**Sources code for creating events**

public class EventPage extends Home {  
 private RecyclerView mBlogList;  
 private DatabaseReference mDatabase;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 LayoutInflater inflater = (LayoutInflater) this.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 //inflate your activity layout here!  
 View contentView = inflater.inflate(R.layout.*activity\_event\_page*, null, false);  
 drawer.addView(contentView, 0);  
 fab.setVisibility(View.*INVISIBLE*);  
 mDatabase = FirebaseDatabase.*getInstance*().getReference("Bazaar").child("Events");  
 mBlogList = (RecyclerView) findViewById(R.id.*post*);  
 mBlogList.setHasFixedSize(true);  
 mBlogList.setLayoutManager(new LinearLayoutManager(this));  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
 FirebaseRecyclerAdapter<EventInfo, BlogViewHolder> fireBaseRecyclerAdapter = new FirebaseRecyclerAdapter<EventInfo, BlogViewHolder>(  
 EventInfo.class,  
 R.layout.*blog\_row*,  
 BlogViewHolder.class,  
 mDatabase  
 ) {  
 @Override  
 protected void populateViewHolder(BlogViewHolder viewHolder, EventInfo model, int position) {  
 viewHolder.setTitle(model.getEventName());  
 viewHolder.setDesc(model.getEventDescription());  
 viewHolder.setImage(getApplicationContext(), model.getEventImageUrl());  
 viewHolder.setDateAndTime("Date: " + model.getEventDate() + ", @ " + model.getEventTime());  
 viewHolder.setLocation("Location: " + model.getEventLocation());  
 }  
 };  
 mBlogList.setAdapter(fireBaseRecyclerAdapter);  
 }  
  
 public static class BlogViewHolder extends RecyclerView.ViewHolder {  
 View mView;  
 public BlogViewHolder(View itemView) {  
 super(itemView);  
 mView = itemView;  
 }  
  
 public void setTitle(String title) {  
 TextView post\_title = (TextView) mView.findViewById(R.id.*post\_title*);  
 post\_title.setText(title);  
 }  
  
 public void setDateAndTime(String dateAndTime) {  
 TextView post\_date\_time = (TextView) mView.findViewById(R.id.*post\_date\_time*);  
 post\_date\_time.setText(dateAndTime);  
 }  
  
 public void setDesc(String desc) {  
 TextView post\_desc = (TextView) mView.findViewById(R.id.*post\_desc*);  
 post\_desc.setText(desc);  
 }  
  
 public void setLocation(String location) {  
 TextView post\_desc = (TextView) mView.findViewById(R.id.*post\_location*);  
 post\_desc.setText(location);  
 }  
  
 public void setImage(Context ctx, String image) {  
  
 ImageView post\_image = (ImageView) mView.findViewById(R.id.*post\_image*);  
 Picasso.*with*(ctx).load(image).into(post\_image);  
 }  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*home*, menu);  
 addItem = menu.findItem(R.id.*addButton*);  
 MenuItem settingsItem = menu.findItem(R.id.*action\_settings*);  
 addItem.setVisible(true);  
 settingsItem.setVisible(true);  
 return super.onCreateOptionsMenu(menu);  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
  
 if (item.getItemId() == R.id.*addButton*) {  
 startActivity(new Intent(this, CreateEventActivity.class));  
 }  
 return super.onOptionsItemSelected(item);  
 }  
}

**Source code for selecting menus from Home**

@Override  
public boolean onNavigationItemSelected(MenuItem item) {  
 // Handle navigation view item clicks here.  
 int id = item.getItemId();  
 Intent intent;  
  
 if (id == R.id.*club*) {  
 intent = new Intent(this, ClubActivity.class);  
 startActivity(intent);  
 } else if (id == R.id.*trade*) {  
 intent = new Intent(this, TradeTypeSelectionActivity.class);  
 startActivity(intent);  
 } else if (id == R.id.*communication*) {  
 intent = new Intent(this, Communication.class);  
 startActivity(intent);  
 } else if (id == R.id.*profile*) {  
 intent = new Intent(this, Memberinfo.class);  
 startActivity(intent);  
 } else if (id == R.id.*search*) {  
 intent = new Intent(this, SearchActivity.class);  
 startActivity(intent);  
 } else if (id == R.id.*logout*) {  
 intent = new Intent(this, MainActivity.class);  
 startActivity(intent);  
 } else if (id == R.id.*home\_page*) {  
 intent = new Intent(this, EventPage.class);  
 startActivity(intent);  
 }  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.*drawer\_layout*);  
 drawer.closeDrawer(GravityCompat.*START*);  
 return true;  
}

