**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TẠI TP.HCM**

⎯🙢🕮🙠⎯



**PHÁT TRIỂN ỨNG DỤNG**

**CHO CÁC THIẾT BỊ DI ĐỘNG**

**ĐỀ TÀI:**

**Ứng dụng “Đặt vé xe khách - CarServices**

**Giảng viên hướng dẫn: Trương Bá Thái**

**Sinh viên thực hiện: Nhóm 5**

1. **Huỳnh Thế An - N14DCCN005**
2. **Ngô Hoàng Quốc Bảo - N14DCCN067**
3. **Nguyễn Tấn Thành - N14DCCN257**
4. **Nguyễn Tấn Bảo Cường - N14DCCN278**
5. **Phan Huy Phương - N14DCCN004**
6. **Trịnh Thông Mãn - N14DCCN021**

**Khoa: Công Nghệ Thông Tin Khóa: 2014**

**Chuyên ngành: Công Nghệ Phần Mềm**

**TP.HCM Tháng 05 năm 2018**

**PHÂN CHIA HOẠT ĐỘNG**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **HỌ TÊN** | **CÔNG VIỆC** | **TỰ ĐÁNH GIÁ** | **NHÓM ĐÁNH GIÁ** | **KÝ TÊN** |
| 1 | Huỳnh Thế An | + Nghiên cứu đề tài.  + Làm màn hình splash screen và code xử lý.  + Làm màn hình giao diện tìm kiếm xe khách và code xử lý.  + Làm màn hình giao diện Đặt vé và code xử lý.  + Viết báo cáo. | 20 | 20 |  |
| 2 | Ngô Hoàng Quốc Bảo | +Nghiên cứu đề tài  +Làm màn hình đăng nhập,thông tin người dùng  +Viết code xử lý  +Viết code giao diện  +Viết báo cáo  +Báo cáo cuối kỳ | 18 | 15 |  |
| 3 | Nguyễn Tấn Thành | +Nghiên cứu đề tài  +Làm màn hình chi tiết xe khách  +Màn hình quản lí vé xe(WinApp)  +Viết code xử lý  +Viết code giao diện  +Viết báo cáo  +Báo cáo cuối kỳ | 18 | 18 |  |
| 4 | Nguyễn Tấn Bảo Cường | +Nghiên cứu đề tài  +Làm màn hình lịch sử và thanh toán  +Viết code xử lý  +Viết code giao diện  +Viết báo cáo  +Báo cáo cuối kỳ | 18 | 20 |  |
| 5 | Phan Huy Phương | +Nghiên cứu đề tài  +Làm đánh giá nhà xe  +Viết code xử lý  +Viết code giao diện  +Viết báo cáo  +Báo cáo cuối kỳ | 15 | 15 |  |
| 6 | Trịnh Thông Mãn | +Nghiên cứu đề tài  +Làm màn hình đăng kí, quên mật khẩu  +Viết code xử lý  +Viết code giao diện  +Viết báo cáo  +Báo cáo cuối kỳ | 18 | 15 |  |

**MỤC LỤC**

[CHƯƠNG 1. MỞ ĐẦU 1](#_Toc513461773)

[1.1. Giới thiệu môn học và nhóm thực hiện Môn Phát triển ứng dụng cho các thiết bị di động này giúp sinh viên phát triển kỹ năng lập trình, tự tiềm kiếm, cập nhật kiến thức mới giúp áp dụng cho các dự án thực tế. 1](#_Toc513461774)

[1.2. Mô tả ứng dụng 1](#_Toc513461775)

[1.3. Ứng dụng trong cuộc sống 2](#_Toc513461776)

[CHƯƠNG 2 PHÂN TÍCH CẤU TRÚC HỆ THỐNG 3](#_Toc513461777)

[2.1 Phân tích thiết hệ thống 3](#_Toc513461778)

[CHƯƠNG 3. CÀI ĐẶT 32](#_Toc513461779)

[CHƯƠNG 4.KẾT QUẢ ĐẠT ĐƯỢC 41](#_Toc513461780)

[4.1 Kết quả đạt được . 41](#_Toc513461781)

[4.2 Kết luận . 41](#_Toc513461782)

# CHƯƠNG 1. MỞ ĐẦU

## 1.1. Giới thiệu môn học và nhóm thực hiện Môn Phát triển ứng dụng cho các thiết bị di động này giúp sinh viên phát triển kỹ năng lập trình, tự tiềm kiếm, cập nhật kiến thức mới giúp áp dụng cho các dự án thực tế.

Kiến thức mới: Hầu như môn này mới học nên tất cả điều là kiến thức mới.

+ Firebase Authentification.

+ Firebase Realtime Database.

+ Fragment

+ AutoComplete EditText.

+ Tìm hiểu về thư viện CircleImageView.

Thành viên thực hiện:

- Huỳnh Thế An

- Ngô Hoàng Quốc Bảo

- Nguyễn Tấn Thành

- Nguyễn Tấn Bảo Cường

- Phan Huy Phương

- Trịnh Thông Mãn

## 1.2. Mô tả ứng dụng

- Tên ứng dụng: CarServices : Là ứng dụng Android giúp người dùng có thể tìm kiếm và đặt vé vé trên điện thoại.

- Cần kết nối mạng - Android 5.0+.

- Đăng ký tài khoản Firebase database bằng email , đăng ký Firebase database bằng gmail.

- Đăng nhập

- Tìm kiếm xe khách.

- Xem thông tin chi tiết xe khách.

- Đặt vé xe.

- Xem lịch sử đặt vé và hủy đặt vé.

- Xem và đánh giá nhà xe qua bình luận.

### 1.3. Ứng dụng trong cuộc sống

Với việc điện thoại thông minh, thiết bị mạng ngày càng phát triển vì vậy các ứng dụng có kết nối mạng được sử dụng liên tục.  
Ứng dụng này giúp mọi người có thể đặt vé xe dễ dàng mọi lúc mọi nơi, không cần đến trực tiếp để đặt và chỉ cần 1 điện thoại android có kết nối mạng.

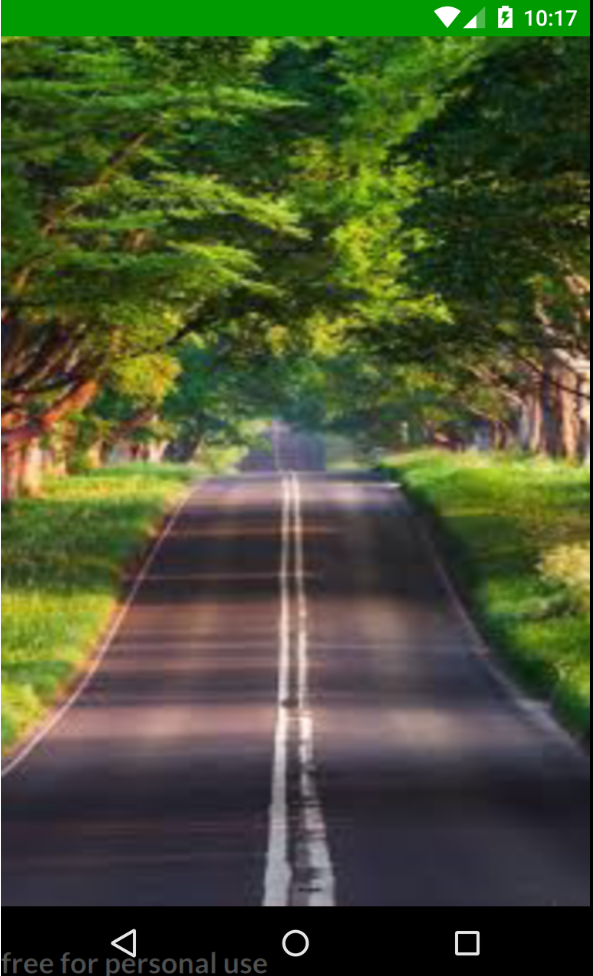
# CHƯƠNG 2 PHÂN TÍCH CẤU TRÚC HỆ THỐNG

## 2.1 Phân tích thiết hệ thống

**Tổng quan:** Đây là ứng dụng dùng để đặt vé xe khách. Ứng dụng cho mọi người đỡ mất thời gian đi đặt vé trực tiếp  
**Phạm vi đề tài:**

⚫ Người dùng phải đăng nhập hoặc phải đăng kí (nếu chưa có tài khoản)  
⚫ Lập trình trên Android Studio 3.1.0  
⚫ Hệ điều hành android: 5.0+  
⚫ Thiết bị thử nghiệm: Genymotion.  
⚫ Độ phân giải màn hình: 768x1280.

**Chức năng #1: Màn hình Splashscreen (Huỳnh Thế An).**



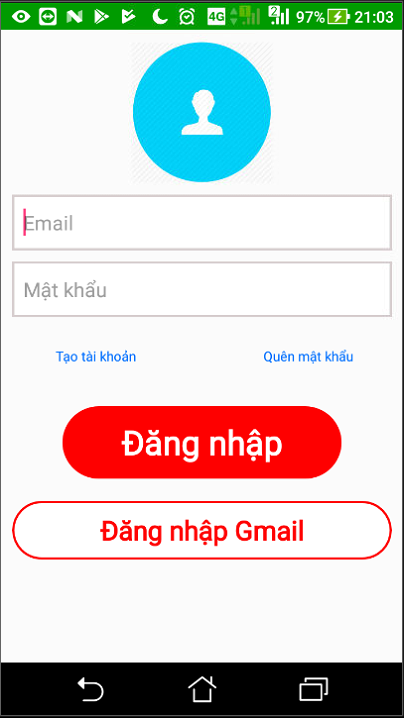
Màn hình Splashscreen.

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình splashscreen. | Màn hình có:  + 1 ConstraintLayout có background là 1 tấm hình | Thiết kế | Hoàn thành |

**Làm thế nào**: Màn hình splashscreen là màn hình xét xem người dùng đã đăng nhập chưa, nếu chưa sẽ chuyển activity đến màn hình đăng nhập. Nếu đã đăng nhập rồi thì sẽ chuyển đến màn hình quản lý của ứng dụng.

