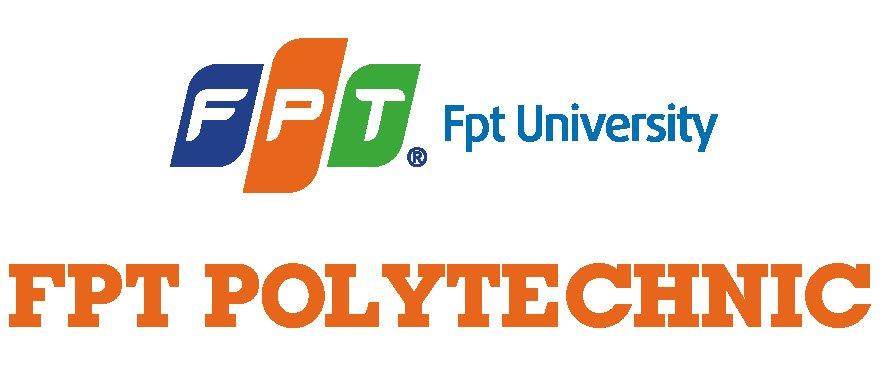
**TRƯỜNG CAO ĐẲNG THỰC HÀNH**

**FPT-POLYTECHNIC TP.HCM**



**BÁO CÁO PRO112 – DỰ ÁN 1**

**App Chat**

**Nhóm thực hiện: Nhóm 2 Lớp: PT12310**

1. **Trần Vinh Quang - Nhóm trưởng**
2. **Hồ Đắc Hiến**
3. **Lý Hải Nam**
4. **Quang Minh**
5. **Phi Long**

**Giảng viên hướng dẫn: Trần Thị Hường**

TP HCM -2014

TP. HỒ CHÍ MINH - 2014

MỤC LỤC

[**1.** **Giới thiệu** 4](#_Toc413744982)

[**2.** **Khảo sát yêu cầu** 5](#_Toc413744983)

[**3.** **Phân tích hệ thống** 6](#_Toc413744984)

[3.1. Chức năng người dùng 6](#_Toc413744985)

[3.2. Phân tích kiến trúc phần cứng 7](#_Toc413744986)

[3.3. Lưu đồ chức năng ứng dụng (Prototype): 8](#_Toc413744987)

[3.4. Thiết kế dữ liệu 8](#_Toc413744988)

[3.4.1. Thiết kế chi tiết 9](#_Toc413744989)

[3.4.2. Câu lệnh SQL cần thiết 9](#_Toc413744990)

[**4.** **Phác thảo giao diện (Mockup):** 11](#_Toc413744991)

[4.1.1. Tất cả ghi chú: 11](#_Toc413744992)

[4.1.2. Activity Thêm ghi chú mới/ Cập nhật ghi chú 11](#_Toc413744993)

[4.1.3. Activity About 11](#_Toc413744994)

[4.1.4. Intro: 11](#_Toc413744995)

[4.1.5. Login: 12](#_Toc413744996)

[**5.** **Thiết kế giao diện ứng dụng** 12](#_Toc413744997)

[5.1.1. Intro Activity: 12](#_Toc413744998)

[5.1.2. LoginActivity: 12](#_Toc413744999)

[5.1.3. MainActivity: 13](#_Toc413745000)

[5.1.4. Activity Thêm ghi chú mới: 13](#_Toc413745001)

[5.1.5. Activity About: 13](#_Toc413745002)

[6. **Xây dựng ứng dụng:** 13](#_Toc413745003)

[6.1. Hệ thống lớp quản lý dữ liệu: 13](#_Toc413745004)

[6.1.1. Lớp DbHelper: 13](#_Toc413745005)

[6.1.2. Lớp quản lý đối tượng (Notes): 13](#_Toc413745006)

[6.1.3. Lớp xử lý đối tượng: 13](#_Toc413745007)

[6.1.4. Lớp quản lí đối tượng: 13](#_Toc413745008)

[6.2. Hệ thống Activity: Code tất cả activitive 14](#_Toc413745009)

[**7.** **Kiểm thử** 14](#_Toc413745010)

[7.1. Mục tiêu kiểm thử: 14](#_Toc413745011)

[7.2. Kỹ thuật kiểm thử 14](#_Toc413745012)

[7.2.1. Công cụ kiểm thử 14](#_Toc413745013)

[7.2.2. Các phương pháp kiểm thử và định nghĩa: 14](#_Toc413745014)

[7.3. Quy trình test 14](#_Toc413745015)

[7.4. TestPlan: 15](#_Toc413745016)

[7.5. TestCase: 16](#_Toc413745017)

[7.5.1. TestNewNote: 16](#_Toc413745018)

[7.5.2. TestNewNotesDetail: 18](#_Toc413745019)

[**8.** **Hướng dẫn sử dụng:** 19](#_Toc413745020)

[8.1. Cài đặt ứng dụng 19](#_Toc413745021)

[8.1.1. Đối với máy tính có cài đặt thiết bị ảo: 19](#_Toc413745022)

[8.2. Sử dụng ứng dụng 19](#_Toc413745023)

[**9.** **Kết luận** 19](#_Toc413745024)

# **Giới thiệu**

Trong bối cảnh nhịp sống hiện đại, mạng xã hội đã trở thành xu thế tất yếu, nhu cầu kết nối và trao đổi qua mạng xã hội ngày càng cao. Đây là ứng dụng giúp chúng ta chia sẻ ảnh trực tuyến.

# **Khảo sát yêu cầu**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Story ID** | **STORY/FEATURE/REQUEST** | | | **Độ ưu tiên** | **B-value** |
| **Người yêu cầu** | **Đặc tả yêu cầu** | **Kết quả mong muốn** |
| RQ01 | **USER** | - Cần một ứng dụng chat | Tôi có thể sử dụng ứng dụng chat để trò chuyện nhanh với bạn bè | 1 |  |
| RQ02 | -Kết bạn với người quen | Có thể kết bạn, chỉ những người đã kết bạn mới có thể nhắn tin với nhau | 1 |  |
| RQ03 | -Chat riêng với bạn bè | Có thể chat giữa hai người với nhau thông qua phngf chat riêng | 2 |  |
| RQ04 | -Hủy kết bạn | Hủy kết bạn để chặn những người dung không mong muốn (spam) | 1 |  |
| RQ05 | Chỉnh sửa tên hiển thị và avatar | Tùy biến thông tin bằng cách chỉnh sửa tên hiển thị và avatar | 1 |  |
| RQ06 | setting | Tùy biến giao diện ứng dụng như thay đổi background, fontsize | 1 |  |
| RQ07 | Xem được thông tin nhôm thực hiện | Tôi có thể chọn để xem các thông tin về nhóm đã viết ra ứng dụng | 3 |  |

# **Phân tích hệ thống**

## Chức năng người dùng

- Đăng ký và đăng nhập bằng Username và Password

- Có thể thay đổi kích thước chữ (Nhỏ, Bình thường, Lớn)

- Có thể thay đổi icon user

- Kết bạn và hủy kết bạn

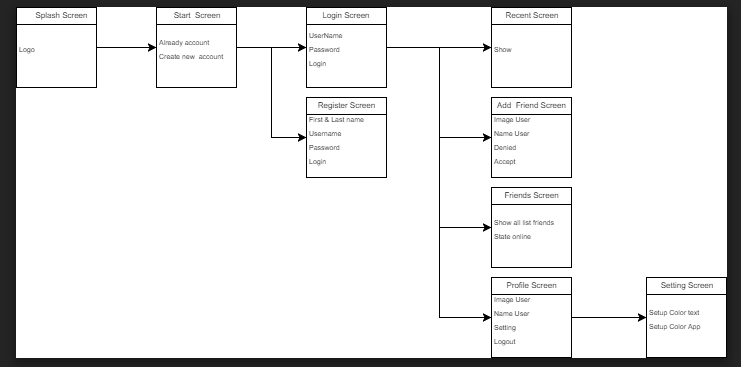
- Tìm kiếm bạn bè

## Phân tích kiến trúc phần cứng

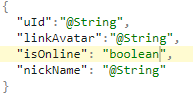
Phần mềm được phát triển sử dụng với các thiết bị hiện có trên thị trường, với yêu cầu phần cứng:

* Thiết bị: Điện thoại/ máy tính bảng đang chạy HĐH Android
* Yêu cầu tối thiểu:
  + RAM 512MB trở lên
  + Chip lõi đơn
  + Android 5.0+
  + Bộ nhớ trống 20mb

## Lưu đồ chức năng ứng dụng (Prototype):

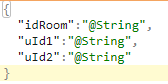


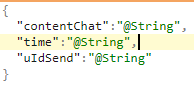
## Thiết kế dữ liệu

Bảng User:

 Bảng Contact:

Bảng Request:

Bảng Roomchat:

Bảng Message:

### Thiết kế chi tiết

**Bảng User:**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| uId | String | Id người dùng |
| linkAvatar | String | Đường dẫn ảnh đại diện người dùng |
| isOnline | boolean | Trạng thái online người dùng |
| nickName | String | Tên hiển thị người dùng |

**Bảng Contact:**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| uId | String | Id bạn bè |

**Bảng Request:**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| uId | String | Id người gửi lời mời kết bạn |

**Bảng RoomChat:**

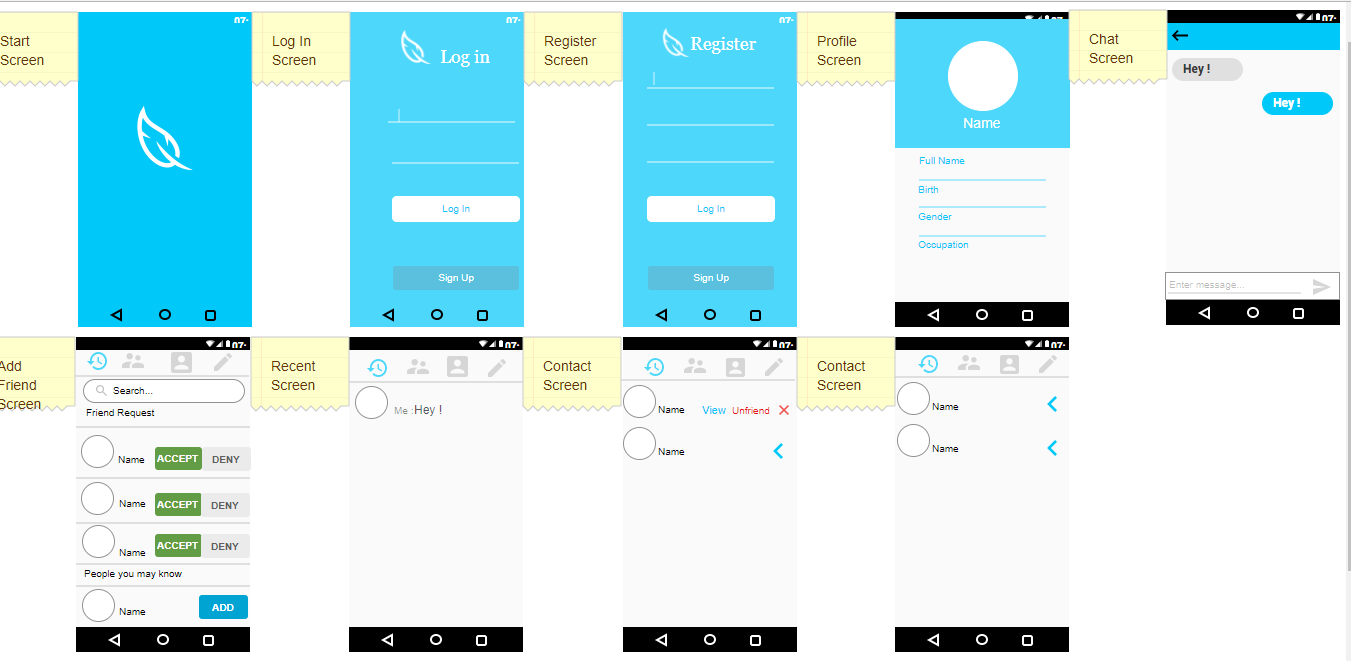
|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| idRoom | String | Id roomchat |
| uId1 | String | Id người chat thứ nhất |
| uId2 | String | Id người chat thứ hai |

**Bảng Message:**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| contentChat | String | Nội dung chat |
| time | String | Thời gian chat |
| uIdSend | String | Id người gửi |