**Parts of Source code for registering a user into the database**

public void registerUser(View view) {  
 goAhead = true;  
 try {  
 *user* = username.getText().toString();  
 *pass* = password.getText().toString();  
 *rpass* = rpassword.getText().toString();  
 *addr* = address.getText().toString();  
 *email* = Email.getText().toString();  
 *pNumber* = phoneNumber.getText().toString();  
 *name* = Name.getText().toString();  
 imagePath = downloadUri.toString();  
 if (*pass*.compareTo(*rpass*) != 0) {  
 Toast.*makeText*(RegisterActivity.this, "Password and Re-enter Password not matching", Toast.*LENGTH\_LONG*).show();  
 goAhead = false;  
 }  
 } catch (NullPointerException e) {  
 Toast.*makeText*(RegisterActivity.this, "Complete all the fields above", Toast.*LENGTH\_LONG*).show();  
 goAhead = false;  
 } catch (Exception e) {  
 Toast.*makeText*(RegisterActivity.this, "Please wait until Upload done notification before proceeding.", Toast.*LENGTH\_LONG*).show();  
 goAhead = false;  
 }  
  
  
 for (int i = 0; i < users.size(); i++) {  
 if (users.get(i).getEmail().compareTo(*email*) == 0) {  
 goAhead = false;  
 Toast.*makeText*(RegisterActivity.this, "Account already created using this email. Try Login", Toast.*LENGTH\_LONG*).show();  
 }  
 if (users.get(i).getUserName().compareTo(*user*) == 0) {  
 goAhead = false;  
 Toast.*makeText*(RegisterActivity.this, "Username already exists. Choose another Username", Toast.*LENGTH\_LONG*).show();  
  
 }  
 }  
  
 if (goAhead == true) {  
 Firebase mRefChild = mFire.child(*user*);  
 mRefChild.push();  
 Firebase newChild = mRefChild.child("name");  
 newChild.push();  
 newChild.setValue(*name*);  
 newChild = mRefChild.child("userName");  
 newChild.setValue(*user*);  
 newChild = mRefChild.child("address");  
 newChild.setValue(*addr*);  
 newChild = mRefChild.child("email");  
 newChild.setValue(*email*);  
 newChild = mRefChild.child("phoneNumber");  
 newChild.setValue(*pNumber*);  
 newChild = mRefChild.child("password");  
 newChild.setValue(*pass*);  
 newChild = mRefChild.child("profilePic\_imageURL");  
 newChild.setValue(imagePath);  
 Intent intent = new Intent(RegisterActivity.this, SignInActivity.class);  
 startActivity(intent);  
 }  
}  
@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 StorageReference userProfilePicture;  
 if (*user* == null || *name* == null || *addr* == null || *email* == null || *pNumber* == null || password == null) {  
 Toast.*makeText*(RegisterActivity.this, "Please complete the fields above.", Toast.*LENGTH\_SHORT*).show();  
 } else {  
 StorageReference userPicture = mStorage.child(*user*);  
 userProfilePicture = userPicture.child("Profile Picture");  
 super.onActivityResult(requestCode, resultCode, data);  
 // When an Image is picked from Gallery  
 if (requestCode == *GALLERY\_INTENT* && resultCode == *RESULT\_OK* && null != data) {  
 progress = new ProgressDialog(this);  
 progress.setMessage("Uploading....");  
 progress.setProgressStyle(ProgressDialog.*STYLE\_HORIZONTAL*);  
 progress.setIndeterminate(true);  
 progress.setProgress(0);  
 progress.show();  
 // Get the Image from data  
 Uri selectedImage = data.getData();  
 //Set the image into imageView  
 myImageView.setImageURI(selectedImage);  
 //UPLOAD IMAGE TO FIREBASE  
 StorageReference filePath = userProfilePicture.child("ProfilePic\_" + *user*);  
 filePath.putFile(selectedImage).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {  
 downloadUri = taskSnapshot.getDownloadUrl();  
 Picasso.*with*(RegisterActivity.this).load(downloadUri).into(myImageView);  
 Toast.*makeText*(RegisterActivity.this, "Upload Done.", Toast.*LENGTH\_SHORT*).show();  
 progress.dismiss();  
 }  
 }).addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 // Handle unsuccessful uploads  
 Toast.*makeText*(RegisterActivity.this, "Upload Unsuccessful.", Toast.*LENGTH\_SHORT*).show();  
 progress.dismiss();  
 }  
 });  
  
