­Trường Đại học Bách Khoa Hà Nội

Viện Công nghệ Thông Tin và Truyền Thông

Đồ án Tốt nghiệp Đại học

Xây dựng ứng dụng điểm danh và quản lý công việc trên thiết bị di động

Phan Thanh Hùng

Hà Nội, 08/2018

Trường Đại học Bách Khoa Hà Nội

Viện Công nghệ Thông Tin và Truyền Thông

Đồ án Tốt nghiệp Đại học

Xây dựng ứng dụng điểm danh và quản lý công việc trên thiết bị di động

|  |  |
| --- | --- |
| Sinh viên thực hiện | Phan Thanh Hùng |
| Người hướng dẫn | TS. Vũ Thị Hương Giang |

­­

Hà Nội, 08/2018

# Lời cam kết

Họ và tên sinh viên: Phan Thanh Hùng

Điện thoại liên lạc: 0986680360 Email: hungpt.hust@gmail.com

Lớp: CN – CNTT 02 K59 Hệ đào tạo: Cử nhân công nghệ.

Tôi – *Phan Thanh Hùng* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *Ts. Vũ Thị Hương Giang*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 14 tháng 08 năm 2018*  Tác giả ĐATN  *Phan Thanh Hùng* |

# Lời cảm ơn

Để hoàn thành đồ án tốt nghiệp này đúng thời gian và đạt được kết quả tốt nhất thì bên cạnh sự cố gắng và nỗ lực của bản thân em xin phép được gửi lời cám ơn chân thành nhất đến TS. Vũ Thị Hương Giang người đã đồng hành cùng em trong suốt thời gian đồ án. TS. Vũ Thị Hương Giang là người đã gợi ý đề tài và hướng dẫn cũng như đưa ra các giải pháp và những hướng đi thiết thực nhất cho ứng dụng.

Bên cạnh đó, em xin gửi lời cám ơn đến gia đình cùng toàn thể các bạn đã động viên giúp đỡ, tạo những điều kiện tốt nhất và là chỗ dựa tinh thần để em có thể vượt qua khó khăn và hoàn thành đồ án tốt nghiệp.

Do thời gian làm đồ án có hạn và trình độ còn nhiều hạn chế nên không thể tránh khỏi những thiếu sót. Vậy nên, em rất mong nhận được sự đóng góp của thầy cô, cũng như các bạn sinh viên để bài đồ án này trở nên hoàn thiện hơn.

Em xin chân thành cảm ơn!!!

# Tóm tắt

Hàng năm có hàng triệu người tìm được việc làm, đây là một con số không hề nhỏ, từ đó đặt ra một thách thức cho các các công ty, doanh nghiệp trong việc quản lý, giám sát và đánh giá năng lực của từng nhân viên. Từ đó hàng loạt các giải pháp được ra đời tuy nhiên cùng với sự phát triển không ngừng của khoa học kỹ thuật và đặc biệt của công nghệ thông tin thì các giải pháp được xây dựng bằng trên nền tảng khoa học và công nghệ thông tin được ưa chuộng hơn hẳn bởi tính chính xác, nhanh chóng và tiện lợi của chúng.

Thực vậy, hàng loạt các thiết bị như máy quét vân tay, máy quét thẻ từ, các ứng dụng quản lý công việc như: Trello, Jira, Zoho… đã được ra đời nhằm phục vụ nhu cầu này. Tuy nhiên bên cạnh những mặt tích cực của các hệ thống thì nó cũng tồn tại không ít các nhược điểm. Với các hệ thống lớn cần đầu tư về chi phí ban đầu khá lớn, các máy chấm công truyền thống sẽ gây ách tắc quá tải và mỗi giờ check in và check out. Các ứng dụng quản lý đầu việc chưa thực sự sát sao, đánh giá đúng năng lực của nhân viên. Đồng thời chưa có một hệ thống nào tích hợp giữa việc điểm danh và quản lý công việc với nhau tạo nên một sự cồng kềnh trong việc quản lý hệ thống.

Nhận thấy được những vấn đề này cũng như áp dụng các xu hướng công nghệ hiện tại thì em đã quyết định việc xậy dựng một ứng dụng *“Điểm danh và quản lý công việc”* trên đi động một thiết bị phổ biến và tiện lợi trong hiện tại. Ứng dụng đã giải quyết được vấn đề kết hợp giữ điểm danh và đánh giá năng lực, hiệu suất của nhân viên. Hai module điểm danh và quản lý công việc có thể hoạt động hoàn toàn độc lập cũng như có thể kết nối với nhau để hoạt động một cách hiệu quả nhất. Đồng thời ứng dụng cho phép đồng bộ dữ liệu liên tục giữa client và server giúp cho ứng dụng có thể hoạt động một cách hoàn toàn chính xác nếu một trong hai phía gặp sự cố.

# Abstract

Every year, millions of people find a job, this is not a small number, which poses a challenge for companies and enterprises to manage, monitor and evaluate their individual capabilities. Then, a lot of solutions have been born. However, because of the continuous development of science and technology and especially of information technology, the solutions which base on science and information technology were preferred due to their accuracy, speed and convenience.

Indeed, a wide range of devices such as fingerprint scanners, magnetic card scanners, and task management applications such as Trello, Jira, Zoho, etc… have been created to serve this need. However, besides the positive aspects of the system, it also has many disadvantages. With large systems, the initial investment is quite large. Timekicks will overload and over time check in and check out. The management applications are not very effective, and it also don’t properly assess the capacity of employees. Other sides, there haven’t been no system that integrates attendance and work management together, that create a cumbersome system.

Recognizing these issues as well as adopting current technology trends, I decided to build a *“Attendance and Work Management”* application on a popular and convenient device. The application solved the problem of connecting between appointments and evaluating the capacity and performance of employees. Two modules which are attendance and task management can operate independently as well as connected to each other to operate most effectively. This application also allows continuous data synchronization between the client and the server, so that the application can operate in exactly the right way if either sides crash.

# Mục lục

[Lời cam kết iii](#_Toc522346104)

[Lời cảm ơn iv](#_Toc522346105)

[Tóm tắt v](#_Toc522346106)

[Abstract vi](#_Toc522346107)

[Mục lục vii](#_Toc522346108)

[Danh mục hình vẽ xii](#_Toc522346109)

[Danh mục bảng xv](#_Toc522346110)

[Danh mục các từ viết tắt xvii](#_Toc522346111)

[Danh mục thuật ngữ xviii](#_Toc522346112)

[Chương 1 Giới thiệu đề tài 1](#_Toc522346113)

[1.1 Đặt vấn đề 1](#_Toc522346114)

[1.2 Mục tiêu và phạm vi đề tài 2](#_Toc522346115)

[1.3 Định hướng giải pháp 4](#_Toc522346116)

[1.4 Bố cục đồ án 7](#_Toc522346117)

[Chương 2 Khảo sát và phân tích yêu cầu 9](#_Toc522346118)

[2.1 Khảo sát hiện trạng [1] [2] 9](#_Toc522346119)

[2.2 Tổng quan chức năng 11](#_Toc522346120)

[2.2.1 Biểu đồ use case tổng quan 11](#_Toc522346121)

[2.2.2 Các biểu đồ use case phân rã. 12](#_Toc522346122)

[2.2.3 Quy trình nghiệp vụ 16](#_Toc522346123)

[2.3 Đặc tả chức năng 19](#_Toc522346124)

[2.3.1 Đặc tả use case đăng nhập 19](#_Toc522346125)

[2.3.2 Đặc tả use case CRUD project 21](#_Toc522346126)

[2.3.3 Đặc tả use case CRUD task 23](#_Toc522346127)

[2.3.4 Use case CRUD ticket của nhân viên 26](#_Toc522346128)

[2.4 Yêu cầu phi chức năng 28](#_Toc522346129)

[2.4.1 Yêu cầu chung về chức năng kỹ thuật 28](#_Toc522346130)

[2.4.2 Yêu cầu về giao diện người dùng 28](#_Toc522346131)

[2.4.3 Yêu cầu về bảo mật 29](#_Toc522346132)

[2.4.4 Yêu cầu về hiệu năng sử dụng 29](#_Toc522346133)

[Chương 3 Công nghệ sử dụng 31](#_Toc522346134)

[3.1 . Hệ quản trị CSDL MySQL 31](#_Toc522346135)

[3.1.1 MySQL là gì 31](#_Toc522346136)

[3.1.2 Ưu điểm của hệ quản trị CSDL MySQL 31](#_Toc522346137)

[3.2 Xử lý hình ảnh OpenCV [4] 33](#_Toc522346138)

[3.2.1 Nhận diện khuôn mặt và thư viện OpenCV 33](#_Toc522346139)

[3.2.2 Nhận diện khuôn mặt bằng phương pháp Haar 33](#_Toc522346140)

[3.3 GPS [6] 36](#_Toc522346141)

[3.3.1 Tổng quan về GPS 36](#_Toc522346142)

[3.3.2 Cơ chế hoạt động của GPS 37](#_Toc522346143)

[3.4 Hàm hash SHA256 39](#_Toc522346144)

[Chương 4 Phát triển và triển khai ứng dụng 40](#_Toc522346145)

[4.1 Thiết kế kiến trúc 40](#_Toc522346146)

[4.1.1 Lựa chọn kiến trúc phần mềm 40](#_Toc522346147)

[4.1.2 Thiết kế tổng quan 43](#_Toc522346148)

[4.1.3 Thiết kế chi tiết gói 45](#_Toc522346149)

[4.2 Thiết kế chi tiết 51](#_Toc522346150)

[4.2.1 Thiết kế giao diện 51](#_Toc522346151)

[4.2.2 Thiết kế lớp 55](#_Toc522346152)

[4.2.3 Thiết kế cơ sở dữ liệu 60](#_Toc522346153)

[4.3 Xây dựng ứng dụng 75](#_Toc522346154)

[4.3.1 Thư viện và công cụ sử dụng 75](#_Toc522346155)

[4.3.2 Kết quả đạt được 76](#_Toc522346156)

[4.3.3 Minh hoạ các chức năng chính 77](#_Toc522346157)

[4.4 Kiểm thử 82](#_Toc522346158)

[4.4.1 Load Testing 82](#_Toc522346159)

[4.4.2 Compatibility Testing 83](#_Toc522346160)

[4.4.3 Black Box Testing 84](#_Toc522346161)

[4.4.4 Tổng kết kiểm thử 88](#_Toc522346162)

[4.5 Triển khai 89](#_Toc522346163)

[Chương 5 Các giải pháp và đóng góp nổi bật 90](#_Toc522346164)

[5.1 Điểm danh bằng xác định vị trí và kiểm tra khuôn mặt 90](#_Toc522346165)

[5.1.1 Đặt vấn đề 90](#_Toc522346166)

[5.1.2 Giải pháp đưa ra 90](#_Toc522346167)

[5.1.3 Kết quả đạt được 93](#_Toc522346168)

[5.2 Tích hợp điểm danh và quản lý công việc 93](#_Toc522346169)

[5.2.1 Đặt vấn đề 93](#_Toc522346170)

[5.2.2 Giải pháp đưa ra 93](#_Toc522346171)

[5.2.3 Kết quả đạt được 95](#_Toc522346172)

[5.3 Giảm tải gánh nặng cho server 98](#_Toc522346173)

[5.3.1 Đặt vấn đề 98](#_Toc522346174)

[5.3.2 Giải pháp đưa ra 99](#_Toc522346175)

[5.3.3 Kết quả đạt được 99](#_Toc522346176)

[5.4 Giải quyết vấn đề xung đột tài nguyên 101](#_Toc522346177)

[5.4.1 Đặt vấn đề 101](#_Toc522346178)

[5.4.2 Giải pháp đưa ra 101](#_Toc522346179)

[5.4.3 Kết quả đạt được 103](#_Toc522346180)

[Chương 6 Kết luận và hướng phát triển 104](#_Toc522346181)

[6.1 Kết luận 104](#_Toc522346182)

[6.2 Hướng phát triển 105](#_Toc522346183)

[Tài liệu tham khảo 106](#_Toc522346184)

# Danh mục hình vẽ

[Hình 1.1 Sơ đồ tổng quan chức năng hệ thống. 3](#_Toc522346185)

[Hình 2.1 Sơ đồ use case tổng quan. 11](#_Toc522346186)

[Hình 2.2 Biểu đồ use case đăng nhập 12](#_Toc522346187)

[Hình 2.3 Biểu đồ use case CRUD task 13](#_Toc522346188)

[Hình 2.4 Biểu đồ use case CRUD project 14](#_Toc522346189)

[Hình 2.5Biểu đồ use case thống kê 15](#_Toc522346190)

[Hình 2.6 Sơ đồ hoạt động quản lý đăng nhập 16](#_Toc522346191)

[Hình 2.7 Sơ đồ hoạt động CRUD project 17](#_Toc522346192)

[Hình 2.8 Sơ đồ hoạt động quản lý task 18](#_Toc522346193)

[Hình 3.1 Ví dụ đặc điểm Haar. 34](#_Toc522346194)

[Hình 3.2 Ví dụ về đặc điểm gương mặt. 35](#_Toc522346195)

[Hình 3.3 Ví dụ kết quả nhận diện khuôn mặt. 36](#_Toc522346196)

[Hình 3.4 Ví dụ cách hoạt động GPS. 38](#_Toc522346197)

[Hình 3.5 Ví dụ về kết quả trả về hàm hash SHA256 39](#_Toc522346198)

[Hình 4.1 Mô hình MVC 40](#_Toc522346199)

[Hình 4.2 Mô hình client – server 41](#_Toc522346200)

[Hình 4.3 Biểu đồ phụ thuộc gói 43](#_Toc522346201)

[Hình 4.4 Thiết kế gói activity 45](#_Toc522346202)

[Hình 4.5 Chi tiết gói res 46](#_Toc522346203)

[Hình 4.6 Chi tiết gói fragment 47](#_Toc522346204)

[Hình 4.7 Chi tiết gói view 48](#_Toc522346205)

[Hình 4.8 Chi tiết gói helper 48](#_Toc522346206)

[Hình 4.9 Chi tiết gói adapter 49](#_Toc522346207)

[Hình 4.10 Chi tiết gói utils 50](#_Toc522346208)

[Hình 4.11 Chi tiết gói service 50](#_Toc522346209)

[Hình 4.12 Biểu đồ dịch chuyển màn hình 52](#_Toc522346210)

[Hình 4.13 Form thông báo hệ thống 53](#_Toc522346211)

[Hình 4.14 Màn hình đăng nhập và màn hình kiểm tra khuôn mặt 53](#_Toc522346212)

[Hình 4.15 Màn hình quản lý công việc và màn hình thống kê 54](#_Toc522346213)

[Hình 4.16 Màn hình quản lý task 54](#_Toc522346214)

[Hình 4.17 Biểu đồ lớp chi tiết LoginActivity 55](#_Toc522346215)

[Hình 4.18 Sơ đồ lớp LocationHelper 57](#_Toc522346216)

[Hình 4.19 Biểu đồ giao tiếp quản lý task 59](#_Toc522346217)

[Hình 4.20Biểu đồ giao tiếp thống kê 59](#_Toc522346218)

[Hình 4.21 Sơ đồ thực thể liên kết (ER) 60](#_Toc522346219)

[Hình 4.22 Mô hình hình CSDL hệ thống 61](#_Toc522346220)

[Hình 4.23 Màn hình Splash và mà hình đăng nhập 77](#_Toc522346221)

[Hình 4.24 Màn hình kiểm tra vân tay và kiểm tra khuôn mặt 78](#_Toc522346222)

[Hình 4.25 Màn hình trang chủ và navigation menu 79](#_Toc522346223)

[Hình 4.26 Màn hình quản lý dự án và màn hình thông báo 80](#_Toc522346224)

[Hình 4.27 Màn hình thống kê và màn hình chỉnh sửa thông tin dự án 81](#_Toc522346225)

# Danh mục bảng

[Bảng 2.1 Bảng đặc tả chức năng đăng nhập 19](#_Toc522346226)

[Bảng 2.2 Dữ liệu đầu vào gồm các trường dữ liệu của chức năng đăng nhập. 21](#_Toc522346227)

[Bảng 2.3 Bảng đặc tả chức năng quản lý dự án 21](#_Toc522346228)

[Bảng 2.4 Dữ liệu đầu vào gồm các trường dữ liệu của chức năng quản lý dự án 22](#_Toc522346229)

[Bảng 2.5 Bảng đặc tả chức năng quản lý task trong dự án 23](#_Toc522346230)

[Bảng 2.6 Bảng dữ liệu đầu vào của chức năng quản lý task 24](#_Toc522346231)

[Bảng 2.7 Bảng dữ liệu đầu đầu ra sau khi thực hiện tìm kiếm task. 25](#_Toc522346232)

[Bảng 2.8 Bảng đặc tả chức năng quản lý ticket của nhân viên 26](#_Toc522346233)

[Bảng 2.9 Bảng dữ liệu đầu vào của chức năng quản lý ticket 27](#_Toc522346234)

[Bảng 2.10 Bảng dữ liệu đầu đầu ra sau khi thực hiện chức năng tìm kiếm ticket. 27](#_Toc522346235)

[Bảng 4.1 Bảng đặc tả các phương thức trong lớp LoginActivity 56](#_Toc522346236)

[Bảng 4.2 Bảng đặc tả các phương thức trong lớp LocationHelper 58](#_Toc522346237)

[Bảng 4.3 Thiết kế chi tiết bảng com\_mst: 62](#_Toc522346238)

[Bảng 4.4 Thiết kế chi tiết bảng Task\_master 63](#_Toc522346239)

[Bảng 4.5 Thiết kế chi tiết bảng Project\_master 64](#_Toc522346240)

[Bảng 4.6 Thiết kế bảng location\_master 66](#_Toc522346241)

[Bảng 4.7 Thiết kế bảng time\_working 67](#_Toc522346242)

[Bảng 4.8 Thiết kế bảng user\_master 68](#_Toc522346243)

[Bảng 4.9 Thiết kế bảng cmt\_tbl 70](#_Toc522346244)

[Bảng 4.10 Thiết kế bảng notification\_tbl 71](#_Toc522346245)

[Bảng 4.11 Thiết kế bảng user\_project\_tbl 72](#_Toc522346246)

[Bảng 4.12 Thiết kế bảng role\_mst 72](#_Toc522346247)

[Bảng 4.13 Thiết kế bảng attendance\_tbl 73](#_Toc522346248)

[Bảng 4.14 Danh sách thư viện và công cụ sử dụng 75](#_Toc522346249)

[Bảng 4.15 Thống kê thông tin ứng dụng 76](#_Toc522346250)

[Bảng 4.16 Thống kê kiểm thử tải 82](#_Toc522346251)

[Bảng 4.17 Thống kê kiểm thử tương thích 83](#_Toc522346252)

[Bảng 4.18 Bảng kiểm thử chức năng điểm danh 84](#_Toc522346253)

[Bảng 4.19 Bảng kiểm thử chức năng quản lý công việc 86](#_Toc522346254)

[Bảng 4.20 Bảng kiểm thử chức năng dự án 87](#_Toc522346255)

[Bảng 4.21 Bảng kế hoạch triển khai 89](#_Toc522346256)

[Bảng 5.1 Bảng chức năng hệ thống 95](#_Toc522346257)

# Danh mục các từ viết tắt

|  |  |
| --- | --- |
| API | Application Programming Interface  Giao diện lập trình ứng dụng |
| **CNTT** | Công nghệ thông tin |
| **ĐATN** | Đồ án tốt nghiệp |
| **SV** | Sinh viên |
| **PM** | Project Manager  (Quản lý dự án) |
| **HR** | Human Resource  Quản lý nhân sự |
| **CDSL** | Cơ sở dữ liệu |
| **SQL** | Structured Query Language  Ngôn ngữ truy vấn cấu trúc |

# Danh mục thuật ngữ

|  |  |
| --- | --- |
| Browser | Trình duyệt |
| **Cache memory** | Bộ nhớ đệm |
| **E-commerce** | Thương mại điện tử |
| **Developer** | Nhà phát triển |
| **Interpreter** | Trình thông dịch |
| **Compiler** | Trình biên dịch |
| **Pakage** | Gói |
| **Request** | Yêu cầu |
| **Response** | Phản hồi |
| **Server** | Máy chủ |
| **Client** | Máy khách |

# Giới thiệu đề tài

## Đặt vấn đề

Vấn đề điểm danh và đánh giá năng lực của nhân viên là một vấn đề chưa bao giờ hết nóng, mặc dù đã có rất nhiều những giải pháp những công nghệ được đưa ra. Tuy nhiên chưa có một công nghệ, giải pháp nào đã thực sự giải quyết được hết những vấn đề mà thực tế đặt ra dẫn đến nhiều hệ lụy.

Thực tế, hiện nay có rất nhiều các công ty start up, các công ty có quy mô vừa và nhỏ, họ không đủ điều kiện để đầu tư đầy đủ cho một hệ thống điểm danh và quản lý công việc một cách chuyên nghiệp. Thay vào đó họ thường sử dụng các công nghệ, ứng dụng với chi phí thấp hoặc được cung cấp miễn phí trên thị trường. Tuy nhiên, do bị hạn chế bởi chi phí nên các công cụ được cung cấp thường chưa thực sự đáp ứng được các nhu cầu thực sự mà họ đã mong muốn. Từ việc không theo dõi được sát sao quá trình làm việc của nhân viên, cho đến việc các công cụ điểm danh hiện tại khá dễ để gian lận, cũng như việc đánh giá sai năng lực của nhân viên cũng tạo nên những tổn thất, thất thoát đáng kể cho công ty và cả nhân viên.

Vì vậy, bài toán đặt ra ở đây là phải xây dựng được một hệ thống, ứng dụng giải quyết được tối đa các vấn đề đã nêu trên. Hệ thống sẽ hướng đến các công ty có quy mô vừa và nhỏ những nơi mà chưa thực sự có giải pháp thỏa đáng nhất cho việc quản lý và đánh giá nhân viên.

Bài toán được giải quyết sẽ giúp cho các chủ sở hữu công ty thực sự yên tâm trong việc trả lương và bên cạnh đó nhân viên cũng cảm thấy năng lực của bản thân được đánh giá một cách chính xác, từ đó so sánh được năng lực mình bỏ ra với mức lương mình nhận được. Bên cạnh đó các công ty sẽ tiết kiệm được một nguồn chi phi đầu tư ban đầu cho các hệ thống điểm danh và quản lý công việc, giảm thiểu được tối đa các công việc của quản lý nhân viên, thống kê, tính lương hàng tháng.

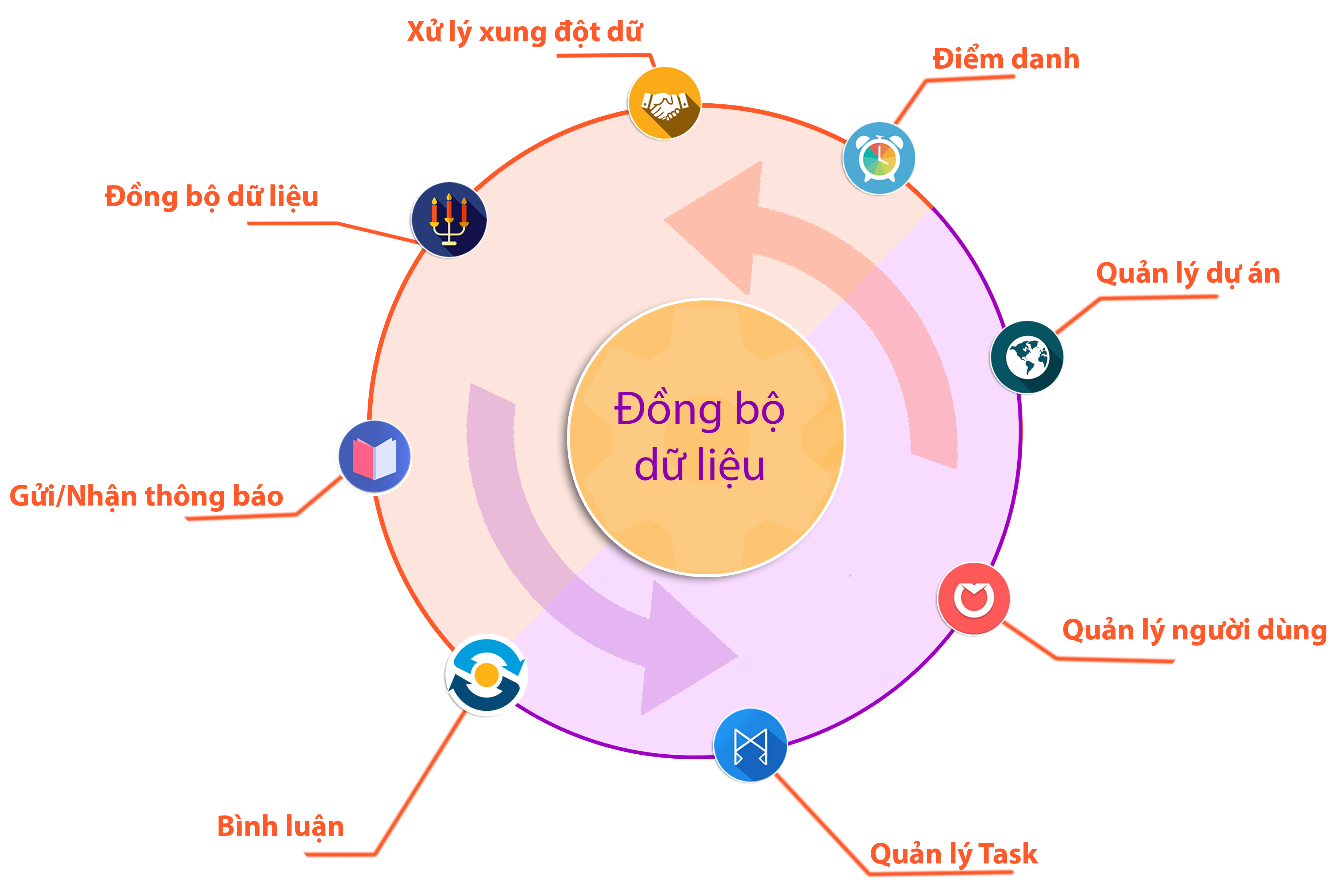
## Mục tiêu và phạm vi đề tài

Để định hướng cho đề tài này, em đã thực hiện nghiên cứu tổng quan một số hệ thống đã được triển khai trên thị trường bao gồm các phần mềm miễn phí cũng như trả phí (chi phí thấp) để đánh giá được các ưu điểm và mặt tồn tại mà chúng mắc phải. Các hệ thống đã nghiên cứu bao gồm hệ thống Trello và Zoho.

Trello là một công cụ để phối hợp công việc hiệu quả giúp cho mọi người trong nhóm chỉ cần nhìn qua là biết được có những đầu việc nào, ai đang làm gì, và làm đến giai đoạn nào rồi. Với phiên bản miễn phí người dùng có đã có thể trải nghiệm đến hơn 70% các chức năng về quản lý công việc mà hệ thống cung cấp. Ưu điểm lớn nhất của Trello bởi tính đơn giản dễ sử dụng và linh hoạt của nó. Vì vậy, Trello được khá nhiều các công ty ưa chuộng và sử dụng. Tuy nhiên hệ thống Trello cũng gặp phải một số vấn đề như: mặc dù Trello cũng có thể điểm danh tuy nhiên hệ thống lại không thể kiểm soát chính xác vị trí mà người dùng điểm danh, bất cứ ai cũng có thể điểm danh hộ từ đó dẫn đến những tình trạng gian lận, điểm danh chống đối. Module quản lý công việc chưa thực sự đáp ứng hết các nhu cầu của người dùng và chặt chẽ trong quá trình theo dõi và đánh giá công việc. Người dùng có thể thực hiện nhiều công việc cùng một lúc mà thực tế thì chuyện này hoàn toàn không thể xảy ra.

Zoho là một công cụ đa năng trong việc hỗ trợ quản lý danh nghiệp với sự tích hợp của hơn 40 ứng dụng khác nhau. Zoho cung cấp rất nhiều các giải pháp về các vấn đề khác nhau, tuy nhiên ở báo cáo xin được phép chỉ đánh giá các ứng dụng giải pháp trong việc quản lý nhân viên và dự án. Zoho cung cấp khá đầy đủ và chuyên nghiệp tính năng quản lý công việc, khi bắt đầu một công việc hệ thống sẽ tự động đếm giờ người dùng hoàn toàn có thể theo dõi được tiến độ lảm việc của mình. Tuy nhiên, cũng như Trello, Zoho cũng gặp những vẫn đề tương tự trong điểm danh, không thể tránh được gian lận, và điểm danh một cách chính xác.

