LoginRegisterActivity.java

public class LoginRegisterActivity extends AppCompatActivity {

FrameLayout frameContainer;

public static AppDatabase db;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_loginregister);

frameContainer= findViewById(R.id.frameContainer);

db= Room.databaseBuilder(this, AppDatabase.class, "MDPM7").build();

showFragment(new LoginFragment());

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.options\_menu, menu);

menu.findItem(R.id.menuLogout).setVisible(false);

return super.onCreateOptionsMenu(menu);

}

@Override

public boolean onOptionsItemSelected(@NonNull MenuItem item) {

switch (item.getItemId()) {

case R.id.menuLogin:

showFragment(new LoginFragment());

break;

case R.id.menuRegister:

showFragment(new RegisterFragment());

break;

}

return super.onOptionsItemSelected(item);

}

private void showFragment(Fragment fragment) {

FragmentTransaction trans= getSupportFragmentManager().beginTransaction();

trans.replace(R.id.frameContainer, fragment);

trans.commit();

}

}

LoginFragment.java

public class LoginFragment extends Fragment {

EditText edtUsername, edtPIN;

Button btnLogin;

Intent page;

List<User> users;

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

return inflater.inflate(R.layout.fragment\_login, container, false);

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

super.onViewCreated(view, savedInstanceState);

edtUsername= view.findViewById(R.id.edtUsername);

edtPIN= view.findViewById(R.id.edtPIN);

btnLogin= view.findViewById(R.id.btnLogin);

users= new ArrayList<>();

new GetAllTask().execute();

btnLogin.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String username= edtUsername.getText().toString();

String PIN= edtPIN.getText().toString();

if (!username.isEmpty() && !PIN.isEmpty()) {

if (username.equals("admin") && PIN.equals("123456")) {

page= new Intent(getContext(), AdminActivity.class);

startActivity(page);

} else if (checkExist(username, PIN) != null) {

page= new Intent(getContext(), MainActivity.class);

page.putExtra("logged", checkExist(username, PIN));

startActivity(page);

} else Toast.makeText(getContext(), "Wrong username or PIN!", Toast.LENGTH\_SHORT).show();

} else Toast.makeText(getContext(), "Field must not empty!", Toast.LENGTH\_SHORT).show();

}

});

}

private User checkExist(String username, String PIN) {

for (User user : users) {

if (user.getUsername().equals(username) && user.getPIN().equals(PIN) && user.getStatus() == 1) {

return user;

}

}

return null;

}

private class GetAllTask extends AsyncTask<Void, Void, Void> {

@Override

protected Void doInBackground(Void... voids) {

if (!LoginRegisterActivity.db.userDAO().getAllUser().isEmpty()) {

users.clear();

users.addAll(LoginRegisterActivity.db.userDAO().getAllUser());

}

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

}

}

}

RegisterFragment.java

public class RegisterFragment extends Fragment {

EditText edtUsername, edtName, edtPIN;

Button btnRegister;

List<User> users;

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

return inflater.inflate(R.layout.fragment\_register, container, false);

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

super.onViewCreated(view, savedInstanceState);

edtUsername= view.findViewById(R.id.edtUsername);

edtName= view.findViewById(R.id.edtName);

edtPIN= view.findViewById(R.id.edtPIN);

btnRegister= view.findViewById(R.id.btnRegister);

users= new ArrayList<>();

new GetAllTask().execute();

btnRegister.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String username= edtUsername.getText().toString();

String name= edtName.getText().toString();

String PIN= edtPIN.getText().toString();

boolean exist= false;

if (!username.isEmpty() && !name.isEmpty() && !PIN.isEmpty()) {

for (User user : users) {

if (user.getUsername().equals(username)) {

exist= true;

break;

}

}

if (!exist) {

new InsertTask().execute(new User(name, username, PIN));

refreshField();

} else Toast.makeText(getContext(), "Username already used!", Toast.LENGTH\_SHORT).show();

} else Toast.makeText(getContext(), "Field must not empty!", Toast.LENGTH\_SHORT).show();

}

});

}

private void refreshField() {

edtUsername.setText("");

edtName.setText("");

edtPIN.setText("");

}

private class InsertTask extends AsyncTask<User, Void, Void> {

@Override

protected Void doInBackground(User... users) {

LoginRegisterActivity.db.userDAO().insert(users[0]);

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

}

}

private class GetAllTask extends AsyncTask<Void, Void, Void> {

@Override

protected Void doInBackground(Void... voids) {

if (!LoginRegisterActivity.db.userDAO().getAllUser().isEmpty()) {

users.clear();

users.addAll(LoginRegisterActivity.db.userDAO().getAllUser());

}

return null;

}

@Override

protected void onPostExecute(Void aVoid) { super.onPostExecute(aVoid); }

}

}

AdminActivity.java

public class AdminActivity extends AppCompatActivity {

FrameLayout frameContainer;

BottomNavigationView bottomNav;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_admin);

frameContainer= findViewById(R.id.frameContainer);

bottomNav= findViewById(R.id.bottomNav);

showFragment(new MasterUserFragment());

bottomNav.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {

@Override

public boolean onNavigationItemSelected(@NonNull MenuItem menuItem) {

switch (menuItem.getItemId()) {

case R.id.menuMasterUser:

showFragment(new MasterUserFragment());

break;

case R.id.menuMasterMerchant:

showFragment(new MasterMerchantFragment());

break;

}

return true;

}

});

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.options\_menu, menu);

menu.findItem(R.id.menuLogin).setVisible(false);

menu.findItem(R.id.menuRegister).setVisible(false);

menu.findItem(R.id.menuLogout).setVisible(true);

return super.onCreateOptionsMenu(menu);

}

@Override

public boolean onOptionsItemSelected(@NonNull MenuItem item) {

if (item.getItemId() == R.id.menuLogout) startActivity(new Intent(this, LoginRegisterActivity.class));

return super.onOptionsItemSelected(item);