### Câu lệnh SQL cần thiết

# **Phác thảo giao diện (Mockup):**

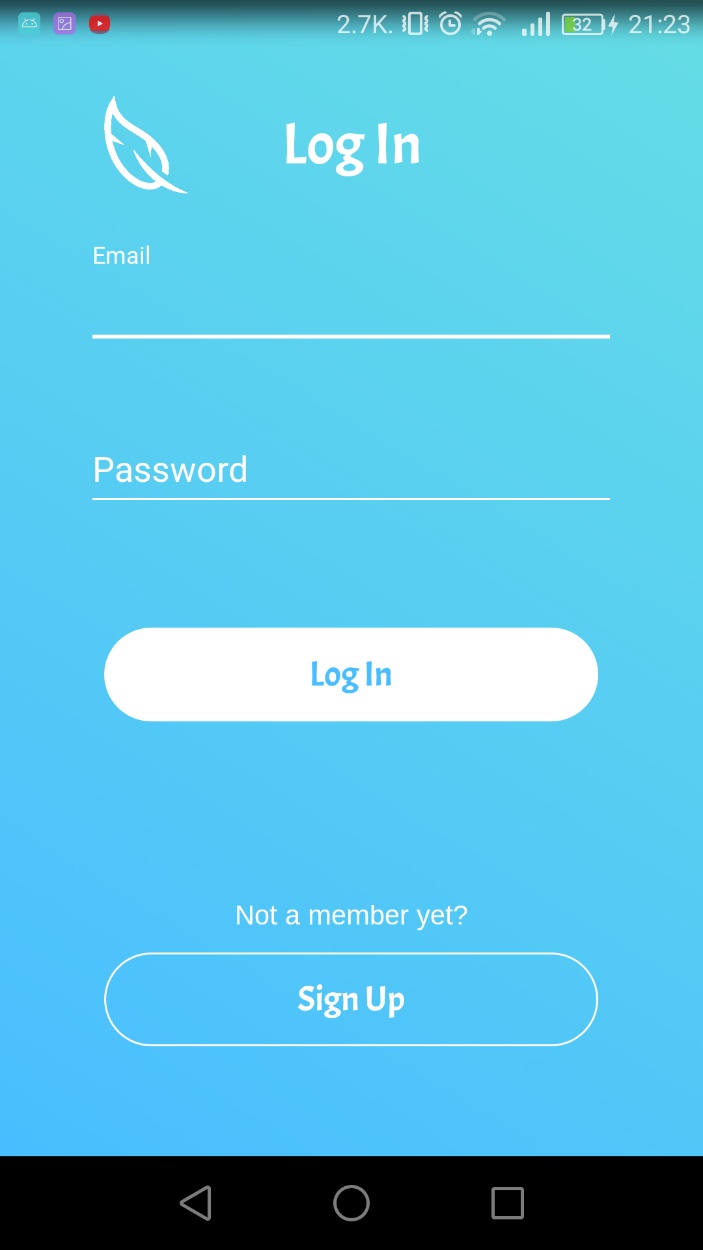


# **Thiết kế giao diện ứng dụng**

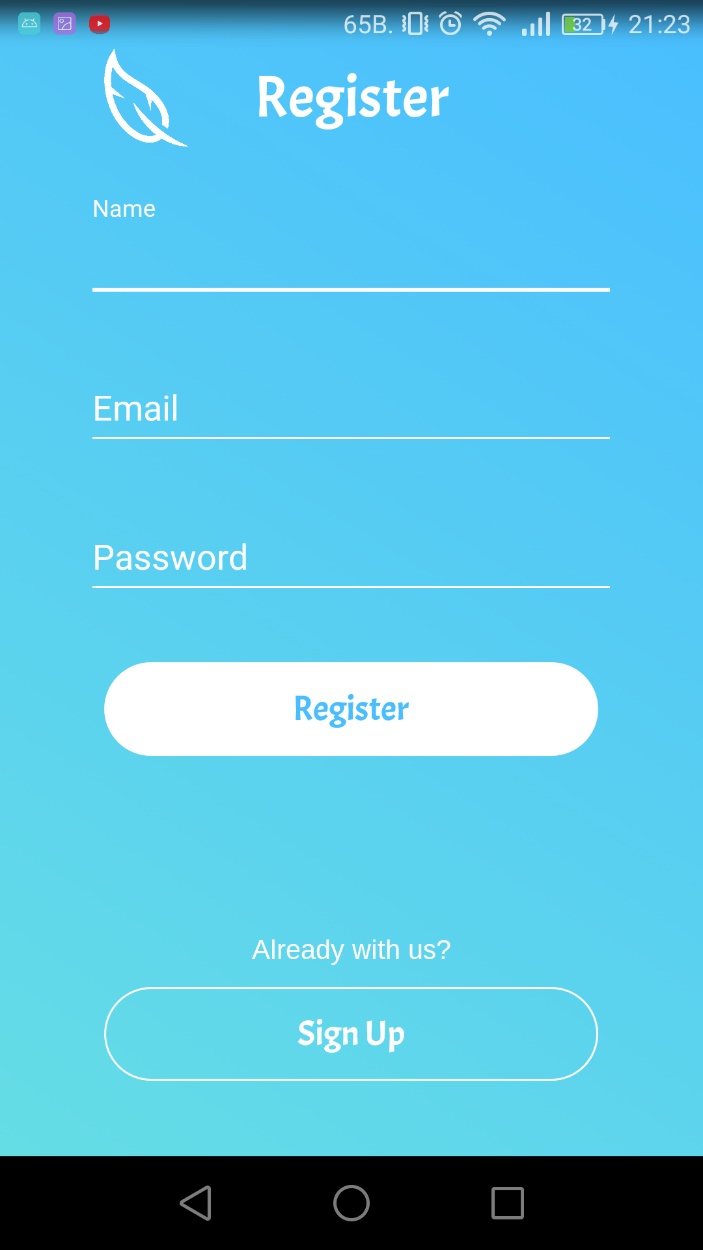
### Intro Activity:



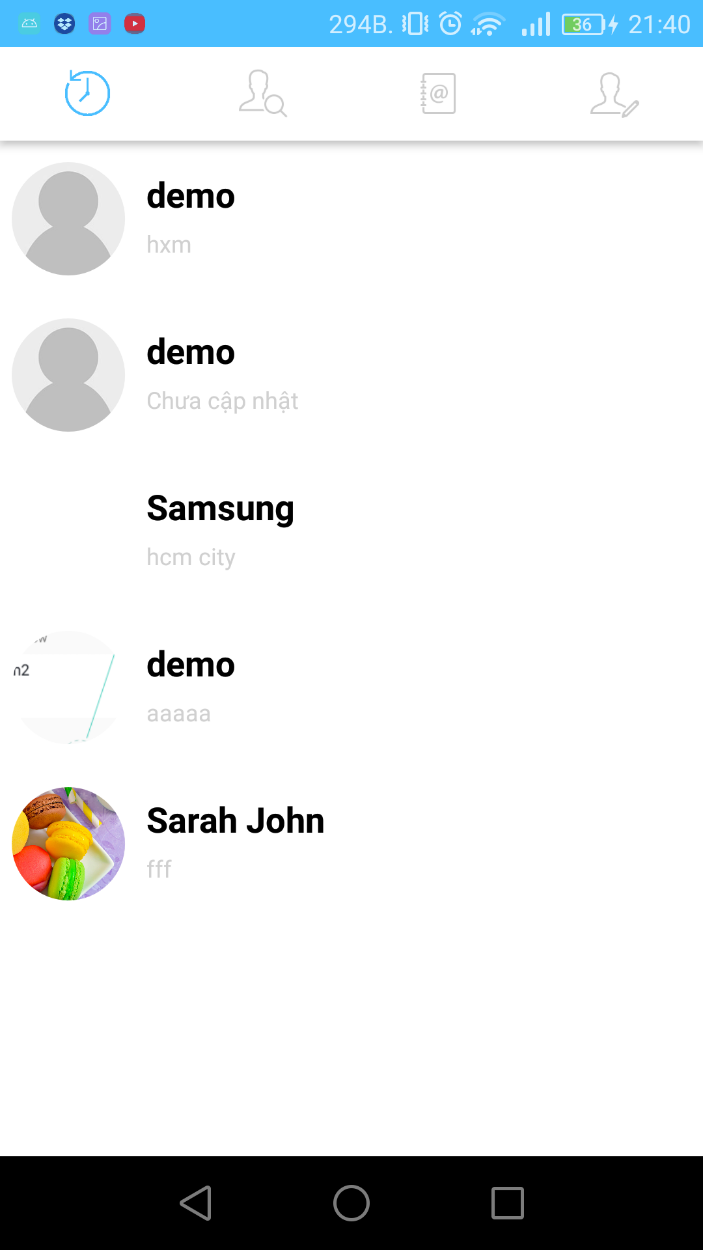
### LoginActivity:



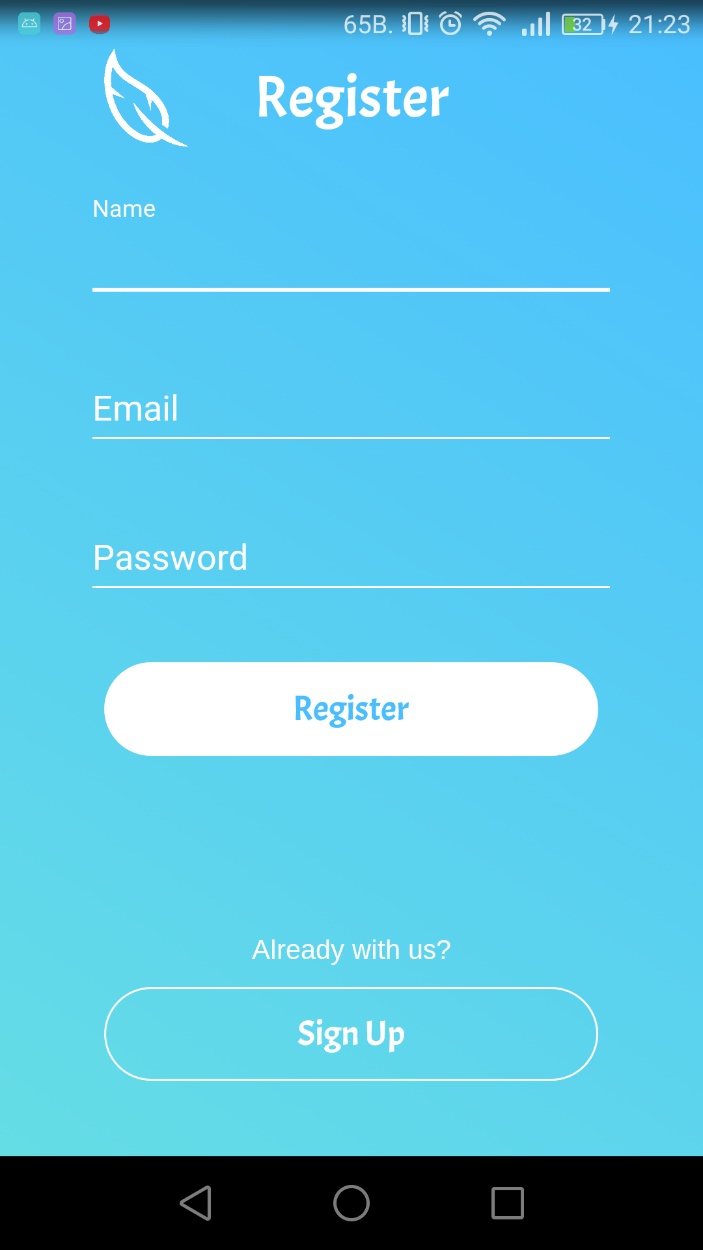
### RegisterActivity:



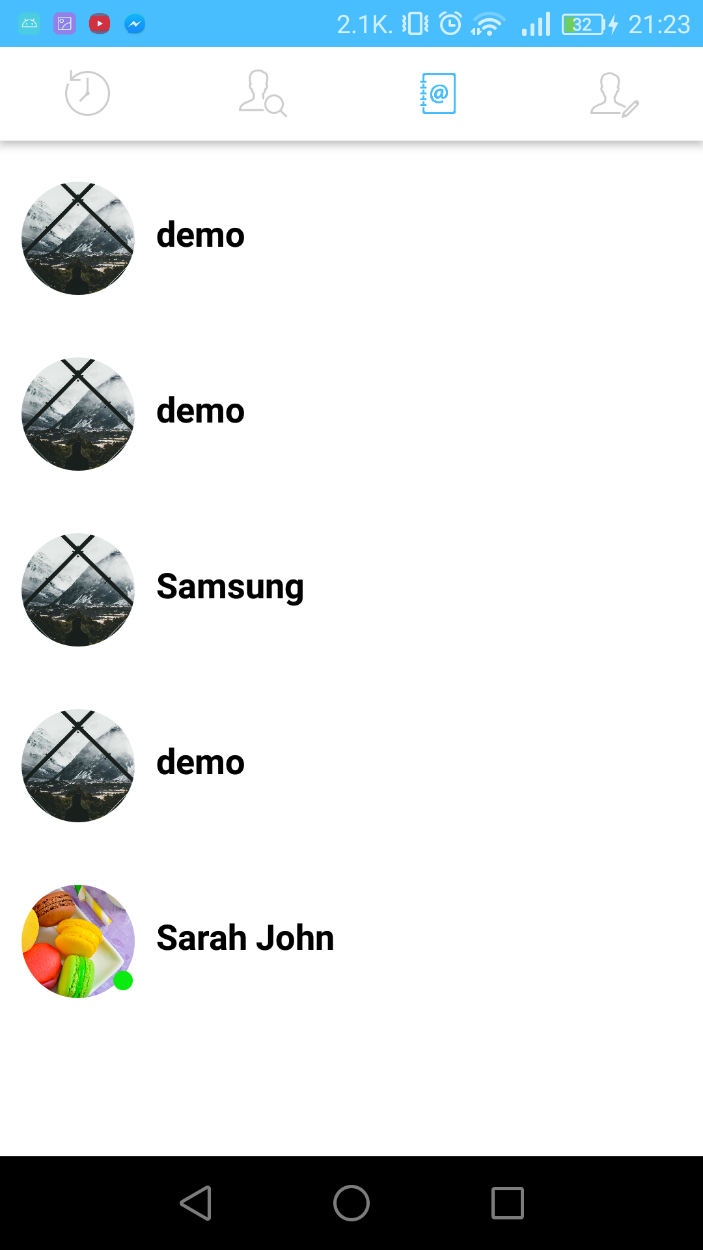
### RecentFragment



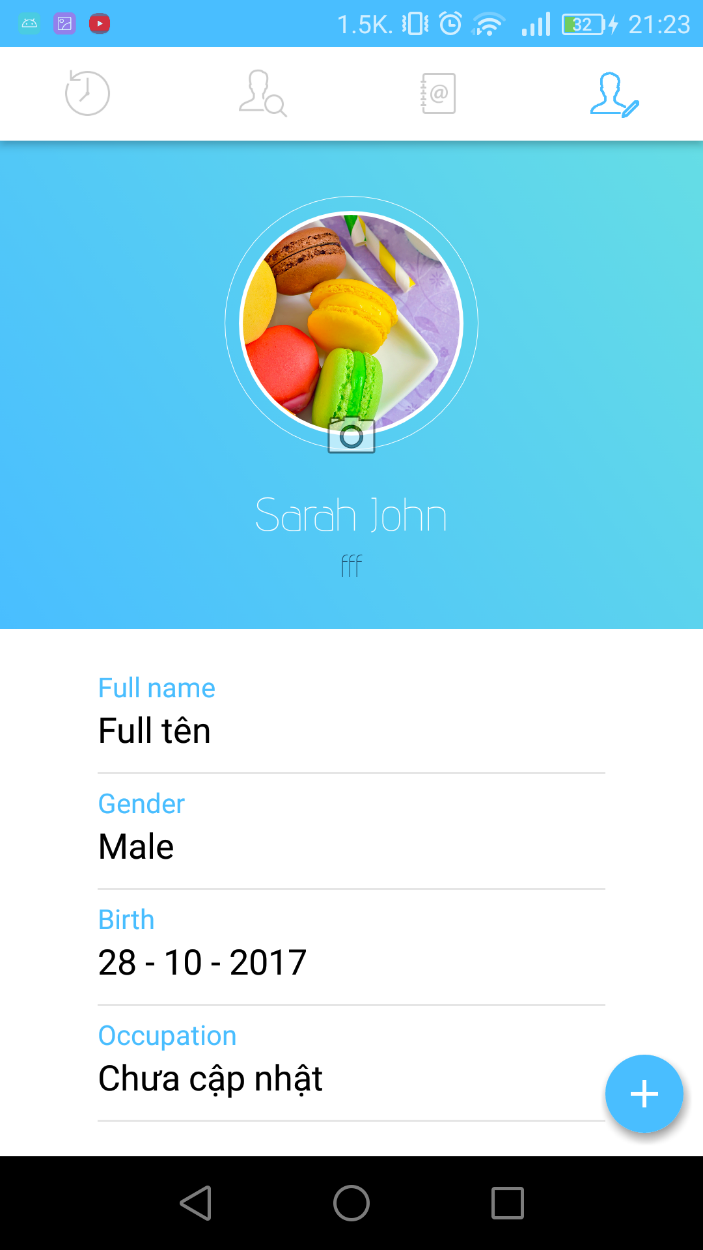
### FindFriendsFragment



### ContactFragment



### ProfileFragment



### 

# **Xây dựng ứng dụng:**

## Hệ thống lớp quản lý dữ liệu:

### Lớp DbHelper:

Lớp DBHelper được dùng để tạo, quản lí CSDL.

|  |
| --- |
| **Code…………………..** |

### Lớp quản lý đối tượng (Notes):

Lớp quản lí đối tượng được dùng để quản lí, khai báo thành phần của các đối tượng, khai báo các phương thức getter, setter cho đối tượng.

|  |
| --- |
| **Contact:**  **public class** Contact {  String **mIdFriend**;   **public** Contact(String mIdFriend) {  **this**.**mIdFriend** = mIdFriend;  }   **public** Contact() {  }   **public** String getmIdFriend() {  **return mIdFriend**;  }   **public void** setmIdFriend(String mIdFriend) {  **this**.**mIdFriend** = mIdFriend;  } }  Message  **public** Message() {  }   **public** String getmContent() {  **return mContent**;  }   **public void** setmContent(String mContent) {  **this**.**mContent** = mContent;  }   **public long** getmTime() {  **return mTime**;  }   **public void** setmTime(**long** mTime) {  **this**.**mTime** = mTime;  }   **public** String getmIdSend() {  **return mIdSend**;  }   **public void** setmIdSend(String mIdSend) {  **this**.**mIdSend** = mIdSend;  } }  Rom chat  **public** RoomChat() {  }   **public** String getmIdRoom() {  **return mIdRoom**;  }   **public void** setmIdRoom(String mIdRoom) {  **this**.**mIdRoom** = mIdRoom;  }   **public** String getmID1() {  **return mID1**;  }   **public void** setmID1(String mID1) {  **this**.**mID1** = mID1;  }   **public** String getmID2() {  **return mID2**;  }   **public void** setmID2(String mID2) {  **this**.**mID2** = mID2;  } }  Recent  **public** Recent() {  }   **public void** setmIdTheOther(String mIdTheOther) {  **this**.**mIdTheOther** = mIdTheOther;  }   **public** String getmTheOtherLinkImage() {  **return mTheOtherLinkImage**;  }   **public void** setmTheOtherLinkImage(String mTheOtherLinkImage) {  **this**.**mTheOtherLinkImage** = mTheOtherLinkImage;  }   **public** String getmTheOtherName() {  **return mTheOtherName**;  }   **public void** setmTheOtherName(String mTheOtherName) {  **this**.**mTheOtherName** = mTheOtherName;  }   **public** String getIdRoom() {  **return idRoom**;  }   **public void** setIdRoom(String idRoom) {  **this**.**idRoom** = idRoom;  } }  RequestFriend  **public class** RequestFriend {  **private** String **mIdRequest**;   **public** RequestFriend(String mIdRequest) {  **this**.**mIdRequest** = mIdRequest;  }   **public** RequestFriend() {  }   **public** String getmIdRequest() {  **return mIdRequest**;  }   **public void** setmIdRequest(String mIdRequest) {  **this**.**mIdRequest** = mIdRequest;  } }  User  **public** String getmFullName() {  **return mFullName**;  }   **public void** setmFullName(String mFullName) {  **this**.**mFullName** = mFullName;  }   **public** String getmGender() {  **return mGender**;  }   **public void** setmGender(String mGender) {  **this**.**mGender** = mGender;  }   **public** User() {  }   **public** String getmProvince() {  **return mProvince**;  }   **public void** setmProvince(String mProvince) {  **this**.**mProvince** = mProvince;  }   **public** String getmOccupation() {  **return mOccupation**;  }   **public void** setmOccupation(String mOccupation) {  **this**.**mOccupation** = mOccupation;  }   **public** String getmBirth() {  **return mBirth**;  }   **public void** setmBirth(String mBirth) {  **this**.**mBirth** = mBirth;  }   **public** String getmId() {  **return mId**;  }   **public void** setmId(String mId) {  **this**.**mId** = mId;  }   **public** String getmName() {  **return mName**;  }   **public void** setmName(String mName) {  **this**.**mName** = mName;  }   **public** String getmLinkImage() {  **return mLinkImage**;  }   **public void** setmLinkImage(String mLinkImage) {  **this**.**mLinkImage** = mLinkImage;  }   **public boolean** ismIsOnline() {  **return mIsOnline**;  }   **public void** setmIsOnline(**boolean** mIsOnline) {  **this**.**mIsOnline** = mIsOnline;  } } |

### Lớp xử lý đối tượng:

Lớp xử lí đối tượng được dùng để quản lí hệ thống chức năng, quản lý các hàm cập nhật, thêm, xóa , sửa (chức năng cơ bản) của ứng dụng.

|  |
| --- |
| **Code…………………..** |

### Lớp quản lí đối tượng:

Dùng để quản lý đối tượng của thanh menu Navigation Drawer.

|  |
| --- |
|  |

## Hệ thống Activity: Code tất cả activitive

### SplashScreen.java

|  |
| --- |
| **public class** SplashScreen **extends** AppCompatActivity **implements** FirebaseAuth.AuthStateListener {  LinearLayout **myLayout**;  AnimationDrawable **animationDrawable**;  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_splash\_screen***);  **myLayout** = (LinearLayout) findViewById(R.id.***myLayout***);   **animationDrawable** = (AnimationDrawable) **myLayout**.getBackground();  **animationDrawable**.setEnterFadeDuration(1000);  **animationDrawable**.setExitFadeDuration(1000);  **animationDrawable**.start();   **final** Handler handler = **new** Handler();  handler.postDelayed(**new** Runnable() {  @Override  **public void** run() {  FirebaseAuth.*getInstance*().addAuthStateListener(SplashScreen.**this**);   }  }, 2000);  }   @Override  **protected void** onResume() {  **super**.onResume();  }   @Override  **protected void** onStart() {  **super**.onStart();  }   @Override  **protected void** onStop() {  **super**.onStop();  FirebaseAuth.*getInstance*().removeAuthStateListener(**this**);  }   @Override  **protected void** onDestroy() {  **super**.onDestroy();   }    **void** goToLoggedInActivity() {  Intent intent = **new** Intent(getApplicationContext(), LoggedInActivity.**class**);  startActivity(intent);  finish();  }   **void** goToStartActivity() {  Intent intent = **new** Intent(getApplicationContext(), StartActivity.**class**);  startActivity(intent);  finish();  }   @Override  **public void** onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {  FirebaseUser user = firebaseAuth.getCurrentUser();  Log.*e*(**"Check session Splash"**, String.*valueOf*(user!=**null**));  **if** (user != **null**) {  *//đã từng đăng nhập* goToLoggedInActivity();  } **else** {  *//chưa từng có phiên đăng nhập trước hoặc đã logout* goToStartActivity();  }  } } |