Từ những vấn đề mắc phải của các ứng dụng trên thì hệ thống mới được xậy dựng phải khắc phục, cũng như đưa ra các hướng giải quyết cho chúng. Hệ thống quản lý điểm danh và giao việc được xây dựng nhằm đưa ra giải pháp thiết thực, hiệu quả nhất cho các công ty với quy mô vừa và nhỏ nhằm quản lý nhân viên và dự án đồng thời tiết kiệm tối đa nhất chi phí đầu tư ban đầu.



Hình 1.1 Sơ đồ tổng quan chức năng hệ thống.

Mục tiêu của hệ thống này là xây dựng một ứng dụng nhằm giải quyết vấn đề điểm danh và quản lý công việc. Việc điểm danh sẽ được thực hiện qua di động của từng nhân viên để tiết kiệm nhất các chi phí đầu tư ban đầu. Việc điểm danh và quản lý công việc được tích hợp với nhau, tạo nên độ xác thực và chính xác cao cho việc đánh giá năng lực của nhân viên. Bên cạnh đó các module này hoàn toàn có thể hoạt động độc lập cũng có thề hoạt động một cách riêng lẻ cho phép người dùng lựa chọn sử dụng hoặc tích hợp với các hệ thống có sẵn, đang được sử dụng mà không cần chuyển đổi công nghệ.

## Định hướng giải pháp

Để giải quyết vấn đề điểm danh một cách chính xác nhất, hệ thống điểm danh được tích hợp trên thiết bị android (API ≥ 18) của nhân viên nhằm tiết kiệm tối đa chi phí đầu tư ban đầu cho thiết bị. Chức năng điểm danh hoạt động dựa trên tính toán GPS vị trí hiện tại của thiết bị android và vị trí thực tế của địa điểm điểm danh. Bên cạnh đó để có thể xác thực chính xác nhất thì hệ thống còn kiểm tra địa chỉ IP của Wifi được kết nối, địa chỉ MAC của máy và yêu cầu chụp lại khuôn mặt với chức nhận diện khuôn mặt để đảm bảo kết quả điểm danh, tránh gian lận.

Chức năng quản lý công việc cũng được xây dựng theo một cách dễ hiểu nhất để người dùng có thể dễ dàng sử dụng. Bao gồm tạo project, đặc tả, thêm thành viên, tạo task, astimate dealine cho từng task. Khi hết giờ làm việc task đang hoạt động sẽ được chuyển trạng thái sang tạm dừng. Người dùng không thể đồng thời thực hiện nhiều công việc một lúc mà chỉ có thể làm một công việc, nếu vi phạm hệ thống sẽ đưa ra cảnh báo và yêu cầu người dùng đóng lại một task.

Hệ thống cho phép thống kê chặt chẽ giữa phần điểm danh và quản lý công việc, từ đó đưa ra được hiệu suất làm việc của từng cá nhân một cách chính xác nhất. Bên cạnh đó người quản trị hệ thống có thể thiết lập các công thức tính lương để hệ thống tự đưa ra kết quả lương hàng tháng của từng nhân viên mà không cần tốn quá nhiều thời gian cho việc tính lương như trước.

Hệ thống được xây dựng trên mô hình client – server với hệ CSDL phân tán với cơ chế đồng bộ dữ liệu liên tục. Mỗi Client có một thư mục đồng bộ. Dữ liệu trong thư mục đó được đưa lên Server. Người dùng có thể thêm/xóa các thư mục/ tệp tin trong thư mục đồng bộ đó. Server thực hiện việc đồng bộ hóa giữa các Client dựa trên trạng thái được cài đặt và thời gian thay đổi dữ liệu, client tự động download và upload dữ liệu. Việc đồng bộ dữ liệu này sẽ được thực hiện tại client là các thiết bị android của người dùng dưới dạng cách service chạy ngầm bên dưới nền ứng dụng vì vậy hoàn toàn không ảnh hưởng đến quá trình thao tác của người dùng.

Bảng 1.1 Bảng đánh giá hệ thống xây dựng với các hệ thống khảo sát

| STT | Tiêu chí | Trello | Zoho | Đánh giá | Giải thích |
| --- | --- | --- | --- | --- | --- |
| 1 | Chức năng: Điểm danh |  | x | A | Hệ thống xây dựng điểm danh với độ chính xác cao do sử dụng điểm danh bằng địa chỉ MAC, wifi và vị trí. |
| 3 | Chức năng: Theo dõi số giờ làm việc |  | x | A | Do độ chính xác cao của hệ thống điểm danh |
| 4 | Chức năng: Quản lý người dùng |  | x | D | Cả hai đều xây dựng đầy đủ các chức năng về quản lý người dùng |
| 5 | Chức năng: Tích hợp quản lý người dùng với quản lý project |  | x | C | Việc kết hợp của hai module này được Zoho xây dựng hoàn thiện và đầy đủ hơn. Cho phép người dùng lên lịch trình kế hoạch thực hiện, dự trù kinh phí, chi phí đi lại, quản lý tài sản. |
| 6 | Chức năng: Tạo project mới | x | x | D |  |
| 7 | Chức năng: Tạo task mới | x | x | D |  |
| 8 | Chức năng: Tính deadline cho từng task | x | x | A | Hệ thống mới cho phép estimate cho từng task tự ngừng đếm thời gian khi hết giờ làm việc……. |
| 9 | Chức năng: Bình luận đính kèm file trong các task | x | x | B | Hệ thống trello làm việc hiệu quả nhất, trực quan, và tích hợp được với nhiều công cụ khác nhau. |
| 10 | Tính thân thiện với người dùng (Dễ sử dụng, dễ học) |  |  | B | Trello đã thực hiện việc tối ưu hóa giao diện, đề cao tính đơn giản dễ sử dụng, hầu như không tốn thời gian học. |
| 11 | Tính đầy đủ |  |  | C | Hệ thố Zoho tích đã tích hợp hơn 40 ứng dụng khác nhau trong việc quản lý công ty. |
| 12 | Tính phổ biến |  |  | B | Trello có tính phổ biến cao nhất trong 3 hệ thống trên, vì hầu hết các chức năng được sử dụng miễn phí nên được các công ty ưa dùng và sử dụng. |
| 13 | Khả năng triển khai | x | x | D |  |

Ghi chú:

* X là có chức năng.
* A là hệ thống xây dựng tối ưu hơn
* B là hệ thống Trello xây dựng tối ưu hơn
* C là hệ thống Zoho tối ưu hơn
* D là độ tối ưu tương đương

## Bố cục đồ án

Báo cáo đồ án này được chia làm sáu chương chính và nội dung của các chương được tóm tắt lần lươt như sau:

Chương 1 là chương giới thiệu đề tài, tại đây nêu ra các vấn tồn tại trong thực tế mà các công ty doanh nghiệp gặp phải, tìm hiểu các giải pháp và công nghệ có sẵn trên thị trường. Trên cơ sở đó tìm hiểu chỉ rõ các yếu tố đã được giải quyết các vấn đề còn tồn đọng, tại đồ án này đã giải quyết và triển khai được những vấn đề gì. Đồng thời so sánh hệ thống xây dựng với các hệ thống có sẵn trên thị trường.

Tại chương 2 em đã khảo sát rõ ràng hơn các vấn đề mắc phải tại công ty, doanh nghiệp mà đề tài hướng tới, cũng như các hệ thống tương tự đã được triển khai xây dựng. Từ đó chỉ rõ các đối tượng người dùng cũng như các chức năng tổng quan cần xây dựng của hệ thống. Phần phân tích nêu rõ những ca sử dụng dành cho từng đối tượng trong hệ thống, miêu tả rõ quy trình nghiệp vụ và chi tiết những chức năng quan trọng, những yêu cầu chức năng và yêu cầu phi chức năng của hệ thống.

Với những khảo sát và đánh giá đã thu thập được ở chương 1 và chương 2 tại chương 3 em đã đưa ra nhưng công nghệ sử dụng trong việc xử lý và giải quyết các vấn đề đã nêu ra. Đầu tiên là việc lựa chọn CSDL cho hệ thống, tại đồ án này em đã sự dụng hệ quản trị CSDL MySQL bởi các lợi ích và tính ứng dụng của nó. Tiếp đến em tích hợp sử dụng hệ thống định vị vị trí GPS và nhận diện khuôn mặt OpenCV cho chức năng điểm danh để tạo nên độ chính xác cao nhất. Cuối cùng là hàm hash SHA256 để mã hóa các dữ liệu đảm bảo tính an toàn và bảo mật cao nhất cho hệ thống.

Ở Chương 4 của báo cáo đồ án này, sẽ trình bày về quá trình phát triển, kiểm thử và triển khai ứng dụng. Trong phần phát triển ứng dụng, nêu rõ kiến trúc sử dụng để xây dựng hệ thống, thiết kế lớp, thiết kế gói, thiết kế cơ sở dữ liệu thông số, thiết kế giao diện hệ thống và trình bày về thư viện và công cụ đã sử dụng cho việc phát triển hệ thống. Trong phần kiểm thử, trình bày những trường hợp kiểm thử và kỹ thuật kiểm thử đã sử dụng. Cuối cùng là báo cáo về việc kế hoạch triển khai hệ thống trên môi trường thực tế.

Trong chương 5 của báo cáo đồ án này, em xin trình bày đóng góp và giải pháp nổi bật của mình khi thực hiện phát triển xong ứng dụng: bao gồm các đóng góp của mình trong việc xây dựng hệ thống điểm danh và quản lý công việc như sử dụng GPS với độ chính xác cao cũng như tích hợp nhận diện khuôn mặt. Đồng thời là vấn đề giảm tải gánh năng cho server cũng như xử lý các xung đột dữ liệu.

Cuối cùng, Chương 6 là kết luận về sản phẩm của em với những thiếu sót trong việc xây dựng hệ thống, những kinh nghiệm đã học hỏi được trong quá trình xây dựng hệ thống và hướng phát triển sau này để hoàn thiện tất cả chức năng mong muốn của hệ thống.

# ­­Khảo sát và phân tích yêu cầu

## Khảo sát hiện trạng [1] [2]

Hiện nay các công ty và doanh nghiệp thường tách rời việc điểm danh và quản lý công việc. Với điểm danh các công ty, doanh nghiệp thường áp dụng các công nghệ như máy quét vân tay, máy quét thẻ để điểm danh. Tuy nhiên cách này cũng gặp phải khá nhiều các vấn đề mắc phải. Đầu tiên các phương thức điểm danh này có thể bị gian lận, với các công ty lớn đông nhân viên có thể gây nên sự ách tắc trong các giờ check in và check out. Với các công ty nhỏ, strart up thì việc đầu tư cho các thiết bị này khá tốn kém, và hiệu quả chưa cao.

Việc quản lý công việc của của nhân viên sẽ được tiến hành bởi các ứng dụng quản lý công việc được cung cấp trên thị trường với phiên bản miễn phí hoặc trả phí. Đa số các công ty đều chọn các phiên bản miễn phí hoặc trả thấp. Mặc dù chúng đã đáp ứng phần nào nhu cầu của người dùng, nhưng chưa thực sự đánh giá sát sao và chính xác.

Em đã khảo sát hai hệ thống hiện có trên thị trường như đã giới thiệu ở mục 1.2 là hệ thống miễn phí Trello và có trả phí Zoho. Cơ bản hai hệ thống này đều có những ưu và nhược điểm riêng. Về ưu điểm Trello như trong các ghi chú mà mọi người ai cũng đều sử dụng qua để ghi chú lại những việc mình cần làm. Trello không có quá nhiều khái niệm phức tạp, nhìn vào có thể hiểu liền và sử dụng ngay được. Tính đơn giản của Trello còn nằm ở chỗ những thao tác trong đó cũng được đơn giản hóa cực kỳ.

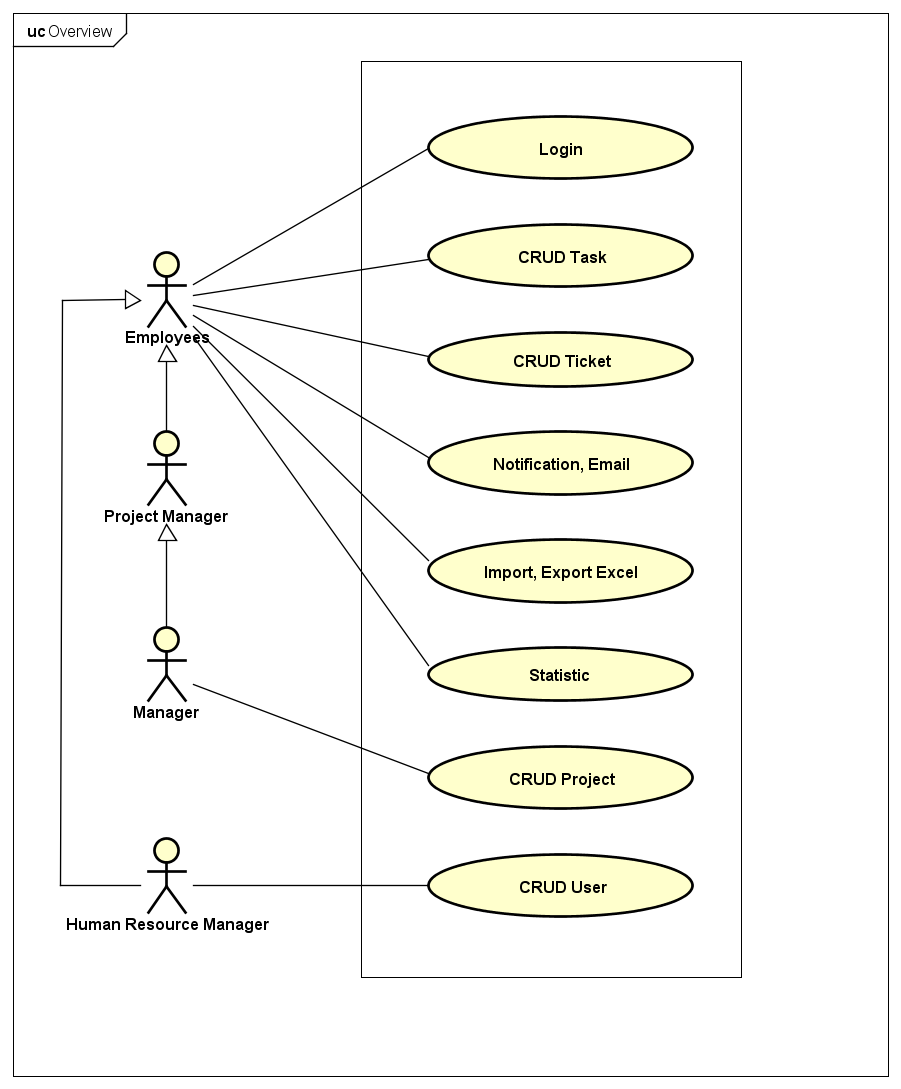
Khác với Trello hệ thống Zoho lại được đánh giá cao bởi tính chuyên nghiệp và cung cấp đầy đủ các tính năng mà người dùng mong muốn. Người dùng có thể lựa chọn sử dụng một hoặc tổ hợp một vài chức năng mà không nhất thiết phải trả phí cho cả một hệ thống do các module mà hệ thống cung cấp hoàn toàn có thể hoạt động độc lập.

Về nhược điểm cả hai hệ thống đều có những nhược điểm chung cũng như riêng. Cả hai hệ thống đều chưa đánh giá được chính xác nhất năng suất làm việc của nhân viên do không theo dõi được vị trí làm việc, và quản lý được số task làm được trong cùng một khoảng thời gian, cũng như số giờ làm việc thực tế. Ở Zoho tuy có đếm giờ nhưng lại không tự ngừng đếm khi hết giờ làm việc. Còn ở Trello thì người dùng phải tự ước lượng và tính toán thời gian, không tự đếm lùi khiến người dùng không ước lượng được phần thời gian đã hao phí và trôi qua, chỉ đến khi gần hết thời gian.

Hơn nữa các hệ thống trên thị trường không riêng gì Trello hay Zoho đều chưa thực sự liên kết được việc điểm danh và tiến độ làm việc cũng như năng lực của nhân viên để cho ra một mức lương hợp lý nhất.

## Tổng quan chức năng

### Biểu đồ use case tổng quan

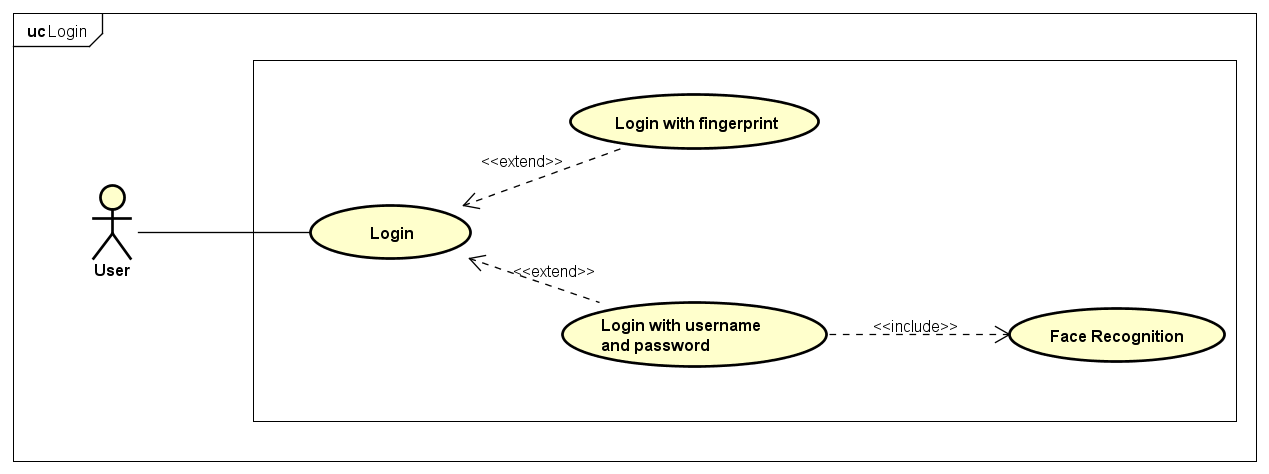


Hình 2.1 Sơ đồ use case tổng quan.

Hình 2.1 Sơ đồ use case tổng quan. Mô tả các chức năng tổng quan, và người dùng của hệ thống. Hệ thống gồm bốn tác nhân chính bao gồm: nhân viên, quản lý nhân sự, quản lý dự án và quản lý. Để sử dụng hệ thống thì người dùng bắt buộc phải sử dụng chức năng đăng nhập. Sau khi đăng nhập thì tùy vào phân quyền khác nhau mà người dùng được sử dụng các chức năng khác nhau.

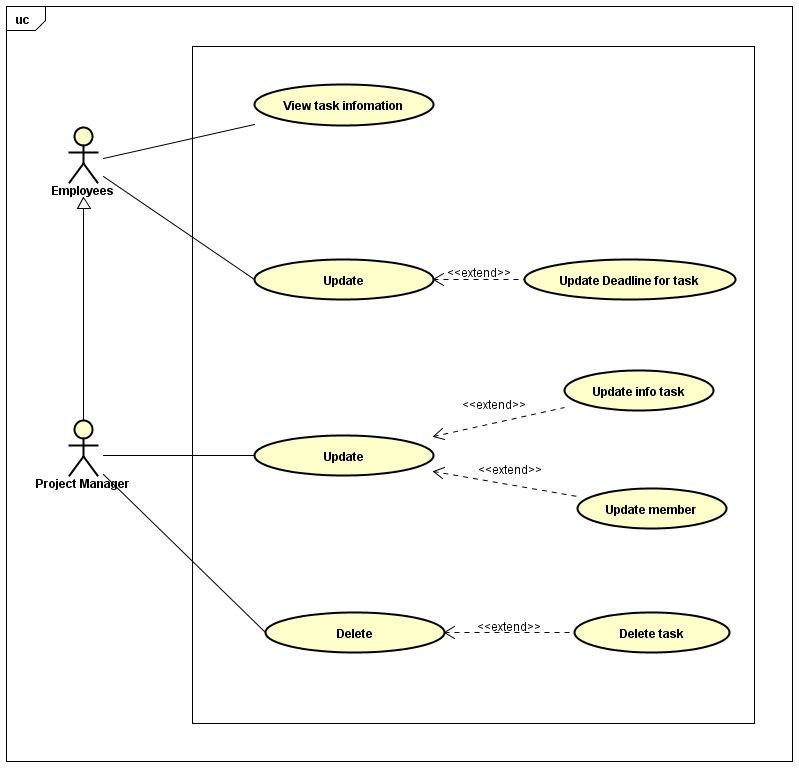
### Các biểu đồ use case phân rã.

#### Biểu đồ use case đăng nhập hệ thống

Hình 2.2 Biểu đồ use case đăng nhập

Hình 2.2 Biểu đồ use case đăng nhập mô tả chức năng đăng nhập của hệ thống. Người dùng bắt buộc phải đăng nhập để bắt đầu điểm danh và sử dụng chức năng hệ thống. Người dùng được phép lựa chọn đăng nhập bằng tài khoản và mật khẩu thông thường hoặc sử dụng vân tay để đăng nhâp. Sau khi xác nhận đúng tài khoản người dùng sẽ được yêu cầu chụp lại khuôn mặt thông qua hệ thống nhận diện khuôn mặt. Chỉ khi thực hiện đầy đủ các bước trên thì mới có thể chính thức điểm danh vào hệ thống.

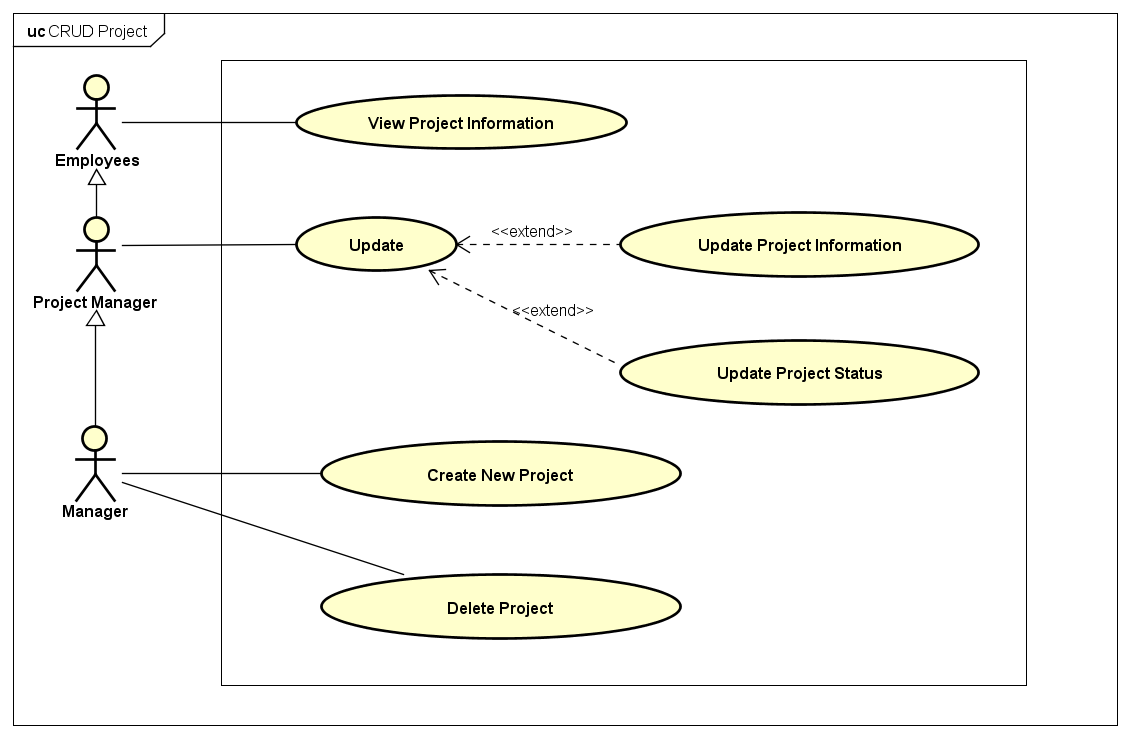
#### Biểu đồ use case CRUD task



Hình 2.3 Biểu đồ use case CRUD task

Hình 2.3 Biểu đồ use case CRUD task mô tả chức năng quản lý các task của một dự án. Sau khi đăng nhập, người dùng lựa chọn chức năng quản lý công việc để bắt đầu. Nhân viên được phép xem thông tin các task và sửa đổi lại thời gian kết thúc công việc trong các task đã được giao. Với quản lý dự án, sau khi đăng nhập có quyền cao nhất trong việc quản lý các task công việc. Quản lý có toàn quyền thêm, sửa, xóa các task khi thấy cần thiết.

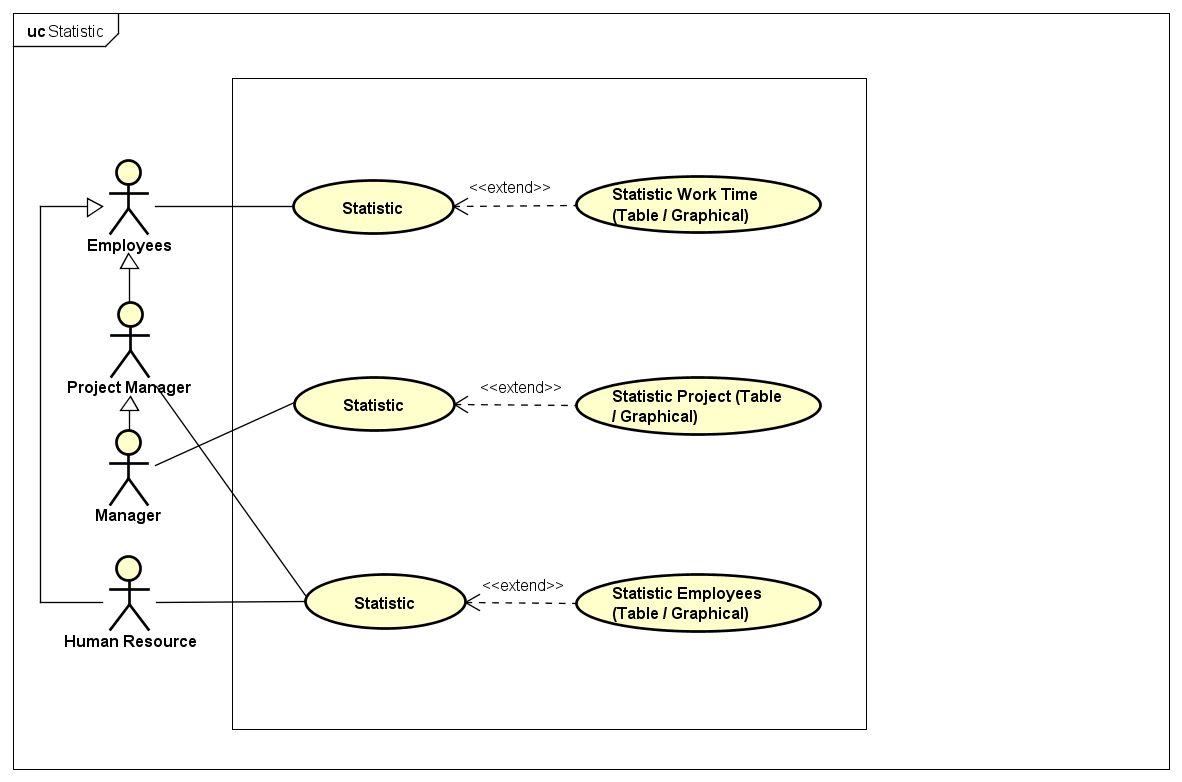
#### Biểu đồ use case CRUD project



Hình 2.4 Biểu đồ use case CRUD project

Hình 2.4 Biểu đồ use case CRUD project mô tả quá trình tạo và quản lý một dự án. Sau khi đăng nhập và hệ thống đối với phân quyền khác nhau thì chức năng người dùng cũng khác nhau. Quản lý sẽ nhận nhiệm vụ tạo hoặc xóa các project. Đối với quản lý dự án sau khi quản lý cập nhập project thì PM sẽ vào xem thông tin và cập nhật lại thông tin thêm thành viên vào dự án. Các thành viên được thêm vào chỉ được xem thông tin trong dự án mà không có quyền chỉnh sửa.