}

private void showFragment(Fragment fragment) {

FragmentTransaction trans= getSupportFragmentManager().beginTransaction();

trans.replace(R.id.frameContainer, fragment);

trans.commit();

}

}

MasterUserFragment.java

public class MasterUserFragment extends Fragment {

RecyclerView rvUser;

List<User> users;

UserAdapter adapter;

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

return inflater.inflate(R.layout.fragment\_masteruser, container, false);

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

super.onViewCreated(view, savedInstanceState);

rvUser= view.findViewById(R.id.rvUser);

users= new ArrayList<>();

new GetAllTask().execute();

adapter= new UserAdapter(users);

rvUser.setHasFixedSize(true);

rvUser.setLayoutManager(new LinearLayoutManager(getContext()));

rvUser.setAdapter(adapter);

adapter.setOnClickCallback(new UserAdapter.OnClickCallback() {

@Override

public void onBtnClick(int position) {

if (users.get(position).getStatus() == 0) users.get(position).setStatus(1);

else if (users.get(position).getStatus() == 1) users.get(position).setStatus(0);

new UpdateTask().execute(users.get(position));

}

});

}

private class UpdateTask extends AsyncTask<User, Void, Void> {

@Override

protected Void doInBackground(User... users) {

LoginRegisterActivity.db.userDAO().update(users[0]);

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

new GetAllTask().execute();

}

}

private class GetAllTask extends AsyncTask<Void, Void, Void> {

@Override

protected Void doInBackground(Void... voids) {

if (!LoginRegisterActivity.db.userDAO().getAllUser().isEmpty()) {

users.clear();

users.addAll(LoginRegisterActivity.db.userDAO().getAllUser());

}

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

adapter.notifyDataSetChanged();

}

}

}

MasterMerchantFragment.java

public class MasterMerchantFragment extends Fragment {

EditText edtName;

Button btnAdd;

ListView lvMerchant;

List<Merchant> merchants;

ArrayAdapter adapter;

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

return inflater.inflate(R.layout.fragment\_mastermerchant, container, false);

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

super.onViewCreated(view, savedInstanceState);

edtName= view.findViewById(R.id.edtName);

btnAdd= view.findViewById(R.id.btnAdd);

lvMerchant= view.findViewById(R.id.lvMerchant);

merchants= new ArrayList<>();

new GetAllTask().execute();

adapter= new ArrayAdapter(getContext(), android.R.layout.simple\_list\_item\_1, merchants);

lvMerchant.setAdapter(adapter);

btnAdd.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String name= edtName.getText().toString();

if (!name.isEmpty()) {

new InsertTask().execute(new Merchant(name));

} else {

Toast.makeText(getContext(), "Field must not empty!", Toast.LENGTH\_SHORT).show();

}

}

});

lvMerchant.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {

@Override

public boolean onItemLongClick(AdapterView<?> adapterView, View view, int i, long l) {

onCreatePopUpMenu(view, i);

return false;

}

});

}

private void onCreatePopUpMenu(View view, final int position) {

PopupMenu popup= new PopupMenu(getContext(), view);

popup.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {

@Override

public boolean onMenuItemClick(MenuItem menuItem) {

if (menuItem.getItemId() == R.id.menuDelete) {

new DeleteTask().execute(merchants.get(position));

}

return false;

}

});

popup.inflate(R.menu.context\_menu);

popup.show();

}

private class InsertTask extends AsyncTask<Merchant, Void, Void> {

@Override

protected Void doInBackground(Merchant... merchants) {

LoginRegisterActivity.db.merchantDAO().insert(merchants[0]);

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

new GetAllTask().execute();

}

}

private class DeleteTask extends AsyncTask<Merchant, Void, Void> {

@Override

protected Void doInBackground(Merchant... merchants) {

LoginRegisterActivity.db.merchantDAO().delete(merchants[0]);

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

new GetAllTask().execute();

}

}

private class GetAllTask extends AsyncTask<Void, Void, Void> {

@Override

protected Void doInBackground(Void... voids) {

if (!LoginRegisterActivity.db.merchantDAO().getAllMerchant().isEmpty()) {

merchants.clear();

merchants.addAll(LoginRegisterActivity.db.merchantDAO().getAllMerchant());

}

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

adapter.notifyDataSetChanged();

}

}

}

MainActivity.java

public class MainActivity extends AppCompatActivity {

FrameLayout frameContainer;

BottomNavigationView bottomNav;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

frameContainer= findViewById(R.id.frameContainer);

bottomNav= findViewById(R.id.bottomNav);

showFragment(new HomeFragment());

bottomNav.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {

@Override

public boolean onNavigationItemSelected(@NonNull MenuItem menuItem) {

switch (menuItem.getItemId()) {

case R.id.menuHome:

showFragment(new HomeFragment());

break;

case R.id.menuHistory:

showFragment(new HistoryFragment());

break;

case R.id.menuPayment:

showFragment(new PaymentFragment());

break;

case R.id.menuTransfer:

showFragment(new TransferFragment());

break;

case R.id.menuMyAccount:

showFragment(new MyAccountFragment());

break;

}

return true;

}

});

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.options\_menu, menu);

menu.findItem(R.id.menuLogin).setVisible(false);

menu.findItem(R.id.menuRegister).setVisible(false);

menu.findItem(R.id.menuLogout).setVisible(true);

return super.onCreateOptionsMenu(menu);

}

@Override

public boolean onOptionsItemSelected(@NonNull MenuItem item) {

if (item.getItemId() == R.id.menuLogout) startActivity(new Intent(this, LoginRegisterActivity.class));

return super.onOptionsItemSelected(item);

}

private void showFragment(Fragment fragment) {

Bundle arguments= new Bundle();

FragmentTransaction trans= getSupportFragmentManager().beginTransaction();

arguments.putSerializable("logged", getIntent().getSerializableExtra("logged"));

fragment.setArguments(arguments);

trans.replace(R.id.frameContainer, fragment);

trans.commit();

}

}

HomeFragment.java

public class HomeFragment extends Fragment {

TextView tvLogged, tvBalance;