### SearchActivity.java

|  |
| --- |
| **public class** SearchActivity **extends** AppCompatActivity **implements** TextWatcher {  EditText **editText**;  TextView **tvResult**;  **private** ListView **listview**;  DatabaseReference **userReference**;  Query **queryByName**;  Query **queryByNick**;  FindAdapter **findAdapter**;  ArrayList<User> **userArrayList**;  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_search***);   initData();  initControl();  initDisplay();    }   **private void** initData() {  **userReference** = FirebaseDatabase.*getInstance*().getReference().child(**"Users"**);  **queryByName** = **userReference**.orderByChild(**"mFullName"**);  **queryByNick** = **userReference**.orderByChild(**"mName"**);   }   **private void** initDisplay() {  initListView();  }   **private void** initListView() {  **userArrayList**= **new** ArrayList<>();  **findAdapter**= **new** FindAdapter(getApplicationContext(),**userArrayList**);  **listview**.setAdapter(**findAdapter**);  }   **private void** initControl() {  **listview** = findViewById(R.id.***lv\_find\_search\_activity***);  **editText** = findViewById(R.id.***edtSearch***);  **editText**.addTextChangedListener(**this**);  **tvResult**=findViewById(R.id.***tv\_result\_status***);  }    @Override  **public void** beforeTextChanged(CharSequence charSequence, **int** i, **int** i1, **int** i2) {   }   @Override  **public void** onTextChanged(CharSequence charSequence, **int** i, **int** i1, **int** i2) {  String contentSearch = String.*valueOf*(charSequence);  **if** (contentSearch.length() >= 1) {  **userArrayList**.clear();  updateLayout();  searchByNickName(contentSearch);  }    }   @Override  **public void** afterTextChanged(Editable editable) {   }   **private void** searchByNickName(**final** String contentSearch) {  **userReference**.addListenerForSingleValueEvent(**new** ValueEventListener() {  @Override  **public void** onDataChange(DataSnapshot dataSnapshot) {  **if** (dataSnapshot.getValue() != **null**) {  **for** (DataSnapshot userSnapshot : dataSnapshot.getChildren()) {  String name = userSnapshot.child(**"mName"**).getValue().toString();  String id = userSnapshot.child(**"mId"**).getValue().toString();  **if** (name.contains(contentSearch)&&!id.equals(Setup.*currentUser*.getmId())){  User user = userSnapshot.getValue(User.**class**);  **userArrayList**.add(user);  }    }  updateLayout();  *//Toast.makeText(getApplicationContext(),dataSnapshot.getValue().toString(),Toast.LENGTH\_SHORT).show();* }   }   @Override  **public void** onCancelled(DatabaseError databaseError) {   }  });  }  **private void** updateLayout(){  **findAdapter**.notifyDataSetChanged();  **if** (**userArrayList**.size()==0){  **tvResult**.setText(**"There is no result matches your searching"**);  }**else** {  **tvResult**.setText(**"Result"**);  }  } } |

### RegisterActivity.java

|  |
| --- |
| **public class** RegisterActivity **extends** DataAdapter **implements** View.OnClickListener, FirebaseAuth.AuthStateListener {  **private** TextView **tv\_have\_account\_register\_activity**;  **private** TextInputLayout **tip\_name\_user\_register\_activity**, **tip\_email\_user\_register\_activity**, **tip\_password\_user\_register\_activity**;  **private** Button **btn\_create\_acc\_register\_activity**, **btn\_haveaccount\_login\_activity**;  **private** FirebaseAuth **mAuth**;  **private** FirebaseDatabase **firebaseDatabase**;  **private** DatabaseReference **databaseReference**;  **private** String **name**, **email**, **password**;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_register***);  initControl();  initData();  initEvent();  }   @Override  **protected void** onStart() {  **super**.onStart();   }   @Override  **protected void** onStop() {  **super**.onStop();  **mAuth**.removeAuthStateListener(**this**);  }   **private void** initEvent() {   }   **private void** createAccount(**final** String email, **final** String password, **final** String name) {  **mAuth**.createUserWithEmailAndPassword(email, password)  .addOnCompleteListener(**new** OnCompleteListener<AuthResult>() {  @Override  **public void** onComplete(@NonNull Task<AuthResult> task) {  **if** (task.isSuccessful()) {  **progressDialog**.dismiss();  Toast.*makeText*(RegisterActivity.**this**, **"thành công:"**, Toast.***LENGTH\_SHORT***).show();  writeNewUser(email, name, password);  } **else** {  **progressDialog**.dismiss();  Toast.*makeText*(RegisterActivity.**this**, **"Lỗi: "** + task.getException().getMessage(), Toast.***LENGTH\_SHORT***).show();  }  *// ...* }  });  }    **private void** writeNewUser(**final** String email, String name, **final** String password) {  FirebaseUser firebaseUser = FirebaseAuth.*getInstance*().getCurrentUser();  String uid = firebaseUser.getUid();  **databaseReference** = FirebaseDatabase.*getInstance*().getReference().child(**"Users"**).child(uid);   **final** User user = **new** User(uid, name, User.***DEFAULT\_AVATAR***, **false**, User.***NOT\_UP\_DATE\_YET***, User.***NOT\_UP\_DATE\_YET***, User.***NOT\_UP\_DATE\_YET***, User.***NOT\_UP\_DATE\_YET***, User.***NOT\_UP\_DATE\_YET***);   **databaseReference**.setValue(user).addOnCompleteListener(**new** OnCompleteListener<Void>() {  @Override  **public void** onComplete(@NonNull Task<Void> task) {  **if** (task.isSuccessful()) {  **mAuth**.addAuthStateListener(RegisterActivity.**this**);   } **else** {  Toast.*makeText*(RegisterActivity.**this**, **"Lỗi: "** + task.getException().getMessage(), Toast.***LENGTH\_SHORT***).show();  }    }  });    }   **private void** initData() {  **mAuth** = FirebaseAuth.*getInstance*();  }   **private void** initControl() {  **btn\_haveaccount\_login\_activity** = findViewById(R.id.***btn\_haveaccount\_login\_activity***);  **tip\_name\_user\_register\_activity** = findViewById(R.id.***tip\_name\_user\_register\_activity***);  **tip\_email\_user\_register\_activity** = findViewById(R.id.***tip\_email\_user\_register\_activity***);  **tip\_password\_user\_register\_activity** = findViewById(R.id.***tip\_password\_user\_register\_activity***);  **btn\_create\_acc\_register\_activity** = findViewById(R.id.***btn\_create\_acc\_register\_activity***);   **btn\_haveaccount\_login\_activity**.setOnClickListener(**this**);  **btn\_create\_acc\_register\_activity**.setOnClickListener(**this**);   }   @Override  **public void** onClick(View v) {  **switch** (v.getId()) {  **case** R.id.***btn\_create\_acc\_register\_activity***:  waitProgress(**"Creating account..."**);  getDataEdt();  **break**;  **case** R.id.***btn\_haveaccount\_login\_activity***:  Intent intent = **new** Intent(RegisterActivity.**this**, LoginActivity.**class**);  startActivity(intent);  **break**;   **default**:  **break**;  }  }   **private void** goToLoggedInActivity() {  Intent intent = **new** Intent(**this**, LoggedInActivity.**class**);  startActivity(intent);  finishAffinity();  }   **private void** getDataEdt() {  **name** = **tip\_name\_user\_register\_activity**.getEditText().getText().toString();  **email** = **tip\_email\_user\_register\_activity**.getEditText().getText().toString();  **password** = **tip\_password\_user\_register\_activity**.getEditText().getText().toString();  hideKeyBoard(); *// if (matcher.matches()) {* **if** (**name**.equals(**""**)) {  **tip\_name\_user\_register\_activity**.setError(**"Please enter the email"**);  **progressDialog**.dismiss();  } **else** {  **tip\_name\_user\_register\_activity**.setError(**""**);  }   **if** (**email**.equals(**""**)) {  **tip\_email\_user\_register\_activity**.setError(**"Please enter the email"**);  **progressDialog**.dismiss();  } **else** {  **tip\_email\_user\_register\_activity**.setError(**""**);  }   **if** (**password**.equals(**""**)) {  **tip\_password\_user\_register\_activity**.setError(**"Please enter the password"**);  **progressDialog**.dismiss();  } **else** {  **tip\_password\_user\_register\_activity**.setError(**""**);  }  **if** (!**email**.isEmpty() && !**password**.isEmpty()) {  createAccount(**email**, **password**, **name**);   }    }   @Override  **public void** onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {  FirebaseUser firebaseUser = firebaseAuth.getCurrentUser();  **boolean** hasUser = firebaseUser != **null**;  Log.*e*(**"Register,hasUser"**, **"changed Auth "** + hasUser);  **if** (firebaseUser != **null**) {  goToLoggedInActivity();  }  } } |

### LoginActivity.java

|  |
| --- |
| **public class** LoginActivity **extends** DataAdapter **implements** View.OnClickListener {  **private** TextInputLayout **tip\_email\_user\_login\_activity**, **tip\_password\_user\_login\_activity**;  **private** Button **btn\_login**, **btn\_signup\_login\_activity**;  **private** FirebaseAuth **mAuth**;  **private** Context **mContext**;    @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_login***); *// initDisplay();* initControl();  initData();  initEvent();  }    @Override  **public void** onBackPressed() {  finish();  }   **private void** loginAccount(String email, String password) {  **mAuth**.signInWithEmailAndPassword(email, password)  .addOnCompleteListener(**new** OnCompleteListener<AuthResult>() {  @Override  **public void** onComplete(@NonNull Task<AuthResult> task) {   **if** (task.isSuccessful()) {  DatabaseReference databaseReference = FirebaseDatabase.*getInstance*().getReference();  String Token = FirebaseInstanceId.*getInstance*().getToken();  String currentUserId = **mAuth**.getCurrentUser().getUid();  databaseReference.child(**"Users"**).child(currentUserId).child(**"mToken"**).setValue(Token).addOnSuccessListener(**new** OnSuccessListener<Void>() {  @Override  **public void** onSuccess(Void aVoid) {  goToLoggedInActivity();  **progressDialog**.dismiss();  }  });    } **else** {  *// giao diện thất bại* **progressDialog**.dismiss();   **tip\_email\_user\_login\_activity**.setError( **"Error: "** + task.getException().getMessage());  **tip\_password\_user\_login\_activity**.setError( **"Error: "** + task.getException().getMessage());   }  }  });  }   **private void** initEvent() {  }   **private void** initData() {  **mAuth** = FirebaseAuth.*getInstance*();  **mContext** = getApplicationContext();  }   **private void** initControl() {  **tip\_email\_user\_login\_activity** = findViewById(R.id.***tip\_email\_user\_login\_activity***);  **tip\_email\_user\_login\_activity**.setErrorEnabled(**true**);  **tip\_password\_user\_login\_activity** = findViewById(R.id.***tip\_password\_user\_login\_activity***);  **tip\_password\_user\_login\_activity**.setErrorEnabled(**true**);  **btn\_login** = findViewById(R.id.***btn\_login\_activity***);  **btn\_login**.setOnClickListener(**this**);  **btn\_signup\_login\_activity** = findViewById(R.id.***btn\_signup\_login\_activity***);  **btn\_signup\_login\_activity**.setOnClickListener(**this**);  }   **private void** initDisplay() {  hideNotificationBar();  }   **private void** goToLoggedInActivity() {  Intent intent = **new** Intent(**mContext**, LoggedInActivity.**class**);  startActivity(intent);  finishAffinity();  }    @Override  **public void** onClick(View v) {  **switch** (v.getId()) {  **case** R.id.***btn\_login\_activity***:  waitProgress(**"Loging..."**);  getDataEdt();  **break**;  **case** R.id.***btn\_signup\_login\_activity***:  Intent intent = **new** Intent(LoginActivity.**this**, RegisterActivity.**class**);  startActivity(intent);  **break**;  **default**:  **break**;  }  }   **private void** getDataEdt() {  String email = **tip\_email\_user\_login\_activity**.getEditText().getText().toString();  String password = **tip\_password\_user\_login\_activity**.getEditText().getText().toString();  hideKeyBoard(); *// if (matcher.matches()) {* **if** (email.equals(**""**)) {  **tip\_email\_user\_login\_activity**.setError(**"Please enter the email"**);  **progressDialog**.dismiss();  } **else** {  **tip\_email\_user\_login\_activity**.setError(**""**);  }   **if** (password.equals(**""**)) {  **tip\_password\_user\_login\_activity**.setError(**"Please enter the password"**);  **progressDialog**.dismiss();  } **else** {  **tip\_password\_user\_login\_activity**.setError(**""**);  }  **if** (!email.isEmpty() && !password.isEmpty()){  loginAccount(email, password);   }    }  *// private boolean validatePassword(String password) { // if (password.equals("")){ // return true; // }else { // return false; // } // // } // // private boolean validateEmail(String email) { //// String emailParttern = "(?:[a-z0-9!#$%&'\*+/=?^\_`{|}~-]+(?:\\.[a-z0-9!#$%&'\*+/=?^\_`{|}~-]+)\*|\"(?:[\\x01-\\x08\\x0b\\x0c\\x0e-\\x1f\\x21\\x23-\\x5b\\x5d-\\x7f]|\\\\[\\x01-\\x09\\x0b\\x0c\\x0e-\\x7f])\*\")@(?:(?:[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?\\.)" //// + "+[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?|\\[(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?|[a-z0-9-]\*[a-z0-9]:(?:[\\x01-\\x08\\x0b\\x0c\\x0e-\\x1f\\x21-\\x5a\\x53-\\x7f]|\\\\[\\x01-\\x09\\x0b\\x0c\\x0e-\\x7f])+)\\])\n"; //// Pattern pattern = Pattern.compile(emailParttern); //// Matcher matcher = pattern.matches(t); // // return false; // }* **private boolean** isValidUrl(String url) {  Pattern p = Patterns.***WEB\_URL***;  Matcher m = p.matcher(url);  **if** (m.matches())  **return true**;  **else  return false**;  } } |

