

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
LỚP CỬ NHÂN TÀI NĂNG**

LÊ THÀNH TÂM

**ỨNG DỤNG QUẢN LÝ GIỜ LÀM VIỆC
CỦA NHÂN VIÊN**

BÀI TEST APPLY CÔNG TY CITYNOW

KHÓA 2012 – 2016

MỤC LỤC

MỤC LỤC	i
Chương 1: Ứng dụng quản lý giờ làm việc nhân viên	1
1.1 Các chức năng hoàn thành.....	1
1.2 Các API (Backend) xây dựng	1
Chương 2: Kiến trúc hệ thống	2
2.1 Backend – Scala và Play framework	2
2.1.1 Cơ sở dữ liệu	2
2.1.2 Kiến trúc	3
2.1.2.1 Kiến trúc DAO	3
2.1.2.2 Controllers	3
2.1.2.3 Views	4
2.2 Frontend – Iphone và Swift	4
2.2.1 Kiến trúc	4
2.2.2 Các thành phần đặc biệt	4
2.2.2.1 Ngôn ngữ ứng dụng	4
2.2.2.2 Slide out Menu	4
2.2.2.3 DialogView	5
2.2.2.4 AutoLayout	5
Chương 3: Giao diện ứng dụng	8
3.1 Đăng nhập.....	8
3.2 Giao diện Camera	9
3.3 Gửi thông tin làm việc	10
3.3.1 Tài khoản hệ thống (Admin).....	11
3.3.1.1 Quản lý tài khoản và cấp quyền cho nhân viên	11
3.3.1.2 Quản lý thời gian làm việc	11

3.3.2	Tài khoản nhân viên.....	13
3.3.3	Menu chức năng.....	14
3.3.4	Màn hình chọn ngôn ngữ.....	15
3.3.5	Màn hình thông tin tác giả	16

Chương 1: Ứng dụng quản lý giờ làm việc nhân viên

1.1 Các chức năng hoàn thành

Hoàn thành hết các chức năng của bài test, trừ chức năng liên kết với google API để lưu dữ liệu thời gian dưới dạng file csv lên google drive. Ngoài ra, một số chức năng sau đây được xây dựng thêm:

- Giao diện menu quản lý ứng dụng.
- Chức năng thay đổi ngôn ngữ ứng dụng: tiếng việt và tiếng anh.
- Chức năng xem danh sách giờ làm việc của các nhân viên trong công ty.

1.2 Các API (Backend) xây dựng

- GET /employees (Lấy danh sách các tài khoản nhân viên)
- POST /employees/:id/delete (Xóa tài khoản)
- GET /getEmployee/:employeeID (Lấy tài khoản nhân viên dựa vào mã nhân viên)
- POST /login (Đăng nhập tài khoản)
- POST /register (Đăng ký tài khoản)
- GET /attendances (Lấy danh sách tất cả giờ làm việc của nhân viên)
- POST /checkIn (Gửi thông tin checkin nhân viên)
- POST /checkOut (Gửi thông tin check out nhân viên)
- GET /getAttendancesByManager/:employeeID (Lấy danh sách giờ làm việc của nhân viên theo theo tài khoản máy)
- GET /getAllRequestPermission (Lấy danh sách tất cả các yêu cầu cấp quyền của nhân viên)
- POST /requestPermission (Gửi yêu cầu cấp quyền của nhân viên – Tài khoản nhân viên)
- POST /updatePermission (Xác nhận yêu cầu cấp quyền nhân viên – Tài khoản Admin quản lý)

Chương 2: Kiến trúc hệ thống

2.1 Backend – Scala và Play framework

Sử dụng IDE là IntelliJ và Postman để test API request.

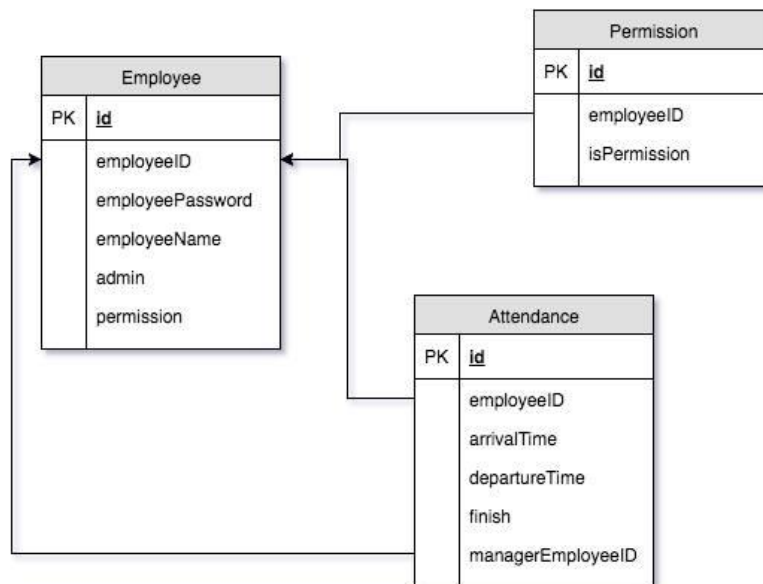
Kiến trúc của Play là MVC gồm các thành phần sau:

- Controllers : Quản lý tất cả các object thuộc về namespace (package) Controller
- Views : Quản lý tất cả các object thuộc về namespace (package) View
- Models : Quản lý tất cả các object thuộc về namespace (package) Model
- Routes : Quản lý các path trong việc định nghĩa URL
- Request (play.api.mvc.Http) quản lý các request từ client lên server (lấy ra các params từ client gửi lên)

2.1.1 Cơ sở dữ liệu

Play kế thừa toàn bộ từ Java vì thế dễ tương tác với DB, Scala Play sử dụng khái niệm DAO (Data access object) để tương tác với DB.

Thư viện sử dụng : JDBC để kết nối và Anorm để truy vấn cơ sở dữ liệu.



Hình 2-1: Kiến trúc cơ sở dữ liệu

Employee: Bảng lưu tài khoản nhân viên và cột Admin dùng để phân loại tài khoản (quản lý và nhân viên). Trong đó, nếu không phải tài khoản Admin thì tài khoản nhân viên có thêm một cột permission để kiểm tra đã cấp quyền hay chưa.

Permission: Bảng lưu trữ các tài khoản nhân viên xin cấp quyền gửi tới các tài khoản Admin. Sau đó, Admin kiểm tra và chấp nhận cấp quyền đối với từng tài khoản.

Attendance: Bảng lưu trữ thời gian làm việc của nhân viên. Trong đó, mỗi tài khoản Admin sẽ quản lý các tài khoản nhân viên riêng của mình.

2.1.2 Kiến trúc

2.1.2.1 Kiến trúc DAO

- Thành phần:
 - 1 Model : Thể hiện các trường trong DB
 - 1 DAO tương ứng : 1 Model sẽ được mapping với một DAO, DAO sẽ chứa tất cả các method access với Database
- Model : các field sẽ tương ứng với các trường trong DB

```
package models  
  
case class Employee(id:Long, employeeID: String, EmployeePassowrd: String...)
```

- DAO : Toàn bộ code thao tác với database :

```
package daos  
  
class EmployeeDAO {  
  
    def all(): TODO  
  
    def getEmployee(id:Long): TODO  
  
    ...  
}
```

- Trong DAO ta sẽ ánh xạ Model với các Database.

2.1.2.2 Controllers

Xây dựng các controller (chức năng) khi View (người dùng) gọi.

Ở ứng dụng này em xây dựng 2 controller: HomeController và AttendanceController.

- HomeController: gồm các controller hỗ trợ cho các request đăng nhập, đăng ký, cấp quyền...
- AttendanceController: Gồm các controller hỗ trợ cho các request checkin, checkout, quản lý thời gian làm việc của nhân viên,...

2.1.2.3 Views

View dùng để xây dựng giao diện người dùng.

2.2 Frontend – Iphone và Swift

Sử dụng IDE Xcode và ngôn ngữ swift.

2.2.1 Kiến trúc

Modul giao diện: Tách ra thành Views con riêng biệt với các ViewControllers để xây dựng giao diện để dễ dàng mở rộng và quản lý.

Phần models: Chứa các model tương ứng để các API request.

Phần API Request: Xây dựng lớp cha ApiRequest để gửi hỗ trợ gửi request lên server và một số phương thức abstract để lớp con kế thừa lại như: loại request (GET, POST,...), nội dung body, url request,...

2.2.2 Các thành phần đặc biệt

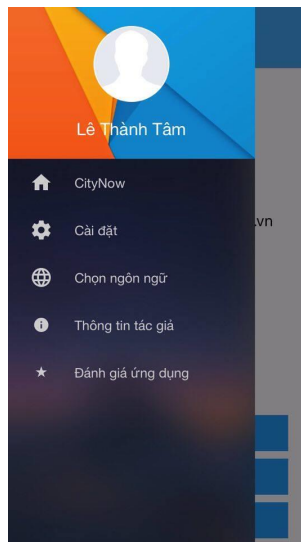
2.2.2.1 Ngôn ngữ ứng dụng

Sử dụng Localizable để xây dựng nhiều ngôn ngữ và sử dụng thư viện Localize_Swift (<https://github.com/marmelroy/Localize-Swift>) để thay đổi ngôn ngữ (Run time) khi mở ứng dụng.