EditText edtAmount;

Button btnTopUp;

RecyclerView rvFeeds;

List<Transaction> feeds;

List<User> users;

User logged;

FeedsAdapter adapter;

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

return inflater.inflate(R.layout.fragment\_home, container, false);

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

super.onViewCreated(view, savedInstanceState);

tvLogged= view.findViewById(R.id.tvLogged);

tvBalance= view.findViewById(R.id.tvBalance);

edtAmount= view.findViewById(R.id.edtAmount);

btnTopUp= view.findViewById(R.id.btnTopUp);

rvFeeds= view.findViewById(R.id.rvFeeds);

feeds= new ArrayList<>();

users= new ArrayList<>();

logged= (User) getArguments().getSerializable("logged");

new GetAllUserTask().execute();

new GetAllTransactionTask().execute(logged.getUsername());

adapter= new FeedsAdapter(feeds);

rvFeeds.setHasFixedSize(true);

rvFeeds.setLayoutManager(new LinearLayoutManager(getContext()));

rvFeeds.setAdapter(adapter);

if (logged != null) {

tvLogged.setText("Welcome, "+logged.getName());

tvBalance.setText("Your Balance: "+logged.getCurrencyFormat());

Toast.makeText(getContext(), logged.toString(), Toast.LENGTH\_SHORT).show();

}

btnTopUp.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

int amount= 0;

if (!edtAmount.getText().toString().isEmpty()) {

amount= Integer.parseInt(edtAmount.getText().toString());

logged.setBalance(amount);

new UpdateTask().execute(logged);

tvBalance.setText("Your Balance: "+logged.getCurrencyFormat());

} else Toast.makeText(getContext(), "Field must not empty!", Toast.LENGTH\_SHORT).show();

}

});

adapter.setOnItemClickCallback(new FeedsAdapter.OnItemClickCallback() {

@Override

public void onItemClicked(int position) {

User other= getUser(feeds.get(position).getOther());

if (other != null) {

logged.setBalance(-feeds.get(position).getNominal());

other.setBalance(feeds.get(position).getNominal());

new UpdateTask().execute(logged);

new UpdateTask().execute(other);

new InsertTask().execute(new Transaction(logged.getUsername(), logged.getName(), other.getName(), feeds.get(position).getNominal(), "out"));

new InsertTask().execute(new Transaction(other.getUsername(), other.getName(), logged.getName(), feeds.get(position).getNominal(), "in"));

new DeleteTask().execute(feeds.get(position));

tvBalance.setText("Your Balance: "+logged.getCurrencyFormat());

}

}

});

}

private User getUser(String username) {

for (User user : users) {

if (user.getUsername().equals(username)) {

return user;

}

}

return null;

}

private class InsertTask extends AsyncTask<Transaction, Void, Void> {

@Override

protected Void doInBackground(Transaction... transactions) {

LoginRegisterActivity.db.transactionDAO().insert(transactions[0]);

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

}

}

private class DeleteTask extends AsyncTask<Transaction, Void, Void> {

@Override

protected Void doInBackground(Transaction... transactions) {

LoginRegisterActivity.db.transactionDAO().delete(transactions[0]);

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

new GetAllTransactionTask();

}

}

private class UpdateTask extends AsyncTask<User, Void, Void> {

@Override

protected Void doInBackground(User... users) {

LoginRegisterActivity.db.userDAO().update(users[0]);

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

}

}

private class GetAllUserTask extends AsyncTask<Void, Void, Void> {

@Override

protected Void doInBackground(Void... voids) {

if (!LoginRegisterActivity.db.userDAO().getAllUser().isEmpty()) {

users.clear();

users.addAll(LoginRegisterActivity.db.userDAO().getAllUser());

}

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

}

}

private class GetAllTransactionTask extends AsyncTask<String, Void, Void> {

@Override

protected Void doInBackground(String... usernames) {

if (!LoginRegisterActivity.db.transactionDAO().getAllFeedsTransactionFrom(usernames[0]).isEmpty()) {

feeds.clear();

feeds.addAll(LoginRegisterActivity.db.transactionDAO().getAllFeedsTransactionFrom(usernames[0]));

}

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

adapter.notifyDataSetChanged();

}

}

}

HistoryFragment.java

public class HistoryFragment extends Fragment {

RadioButton rbIn, rbOut, rbAll;

EditText edtUsername;

Button btnShow;

RecyclerView rvHistories;

User logged;

List<Transaction> displayHistory;

HistoryAdapter adapter;

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

return inflater.inflate(R.layout.fragment\_history, container, false);

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

super.onViewCreated(view, savedInstanceState);

rbIn= view.findViewById(R.id.rbIn);

rbOut= view.findViewById(R.id.rbOut);

rbAll= view.findViewById(R.id.rbAll);

edtUsername= view.findViewById(R.id.edtUsername);

btnShow= view.findViewById(R.id.btnShow);

rvHistories= view.findViewById(R.id.rvHistories);

logged= (User) getArguments().getSerializable("logged");

displayHistory= new ArrayList<>();

new GetAllTask().execute(logged.getUsername());

adapter= new HistoryAdapter(displayHistory);

rvHistories.setHasFixedSize(true);

rvHistories.setLayoutManager(new LinearLayoutManager(getContext()));

rvHistories.setAdapter(adapter);

btnShow.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String username= edtUsername.getText().toString();

if (rbIn.isChecked()) {

displayHistory.clear();

if (!username.isEmpty()) new GetAllByTypeWithUsernameTask().execute(logged.getUsername(), username, "in");

else new GetAllByTypeTask().execute(logged.getUsername(), "in");

} else if (rbOut.isChecked()) {

displayHistory.clear();

if (!username.isEmpty()) new GetAllByTypeWithUsernameTask().execute(logged.getUsername(), username, "out");

else new GetAllByTypeTask().execute(logged.getUsername(), "out");

} else if (rbAll.isChecked()) {

displayHistory.clear();

if (!username.isEmpty()) new GetAllByUsernameTask().execute(logged.getUsername(), username);

else new GetAllTask().execute(logged.getUsername());