#### Biểu đồ use case thống kê

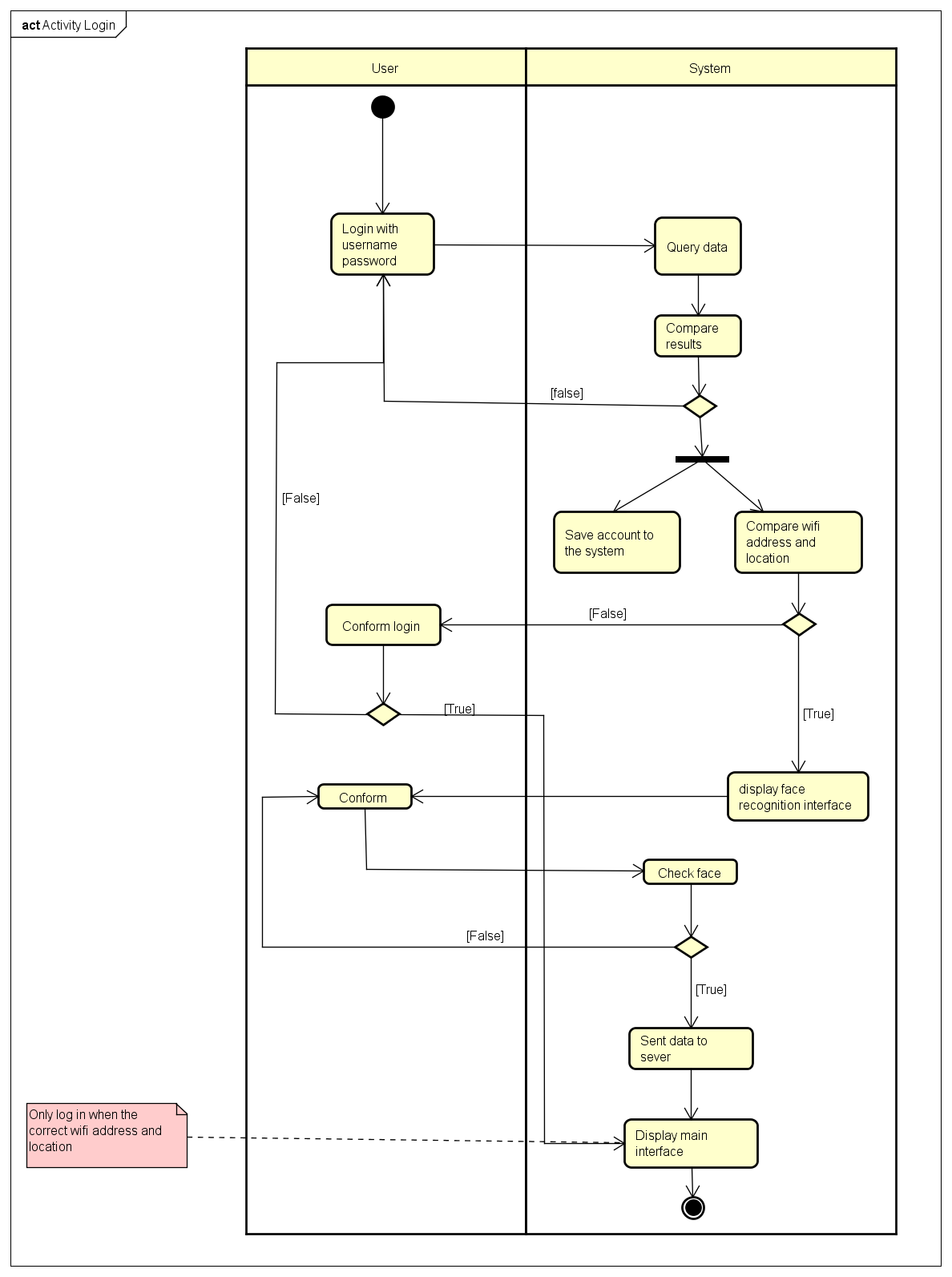


Hình 2.5Biểu đồ use case thống kê

Hình 2.5Biểu đồ use case thống kê mô tả chức năng thống kê của hệ thống. Sau khi đăng nhập vào hệ thống người dùng lựa chọn chức năng thống kê để thống kê. Với người dùng là nhân viên có thể lựa chọn chứ năng thống kê giờ làm việc. Với người dùng là quản lý nhân sự bên cạnh thống kê giờ làm việc sẽ được thống kế thêm người dùng. Quản lý dự án sẽ có quyền thống kê tương tự như quản lý nhân sự nhưng phạm vi thống kê người dùng sẽ khác nhau. Quản lý sẽ có quyền thống kê cao nhất, bên cạnh thống kê công việc và thống kê người dùng quản lý còn có quyền thống kê project.

### Quy trình nghiệp vụ

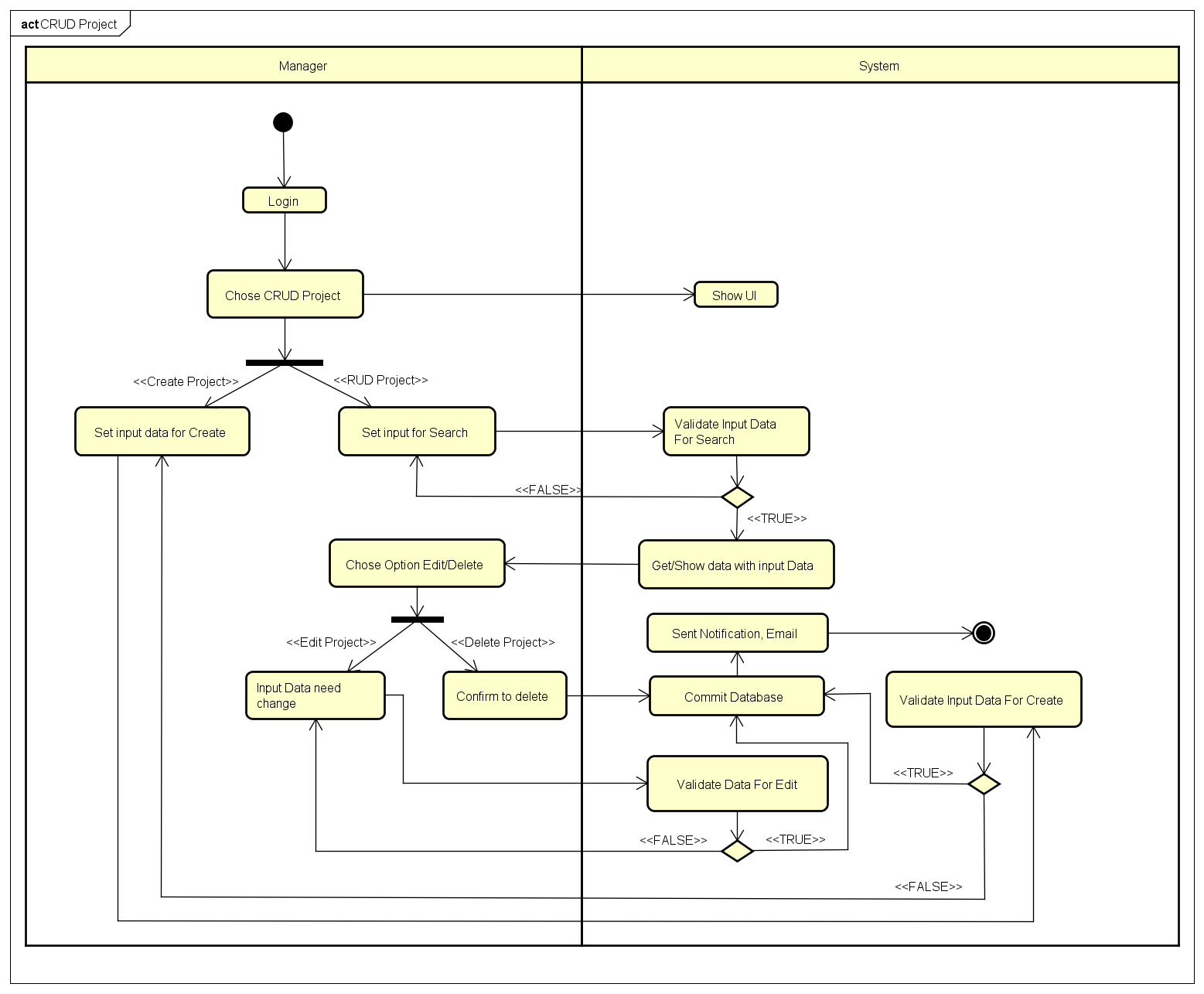
Hệ thống cần xây dựng gồm một số quy trình nhiệm vụ chính sau. Đầu tiên là quy trình đăng nhập và điểm danh.



Hình 2.6 Sơ đồ hoạt động quản lý đăng nhập

Để sử dụng ứng dụng và bắt đầu điểm danh người dùng bắt buộc phải đăng nhập vào hệ thống. Người dùng phải đến vị trí được yêu cầu điểm danh và kết nối với wifi của công ty và sử dụng đúng thiết bị được cấp phép để đăng nhập. Người dùng được lựa chọn sử dụng đăng nhập và vân tay hoặc đăng nhập với tài khoản và mật khẩu. Sau khi đăng nhập hệ thống sẽ yêu cầu người dùng chụp lại khuôn mặt thông qua hệ thống nhận diện khuôn mặt. Nếu ngoài giờ làm việc hoặc không trong vị trí điểm danh người dùng vẫn có quyền đăng nhập tuy nhiên sẽ không được tính điểm danh và bị hạn chế một số chức năng trong hệ thống.

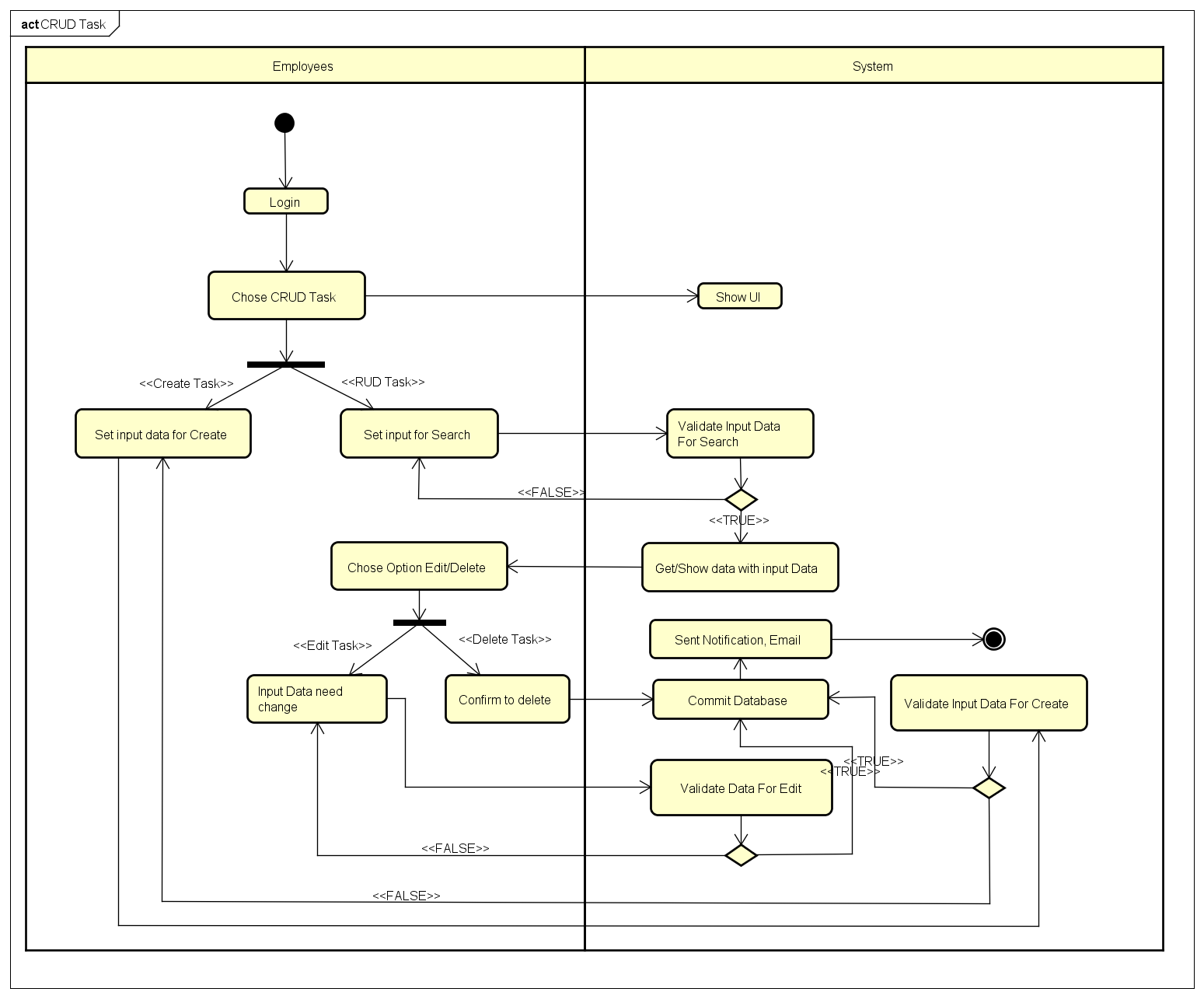
Tiếp theo là quy trình CRUD project.



Hình 2.7 Sơ đồ hoạt động CRUD project

Hình 2.7 Sơ đồ hoạt động CRUD project mô tả chi tiết quy trình tạo và quản lý các dự án của hệ thống kể từ bước đầu đăng nhập, tạo dự án cho đến khi dự án kết thúc.

Quy trình quản lý task



Hình 2.8 Sơ đồ hoạt động quản lý task

Hình 2.8 Sơ đồ hoạt động quản lý task mô tả chi tiết quy trình thực hiện và quả lý dự án bằng các task của nhân viên trong hệ thống từ việc tạo task, cập nhật task từ khi dự án bắt đầu đến khi dự án kết thúc. Khi được thêm vào 1 dự án, nhân viên sẽ đọc phần mô tả dự án để biết công việc của mình, từ đó tự tạo các task công việc, dự trù thời gian hợp lý, và cập nhật lại thông tin của các task nếu thấy cần thiết.

## Đặc tả chức năng

### Đặc tả use case đăng nhập

Bảng 2.1 Bảng đặc tả chức năng đăng nhập

| Mã Usecase | UC0001 | Tên Usecase | Login |
| --- | --- | --- | --- |
| Tác nhân hệ thống | Tất cả các tác nhân trong hệ thống đều có thể sử dụng chức năng này. | | |
| Tiền điều kiện | Cài đặt Mobile App | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện | Hành động | | 1. | Tất cả người dùng | Chọn chức năng đăng nhập | | 2. | Tất cả người dùng | Chọn kiểu đăng nhập | | 3. | Tất cả người dùng | Đăng nhập | | 3.1. | Tất cả người dùng | Đăng nhập bằng vân tay | | 3.2. | Tất cả người dùng | Đăng nhập bằng tài khoản | | 3.2.1 | Tất cả người dùng | Nhập tài khoản và mật khẩu | | 3.2.2 | Tất cả người dùng | Đăng nhập thành công, chụp ảnh khuôn mặt lưu trữ vào cơ sở dữ liệu | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện | Hành động | | 3.1a | Hệ thống | Thông báo vân tay đăng nhập không trùng | | 3.1b | Hệ thống | Thông báo sai vị trí đăng nhập | | 3.1c | Hệ thống | Thông báo sai địa chỉ ip wifi | | 3.1d | Hệ thống | Không đúng thiết bị được cấp phép điểm danh | | 3.2.1a | Hệ thống | Thông báo sai Tài khoản, mật khẩu | | 3.1b | Hệ thống | Không thể đăng nhập vân tay do chưa đăng nhập lần đầu bằng use name và password | | 3.2.2a | Hệ thống | Không nhận diện được khuôn mặt | | 3.2.2b | Hệ thống | Thông báo ra khỏi vị trí đăng nhập. Logout khỏi hệ thống. | | 3.2.2c | Hệ thống | Thông báo hết giờ làm. Logout khỏi hệ thống. | | | |
| Hậu điều kiện | Duy trì đăng nhập trong vị trí điểm danh và kết nối wifi | | |

Bảng 2.2 Dữ liệu đầu vào gồm các trường dữ liệu của chức năng đăng nhập.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Trường dữ liệu | Mô tả | Bắt buộc | Ví dụ hợp lệ |
| 1 | UserName | Tên tài khoản đăng nhập | x | Hungpt |
| 2 | Password | Mật khẩu đăng nhập | x | hungpt@1 |
| 3 | Vân tay | Vân tay mã hóa |  |  |
| 4 | Hình ảnh | Hình ảnh đươc gửi dưới dạng bitmap | x |  |

### Đặc tả use case CRUD project

Bảng 2.3 Bảng đặc tả chức năng quản lý dự án

| Mã Usecase | UC0002 | Tên Usecase | CRUD Project |
| --- | --- | --- | --- |
| Tác nhân hệ thống | Nhân viên, Quản lí dự án, Giám đốc | | |
| Tiền điều kiện | Đăng nhập hệ thống thành công | | |
| Luồng sự kiện chính | | STT | Thực hiện | Hành động | | --- | --- | --- | | 1. | Nhân viên | Chọn dự án, xem thông tin dự án | | 2. | Quản lí dự án | Tìm kiếm dự án dựa trên thông tin nhập vào | | 2.1 | Quản lí dự án | Chọn dự án cần cập nhật thông tin, điền thông tin cần sửa đổi | | 2.2 | Quản lí dự án | Submit những thay đổi | | 3. | Giám đốc | Chọn chức năng quản lí dự án | | 3.1 | Giám đốc | Chọn thêm dự án, điền thông tin về dự án | | 3.1.1 | Giam đốc | Submit thêm dự án | | 3.2 | Giám đốc | Chọn xóa dự án | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 2.1a | Hệ thống | Báo ra lỗi dữ liệu truyền vào không phù hợp | | 3.1a | Hệ thống | Báo ra lỗi dữ liệu truyền vào không phù hợp | | | |
| Hậu điều kiện | Không | | |

Bảng 2.4 Dữ liệu đầu vào gồm các trường dữ liệu của chức năng quản lý dự án

| STT | Trường dữ liệu | Mô tả | Bắt buộc | Ví dụ hợp lệ |
| --- | --- | --- | --- | --- |
| 1 | Project Name | Tên dự án | X | Dự án quản lí bàn hàng ACXN |
| 2 | Project Description | Mô tả về dự án |  | Dự án ACXN hỗ trợ các doanh nghiệp vừa và nhỏ trong việc quản lí bán hàng |
| 3 | Project Status | Trạng thái của dự án | x | Mở, đóng |

### Đặc tả use case CRUD task

Bảng 2.5 Bảng đặc tả chức năng quản lý task trong dự án

| Mã Usecase | UC0003 | Tên Usecase | CRUD Task |
| --- | --- | --- | --- |
| Tác nhân hệ thống | Nhân viên, quản lí dự án | | |
| Tiền điều kiện | Đăng nhập thành công | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Nhân viên | Chọn chức năng xem task | | 2 | Nhân viên | Chọn chức năng xem task và nhấn sửa đổi | | 2.1 | Nhân viên | Nhập thời gian dự tính hoàn thành công việc hoặc thay đổi trạng thái đầu việc | | 3 | Quản lí dự án | Chọn chức năng xem task và nhấn sửa đổi | | 3.1 | Quản lí dự án | Nhập thông tin cần thay đổi, thành viên của dự án | | | |
| Luồng sự kiện thay thế | | STT | Thực hiện | Hành động | | --- | --- | --- | | 2.1a | Hệ thống | Thông báo dự liệu nhập vào không đúng | | 3.1a | Hệ thống | Thông báo dự liệu nhập vào không đúng | | | |
| Hậu điều kiện | Không | | |

Bảng 2.6 Bảng dữ liệu đầu vào của chức năng quản lý task

| STT | Trường dữ liệu | Mô tả | Bắt buộc | Ví dụ hợp lệ |
| --- | --- | --- | --- | --- |
| 1 | Project ID | Id dự án | x | PR0001 |
| 2 | Task Name | Tên của đầu việc | x | Create new requirement |
| 3 | Task Description | Chú thích cho đầu việc |  |  |
| 4 | Task Status | Trạng thái của Task | x | Open |
| 5 | Task Priority | Mức độ ưu tiên của Task | x | Normal |
| 6 | Task Estimate | Thời gian dự tính làm xong Task |  | 48h |

Bảng 2.7 Bảng dữ liệu đầu đầu ra sau khi thực hiện tìm kiếm task.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Trường dữ liệu | Mô tả | Ví dụ hợp lệ |
| 1 | Project ID | Id dự án | PR0001 |
| 2 | Task Name | Tên của đầu việc | Create new requirement |
| 3 | Task Description | Chú thích cho đầu việc |  |
| 4 | Task Status | Trạng thái của Task | Open |
| 5 | Task Priority | Mức độ ưu tiên của Task | Normal |
| 6 | Task Estimate | Thời gian dự tính làm xong Task | 48h |

### Use case CRUD ticket của nhân viên

Bảng 2.8 Bảng đặc tả chức năng quản lý ticket của nhân viên

| Mã Usecase | UC0004 | Tên Usecase | CRUD Ticket |
| --- | --- | --- | --- |
| Tác nhân hệ thống | Nhân viên, Quản lí tài nguyên, Giám đốc | | |
| Tiền điều kiện | Đăng nhập hệ thống thành công | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1. | Nhân viên | Chọn chức năng tạo Ticket | | 1.1 | Nhân viên | Nhập thông tin tạo ticket | | 2 | Quản lí dự án / Giám đốc | Nhận thông báo có ticket, chọn chức năng xem ticket | | 2.1 | Quản lí dự án / Giám đốc | Cập nhật trạng thái, comment. | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1.1a | Hệ thống | Báo ra lỗi dữ liệu truyền vào không phù hợp | | 2.1a | Hệ thống | Báo ra lỗi dữ liệu truyền vào không phù hợp | | | |
| Hậu điều kiện | Không | | |

Bảng 2.9 Bảng dữ liệu đầu vào của chức năng quản lý ticket

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Trường dữ liệu | Mô tả | Bắt buộc | Ví dụ hợp lệ |
| 1 | Ticket Name | Tên ticket | X | Xin nghỉ |
| 2 | Ticket Description | Mô tả về ticket |  |  |
| 3 | Approver ID | Người sẽ đồng ý, cho tiến hành các ticket | X | US0001 |
| 4 | Ticket Priority | Mức độ ưu tiên của ticket | X | Normal |
| 5 | Ticket Status | Trạng thái của ticket | X | Open / Processing / close |

Bảng 2.10 Bảng dữ liệu đầu đầu ra sau khi thực hiện chức năng tìm kiếm ticket.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Trường dữ liệu | Mô tả | Ví dụ hợp lệ |
| 1 | Ticket Name | Tên ticket | Xin nghỉ |
| 2 | Ticket Description | Mô tả về ticket |  |
| 3 | Approver ID | Người sẽ đồng ý, cho tiến hành các ticket | US0001 |
| 4 | Ticket Priority | Mức độ ưu tiên của ticket | Normal |
| 5 | Ticket Status | Trạng thái của ticket | Open / Processing / close |

## Yêu cầu phi chức năng

### Yêu cầu chung về chức năng kỹ thuật

Cho phép phân quyền đến từng chức năng của hệ thống, từng trang thành phần. Hỗ trợ khả năng định nghĩa và phân quyền theo vai trò. Hỗ trợ khả năng tìm kiếm Việt/Anh theo chuẩn unicode TCVN 6909:2001 và tìm kiếm nâng cao. Sao lưu và phục hồi dữ liệu: Cung cấp công cụ cho người quản trị thực hiện sao lưu định kỳ, sao lưu đột xuất và cho phép phục hồi dữ liệu theo phiên bản đã được lưu trữ khi cần thiết hoặc khi có sự cố xảy ra.

An toàn, bảo mật: Thực hiện các cơ chế xác thực, cấp phép truy cập trên cổng thông tin, đồng thời hỗ trợ cơ chế trao đổi thông tin, dữ liệu được mã hóa đảm bảo an toàn cho hệ thống cổng thông tin trong quá trình khai thác, vận hành.

### Yêu cầu về giao diện người dùng

Giao diện cần đẹp và tạo ấn tượng cho người xem, bố cục trình bày dễ xem và dễ sử dụng, giao diện (hệ thống menu, hệ thống trợ giúp, v.v...) nhất quán, dễ dùng, thuận tiện cho người sử dụng. Hệ thống phần mềm được thiết kế sử dụng ngôn ngữ Việt Nam và tiếng Anh cho phép người dùng chuyển đổi ngôn ngữ để dễ ràng sử dụng. Hỗ trợ chuẩn ngôn ngữ Unicode (character set UTF8).

Giao diện sử dụng phải đảm bảo các tính chất như thân thiện với người dùng, tính dễ học và sử dụng. Các sự kiện khi tương tác với giao diện phải có tính phản hồi, khi phải chờ đợi xử lý tiến trình thì cần có thông báo cho người dùng.

### Yêu cầu về bảo mật

Các tiêu chuẩn về bảo mật và toàn vẹn dữ liệu cần đáp ứng các yêu cầu:

* Bảo mật chia làm nhiều mức
* Mã hóa các dữ liệu có yêu cầu bảo mật cao
* Sử dụng hỗ trợ an toàn bảo mật của các ứng dụng nền
* Sử dụng giải pháp xác thực người dùng đa nhân tố
* Tính sẵn sàng cao của hệ thống hạ tầng Clustering server (thiết bị phần cứng)
* Tính sẵn sàng khôi phục lại trạng thái trước đó (Restore Database)
* Luôn luôn đề phòng rủi ro: Tự động sao lưu dữ liệu theo định kỳ.

### Yêu cầu về hiệu năng sử dụng

Giải pháp kỹ thuật xây dựng phần mềm phải đáp ứng các yêu cầu sau đây:

* Tính khả thi: Giải pháp đưa ra phải giải quyết được các yêu cầu đang đặt ra, phù hợp với điều kiện thực tế của các công ty, và quy mô của họ.
* Tính hiện đại: Các giải pháp đưa ra dựa trên các công nghệ mới hiện đại và đang được sử dụng phổ biến.
* Tính mở: Giải pháp đưa ra phải dễ dàng kết nối cũng như tích hợp thêm các giải pháp khác khi cần thiết.
* Tính linh động: hệ thống cần phải linh động để đáp ứng được các thay đổi dựa trên yêu cầu từ phía người sử dụng cũng như các yêu cầu phát sinh từ hệ thống.
* Tính toàn vẹn: giải pháp phải có các cơ chế sao lưu phục hồi khi hệ thống có lỗi để tránh việc mất mát dữ liệu.
* Dễ dàng mở rộng hệ thống khi có nhu cầu như nhu cầu tích hợp thêm các nguồn thông tin mới ...
* Hiệu năng thực hiện cao (theo nghĩa sử dụng tài nguyên, tốc độ đáp ứng yêu cầu của người khai thác thông tin, ...).

# ****Công**** nghệ sử dụng

## . Hệ quản trị CSDL MySQL

### MySQL là gì

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh.

Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. MySQL miễn phí hoàn toàn cho nên bạn có thể tải về MySQL từ trang chủ. Nó có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS…

Bên cạnh đó MySQL là một trong những ví dụ rất cơ bản về hệ quản trị CSDL quan hệ sử dụng ngôn ngữ truy vấn có cấu trúc (SQL). MySQL được sử dụng cho việc bổ trợ PHP, Perl, và nhiều ngôn ngữ khác, nó làm nơi lưu trữ những thông tin trên các trang web viết bằng PHP hay Perl.

### Ưu điểm của hệ quản trị CSDL MySQL

Hệ quản trị CSDL MySQL có rất nhiều ưu điểm khác nhau, đầu tiên phải nói đến tốc độ xử lý. MySQL rất nhanh, những nhà phát triển cho rằng MySQL là cơ sở dữ liệu nhanh nhất mà người dùng có thể có. Tiếp đến là tính dễ sử dụng, MySQL tuy có tính năng cao nhưng thực sự là một hệ thống cơ sở dữ liệu rất đơn giản và ít phức tạp khi cài đặt và quản trị hơn các hệ thống lớn.

Tiếp theo là về giá thành, MySQL là miễn phí cho hầu hết các việc sử dụng trong một tổ chức. Hơn nữa, MySQL còn hỗ trợ ngôn ngữ truy vấn: MySQL hiểu SQL, là ngôn ngữ của sự chọn lựa cho tất cả các hệ thống cơ sở dữ liệu hiện đại. Bạn cũng có thể truy cập MySQL bằng cách sử dụng các ứng dụng mà hỗ trợ ODBC (Open Database Connectivity - một giao thức giao tiếp cơ sở dữ liệu được phát triển bởi Microsoft).

Về Năng lực, MySQL cho phép nhiều client có thể truy cập đến server trong cùng một thời gian. Các client có thể sử dụng nhiều cơ sở dữ liệu một cách đồng thời. Bạn có thể truy cập MySQL tương tác với sử dụng một vài giao diện để bạn có thể đưa vào các truy vấn và xem các kết quả: các dòng yêu cầu của khách hàng, các trình duyệt Web…

Về khả năng kết nối và bảo mật, MySQL được nối mạng một cách đầy đủ, các cơ sở dữ liệu có thể được truy cập từ bất kỳ nơi nào trên Internet do đó bạn có thể chia sẽ dữ liệu của bạn với bất kỳ ai, bất kỳ nơi nào. Nhưng MySQL kiểm soát quyền truy cập cho nên người mà không nên nhì n thấy dữ liệu của bạn thì không thể nhìn được.