**Chức năng #2: Màn hình đăng nhập (Ngô Hoàng Quốc Bảo).**

* Giao diện người dùng:



Màn hình Đăng nhập

* Yêu cầu chức năng:

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình đăng nhập | Màn hình có:   * 1 ImageView * 2 Button * 2 EditText * 2 TextView | Thiết kế | Hoàn thành |
| EditText(Email,Pass) | Nhập Email và Pass đã đăng kí | Nhập dữ liệu | Nhập đúng dữ liệu   * Email có dạng [....@gmail.com](mailto:....@gmail.com) * Pass có 6 chữ số |
| TextView (Tạo tài khoản) | Chuyển sang giao diện Tạo tài khoản | Click | Chuyển sang giao diện Tạo tài khoản |
| TextView (Quên mật khẩu) | Chuyển sang giao diện Quên mật khẩu | Click | Chuyển sang giao diện Quên mật khẩu |
| Button (Đăng nhập) | Kiểm tra, xử lý dữ liệu | Click | Tự động vào Firebase kiểm tra xem đã có và đúng Email, mật khẩu đã đăng kí |
| Button(Đăng nhập Gmail) | Đăng nhập bằng Gmail có sẵn trên điện thoại | Click | Xác nhận xem đã có Gmail đó chưa và Login vào |

Làm thế nào:

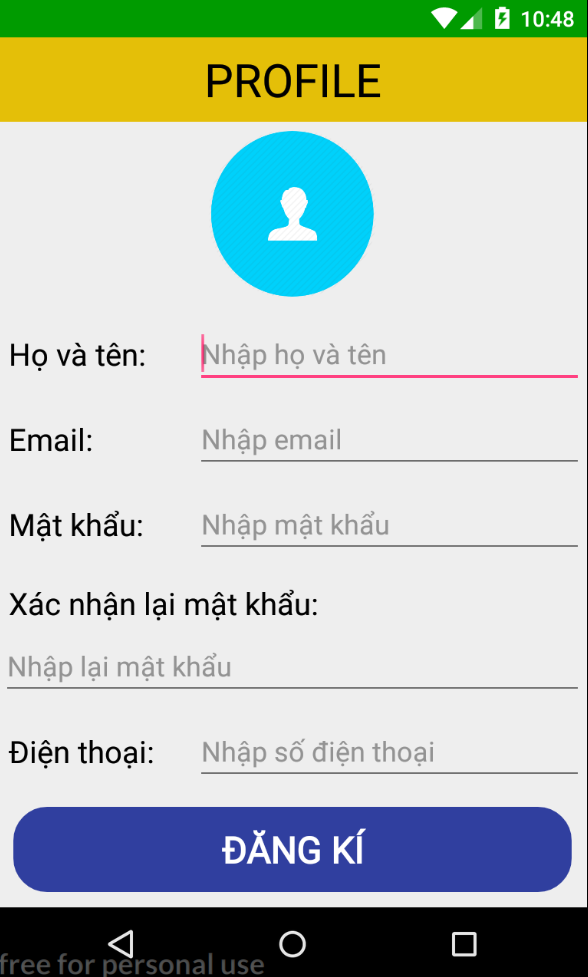
EditText : Nhận dữ liệu được nhập từ người dùng bao gồm : Email và mật khẩu đã đăng kí trước đó hoặc bằng tài khoản Gmail của Google.

Buton “Đăng nhập”: Khi click vào button này, Email và mật khẩu sẽ được vào Firebase kiểm tra xem đã có chưa nếu đăng nhập thành công sẽ chuyển giao diện khác còn ngược lại sẽ có thông báo Email hoặc mật khẩu sai.

Button ‘‘Đăng nhập Gmail’’ :Khi click vào sẽ hiện lên giao diện đăng nhập Gmail của Google, bắt người dùng nhập tài khoản gmail, nếu đăng nhập thành công sẽ chuyển giao diện khác còn ngược lại sẽ có thông báo Email hoặc mật khẩu sai.

**Chức năng #3 : Màn hình đăng kí tài khoản (Trịnh Thông Mãn).**

* Giao diẹn người dùng :



Màn hình nhập thông tin đăng ký tài khoản Firebase.

* Yêu cầu chức nang :

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu Đề | Mô tả | Quá Trình | Kết Quả |
| Màn Hình Tạo Tài Khoản | Màn Hình bao gồm:  + 5 Edit Text để nhập thông tin người dung  + 1 Buton Đăng Kí  +1 Circle Image View(Down từ GitHub source mở về) | Thiết kế | Hoàn Thành |
| Edit Text | Nhập Dữ liệu thông tin người dung | Thiết kế | Lấy được đúng dữ liệu đã nhập |
| Buton Click | Xử lí dữ liệu | Click | Đưa dữ liệu nhận vào gửi lên Auth Firebase để khởi tạo 1 tài khoản mới |

Làm thế nào:

EditText: Nhận dữ liệu được nhập từ người dùng bao gồm các thông tin Họ Tên, Email , Pass, Nhập lại Pass , Số Điện Thoại.

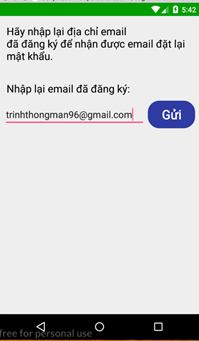
Buton “Đăng Kí”:

+ Khi nhận dữ liệu các thông tin Họ Tên, Email , Pass , Nhập lại Pass , Số Điện Thoại, lúc này trong sự kiện sẽ kiểm tra các giá trị nhập vào có hợp lệ hay chưa. Nếu chưa sẽ báo lỗi, còn hợp lệ sẽ thực hiện chức năng đăng kí tài khoản Auth Firebase đồng thời lưu trữ số điện thoại lên DataBase với khoá UserID tự sinh của Tài Khoản.

+Nếu thành công sẽ thông báo lên màn hình đã đăng kí thành công và ngược lại Thất bại.

**Chức năng #4: Màn hình lấy lại mật khẩu thông qua Email (Trịnh Thông Mãn)**

* Giao diẹn người dùng:



Màn hình đổi lại mật khẩu qua Email

* Yêu cầu chức nang

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu Đề | Mô tả | Quá Trình | Kết Quả |
| Màn Hình Đổi Mật Khẩu thông qua Email | Màn Hình bao gồm:  + 1 Edit Text để nhập Email người dung  + 1 Buton Gửi | Thiết kế | Hoàn Thành |
| Edit Text | +Nhập Email  người dung | Thiết kế | Lấy được đúng dữ liệu đã nhập |
| Buton Click | Xử lí dữ liệu | Click | Đưa dữ liệu nhận vào gửi lên Auth Firebase để gửi về 1 request từ Firebase vào Email người dung. Từ đó người dung có thể thay đổi mật khẩu |

Làm thế nào:

EditText : Nhận dữ liệu được nhập từ người dùng Email.

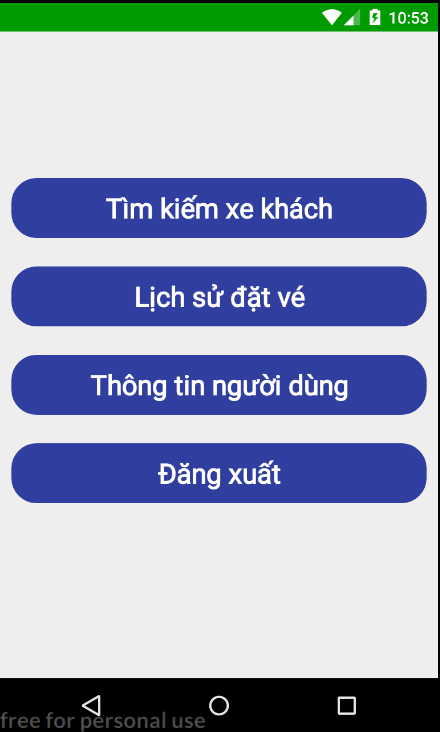
Buton “Gửi”:

+ Khi nhận dữ liệu các thông tin Email lúc này trong sự kiện sẽ kiểm tra các giá trị nhập vào có hợp lệ hay chưa. Nếu chưa sẽ báo lỗi, còn hợp lệ sẽ thực hiện chức năng Auth Firebase Request link gửi trực tiếp vào email người dung để họ thay đổi mật khẩu.

+Nếu thành công sẽ thông báo lên màn hình đã đăng kí thành công và ngược lại Thất bại. Sau đó tự động quay về màn hình đăng nhập.

**Chức năng #5: Màn hình menu chính. (Huỳnh Thế An).**

**Giao diện người dùng:**



**Menu chính**

**Yêu cầu chức năng:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình menu | Màn hình có:  + 4 Button   * Tìm kiếm xe khách * Lịch sử đặt vé * Thông tin người dùng * Đăng suất | Thiết kế | Hoàn thành |
| Button(Tìm kiếm xe khách) | Chuyển sang giao diện tìm kiếm xe khách | Click | Sang giao diện tìm kiếm xe khách |
| Button(Lịch sử đặt vé) | Chuyển sang giao diện lịch sử đặt vé của Email bạn | Click | Sang giao diện lịch sử đặt vé |
| Button(Thông tin người dùng) | Chuyển sang giao diện thông tin người dùng | Click | Sang giao diện thông tin người dùng |
| Button(Đăng suất) | Đăng suất tài khoản hiện tại và chuyển sang giao diện Đăng nhập | Click | Đăng suất và chuyển sang giao diện đăng nhập |

Làm thế nào:

Button “Tìm kiếm xe khách”: Nhấn vào sẽ chuyển đến activity tìm kiếm.

Button “Lịch sử đặt vé: Nhấn vào sẽ chuyển đến activity xem lịch sử các vé đã đặt.

Button “Thông tin người dùng”: nhấn vào sẽ xem được thông tin bạn đã đăng ký.

Button “Đăng xuất”: Đăng xuất.

**Chức năng #6: Màn hình Tìm kiếm xe khách. (Huỳnh Thế An).**

Giao diện người dùng:



Màn hình Tìm kiếm xe khách.

Yêu cầu chức năng:

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình menu | Màn hình có:  + 2 AutoCompleteTextview:  + 1 Textview:  + 2 Button  + 1 Listview | Thiết kế | Hoàn thành |
| AutoCompleteTextview (Điểm đi). | Cho người dùng nhập vào điểm đi. | Nhập chuỗi. | Hoàn thành |
| AutoCompleteTextview (Điểm đến). | Cho người dùng nhập vào điểm đến. | Nhập chuỗi. | Hoàn thành |
| Button ( Chọn ngày) | Nhấp vào để chọn ngày khởi hành từ DatePicker. | Click | Hoàn thành |
| TextView (Hiện ngày đã chọn) | Hiển thị ngày khởi hành mà bạn đã chọn | Hiển thị | Hoàn thành |
| Button ( Tìm kiếm) | Nhấp vào để tiến hành tìm kiếm xe khách. | Click | Hoàn thành |
| ListView | Hiển thị danh sách các chuyến xe khách vừa tìm được. |  |  |

Làm thế nào:

AutoCompleteTextView: Khi nhập vài ký tự vào thì nó sẽ có chức năng gợi ý một số tỉnh thành, bạn có thể sẽ thấy nơi bạn xuất phát.