}

}

});

}

private class GetAllTask extends AsyncTask<String, Void, Void> {

@Override

protected Void doInBackground(String... usernames) {

if (!LoginRegisterActivity.db.transactionDAO().getAllTransactionFrom(usernames[0]).isEmpty()) {

displayHistory.clear();

displayHistory.addAll(LoginRegisterActivity.db.transactionDAO().getAllTransactionFrom(usernames[0]));

}

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

adapter.notifyDataSetChanged();

}

}

private class GetAllByUsernameTask extends AsyncTask<String, Void, Void> {

@Override

protected Void doInBackground(String... usernames) {

if (!LoginRegisterActivity.db.transactionDAO().getAllTransactionByUsername(usernames[0], usernames[1]).isEmpty()) {

displayHistory.clear();

displayHistory.addAll(LoginRegisterActivity.db.transactionDAO().getAllTransactionByUsername(usernames[0], usernames[1]));

}

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

adapter.notifyDataSetChanged();

}

}

private class GetAllByTypeTask extends AsyncTask<String, Void, Void> {

@Override

protected Void doInBackground(String... strings) {

if (!LoginRegisterActivity.db.transactionDAO().getAllTransactionByType(strings[0], strings[1]).isEmpty()) {

displayHistory.clear();

displayHistory.addAll(LoginRegisterActivity.db.transactionDAO().getAllTransactionByType(strings[0], strings[1]));

}

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

adapter.notifyDataSetChanged();

}

}

private class GetAllByTypeWithUsernameTask extends AsyncTask<String, Void, Void> {

@Override

protected Void doInBackground(String... strings) {

if (!LoginRegisterActivity.db.transactionDAO().getAllTransactionByTypeWithUsername(strings[0], strings[1], strings[2]).isEmpty()) {

displayHistory.clear();

displayHistory.addAll(LoginRegisterActivity.db.transactionDAO().getAllTransactionByTypeWithUsername(strings[0], strings[1], strings[2]));

}

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

adapter.notifyDataSetChanged();

}

}

}

PaymentFragment.java

public class PaymentFragment extends Fragment {

Spinner spinMerchant;

EditText edtNominal;

Button btnPay;

List<Merchant> merchants;

ArrayAdapter adapter;

User logged;

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

return inflater.inflate(R.layout.fragment\_payment, container, false);

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

super.onViewCreated(view, savedInstanceState);

spinMerchant= view.findViewById(R.id.spinMerchant);

edtNominal= view.findViewById(R.id.edtNominal);

btnPay= view.findViewById(R.id.btnPay);

merchants= new ArrayList<>();

new GetAllTask().execute();

adapter= new ArrayAdapter(getContext(), android.R.layout.simple\_spinner\_dropdown\_item, merchants);

logged= (User) getArguments().getSerializable("logged");

spinMerchant.setAdapter(adapter);

spinMerchant.setSelection(0);

btnPay.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Merchant merchant= (Merchant) spinMerchant.getSelectedItem();

int nominal= 0;

if (merchant != null && !edtNominal.getText().toString().isEmpty()) {

nominal= Integer.parseInt(edtNominal.getText().toString());

if (logged.getBalance() > nominal) {

logged.setBalance(-nominal);

new UpdateTask().execute(logged);

new InsertTask().execute(new Transaction(logged.getUsername(), logged.getName(), merchant.getName(), nominal, "out", true));

} else Toast.makeText(getContext(), "Your balance not enough!", Toast.LENGTH\_SHORT).show();

} else Toast.makeText(getContext(), "Field must not empty!", Toast.LENGTH\_SHORT).show();

}

});

}

private class InsertTask extends AsyncTask<Transaction, Void, Void> {

@Override

protected Void doInBackground(Transaction... transactions) {

LoginRegisterActivity.db.transactionDAO().insert(transactions[0]);

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

}

}

private class UpdateTask extends AsyncTask<User, Void, Void> {

@Override

protected Void doInBackground(User... users) {

LoginRegisterActivity.db.userDAO().update(users[0]);

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

}

}

private class GetAllTask extends AsyncTask<Void, Void, Void> {

@Override

protected Void doInBackground(Void... voids) {

if (!LoginRegisterActivity.db.merchantDAO().getAllMerchant().isEmpty()) {

merchants.clear();

merchants.addAll(LoginRegisterActivity.db.merchantDAO().getAllMerchant());

}

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

adapter.notifyDataSetChanged();

}

}

}

TransferFragment.java

package com.example.mdpm7\_6615;

import android.os.AsyncTask;

import android.os.Bundle;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.EditText;

import android.widget.RadioButton;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.fragment.app.Fragment;

import java.util.ArrayList;

import java.util.List;

public class TransferFragment extends Fragment {

RadioButton rbTransfer, rbRequest;

EditText edtNominal, edtDestination;

Button btnSubmit;

List<User> users;

User logged;

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

return inflater.inflate(R.layout.fragment\_transfer, container, false);

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

super.onViewCreated(view, savedInstanceState);

rbTransfer= view.findViewById(R.id.rbTransfer);

rbRequest= view.findViewById(R.id.rbRequest);

edtNominal= view.findViewById(R.id.edtNominal);

edtDestination= view.findViewById(R.id.edtDestination);

btnSubmit= view.findViewById(R.id.btnSubmit);

users= new ArrayList<>();

new GetAllTask().execute();

logged= (User) getArguments().getSerializable("logged");

btnSubmit.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String type= "";

if (rbTransfer.isChecked()) type= "out";

else if (rbRequest.isChecked()) type= "request";

if (!type.isEmpty() && !edtNominal.getText().toString().isEmpty() && !edtDestination.getText().toString().isEmpty()) {

int nominal= Integer.parseInt(edtNominal.getText().toString());