Về tính linh động, MySQL chạy trên nhiều hệ thống UNIX cũng như không phải UNIX chẳng hạn như Windows hay OS/2. MySQL chạy được các với mọi phần cứng từ các máy PC ở nhà cho đến các máy server.

Về sự phân phối rộng, MySQL rất dễ dàng đạt được, chỉ cần sử dụng trình duyệt web của bạn. Nếu bạn không hiểu làm thế nào mà nó làm việc hay tò mò về thuật toán, bạn có thể lấy mã nguồn và tìm tòi nó. Nếu bạn không thích một vài cái, bạn có thể thay đổi nó.

Cuối cùng về khả năng hỗ trợ, người dùng có thể tìm thấy các tài nguyên có sẵn mà MySQL hỗ trợ. Cộng đồng MySQL rất có trách nhiệm. Họ trả lời các câu hỏi trên mailing list thường chỉ trong vài phút. Khi lỗi được phát hiện, các nhà phát triển sẽ đưa ra cách khắc phục trong vài ngày, thậm chí có khi trong vài giờ và cách khắc phục đó sẽ ngay lập tức có sẵn trên Internet.

Chính vì những ưu điểm vượt trội đó mà MySQL được rất nhiều người dùng ưu chuộng và sử dụng.

## Xử lý hình ảnh OpenCV [4]

### Nhận diện khuôn mặt và thư viện OpenCV

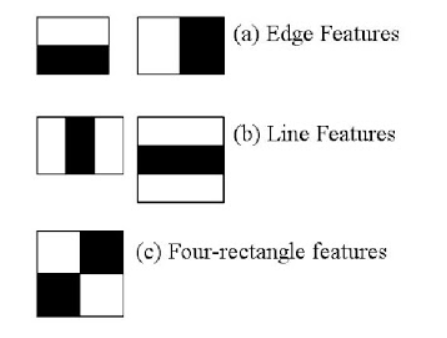
Nhận diện khuôn mặt với con người là nhiệm vụ rất dễ dàng, trẻ em từ một đến ba ngày tuổi cũng có thể phân biệt được gương mặt đã biết do bộ não của con người có những tế bào thần kinh chuyên biệt để xử lý thông tin về hình ảnh. Khác với cơ chế sinh học của bộ não con người, để máy tính có thể nhận diện được gương mặt cần phải có những thư viện hỗ trợ xử lý hình ảnh. Sau khi tìm hiểu, em sử dụng thư viện OpenCV để thực hiện xây dựng chức năng nhận diện gương mặt.

OpenCV (Open Soure Computer Vision) là một thư viện xử lý đồ họa đa nền tảng được Intel xây dựng từ năm 1999. OpenCV được phát hành theo giấy phép BSD, do đó nó hoàn toàn miễn phí cho cả học thuật và thương mại. Nó có các interface C++, C, Python, Java và hỗ trợ Windows, Linux, Mac OS, iOS và Android. OpenCV được thiết kế để tính toán hiệu quả và với sự tập trung nhiều vào các ứng dụng thời gian thực. Được viết bằng tối ưu hóa C/C++, thư viện có thể tận dụng lợi thế của xử lý đa lõi. Được sử dụng trên khắp thế giới, OpenCV có cộng đồng hơn 47 nghìn người dùng và số lượng download vượt quá 6 triệu lần. Từ phiên bản OpenCV 2.4 trở đi, với lớp FaceRecognizer đi kèm, chúng ta có thể xây dựng chức năng nhận diện khuôn mặt.

### Nhận diện khuôn mặt bằng phương pháp Haar

Phát hiện một đối tượng bằng cách sử dụng Haar là một phương pháp phát hiện đối tượng hiệu quả được đề xuất bởi Paul Viola và Michael Jones vào năm 2001. Nó là một phương pháp tiếp cận dựa trên máy học những tập hình ảnh đối tượng ban đầu rồi sau đó sử dụng để phát hiện ra các đối tượng cùng loại trong các hình ảnh khác.

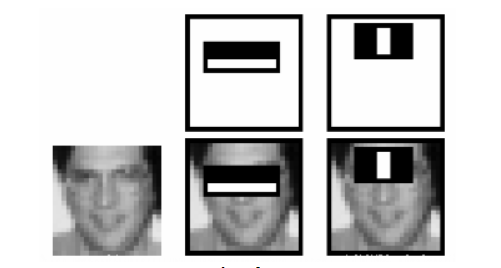
Trong trường hợp nhận diện khuôn mặt, ban đầu thuật toán cần nhiều hình ảnh tích cực (hình ảnh có khuôn mặt) và hình ảnh tích cực (hình ảnh không có khuôn mặt) để huấn luyện phân loại. Từ đó, chúng ta trích xuất được những đặc điểm khuôn mặt, các đặc điểm Haar được thể hiện như Hình 3.1 sau đây. Nó như những giá trị cơ bản ban đầu (hạt nhân). Mỗi đối tượng là một giá trị duy nhất thu được bằng cách trừ tổng số điểm ảnh trong hình chữ nhật màu trắng từ tổng số pixel dưới hình chữ nhật đen.



Hình 3.1 Ví dụ đặc điểm Haar.

Bây giờ, tất cả các kích thước và vị trí có thể có của mỗi hạt nhân được sử dụng để tính toán rất nhiều đặc điểm (với 24\*24 cử sổ đã có hơn 160000 đặc điểm). Đối với mỗi phép tính tính năng, chúng ta cần tìm tổng các pixel dưới hình chữ nhật trắng và đen. Để giải quyết điều này, họ sử dụng hình ảnh tích phân. Tuy nhiên, với hình ảnh lớp, nó giảm việc tính toán cho một pixel nhất định cho một hoạt động liên quan đến chỉ bốn điểm ảnh.

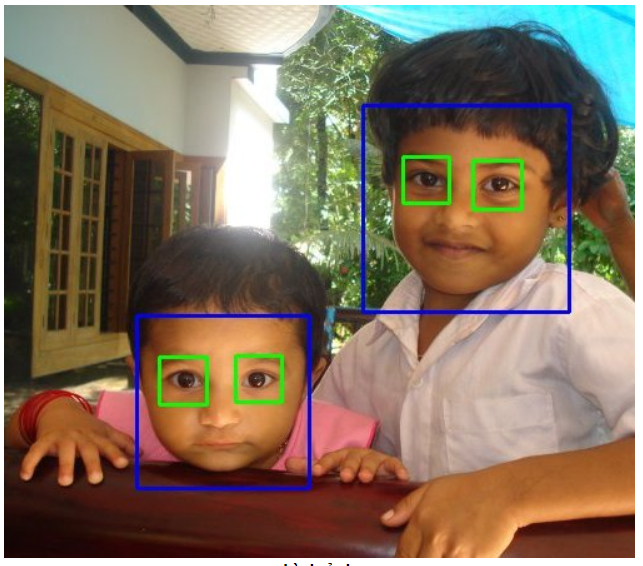
Trong hàng loạt các đặc điểm gương mặt tính toán, hầu hết chúng ta không cần quan tâm. Hai đặc điểm quan trọng và chúng ta cần tập trung vào là vùng mắt thường tối hơn vùng mũi và má và đôi mắt tối hơn sống mũi. Nhưng có rất nhiều những cửa số chúng ta không quan tâm, vậy cần làm cách nào để chọn ra các đặc điểm cần thiết trong số hơn 160000 đặc điểm.



Hình 3.2 Ví dụ về đặc điểm gương mặt.

Để chọn ra những phần chúng ta quan tâm, đối với mỗi vùng đối tượng, tìm ra ngưỡng tốt nhất sẽ phân loại các mặt thành dương và âm. Chắc chắn sẽ có các lỗi phân loại sai, chúng ta chọn ra những đặc điểm có tỷ lệ lỗi tối thiểu, có nghĩa chúng là các tính năng phân loại chính xác nhất hình ảnh khuôn mặt và hình ảnh không phải khuôn mặt

Trong việc phân loại cuối cùng là việc tính tổng trọng số của các bộ phận phân loại yếu. Nó được gọi là yếu bởi vì nó một mình không thể phân loại hình ảnh, nhưng cùng với những người khác tạo thành một phân loại mạnh mẽ. Với 200 đặc điểm cung cấp khả năng phát hiện với độ chính xác 95%. Thiết lập cuối cùng của họ có khoảng 6000 đặc điểm (giảm từ hơn 16000 đặc điểm). Kết quả đặt được sẽ tương tự như Hình 3.3 Ví dụ kết quả nhận diện khuôn mặt.



Hình 3.3 Ví dụ kết quả nhận diện khuôn mặt.

## GPS [6]

### Tổng quan về GPS

GPS là viết tắt của "global positioning system" (hệ thống định vị toàn cầu), thực chất là một mạng lưới bao gồm 27 vệ tinh quay xung quanh trái đất. Trong số 27 vệ tinh này, 24 vệ tinh đang hoạt động, 3 vệ tinh còn lại đóng vai trò dự phòng trong trường hợp 1 trong số 24 vệ tinh chính bị hư hỏng. Dựa vào cách sắp đặt của các vệ tinh này, khi đứng dưới mặt đất, bạn có thể nhìn được ít nhất là 4 vệ tinh trên bầu trời tại bất kì thời điểm nào.

GPS cho phép mọi người trên thế giới sử dụng một số chức năng của GPS miễn phí. Nên bạn có thể sử dụng định vị trên các thiết bị thu GPS để xác định vị trí của mình một cách chính xác và hoàn toàn miễn phí.

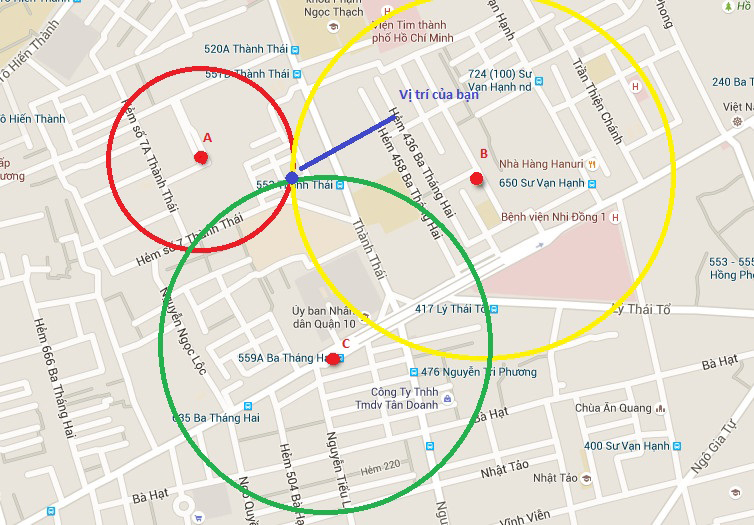
Ngoài việc sử dụng GPS để xác định vị trí hiện tại, các thiết bị di động ngày nay còn sử dụng một hệ thống định vị tương tự do Nga phát triển cùng thời điểm với GPS là GLONASS. Ban đầu, cả 2 công nghệ định vị này chỉ dành cho các mục đích quân sự nhưng từ năm 1980, chính phủ Mỹ đã cho phép sử dụng GPS ở mọi nơi trên Trái Đất, 24h mỗi ngày và không yêu cầu thuê bao hay bất kỳ loại phí dịch vụ nào.

### Cơ chế hoạt động của GPS

Các vệ tinh GPS bay hai vòng trong một ngày theo một quỹ đạo đã được tính toán chính xác và liên tục phát các tín hiệu có thông tin xuống Trái Đất. Các máy thu GPS nhận các tín hiệu này và giải mã bằng các phép tính lượng giác, qua đó sẽ tính toán và hiển thị được vị trí của người dùng.

Thiết bị smartphone (Android, Windows Phone, iOS) của bạn là một đầu thu GPS, nó sẽ thu dữ liệu từ các vệ tinh GPS ở trên bầu trời. Dữ liệu gì? Nói một cách đơn giản, mỗi vệ tinh cho bạn biết khoảng cách chính xác từ vị trí của bạn đến vệ tinh đó hoặc một điểm nào đó trên trái đất.

Cơ chế hoạt động của GPS rất đơn giản, bạn có thể tưởng tượng như sau. Trên bản đồ có 3 điểm cố định A, B, C. Dữ liệu GPS cho bạn biết khoảng cách lần lượt từ điểm A, B, C đến nơi bạn đứng là 1, 3km, 2km. Sau đó bạn vẽ 3 vòng tròn có tâm là A, B, C với bán kính lần lượt là 1km, 3km và 2km. Ví dụ như Hình 3.4



Hình 3.4 Ví dụ cách hoạt động GPS.

Vị trí giao nhau của ba vòng tròn chính là vị trí của bạn.

Thiết bị smartphone phải nhận được tín hiệu của ít nhất ba vệ tinh để cho ra vị trí hai chiều và để theo dõi được chuyển động của bạn. Khi nhận được tín hiệu của ít nhất bốn vệ tinh, máy sẽ cho ra được vị trí ba chiều. Một khi vị trí của bạn đã tính được thì thiết bị smartphone (Android, Windows Phone, iOS) có thể tính các thông tin khác, như tốc độ di chuyển, hướng chuyển động, bám sát di chuyển, khoảng hành trình, khoảng cách đích đến và nhiều thứ khác nữa.

Để đưa ra vị trí chính xác, rất nhiều thiết bị GPS kết nối tới ít nhất là 4 vệ tinh. Đó là lý do vì sao đôi khi để tìm ra vị trí chính xác của bạn, hệ thống GPS lại mất nhiều thời gian tới vậy. Đó cũng là lý do vì sao đôi khi bạn bị mất sóng GPS: thiết bị di động của bạn có thể đã kết nối tới 1 hoặc 2 vệ tinh, song 2 vệ tinh vẫn là không đủ.

## Hàm hash SHA256

Hàm băm (hash) là giải thuật được xây dựng để ánh xạ một ký tự có độ dài bất kỳ thành một giá trị băm có độ dài giống nhau và đảm bảo mỗi ký tự đầu vào khác nhau sẽ tạo ra được một giá trị băm khác nhau. Hàm băm được xây dựng tốt là một phép biến đổi “một chiều”, nghĩa là không có một phương pháp nào để xác định lại giá trị đầu vào ban đầu từ giá trị băm nhận được.

Hàm băm được sử dụng trong việc phát triển ứng dụng là hàm băm SHA256, đây là một trong số những giải thuật được phát triển bởi viện công nghệ và tiêu chuẩn quốc gia Mỹ. Giải thuật băm SHA (viết tắt của Secure Hash Algorithm) gồm nhiều phiên bản với những đặc điểm khác nhau về độ dài giá trị băm: Giải thuật SHA 256 trả về dữ liệu có độ dài 256 bit, SHA1 trả về kết quả dài 160 bit, SHA224 trả về kết quả dài 224 bit, SHA384 với độ dài kết quả 384bit và cuối cùng là SHA512 với kết quả dài nhất là 512 bit. Hình 3.5 mô tả ví dụ về kết quả đầu ra sau khi sử dụng giải thuật băm SHA256.



Hình 3.5 Ví dụ về kết quả trả về hàm hash SHA256

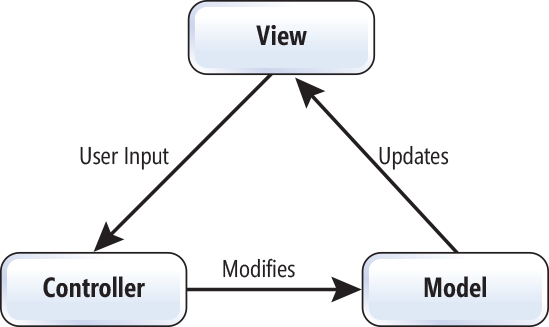
# Phát triển và triển khai ứng dụng

## Thiết kế kiến trúc

### Lựa chọn kiến trúc phần mềm

#### Mô hình kiến trúc MVC

Mô hình MVC (Model - View - Controller) là một kiến trúc phần mềm hay mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. Nó giúp cho các developer tách ứng dụng của họ ra 3 thành phần khác nhau Model, View và Controller. Mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần khác.

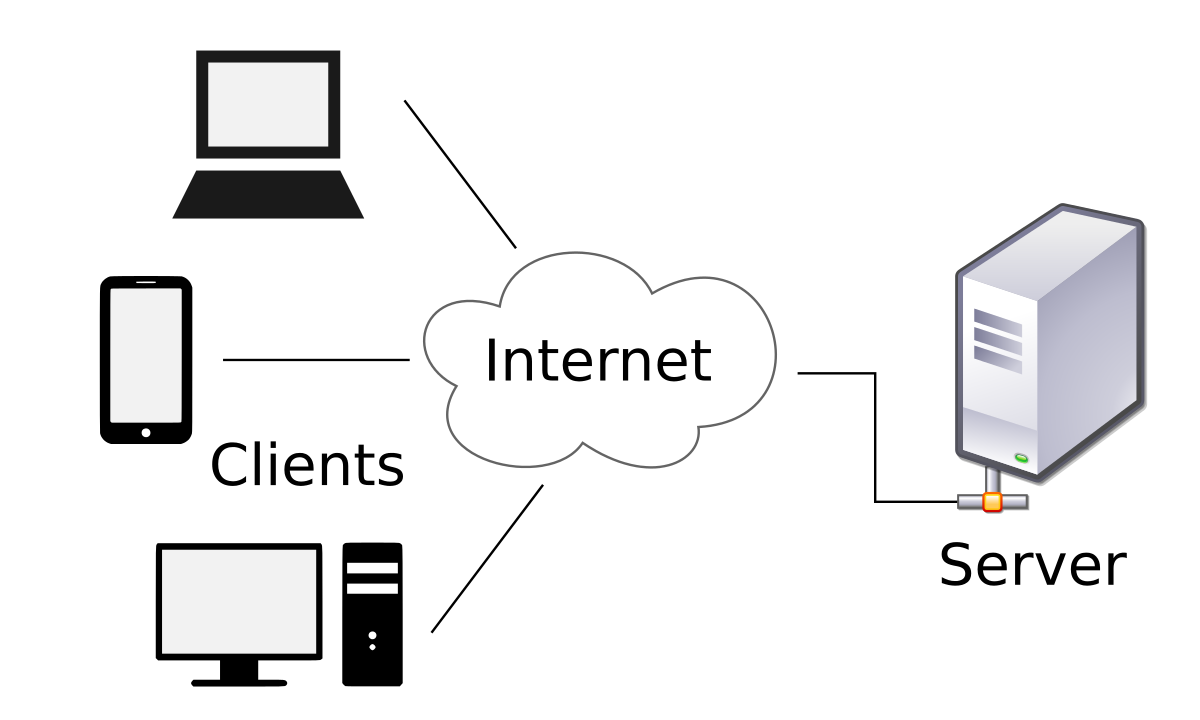


Hình 4.1 Mô hình MVC

Model đây là thành phần chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất database, đối tượng mô tả dữ liệu như các Class, hàm xử lý. View đảm nhận việc hiển thị thông tin, tương tác với người dùng, nơi chứa tất cả các đối tượng GUI như textbox, images...Hiểu một cách đơn giản, nó là tập hợp các form hoặc các file HTML. Controller giữ nhiệm vụ nhận điều hướng các yêu cầu từ người dùng và gọi đúng những phương thức xử lý chúng... Chẳng hạn thành phần này sẽ nhận request từ url và form để thao tác trực tiếp với Model.

#### Mô hình client – server

Mô hình client-server là một mô hình nổi tiếng trong [mạng máy tính](https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_m%C3%A1y_t%C3%ADnh), được áp dụng rất rộng rãi và là mô hình của mọi trang web hiện có. Ý tưởng của mô hình này là máy con (đóng vài trò là máy khách) gửi một yêu cầu (request) để máy chủ (đóng vai trò người cung ứng dịch vụ), máy chủ sẽ xử lý và trả kết quả về cho máy khách.



Hình 4.2 Mô hình client – server

Client trong mô hình client/server là một máy trạm mà chỉ được sử dụng bởi 1 người dùng với để muốn thể hiện tính độc lập cho nó. Bản thân mỗi một client cũng đã được tích hợp nhiều chức năng trên hệ điều hành mà nó chạy, nhưng khi được nối vào một mạng internet theo mô hình client/server thì nó còn có thể sử dụng thêm các chức năng do hệ điều hành mạng cung cấp với nhiều dịch vụ khác nhau (cụ thể là các dịch vụ do các server trên mạng này cung cấp), ví dụ như nó có thể yêu cầu lấy dữ liệu từ một server hay gửi dữ liệu lên server đó.

Server còn được định nghĩa như là một máy tính nhiều người sử dụng (multiuser computer). Vì một server phải quản lý nhiều yêu cầu từ các client trên mạng cho nên nó hoạt động sẽ tốt hơn nếu hệ điều hành của nó là đa nhiệm với các tính năng hoạt động độc lập song song với nhau. Server cung cấp và điều khiển các tiến trình truy cập vào tài nguyên của hệ thống. Các ứng dụng chạy trên server phải được tách rời nhau để một lỗi của ứng dụng này không làm hỏng ứng dụng khác. Tính đa nhiệm đảm bảo một tiến trình không sử dụng toàn bộ tài nguyên hệ thống.

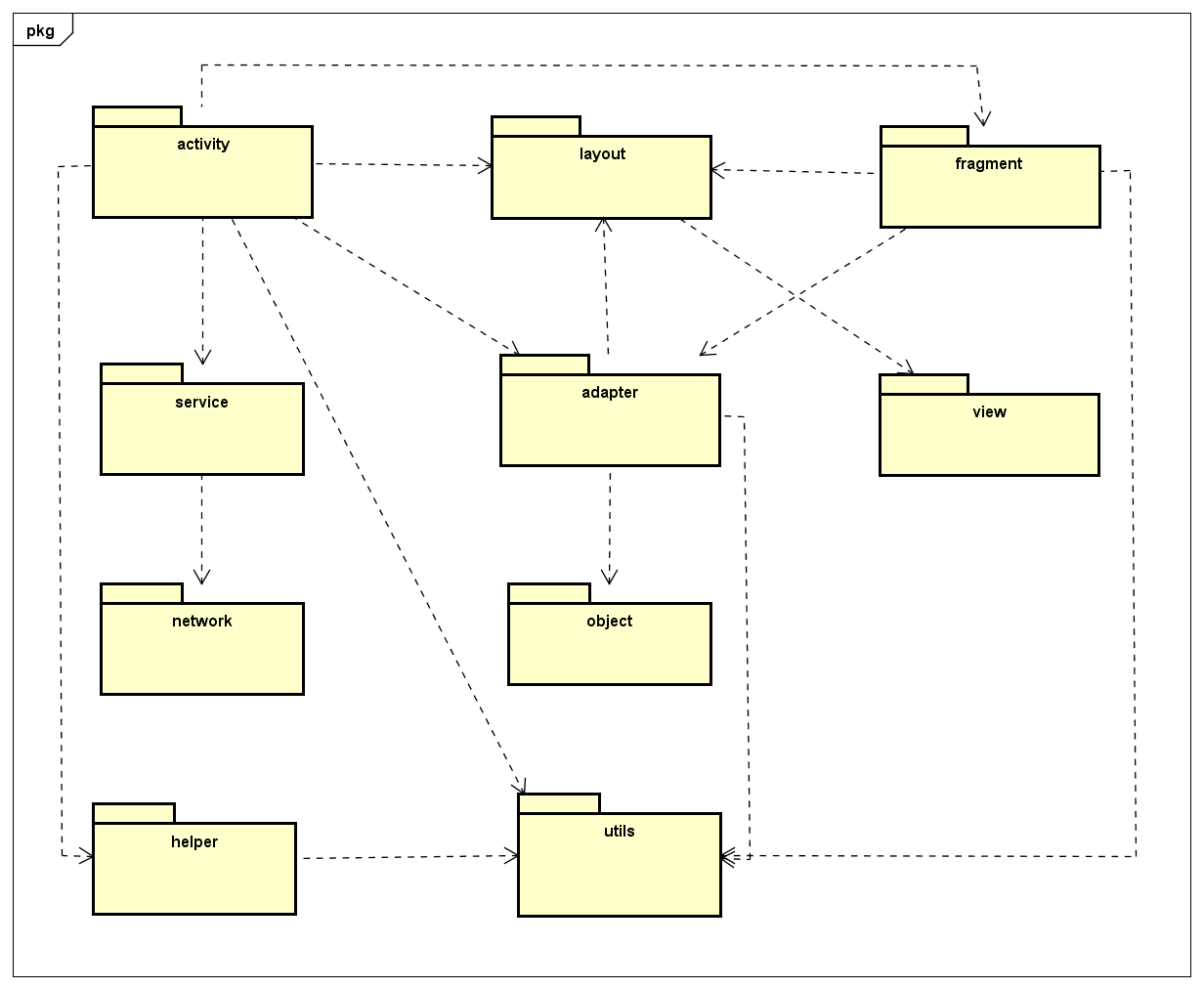
#### Mô hình CSDL phân tán

CSDL phân tán là một tuyển tập dữ liệu có quan hệ với nhau được phân bố trên các máy tính khác nhau của một mạng máy tính, đồng thời cho phép quản trị dữ liệu một cách phân tán cũng như đảm bảo tính trong suốt về sự phân tán với người dùng. Nó được yêu cầu và thực hiện bởi một máy tính tại một nút trong hệ và chỉ được truy cập tới các dữ liệu liên quan tới nó.

Một cơ sở dữ liệu phân tán là một cơ sở dữ liệu logic đơn lẻ mà được trải ra về mặt vật lý trên nhiều máy tính ở nhiều vị trí địa lý khác nhau. Có hai kiểu chung nhất của các hệ cơ sở dữ liệu phân tán là: hệ cơ sở dữ liệu phân tán thuần nhất và hệ cơ sở dữ liệu phân tán không thuần nhất.

Hệ cơ sở dữ liệu phân tán thuần nhất, khi áp dụng đối với các hệ cơ sở dữ liệu, thuật ngữ thuần nhất có nghĩa là công nghệ cơ sở dữ liệu là như nhau (hay ít nhất có thể tương thích) tại mỗi vị trí địa lý khác nhau cũng có thể tương thích. Đối với các hệ cơ sở dữ liệu phân tán thuần nhất này, các điều kiện sau đây có thể tồn tại: Các hệ điều hành máy tính tại mỗi vị trí địa lý là như nhau hay ít nhất chúng có khả năng tương thích cao, các mô hình dữ liệu được sử dụng tại mỗi vị trí địa lý là như nhau, các hệ quản trị cơ sở dữ liệu được sử dụng tại mỗi vị trí địa lý là như nhau hay ít nhất chúng có khả năng tương thích cao. Đối với hệ cơ sở dữ liệu phân tán không thuần nhất các máy tính khác nhau và các hệ điều hành khác nhau có thể được sử dụng tại mỗi vị trí địa lý. Các mô hình dữ liệu khác nhau và các hệ quản tri cơ sở dữ liệu khác nhau cũng có thể được lựa chọn sử dụng. Ví dụ, một vị trí có thể sử dụng công nghệ cơ sở dữ liệu quan hệ mới nhất, trong khi một vị trí khác có thể lưu trữ dữ liệu sử dụng các tệp truyền thống hay các cơ sở dữ liệu mạng, phân cấp cũ hơn. Phức tạp hơn nữa, dữ liệu trên các vị trí thường không tương thích. Các mâu thuẫn điển hình bao gồm các khác biệt về cú pháp (sự biểu diễn khác nhau của các khoản mục dữ liệu tại hai vị trí) và các khác biệt về ngữ nghĩa.