2.2.2.2 Slide out Menu

Xây dựng Menu để quản lý các chức năng của ứng dụng để người dùng tương tác một cách dễ dàng hơn. Sử dụng thư viện SlideMenuController:

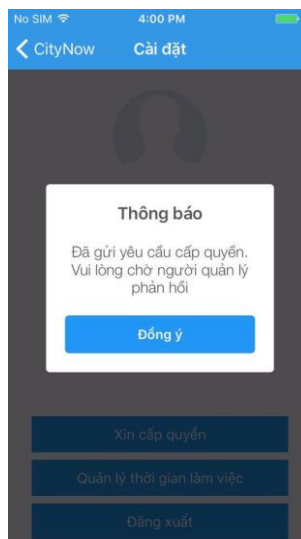
<https://github.com/dekatotoro/SlideMenuControllerSwift>



2.2.2.3 *DialogView*

DialogView dùng để hiển thị thông báo khi tương tác một chức năng nào đó của ứng dụng như: hiện thông báo thành công, lỗi khi gửi request, chọn ngôn ngữ,...

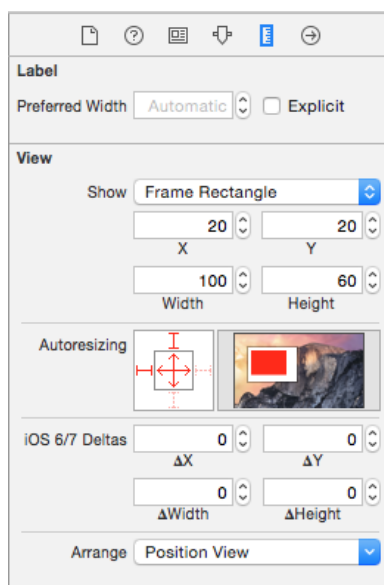
Sử dụng thư viện DialogView: <https://github.com/manGoweb/DialogView>



2.2.2.4 *AutoLayout*

Trước đây khi viết những ứng dụng chạy trên nhiều màn hình hoặc hỗ trợ màn hình xoay ngang, xoay dọc mọi người hay dùng code để có thể chỉnh được những vị trí cũng

như kích thước của những đối tượng. Hoặc có thể dùng Autosizing để tự động canh chỉnh những đối tượng nhưng không tối ưu và tiện lợi cho lắm, hình minh họa ở dưới:

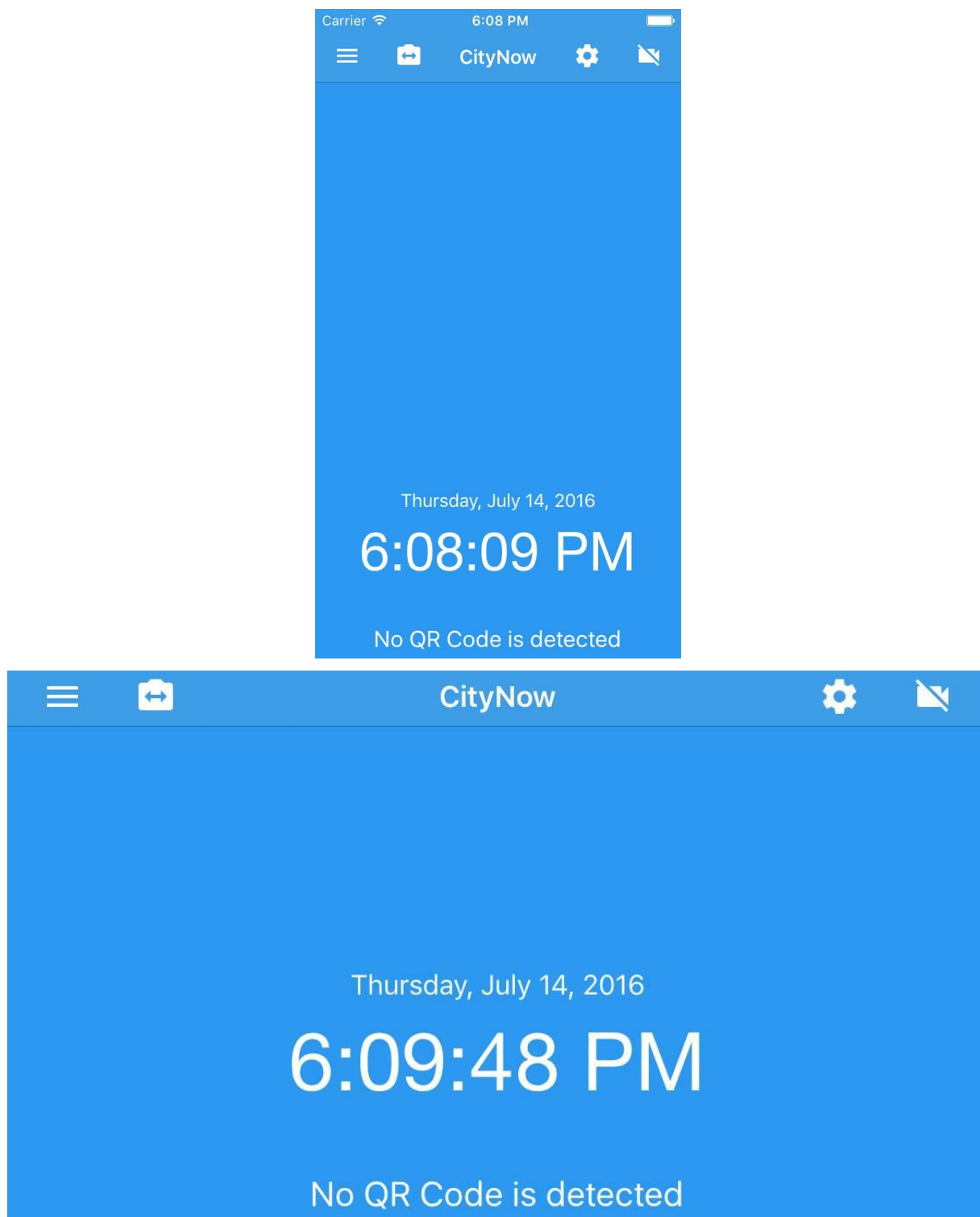


Hình 2-2: Autosizing

Auto layout là hệ thống các ràng buộc (constraint) giữa các thành phần giao diện của chúng ta. Để với một giao diện có thể hiển thị đúng với thiết kế trên các thiết bị có kích thước màn hình khác nhau thì auto layout sẽ hỗ trợ việc này.

Một số đặc trưng sau sẽ dùng auto layout:

- Lập trình đơn giản
- Dễ dàng bảo trì ứng dụng
- Đặc biệt rút ngắn được thời gian viết code khi ứng dụng chạy trên nhiều thiết bị có kích thước màn hình khác nhau



Hình 2-3: Thiết kế giao diện hỗ trợ nhiều thiết bị sử dụng AutoLayout

Chương 3: **Giao diện ứng dụng**

3.1 **Đăng nhập**

Hệ thống sẽ cung cấp tài khoản để đăng nhập được tạo sẵn. Trước khi sử dụng ứng dụng cần đăng nhập để gửi thông tin làm việc của nhân viên. Ngoài tài khoản hệ thống, còn cung cấp thêm tài khoản cho nhân viên đăng nhập. Tuy nhiên, tài khoản nhân viên thì không thể gửi thông tin làm việc của người khác, trừ khi xin cấp quyền.



No SIM 3:50 PM

Đăng nhập Thoát

Mã nhân viên:

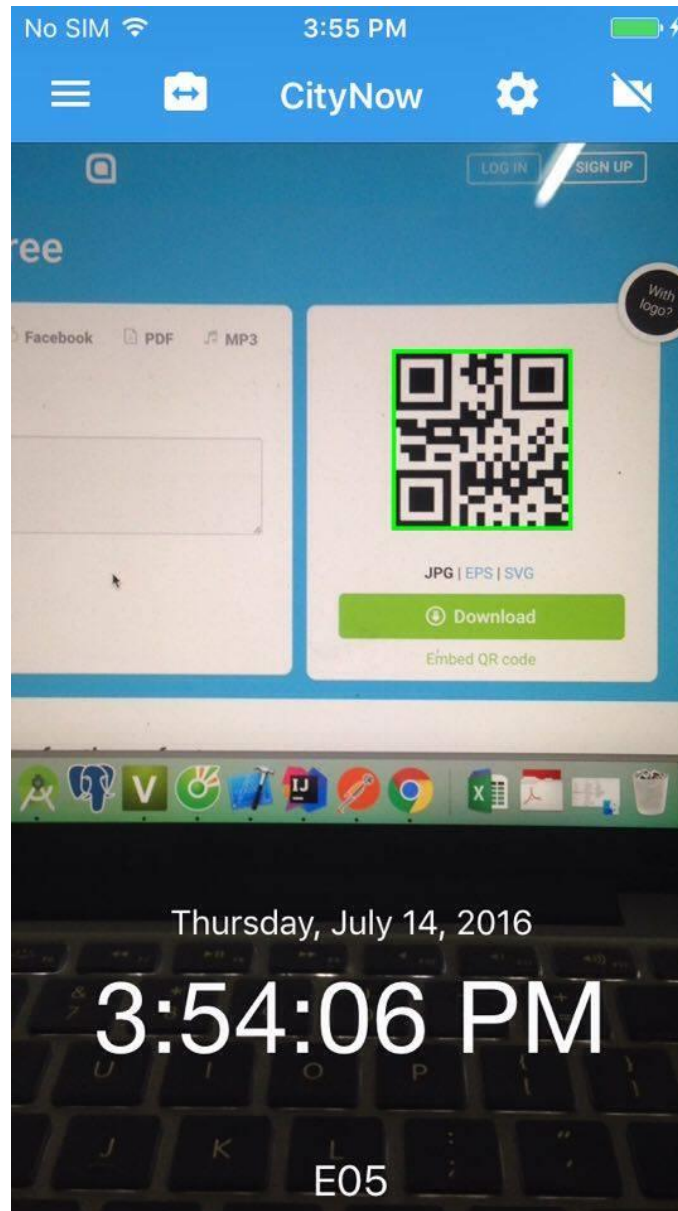
Mật khẩu:

Đăng nhập

Hình 3-1: Màn hình đăng nhập

3.2 Giao diện Camera

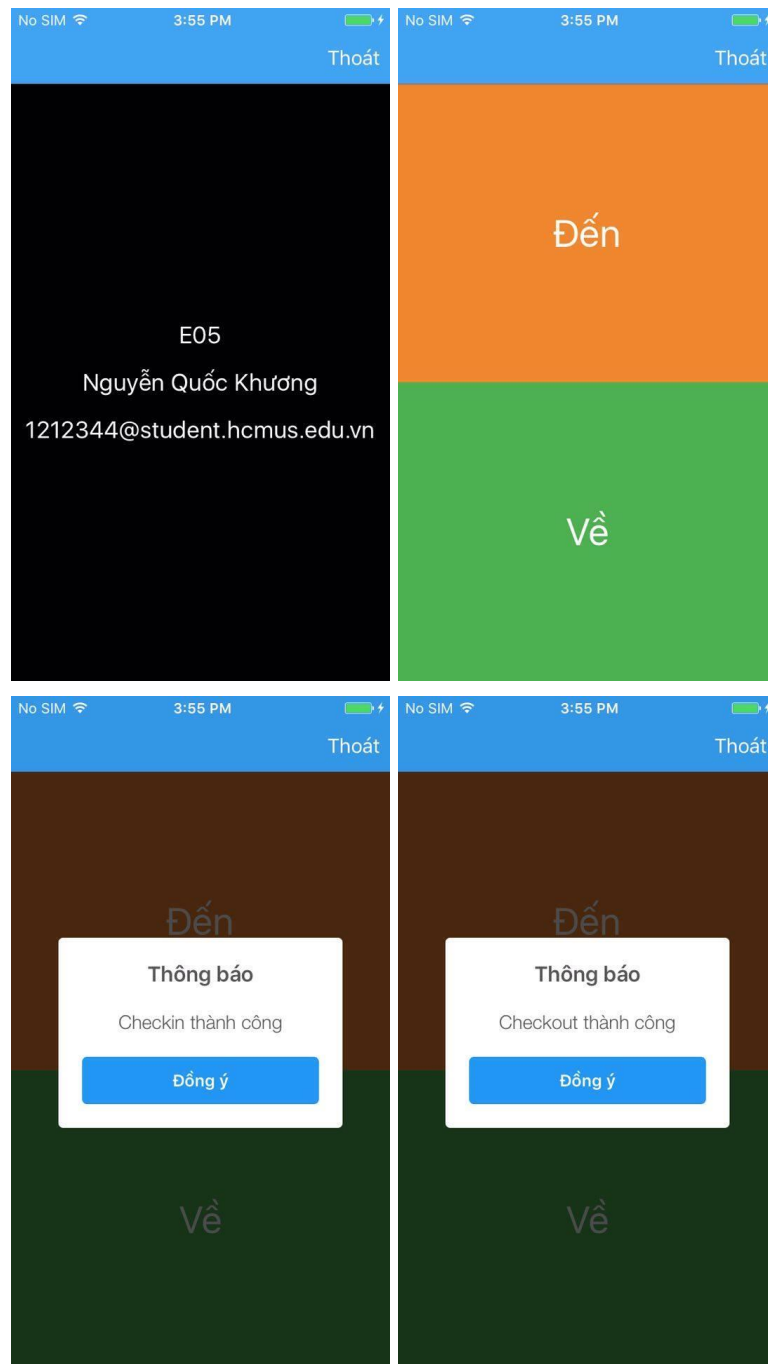
Giao diện camera dùng quét mã thẻ nhân viên, ngoài ra màn hình có thể coi thời gian, ngày tháng. Có thể tắt mở camera, thay đổi camera trước hoặc sau.



Hình 3-2: Màn hình camera

3.3 Gửi thông tin làm việc

Khi người dùng đưa mã thẻ vào thì ứng dụng sẽ hiện thông tin người dùng lên, sau đó người dùng chọn checkin hoặc checkout.