### LoggedlnActivity.java

|  |
| --- |
| **public class** LoggedInActivity **extends** AppCompatActivity **implements** View.OnClickListener {  **private** ViewPager **viewPager**;  **private** ImageView **ivRecent**, **ivFindFriend**, **ivContact**, **ivProfile**;  **private** RelativeLayout **sectionRecent**, **sectionFindFriend**, **sectionContact**, **sectionProfile**;  **private** FirebaseAuth **mAuth**;  **private boolean doubleBackpressedtoExit** = **false**;  **private final static int *TIMEBACKPRESSED*** = 2000;  **private int countBack** = 0;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***loggedin\_activity***);  initControls();  initData();  initDisplay();  initEvents();  }   **private void** initData() {  **mAuth** = FirebaseAuth.*getInstance*();  Setup.*currentUser* = **new** User();  **final** String uid = FirebaseAuth.*getInstance*().getCurrentUser().getUid();  Setup.*currentUser*.setmId(uid);  updateOnline(**true**);  **final** DatabaseReference databaseReference = FirebaseDatabase.*getInstance*().getReference();  databaseReference.child(**"Users"**).child(uid).addValueEventListener(**new** ValueEventListener() {  @Override  **public void** onDataChange(DataSnapshot dataSnapshot) {  Setup.*currentUser* = dataSnapshot.getValue(User.**class**);   }   @Override  **public void** onCancelled(DatabaseError databaseError) {  Toast.*makeText*(LoggedInActivity.**this**, databaseError.getMessage(), Toast.***LENGTH\_SHORT***).show();  }  });  }   @Override  **protected void** onResume() {  **super**.onResume();  updateOnline(**true**);  }   @Override  **protected void** onDestroy() {  **super**.onDestroy();  updateOnline(**false**);  Setup.*currentUser* = **new** User();  }   @Override  **public void** onBackPressed() {  **countBack**++;  **if** (**doubleBackpressedtoExit** == **true**) {  **super**.onBackPressed();  finishAffinity();  **return**;  }**if** (**countBack** == 2 ){  AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**)  .setTitle(**"Log-Out"**)  .setMessage(**"Do you want Log-out?"**)  .setCancelable(**true**)  .setPositiveButton(**"Yes"**, **new** DialogInterface.OnClickListener() {  @Override  **public void** onClick(DialogInterface dialogInterface, **int** i) {  **mAuth**.*getInstance*().signOut();  Intent intent = **new** Intent(LoggedInActivity.**this**, StartActivity.**class**);  startActivity(intent);  finish();  }  })  .setNegativeButton(**"No"**, **null**);  AlertDialog alertDialog = builder.create();  alertDialog.show();  alertDialog.getButton(alertDialog.***BUTTON\_POSITIVE***).setTextColor(getResources().getColor(R.color.***colorDone***));  alertDialog.getButton(alertDialog.***BUTTON\_NEGATIVE***).setTextColor(getResources().getColor(R.color.***colorCancel***));  }   **new** Handler().postDelayed(**new** Runnable() {  @Override  **public void** run() {  **doubleBackpressedtoExit** = **false**;  **countBack** = 0;  }  }, ***TIMEBACKPRESSED***);  }      **private void** initEvents() {  **viewPager**.addOnPageChangeListener(**new** ViewPager.OnPageChangeListener() {  @Override  **public void** onPageScrolled(**int** position, **float** positionOffset, **int** positionOffsetPixels) {   }   @Override  **public void** onPageSelected(**int** position) {  **ivRecent**.setImageResource(R.drawable.***recent\_normal***);  **ivFindFriend**.setImageResource(R.drawable.***find\_friend\_normal***);  **ivContact**.setImageResource(R.drawable.***contact\_normal***);  **ivProfile**.setImageResource(R.drawable.***profile\_normal***);   **switch** (position) {  **case** 0:  **ivRecent**.setImageResource(R.drawable.***recent\_active2***);  **break**;  **case** 1:  **ivFindFriend**.setImageResource(R.drawable.***find\_friend\_active2***);  **break**;  **case** 2:  **ivContact**.setImageResource(R.drawable.***contact\_active2***);  **break**;  **case** 3:  **ivProfile**.setImageResource(R.drawable.***profile\_active2***);  **break**;  }  }   @Override  **public void** onPageScrollStateChanged(**int** state) {   }  });  **sectionRecent**.setOnClickListener(**this**);  **sectionFindFriend**.setOnClickListener(**this**);  **sectionContact**.setOnClickListener(**this**);  **sectionProfile**.setOnClickListener(**this**);  }   **private void** initDisplay() {  initViewPager();  initTopBar();  }   **private void** initTopBar() {  **ivRecent**.setImageResource(R.drawable.***recent\_active2***);  }   **private void** initViewPager() {  ViewPagerAdapter viewPagerAdapter = **new** ViewPagerAdapter(getSupportFragmentManager());  **viewPager**.setAdapter(viewPagerAdapter);  **viewPager**.setOffscreenPageLimit(4);  }   **private void** initControls() {  **viewPager** = findViewById(R.id.***vp\_loggedin\_activity***);  **ivRecent** = findViewById(R.id.***iv\_recent\_loggedin\_activity***);  **ivFindFriend** = findViewById(R.id.***iv\_findfriend\_loggedin\_activity***);  **ivContact** = findViewById(R.id.***iv\_contact\_loggedin\_activity***);  **ivProfile** = findViewById(R.id.***iv\_profile\_loggedin\_activity***);  **sectionRecent** = findViewById(R.id.***section\_recent\_loggedin\_activity***);  **sectionFindFriend** = findViewById(R.id.***section\_findfriend\_loggedin\_activity***);  **sectionContact** = findViewById(R.id.***section\_contact\_loggedin\_activity***);  **sectionProfile** = findViewById(R.id.***section\_profile\_loggedin\_activity***);  }    @Override  **public void** onClick(View view) {  **switch** (view.getId()) {  **case** R.id.***section\_recent\_loggedin\_activity***:  **viewPager**.setCurrentItem(0, **true**);  **break**;  **case** R.id.***section\_findfriend\_loggedin\_activity***:  **viewPager**.setCurrentItem(1, **true**);  **break**;  **case** R.id.***section\_contact\_loggedin\_activity***:  **viewPager**.setCurrentItem(2, **true**);  **break**;  **case** R.id.***section\_profile\_loggedin\_activity***:  **viewPager**.setCurrentItem(3, **true**);  **break**;  }  }   **private void** updateOnline(**boolean** isOnline) {  DatabaseReference databaseReference = FirebaseDatabase.*getInstance*().getReference().child(**"Users"**).child(Setup.*currentUser*.getmId()).child(**"mIsOnline"**);  databaseReference.onDisconnect().setValue(**false**);  databaseReference.setValue(isOnline)  .addOnCompleteListener(**new** OnCompleteListener<Void>() {  @Override  **public void** onComplete(@NonNull Task<Void> task) {  **if** (!task.isSuccessful()) {  Toast.*makeText*(LoggedInActivity.**this**, **"lỗi: "** + task.getException().getMessage(), Toast.***LENGTH\_SHORT***).show();  }   }  });  }   } |