 }  
 //When image is picked from camera  
 else if (requestCode == *CAMERA\_REQUEST\_CODE* && resultCode == *RESULT\_OK* && null != data) {  
 progress = new ProgressDialog(this);  
 progress.setMessage("Uploading....");  
 progress.setProgressStyle(ProgressDialog.*STYLE\_HORIZONTAL*);  
 progress.setIndeterminate(true);  
 progress.setProgress(0);  
 progress.show();  
  
 // Get the Image from Camera  
 Uri cameraImage = data.getData();  
 //Set the image into imageView  
 myImageView.setImageURI(cameraImage);  
 //UPLOAD IMAGE TO FIREBASE  
 StorageReference filePathCamera = userProfilePicture.child("ProfilePic\_" + *user*);  
 filePathCamera.putFile(cameraImage).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {  
 downloadUri = taskSnapshot.getDownloadUrl();  
 Picasso.*with*(RegisterActivity.this).load(downloadUri).into(myImageView);  
 Toast.*makeText*(RegisterActivity.this, "Upload Done.", Toast.*LENGTH\_SHORT*).show();  
 progress.dismiss();  
 }  
 }).addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 // Handle unsuccessful uploads  
 Toast.*makeText*(RegisterActivity.this, "Upload Unsuccessful.", Toast.*LENGTH\_SHORT*).show();  
 progress.dismiss();  
 }  
 });  
  
 }  
 }  
  
}

**Source code for attempting login**

public void attemptLogin(View view) {  
 setUsername(user.getText().toString());  
 password = pass.getText().toString();  
 if (*username*.isEmpty() || password.isEmpty()) {  
 error.setText("Input fields cannot be empty.");  
 } else {  
 System.*out*.println("password is: " + password);  
 for (int i = 0; i < userList.size(); i++) {  
  
 if (userList.get(i).getUserName().contains(*username*)) {  
 System.*out*.println("username is: " + userList.get(i).getUserName());  
 if (userList.get(i).getPassword().compareTo(password) == 0) {  
 *setName*(userList.get(i).getName());  
 *setEmail*(userList.get(i).getEmail());  
 *setAddress*(userList.get(i).getAddress());  
 *setPhone*(userList.get(i).getPhoneNumber());  
 *setProfile\_url*(userList.get(i).getProfilePic\_imageURL());  
 System.*out*.println(userList.get(i).getPassword());  
 Toast.*makeText*(getApplicationContext(), "Logging in...",  
 Toast.*LENGTH\_SHORT*).show();  
 intent = new Intent(this, EventPage.class);  
 startActivity(intent);  
 } else {  
 error.setText("Username or Password Invalid!!");  
 }  
 }  
 }  
 }  
  
}

**Source code for background process for sending mail**

protected Void doInBackground(Void... params) {  
 //Creating properties  
 Properties props = new Properties();  
  
 //Configuring properties for gmail  
 //If you are not using gmail you may need to change the values  
 props.put("mail.smtp.host", "smtp.gmail.com");  
 props.put("mail.smtp.socketFactory.port", "465");  
 props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");  
 props.put("mail.smtp.auth", "true");  
 props.put("mail.smtp.port", "465");  
  
 //Creating a new session  
 session = Session.*getDefaultInstance*(props,  
 new javax.mail.Authenticator() {  
 //Authenticating the password  
 protected PasswordAuthentication getPasswordAuthentication() {  
 return new PasswordAuthentication(Config.*EMAIL*, Config.*PASSWORD*);  
 }  
 });  
  
 try {  
 //Creating MimeMessage object  
 MimeMessage mm = new MimeMessage(session);  
 //Setting sender address  
 mm.setFrom(new InternetAddress(Config.*EMAIL*));  
 //Adding receiver  
 mm.addRecipient(Message.RecipientType.*TO*, new InternetAddress(email));  
 mm.setText(message);  
 //Adding subject  
 mm.setSubject(subject);  
 //Sending email  
 Transport.*send*(mm);  
  
 } catch (MessagingException e) {  
 e.printStackTrace();  
 }  
 return null;  
}