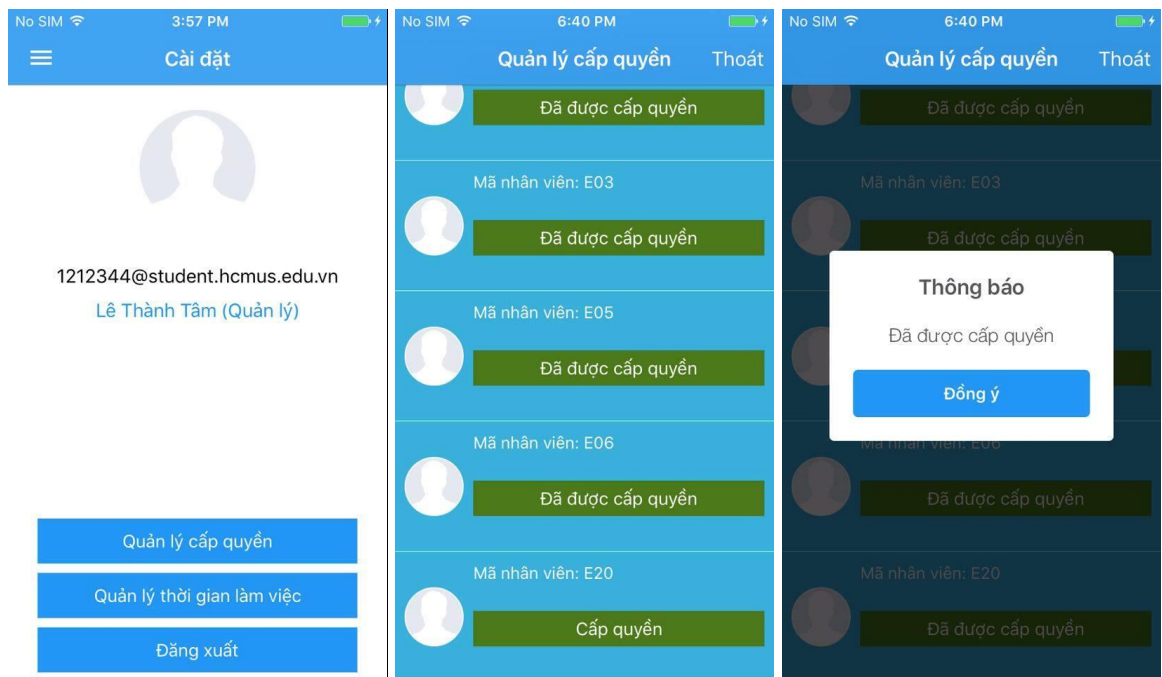
Hình 3-3: Màn hình gửi thông tin làm việc của nhân viên

3.3.1 Tài khoản hệ thống (Admin)

Vào cài đặt, ứng dụng sẽ hiện thị thông tin tài khoản của máy với nhiều chức năng khác nhau tùy vào loại tài khoản

3.3.1.1 Quản lý tài khoản và cấp quyền cho nhân viên

Tài khoản người quản lý (Admin) sẽ có các quyền mà tài khoản thường không có là xem danh sách xin cấp quyền của nhân viên để kiểm tra và đồng ý cấp quyền.



Hình 3-4: Màn hình quản lý của tài khoản hệ thống

3.3.1.2 Quản lý thời gian làm việc





Mỗi tài khoản hệ thống sẽ quản lý riêng các nhân viên của mình, ví dụ trong công ty có nhiều phòng và mỗi phòng có một người quản lý. Do đó, mỗi máy sẽ có một tài khoản riêng để quản lý nhân viên.