### ChatActivity.java

|  |
| --- |
| **public class** ChatActivity **extends** AppCompatActivity {  RecyclerView **recyclerView**;  **private** String **idRoom**, **friendName**, **friendId**;  EditText **edtMessage**;  ImageView **iv\_send\_chat\_activity**, **iv\_send\_img\_chat\_activity**;  ProgressBar **progressBar**;  **private** Toolbar **toolbar**;  ArrayList<Message> **messageArrayList** = **new** ArrayList<>();  ArrayList<Message> **temporaryMessageList** = **new** ArrayList<>();  ChatAdapterRecyclerAdapter **adapterRecyclerAdapter**;  DatabaseReference **messageReference**;  DatabaseReference **recentReference**;  DatabaseReference **contactReference**;  LinearLayoutManager **linearLayoutManager**;  **private static final int *GALLERY\_PICK*** = 1;  **long timeOfMessageFirstIndex**;  **boolean loadMorePermission** = **true**;  **boolean isLastMessageLoaded**=**false**;    **private** FirebaseUser **firebaseUser**;  StorageReference **storageImg**;  DatabaseReference **databaseReference**;  String **content**;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_chat***);  initData();  initControl();  initDisplay();   **iv\_send\_chat\_activity**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View view) {   **final** String content = **edtMessage**.getText().toString().trim();  **if** (!content.equals(**""**)) {   **contactReference**.child(**friendId**).child(Setup.*currentUser*.getmId()).addListenerForSingleValueEvent(**new** ValueEventListener() {  @Override  **public void** onDataChange(DataSnapshot dataSnapshot) {  **if** (dataSnapshot.exists()) { *// xem có tồn tại liên hệ với người đang chat ko, nếu có thì cho phép gửi tin nhắn* pushMessage(content);  } **else** { *//nếu ko báo lỗi bị unfriend hoặc block* **final** Snackbar mySnackbar = Snackbar.*make*(findViewById(R.id.***contentActivityChat***),  **"You can't send message because two of you are not friends"**, Snackbar.***LENGTH\_INDEFINITE***);  mySnackbar.setAction(**"Close"**, **new** View.OnClickListener() {  @Override  **public void** onClick(View view) {  mySnackbar.dismiss();  }  }).setActionTextColor(Color.***RED***);    mySnackbar.show();  }  }   @Override  **public void** onCancelled(DatabaseError databaseError) {   }   });   }   }  });    **edtMessage**.setOnFocusChangeListener(**new** View.OnFocusChangeListener() {  @Override  **public void** onFocusChange(View v, **boolean** hasFocus) {  **if** (hasFocus) {  **recyclerView**.postDelayed(**new** Runnable() {  @Override  **public void** run() {  **recyclerView**.scrollToPosition(**messageArrayList**.size() - 1);  }  }, 500);  }  }  });  **recyclerView**.addOnScrollListener(**new** RecyclerView.OnScrollListener() {  @Override  **public void** onScrollStateChanged(RecyclerView recyclerView, **int** newState) {  **super**.onScrollStateChanged(recyclerView, newState);  **int** firstItem = **linearLayoutManager**.findFirstCompletelyVisibleItemPosition();  Log.*e*(**"permission"**,**loadMorePermission**+**" "**+**isLastMessageLoaded**);  **if** (firstItem == 0 && **loadMorePermission** == **true** && **isLastMessageLoaded**==**false**) {  **loadMorePermission** = **false**; *//ko cho loadmore nếu chưa loadmore xong* loadMoreMessage();  }  }   });    }   @Override  **protected void** onStart() {  **super**.onStart();  updateSeenRecent(**true**);  }   @Override  **protected void** onStop() {  **super**.onStop();  updateSeenRecent(**true**);  updateContactStatus();  }   **private void** initData() {  Intent intent = getIntent();  **idRoom** = intent.getStringExtra(**"idRoom"**);  **friendName** = intent.getStringExtra(**"friendName"**);  **friendId** = intent.getStringExtra(**"friendId"**);  **messageReference** = FirebaseDatabase.*getInstance*().getReference().child(**"Messages"**).child(**idRoom**);  **recentReference** = FirebaseDatabase.*getInstance*().getReference().child(**"Recents"**).child(**idRoom**);  **contactReference** = FirebaseDatabase.*getInstance*().getReference().child(**"Contacts"**);  }   **private void** initDisplay() {  customToolbar();  initRecyclerView();  updateMessageRecycler();  sendImg();  checkEdt();  }   **private void** checkEdt(){  **edtMessage**.addTextChangedListener(**new** TextWatcher() {  @Override  **public void** beforeTextChanged(CharSequence s, **int** start, **int** count, **int** after) {   }   @Override  **public void** onTextChanged(CharSequence s, **int** start, **int** before, **int** count) {  **if** (**edtMessage**.getText().length() > 0){  **iv\_send\_chat\_activity**.setImageResource(R.drawable.***ic\_send\_blue***);  Log.*i*(**"Kiem tra edt"**, **"checkEdt: co noi dung"**);  }**else** {  Log.*i*(**"Kiem tra edt"**, **"checkEdt: khong co noi dung"**);  **iv\_send\_chat\_activity**.setImageResource(R.drawable.***ic\_send***);  }   }   @Override  **public void** afterTextChanged(Editable s) {   }  });   }    **private void** initRecyclerView() {  **adapterRecyclerAdapter** = **new** ChatAdapterRecyclerAdapter(getApplicationContext(), **messageArrayList**);  **linearLayoutManager** = **new** LinearLayoutManager(getApplicationContext());  **linearLayoutManager**.setOrientation(LinearLayoutManager.***VERTICAL***);  **recyclerView**.setItemAnimator(**new** DefaultItemAnimator());  **recyclerView**.setLayoutManager(**linearLayoutManager**);  **recyclerView**.setAdapter(**adapterRecyclerAdapter**);  **recyclerView**.startLayoutAnimation();  }   **private void** initControl() {  **toolbar** = findViewById(R.id.***actionbar\_chat\_activity***);  setSupportActionBar(**toolbar**);  **recyclerView** = findViewById(R.id.***recyclerChat***);  **edtMessage** = findViewById(R.id.***edt\_message\_chat\_activity***);  **iv\_send\_chat\_activity** = findViewById(R.id.***iv\_send\_chat\_activity***);  **iv\_send\_img\_chat\_activity** = findViewById(R.id.***iv\_send\_img\_chat\_activity***);  **progressBar** = findViewById(R.id.***progress\_chat\_activity***);  }   @Override  **public boolean** onSupportNavigateUp() {  finish();  **return true**;  }   **private void** customToolbar() {  Drawable arrow = getApplicationContext().getResources().getDrawable(R.drawable.***ic\_back***);  **toolbar**.setTitleTextColor(Color.***WHITE***);  getSupportActionBar().setTitle(**friendName**);  getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  getSupportActionBar().setHomeAsUpIndicator(arrow);   }   **private void** sendImg() {  **iv\_send\_img\_chat\_activity**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  Intent openGallery = **new** Intent(Intent.***ACTION\_GET\_CONTENT***);  openGallery.setType(**"image/\*"**);  startActivityForResult(Intent.*createChooser*(openGallery, **"select img"**), ***GALLERY\_PICK***);   }  });  }   @Override  **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  **super**.onActivityResult(requestCode, resultCode, data);   Uri uri = data.getData();  String currentID = **firebaseUser**.getUid();  **if** (requestCode == ***GALLERY\_PICK*** && resultCode == ***RESULT\_OK***) {  Toast.*makeText*(**this**, **"path: "** + uri, Toast.***LENGTH\_SHORT***).show();  StorageReference path = **storageImg**.child(**"SendIMG"**).child(currentID + **".bmp"**);  path.putFile(uri).addOnCompleteListener(**new** OnCompleteListener<UploadTask.TaskSnapshot>() {  @Override  **public void** onComplete(@NonNull Task<UploadTask.TaskSnapshot> task) {  **if** (task.isSuccessful()) {  String upLoadIMG = task.getResult().getDownloadUrl().toString();  **databaseReference**.child(**"mContent"**).setValue(upLoadIMG).addOnCompleteListener(**new** OnCompleteListener<Void>() {  @Override  **public void** onComplete(@NonNull Task<Void> task) {  **if** (task.isSuccessful()) {  Toast.*makeText*(ChatActivity.**this**, **"success"**, Toast.***LENGTH\_SHORT***).show();   }   }  });  }  }  });  }  }    **private void** updateMessageRecycler() {   Query query = **messageReference**.orderByChild(**"mTime"**).limitToLast(15);  query.addChildEventListener(**new** ChildEventListener() {  @Override  **public void** onChildAdded(DataSnapshot dataSnapshot, String s) {  Message message = dataSnapshot.getValue(Message.**class**);  **messageArrayList**.add(message);  **adapterRecyclerAdapter**.notifyDataSetChanged();  **recyclerView**.scrollToPosition(**messageArrayList**.size() - 1);  **if** (**messageArrayList**.size() > 0) {  **timeOfMessageFirstIndex** = **messageArrayList**.get(0).getmTime();  }   }   @Override  **public void** onChildChanged(DataSnapshot dataSnapshot, String s) {   }   @Override  **public void** onChildRemoved(DataSnapshot dataSnapshot) {   }   @Override  **public void** onChildMoved(DataSnapshot dataSnapshot, String s) {   }   @Override  **public void** onCancelled(DatabaseError databaseError) {   }  });   }  **private void** loadMoreMessage() {  **progressBar**.setVisibility(View.***VISIBLE***);  Query query = **messageReference**.orderByChild(**"mTime"**).endAt(**timeOfMessageFirstIndex** - 1).limitToLast(15);  query.addListenerForSingleValueEvent(**new** ValueEventListener() {  @Override  **public void** onDataChange(DataSnapshot dataSnapshot) {  **for** (DataSnapshot messageSnapshot : dataSnapshot.getChildren()) {  Message message = messageSnapshot.getValue(Message.**class**);  **temporaryMessageList**.add(message);  }  **final int** indexNeedToScrollWhenLoaded=**temporaryMessageList**.size()+(**int**)(**temporaryMessageList**.size()/2);  Log.*e*(**"index"**,indexNeedToScrollWhenLoaded+**""**);  **if** (**temporaryMessageList**.size() > 0) {  **timeOfMessageFirstIndex** = **temporaryMessageList**.get(0).getmTime();  **messageArrayList**.addAll(0, **temporaryMessageList**);  **temporaryMessageList**.clear();  }**else** {  **isLastMessageLoaded**=**true**;  }   Handler handler = **new** Handler(); *//server load nhanh quá delay lại 1s* handler.postDelayed(**new** Runnable() {  @Override  **public void** run() {  **adapterRecyclerAdapter**.notifyDataSetChanged();  **recyclerView**.scrollToPosition(indexNeedToScrollWhenLoaded);  **loadMorePermission** = **true**; *// load xong cho load tiếp* **progressBar**.setVisibility(View.***GONE***);  }  },1000);   }   @Override  **public void** onCancelled(DatabaseError databaseError) {  }  });  }       **private void** pushMessage(String content) {  String key = **messageReference**.push().getKey();  **final** HashMap<String, Object> messageMap = **new** HashMap<>();  messageMap.put(**"mContent"**, content);  messageMap.put(**"mTime"**, ServerValue.***TIMESTAMP***);  messageMap.put(**"mIdSend"**, Setup.*currentUser*.getmId());    **messageReference**.child(key).setValue(messageMap).addOnCompleteListener(**new** OnCompleteListener<Void>() {  @Override  **public void** onComplete(@NonNull Task<Void> task) {  **if** (task.isSuccessful()) {  **edtMessage**.setText(**""**);  updateLastMessage(messageMap);  **recyclerView**.scrollToPosition(**messageArrayList**.size() - 1);  } **else** {  Toast.*makeText*(getApplicationContext(), **"Lỗi: "** + task.getException().getMessage(), Toast.***LENGTH\_SHORT***).show();  }  }  });  }   **private void** updateLastMessage(HashMap<String, Object> recentMap) {  String[] id = Utility.*generateIdRoomBasedOnIdUsers*(**friendId**, Setup.*currentUser*.getmId());  recentMap.put(id[1], **true**);  recentMap.put(id[2], **true**);  recentMap.put(**"mID1"**, id[1]);  recentMap.put(**"mID2"**, id[2]);  recentMap.put(**"mSeen"**, **false**);  **recentReference**.setValue(recentMap);  }   **private void** updateSeenRecent(**final boolean** isSeen) {  **final** DatabaseReference recentReference = FirebaseDatabase.*getInstance*().getReference().child(**"Recents"**).child(**idRoom**);  recentReference.addListenerForSingleValueEvent(**new** ValueEventListener() {  @Override  **public void** onDataChange(DataSnapshot dataSnapshot) {  **if** (dataSnapshot.exists()) {  String idSend = dataSnapshot.child(**"mIdSend"**).getValue().toString();  **if** (!idSend.equals(Setup.*currentUser*.getmId())) {  recentReference.child(**"mSeen"**).setValue(isSeen);  }  }  }   @Override  **public void** onCancelled(DatabaseError databaseError) {   }  });  }   **private void** updateContactStatus() {  **contactReference**.child(**friendId**).child(Setup.*currentUser*.getmId()).addListenerForSingleValueEvent(**new** ValueEventListener() {  @Override  **public void** onDataChange(DataSnapshot dataSnapshot) {  **if** (!dataSnapshot.exists()) {  **recentReference**.removeValue();  }  }   @Override  **public void** onCancelled(DatabaseError databaseError) {   }  });  }   } |

### AbouUsActivity.java

|  |
| --- |
| **public class** AbouUsActivity **extends** DataAdapter {   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_abou\_us***);  ImageView icBack = findViewById(R.id.***ic\_back\_aboutus\_activity***);  icBack.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  AbouUsActivity.**this**.finish();  }  });  } } |

### ViewProfileFriends.java

|  |
| --- |
| **public class** ViewProfileFriends **extends** AppCompatActivity {  **private** TextView **tv\_name\_view\_friend\_activity**, **tv\_full\_name\_view\_friend\_activity**, **tv\_gender\_view\_friend\_activity**, **tv\_birth\_view\_friend\_activity**,**tv\_occupation\_view\_friend\_activity**,**tv\_provine\_view\_friend\_activity**;  **private** ImageView **iv\_user\_view\_friend\_activity**;  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_view\_profile\_friends***);  initControl();  **tv\_name\_view\_friend\_activity**.setText(Setup.*selectUser*.getmName());  **tv\_birth\_view\_friend\_activity**.setText(Setup.*selectUser*.getmBirth());  **tv\_occupation\_view\_friend\_activity**.setText(Setup.*selectUser*.getmOccupation());  **tv\_full\_name\_view\_friend\_activity**.setText(Setup.*selectUser*.getmFullName());  **tv\_gender\_view\_friend\_activity**.setText(Setup.*selectUser*.getmGender());  **tv\_provine\_view\_friend\_activity**.setText(Setup.*selectUser*.getmProvince());  Picasso.*with*(**this**).load(Setup.*selectUser*.getmLinkImage()).resize(120,120).into(**iv\_user\_view\_friend\_activity**);   }   **private void** initControl() {  **iv\_user\_view\_friend\_activity** = findViewById(R.id.***iv\_user\_view\_friend\_activity***);  **tv\_name\_view\_friend\_activity** = findViewById(R.id.***tv\_name\_view\_friend\_activity***);  **tv\_provine\_view\_friend\_activity** = findViewById(R.id.***tv\_provine\_view\_friend\_activity***);  **tv\_full\_name\_view\_friend\_activity** = findViewById(R.id.***tv\_full\_name\_view\_friend\_activity***);  **tv\_gender\_view\_friend\_activity** = findViewById(R.id.***tv\_gender\_view\_friend\_activity***);  **tv\_occupation\_view\_friend\_activity** = findViewById(R.id.***tv\_occupation\_view\_friend\_activity***);  **tv\_birth\_view\_friend\_activity** = findViewById(R.id.***tv\_birth\_view\_friend\_activity***);   } } |