User other= checkExist(edtDestination.getText().toString());

if (type.equals("out")) {

if (logged.getBalance() >= nominal) {

if (other != null) {

logged.setBalance(-nominal);

other.setBalance(nominal);

new UpdateTask().execute(logged);

new UpdateTask().execute(other);

new InsertTask().execute(new Transaction(logged.getUsername(), logged.getName(), other.getName(), nominal, "out"));

new InsertTask().execute(new Transaction(other.getUsername(), other.getName(), logged.getName(), nominal, "in"));

} else Toast.makeText(getContext(), "ID not found!", Toast.LENGTH\_SHORT).show();

} else Toast.makeText(getContext(), "Your balance not enough!", Toast.LENGTH\_SHORT).show();

} else if (type.equals("request")) {

if (other != null) {

new InsertTask().execute(new Transaction(other.getUsername(), other.getName(), logged.getName(), nominal, "request"));

} else Toast.makeText(getContext(), "ID not found!", Toast.LENGTH\_SHORT).show();

}

} else Toast.makeText(getContext(), "All field must not empty!", Toast.LENGTH\_SHORT).show();

}

});

}

private User checkExist(String username) {

for (User user : users) {

if (user.getUsername().equals(username)) {

return user;

}

}

return null;

}

private class InsertTask extends AsyncTask<Transaction, Void, Void> {

@Override

protected Void doInBackground(Transaction... transactions) {

LoginRegisterActivity.db.transactionDAO().insert(transactions[0]);

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

}

}

private class UpdateTask extends AsyncTask<User, Void, Void> {

@Override

protected Void doInBackground(User... users) {

LoginRegisterActivity.db.userDAO().update(users[0]);

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

}

}

private class GetAllTask extends AsyncTask<Void, Void, Void> {

@Override

protected Void doInBackground(Void... voids) {

if (!LoginRegisterActivity.db.userDAO().getAllUser().isEmpty()) {

users.clear();

users.addAll(LoginRegisterActivity.db.userDAO().getAllUser());

}

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

}

}

}

MyAccountFragment.java

public class MyAccountFragment extends Fragment {

TextView tvUsername;

EditText edtName, edtNewPIN, edtOldPIN;

Button btnSave;

User logged;

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

return inflater.inflate(R.layout.fragment\_myaccount, container, false);

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

super.onViewCreated(view, savedInstanceState);

tvUsername= view.findViewById(R.id.tvUsername);

edtName= view.findViewById(R.id.edtName);

edtNewPIN= view.findViewById(R.id.edtNewPIN);

edtOldPIN= view.findViewById(R.id.edtOldPIN);

btnSave= view.findViewById(R.id.btnSave);

logged= (User) getArguments().getSerializable("logged");

if (logged != null) {

tvUsername.setText(logged.getUsername());

edtName.setText(logged.getName());

}

btnSave.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String name= edtName.getText().toString();

String newPIN= edtNewPIN.getText().toString();

String oldPIN= edtOldPIN.getText().toString();

if (!newPIN.isEmpty() && !oldPIN.isEmpty()) {

if (logged.getPIN().equals(oldPIN)) logged.setPIN(newPIN);

else Toast.makeText(getContext(), "Wrong old PIN!", Toast.LENGTH\_SHORT).show();

}

if (!name.isEmpty()) logged.setName(name);

new UpdateTask().execute(logged);

}

});

}

private class UpdateTask extends AsyncTask<User, Void, Void> {

@Override

protected Void doInBackground(User... users) {

LoginRegisterActivity.db.userDAO().update(users[0]);

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

}

}

}

UserAdapter.java

public class UserAdapter extends RecyclerView.Adapter<UserAdapter.UserViewHolder> {

private List<User> users;

private OnClickCallback onClickCallback;

public UserAdapter(List<User> users) {

this.users = users;

}

public void setOnClickCallback(OnClickCallback onClickCallback) {

this.onClickCallback = onClickCallback;

}

@NonNull

@Override

public UserAdapter.UserViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

return new UserAdapter.UserViewHolder(LayoutInflater.from(parent.getContext()).inflate(R.layout.layout\_user, parent, false));

}

@Override

public void onBindViewHolder(@NonNull UserAdapter.UserViewHolder holder, final int position) {

User user= users.get(position);

holder.ivUser.setImageResource(R.drawable.account);

holder.tvName.setText(user.getName());

if (user.getStatus() == 0) holder.btnActive.setText("Enable");

else if (user.getStatus() == 1) holder.btnActive.setText("Disable");

holder.btnActive.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

if (onClickCallback != null) {

onClickCallback.onBtnClick(position);

}

}

});

}

@Override

public int getItemCount() {

return this.users.size();

}

public class UserViewHolder extends RecyclerView.ViewHolder {

ImageView ivUser;

TextView tvName;

Button btnActive;

public UserViewHolder(@NonNull View itemView) {

super(itemView);

this.ivUser= itemView.findViewById(R.id.ivUser);

this.tvName= itemView.findViewById(R.id.tvName);

this.btnActive= itemView.findViewById(R.id.btnActive);

}

}

public interface OnClickCallback { void onBtnClick(int position); }

}

FeedsAdapter.java

public class FeedsAdapter extends RecyclerView.Adapter<FeedsAdapter.FeedsViewHolder> {

private List<Transaction> feeds;

private OnItemClickCallback onItemClickCallback;

public FeedsAdapter(List<Transaction> feeds) {

this.feeds = feeds;

}

public void setOnItemClickCallback(OnItemClickCallback onItemClickCallback) { this.onItemClickCallback = onItemClickCallback; }

@NonNull

@Override

public FeedsAdapter.FeedsViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

return new FeedsAdapter.FeedsViewHolder(LayoutInflater.from(parent.getContext()).inflate(R.layout.layout\_feeds, parent, false));

}

@Override

public void onBindViewHolder(@NonNull FeedsAdapter.FeedsViewHolder holder, final int position) {

Transaction feed= this.feeds.get(position);

if (feed.getType().equals("in")) {

holder.tvDescription.setText(feed.getOther()+" telah mengirimkan dana sebesar "+feed.getCurrencyFormat());

holder.linearContainer.setBackgroundColor(Color.GREEN);

holder.btnSend.setVisibility(View.INVISIBLE);