**Source code for uploading sell item to the database**

public void UploadSellItems(View view) {  
 boolean goAhead = true;  
 final String userName;  
 try {  
 *iDescription* = itemDescription.getText().toString();  
 *iDes* = itemDes.getText().toString();  
 *iPrice* = itemPrice.getText().toString();  
 *iQuantity* = itemQuantity.getText().toString();  
 *sType* = sellType.getSelectedItem().toString();  
 imagePath = downloadUri.toString();  
 } catch (NullPointerException e) {  
 Toast.*makeText*(sellActivity.this, "Complete all the fields above", Toast.*LENGTH\_LONG*).show();  
 goAhead = false;  
 } catch (IllegalStateException e) {  
 Toast.*makeText*(sellActivity.this, "Please wait until Upload done notification before proceeding.", Toast.*LENGTH\_LONG*).show();  
 goAhead = false;  
 }  
 userName = new SignInActivity().getUsername();  
 if (*sType*.compareTo("Sell") == 0) {  
 mFire = mFire.child("Buy\_Items");  
  
 } else if (*sType*.compareTo("Exchange") == 0) {  
 mFire = mFire.child("Exchange\_Items");  
 } else {  
 mFire = mFire.child("Borrow\_Items");  
 }  
 if (goAhead == true) {  
  
 Firebase mRefChild = mFire.child(*iDescription*);  
 mRefChild.push();  
 Firebase newChild = mRefChild.child("itemDescription");  
 newChild.push();  
 newChild.setValue(*iDescription*);  
 newChild = mRefChild.child("itemQuantity");  
 newChild.setValue(*iQuantity*);  
 newChild = mRefChild.child("itemPrice");  
 newChild.setValue(*iPrice*);  
 newChild = mRefChild.child("sellType");  
 newChild.setValue(*sType*);  
 newChild = mRefChild.child("sellItem\_imageURL");  
 newChild.setValue(imagePath);  
 newChild = mRefChild.child("itemDes");  
 newChild.setValue(*iDes*);  
 newChild = mRefChild.child("userName");  
 newChild.setValue(userName);  
 Intent intent = new Intent(sellActivity.this, BankInfoActivity.class);  
 startActivity(intent);  
 }  
}  
  
  
@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 if (*iDescription* == null || *iQuantity* == null || *iPrice* == null) {  
 Toast.*makeText*(sellActivity.this, "Please complete the fields above.", Toast.*LENGTH\_SHORT*).show();  
 } else {  
 final String userName = new SignInActivity().getUsername();  
 StorageReference sellItemPicture = mStorage.child(userName);  
 StorageReference itemPicture = sellItemPicture.child("Sell Items");  
 super.onActivityResult(requestCode, resultCode, data);  
 // When an Image is picked from Gallery  
 if (requestCode == *GALLERY\_INTENT* && resultCode == *RESULT\_OK* && null != data) {  
 progress = new ProgressDialog(this);  
 progress.setMessage("Uploading....");  
 progress.setProgressStyle(ProgressDialog.*STYLE\_HORIZONTAL*);  
 progress.setIndeterminate(true);  
 progress.setProgress(0);  
 progress.show();  
 // Get the Image from data  
 Uri selectedImage = data.getData();  
 //Set the image into imageView  
 myImageView.setImageURI(selectedImage);  
 //UPLOAD IMAGE TO FIREBASE  
 StorageReference filePath = itemPicture.child("SellItem\_" + selectedImage.getLastPathSegment());  
 filePath.putFile(selectedImage).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {  
 downloadUri = taskSnapshot.getDownloadUrl();  
 ItemInfo item = new ItemInfo(*iDescription*, *iDes*, *iQuantity*, *iPrice*, *sType*, imagePath, userName);  
 Picasso.*with*(sellActivity.this).load(downloadUri).into(myImageView);  
 Toast.*makeText*(sellActivity.this, "Upload Done.", Toast.*LENGTH\_SHORT*).show();  
 progress.dismiss();  
 }  
 }).addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 // Handle unsuccessful uploads  
 Toast.*makeText*(sellActivity.this, "Upload Unsuccessful.", Toast.*LENGTH\_SHORT*).show();  
 progress.dismiss();  
 }  
 });  
  