AutoCompleteTextView: Khi nhập vài ký tự vào thì nó sẽ có chức năng gợi ý một số tỉnh thành, bạn có thể sẽ thấy nơi bạn đến.

Button ( chọn ngày ) : khi nhấn vào sẽ hiển thị một DatePicker cho bạn chọn ngày khởi hành.

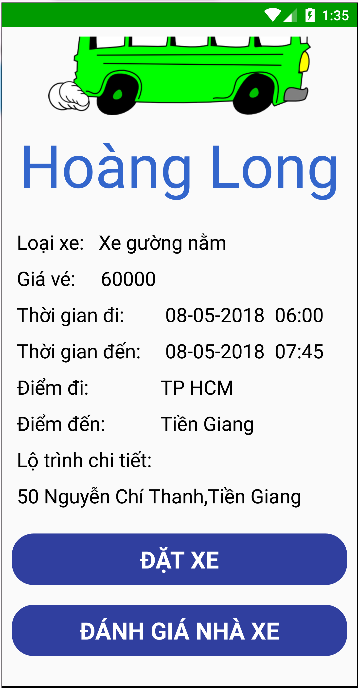
TextView : Ngày khởi hành mà bạn chọn sẽ được hiển thị ở đây.

Button “Tìm kiếm”: Khi nhấp vào sẽ lấy thông tin bạn đã nhập để tìm kiếm các xe khách bạn cần.

ListView: Hiển thị kết quả tìm kiếm.

**Chức năng #8: Màn hình hiển thị chi tiết xe khách. (Nguyễn Tấn Thành).**

Giao diện người dùng:

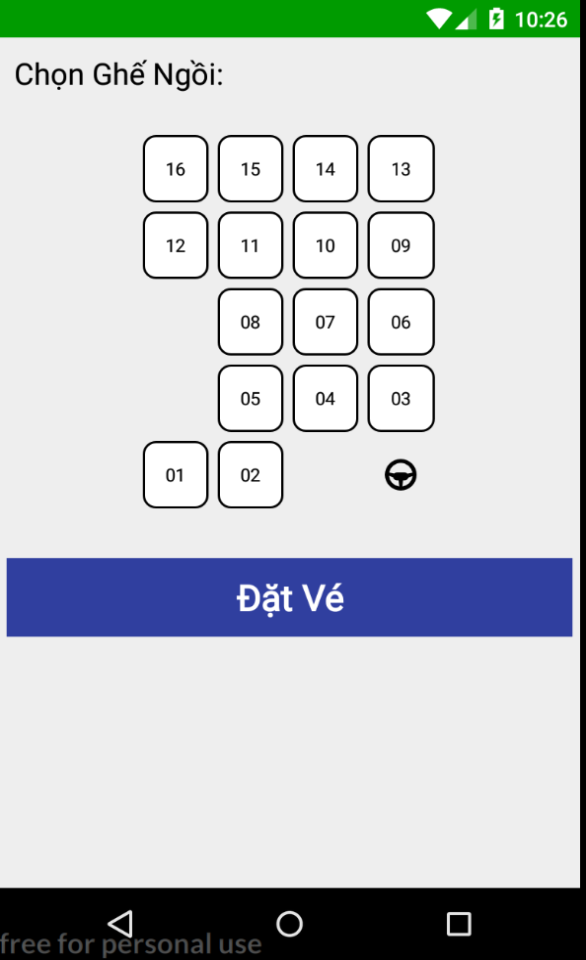
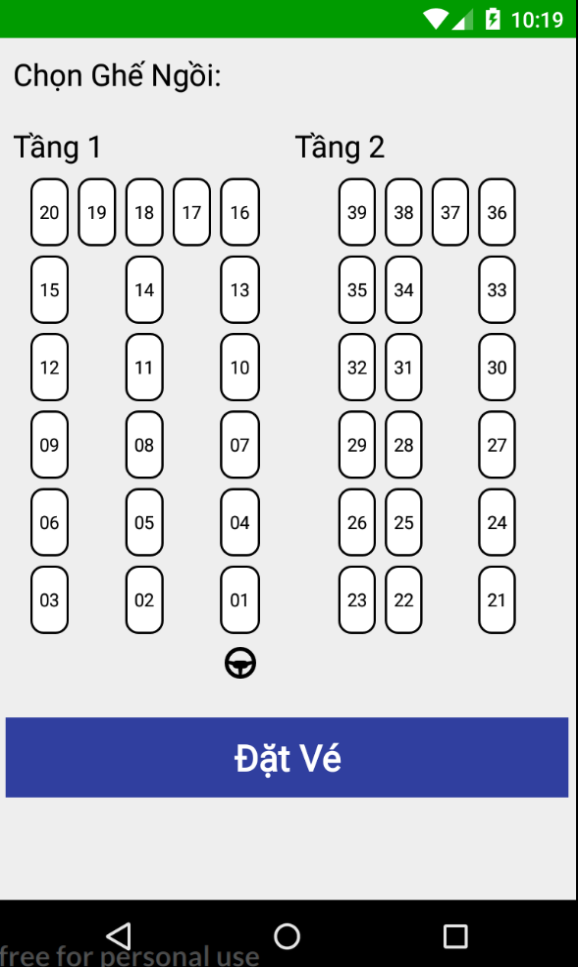


Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình xem chi tiết xe | Màn hình có:  -Thông tin xe: Loại xe, giá vé, thời gian đi, thời gian đến, điểm đi, điểm đến, lộ trình chi tiết.  -2 Button: Đặt xe, Đánh giá nhà xe | Thiết kế | Hoàn thành |
| Button(Đặt xe) | Chuyển qua giao diện đặt chỗ | Click | Hoàn thành |
| Button(Đánh giá nhà xe) | Chuyển qua giao diện đánh giá nhà xe | Click | Hoàn thành |

**Chức năng #9: Màn hình Đặt vé (Huỳnh Thế An).**

Giao diện người dùng :



Yêu cầu chức năng :

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình menu | Màn hình có:  + Button (Đặt vé)  + FrameLayout | Thiết kế | Hoàn thành |
| Button (Đặt vé) | Nhấp vào để đăng ký các vé đã chọn | Click | Hoàn thành |
| FrameLayout | Hiển thị sơ đồ ghế ngồi theo từng loại xe. | Cick | Hoàn thành |

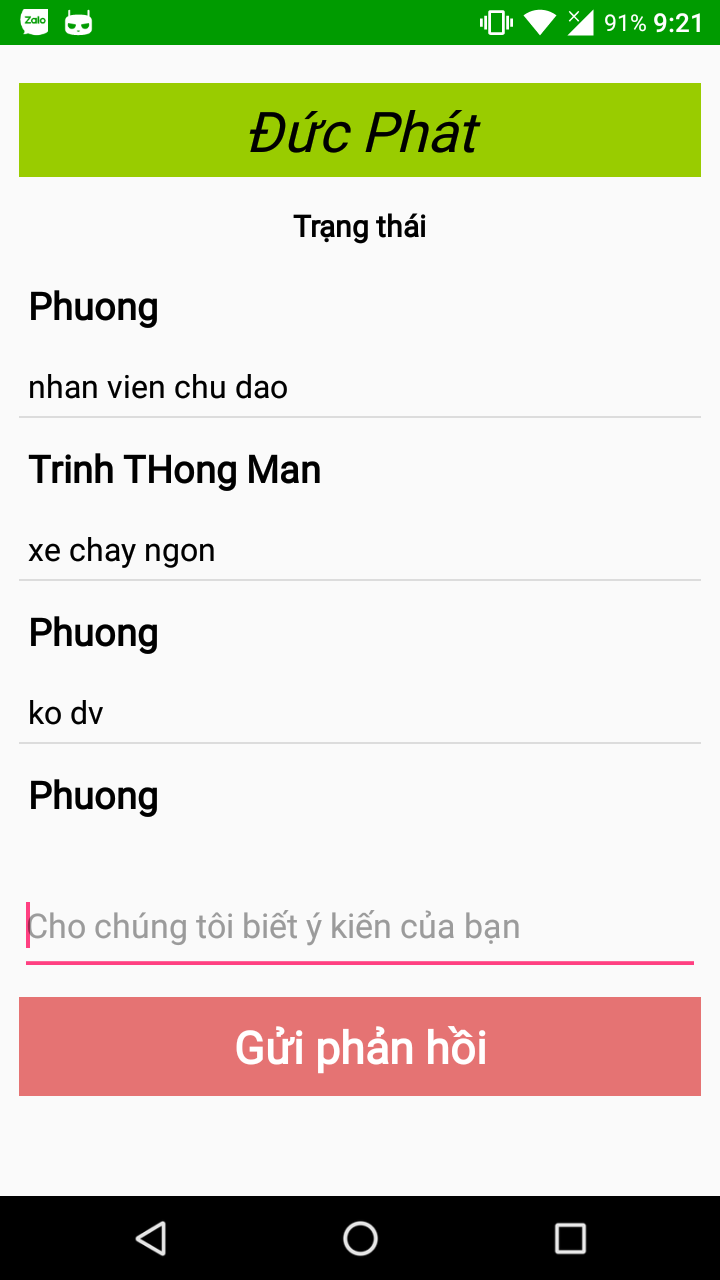
Làm thế nào :

Button : Khi nhấp vào sẽ đăng ký vé xe.

Framelayout : để thêm Fragment xử lý khi chọn đặt chỗ.

**Chức năng #10: Màn hình Đánh giá nhà xe (Phan Huy Phương).**

Giao diện người dùng :



Yêu cầu chức năng :

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu Đề | Mô tả | Quá Trình | Kết Quả |
| Màn Hình Đánh Giá | Màn Hình bao gồm:  + 1 Edit Text  + 1 Buton  +1 listview  + 2 textview | Thiết kế | Hoàn Thành |
| Edit Text | Nhập Dữ liệu thông tin phản hồi | Thiết kế | Lấy được đúng dữ liệu đã nhập |
| Buton Click | Xử lí dữ liệu | Click | Đưa dữ liệu nhận vào gửi lên Firebase |
| Listview | Hiển thị phản hồi |  | Nhận và hiển thị dữ liệu từ firebase |

Làm thế nào :

EditText : Nhận dữ liệu phản hồi được nhập từ người dùng.