No SIM 6:48 PM

Quản lý thời gian làm việc Thoát

Mã người quản lý: E01

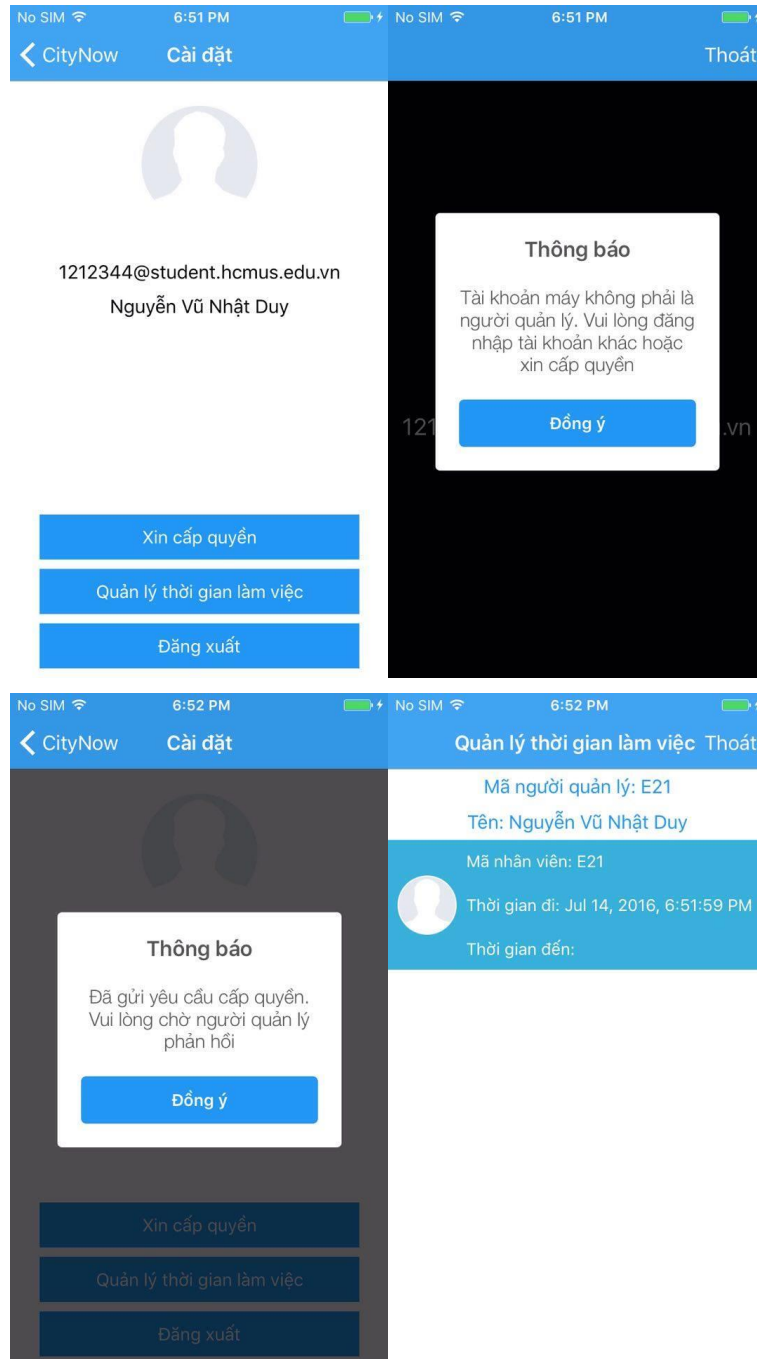
Tên: Lê Thành Tâm

Mã nhân viên: E05		Thời gian đi: Jul 14, 2016, 3:55:41 PM
		Thời gian đến: Jul 14, 2016, 3:55:56 PM
Mã nhân viên: E05		Thời gian đi: Jul 14, 2016, 3:57:38 PM
		Thời gian đến: Jul 14, 2016, 3:57:43 PM
Mã nhân viên: E01		Thời gian đi: Jul 14, 2016, 6:12:16 PM
		Thời gian đến:
Mã nhân viên: E20		Thời gian đi: Jul 14, 2016, 6:48:02 PM
		Thời gian đến:

Hình 3-5: Màn hình quản lý giờ làm việc

3.3.2 Tài khoản nhân viên

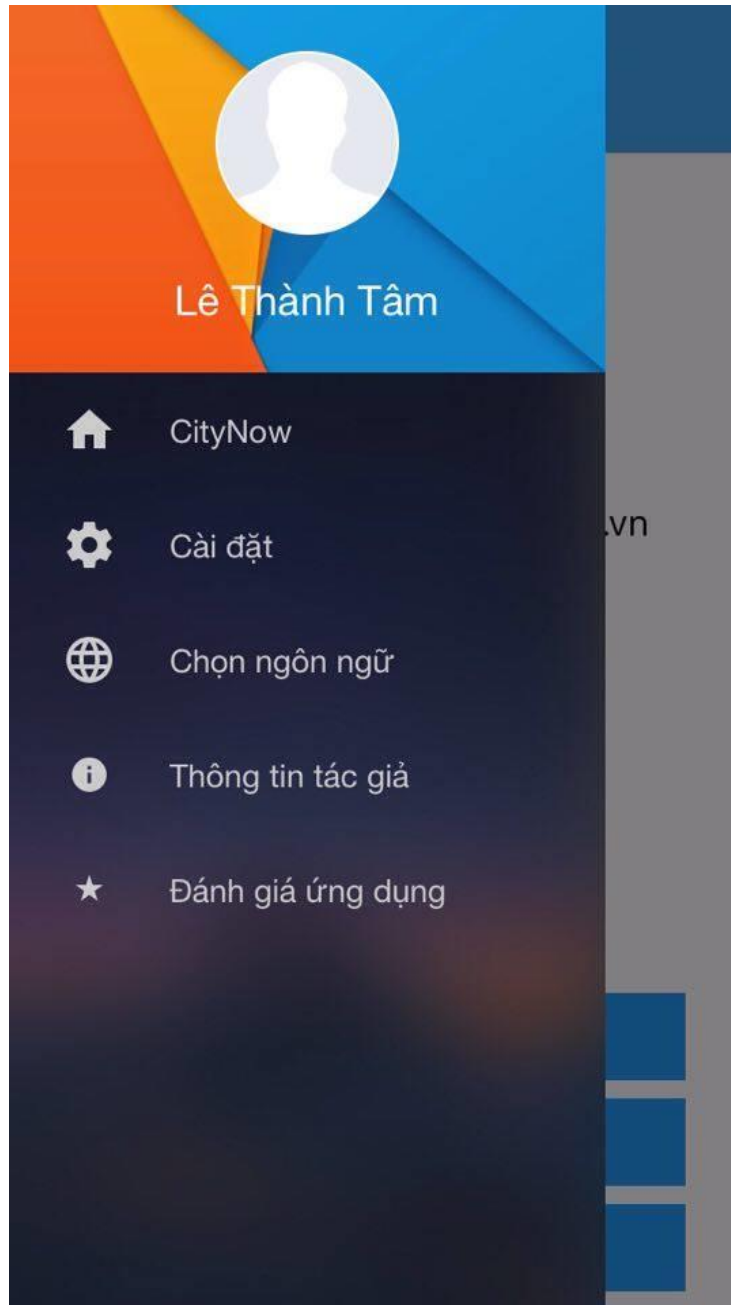
Tài khoản thường chỉ có quyền gửi thông tin check in check out của chính mình, do đó để có thể gửi thông tin của người khác thì cần xin cấp quyền đến Admin.