} else if (feed.getType().equals("request")) {

holder.tvDescription.setText(feed.getOther()+" telah meminta dana sebesar "+feed.getCurrencyFormat());

holder.linearContainer.setBackgroundColor(Color.YELLOW);

holder.btnSend.setVisibility(View.VISIBLE);

}

holder.tvDateTime.setText(feed.getDateTime());

holder.btnSend.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

if (onItemClickCallback != null) {

onItemClickCallback.onItemClicked(position);

}

}

});

}

@Override

public int getItemCount() {

return this.feeds.size();

}

public class FeedsViewHolder extends RecyclerView.ViewHolder {

LinearLayout linearContainer;

TextView tvDescription, tvDateTime;

Button btnSend;

public FeedsViewHolder(@NonNull View itemView) {

super(itemView);

this.linearContainer= itemView.findViewById(R.id.linearContainer);

this.tvDescription= itemView.findViewById(R.id.tvDescription);

this.tvDateTime= itemView.findViewById(R.id.tvDateTime);

this.btnSend= itemView.findViewById(R.id.btnSend);

}

}

public interface OnItemClickCallback { void onItemClicked(int position); }

}

HistoryAdapter.java

public class HistoryAdapter extends RecyclerView.Adapter<HistoryAdapter.HistoryViewHolder>{

private List<Transaction> histories;

public HistoryAdapter(List<Transaction> histories) {

this.histories = histories;

}

@NonNull

@Override

public HistoryAdapter.HistoryViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

return new HistoryViewHolder(LayoutInflater.from(parent.getContext()).inflate(R.layout.layout\_history, parent, false));

}

@Override

public void onBindViewHolder(@NonNull HistoryAdapter.HistoryViewHolder holder, int position) {

Transaction history= histories.get(position);

holder.tvDateTime.setText(history.getDateTime());

holder.tvMoney.setText(history.getCurrencyFormat());

if (history.getType().equals("in")) {

holder.tvDescription.setText("Uang masuk dari "+history.getOther());

holder.tvCode.setText("CR");

holder.tvMoney.setTextColor(Color.GREEN);

holder.tvCode.setTextColor(Color.GREEN);

} else if (history.getType().equals("out")) {

if (history.isPayment()) holder.tvDescription.setText("Payment ke "+history.getOther());

else holder.tvDescription.setText("Transfer ke "+history.getOther());

holder.tvCode.setText("DB");

holder.tvMoney.setTextColor(Color.RED);

holder.tvCode.setTextColor(Color.RED);

}

}

@Override

public int getItemCount() {

return this.histories.size();

}

public class HistoryViewHolder extends RecyclerView.ViewHolder {

TextView tvDateTime, tvMoney, tvDescription, tvCode;

public HistoryViewHolder(@NonNull View itemView) {

super(itemView);

this.tvDateTime= itemView.findViewById(R.id.tvDateTime);

this.tvMoney= itemView.findViewById(R.id.tvMoney);

this.tvDescription= itemView.findViewById(R.id.tvDescription);

this.tvCode= itemView.findViewById(R.id.tvCode);

}

}

}

User.java

@Entity(tableName = "user")

public class User implements Serializable {

@PrimaryKey(autoGenerate = true)

@ColumnInfo(name = "id")

private int id;

@ColumnInfo(name = "name")

private String name;

@ColumnInfo(name = "username")

private String username;

@ColumnInfo(name = "PIN")

private String PIN;

@ColumnInfo(name = "balance")

private int balance;

@ColumnInfo(name = "status")

private int status;

public User(String name, String username, String PIN) {

this.name = name;

this.username = username;

this.PIN = PIN;

this.status= 1;

}

@Override

public String toString() { return this.username+" - "+this.PIN+" - " +this.status; }

public String getCurrencyFormat() {

DecimalFormat currency= (DecimalFormat) DecimalFormat.getCurrencyInstance();

DecimalFormatSymbols rupiah= new DecimalFormatSymbols();

rupiah.setCurrencySymbol("Rp ");

rupiah.setMonetaryDecimalSeparator(',');

rupiah.setGroupingSeparator('.');

currency.setDecimalFormatSymbols(rupiah);

return currency.format(this.balance);

}

public int getId() { return id; }

public String getName() { return name; }

public String getUsername() { return username; }

public String getPIN() { return PIN; }

public int getBalance() { return balance; }

public int getStatus() { return status; }

public void setId(int id) { this.id = id; }

public void setName(String name) { this.name = name; }

public void setUsername(String username) { this.username = username; }

public void setPIN(String PIN) { this.PIN = PIN; }

public void setBalance(int balance) { this.balance += balance; }

public void setStatus(int status) { this.status = status; }

}

Merchant.java

@Entity(tableName = "merchant")

public class Merchant {

@PrimaryKey(autoGenerate = true)

@ColumnInfo(name = "id")

private int id;

@ColumnInfo(name = "name")

private String name;

@ColumnInfo(name = "balance")

private int balance;

public Merchant(String name) {

this.name = name;

this.balance = balance;

}

@Override

public String toString() { return this.name; }

public int getId() { return id; }

public String getName() { return name; }

public int getBalance() { return balance; }

public void setId(int id) { this.id = id; }

public void setName(String name) { this.name = name; }

public void setBalance(int balance) { this.balance = balance; }

}

Transaction.java

@Entity(tableName = "trans")

public class Transaction {

@PrimaryKey(autoGenerate = true)

@ColumnInfo(name = "id")

private int id;

@ColumnInfo(name = "user")

private String user;

@ColumnInfo(name = "current")

private String current;

@ColumnInfo(name = "other")

private String other;

@ColumnInfo(name = "nominal")

private int nominal;

@ColumnInfo(name = "type")

private String type;

@ColumnInfo(name = "dateTime")

private String dateTime;