Button “Gửi phản hồi”:

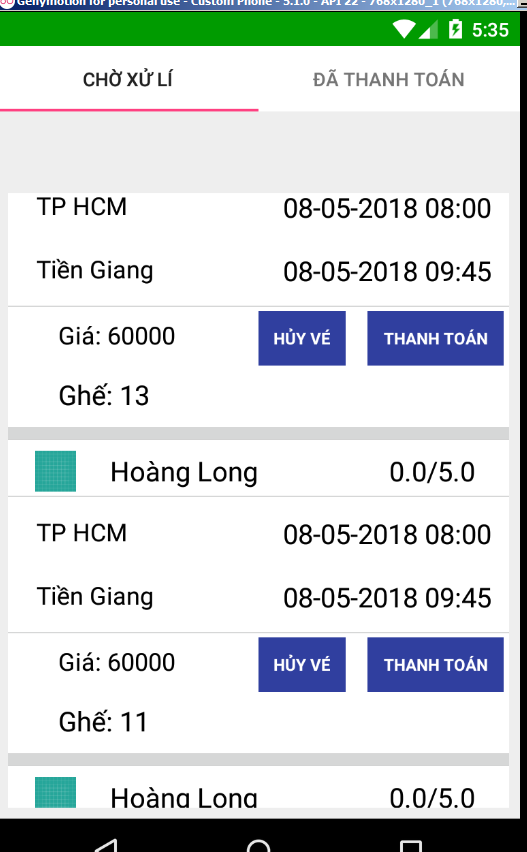
+Click button gửi phản hồi.

+Lưu dữ liệu vào database.

Listview: Hiển thị thông tin phản hồi của người dùng.

**Chức năng #11: Màn hình Lịch sử đặt vé (Nguyễn Tấn Bảo Cường).**

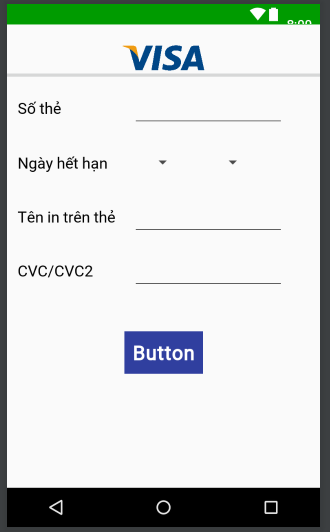
Giao diện người dùng :



Yêu cầu chức năng :

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình xem lịch sử | Màn hình có:   * 1 Tab Page * 1 ViewPager * Thiết kế layout\_tabpage\_lich\_su\_dat\_ve để add vào ViewPager * Layout\_tabpage\_lich\_su\_dat\_ve gồm:   + 2 button | Thiết kế | Hoàn thành |
| Button(Thanh toán) | Chuyển qua giao diện thanh toán | Click | Hoàn thành |
| Button(Hủy vé) | Hủy vé | Thay đổi status của vé  Xóa nút vé trong danh sách ticketList của user trên firebase | Hoàn thành |

**Màn hình Thanh toán**

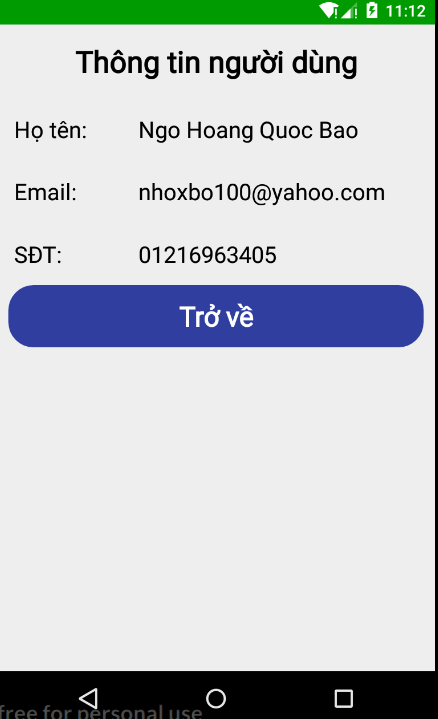


Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình thanh toán | Màn hình có:   * 3 EditText * 2 Spinner * 1 Button | Thiết kế | Hoàn thành |
| EditText(Số thẻ) | Số thẻ trên Visa | Thiết kế | Lấy được đúng dữ liệu đã nhập, bắt lỗi theo điều kiện thẻ visa, mã số không dưới 16 kí tự |
| EditText(Tên trên thẻ) | Tên trên thẻ Visa | Thiết kế | Lấy được đúng dữ liệu đã nhập, bắt lỗi theo điều kiện thẻ visa, tên không để trống |
| EditText(CVC/CVC2) | Mã bảo mật trên thẻ Visa | Thiết kế | Lấy được đúng dữ liệu đã nhập, bắt lỗi theo điều kiện thẻ visa, mã bảo mật phải 3 kí tự |
| Spinner(Tháng)  Spinner(Năm) | Chọn tháng và năm trên thẻ Visa | Thiết kế | Hoàn thành |
| Button(Thanh toán) | Xác nhận thanh toán và trở về giao diện xem lịch sử | Click | Xác nhận thanh toán và chuyển view về xem lịch sử |

**Chức năng #12: Màn hình Thông tin người dùng (Ngô Hoàng Quốc Bảo).**

Giao diện người dùng :



Màn hình xem thông tin User

Yêu cầu chức năng :

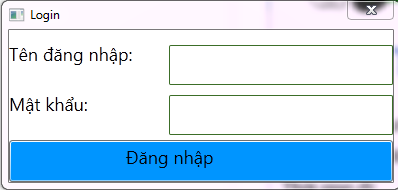
|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình UserInfo | Màn hình có:   * 3 TextView | Thiết kế | Hoàn thành |
| TextView(Họ tên) | Tại đây nó lấy Họ tên từ Email, mật khẩu đã đăng nhập từ đầu trên Firebase | Tự động hiện | Hiện kết quả đã lấy được từ Firebase |
| TextView (Email) | Tại đây nó lấy Email từ Email, mật khẩu đã đăng nhập từ đầu trên Firebase | Tự động hiện | Hiện kết quả đã lấy được từ Firebase |
| TextView (SĐT) | Tại đây nó lấy SDT từ Email, mật khẩu đã đăng nhập từ đầu trên Firebase | Tự động hiện | Hiện kết quả đã lấy được từ Firebase |

Làm thế nào :

- Các TextView hiện các thông tin được lấy từ Email và Pass đã đăng nhập ban đầu như: Họ tên, Email, SĐT.

**Quản lý vé xe và quản lý chuyến xe (Win app) (Nguyễn Tấn Thành)**

Màn hình đăng nhập



Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình thanh toán | Màn hình có:   * 2 EditText * 2 TextBlock * 1 Button | Thiết kế | Hoàn thành |
| EditText(Tên đăng nhập) | Email của người dùng | Nhập từ bàn phím | Email của người dùng |
| EditText(Mật khẩu) | Mật khẩu của người dùng | Nhập từ bàn phím | Mật khẩu của người dùng |
| Button (Đăng nhập) |  | Click | Chuyển đến màn hình quản lý |

Cách hoạt động :

-Button(Thanh toán) : Khi click sẽ gọi hàm LoginBtn(), lấy danh sách user từ Firebase sao đó so sánh từng user email và pass với nội dung nhập từ 2 TextBox nếu trùng và quyền user là admin thì đăng nhập thành công.

private void LoginBtn(object sender, RoutedEventArgs e)

{

bool isHave = false;

userList = getUserList();

foreach (var u in userList)

{

if (u.email == email.Text && u.pass == pass.Password && u.rule == "admin")

{

isHave = true;

break;

}

}

if(isHave == true)

{

MainWindow main = new MainWindow();

main.Show();

this.Close();