 }  
 //When image is picked from camera  
 else if (requestCode == *CAMERA\_REQUEST\_CODE* && resultCode == *RESULT\_OK* && null != data) {  
 progress = new ProgressDialog(this);  
 progress.setMessage("Uploading....");  
 progress.setProgressStyle(ProgressDialog.*STYLE\_HORIZONTAL*);  
 progress.setIndeterminate(true);  
 progress.setProgress(0);  
 progress.show();  
 // Get the Image from Camera  
 Uri cameraImage = data.getData();  
 //Set the image into imageView  
 myImageView.setImageURI(cameraImage);  
 //UPLOAD IMAGE TO FIREBASE  
 StorageReference filePathCamera = itemPicture.child("SellItem\_" + cameraImage.getLastPathSegment());  
 filePathCamera.putFile(cameraImage).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {  
 @Override  
 public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {  
 downloadUri = taskSnapshot.getDownloadUrl();  
 ItemInfo item = new ItemInfo(*iDescription*, *iDes*, *iQuantity*, *iPrice*, *sType*, imagePath, userName);  
 Picasso.*with*(sellActivity.this).load(downloadUri).into(myImageView);  
 Toast.*makeText*(sellActivity.this, "Upload Done.", Toast.*LENGTH\_SHORT*).show();  
 progress.dismiss();  
 }  
 }).addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 // Handle unsuccessful uploads  
 Toast.*makeText*(sellActivity.this, "Upload Unsuccessful.", Toast.*LENGTH\_SHORT*).show();  
 progress.dismiss();  
  
 }  
 });  
  
 }  
 }

**Source code for displaying the information of the registered users**

public class SearchResultsMember extends Home {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 LayoutInflater inflater = (LayoutInflater) this.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 //inflate your activity layout here!  
 View contentView = inflater.inflate(R.layout.*activity\_search\_results\_member*, null, false);  
 drawer.addView(contentView, 0);  
 // initializing the TextView attributes in activity\_search\_results\_member xml file  
 TextView username = (TextView) findViewById(R.id.*display\_username*);  
 TextView name = (TextView) findViewById(R.id.*display\_name*);  
 TextView email = (TextView) findViewById(R.id.*display\_email*);  
 TextView address = (TextView) findViewById(R.id.*display\_address*);  
 TextView phone = (TextView) findViewById(R.id.*display\_phone*);  
 ImageView image = (ImageView) findViewById(R.id.*profile\_pic*);  
 //setting the member details to display in its corresponding TextView  
 username.setText(SearchActivity.*getUsername*());  
 name.setText(SearchActivity.*getName*());  
 email.setText(SearchActivity.*getEmail*());  
 address.setText(SearchActivity.*getAddress*());  
 phone.setText(SearchActivity.*getPhone*());  
 Picasso.*with*(getApplicationContext()).load(SearchActivity.*getProfile\_url*()).into(image);  
 }  
}

**Source code to get the image view when clicked at the particular image in the GridVIew**

public View getView(int position, View convertView, ViewGroup parent) {  
 ImageView imageView;  
 if (convertView == null) {  
 imageView = new ImageView(context);  
 imageView.setLayoutParams(new GridView.LayoutParams(400, 400));  
 imageView.setScaleType(ImageView.ScaleType.*CENTER\_CROP*);  
 imageView.setPadding(10, 10, 10, 10);  
 } else {  
 imageView = (ImageView) convertView;  
 }  
  
 String imageUrl = *itemList*.get(position).getSellItem\_imageURL();  
 Uri myImageUri = Uri.*parse*(imageUrl);  
 if (myImageUri != null) {  
  
 Picasso.*with*(Borrow\_items.this).load(myImageUri).fit().centerCrop().into(imageView);  
 }  
 return imageView;  
  
}

**Source code to change the password**

SignInActivity login = new SignInActivity();  
for (int i=0;i<users.size();i++){  
 if (users.get(i).getUserName().compareTo(login.getUsername())==0){  
 if (users.get(i).getPassword().compareTo(oldPass)==0) {  
 if (newPassword.compareTo(confirmPass) == 0) {  
 Firebase usersData = mRef.child(users.get(i).getUserName());  
 usersData.push();  
 user = users.get(i);  
 user.setPassword(newPassword);  
 users.set(i, user);  
 System.*out*.println("the new password is: " + user.getPassword());  
 Firebase newChild = usersData.child("name");  
 newChild.push();  
 newChild.setValue(user.getName());  
 newChild = usersData.child("userName");  
 newChild.setValue(user.getUserName());  
 newChild = usersData.child("address");  
 newChild.setValue(user.getAddress());  
 newChild = usersData.child("email");  
 newChild.setValue(user.getEmail());  
 newChild = usersData.child("phoneNumber");  
 newChild.setValue(user.getPhoneNumber());  
 newChild = usersData.child("password");  
 newChild.setValue(user.getPassword());  
 newChild = usersData.child("profilePic\_imageURL");  
 newChild.setValue(user.getProfilePic\_imageURL());  
 Toast.*makeText*(getApplicationContext(),"Password successfully changed.",Toast.*LENGTH\_SHORT*).show();  
 startActivity(new Intent(ChangePasswordActivity.this, Memberinfo.class));  
 } else {  
 error.setText("The new passwords entered didn't match");  
 }  
 }else{  
 error.setText("Old password didn't match");  
 }  
 }  
}