@ColumnInfo(name = "payment")

private boolean payment;

public Transaction(String user, String current, String other, int nominal, String type) {

this.user= user;

this.current= current;

this.other= other;

this.nominal= nominal;

this.type= type;

this.dateTime= this.getDateTimeFormat();

}

@Ignore

public Transaction(String user, String current, String other, int nominal, String type, boolean payment) {

this.user= user;

this.current= current;

this.other= other;

this.nominal= nominal;

this.type= type;

this.dateTime= this.getDateTimeFormat();

this.payment= payment;

}

@Override

public String toString() {

return this.current+" - "+this.other+" - "+this.nominal+" - "+this.type;

}

public String getCurrencyFormat() {

DecimalFormat currency= (DecimalFormat) DecimalFormat.getCurrencyInstance();

DecimalFormatSymbols rupiah= new DecimalFormatSymbols();

rupiah.setCurrencySymbol("Rp ");

rupiah.setMonetaryDecimalSeparator(',');

rupiah.setGroupingSeparator('.');

currency.setDecimalFormatSymbols(rupiah);

return currency.format(this.nominal);

}

private String getDateTimeFormat() {

Date date= Calendar.getInstance().getTime();

DateFormat formatter= new SimpleDateFormat("dd MMM, hh:mm");

return formatter.format(date);

}

public int getId() { return id; }

public String getUser() { return user; }

public String getCurrent() { return current; }

public String getOther() { return other; }

public int getNominal() { return nominal; }

public String getType() { return type; }

public String getDateTime() { return dateTime; }

public boolean isPayment() { return payment; }

public void setId(int id) { this.id = id; }

public void setUser(String user) { this.user = user; }

public void setCurrent(String current) { this.current = current; }

public void setOther(String other) { this.other = other; }

public void setNominal(int nominal) { this.nominal = nominal; }

public void setType(String type) { this.type = type; }

public void setDateTime(String dateTime) { this.dateTime = dateTime; }

public void setPayment(boolean payment) { this.payment = payment; }

}

UserDAO.java

@Dao

public interface UserDAO {

@Insert

void insert(User user);

@Update

void update(User user);

@Query("SELECT \* FROM user")

List<User> getAllUser();

}

Merchant.java

@Dao

public interface MerchantDAO {

@Insert

void insert(Merchant merchant);

@Delete

void delete(Merchant merchant);

@Query("SELECT \* FROM merchant")

List<Merchant> getAllMerchant();

}

Transaction.java

@Dao

public interface TransactionDAO {

@Insert

void insert(Transaction transaction);

@Delete

void delete(Transaction transaction);

@Query("SELECT \* FROM trans WHERE user = :username AND (type = 'in' OR type = 'out')")

List<Transaction> getAllTransactionFrom(String username);

@Query("SELECT \* FROM trans WHERE user = :username AND (type = 'in' OR type = 'request')")

List<Transaction> getAllFeedsTransactionFrom(String username);

@Query("SELECT \* FROM trans WHERE user = :username AND (current = :search OR other = :search)")

List<Transaction> getAllTransactionByUsername(String username, String search);

@Query("SELECT \* FROM trans WHERE user = :username AND type = :type")

List<Transaction> getAllTransactionByType(String username, String type);

@Query("SELECT \* FROM trans WHERE user = :username AND (current = :search OR other = :search) AND type = :type")

List<Transaction> getAllTransactionByTypeWithUsername(String username, String search, String type);

}

AppDatabase.java

@Database(entities = {User.class, Merchant.class, Transaction.class}, version = 1)

public abstract class AppDatabase extends RoomDatabase {

public abstract UserDAO userDAO();

public abstract MerchantDAO merchantDAO();

public abstract TransactionDAO transactionDAO();

}

activity\_admin.xml

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout 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"

tools:context=".AdminActivity">

<FrameLayout

android:id="@+id/frameContainer"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</FrameLayout>

<com.google.android.material.bottomnavigation.BottomNavigationView

android:id="@+id/bottomNav"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

app:menu="@menu/bottom\_navigation2"/>

</RelativeLayout>

activity\_loginregister.xml

<?xml version="1.0" encoding="utf-8"?>

<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"

tools:context=".LoginRegisterActivity">

<FrameLayout

android:id="@+id/frameContainer"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</FrameLayout>

</LinearLayout>

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout 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"

tools:context=".MainActivity">

<FrameLayout

android:id="@+id/frameContainer"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</FrameLayout>

<com.google.android.material.bottomnavigation.BottomNavigationView

android:id="@+id/bottomNav"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

app:menu="@menu/bottom\_navigation"/>

</RelativeLayout>

fragment\_history.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="10pt">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="8pt"

android:text="Jenis Transaksi:"/>

<RadioGroup

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:orientation="horizontal">

<RadioButton

android:id="@+id/rbIn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginRight="5pt"

android:textSize="6pt"

android:text="Uang Masuk"/>

<RadioButton

android:id="@+id/rbOut"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginRight="5pt"

android:textSize="6pt"

android:text="Uang Keluar"/>

<RadioButton

android:id="@+id/rbAll"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="6pt"

android:text="Semua"/>

</RadioGroup>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="8pt"

android:text="Pencarian bedasarkan username:"/>

<EditText

android:id="@+id/edtUsername"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="8pt"

android:hint="Username"/>

<Button

android:id="@+id/btnShow"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="8pt"

android:text="Tampilkan"/>

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/rvHistories"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt">

</androidx.recyclerview.widget.RecyclerView>

</LinearLayout>

fragment\_home.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="10pt">

<TextView

android:id="@+id/tvLogged"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="10pt"/>

<TextView

android:id="@+id/tvBalance"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<EditText

android:id="@+id/edtAmount"

android:inputType="number"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:layout\_marginRight="5pt"

android:textSize="10pt"

android:hint="Amount"/>

<Button

android:id="@+id/btnTopUp"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_weight="2"

android:textSize="10pt"

android:text="Top Up"/>

</LinearLayout>

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/rvFeeds"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt">