}

else

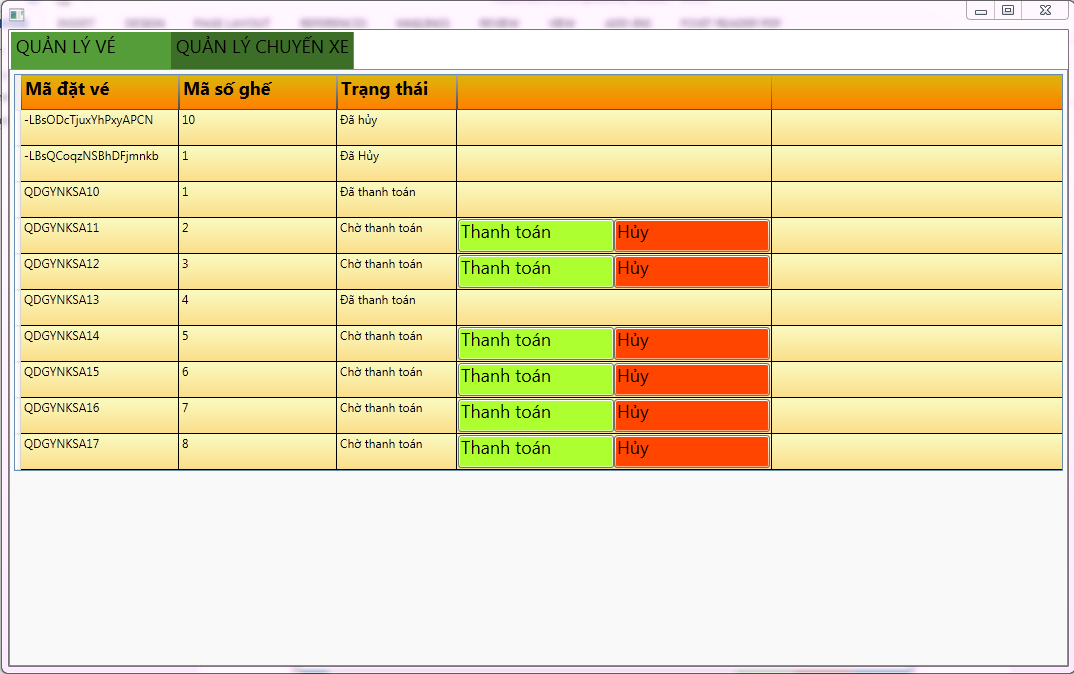
{

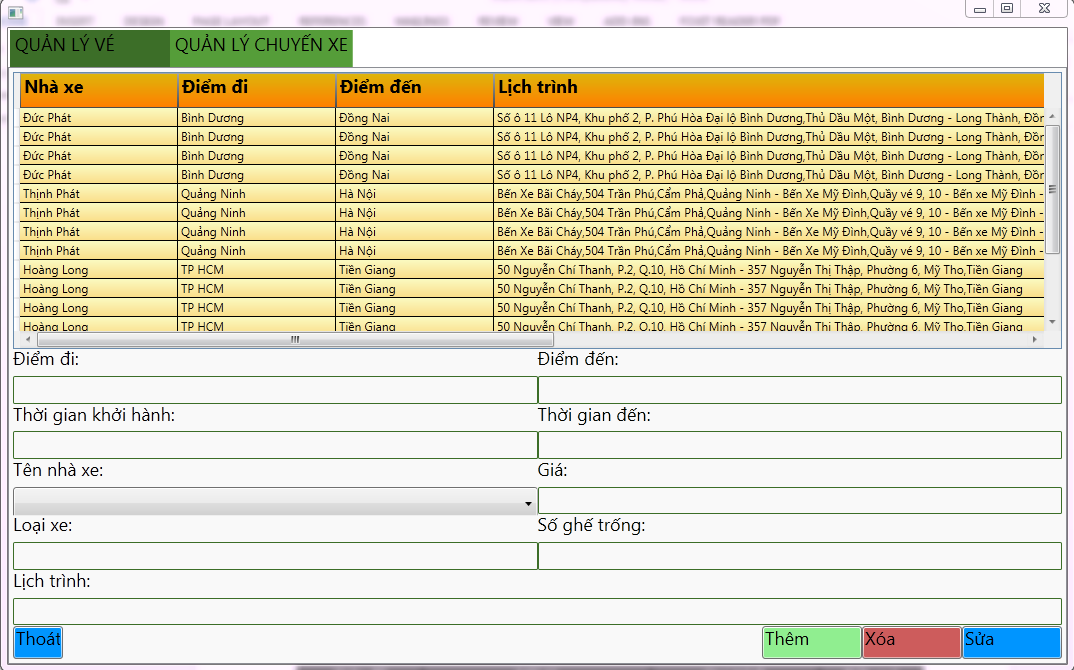
MessageBox.Show("Tài khoản không tồn tại");

}

}

Màn hình quản lý





Yêu cầu chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu đề | Mô tả | Qui trình | Kết quả |
| Màn hình thanh toán | Màn hình có:  -2 TabItem, 2 DataGrid | Thiết kế | Hoàn thành |
| Tab Quản lý vé | - Hiển thị Thông tin vế gồm: Mã đặt vé, trạng thái vé,…  - Mỗi hàng dữ liệu có 2 Button (thanh toán và hủy vé) để thay đổi trang thái vé | Thiết kế | Hoàn thành |
| Tab Quản lý xe khách | -Hiển thị thông tin xe khách và các Button (thêm, xóa, sửa, lưu, hủy) để thao tác với danh sách xe khách  -Button (thoát) để trở về màn hình đăng nhập | Thiết kế | Hoàn thành |

Cách hoạt động :

- Sau khi đăng nhập Giao diện quản lý sẽ được khởi tạo và gọi các hàm : getTicketList, getCoachList, companyList:

InitializeComponent();

ticketList = getTicketList();

dg1.ItemsSource = null;

dg1.ItemsSource = ticketList;

coachList = getCoachList();

dg2.ItemsSource = null;

dg2.ItemsSource = coachList;

companyList = getCompanyList();

cbb.ItemsSource = companyList;

các hàm này sẽ lấy dữ liệu từ firebase và show ra view bằng datagrid.

- Khi ấn nút thanh toán hàm PayButton sẽ được gọi, hàm này thay đổi thuộc tính status của vé đang chọn thành « Đã thanh toán » và cập nhật dữ liệu vào firebase.

private void PayButton(object sender, RoutedEventArgs e)

{

Button b = (Button)sender;

string id = ((Ticket)b.DataContext).id;

foreach(var t in ticketList)

{

if (t.id == id)

{

t.status = TicketStatusPayed;

FBUpdate("https://carservices-cb0bd.firebaseio.com/TicketList/" + t.id + ".json", JsonConvert.SerializeObject(t));

dg1.ItemsSource = null;

dg1.ItemsSource = ticketList;

break;

}

}

}

- Khi ấn nút thanh toán hàm CancelButton sẽ được gọi, hàm này thay đổi thuộc tính status của vé đang chọn thành « Đã hủy » và cập nhật dữ liệu vào firebase.

private void CancelButton(object sender, RoutedEventArgs e)

{

Button b = (Button)sender;

string id = ((Ticket)b.DataContext).id;

foreach (var t in ticketList)

{

if (t.id == id)

{

t.status = TicketStatusCancel;

dg1.ItemsSource = null;

dg1.ItemsSource = ticketList;

FBUpdate("https://carservices-cb0bd.firebaseio.com/TicketList/" + t.id + ".json", JsonConvert.SerializeObject(t));

}

}

}

- Khi ấn nút thêm hàm AddCoach được gọi, hàm này thêm 1 đối tượng Coach vào CoachList cho phép thay đổi thông tin đối tượng này và cập nhập lại view

private void AddCoach(object sender, RoutedEventArgs e)

{

isAdd = true;

canvas.Visibility = Visibility.Collapsed;

edit.Visibility = Visibility.Visible;

noEdit.Visibility = Visibility.Collapsed;

Coach c = new Coach();

coachList.Add(c);

dg2.SelectedItem = c;

currentCoach = c;

}

- Khi ấn nút xóa hàm DelCoach được gọi, hàm này xóa đối tượng Coach đang được chọn khỏi CoachList và cập nhập lại view

private void DelCoach(object sender, RoutedEventArgs e)

{

if (dg2.SelectedItem != null)

{

coachList.Remove(currentCoach);

}

}

- Khi ấn nút sửa hàm EditCoach được gọi, hàm này cho phép thay đổi thông tin đối tượng Coach đang được chọn.

private void EditCoach(object sender, RoutedEventArgs e)

{

if (dg2.SelectedItem != null)

{

isEdit = true;

beforeCoach = new Coach();

beforeCoach.copyCoach(currentCoach);

canvas.Visibility = Visibility.Collapsed;

edit.Visibility = Visibility.Visible;

noEdit.Visibility = Visibility.Collapsed;

}

}

- Khi ấn nút lưu hàm SaveCoach được gọi, hàm này xét xem nếu thay đổi là thêm thì sẽ thêm 1 đối tượng mới lên firebase còn nếu thay đổi là sửa thì sẽ cập nhật lại đối tượng này trên firebase

private void SaveCoach(object sender, RoutedEventArgs e)

{

string s = JsonConvert.SerializeObject(currentCoach);

if (isEdit == true)

{

isEdit = false;

FBDel("https://carservices-cb0bd.firebaseio.com/CoachList/" + currentCoach.id + ".json");

currentCoach.id = currentCoach.from + "-" + currentCoach.to + "-" + currentCoach.timeStart + "-" + currentCoach.companyId;

FBSet("https://carservices-cb0bd.firebaseio.com/CoachList/" + currentCoach.id + ".json", s);

}

else if (isAdd == true)

{

isAdd = false;

currentCoach.id = currentCoach.from + "-" + currentCoach.to + "-" + currentCoach.timeStart + "-" + currentCoach.companyId;

FBSet("https://carservices-cb0bd.firebaseio.com/CoachList/" + currentCoach.id + ".json", s);

}

canvas.Visibility = Visibility.Visible;

edit.Visibility = Visibility.Collapsed;

noEdit.Visibility = Visibility.Visible;

}

- Khi ấn nút hủy hàm Cancel được gọi, hàm này ngưng không cho user thay đổi thông tin trên view và trả thông tin về như lúc chưa ấn nút thêm hoặc sửa

private void Cancel(object sender, RoutedEventArgs e)

{

if (isAdd == true)

{

coachList.Remove(currentCoach);

isAdd = false;

}

else if (isEdit == true)

{

for (int i=0; i<coachList.Count; i++)

{

if (coachList[i].id == currentCoach.id)

{

coachList[i].copyCoach(beforeCoach);

dg2.ItemsSource = null;

dg2.ItemsSource = coachList;

}

}

isEdit = false;

}

canvas.Visibility = Visibility.Visible;

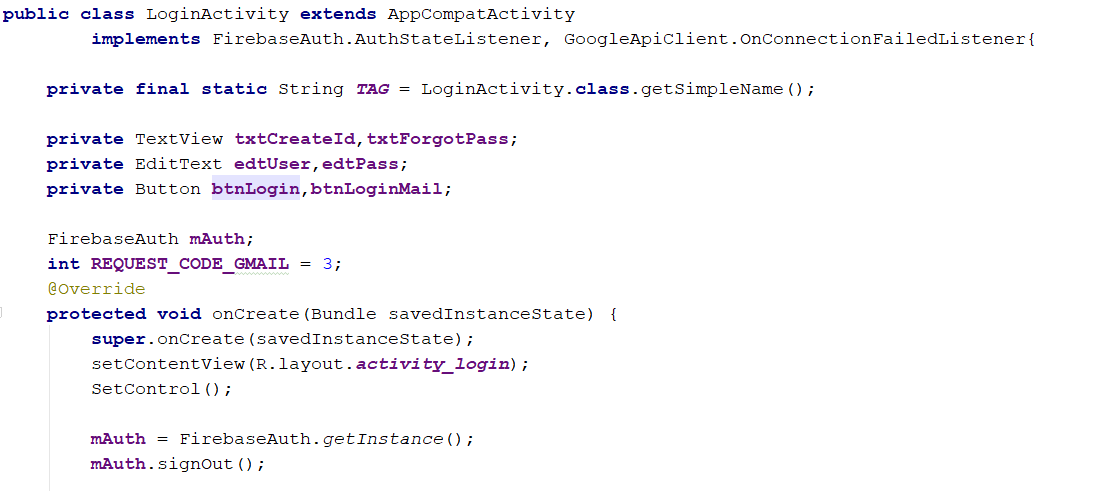
edit.Visibility = Visibility.Collapsed;

noEdit.Visibility = Visibility.Visible;