### ChatAdapterRecyclerAdapter.java

### Activity\_splash\_screen.xml

|  |
| --- |
| <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="@drawable/my\_bg\_anim"  android:gravity="center"  android:fitsSystemWindows="true"  android:id="@+id/myLayout"   tools:context="com.hdh.appchatgr2.Activities.SplashScreen"**>    <**ImageView   android:id="@+id/iv\_logo"  android:layout\_width="100dp"  android:layout\_height="100dp"  android:layout\_gravity="center"  android:src="@drawable/logov3"** />  </**LinearLayout**> |

### Activity\_login.xml

|  |
| --- |
| <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"  android:theme="@style/themeLoginRegisterActivity"  android:background="@drawable/bg\_login\_activity"  android:gravity="center\_horizontal"  android:fitsSystemWindows="true"  tools:context=".Activities.LoginActivity"**>   <**LinearLayout   android:orientation="horizontal"  android:layout\_width="match\_parent"  android:layout\_height="100dp"**>   <**ImageView  android:layout\_marginLeft="50dp"  android:layout\_gravity="center"  android:src="@drawable/logov3"  android:layout\_width="50dp"  android:layout\_height="50dp"** />   <**TextView  android:gravity="center"  android:layout\_width="160dp"  android:layout\_height="wrap\_content"  android:layout\_gravity="center"  android:fontFamily="@font/acme"  android:text="Log In"  android:textColor="#fff"  android:textSize="30sp"** />  </**LinearLayout**>     <**android.support.design.widget.TextInputLayout  android:textColorHint="#fff"  android:id="@+id/tip\_email\_user\_login\_activity"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"**>   <**EditText  android:maxLength="20"  android:textColor="#b6ffffff"  android:textColorHint="#b6ffffff"  android:backgroundTint="#fff"  android:ems="13"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:hint="Email"**/>   </**android.support.design.widget.TextInputLayout**>   <**android.support.design.widget.TextInputLayout  android:textColorHint="#fff"  android:id="@+id/tip\_password\_user\_login\_activity"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"**>   <**EditText  android:maxLength="20"  android:textColor="#b6ffffff"  android:textColorHint="#b6ffffff"  android:backgroundTint="#fff"  android:ems="13"  android:id="@+id/edt\_password\_user\_register\_activity"  android:layout\_width="wrap\_content"  android:inputType="textPassword"  android:layout\_height="wrap\_content"  android:hint="Password"** />  </**android.support.design.widget.TextInputLayout**>   <**Button  android:id="@+id/btn\_login\_activity"   android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginTop="30dp"  android:background="@drawable/drawble\_btn\_logout"  style="?android:attr/borderlessButtonStyle"  android:ems="11"  android:textSize="18sp"  android:fontFamily="@font/acme"  android:text="Log In"  android:textAllCaps="false"  android:textColor="#49beff"** />    <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginTop="90dp"  android:fontFamily="@font/arimo"  android:text="Not a member yet?"  android:textColor="#fff"** />   <**Button  style="?android:attr/borderlessButtonStyle"  android:ems="11"  android:background="@drawable/btn\_border\_white"  android:layout\_marginTop="10dp"  android:id="@+id/btn\_signup\_login\_activity"  android:text="Sign Up"  android:textAllCaps="false"  android:textColor="#fff"  android:textSize="18sp"  android:fontFamily="@font/acme"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"** />       </**LinearLayout**> |

### Activity\_Chat.xml

|  |
| --- |
| <**android.support.design.widget.CoordinatorLayout  xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="#fff"  android:id="@+id/contentActivityChat"  xmlns:app="http://schemas.android.com/apk/res-auto"**> <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="#fff"  xmlns:app="http://schemas.android.com/apk/res-auto"  android:orientation="vertical"  tools:context="com.hdh.appchatgr2.Activities.ChatActivity"**>   <**android.support.v7.widget.Toolbar  android:id="@+id/actionbar\_chat\_activity"  android:layout\_width="match\_parent"  android:layout\_height="?android:actionBarSize"  android:background="@color/colorAccent"  android:elevation="5dp"**>   </**android.support.v7.widget.Toolbar**>  <**ProgressBar  android:layout\_gravity="center"  android:id="@+id/progress\_chat\_activity"  android:visibility="gone"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"** />  <**android.support.v7.widget.RecyclerView  android:paddingTop="10dp"  android:id="@+id/recyclerChat"  android:overScrollMode="never"    android:layout\_weight="1"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>   </**android.support.v7.widget.RecyclerView**>    <**View  android:background="#f2f2f2"  android:layout\_width="match\_parent"  android:layout\_height="1dp"**/>  <**LinearLayout  android:orientation="horizontal"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"**>   <**ImageView  android:id="@+id/iv\_send\_img\_chat\_activity"  android:layout\_gravity="center"  android:src="@drawable/ic\_sendimg"  android:layout\_width="40dp"  android:layout\_height="40dp"** />  <**EditText  android:layout\_marginLeft="5dp"  android:hint="Enter message..."  android:id="@+id/edt\_message\_chat\_activity"  android:layout\_weight="1"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"** />  <**ImageView  android:padding="5dp"  android:layout\_margin="5dp"  android:src="@drawable/ic\_send"  android:id="@+id/iv\_send\_chat\_activity"  android:layout\_width="40dp"  android:layout\_height="40dp"** />  </**LinearLayout**>    </**LinearLayout**> </**android.support.design.widget.CoordinatorLayout**> |

### Activity\_abou\_us.xml

|  |
| --- |
| <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"  android:id="@+id/layout\_about\_us"  tools:context="com.hdh.appchatgr2.Activities.AbouUsActivity"**>    <**RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="200dp"  android:background="@color/colorAccent"  android:orientation="vertical"**>   <**ImageView  android:id="@+id/ic\_back\_aboutus\_activity"  android:layout\_marginTop="20dp"  android:layout\_marginLeft="10dp"  android:src="@drawable/ic\_back"  android:layout\_width="30dp"  android:layout\_height="wrap\_content"** />   <**ImageView  android:id="@+id/logoa"  android:layout\_below="@id/ic\_back\_aboutus\_activity"  android:layout\_width="50dp"  android:layout\_height="50dp"  android:layout\_marginTop="-20dp"  android:layout\_marginLeft="50dp"  android:src="@drawable/logov3"** />   <**TextView  android:layout\_below="@id/logoa"  android:id="@+id/nameapp"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="50dp"  android:layout\_marginTop="10dp"  android:fontFamily="@font/convergence"  android:text="AppChatG2"  android:textColor="#fff"  android:textSize="25sp"** />   <**TextView  android:layout\_below="@id/nameapp"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="50dp"  android:layout\_marginTop="10dp"  android:fontFamily="@font/convergence"  android:text="Ho Chi Minh City"  android:textColor="#f0f0f0"  android:textSize="17sp"** />  </**RelativeLayout**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="100dp"  android:orientation="horizontal"**>   <**ImageView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_gravity="center\_vertical"  android:layout\_marginLeft="15dp"  android:src="@drawable/ic\_phone\_aboutus"** />   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:gravity="bottom"  android:orientation="vertical"  android:paddingLeft="20dp"**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="097 888 999 9"  android:textColor="#000"  android:textSize="18sp"  android:textStyle="bold"** />   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="10dp"  android:text="Tran Vinh Quang (Mr.)"  android:textColor="#d6d6d6"** />   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="15dp"  android:text="Work"  android:textColor="#e6e5e5"** />   <**View  android:layout\_width="match\_parent"  android:layout\_height="2dp"  android:background="#f2f2f2"** />   </**LinearLayout**>   </**LinearLayout**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="100dp"  android:orientation="horizontal"**>   <**ImageView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_gravity="center\_vertical"  android:layout\_marginLeft="15dp"  android:src="@drawable/ic\_email\_aboutus"** />   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:gravity="bottom"  android:orientation="vertical"  android:paddingLeft="20dp"**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="AppChatG2@gmail.com"  android:textColor="#000"  android:textSize="18sp"  android:textStyle="bold"** />   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="10dp"  android:text="Tran Vinh Quang (Mr.)"  android:textColor="#d6d6d6"** />   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="15dp"  android:text="Work"  android:textColor="#e6e5e5"** />   <**View  android:layout\_width="match\_parent"  android:layout\_height="2dp"  android:background="#f2f2f2"** />   </**LinearLayout**>   </**LinearLayout**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="140dp"  android:orientation="horizontal"**>   <**ImageView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_gravity="center\_vertical"  android:layout\_marginLeft="15dp"  android:src="@drawable/ic\_editapp"** />   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:gravity="bottom"  android:orientation="vertical"  android:paddingLeft="20dp"**>     <**TextView  android:lineSpacingMultiplier="6"  android:lineSpacingExtra="2.5dp"  android:textStyle="bold"  android:textSize="18sp"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Team 2"  android:textColor="#000"** />   <**TextView  android:lineSpacingMultiplier="6"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Tran Vinh Quang (Leader) (Mr.)"  android:textColor="#414141"** />   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Ly Hai Nam (Mr.)"  android:textColor="#414141"** />   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Nguyen Tran Phi Long (Mr.)"  android:textColor="#414141"** />   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Hong Quang Minh (Mr.)"  android:textColor="#414141"** />   <**TextView  android:layout\_marginBottom="10dp"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Ho Dac Hien (Mr.)"  android:textColor="#414141"** />   <**View  android:layout\_width="match\_parent"  android:layout\_height="2dp"  android:background="#f2f2f2"** />   </**LinearLayout**>   </**LinearLayout**>   <**LinearLayout  android:layout\_marginTop="5dp"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal"**>   <**TextView  android:textColor="@color/colorAccent"  android:text="Version"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_gravity="center\_vertical"  android:layout\_marginLeft="10dp"** />   <**TextView  android:textColor="#000"  android:layout\_marginLeft="10dp"  android:text="v1.0"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"** />     </**LinearLayout**>    </**LinearLayout**> |

### Activity\_search.xml

|  |
| --- |
| <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:orientation="vertical"  android:background="#fff"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context="com.hdh.appchatgr2.Activities.SearchActivity"**>  <**EditText  android:id="@+id/edtSearch"  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:layout\_marginLeft="10dp"  android:layout\_marginRight="10dp"  android:layout\_marginTop="5dp"  android:background="@drawable/background\_search"  android:drawableLeft="@drawable/search"  android:drawablePadding="5dp"  android:hint="Input your friend's nick name here"  android:paddingLeft="10dp"  android:theme="@style/mySearchInput"** />  <**TextView  android:layout\_marginTop="10dp"  android:layout\_marginBottom="10dp"  android:id="@+id/tv\_result\_status"  android:layout\_marginLeft="10dp"  android:textStyle="bold"  android:text=""  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"** />  <**ListView  android:id="@+id/lv\_find\_search\_activity"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"**>   </**ListView**> </**LinearLayout**> |