**Source code to display the buy item details**

@Override  
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 LayoutInflater inflater = (LayoutInflater) this.getSystemService(Context.*LAYOUT\_INFLATER\_SERVICE*);  
 //inflate your activity layout here!  
 View contentView = inflater.inflate(R.layout.*activity\_buy\_item\_details*, null, false);  
 drawer.addView(contentView, 0);  
 fab.setVisibility(View.*INVISIBLE*);  
 TextView desc = (TextView) findViewById(R.id.*buy\_desTextView*);  
 TextView price = (TextView) findViewById(R.id.*buy\_priceTextView*);  
 TextView name = (TextView) findViewById(R.id.*buy\_nameTextView*);  
 int position = getIntent().getExtras().getInt("id");  
 ImageView imageView = (ImageView) findViewById(R.id.*profile\_pic*);  
  
 String imageUrl = *itemList*.get(position).getSellItem\_imageURL();  
 Uri myImageUri = Uri.*parse*(imageUrl);  
  
 Picasso.*with*(Buy\_item\_details.this).load(myImageUri).into(imageView);  
 desc.setText(*itemList*.get(position).getItemDes());  
  
 price.setText("$"+*itemList*.get(position).getItemPrice());  
 name.setText(*itemList*.get(position).getItemDescription());  
}

**Source code to display the clubs in the list**

lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
 *setSelected*(adapterView.getItemAtPosition(i).toString());  
 for (int j = 0; j < users.size(); j++) {  
 ClubInfo tempUser = users.get(j);  
 System.*out*.println("here are: " + users.get(j).getClubName());  
 ;  
 if (tempUser.getClubName().compareTo(*selected*) == 0) {  
 *setClub\_name*(*selected*);  
 *setClub\_admin*(tempUser.getClubAdmin());  
 *setClub\_desc*(tempUser.getClubDescription());  
 *setClub\_category*(tempUser.getClubCategory());  
 Intent intent = new Intent(ClubActivity.this, AllClubActivity.class);  
 startActivity(intent);  
 }  
 }  
  
 }  
});

**Code for sending the password reset link with temporary password and updating the database with temporary password**

for (int i=0;i<users.size();i++){  
 if (users.get(i).getEmail().compareTo(emailAddress)==0){  
 Firebase usersData = mRef.child(users.get(i).getUserName());  
 usersData.push();  
 user = users.get(i);  
 user.setPassword(tempPass);  
 users.set(i,user);  
 System.*out*.println("the new password is: "+user.getPassword());  
 Firebase newChild = usersData.child("name");  
 newChild.push();  
 newChild.setValue(user.getName());  
 newChild = usersData.child("userName");  
 newChild.setValue(user.getUserName());  
 newChild = usersData.child("address");  
 newChild.setValue(user.getAddress());  
 newChild = usersData.child("email");  
 newChild.setValue(user.getEmail());  
 newChild = usersData.child("phoneNumber");  
 newChild.setValue(user.getPhoneNumber());  
 newChild = usersData.child("password");  
 newChild.setValue(user.getPassword());  
 newChild = usersData.child("profilePic\_imageURL");  
 newChild.setValue(user.getProfilePic\_imageURL());  
  
 String subject = "Your Temporary password";  
 String message = "Here is the temporary password for your account.\n" +  
 "Please change the change password by clicking 'change password'\n in you profile page at your next login.\n" +  
 "\n\tTemporary Password: "+tempPass+"\n\n\n\nThank You,\nBazaar Team";  
  
 SendMail sm = new SendMail(PasswordRetrivalActivity.this, emailAddress, subject, message);  
 sm.execute();  
  
 Intent intent = new Intent(PasswordRetrivalActivity.this, SignInActivity.class);  
 startActivity(intent);  
 }else{  
 status = false;  
 }  
}  
if (status == false){  
 Toast.*makeText*(getApplicationContext(),"Email not valid. Please Enter the email linked to the account.", Toast.*LENGTH\_LONG*).show();  
}