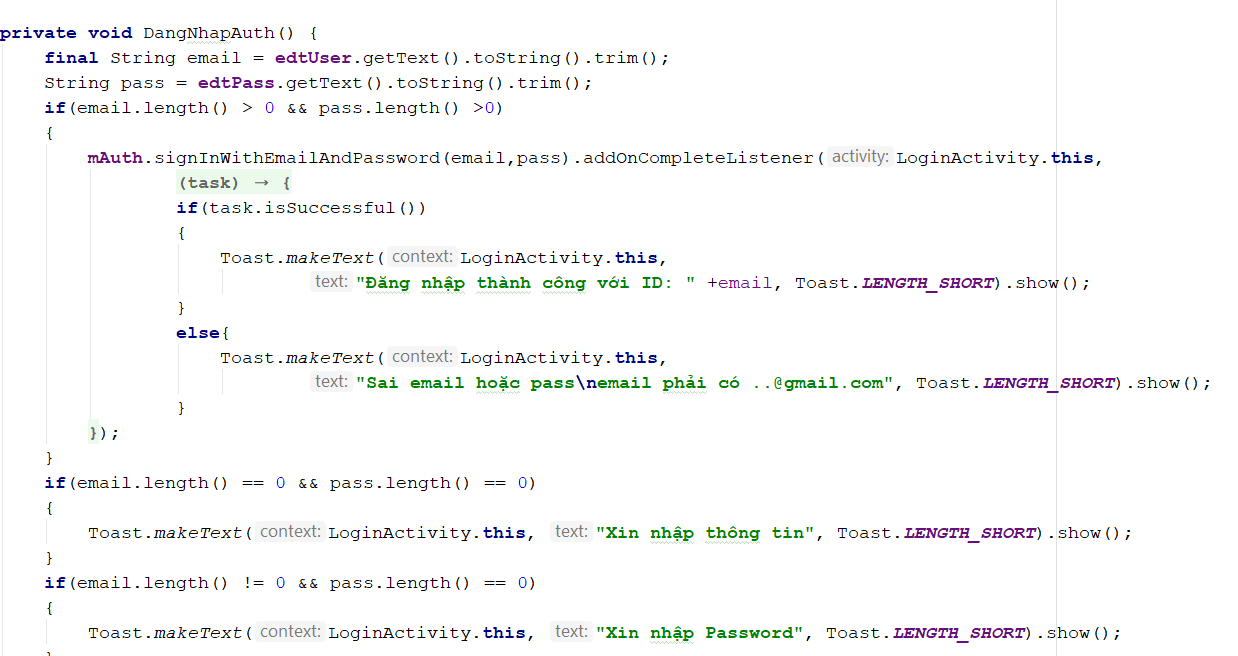
}

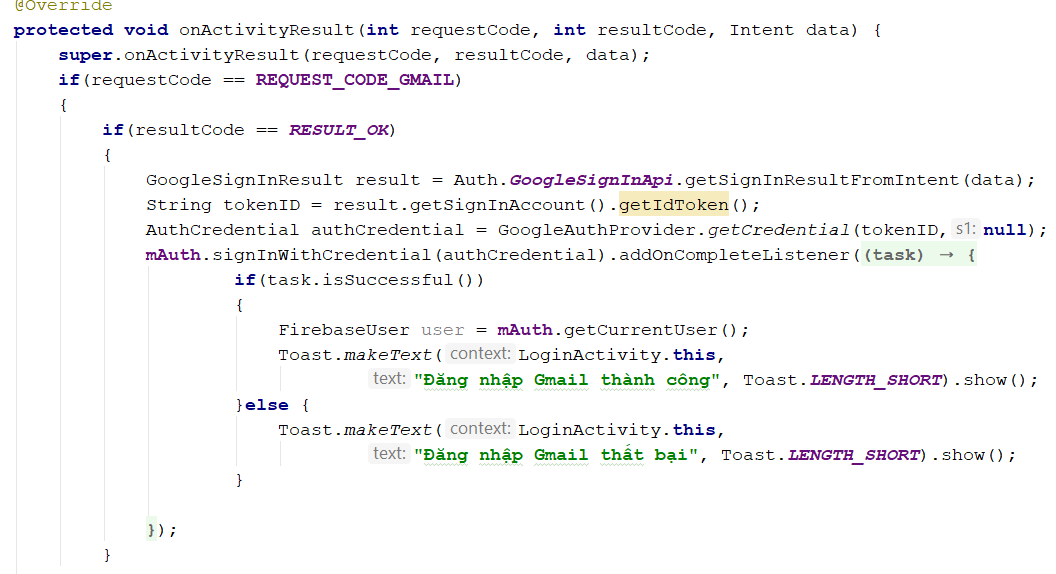
# CHƯƠNG 3. CÀI ĐẶT

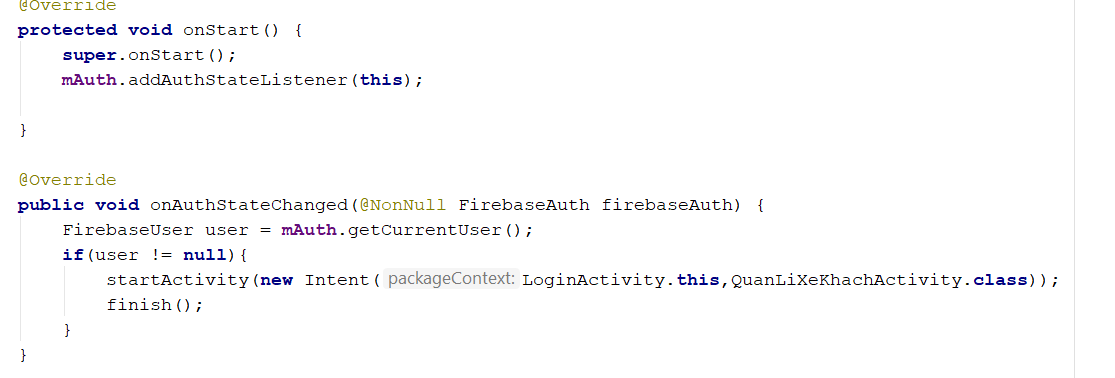
**Màn hình Đăng Nhập**

****

****

****

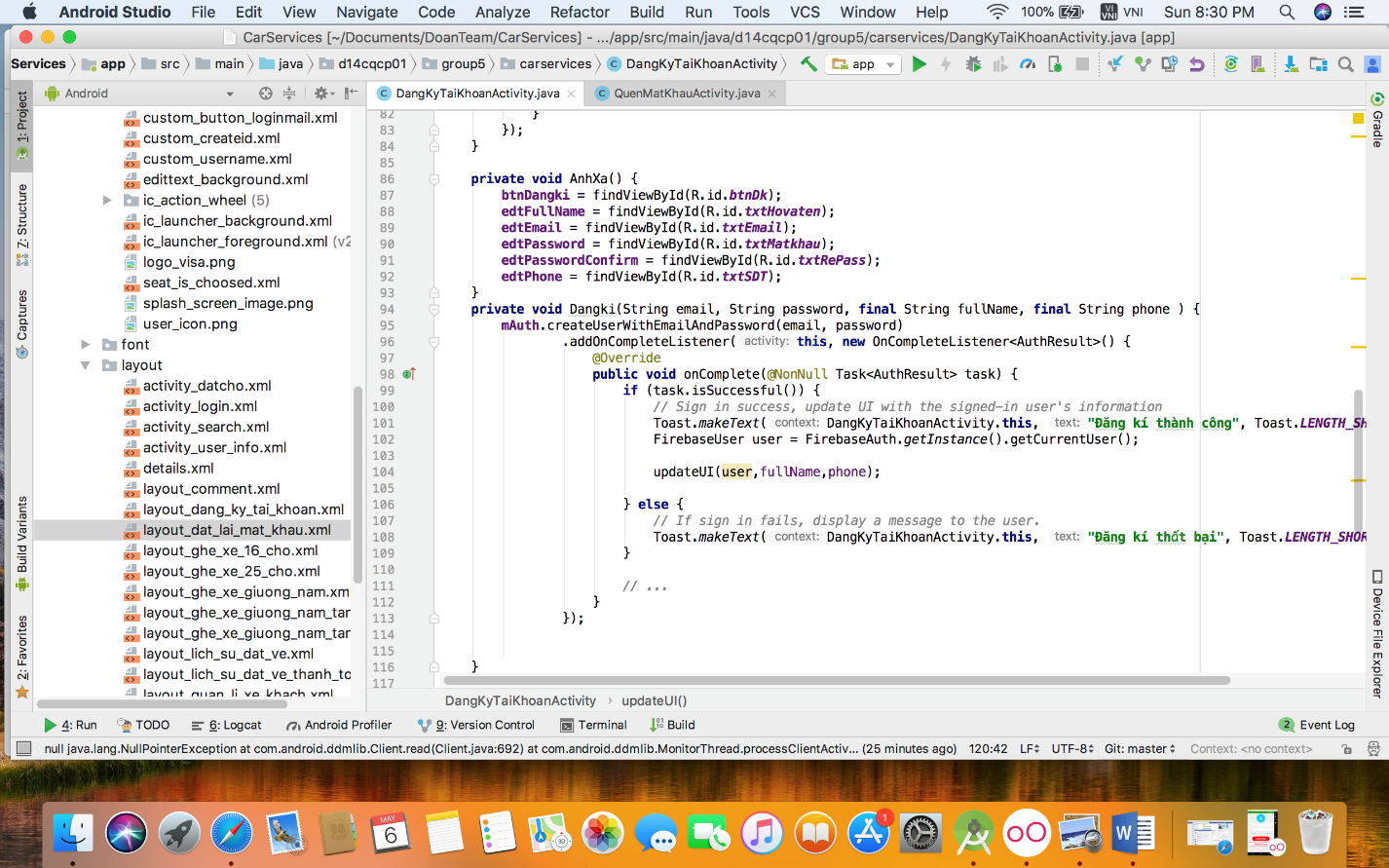
****

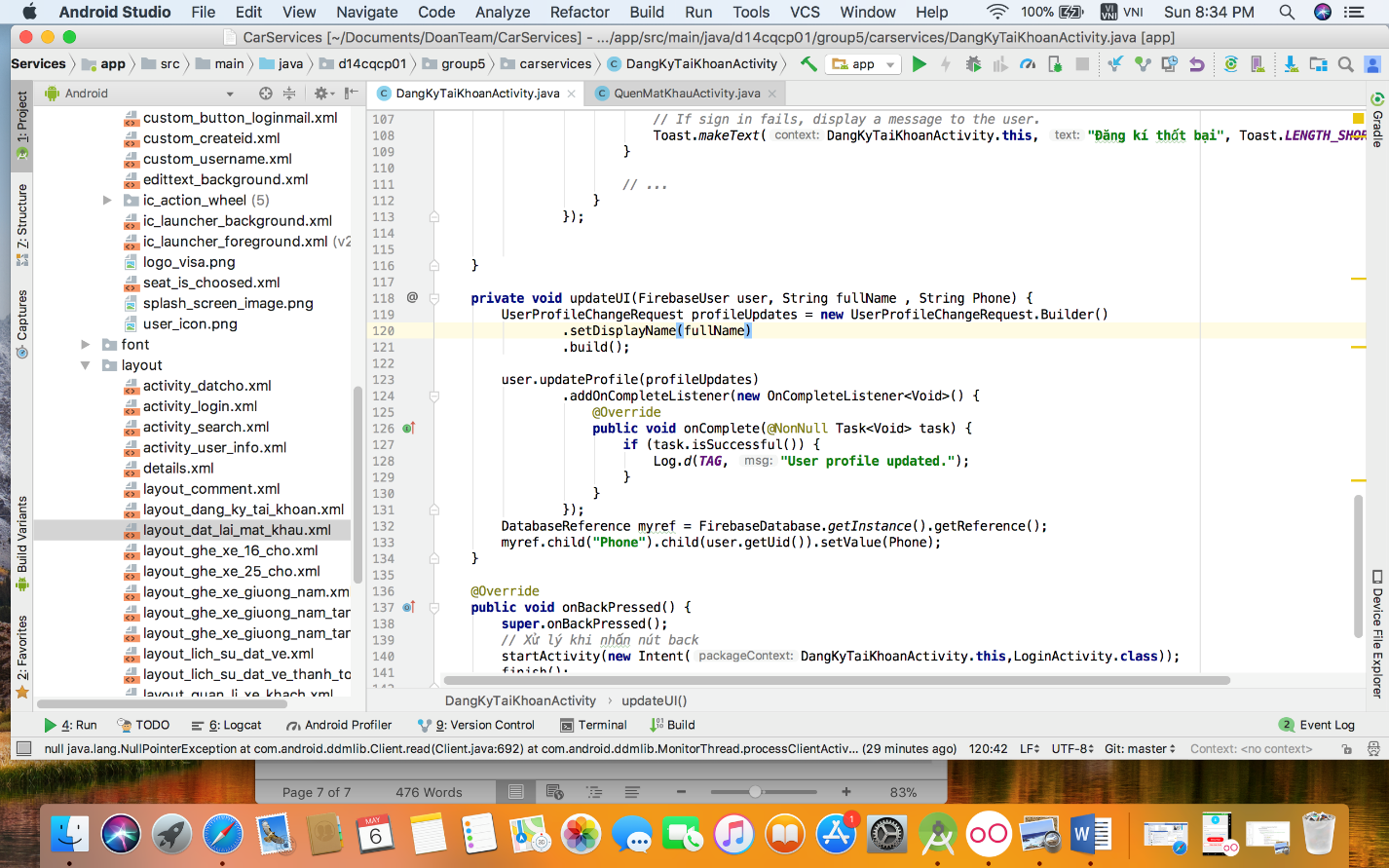