</androidx.recyclerview.widget.RecyclerView>

</LinearLayout>

fragment\_login.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="10pt">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:textSize="14pt"

android:text="Login"/>

<EditText

android:id="@+id/edtUsername"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"

android:hint="Username"/>

<EditText

android:id="@+id/edtPIN"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"

android:hint="PIN"/>

<Button

android:id="@+id/btnLogin"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"

android:text="Login"/>

</LinearLayout>

fragment\_mastermerchant.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="10pt">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="10pt"

android:text="Add Merchants"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:layout\_marginTop="5pt">

<EditText

android:id="@+id/edtName"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:textSize="10pt"

android:hint="Name"/>

<Button

android:id="@+id/btnAdd"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="10pt"

android:text="Add"/>

</LinearLayout>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="14pt"

android:text="List of Merchants"/>

<ListView

android:id="@+id/lvMerchant"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"></ListView>

</LinearLayout>

fragment\_masteruser.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="10pt">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="14pt"

android:text="List of Users"/>

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/rvUser"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt">

</androidx.recyclerview.widget.RecyclerView>

</LinearLayout>

fragment\_account.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="10pt">

<ImageView

android:layout\_width="match\_parent"

android:layout\_height="40pt"

android:src="@drawable/account"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="7pt"

android:text="Username:"/>

<TextView

android:id="@+id/tvUsername"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="7pt"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="7pt"

android:text="Your Name:"/>

<EditText

android:id="@+id/edtName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="7pt"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="7pt"

android:text="Change PIN:"/>

<EditText

android:id="@+id/edtNewPIN"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="5pt" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="7pt"

android:text="Old PIN:"/>

<EditText

android:id="@+id/edtOldPIN"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="7pt"/>

<Button

android:id="@+id/btnSave"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="7pt"

android:text="Save Changes"/>

</LinearLayout>

fragment\_payment.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="10pt">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="14pt"

android:text="List of Merchants"/>

<Spinner

android:id="@+id/spinMerchant"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt">

</Spinner>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"

android:text="Masukkan Nominal:"/>

<EditText

android:id="@+id/edtNominal"

android:inputType="number"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"/>

<Button

android:id="@+id/btnPay"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"

android:text="Pay"/>

</LinearLayout>

fragment\_register.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="10pt">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:textSize="14pt"

android:text="Register"/>

<EditText

android:id="@+id/edtUsername"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"

android:hint="Username"/>

<EditText

android:id="@+id/edtName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"

android:hint="Name"/>

<EditText

android:id="@+id/edtPIN"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"

android:hint="PIN"/>

<Button

android:id="@+id/btnRegister"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"

android:text="Register"/>

</LinearLayout>

fragment\_transfer.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="10pt">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="10pt"

android:text="Jenis Transaksi: "/>

<RadioGroup

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:orientation="horizontal">

<RadioButton

android:id="@+id/rbTransfer"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginRight="5pt"

android:textSize="8pt"

android:text="Transfer Dana"/>

<RadioButton

android:id="@+id/rbRequest"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="8pt"

android:text="Request Dana"/>

</RadioGroup>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"

android:text="Masukkan Nominal Dana:"/>

<EditText

android:id="@+id/edtNominal"

android:inputType="number"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"

android:text="Masukkan ID Tujuan:"/>

<EditText

android:id="@+id/edtDestination"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt"

android:textSize="10pt"/>

<Button

android:id="@+id/btnSubmit"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10pt"

android:textSize="10pt"

android:text="Submit"/>

</LinearLayout>

layout\_feeds.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/linearContainer"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:id="@+id/tvDescription"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="8pt"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:id="@+id/tvDateTime"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:textSize="8pt"/>

<Button

android:id="@+id/btnSend"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="8pt"

android:text="Send Money"/>

</LinearLayout>

</LinearLayout>

layout\_history.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10pt">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/tvDateTime"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:textSize="8pt"/>

<TextView

android:id="@+id/tvMoney"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:gravity="right"

android:textSize="8pt"/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:layout\_marginTop="5pt">

<TextView

android:id="@+id/tvDescription"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:textSize="8pt"/>

<TextView

android:id="@+id/tvCode"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:gravity="right"

android:textSize="8pt"/>

</LinearLayout>

</LinearLayout>

layout\_user.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="horizontal"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="5pt">

<ImageView

android:id="@+id/ivUser"

android:layout\_width="wrap\_content"

android:layout\_height="40pt"

android:layout\_weight="1"

android:src="@drawable/account"/>

<TextView

android:id="@+id/tvName"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:gravity="center"

android:textSize="10pt"/>

<Button

android:id="@+id/btnActive"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:textSize="10pt"/>

</LinearLayout>

Bottom\_navigation.xml

<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:android="http://schemas.android.com/apk/res/android">

<item

android:id="@+id/menuHome"

android:title="Home"

android:icon="@drawable/home"/>

<item

android:id="@+id/menuHistory"

android:title="History"

android:icon="@drawable/history"/>

<item

android:id="@+id/menuPayment"

android:title="Payment"

android:icon="@drawable/payment"/>

<item

android:id="@+id/menuTransfer"

android:title="Transfer"

android:icon="@drawable/transfer"/>

<item

android:id="@+id/menuMyAccount"

android:title="My Account"

android:icon="@drawable/account"/>

</menu>

bottom\_navigation2.xml

<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:android="http://schemas.android.com/apk/res/android">

<item

android:id="@+id/menuMasterUser"

android:title="Master User"

android:icon="@drawable/user"/>

<item

android:id="@+id/menuMasterMerchant"

android:title="Master Merchant"

android:icon="@drawable/merchant"/>

</menu>

context\_menu.xml

<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:android="http://schemas.android.com/apk/res/android">

<item

android:id="@+id/menuDelete"

android:title="Delete"/>

</menu>

options\_menu.xml

<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:android="http://schemas.android.com/apk/res/android">

<item

android:id="@+id/menuLogin"

android:title="Login"/>

<item

android:id="@+id/menuRegister"

android:title="Register"/>

<item

android:id="@+id/menuLogout"

android:title="Logout"/>

</menu>