### Thiết kế tổng quan



Hình 4.3 Biểu đồ phụ thuộc gói

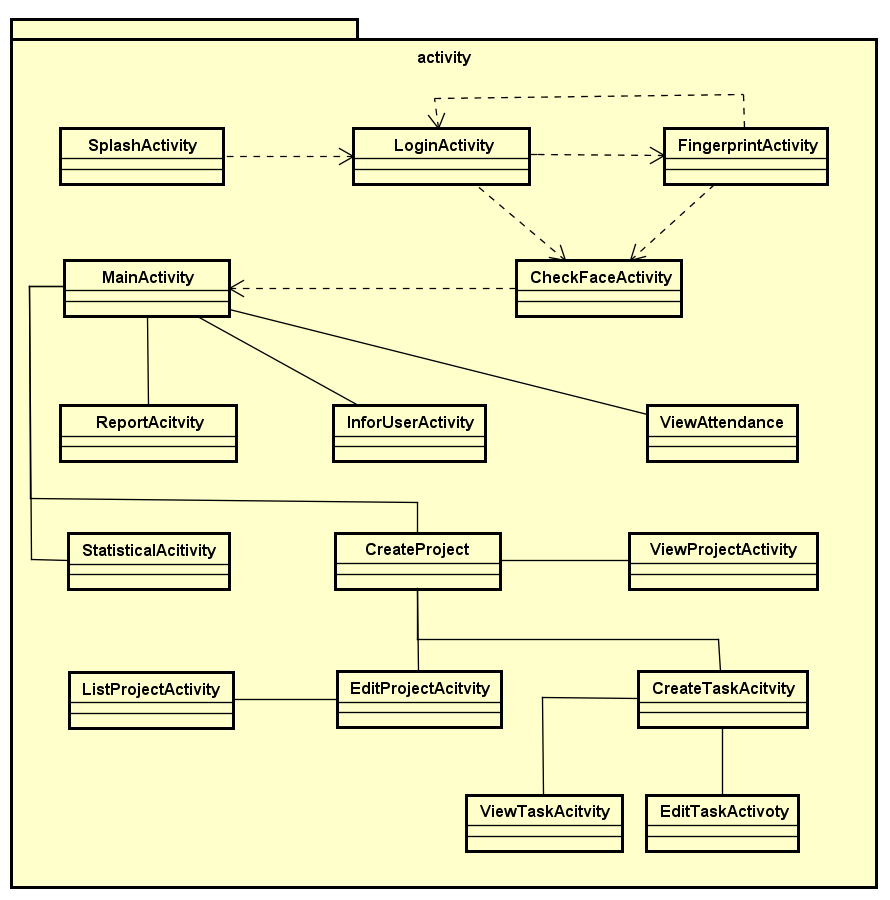
Hình 4.3 Biểu đồ phụ thuộc gói là sơ đồ các gói trong hệ thống trong đó gói activity là nơi chưa nhưng file điều khiển của ứng dụng, điều hướng các trang và hiện thị giao diện người dùng, bắt các sự kiện click và trả về các phản hồi của hê thống. Gói layout là nơi chưa các thiết kế vể giao diện người dùng được gọi đến và hiển thị trên các activity của ứng dụng. Tiếp theo gói fragment là nơi chứa là một thành phần android **độc lập**, được sử dụng bởi một activity, giống như một **sub-activity**. Fragment có **vòng đời và giao diện riêng**. Các Fragment thường có một file java đi kèm với file giao diện xml. Các fragment không có file giao diện xml thường được gọi là **headless fragments.**

Tiếp đógói service là nơi lưu trữ các dịch vụ chạy nền của ứng dụng, cho phép ứng dụng tiếp tục hoạt động ngay cả khi người dùng đóng ứng dụng. Gói view là nơi chứa các file thiết kế giao diện đặc biệt không được hỗ trợ bằng xml. Gói network là nơi chứa các các file hỗ trợ, quản lý kết nối internet của hệ thống. Gói helper là nơi chưa các file quản lý và hỗ trợ giao tiếp với Sqlite. Gói adapter là nơi chứa các đối tượng của một lớp cài đặt giao diện. Nó đóng vai trò như là một liên kết giữa một tập hợp dữ liệu và một đối tượng của một lớp thừa kế lớp trừu tượng. Tập hợp dữ liệu có thể là bất cứ điều gì mà trình bày dữ liệu một cách có cấu trúc.

Gói object là nơi chứa các đối tượng là thực thể mang tính vật lý cũng như mang tính logic, trong khi lớp chỉ là thực thể logic, có các trạng thái và các hành vi. Cuối cùng là gói ultils đây là nơi chứa các file trợ giúp hỗ trợ giúp cho việc xử lý dữ liệu và hoạt động của ứng dụng trở nên đơn giản hơn.

### Thiết kế chi tiết gói

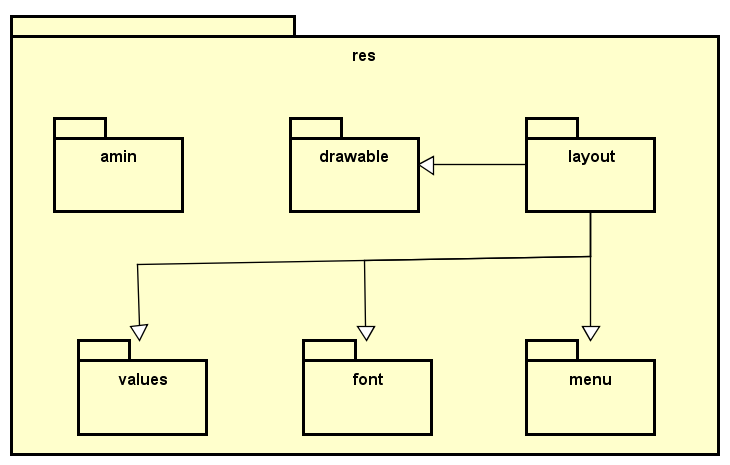
#### Thiết kế chi tiết gói Activity



Hình 4.4 Thiết kế gói activity

Hình 4.4 Thiết kế gói activity thể hiện đầy đủ các lớp và mối quan hệ và luồng dịch chuyển giữa các acitvity trong hệ thống. Hệ thống gồm 16 các acivity khác nhau đại diện cho 16 lớp mà hình chức năng chính mà người dùng tương tác với hệ thống.

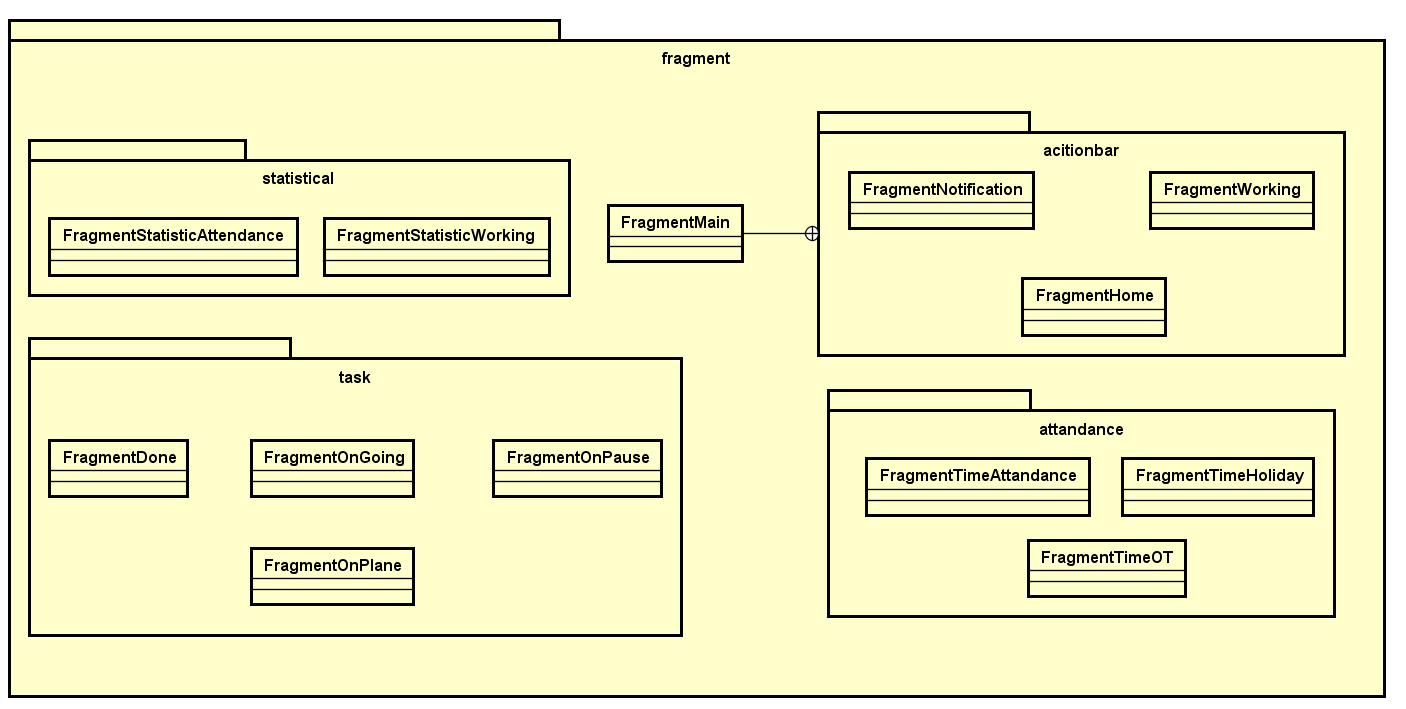
#### Thiết kế chi tiết gói layout



Hình 4.5 Chi tiết gói res

Hình 4.5 Chi tiết gói res thể hiện các gói bên trong của nó. Đây là các gói hỗ trợ cho việc hiển thị giao diện trong đó gói amin là nơi chứa các file hiệu ứng của hệ thống. Gói drawable là nơi chứ các file hình ảnh, gói layout là nơi chứa các file thiết kế giao diện, gói font chứa các tệp tin về font chữ, gói menu chứa các menu của hệ thống. Và cuối cùng là gói values là nơi chứa các tệp tin định nghĩa cho ứng dụng như file string, file colors, file dimen, file style.

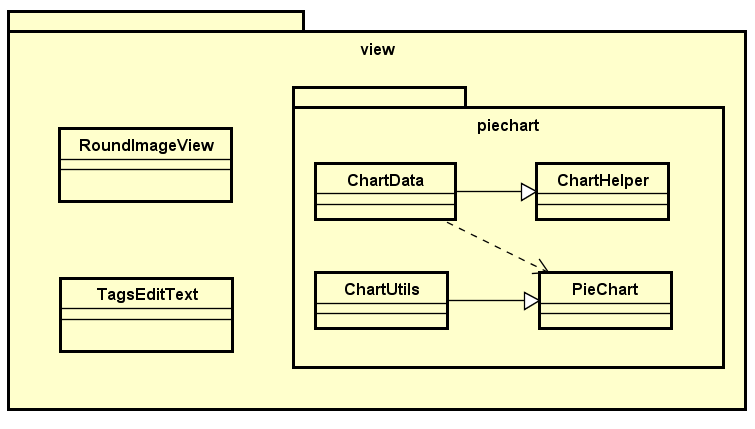
#### Chi tiết gói fragment



Hình 4.6 Chi tiết gói fragment

Hình 4.6 Chi tiết gói fragment thể hiện cấu trúc của gói fragment. FragmentMain là nơi chứa các viewpage của giao diện chính, gói actionbar chính là nơi chứa các fragment trong viewpage của FragmentMain. Gói statistical là nơi chứa các fragment về thống kê (Bao gồm thống kê điểm danh và thống kê công việc). Gói task là nơi chứa các fragment về thẻ công việc trong project. Gói attendance là nơi chứa các fragment về dữ liệu điểm danh bao gồm dữ liệu điểm danh hàng ngày, dữ liệu điểm danh khi làm thêm và dữ liệu điểm danh trong ngày nghỉ.

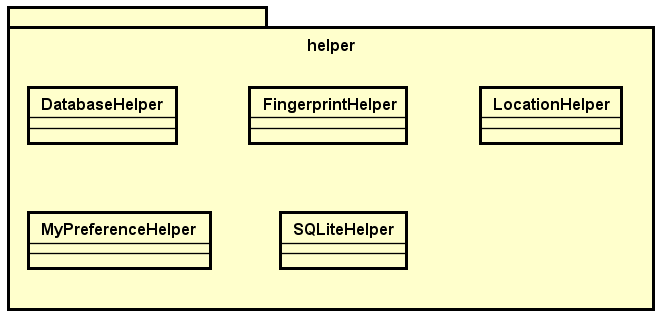
#### Chi tiết gói view



Hình 4.7 Chi tiết gói view

Hình 4.7 Chi tiết gói view thể hiện cấu trúc của gói view. Đây là nơi chứa phần giao diện mà khó có thể thực hiện chỉ bằng các file layout trong xml. Bao gồm các file custom lại sự hiển thị của hình ảnh, và các bộ hỗ trợ vẽ biểu đồ trong thống kê bao gồm biểu đồ hình tròn và biểu đồ hình cột.

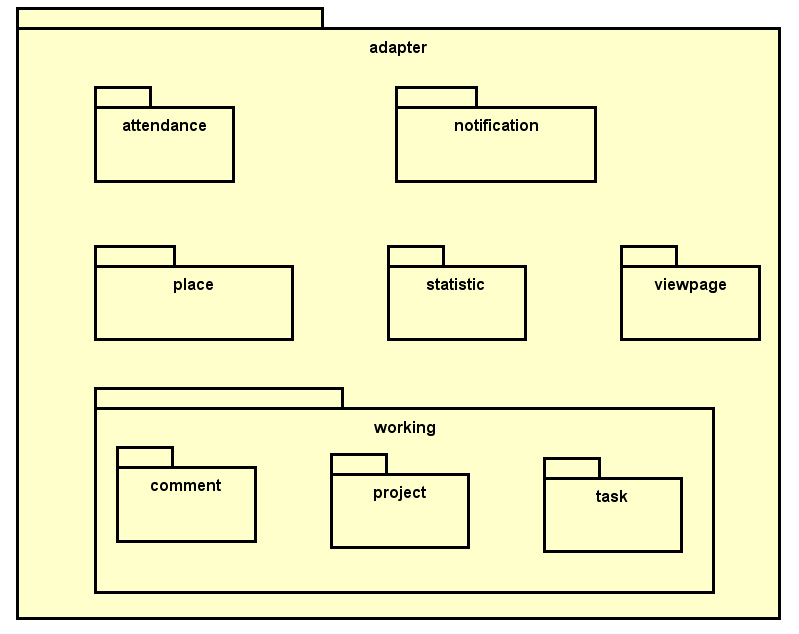
#### Chi tiết gói helper



Hình 4.8 Chi tiết gói helper

Hình 4.8 Chi tiết gói helper mô tả chi tiết cấu trúc của gói helper nơi chứa các tiện ích trợ giúp đuợc xây dựng sẵn để sử dụng trong quá trình xây dựng ứng dụng bao gồm hỗ trợ về vân tay, hỗ trợ về kết nối và truy vấn cơ sở dữ liệu, hỗ trợ về xác định vị trí và địa điểm, hỗ trợ về tìm đường và gợi ý địa điểm.

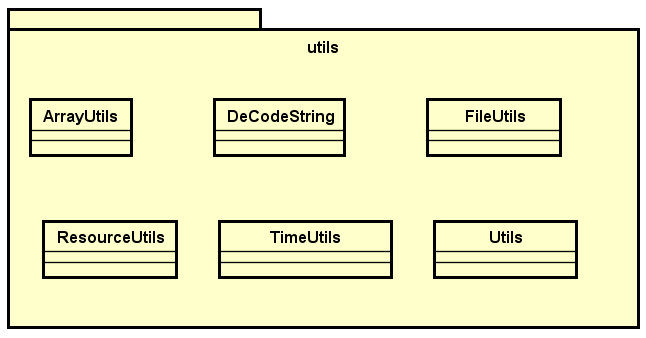
#### Chi tiết gói adapter



Hình 4.9 Chi tiết gói adapter

Hình 4.9 Chi tiết gói adapter mô tả cấu trúc của gói adapter nói chứa các file định nghĩa hiển thị dữ liệu trên màn hình người dùng, xử lý các dữ liệu đầu vào và ra cũng như góp phần điều hướng cho ứng dụng.

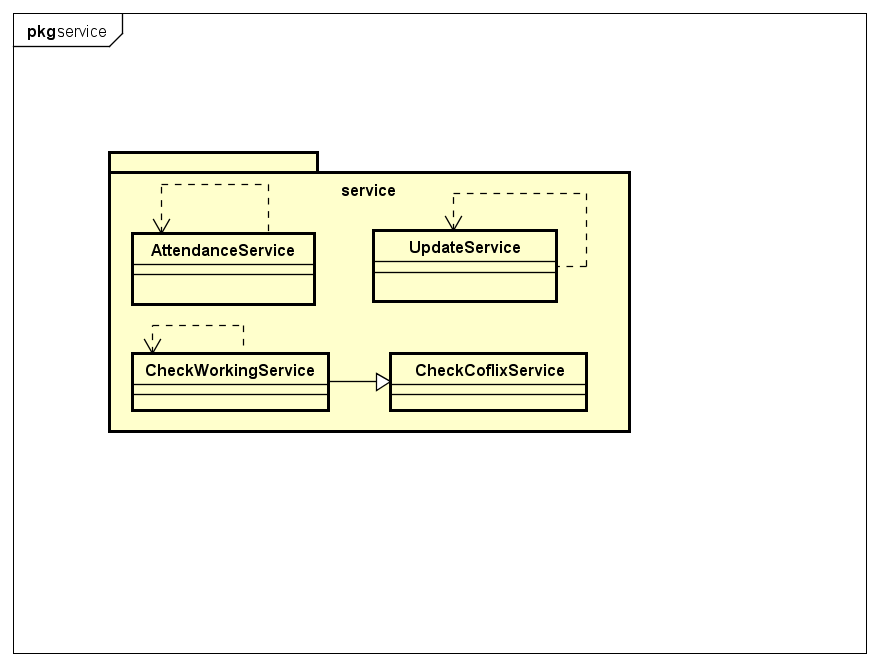
#### Chi tiết gói utils



Hình 4.10 Chi tiết gói utils

Hình 4.10 Chi tiết gói utils thể hiện cấu trúc của gói chưa các file trợ giúp bao gồm trợ giúp tạo và quản lý file, trợ giúp và cấu hình thời gian, kiểm tra và quản lý internet, phân giải ảnh thành bitmap và chuyển đổi bitmap thành dữ liệu hiển thị ảnh trên ứng dụng từ đó rễ dàng cho việc xây dựng ứng dụng, người thiết kế có thể tái sử dụng lại code, người phát triển và sửa đổi dễ dàng quản lý và cập nhật.

#### Chi tiết gói service



Hình 4.11 Chi tiết gói service

Hình 4.11 mô tả chi tiết cấu trúc gói service nơi chứa các dịch vụ hoạt động ngầm dưới nền ứng dụng đảm bảo sự trong suốt của sự phân tán trong hệ thống. Bao gồm các service quản lí cập nhật dữ liệu, kiểm tra điểm danh, kiểm tra các tiến trình công việc và xử lý các ngoại lệ trong cập nhật dữ liệu.

## Thiết kế chi tiết

### Thiết kế giao diện

#### Yêu cầu về thiết kế

Ứng dụng phả được thiết kế và tích hợp sử dụng được trên tất cả các thiết bị android có API ≥ 18 trừ đồng hồ android. Giao diện phải được hiển thị và co dãn phù hợp trên từng loại màn hình khác nhau, đặc biệt không bị nhảy giao diện gây khó chịu cho người dùng.

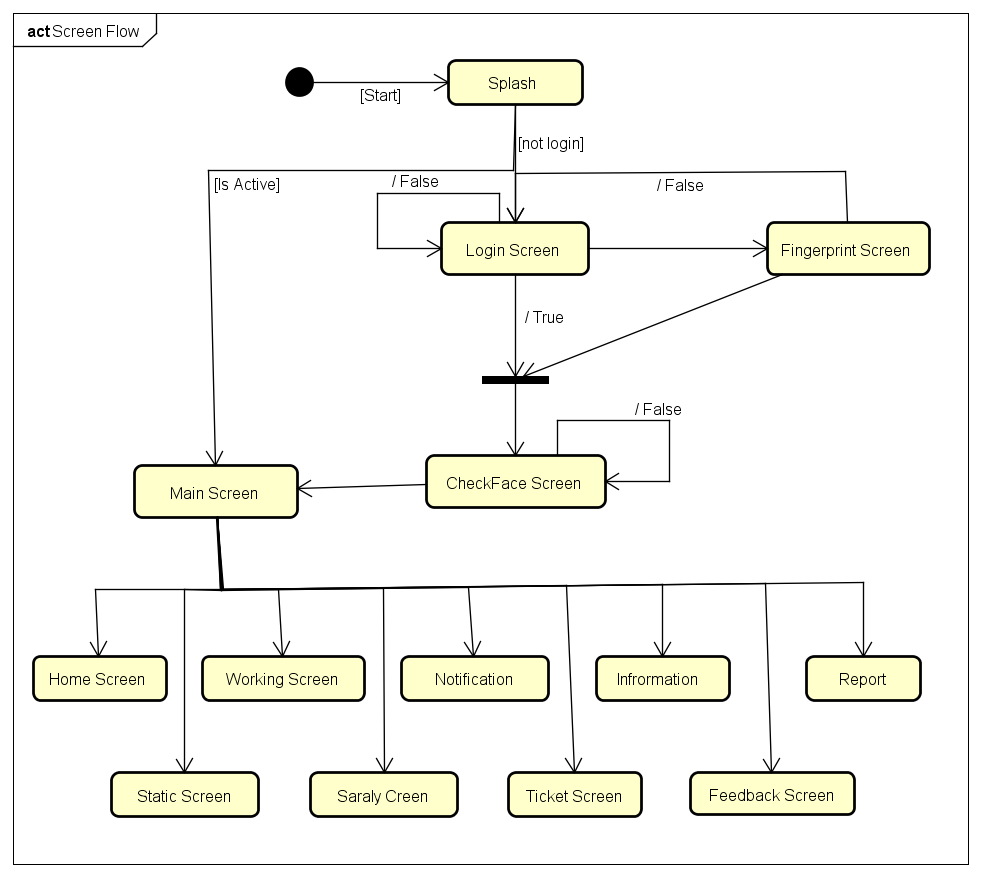
Màu sắc sử dụng hài hòa sử dụng tông màu sáng và dịu không gây ức chế cho người sử dụng. Hình ảnh phải mang tính gợi ý, không đánh đố người dùng. Các giao diện phải sử dụng đồng dạng các thiết kế, âm báo phải giống nhau cho các thông báo giống nhau.

Ứng dụng phải đảm bảo tính phản hồi, khi chạm vào các button thì phải có thông báo cho người dùng như có hiệu ứng khi ấn vào hoặc đơn giản là sáng lên tuy nhiên cần thống nhất chung cho các button có nhiệm vụ giống và tương tự nhau để giảm thiểu tối đa thời gian học hệ thống của người dùng. Khi có tiến trình hoạt động và yêu cầu người dùng chờ thì phải có thông báo hiển thị.

Ứng dụng phải đảm bảo các rằng buộc về vật lý và logic và ánh xạ chính xác. Tóm lại giao diện cần thiết kế một cách nhất quán và xuyên suốt từ đâu đến cuối ứng dụng cung cấp đầy đủ phím tắt và sự phản hồi cũng như phải hỗ trợ được người dùng kiểm soát được hệ thống.

#### Thiết kế màn hình

##### Sơ đồ dịch chuyển màn hình

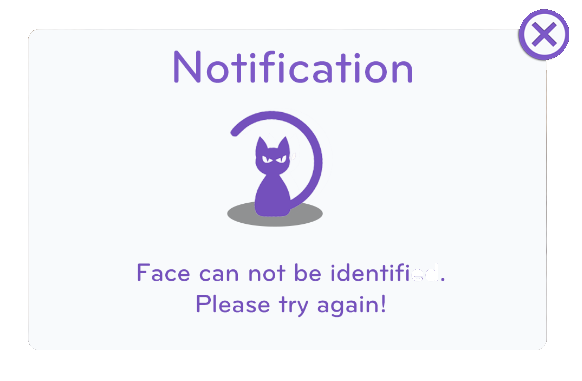


Hình 4.12 Biểu đồ dịch chuyển màn hình

Hình 4.12 Biểu đồ dịch chuyển màn hình mô tả chi tiết đường đi và các màn hình của hệ thống từ lúc đăng nhập đến các phiên sử dụng trong hệ thống.

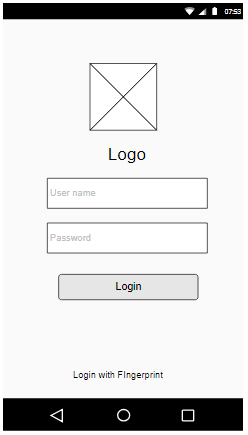
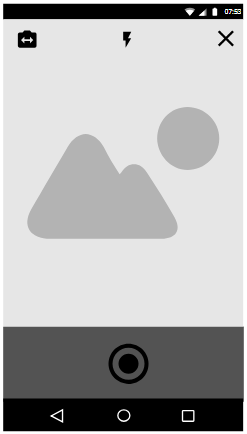
##### Thiết kế màn hình

Tất cả các thiết kế đường vẽ bằng Photoshop phiên bản CC 2018 để tạo nên sự thuận tiện nhất cho việc thiết kế giao diện sau này. Dưới đây là một vài form thông báo cũng như một số thiết kế mà hình cơ bản.



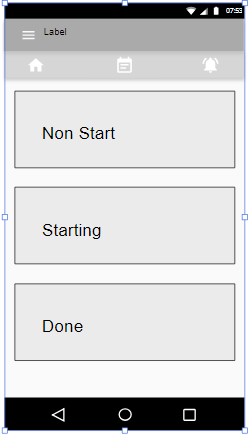
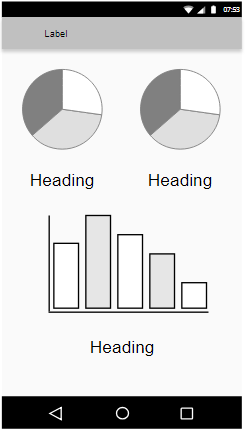
Hình 4.13 Form thông báo hệ thống

Hình 4.13 Form thông báo hệ thống là các thiết kế cơ bản về các thông báo hệ thống bao gồm thông báo lỗi, các thông báo xác nhận tới người dùng.

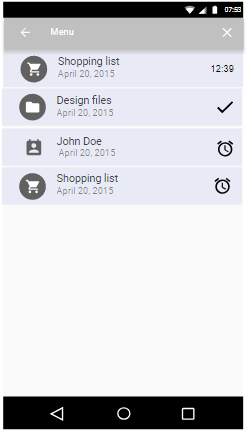
Hình 4.14 Màn hình đăng nhập và màn hình kiểm tra khuôn mặt

Hình 4.14 là bản thiết kế cho giao diện đăng nhập và giao diện chụp và kiểm tra khuôn mặt của người dùng trong mỗi phiên điểm danh và đăng nhập.

Hình 4.15 Màn hình quản lý công việc và màn hình thống kê

Hình 4.15 là những thiết kế cơ bản cho màn hình quản lý công việc, và thống kê trong hệ thống bao gồm thống kê công việc và thống kê giờ làm việc.

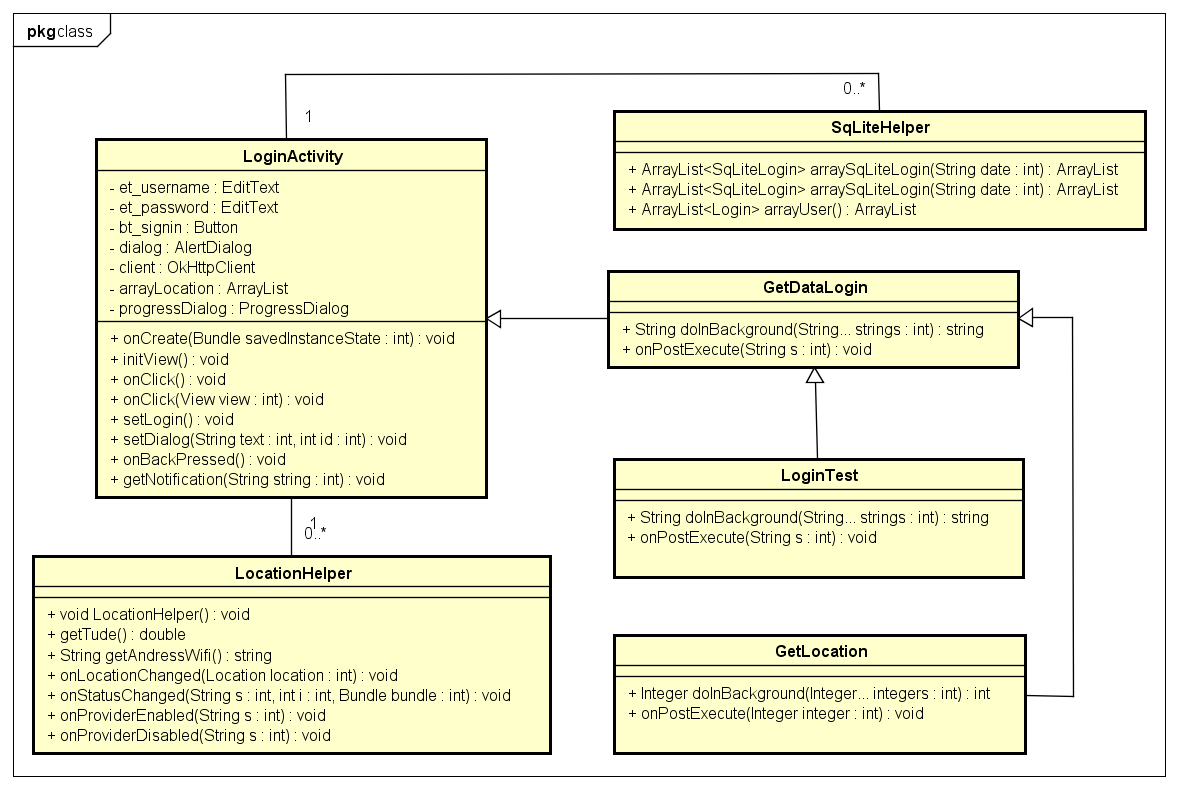
Hình 4.16 Màn hình quản lý task

### Thiết kế lớp

#### Thiết kế lớp chi tiết

Dưới đây là mô tả chi tiết của một số lớp đặc trưng trong hệ thống.