Hình 3-6: Màn hình quản lý của tài khoản nhân viên

3.3.3 Menu chức năng

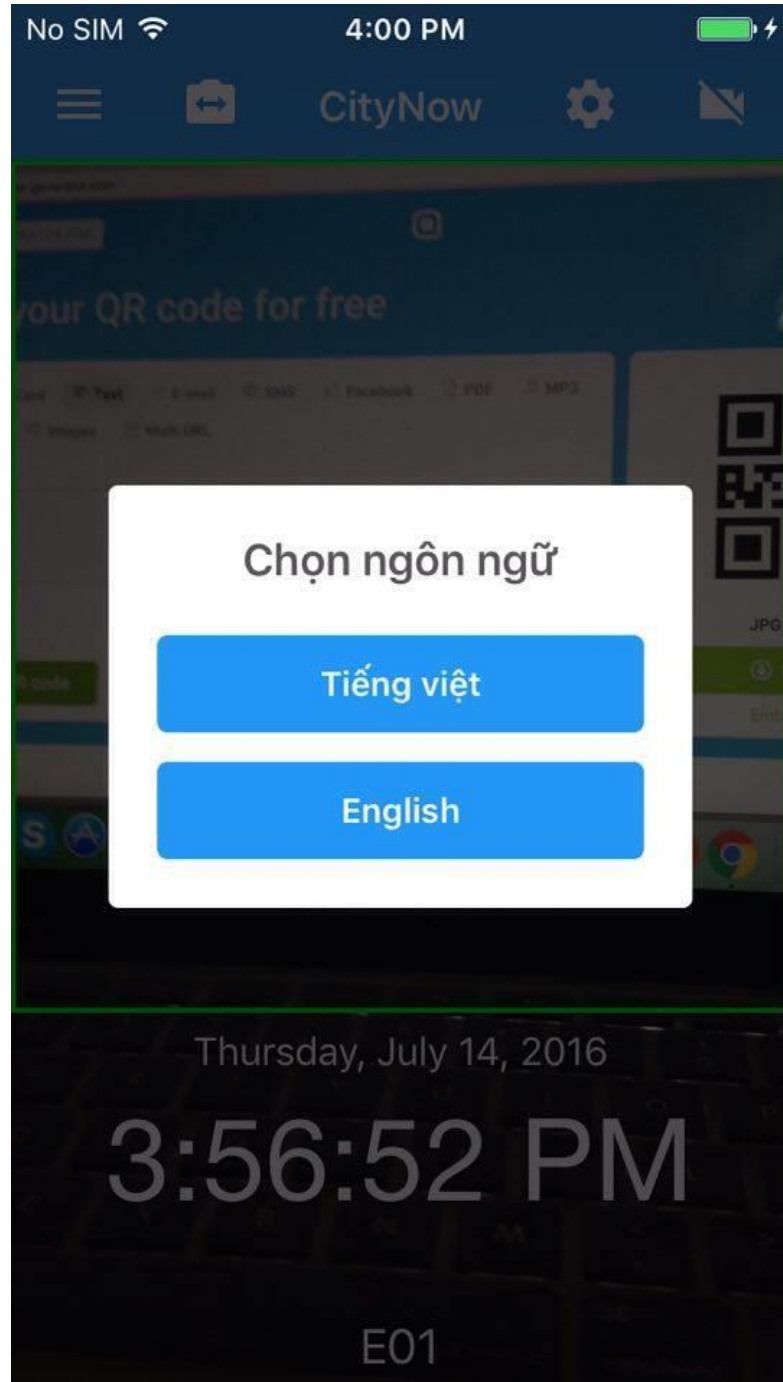
Nhấn vào biểu tượng menu ☰ hoặc lướt từ trái sang phải thì menu chức năng hiển thị gồm: trang camera, trang cài đặt, chọn ngôn ngữ, trang thông tin tác giả và chức năng đánh giá ứng dụng.



Hình 3-7: Màn hình menu chức năng

3.3.4 Màn hình chọn ngôn ngữ

Thay đổi ngôn ngữ: Tiếng việt hoặc tiếng anh. Ngôn ngữ sẽ được thay đổi tại thời điểm chạy run time.



Hình 3-8: Màn hình thay đổi ngôn ngữ

3.3.5 Màn hình thông tin tác giả



Hình 3-9: Màn hình thông tin tác giả