****

**Màn hình đăng kí**

Xữ lí dữ liệu và Up dữ liệu lên Auth và DataBase :

Dangki() && updateUI()

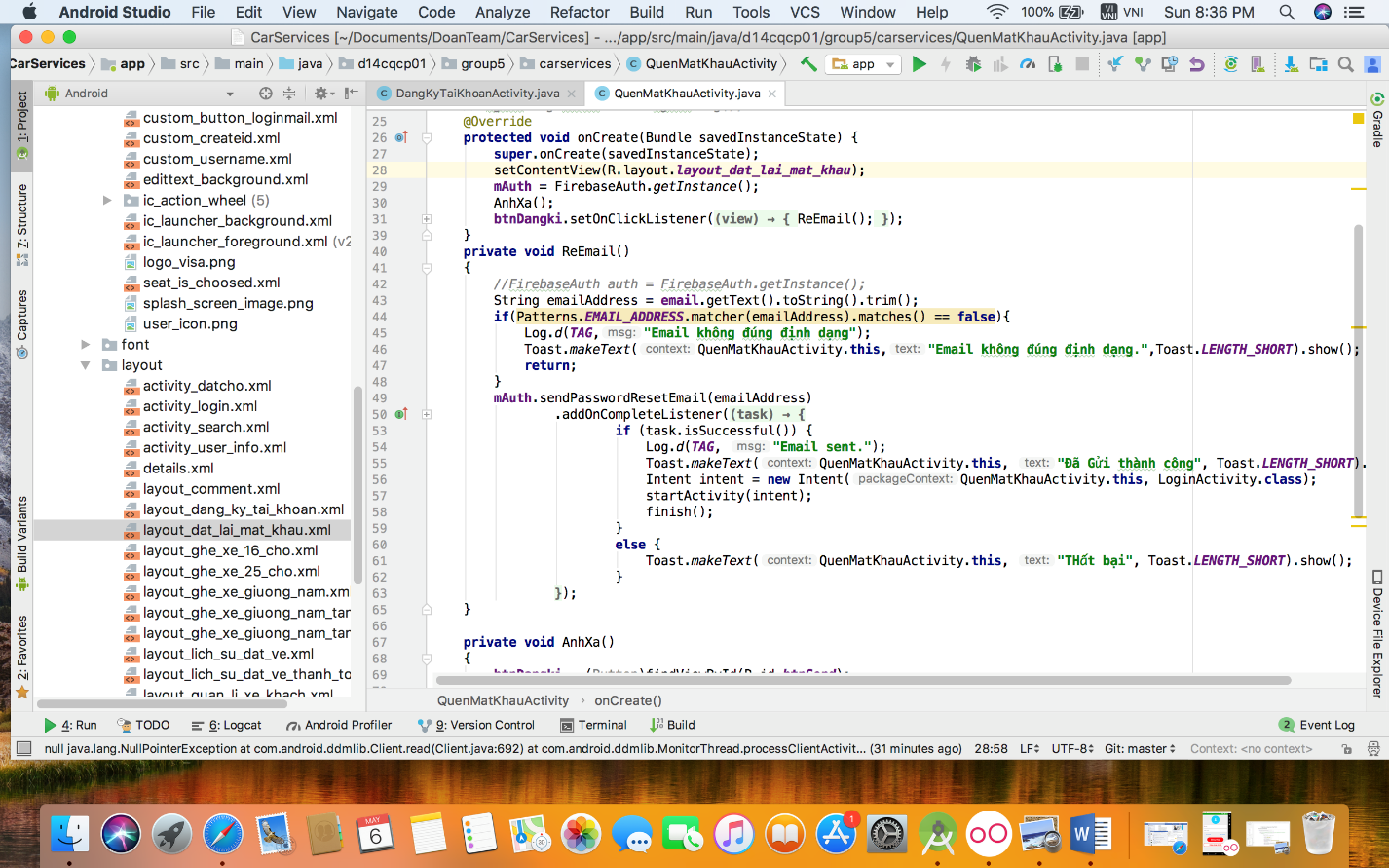




**Màn hình Gửi lại Mật khẩu**

Xử lí và gửi request by AuthFireBase

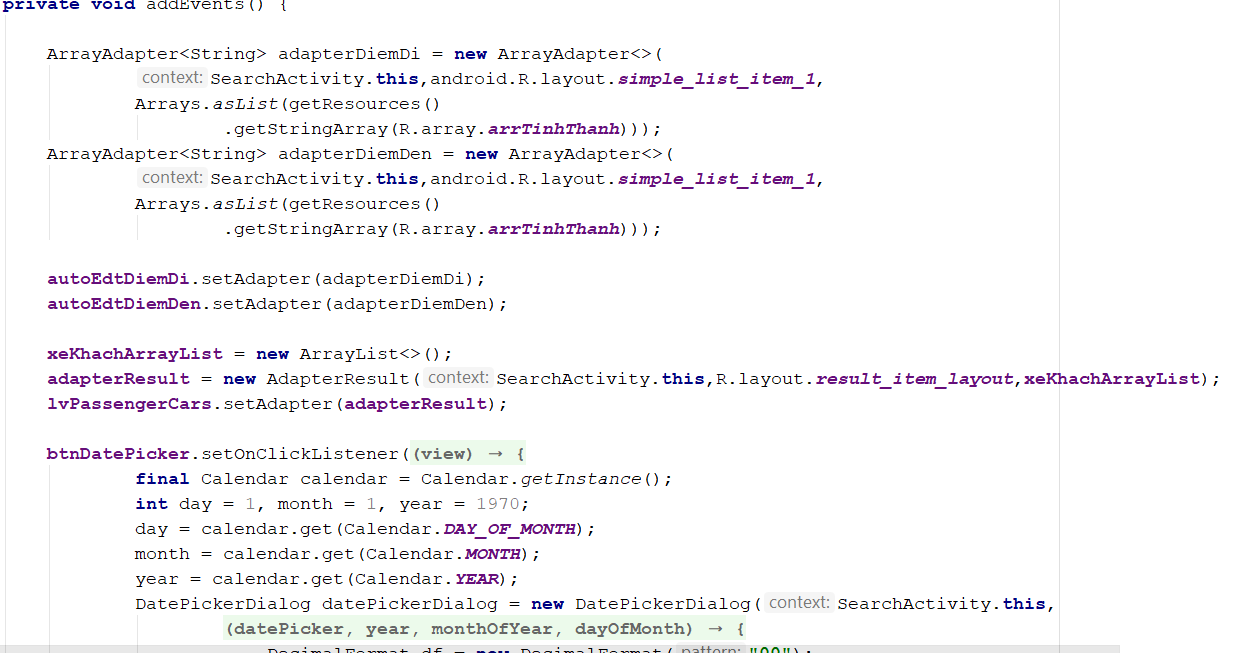
ReEmail()