Lớp LoginActivity



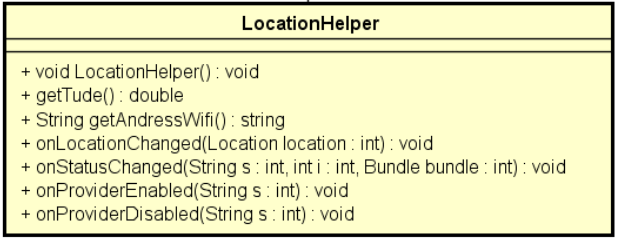
Hình 4.17 Biểu đồ lớp chi tiết LoginActivity

Hình 4.17 mô tả chi tiết các thuộc tính phương thức và các phụ thuộc nó trong quá trình xây dựng.

Bảng 4.1 Bảng đặc tả các phương thức trong lớp LoginActivity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tên lớp: | | ActivityLogin | | | |
| Danh sách thuộc tính | | | | | |
| Tên thuộc tính | **Kiểu dữ liệu** | | | **Ý nghĩa** | |
| et\_username | EditText | | | Ô nhập dữ liệu đầu vào user name | |
| et\_password | EditText | | | Ô nhập dữ liệu đầu vào password | |
| bt\_singin | Button | | | Button đăng nhập hệ thống | |
| dialog | AlertDialog | | | Thông báo hiển thị, thông báo lỗi | |
| client | OkHttpClient | | | Giao thức kết nối server | |
| Danh sách phương thức | | | | | |
| Tên phương thức | | **Đầu vào** | **Đầu ra** | | **Ý nghĩa** |
| onCreate | | Bundle | void | | Phương thức khởi tạo ban đầu khi gọi đến Activity |
| initView() | | void | void | | Phương thức ánh xạ tới tới từng thành phần giao diện của acitvity |
| onClick() | | void | void | | Khởi tạo phương thức click tại activity |
| onClick | | View | void | | Cài đặt các sự kiện click trong acitvbity |
| setLogin() | | void | void | | Gọi đến lớp xử lý đăng nhập |
| setDialog | | String | void | | Hiển thị các thông báo lỗi. |

Lớp LocationHelper



Hình 4.18 Sơ đồ lớp LocationHelper

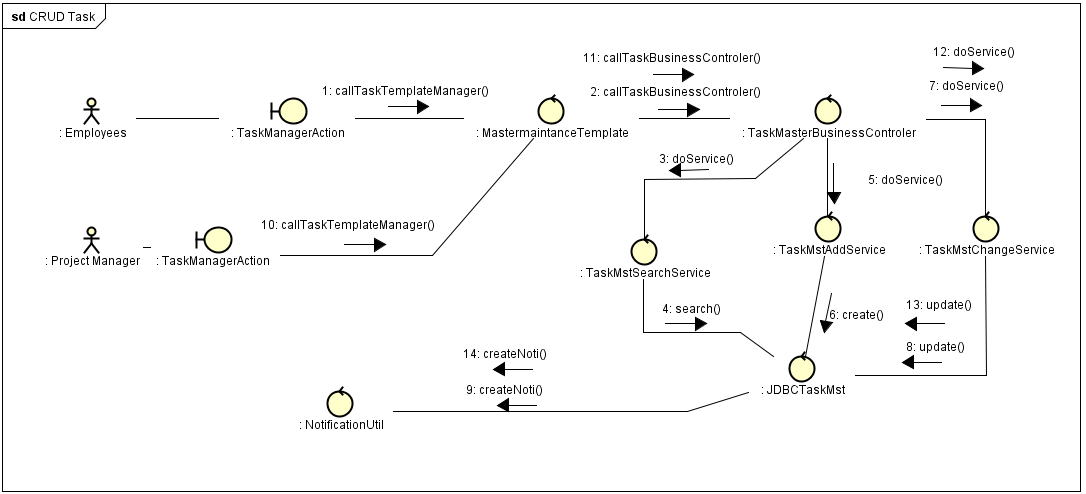
Hình 4.18 Sơ đồ lớp LocationHelper mô tả chi tiết các phương thức trong lớp LocationHeler, nơi giúp đỡ lấy ra vị trí hiện tại của người dùng. Tên các phương thức, hàm trong lớp sẽ được mô tả chi tiết tại Bảng 4.2 Bảng đặc tả các phương thức trong lớp LocationHelper

Bảng 4.2 Bảng đặc tả các phương thức trong lớp LocationHelper

|  |  |  |  |
| --- | --- | --- | --- |
| Tên lớp: | ActivityLogin | | |
| Danh sách phương thức | | | |
| Tên phương thức | **Đầu vào** | **Đầu ra** | **Ý nghĩa** |
| LocatiobHelper() | void | void | Khởi tạo các loation service |
| getTude() | void | void | Phương thức trả về tọa độ vị trí của người dùng |
| getAndressWifi() | void | String | Phương thức trả về địa chỉ wifi đang được kết nối |
| onLocationChanged() | Location | void | Phương thức cập nhật lại vị trí người dùng. |
| onStatusChanged() | Bundel | void | Trả về các thông báo hệ thống |
| onProviderEnabled | String | void | Thông báo khởi tạo GPS |
| onProviderEnabled | String | void | Phương thức khởi tạo kết nối GPS |
| onProviderDisabled | String | void | Thông báo đóng kết nối GPS |

#### Thiết kế luồng thông điệp

Dưới đây là một số biểu đồ giao tiếp nổi bật của hệ thống nhằm làm rõ các luồng thông điệp.



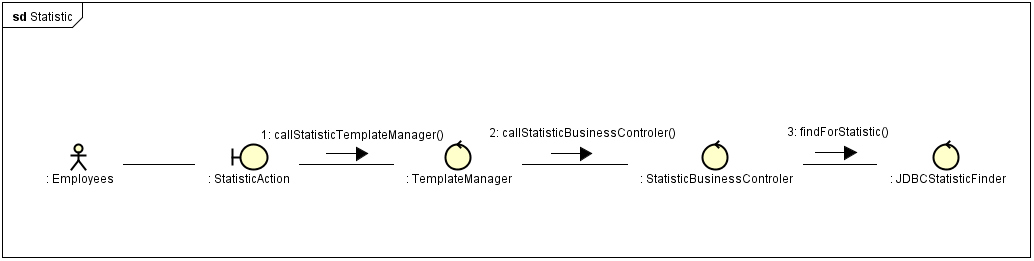
Hình 4.19 Biểu đồ giao tiếp quản lý task

Thứ tự luồng thông điệp 1 – 2 – 3 – 4 thể hiện chức năng tìm kiếm công việc (Task).

Thứ tự luồng thông điệp 1 – 2 – 5 – 6 thể hiện quá trình tạo mới công việc (Task).

Thứ tự luồng thông điệp 1 – 2 – 7 – 8 – 9 thể hiện quá trình cập nhật thông tin hoặc cập nhập trạng thái từ người tạo công việc hoặc người giao công việc, sau đó được gửi thông báo tới những người liên quan thông qua lớp NotificationUtil.

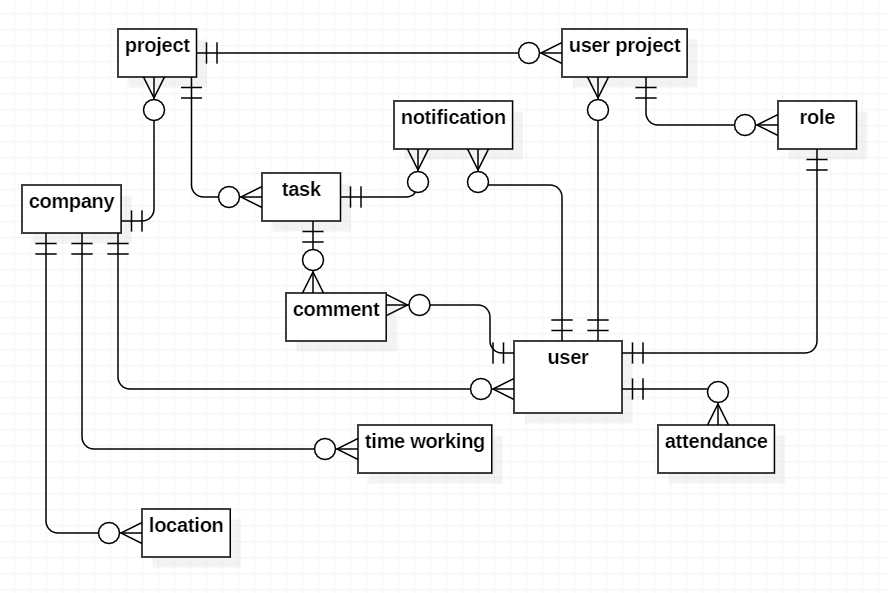
Thứ tự luồng thông điệp 10 – 11 – 12 – 13 – 14 thể hiện quá trình cập nhật trạng thái cho công việc của người quản lý dự án (ProjectManager) sau đó thông báo sẽ được gửi đến người tạo và người được giao công việc



Hình 4.20Biểu đồ giao tiếp thống kê

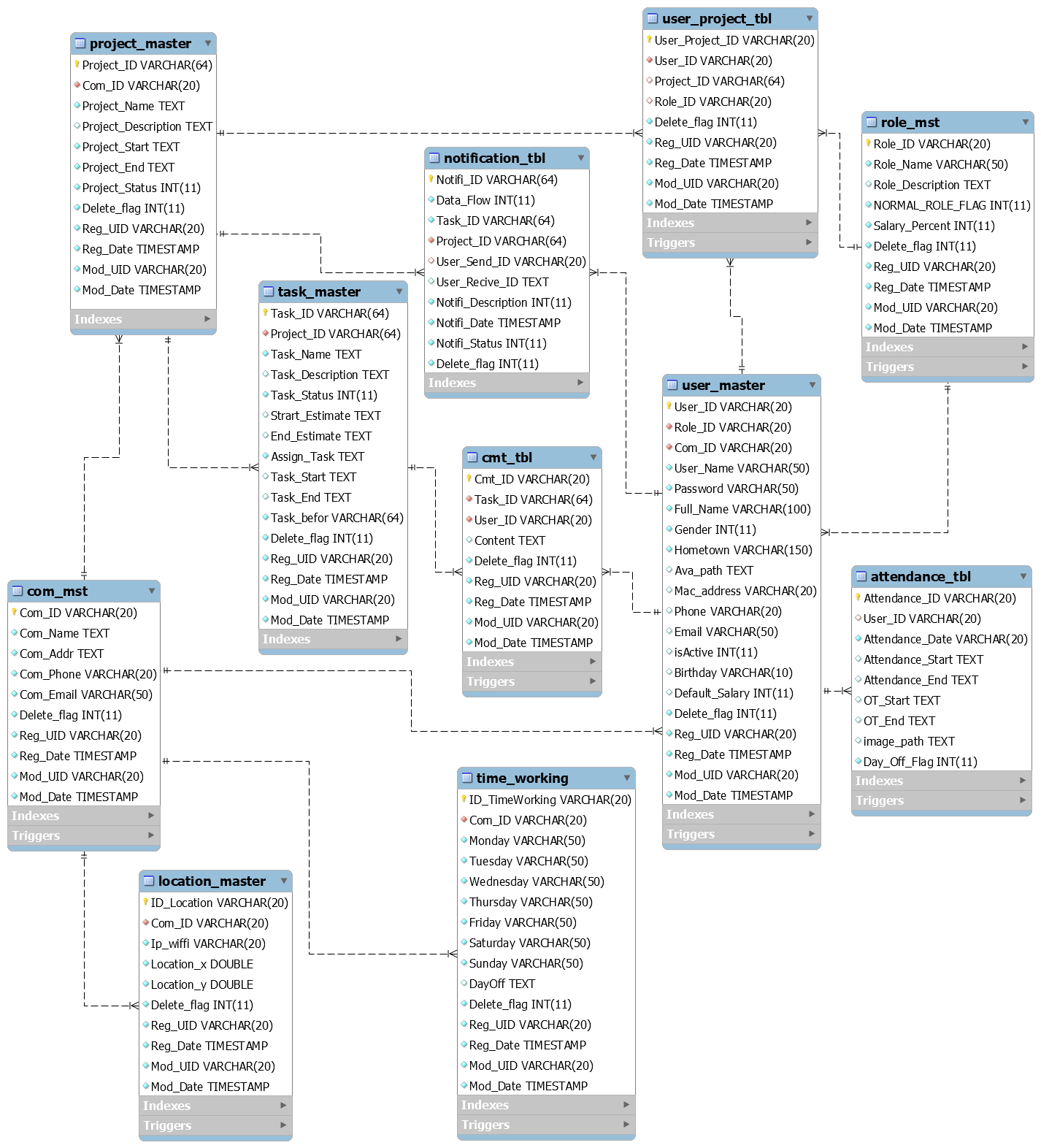
Luồng thông điệp 1 – 2 – 3 là quá trình truy vấn lấy dữ liệu của chức năng thống kê.

### Thiết kế cơ sở dữ liệu



Hình 4.21 Sơ đồ thực thể liên kết (ER)

Hình 4.21 Sơ đồ thực thể liên kết (ER) mô tả sơ đồ mối quan hệ của cơ sở dữ liệu trong hệ thống. Một người dùng (User) có thể chứa nhiều dữ liệu về điểm danh (Attendance), nhận được nhiều thông báo (Notification), sở hữu nhiều công việc (Task) và các thẻ yêu cầu (Ticket), người dùng (User) chỉ có một chức vụ (Role). Ngược lại, một dự án (Project) sẽ có nhiều người dùng (User) và các công việc phải làm (Task). Cở sở dữ liệu được xây dựng trên hệ quản trị cơ sở dữ liệu MySQL.



Hình 4.22 Mô hình hình CSDL hệ thống

Hình 4.22 là sơ đồ tổng quan về thiết kế CSDL của hệ thống, chi tiết hơn về thiết kế sẽ được trình bày rõ ràng hơn từ Bảng 4.3 đến Bảng 4.13

Bảng 4.3 Thiết kế chi tiết bảng com\_mst:

| Thuộc tính | Kiểu dữ liệu | Kích thước | Giải thích | Ràng buộc |
| --- | --- | --- | --- | --- |
| Com\_ID | Varchar | 20 | Id công ty | Khóa chính |
| Com\_Name | Text | - | Tên công ty | - |
| Com\_Addr | Text | - | Địa chỉ công ty | - |
| Com\_Phone | Varchar | 20 | Số điện thoại công ty | - |
| Com\_Email | Varchar | 50 | Email công ty | - |
| Delete\_flag | Int | 11 | Cờ xóa | - |
| Reg\_UserID | Varchar | 20 | Id người tạo | - |
| Reg\_Date | Datetime | - | Thời gian tạo | - |
| Mod\_UserID | Varchar | 20 | Id người sửa | - |
| Mod\_date | Datetime | - | Thời gian sửa | - |

Bảng 4.4 Thiết kế chi tiết bảng Task\_master

| Thuộc tính | Kiểu dữ liệu | Kích thước | Giải thích | Ràng buộc |
| --- | --- | --- | --- | --- |
| Task\_ID | Varchar | 10 | Id công việc | Khóa chính |
| Project\_ID | Varchar | 10 | ID dự án | Khóa ngoài |
| Task\_Name | Varchar | 50 | Tên công việc | - |
| Task\_description | Text | - | Mô tả | - |
| Task\_Status | Int | 11 | Trạng thái của công việc:  1 - Open  2 - Process  4 - Close  6 - Done  8 - Pending | - |
| Task\_Priority | Int | 11 | Loại công việc:  1 - Chính  2 - Phụ  3 - Cần gấp | - |
| Task\_Estimate | Int | 11 | Thời gian dự kiến hoàn thành | - |
| Assign\_Task | Varchar | 10 | ID người được giao việc | - |
| Task\_Start | Text | - | Những thời gian công việc được bắt đầu | - |
| Task\_End | Text |  | Những thời gian ngừng công việc | - |
| Delete\_flag | Int | 11 | Cờ xóa | - |
| Reg\_UserID | Varchar | 20 | Id người tạo | - |
| Reg\_Date | Datetime | - | Thời gian tạo | - |
| Mod\_UserID | Varchar | 20 | Id người sửa | - |
| Mod\_date | Datetime | - | Thời gian sửa | - |

Bảng 4.5 Thiết kế chi tiết bảng Project\_master

| Thuộc tính | Kiểu dữ liệu | Kích thước | Giải thích | Ràng buộc |
| --- | --- | --- | --- | --- |
| Project\_ID | Varchar | 10 | ID dự án | Khóa chính |
| Com\_ID | Varchar | 50 | ID Công ty | Khóa ngoài |
| Project\_Name | Text | - | Tên dự án | - |
| Project\_description | Text | - | Mô tả | - |
| Project \_Start | Text | - | Thời gian bắt đầu làm dự án | - |
| Project\_End | Text | - | Thời gian kết thúc dự án | - |
| Project \_Status | Int | 11 | Trạng thái của công việc:  0 – Chưa thực hiện  1 – Đang thực hiên  2 – Tạm dừng  3 – Đã đóng | - |
| Delete\_flag | Int | 11 | Cờ xóa | - |
| Reg\_UserID | Varchar | 20 | Id người tạo | - |
| Reg\_Date | Datetime | - | Thời gian tạo | - |
| Mod\_UserID | Varchar | 20 | Id người sửa | - |
| Mod\_date | Datetime | - | Thời gian sửa | - |

Bảng 4.6 Thiết kế bảng location\_master

| Thuộc tính | Kiểu dữ liệu | Kích thước | Giải thích | Ràng buộc |
| --- | --- | --- | --- | --- |
| Location\_ID | Varchar | 10 | ID vị trí | Khóa chính |
| Com\_ID | Varchar | 50 | ID Công ty | Khóa ngoài |
| Ip\_wifi | Varchar | 20 | Địa chỉ Ip wifi công ty | - |
| Location\_x | Float | - | Vị trí kinh độ | - |
| Location\_y | Float | - | Vị trí vĩ độ | - |
| Delete\_flag | Int | 11 | Cờ xóa | - |
| Reg\_UserID | Varchar | 20 | Id người tạo | - |
| Reg\_Date | Datetime | - | Thời gian tạo | - |
| Mod\_UserID | Varchar | 20 | Id người sửa | - |
| Mod\_date | Datetime | - | Thời gian sửa | - |

Bảng 4.7 Thiết kế bảng time\_working

| Thuộc tính | Kiểu dữ liệu | Kích thước | Giải thích | Ràng buộc |
| --- | --- | --- | --- | --- |
| TimeWorking\_ID | Varchar | 10 | ID Thời gian làm việc | Khóa chính |
| Com\_ID | Varchar | 50 | ID Công ty | Khóa ngoài |
| Monday | Varchar | 50 | Thời gian làm việc ngày thứ 2 | - |
| Tuesday | Varchar | 50 | Thời gian làm việc ngày thứ 3 | - |
| Wednesday | Varchar | 50 | Thời gian làm việc ngày thứ 4 | - |
| Thursday | Varchar | 50 | Thời gian làm việc ngày thứ 5 | - |
| Friday | Varchar | 50 | Thời gian làm việc ngày thứ 6 | - |
| Saturday | Varchar | 50 | Thời gian làm việc ngày thứ 7 | - |
| Sunday | Varchar | 50 | Thời gian làm việc ngày chủ nhật | - |
| DayOff | Text | - | Ngày nghỉ | - |
| Delete\_flag | Int | 11 | Cờ xóa | - |
| Reg\_UserID | Varchar | 20 | Id người tạo | - |
| Reg\_Date | Datetime | - | Thời gian tạo | - |
| Mod\_UserID | Varchar | 20 | Id người sửa | - |
| Mod\_date | Datetime | - | Thời gian sửa | - |

Bảng 4.8 Thiết kế bảng user\_master

| Thuộc tính | Kiểu dữ liệu | Kích thước | Giải thích | Ràng buộc |
| --- | --- | --- | --- | --- |
| User\_ID | Varchar | 10 | ID Người dùng | Khóa chính |
| Role\_ID | Varchar | 20 | ID Quyền người dùng | Khóa ngoài |
| Com\_ID | Varchar | 50 | ID Công ty | Khóa ngoài |
| User\_Name | Varchar | 50 | Tên đăng nhập | - |
| Password | Varchar | 50 | Mật khẩu đăng nhập | - |
| Full\_Name | Varchar | 100 | Họ và tên người dùng | - |
| Gender | Int | 11 | Giới tính | - |
| Hometown | Varchar | 150 | Địa chỉ quê quán người dùng | - |
| Ava\_patch | Text | - | Ảnh đại diện người dùng | - |
| Mac\_address | Varchar | 20 | Địa chỉ Mac wifi trên thiết bị di động người dùng | - |
| Phone | Varchar | 20 | Số điện thoại người dùng | - |
| Email | Varchar | 50 | Email người dùng | - |
| isActive | Int | 11 | Trạng thái hoạt động người dùng | - |
| Birthday | Varchar | 10 | Ngày sinh | - |
| Delete\_flag | Int | 11 | Cờ xóa | - |
| Reg\_UserID | Varchar | 20 | Id người tạo | - |
| Reg\_Date | Datetime | - | Thời gian tạo | - |
| Mod\_UserID | Varchar | 20 | Id người sửa | - |
| Mod\_date | Datetime | - | Thời gian sửa | - |

Bảng 4.9 Thiết kế bảng cmt\_tbl

| Thuộc tính | Kiểu dữ liệu | Kích thước | Giải thích | Ràng buộc |
| --- | --- | --- | --- | --- |
| Cmt\_ID | Varchar | 10 | ID Bình luận | Khóa chính |
| Task\_ID | Varchar | 20 | ID Tác vụ | Khóa ngoài |
| User\_ID | Varchar | 50 | ID Người dùng | Khóa ngoài |
| Content | Text | - | Nội dung bình luận | - |
| Delete\_flag | Int | 11 | Cờ xóa | - |
| Reg\_UserID | Varchar | 20 | Id người tạo | - |
| Reg\_Date | Datetime | - | Thời gian tạo | - |
| Mod\_UserID | Varchar | 20 | Id người sửa | - |
| Mod\_date | Datetime | - | Thời gian sửa | - |

Bảng 4.10 Thiết kế bảng notification\_tbl

| Thuộc tính | Kiểu dữ liệu | Kích thước | Giải thích | Ràng buộc |
| --- | --- | --- | --- | --- |
| Notifi\_ID | Varchar | 10 | ID Thông báo | Khóa chính |
| Project\_ID | Varchar | 20 | ID dự án | Khóa ngoài |
| Data\_Flow | Int | 11 | Đường dẫn thông báo | - |
| Task\_ID | Varchar | 64 | ID Tác vụ | - |
| User\_Send\_ID | Varchar | 20 | ID Người gửi thông báo | - |
| User\_Recive\_ID | Text | - | ID Người nhận thông báo | - |
| Notifi\_Description | Int | 11 | Chi tiết thông báo | - |
| Notifi\_Date | Timestamp | - | Thời gian thông báo | - |
| Notifi\_Status | Int | 11 | Trạng thái thông báo | - |
| Delete\_flag | Int | 11 | Cờ xóa | - |

Bảng 4.11 Thiết kế bảng user\_project\_tbl

| Thuộc tính | Kiểu dữ liệu | Kích thước | Giải thích | Ràng buộc |
| --- | --- | --- | --- | --- |
| User\_Project\_ID | Varchar | 20 | ID | Khóa chính |
| Role\_ID | Varchar | 20 | ID Quyền người dùng | Khóa ngoài |
| Project\_ID | Varchar | 20 | ID Công ty | Khóa ngoài |
| User\_Name | Varchar | 50 | ID Người dùng | Khóa ngoài |
| Delete\_flag | Int | 11 | Cờ xóa | - |
| Reg\_UserID | Varchar | 20 | Id người tạo | - |
| Reg\_Date | Datetime | - | Thời gian tạo | - |
| Mod\_UserID | Varchar | 20 | Id người sửa | - |
| Mod\_date | Datetime | - | Thời gian sửa | - |

Bảng 4.12 Thiết kế bảng role\_mst

| Thuộc tính | Kiểu dữ liệu | Kích thước | Giải thích | Ràng buộc |
| --- | --- | --- | --- | --- |
| Role\_ID | Varchar | 20 | ID Quyền người dùng | Khóa chính |
| Role\_Name | Varchar | 50 | Tên quyền người dùng | - |
| Role\_Description | Text | - | Mô tả quyền | - |
| NORMAL\_ROLE  \_FLAG | Int | 11 | Chỉ số quyền mặc đinh:  0: Quyền mặc định  1: Quyền mở rộng | - |
| Delete\_flag | Int | 11 | Cờ xóa | - |
| Reg\_UserID | Varchar | 20 | Id người tạo | - |
| Reg\_Date | Datetime | - | Thời gian tạo | - |
| Mod\_UserID | Varchar | 20 | Id người sửa | - |
| Mod\_date | Datetime | - | Thời gian sửa | - |

Bảng 4.13 Thiết kế bảng attendance\_tbl

| Thuộc tính | Kiểu dữ liệu | Kích thước | Giải thích | Ràng buộc |
| --- | --- | --- | --- | --- |
| Attendance\_ID | Varchar | 20 | ID Điểm danh | Khóa chính |
| User\_ID | Varchar | 20 | ID Người dùng | Khóa ngoài |
| Attendance\_Start | Text | - | Thời gian bắt đầu điểm danh | - |
| Attendance\_End | Text | - | Thời gian đăng xuất điểm danh | - |
| OT\_Start | Text | - | Thời gian bắt đầu điểm danh tăng ca | - |
| OT\_End | Text | - | Thời gian đăng xuất điểm danh tăng ca | - |
| Image\_path | Text | - | Đường dẫn ảnh | - |
| Day\_Off\_Flag | Int | 11 | Chỉ số ngày nghỉ:  0: ngày làm việc bình thường  1: ngày nghỉ | - |

## Xây dựng ứng dụng

### Thư viện và công cụ sử dụng

Bảng 4.14 Danh sách thư viện và công cụ sử dụng

|  |  |  |
| --- | --- | --- |
| **Mục đích** | **Công cụ** | **Địa chỉ URL** |
| IDE lập trình | Android Studio 3.1.2 for Windows 64-bit | https://developer.android.com/studio/ |
| Ngôn ngữ lập trình | Java 8 | <http://www.oracle.com> |
| Quản trị CSDL | MySql 8.0 | https://www.mysql.com/ |
| Công cụ giả lập máy chủ | XAMPP for Windows 7.2.5 | https://www.apachefriends.org/index.html |
| Công cụ hỗ trợ lập trình | Atom 1.27.0 for Windows 64-bit | https://atom.io/ |
| Thư viện | Layout percent | https://developer.android.com/topic/libraries/support-library/ |
| Thư viện | OkHttp3 | https://github.com/square/okhttp |
| Thư viện | Gson | https://github.com/google/gson |
| Thư viện | picasso | https://github.com/square/picasso |
| Thư viện | google-play-service | https://developers.google.com/android/guides/releases |

### Kết quả đạt được

Sau khi triển khai xây dựng ứng dụng, em đã xây dựng được thành công ứng dụng “quản lý công việc”. Bao gồm các chức năng chính 7 chức năng chính.

* Quản lý điểm dannh bằng điện thoại thông qua vị trí điểm danh.
* Quản lý dự án (CRUD Project): giúp quản lý thực hiện việc quản lý dự án từ lúc khởi tạo đến kết thúc
* Quản lý công việc (CRUD Task): Chia nhỏ các công việc trong project, giao cho từng thành viên. Nhân viên xem các task, ước lượng thời gian, quản lý dự án, trên cơ sở đó quản lý tiến độ dự án.
* Thống kê (Statistic): Giúp người dùng có được các phép thống kê cần thiết để thấy được thành phần của công ty, dự án.
* Report: Giúp người dùng báo cáo các sai phạm trong sử dụng hệ thống
* Inforuser: Xem và cập nhật thông tin cá nhân.
* Đồng bộ dữ liệu hệ thống (Đồng bộ 2 chiều client – server).