### Activity\_register.xml

|  |
| --- |
| <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"  android:gravity="center\_horizontal"  android:background="@drawable/bg\_register\_activity"  tools:context=".Activities.RegisterActivity"**>   <**LinearLayout  android:orientation="horizontal"  android:layout\_width="match\_parent"  android:layout\_height="100dp"**>  <**ImageView  android:layout\_marginLeft="50dp"  android:layout\_gravity="center"  android:src="@drawable/logov3"  android:layout\_width="50dp"  android:layout\_height="50dp"** />  <**TextView  android:layout\_width="160dp"  android:layout\_height="wrap\_content"  android:layout\_gravity="center"  android:fontFamily="@font/acme"  android:gravity="center"  android:text="Register"  android:textColor="#fff"  android:textSize="30sp"** />  </**LinearLayout**>     <**android.support.design.widget.TextInputLayout   android:textColorHint="#fff"  android:id="@+id/tip\_name\_user\_register\_activity"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"**>   <**EditText  android:maxLength="20"  android:inputType="textCapWords"  android:textColor="#b6ffffff"  android:textColorHint="#b6ffffff"  android:backgroundTint="#fff"  android:ems="13"  android:id="@+id/edt\_name\_user\_register\_activity"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:hint="Name"** />  </**android.support.design.widget.TextInputLayout**>   <**android.support.design.widget.TextInputLayout  android:layout\_marginTop="20dp"  android:textColorHint="#fff"  android:id="@+id/tip\_email\_user\_register\_activity"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"**>   <**EditText  android:maxLength="20"  android:textColor="#b6ffffff"  android:textColorHint="#b6ffffff"  android:backgroundTint="#fff"  android:ems="13"  android:id="@+id/edt\_email\_user\_register\_activity"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:hint="Email"** />  </**android.support.design.widget.TextInputLayout**>   <**android.support.design.widget.TextInputLayout  android:layout\_marginTop="20dp"  android:textColorHint="#fff"  android:id="@+id/tip\_password\_user\_register\_activity"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"**>   <**EditText  android:maxLength="20"  android:inputType="textPassword"  android:textColor="#b6ffffff"  android:textColorHint="#b6ffffff"  android:backgroundTint="#fff"  android:ems="13"  android:id="@+id/edt\_password\_user\_register\_activity"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:hint="Password"** />  </**android.support.design.widget.TextInputLayout**>     <**Button  android:layout\_marginTop="30dp"  android:ems="11"  android:background="@drawable/drawble\_btn\_logout"  android:text="Register"  style="?android:attr/borderlessButtonStyle"  android:textSize="18sp"  android:fontFamily="@font/acme"  android:textAllCaps="false"  android:textColor="#49beff"  android:id="@+id/btn\_create\_acc\_register\_activity"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"** />   <**TextView  android:textColor="#fff"  android:layout\_marginTop="90dp"  android:fontFamily="@font/arimo"  android:text="Already with us?"  android:id="@+id/tv\_have\_account\_register\_activity"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"** />   <**Button  style="?android:attr/borderlessButtonStyle"  android:ems="11"  android:background="@drawable/btn\_border\_white"  android:layout\_marginTop="10dp"  android:id="@+id/btn\_haveaccount\_login\_activity"  android:text="Sign In"  android:textAllCaps="false"  android:textColor="#fff"  android:textSize="18sp"  android:fontFamily="@font/acme"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"** />   </**LinearLayout**> |

### Activity\_start.xml

|  |
| --- |
| <**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="@drawable/bg\_start"  tools:context="com.hdh.appchatgr2.Activities.StartActivity"**>    <**TextView  android:id="@+id/layout99"    android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerHorizontal="true"  android:layout\_marginTop="80dp"  android:fontFamily="@font/alex\_brush"  android:text="Feather"  android:textColor="#fff"  android:textSize="50sp"** />   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_below="@+id/layout99"  android:gravity="center\_horizontal"  android:orientation="vertical"**>   <**Button  android:id="@+id/btn\_create\_account\_start\_activity"   android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginTop="170dp"  android:background="@drawable/btn\_border\_white"  android:ems="10"  android:fontFamily="@font/andika"  android:text="Sign Up"  android:textColor="#fff"  android:textSize="18sp"** />   <**Button  android:id="@+id/btn\_already\_account\_start\_activity"  android:layout\_width="wrap\_content"   android:layout\_height="wrap\_content"  android:layout\_centerHorizontal="true"  android:layout\_marginTop="30dp"  android:background="@drawable/bg\_btn\_login\_start\_activity"  android:ems="10"  android:fontFamily="@font/andika"  android:text="Log In"  android:textColor="#49beff"  android:textSize="18sp"** />   </**LinearLayout**>  </**RelativeLayout**> |

### Activity\_view\_profile\_friends.xml

|  |
| --- |
| <**android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:fab="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="#fff"  android:orientation="vertical"**>   <**RelativeLayout  android:id="@+id/relativeLayout"  android:layout\_width="match\_parent"  android:layout\_height="250dp"  android:background="@drawable/bg\_login\_activity"  android:gravity="center\_horizontal"  android:orientation="vertical"**>   *<!--<ImageView-->   <!--android:id="@+id/iv\_user\_view\_friend\_activity"-->  <!--android:layout\_width="match\_parent"-->  <!--android:layout\_height="175dp"-->  <!--android:adjustViewBounds="true"-->  <!--android:scaleType="fitXY"-->  <!--android:src="@drawable/avatar\_demo" />-->* <**RelativeLayout  android:id="@+id/lo\_avatar\_section"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerHorizontal="true"  android:layout\_marginTop="20dp"**>   <**View  android:layout\_width="130dp"  android:layout\_height="130dp"  android:layout\_centerInParent="true"  android:background="@drawable/circle\_border"**></**View**>   <**de.hdodenhof.circleimageview.CircleImageView  android:id="@+id/iv\_user\_view\_friend\_activity"  android:layout\_width="115dp"  android:layout\_height="115dp"  android:layout\_centerInParent="true"  android:src="@drawable/anonymous"  app:civ\_border\_color="#fff"  app:civ\_border\_width="2dp"** />   </**RelativeLayout**>   <**TextView  android:id="@+id/tv\_name\_view\_friend\_activity"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_below="@id/lo\_avatar\_section"  android:layout\_centerHorizontal="true"  android:layout\_marginTop="10dp"  android:fontFamily="@font/advent\_pro\_thin"  android:text=""  android:textColor="#ffffff"  android:textSize="24sp"** />   <**TextView  android:id="@+id/tv\_provine\_view\_friend\_activity"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_below="@+id/tv\_name\_view\_friend\_activity"  android:layout\_centerHorizontal="true"  android:layout\_marginTop="3dp"  android:fontFamily="@font/convergence"  android:text=""  android:textColor="#1d4b69"  android:textSize="16sp"** />   </**RelativeLayout**>   <**RelativeLayout  android:background="#fff"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_below="@id/relativeLayout"  android:orientation="horizontal"**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="45dp"  android:layout\_marginRight="45dp"  android:layout\_marginTop="20dp"  android:orientation="vertical"**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="5dp"  android:text="Full name"  android:textColor="#49beff"** />   <**TextView  android:id="@+id/tv\_full\_name\_view\_friend\_activity"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="5dp"  android:layout\_marginLeft="5dp"  android:text=""  android:textColor="#000"  android:textSize="18sp"** />   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:layout\_margin="5dp"  android:background="#e5e5e5"**></**View**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="5dp"  android:text="Gender"  android:textColor="#49beff"** />   <**TextView  android:id="@+id/tv\_gender\_view\_friend\_activity"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="5dp"  android:layout\_marginLeft="5dp"  android:text=""  android:textColor="#000"  android:textSize="18sp"** />   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:layout\_margin="5dp"  android:background="#e5e5e5"**></**View**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="15dp"  android:layout\_marginLeft="5dp"  android:text="Birth"  android:textColor="#49beff"** />   <**TextView  android:id="@+id/tv\_birth\_view\_friend\_activity"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="5dp"  android:layout\_marginLeft="5dp"  android:text=""  android:textColor="#000"  android:textSize="18sp"** />   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:layout\_margin="5dp"  android:background="#e5e5e5"**></**View**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="5dp"  android:text="Occupation"  android:textColor="#49beff"** />   <**TextView  android:id="@+id/tv\_occupation\_view\_friend\_activity"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="5dp"  android:layout\_marginLeft="5dp"  android:text=""  android:textColor="#000"  android:textSize="18sp"** />   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:layout\_margin="5dp"  android:background="#e5e5e5"**>   </**View**>  </**LinearLayout**>   </**RelativeLayout**>   </**LinearLayout**> </**android.support.design.widget.CoordinatorLayout**> |

# **Kiểm thử**

## Mục tiêu kiểm thử:

Việc kiểm thử ứng dụng được triển khai nhằm mục đích xác định, kiểm tra các thành phần giao diện, hoạt động, chức năng và các lỗi có thể xảy ra của ứng dụng, hướng tới ứng dụng hoàn thiện, hoàn chỉnh về mặt giao diện cũng như chức năng.

Song song đó, việc kiểm thử ứng dụng giúp nhóm phát triển xác nhận việc ứng dụng đã hoạt động theo mong muốn hay chưa.

## Kỹ thuật kiểm thử

### Công cụ kiểm thử

* Sử dụng Robotium để kiểm thử ứng dụng.
* Sử dụng device thật

### Các phương pháp kiểm thử và định nghĩa:

* Blackbox Testing
* Whitebox Testing
* GreyBox Testing
* Regression Testing

## Quy trình test

* Lập TestPlan
* Lập TestCase
* Tiến hành kiểm thử
* Báo cáo

## TestPlan:



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Đề mục muốn test** | **Người thực hiện** | **Bắt đầu** | **Kết thúc** | **Trạng thái** |
| 1 | Tài liệu mô tả yêu cầu dự án |  |  |  |  |
| 2 | Tài liệu mô tả chức năng ứng dụng |  |  |  |  |
| 3 | Tài liệu kế hoạch dự án |  |  |  |  |
| 4 | Tài liệu phân tích thiết kế |  |  |  |  |
| 5 | Tài liệu hướng dẫn cài đặt |  |  |  |  |
| 6 | Tài liệu hướng dẫn sử dụng |  |  |  |  |
| 7 | Giao diện người dùng |  |  |  |  |
| 8 | Chức năng Thêm Ghi Chú |  |  |  |  |
| 9 | Chức năng Cập Nhật ghi chú |  |  |  |  |
| 10 | Chức năng Xóa ghi chú |  |  |  |  |
| 11 | Chức năng Vẽ tay |  |  |  |  |
| 12 | Chức năng liên kết Google+ |  |  |  |  |
| 13 | Thông tin nhóm phát triển |  |  |  |  |
| 14 | Chức năng Thông báo (nhắc nhở) |  |  |  |  |
| 15 | Tạo CSDL |  |  |  |  |

## TestCase:

### TestNewNote:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. **Người thực hiện:** | |  | | |
| **Kiểu Test** | | ACTIVITY, Layout | | |
| **Trường hợp Test** | | 1 | | |
| **Tên trường hợp** | | TestNewNote | | |
| **Miêu tả** | | * Kiểm tra sự tồn tại của các controls trên giao diện * Kiểm tra layout hiện tại * Kiểm tra thuộc tính click của các Button | | |
| **Các thành phần tham gia** | | | | |
| 1 | edt\_title\_newnotes, edt\_content\_newnotes, | | | |
| 2 | btn\_timer\_new\_notes, btn\_bgcolor\_new\_notes, btn\_save\_new\_notes | | | |
| **Miêu tả** | | | | |
| **Giá trị nhập vào** | | | | **Kết quả mong muốn** |
|  | | | | - Các controls có tồn tại  -Nút button nhận được sự kiện onclick  - NewNotesActivity là activity hiện tại |
| **Thao tác thực hiện** | | | | |
| 1 | Gọi đối tượng Solo | | | |
| 2 | Thực hiện viết lệnh xác dịnh id cho biến | | | |
| 3 | Kiểm tra sự tồn tại của id (biến) đó | | | |
| 4 | Thiết lập sự kiện click cho các button | | | |
| **Kết quả:** | | |  | |

### 7.5.2. TestNewNotesDetail:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. **Người thực hiện:** | |  | | |
| **Kiểu Test** | | ACTIVITY, Layout | | |
| **Trường hợp Test** | | 1 | | |
| **Tên trường hợp** | | TestNewNotesDetail | | |
| **Miêu tả** | | * Kiểm tra sự tồn tại của các controls trên giao diện * Kiểm tra layout hiện tại * Kiểm tra truyền dữ liệu | | |
| **Các thành phần tham gia** | | | | |
| 1 | edt\_title\_detail, edt\_content\_detail, | | | |
| 2 | btn\_bgcolor\_detail, btn\_update\_detail | | | |
| **Miêu tả** | | | | |
| **Giá trị nhập vào** | | | | **Kết quả mong muốn** |
|  | | | | - Các controls có tồn tại  - Dữ liệu có được truyền qua intent  - NoteDetailActivity là activity hiện tại |
| **Thao tác thực hiện** | | | | |
| 1 | Gọi đối tượng solo | | | |
| 2 | Truyền dữ liệu qua intent | | | |
| 3 | Thực hiện viết lệnh xác dịnh id cho biến | | | |
| 4 | Kiểm tra sự tồn tại của id (biến) đó | | | |
| **Kết quả:** | | |  | |

# **Hướng dẫn sử dụng:**

## Cài đặt ứng dụng

### Đối với máy tính có cài đặt thiết bị ảo:

#### Thao tác với file apk (cài đặt nhanh):

#### Thao tác với thư mục gốc (source)

#### Thao tác với thiết bị thực:

## Sử dụng ứng dụng

# **Kết luận**

* Tóm tắt
* Khó khăn
* Hướng khắc phục

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

**TP.HCM, ngày … /… /20…**

**GIẢNG VIÊN HD**