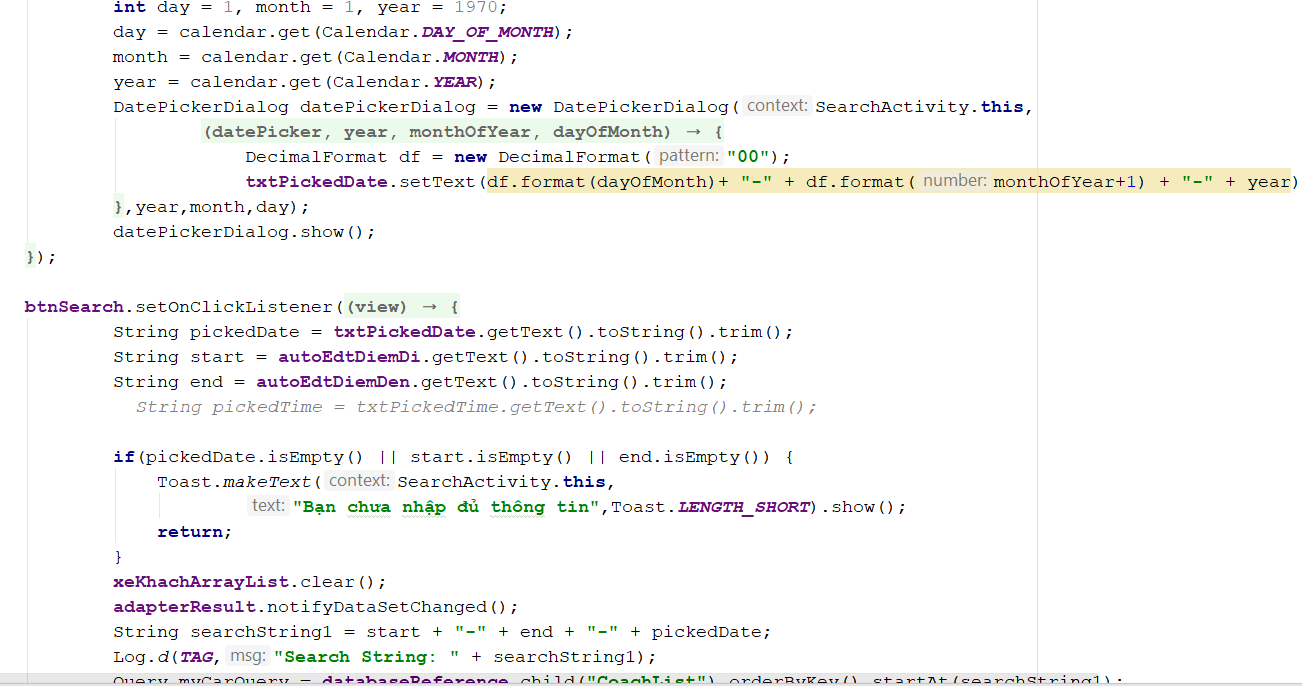


**Màn hình Menu chính**

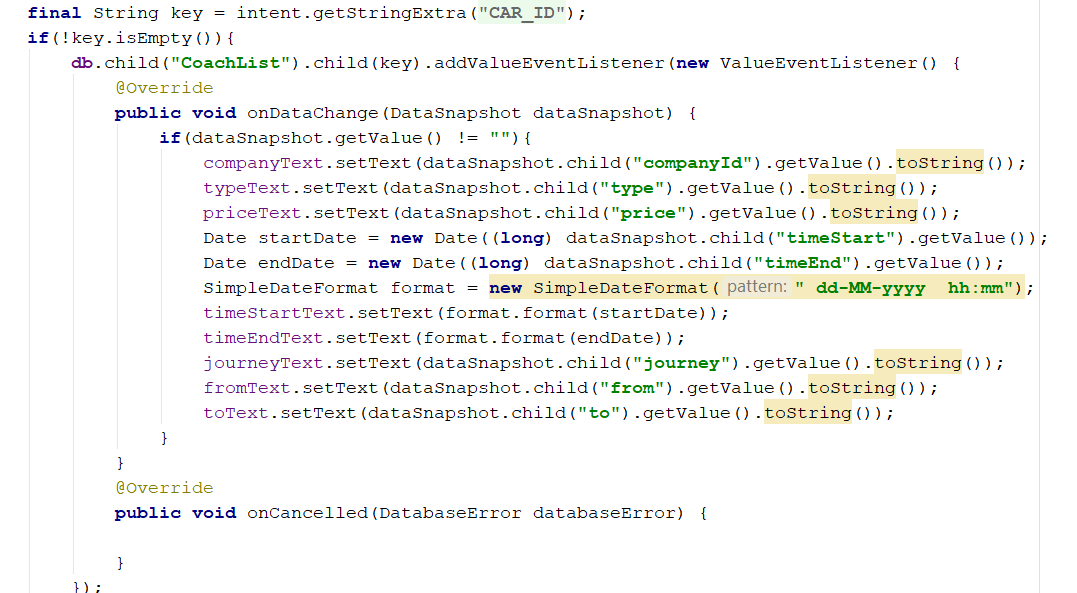
****

**Tìm kiếm xe khách**

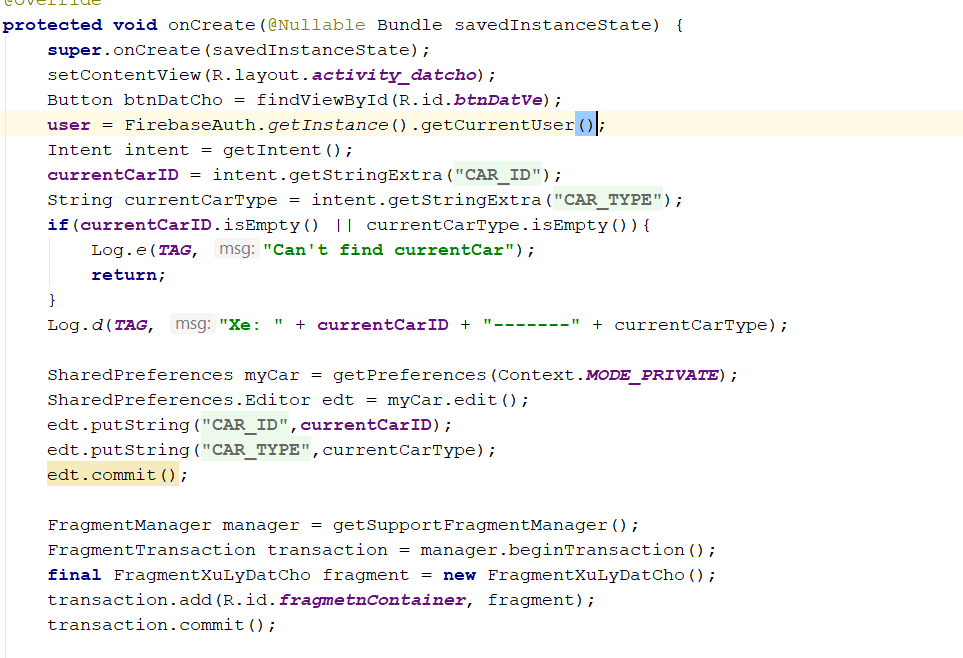
****

****

**Chi tiết xe khách**

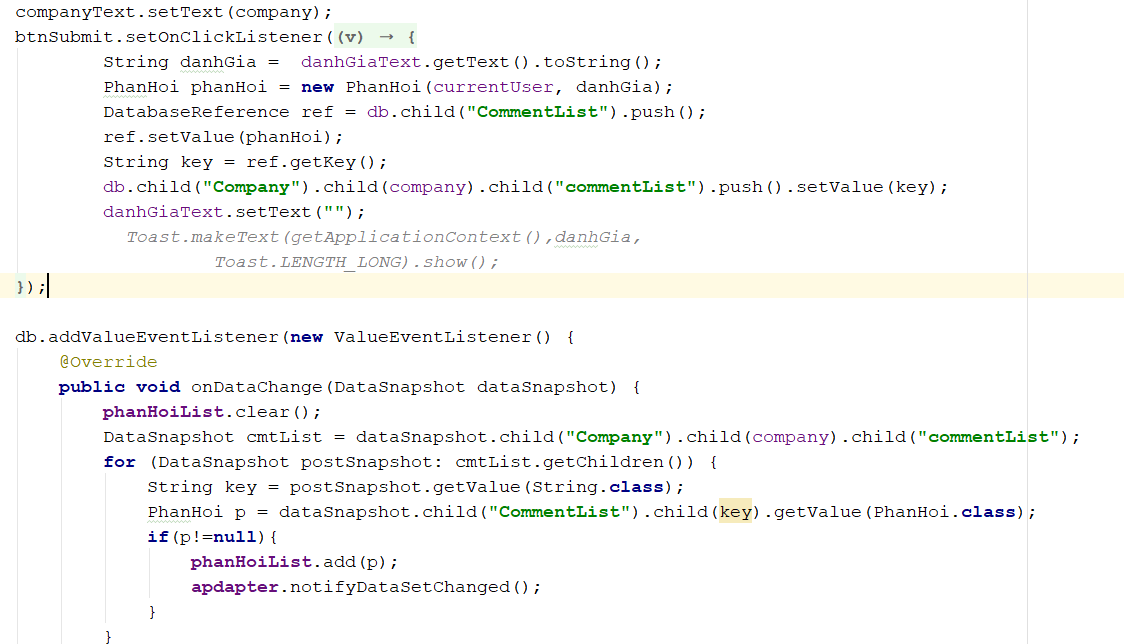
****

**Màn hình đặt vé**

****

****

**Đánh giá xe**

****

**Lịch sử đặt vé**

****

# CHƯƠNG 4.KẾT QUẢ ĐẠT ĐƯỢC

## 4.1 Kết quả đạt được .

• Đăng ký và đăng nhập thành công với Firebase Authentification.

• Truy vấn và tìm kiếm được trên firebase.

• Cho phép đặt vé xe khách

• Giả lập thanh toán Visa.

• Xem lịch sử đặt vé.

## 4.2 Kết luận .

Những việc đã làm đuợc:

- Tìm kiếm lịch sử trên firebase theo yêu cầu.

- Đăng nhập firebase thành công.

- Lưu lịch sử đặt vé thành công.

Những việc chưa làm được:

- Còn nhiều bug.

- Giao diện chưa tối ưu

- Chưa làm được màn hình load động

- Chưa xử lý khi không có internet