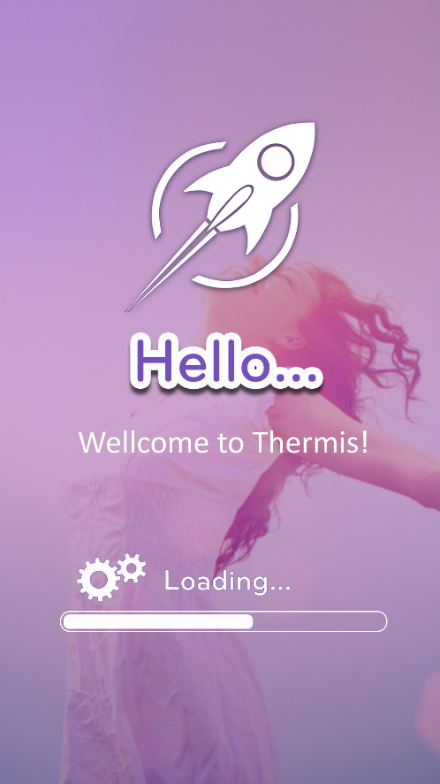
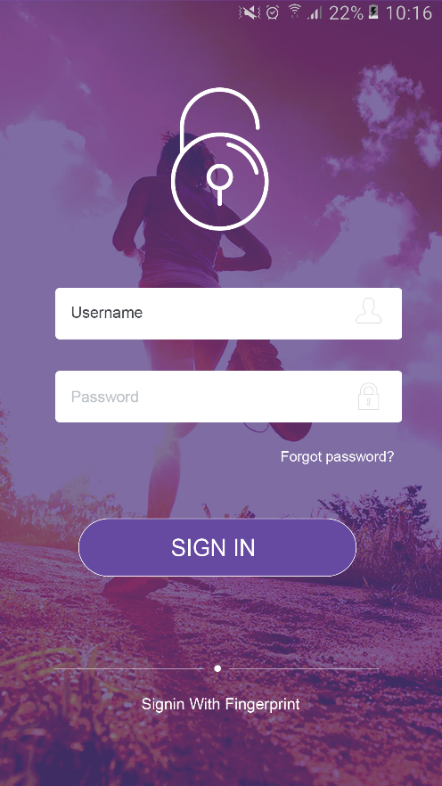
Thống kê về thông tin ứng dụng được thể hiện ở Bảng 4.15

Bảng 4.15 Thống kê thông tin ứng dụng

| Thông tin | Thống kê |
| --- | --- |
| Số lớp trong mã nguồn android | 110 class java |
| Số gói trong mã nguồn android | 25 gói |
| Số file giao diện android (xml) | 60 file xml |
| Dung lượng mã nguồn android | 200 Mb |
| Dung lượng file apk | 17Mb trên thiết bị di động |
| Số file PHP trong mã nguồn website | 11 file PHP |
| Số bảng trong cơ sở dữ liệu | 11 bảng |

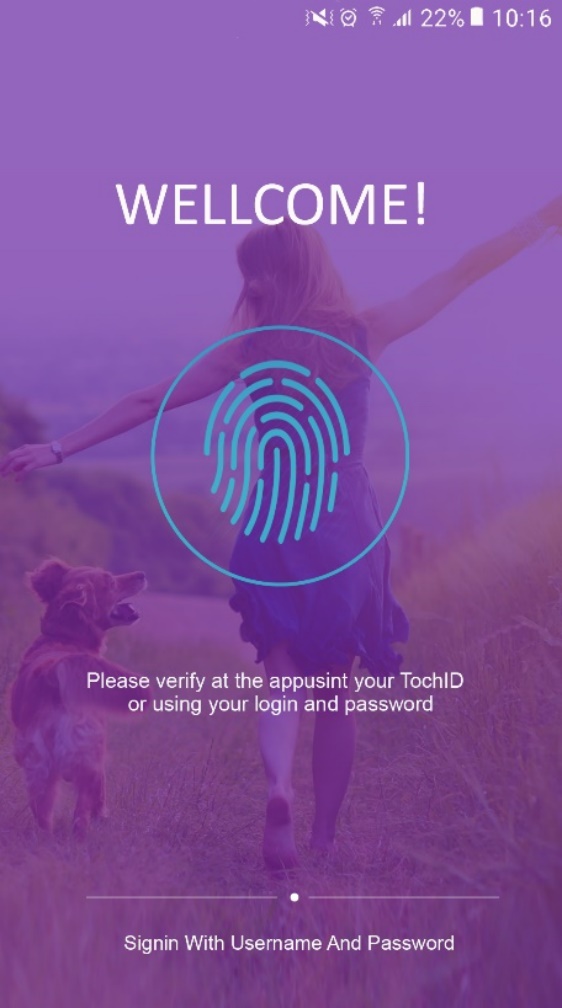
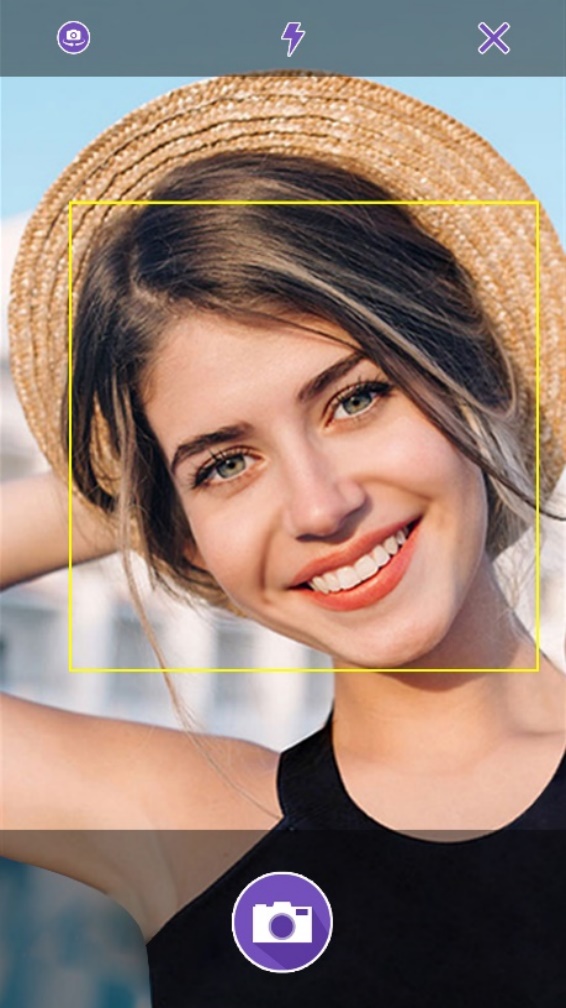
### Minh hoạ các chức năng chính

Dưới đây là một số hình ảnh minh họa cho sản phẩm sau quá trình xây dựng.

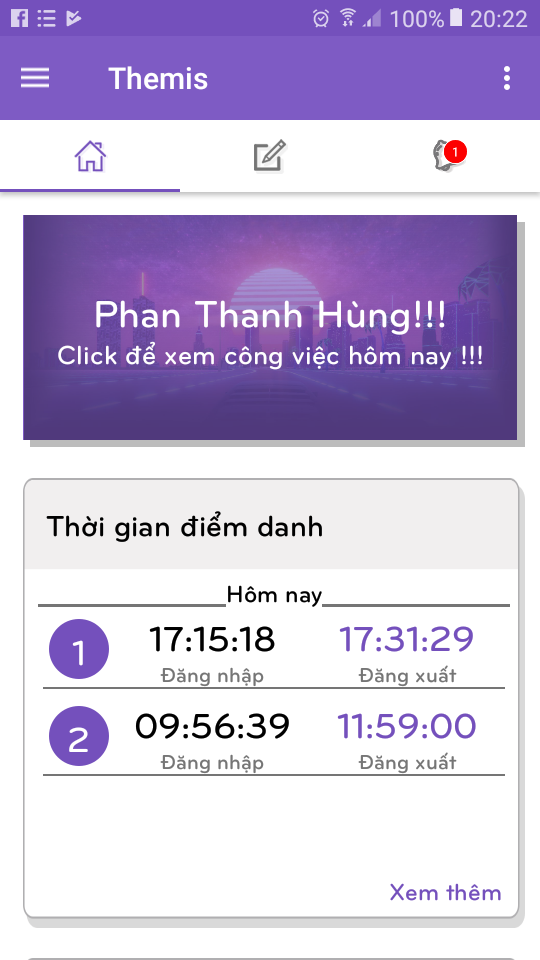
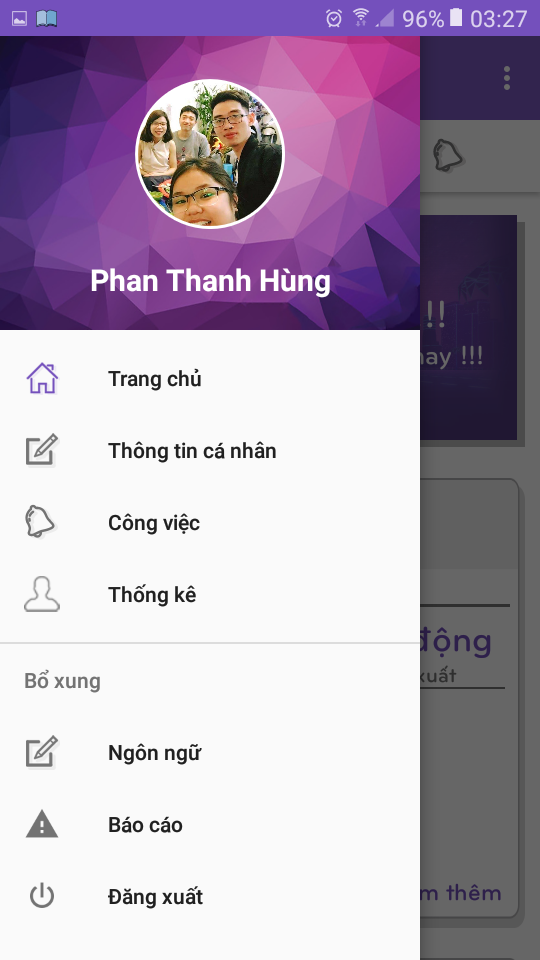
Hình 4.23 Màn hình Splash và mà hình đăng nhập

Hình 4.23 là mà hình chờ kiểm tra cập nhật lúc khởi động ứng dụng và màn hình đăng nhập đăng nhập vào hệ thống. Khi khởi động ứng dụng màn hình Splash đầu tiên sẽ được khởi tạo nhiệm vụ của nó là kiểm tra các cập nhật cần thiết cho ứng dụng hoạt động. Tiếp sau đó màn hình đăng nhập được khởi tạo, ở màn hình này bao gồm hai ô Edittext tài khoản và mật khẩu, một Button đăng nhập và Textview quên mật khẩu và đăng nhập bằng vân tay.

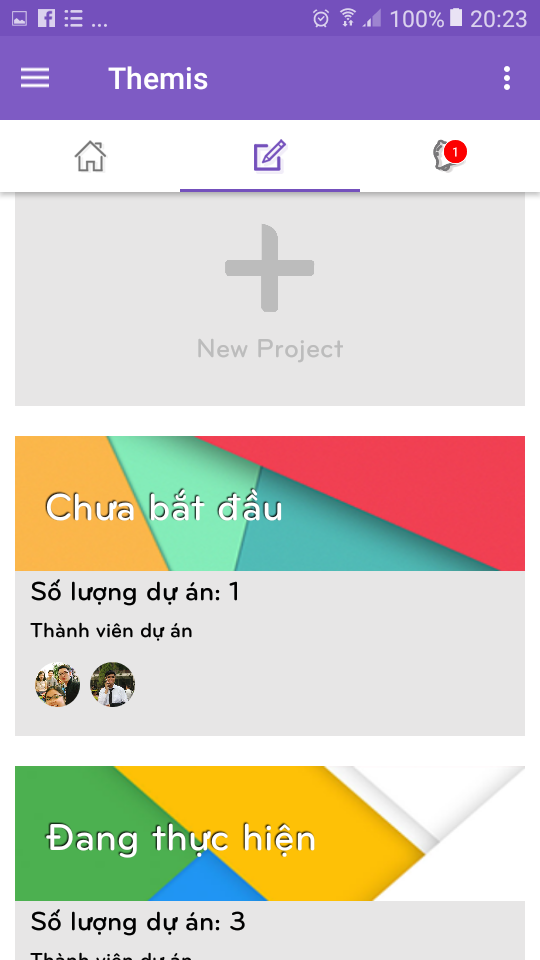
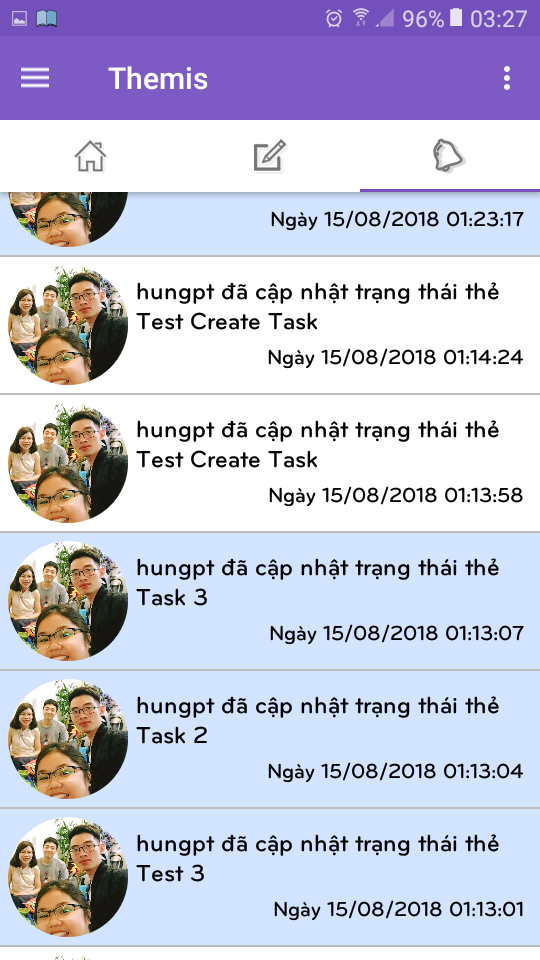
Hình 4.24 Màn hình kiểm tra vân tay và kiểm tra khuôn mặt

Hình 4.24 là hai màn hình đăng nhập ứng dụng bằng vân tay và màn hình nhận diện và chụp lại khuôn mặt. Màn hình đăng nhập ứng dụng bằng vân tay chỉ có thể khởi tạo nếu thiết bị di động hỗ trợ đăng nhập với vân tay. Sau khi đăng nhập bằng tài khoản và mật khẩu hoặc sử dụng vân tay thì màn hình chụp và kiểm tra khuôn mặt được khởi tạo. Tại đây người dung phải chụp lại khuôn mặt mình mới chính thức đăng nhập thành công.

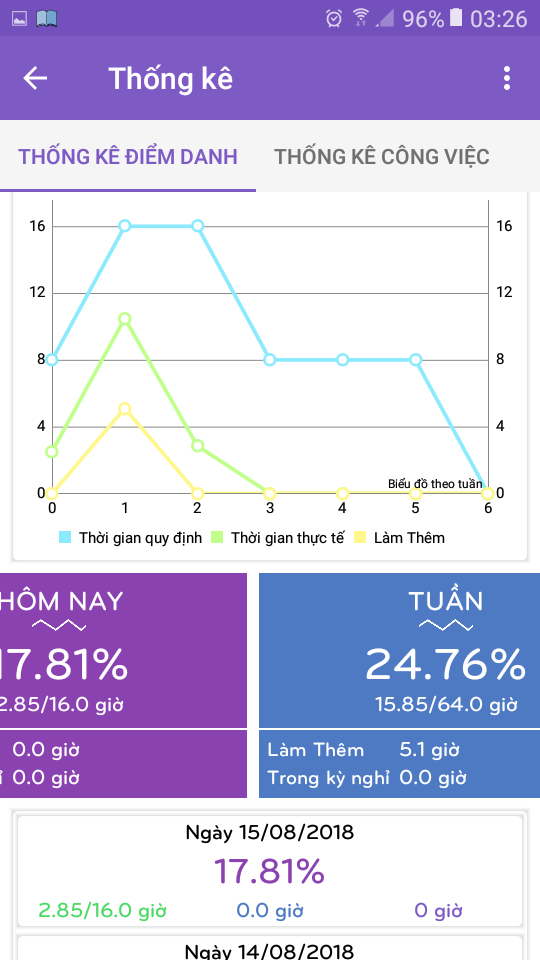
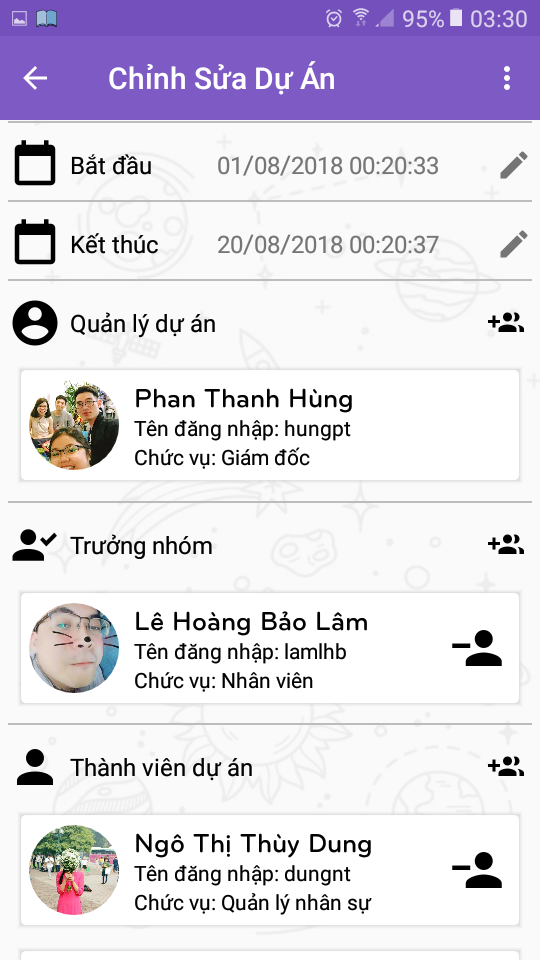
Hình 4.25 Màn hình trang chủ và navigation menu

Hình 4.25 là màn hình đầu tiên xuất hiện khi đăng nhập vào hệ thống. Tab đầu tiên là trang tổng quan về các dữ liệu đăng nhâp. Hình bên cạnh là hình thể hiện menu điều hướng của ứng dụng đến các trang chức năng.

Hình 4.26 Màn hình quản lý dự án và màn hình thông báo

Hình 4.26 là hai tab tiếp theo của màn hình chính khi đăng nhập vào hệ thống. Hình bên trái là hình tab quản lý công việc bao gồm tạo các dự án, các dự án chưa bắt đầu, các dự án đang hoạt động và các dự án đã hoàn thành. Màn hình bên phải là màn hình thông báo, bao gồm các thông báo đến từ hệ thống hoặc thông báo đến từ những người dùng khác.

Hình 4.27 Màn hình thống kê và màn hình chỉnh sửa thông tin dự án

Hình 4.27 là màn hình thống kê hệ thống (bên trái) bao gồm các biểu đồ và các thẻ thể hiện các dữ liệu điểm danh và công việc người dung theo thời gian. Hình bên phải là giao diện của chức năng chỉnh sửa thông tin dự án. Tại đây người dung có thể chỉnh sửa các thông tin cơ bản của dự án cũng như thêm thành viên và phân quyền cho dự án.

## Kiểm thử

### Load Testing

Với kiểu kiểm thử Load Testing, ứng dụng sẽ được kiểm tra phản ứng trong các điều kiện tải khác nhau. Trong phần thực hiện kiểm thử này, em sử dụng công cụ Apache Jmeter để kiểm tra khả năng đáp ứng của máy chủ trong các yêu cầu lượng truy cập khác nhau. Chi tiết về dữ kết quả kiểm thử được thể hiện trong Bảng 4.16

Bảng 4.16 Thống kê kiểm thử tải

| Chức năng | Đầu vào | Đầu ra mong muốn | Kết quả |
| --- | --- | --- | --- |
| Đăng nhập | Số lượng threads: 50  Thời gian giữa các yêu cầu: 0s  Số vòng lặp:1 | Thời gian phản hồi trung bình: <5s  Thời gian phản hồi lâu nhất: <10s  Số request xử lý trong 1 phút: >20000 request  Số request không xử lý được: <3% | Đạt |
| Đăng nhập | Số lượng threads: 100  Thời gian giữa các yêu cầu: 0s  Số vòng lặp:1 | Thời gian phản hồi trung bình: <10s  Thời gian phản hồi lâu nhất: <20s  Số request xử lý trong 1 phút: >20000 request  Số request không xử lý được: <3% | Đạt |
| Xem thông tin project | Số lượng threads: 200  Thời gian giữa các yêu cầu: 0s  Số vòng lặp:1 | Thời gian phản hồi trung bình: <15s  Thời gian phản hồi lâu nhất: <30s  Số request xử lý trong 1 phút: >15000 request  Số request không xử lý được: <3% | Đạt |

### Compatibility Testing

Với đặc trưng của hệ điều hành Android là có thể chạy trên nhiều thiết bị phần cứng khác nhau thì việc thực hiện kiểm thử khả năng tương thích sẽ giúp kiểm tra khả năng hoạt động của ứng dụng trên các thiết bị khác nhau về thông số phần cứng, phiên bản phần mềm. Chi tiết về thiết bị kiểm thử được thể hiện ở Bảng 4.17

Bảng 4.17 Thống kê kiểm thử tương thích

| Thiết bị | Thông tin thiết bị | Giao diện | Chức năng |
| --- | --- | --- | --- |
| Sony M4 | Màn hình: 5” HD, Ram: 2GB, ROM: 16GB Android 5.0, CPU: Snapdragon 615 | Đạt | Đạt |
| Samsung J2 | Màn hình: 5” HD, Ram: 1.5GB, ROM: 8GB Android 6.0, CPU: Mediateck MT6737 | Đạt | Đạt |
| Zenfone 2 | Màn hình: 5.5” HD, Ram: 1.5GB, ROM: 16GB, Android 5.0, CPU: Intel Z3560 | Đạt | Đạt |
| Samsung S6 | Màn hình: 5” QHD, Ram: 3GB, ROM: 32GB, Android 7.0, CPU: Exynos 7420 | Đạt | Đạt |
| Xiaomi Redmi Note 4X | Màn hình: 5.5” FHD, Ram: 2GB, ROM: 32GB, Android 7.0, CPU: Snapdragon 625 | Đạt | Đạt |

### Black Box Testing

#### Kiểm thử chức năng điểm danh

Bảng 4.18 Bảng kiểm thử chức năng điểm danh

| Chức năng | Giá trị nhập vào | Giá trị đầu ra cần đạt | Kết quả |
| --- | --- | --- | --- |
| Đăng nhập | Bỏ trống “User name” | Thông báo lỗi thiếu tển đăng nhập | ĐẠT |
| Đăng nhập | Bỏ trống “Password” | Thông báo lỗi thiếu Password | ĐẠT |
| Đăng nhập | Sai tên đăng nhập hoặc sai password | Thông báo sai tên đăng nhập hoặc password | ĐẠT |
| Đăng nhập | Sai vị trí điểm danh | Thông báo không trong vị trí điểm danh | ĐẠT |
| Đăng nhập | Sai địa chỉ wifi | Thông báo địa chỉ wifi không chính xác | ĐẠT |
| Đăng nhập | Sai địa chỉ MAC | Thông báo thiết bị không được hỗ trợ điểm danh | ĐẠT |
| Đăng nhập | Quá giờ làm việc | Thông báo hết giờ làm việc | Đạt |
| Đăng nhập với vân tay | Thiết bị không hỗ trợ vân tay | Thông báo thiết bị không hỗ trợ vân tay | Đạt |
| Đăng nhập với vân tay | Vân tay không phù hợp | Thông báo vân tay không phù hợp | Đạt |
| Đăng nhập với vân tay | Chưa đăng nhập bằng user name và password | Thông báo yêu cầu đăng nhập với user name và password để sử dụng chức năng này | Đạt |
| Kiểm tra khuôn mặt | Không có khuôn mặt | Thông báo không nhận diện được khuôn mặt | Đạt |
| Điểtm danh | Ngắt kết nối internet | Vẫn hoạt động bình thường | Đạt |
| Điểm danh | Ngắt server | Vẫn hoạt động bình thường | Đạt |

#### Kiểm thử cho chức năng quản lý công việc (CRUD Task)

Bảng 4.19 Bảng kiểm thử chức năng quản lý công việc

| Chức năng | Giá trị nhập vào | Giá trị đầu ra cần đạt | Kết quả |
| --- | --- | --- | --- |
| Tạo mới công việc | Bỏ trống “tên công việc” | Thông báo lỗi thiếu tên công việc | ĐẠT |
| Tạo mới công việc | Bỏ trống “mô tả” | Thông báo lỗi thiếu mô tả | ĐẠT |
| Tạo mới công việc | Bỏ trống “dự tính” | Thông báo lỗi thiếu dự tính | ĐẠT |
| Tạo mới công việc | Điền vào “dự tính”:  “test” | Thông báo lỗi sai định dạng | ĐẠT |
| Tạo mới công việc | Tên công việc: Tao TDD  Mô tả: Tai lieu TDD  Chọn dự án  Dự tính: 48  Chọn giao việc | Thông báo tạo công việc thành công | ĐẠT |
| Cập nhật thông tin công việc | Bỏ trống “tên công việc” | Thông báo lỗi thiếu tên công việc | ĐẠT |
| Cập nhật thông tin công việc | Điền vào “dự tính”:  “test” | Thông báo lỗi sai định dạng | ĐẠT |
| Cập nhật trạng thái công việc | Khi đổi trạng thái công việc sang trạng thái “thực hiện”. Người dùng có công việc khác đang ở trạng thái thực hiện. | Có thông báo confirm hiện ra. | ĐẠT |
| Cập nhật trạng thái công việc | Chọn có trong thông báo confirm | Công việc đang thực hiện được đổi trạng thái thành tạm dừng và đổi trạng thái của công việc đang yêu cầu thành thực hiện | ĐẠT |

#### Kiểm thử cho chức năng quản lý dự án (CRUD Project)

Bảng 4.20 Bảng kiểm thử chức năng dự án

| Chức năng | Giá trị nhập vào | Giá trị đầu ra cần đạt | Kết quả |
| --- | --- | --- | --- |
| Tạo mới dự án | Bỏ trống “tên dự án” | Thông báo lỗi thiếu tên dự án | ĐẠT |
| Tạo mới dự án | Bỏ trống “mô tả” | Thông báo lỗi thiếu mô tả | ĐẠT |
| Tạo mới dự án | Tên công việc: Dự án phát triển ứng dụng bán hàng  Mô tả: Dự án phát triển ứng dụng bán hàng | Thông báo tạo dự án thành công | ĐẠT |
| Cập nhật thông tin dự án | Bỏ trống “tên dự án” | Thông báo lỗi thiếu tên công việc | ĐẠT |
| Cập nhật thông tin dự án | Điền vào “dự tính”:  “test” | Thông báo lỗi sai định dạng | ĐẠT |
| Tìm kiếm nâng cao | Mã dự án: xxxxxxxx | Thông báo lỗi không có dữ liệu | ĐẠT |

### Tổng kết kiểm thử

Như vậy, trong hầu hết các trường hợp kiểm thử, những chức năng chính của hệ thống đều đạt. Những trường hợp không đạt đã trình bày ngay dưới bảng kiểm thử của từng phương pháp kiểm thử.

## Triển khai

Sau khi phát triển thành công hệ thống, hệ thống gồm 3 phần: ứng dụng trên thiết bị android, máy chủ PHP cùng với hệ quản trị cơ sở dữ liệu MySQL.

Bảng 4.21 Bảng kế hoạch triển khai

|  |  |  |
| --- | --- | --- |
| Thành phần | Kế hoạch triển khai | Yêu cầu cấu hình |
| Ứng dụng di động (android) | Đưa ứng dụng lên cửa hàng ứng dụng Google Play | Đưa ứng dụng lên cửa hàng ứng dụng Google Play  Hệ điều hành: Android 4.0 trở lên  Bộ nhớ Ram: 1GB trở lên |
| Máy chủ PHP | Chạy trên máy chủ thực tế | Hệ điều hành: Linux hoặc Window  Bộ nhớ trống: 1GB trở lên  Cài đặt: Apache, NginX, MySQL  Băng thông: lớn hơn 100GB |

# Các giải pháp và đóng góp nổi bật

Ở chương này em xin phép được trình bày bốn đóng góp nổi bật của em trong việc xử lý bài toán điểm danh và quản lý công việc. Bao gồm việc xử lý điểm danh bằng xác định vị trí và nhận diện khuôn mặt, tiếp đó là việc tích hợp điểm danh với quản lý công việc, giảm tải các gánh nặng cho server và cuối cùng là việc xử lý các xung động dữ liệu khi đồng bộ. Chi tiết về các đóng góp sẽ được trình bày dưới đây.

## Điểm danh bằng xác định vị trí và kiểm tra khuôn mặt

### Đặt vấn đề

Vấn đề điểm danh là vấn đề luôn được quan tâm hàng đầu trong việc quản lý nhân viên tuy, và có rất nhiều các phần mềm và thiết bị được tạo ra để thực hiện công việc này. Tuy nhiên có rất nhiều bất cập xảy ra, nếu nhân viên cố ý gian lận, đặc biệt trong việc nhân viên chỉ điểm danh chống đối, mà không tiếp tục làm việc ở vị trí điểm danh. Điều này khá khó để kiểm soát với các thiết bị và ứng dụng hiện tại. Chính vì vậy việc ra đời của một giải pháp khác phục các hạn chế trên là điều tất yếu.

### Giải pháp đưa ra

Trước các thực tại và vấn đề đã đặt ra ở phần 5.1.1 em đã quyết định xây dựng một ứng dụng trên thiết bị di động nhằm giải quyết các vấn đề này.

Thiết bị di động được lựa chọn bởi lẽ đây là một trong những thiết bị phổ biến nhất hiện tại mà bất cứ một ai cũng có. Giá thành không quá đắt đỏ, hơn thế phía bên công ty có thể tận dụng các thiết bị có sẵn của nhân viên để triển khai ứng dụng từ đó tiết kiệm từ đó tiết kiệm từ hàng chục đến hàng trăm triệu đồng mỗi năm để đầu tư và bảo trì các thiết bị kiểm tra và giám sát điểm danh.

Các thiết bị di động android hiện tại đều được trang bị GPS từ đó dễ dàng cho việc theo dõi vị trí người dùng. Tính đúng đắn và khả thi của việc sử dụng GPS trong việc xác định vị trí đã được nêu rõ tại mục 3.3. Chính vì vậy việc điểm danh bằng cách xác định vị trí có thể hạn chế tối đa các sai phạm trong điểm danh, đặc biệt là việc nhân viên không nằm trong vị trí làm việc được quy định trước.

Tuy nhiên GPS cũng có độ sai số nhất định (xấp xỉ 3 – 5m tính theo chiều ngang và 5 – 10m tính theo độ cao), chính vì vậy sai số có thể là khá lớn với với các công ty có quy mô nhỏ hay các công ty có thuê chung một tầng của tòa nhà. Chính vì vậy em đưa ra giải pháp là phải kiểm tra bằng đa nhân tố khác nhau, các nhân tố có thể không phải là duy nhất, tuy nhiên khi tổng hợp lại nó phải tạo thành một khóa duy nhất để có thể kiểm tra được chính xác nhất.

Các công ty cạnh nhau có thể có các địa điểm tương tự nhau nhưng mỗi công ty sẽ phải sử dụng các model wifi là không giống nhau chính vì vậy em đã đưa ra đề xuất là bên cạnh việc kiểm tra vị trí thì hệ thống còn kiểm tra địa chỉ wifi mà các thiết bị điểm danh đang kết nối vào. Từ đó sẽ hạn chế được phần nào những sai số gặp phải trong việc tính toán vị trí của GPS.

Để hạn chế việc gian lận trong việc điểm danh như người dùng để thiết bị điểm danh ngay tại vị trí điểm danh thì cứ hết mỗi phiên làm việc hệ thống sẽ tự động đăng xuất. Đồng thời giới hạn số thiết bị truy nhập vào hệ thống, hay nói cách khác mối một người dùng chỉ được đăng nhập vào từ một thiết bị bằng cách với mỗi tài khoản được cài đặt với một tài địa chỉ MAC của điện thoại di động được nhân viên đăng ký từ trước. Khi gặp sự cố người dùng muốn thay đổi địa chỉ MAC thì cần liên hệ trực tiếp với quản lý nhân sự hay người quản lý tài khoản và nhân viên tại công ty để thay đổi chứ không thể tự do thay đổi.

Khác với các hệ thống điểm danh bằng vân tay như ở các công ty, thì điện thoại di động khó lòng xác định được danh tính chính xác của người đăng nhập vì rất có thể người dùng sẽ nhờ người quen để điểm danh hộ. Để giảm thiểu khả năng này xảy ra sau khi hệ thống kiểm tra chính xác về vị trí điểm danh, địa chỉ MAC của thiết bị di động, cũng như địa chỉ wifi kết nối, hệ thống sẽ yêu tiếp tục yêu cầu người chụp lại khuôn mặt của mình và lưu trữ ảnh cộng với dữ liệu điểm danh lên server để có thể kiểm tra lại mỗi khi nghi ngờ có sai phạm.

Bên cạnh đó, hệ thống nhận diện khuôn mặt còn được được tích hợp thêm một phần trí tuệ nhân tạo. Mặc dù trí tuệ nhân tạo chỉ hoạt động ở mức hết sức cơ bản tuy nhiên nó có thể nhận diện được việc tồn tại khuôn mặt chính xác đến 98% .cơ chế hoạt động của nó đã được chỉ rõ tại mục 3.2 về thư viện hỗ trợ nhận diện khuôn mặt OpenCV. Từ đó lưu lại các kết quả điểm danh một cách chính xác nhất, hạn chế việc gian lận đến tối đa.

Để tăng trải nghiệm người dùng, giảm sự phiền toái trong sử dụng hệ thống còn tích hợp thêm phần hỗ trợ đăng nhập bằng vân tay, từ đó khiến người dùng cảm thấy thoải mái và thuận tiện nhất. Các thiết bị không có vân tay vẫn có thể đăng nhập bình thường với tên đăng nhập và mật khẩu.

Ứng dụng cho phép người dùng theo dõi trực tiếp thời gian làm việc của bản thân từ đó giải quyết sự nhập nhằng trong việc điểm danh và tạo nên sự minh bạch từ hai phía người dùng và công ty.

Ứng dụng sử dụng các service chạy ngầm bên dưới nền ứng dựng, từ đó tăng tính chịu lỗi của ứng dụng lên tối đa, ứng dụng vẫn hoạt động ngay cả khi người dùng đóng ứng dụng.

### Kết quả đạt được

Sau khi nguyên cứu và thực hiện theo các giải pháp đã được đưa ra ở mục 5.3.2 em đã thành công xây dựng nên module đăng nhập và điểm danh thông qua việc định vị vị trí, kiểm tra địa chỉ wifi kết nối, địa chỉ MAC thiết bị truy cập, và việc xác nhận khuôn mặt. Tuy quy trình của việc kiểm tra và xác định đăng nhập khác phác tạp và dài dòng, nhưng thực tế thời gian thao tác thực tế của người dùng chỉ mất vài giây để thực hiện, không hề tốn thời gian hơn quá nhiều so với các thiết bị điểm danh truyền thống.

## Tích hợp điểm danh và quản lý công việc

### Đặt vấn đề

Hầu hết các công cụ, cũng như thiết bị điểm danh hiện tại chỉ có thể quản lý công việc hoắc chỉ có thể điểm danh. Hoặc có thể kết hợp cả hai nhưng việc kết hớp này lại khá lỏng lẻo và tính chính xác không cao. Vì vậy tại hệ thống này em mong muốn nó phải giải quyết được vấn đề khi hết thời gian làm việc thì các công việc đang thực phải được tạm dừng tránh việc khi ngưng làm việc mà các task công việc vẫn bị tính giờ.

### Giải pháp đưa ra

Học hỏi các công nghệ đi trước như Trello và Zoho, cũng như sửa đổi một số lỗi trong các thức hoạt động của chúng từ đó thành công xây dựng module quản lý công việc với các ưu điểm thực tế trực quan, dễ sử dụng, phân quyền người dùng hết sức chặt trẽ. Trong cùng một khoảng thời gian người dùng sẽ chỉ được thực hiện một công việc nhất định nếu như chưa kết thúc một công việc người dùng lại khởi tạo một công việc mới thì hệ thống sẽ đưa ra thông báo và yêu cầu người dùng tạm dừng một trong hai công việc đang chọn vì thực tế không một ai có thể đồng thời thực hiện hai công việc một lúc.

Với nhân viên bình thường chỉ được phép thao tác với các task mình tạo ra, chỉ có quản lý dự án mới có quyền thao tác với tất cả các task. Khi nhân viên xác nhận đã hoàn thành task đồng hồ đếm giờ của task sẽ dừng lại và ghi nhận thời gian làm việc. Khi đó hệ thống sẽ gửi một thông báo tới PM và yêu cầu xác nhận mức độ hoàn thành của task. Nếu PM đánh giá task chưa đạt yêu cầu, thì hệ thống sẽ chuyển task đó quay về trang trạng thái chờ và thông báo cho nhân viên biết. Khi nhân viên bắt đầu tiếp tục làm task đó đồng hồ sẽ tiếp tục đếm giờ kể từ thời điểm người dùng dừng task đó lại tạo nên một sự chính xác cao trong việc quản lý công việc.

Một nhân viên có thể tham gia vào nhiều các dự án khác nhau và ở các vị trí khác nhau. Hệ thống sẽ ghi nhận lại thông qua việc phân quyền người dùng từ đó tạo nên những đánh giá chân thực và chính xác hơn hẳn các hệ thống hiện có trên thị trường.

Hai module điểm danh và quản lý công việc này được xây dựng hoàn toàn độc lập tuy nhiên chúng có thể liên kết với nhau để tạo nên sự chính xác hơn trong việc đánh giá hiệu năng làm việc của nhân viên.

Cứ kết thúc giờ làm việc hệ thống sẽ tự động đóng tất cả các task đang hoạt động và chuyển nó về trạng thái chờ đồng thời lưu lại thời gian đã làm của task để trong phiên làm việc tiếp theo người dùng có thể tiếp tục làm việc với task đó mà không phát sinh thêm các khoảng thời gian thừa, từ đó góp phần đánh giá đúng đến 80% thời gian làm việc thực của task.

Khi thực hiện tính lương hệ thống sẽ thực hiện việc đối chiếu giữa thời gian làm việc trên công việc, và thời gian điểm danh, từ đó tính toán chính xác được thời gian làm việc thực tế của người dùng. Trên cơ sở đó công hệ thống có ưu thế hơn hẳn so với các hệ thống có cùng quy mô và thậm chí cả một vài hệ thống có quy mô lớn hơn.

### Kết quả đạt được

Sau thời gian tìm hiểu và xây dựng hệ thống đã được hoàn thiện và triển khai thành công một số chức năng sau:

Bảng 5.1 Bảng chức năng hệ thống

| STT | Nhóm CN | Tên CN | Mô Tả | NV | HR | Manage | PM |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Điểm danh | Điểm danh | Ghi nhận thời gian check in, check out | x | x | x | x |
| Theo dõi số giờ làm việc, số giờ nghỉ. | Cho phép theo dõi số giờ làm việc của người dùng | x | x | x | x |
| Tính lương | Xem và ước tính lương | x | x | x | x |
| Tạo ticket | Cho phép người dùng tạo các đơn xin nghỉ việc, đến muộn. | x | x | x | x |
| Tạo report | Cho phép người dùng báo cáo sai phạm trong việc điểm danh | x | x | x | x |
| Check report | Kiểm tra các report được gửi về hệ thống. |  | x |  |  |
| Check ticket | Phê duyệt các đơn xin nghỉ, đến muộn. |  |  | x | x |
| 2 | Thông bá0 | Thông báo từ hệ thống | Các thông báo được tự sinh bởi hệ thống | x | x | x | x |
| Gửi thông báo | Tạo các thông báo đột xuất hay các thông báo định kỳ. |  | x | x | x |
| Nhận thông báo | Nhận các thông báo từ hệ thống cũng như các thông báo được gửi từ các cấp. | x | x | x | x |
| 3 | Quản lý công việc | Tạo project mới | Tạo ra các dự án mới khi có đồng thời thêm người dùng vào hệ thống cũng như phân quyền project manager cũng như các vị trí trong dự án |  |  | x |  |
| Update thông tin project | Thêm các đặc tả, dealine cũng như trạng thái của project |  |  | x | x |
| Improt, Export project | Improt, Export project từ (ra) Excel |  |  | x |  |
| Tạo task cho project | Tạo các task công việc cho từng project |  |  | x | x |
| Update các task trong project | Update thông tin các task | x |  | x | x |
| 4 | Thống kê | Thống kê số giờ làm việc, | Nhân viên thống kê số giờ làm việc theo thời gian | x | x | x | x |
| Thống kê project | Thống kê số project đã hoàn thành, đang thực hiện theo thời gian |  |  | x |  |

Hệ thống sau khi xây dựng đã chạy ổn định trên thực tế, hỗ trợ hầu hết các thiết bị android và đã được kiểm nghiệm từ phiên bản API 18 cho đến phiên bản API 28 (Andorid P). Hệ thống đã giải quyết được phần lớn các bài toán đặt ra ở phần 5.1.1 bao gồm vấn đề quản lý điểm danh, vấn đề trong việc quản lý công việc, tính lương, thưởng dựa trên việc gắn bó tích hợp hai module điểm danh và quản lý công việc.

## Giảm tải gánh nặng cho server

Hệ thống được xây dựng với mong muốn triển khai mới quy mô lớn dùng cho nhiều công ty, và doanh nghiệp và các đối tượng sử dụng khác nhau, chính vì vậy số lượng người dùng là rất lớn, số lượng request và reponse là vô cùng lớn vì vậy nếu chỉ xây dựng theo mô hình client server thông thường thì gánh nặng cho server là rất lớn bởi lẽ server sẽ phải xử lý rất nhiều các yêu cầu, gây nên việc quá tải, và tác nghẽn cho server là điều khó tránh khỏi. Chính vì vậy hệ thống em mong muốn xây dựng phải giảm thiểu được tối đa gánh nặng trong việc xử lý dữ liệu cũng như các request từ các client.

### Đặt vấn đề

Hiện tại đa số các ứng dụng trong thực tế đều được xây dựng dựa trên mô hình client – server tuy nhiên với sự phát triển ngày càng mạnh mẽ của công nghệ thông tin thì mô hình này càng bộc lộ nhiều nhược điểm.

Mô hình client – server đòi hỏi cấu hình của máy chủ phải cao, nếu có quá nhiều client truy cập, server không thể cung cấp trả kết quả cho client, server có thể bị treo. Để đảm bảo tính an toàn trên mạng cho nên server này còn có vai trò như là một nhà quản lý toàn bộ quyền truy cập dữ liệu của các máy clients, nói cách khác đó là vai trò quản trị mạng. Hơn thế tính bảo mật dữ liệu không cao. Nếu máy chủ lỗi thì toàn bộ hệ thống sẽ đi xuống. Vì vậy vai trò máy chủ rất cao. Luôn luôn phải trong trạng thái tốt.

Trong thực tế không phải lúc nào máy chủ cũng có thể hoạt động hoặc đường truyền internet không ổn định. Từ đó hệ sẽ dẫn đến hiện trạng ứng dụng hoạt động chậm, tốn tài nguyên, thậm chí không phản hồi, gây nên sự ức chế, khó chịu và thậm chí ảnh hưởng tới toàn bộ tiến độ làm việc của công ty hay dự án. Hơn nữa việc đầu tư cho server hệ thống lớn cũng là cả một gánh nặng về kinh tế với các công ty start up và các công ty nhỏ.

### Giải pháp đưa ra

Từ vấn đề đặt ra ở mục trên dựa theo kinh nghiệm và kiến thức nghiên cứu được trong thời gian qua em đã đưa ra một số giải pháp. Ứng dụng được xây dựng vẫn tiếp tục được xây dựng trên mô hình client – server tuy nhiên sẽ áp dụng thêm các mô hình phát triển phần mềm phân tán cũng như CSDL phân tán nhằm giải quyết vấn đề tồn đọng của mô hình clent – server.

Việc triển khai mô hình phân tán sẽ giúp khái niệm các công việc phức tạp và xử lý nó trên nhiều thiết bị khác nhau. Các công việc này sẽ là vô cùng to lớn nếu để máy chủ giải quyết bởi lẽ, máy chủ phải chịu tải không chỉ là từ một thiết bị mà là từ rất nhiều các thiết bị khác nhau, như vậy dữ liệu sẽ tăng lên theo cấp số nhân. Tuy nhiên nếu chia sẻ việc sử lý ra các thiết bị khác nhau thì với mỗi thiết bị chỉ phải sử lý một lượng dữ liệu hữu hạn. Hơn thế các thiết bị di động hiện tại với cấu hình ngày càng được nâng cao, từ đó khả năng và tốc xử lý của nó khá là mạnh mẽ, chính vị vậy nếu như dữ liệu được xử lý trực tiếp tại client thì hoàn toàn không ảnh hường đến hiệu năng và trải nghiệm của người dùng. Từ đó tận dụng được không gian bộ nhớ và quá trình xử lý trên từng thiết bị, server chỉ quản lý các tiến trình và cập nhật thông tin theo định kỳ từ đó giảm nhẹ đi gánh nặng về công việc cho server chính.

### Kết quả đạt được

Sau khi thực hiện theo mô hình xây dựng trên em đã thực hiện việc test thử chịu tải đối với verver và kết quả đạt được khá khả quan trong việc giảm gánh nặng cho server. Kết quả test kiểm thử được thể hiện rõ ở bảng sau:

Bảng 5.1 Kết quả test chịu tải của máy chủ

| Chức năng | Đầu vào | Đầu ra mong muốn | Kết quả |
| --- | --- | --- | --- |
| Đăng nhập | Số lượng threads: 50  Thời gian giữa các yêu cầu: 0s  Số vòng lặp:1 | Thời gian phản hồi trung bình: <5s  Thời gian phản hồi lâu nhất: <10s  Số request xử lý trong 1 phút: >20000 request  Số request không xử lý được: <3% | Đạt |
| Đăng nhập | Số lượng threads: 100  Thời gian giữa các yêu cầu: 0s  Số vòng lặp:1 | Thời gian phản hồi trung bình: <10s  Thời gian phản hồi lâu nhất: <20s  Số request xử lý trong 1 phút: >20000 request  Số request không xử lý được: <3% | Đạt |
| Xem thông tin project | Số lượng threads: 200  Thời gian giữa các yêu cầu: 0s  Số vòng lặp:1 | Thời gian phản hồi trung bình: <15s  Thời gian phản hồi lâu nhất: <30s  Số request xử lý trong 1 phút: >15000 request  Số request không xử lý được: <3% | Đạt |

## Giải quyết vấn đề xung đột tài nguyên

### Đặt vấn đề

Vấn đề đầu tiên cần phải được nhắc đến đó là việc không phải lúc nào hệ thống cũng đảm bảo được đường truyền internet ổn định để truyền tải dữ liệu từ server cũng như cập nhật dữ liệu lên server. Đặc biệt hệ thống còn được triển khai cả trên thiết bị di động cũng như hệ thống web chính vì vậy không thế tránh được việc xuất hiện các xung đột trong việc cập nhật và đồng bộ dữ liệu gây nên những lỗi không đáng mắc phải trong quá trình hoạt động. Vì vậy ở hệ thống mới em xây dựng sẽ phải giải quyết được hết các vấn đề này đó chính là đảm bảo ứng dụng hoạt động khi không có internet, và đảm bảo xử lý các xung đột gặp phải trong quá trình hoạt động

### Giải pháp đưa ra

Từ những vấn đề gặp phải, sau một thời gian tìm hiểu và nghiên cứu em đã tìm được hai giải pháp hữu hiệu và có thể sử dụng nhất cho hệ thống hiện tại. Giải pháp đầu tiên mà em đưa ra là RMI (Remote Method Invocation). RMI giúp cho việc truyền thông giữa các đối tượng phân tán được dễ dàng hơn. RMI là API bậc cao được xây dựng dựa trên lập trình Socket. Hệ thống RMI cho phép một đối tượng chạy trên một máy ảo Java này có thể kích hoạt một phương thức của một đối tượng đang chạy trên một máy ảo Java khác. Đối tượng có phương thức được gọi từ xa gọi là các đối tượng ở xa (Remote Object). Một ứng dụng RMI thường bao gồm 2 phần phân biệt: Một chương trình Server và một chương trình Client.  Chương trình Server tạo một số các Remote Object, tạo các tham chiếu (reference) đến chúng và chờ những chương trình Client kích hoạt các phương thức của các Remote Object này.  Chương trình Client lấy một tham chiếu đến một hoặc nhiều Remote Object trên Server và kích hoạt các phương thức từ xa thông qua các tham chiếu. Một chương trình Client có thể kích hoạt các phương thức ở xa trên một hay nhiều Server - sự thực thi của chương trình được trải rộng trên nhiều máy tính.

Giải pháp thứ 2 NTP (Network Time Protocol). Server thực hiện việc đồng bộ hóa dữ liệu giữa các Client dựa vào 2 yếu tố là: Kích thước của file dữ liệu - Thời gian thay đổi file dữ liệu. Trong đó, để đồng bộ dựa vào thời gian thay đổi file dữ liệu, trước hết phải đồng bộ đồng hồ của của các máy Client và Server . Sử dụng giao thức NTP. Cơ chế hoạt động của NTP như sau: NTP client gửi một gói tin, trong đó chứa một thẻ thời gian tới cho NTP server. NTP server nhận được gói tin, gửi trả lại NTP client một gói tin khác, có thẻ thời gian là thời điểm nó gửi gói tin đó đi. - NTP client nhận được gói tin đó, tính toán độ trễ, dựa và thẻ thời gian mà nó nhận được cùng với độ trễ đường truyền, NTP client sẽ set lại thời gian của nó.

Từ quy mô của hiện tại và thời gian hoàn thiện của dự án em đã lựa chọn phương pháp NTP để giải quyết bài toán này. Em đã xử lý các việc đồng bộ các dữ liệu hệ thống dựa trên việc kiểm soát, so sánh kích thước các gói tin, cũng như thời gian cập nhật của các của chúng từ đó lựa chọn phần nào được giữ lại phần nào bị loại bỏ, sau đó cập nhật lại thời gian cho các gói tin đó, đảm bảo thông tin và dữ liệu của các gói tin là mới nhất và chính xác nhất.

Tuy nhiên đối với các dữ liệu điểm danh thì bản thân nó là các gói tin về thời gian nên việc sắp xếp và đồng bộ nó đỡ phức tạp hơn là các gói tin khác. Các dữ liệu điểm danh này được lưu dưới dạng string của các thông tin thời gian được ngăn cách nhau bởi dấu phẩy. Khi server gửi một gói tin về dữ liệu điểm danh xuống client thì đầu tiên client sẽ kiểm tra xem dữ liệu ngày hôm nó đã có trong hệ thống chưa, nếu chưa có thì thực hiện insert vào CSDL. Ngược lại nếu dữ liệu điểm danh ngày hôm đó đã tồn tại thì thực hiện việc gép nối 2 gói tin dữ liệu của ngày hôm đó với nhau, sau đó dùng thuật toán loại bỏ các giá trị trùng nhau, tiếp đó tiếp tục việc xắp xếp ại các giá trị thời gian này theo giá trị tăng dần. Sau đó cập nhật lại gói tin tin này vào trong CSDL, rồi trả kết quả cập nhật lại cho server. Như vậy ta luôn đảm bảo được cấc dữ liệu điểm danh là đầy đủ và chính xác nhất.

### Kết quả đạt được

Sau quá trình nghiên cứu và xây dựng và triển khai ứng dụng em đã thành công xây dựng ứng dựng áp dụng đúng quy trình phát triển một phần mềm phân tán. Từ đó giải quyết các vấn đề trong việc hiệu năng sử dụng và đồng bộ dữ liệu trong hoạt động, Thực vậy ứng dụng hoàn toàn có thể hoạt động trong điều kiện không có internet hoặc server bị trục trặc. Tất cả các dữ liệu sẽ được lưu trữ và bộ nhớ của mấy và đợi cập nhật. Trong quá trình đó người dùng sẽ không cảm nhận được do dữ liệu đã được sử lý trực tiếp tại máy.

Các cơ chế đồng bộ và giải quyết xung đột đã giúp cho việc đảm bảo chính xác nhất về thông tin cũng như của người dùng được truyền đến CSDL tập trung mà không tạo quá nhiều gánh nặng cho server. Từ đó cho thấy những mặt vượt trội hơn so với hầu hết các ứng dụng cùng loại có mặt trong thị trường.

# Kết luận và hướng phát triển

Ứng dụng ra đời đã giải quyết được phần nào các vấn đề gặp phải trong việc điểm danh và quản lý công việc. Tuy nhiên do được phát triển và xây dựng trong thời gian ngắn ứng dụng còn gặp phải nhiều vấn đề và hạn chế cần phải nâng cấp phát triển và cập nhật liên tục mới có thể đáp ứng đầy đủ các nhu cầu của người dùng.

## Kết luận

Sau quá trình làm đồ án, em đã xây dựng được thành công ứng dụng “Điểm danh và quản lý công việc trên thiết bị di động” với các chức năng cơ bản như sau: Điểm danh, quản lý dự án, cập nhật thông tin người dùng, đồng bộ dữ liệu hai chiều, xử lý các xung đột dữ liệu khi đồng bộ.

Hệ thống có điểm nổi bật đó là khả năng điểm danh với độ chính xác cao đồng thời có thể tận dụng được tối đa nguồn trang thiết bị đó là các thiết bị di động của người dùng. Gắn liền việc điểm danh với quản lý công việc từ đó khiến cho việc đánh giá năng lực của nhân viên được chính xác nhất.

Hệ thống được xây dựng trên mô hình phân tán và đảm bảo sự trong suốt trong việc đồng bộ dữ liệu, toàn bộ quá trình đồng bộ dữ liệu giữa client – và server được tiến hành trên các service chạy ngầm, ứng dụng hoạt động hoàn toàn bình thường khi không có mạng hoặc kết nối đến server bị lỗi.

Song vẫn còn tồn tại các hạn chế như: hệ thống chỉ có thể áp dụng cho các công ty vừa và nhỏ, dù đã sử dụng framework xong giao diện vẫn chưa thực sự bắt mắt. Khả năng kết hợp với các hệ thống khác hạn chế (chỉ thông qua nhập xuất excel) mất công sức để tạo thành file excel.

Ngoài việc xây dựng thành công được ứng dụng, em còn rút ra được một số kinh nghiệm cho bản thân như việc học tập và áp dụng một công nghệ mới vào ứng dụng. Việc xây dựng ứng dụng, từ đầu đến cuối giúp bản thân em sử dụng được hết tất cả kiến thức đã được học trong trường từ phân tích, thiết kế hệ thống, phát triển xây dựng ứng dụng, kiểm thử, triển khai. Bên cạnh đó, là các kỹ năng mềm như viết báo cáo, làm việc nhóm, tìm kiếm thông tin…

## Hướng phát triển

Khi phát triển ứng dụng do sự hạn chế về thiết bị và thời gian nên một số các nghiệp vụ chưa thực sự bám sát với thực tế. Vì thế, trong tương lai khi thực sự được trải nghiệm trong môi trường làm việc tại các công ty cũng như việc khảo sát thêm các hệ thống khác em sẽ khắc phục, nâng cấp để các nghiệp vụ trong ứng dụng phản ánh đúng thực tế, đem lại sự thân thuộc cho người dùng nhất.

Bên cạnh đó, để khắc phục việc giao tiếp, chuyển đổi giữ liệu giữa các hệ thống quản lý như Trello, Zoho, Jira em sẽ nghiên cứu và tích hợp API của các hệ thống kể trên vào ứng dụng, giúp việc chuyển đổi dữ liệu giữa ứng dụng với các hệ thống khác trở nên dễ dàng, tiện lợi, nhanh chóng hơn.

*Kết luận:*  Ngoài việc hoàn thành việc xây dựng ứng dụng “điểm danh và quản lý dự án”, quá trình làm đồ án tốt nghiệp còn rèn luyện cho em các kỹ năng mềm như viết báo cáo hay tìm kiếm thông tin. Dù ứng dụng sau khi hoàn thành vẫn còn những hạn chế nhất định nhưng em tin nếu đi đúng theo hướng phát triển, dần hoàn thiện các chức năng thì ứng dụng này hoàn toàn có thể được sử dụng rộng rãi tại các công ty.

# Tài liệu tham khảo

1. Trello, Atlassian, https://trello.com/, last visited August 2018.
2. Zoho, Zoho Corporation Pvt. Ltd, https://www.zoho.com/, last visited August 2018.
3. Android, <https://developer.android.com/topic/libraries/support-library/>, last visited August 2018.
4. Open CV, <https://opencv.org/platforms/android/>, last visited August 2018.
5. PGS.TS. Đỗ Xuân Lôi, Cấu trúc dữ liệu và giải thuật, Nhà xuất bản giáo dục, 2009
6. GPS, <https://www.gps.gov/> , last visited August 2018