```java
package com.marktrs.macapp.Fragment.Recruiter;


import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.Toast;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.marktrs.macapp.Fragment.WorkerSetUpFragment;
import com.marktrs.macapp.Model.Job;
import com.marktrs.macapp.R;

/**
 * A simple {@link Fragment} subclass.
 */
public class AddNewJobFragment extends Fragment {


    private Button confirmButton;
    private EditText jobNameET;
    private EditText symptomTypeET;
    private EditText requiredEducationET;
    private EditText requiredSkillET;
    private EditText workplaceET;
    private EditText paymentET;

    private DatabaseReference mDatabase;
    private FirebaseAuth mFirebaseAuth;
    private FirebaseUser mFirebaseUser;

    private ImageButton fab;

    public AddNewJobFragment() {
        // Required empty public constructor
    }


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        mDatabase = FirebaseDatabase.getInstance().getReference();
        mFirebaseAuth = FirebaseAuth.getInstance();
        mFirebaseUser = mFirebaseAuth.getCurrentUser();
        return inflater.inflate(R.layout.fragment_add_new_job, container, false);
    }

    @Override
    public void onViewCreated(View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        jobNameET = (EditText) view.findViewById(R.id.job_name);
        symptomTypeET = (EditText) view.findViewById(R.id .job_symptoms);
        requiredEducationET = (EditText) view.findViewById(R.id.job_education);
        requiredSkillET = (EditText) view.findViewById(R.id.job_skill);
        workplaceET = (EditText) view.findViewById(R.id.job_workplace);
        paymentET = (EditText) view.findViewById(R.id.job_payment);

        confirmButton = (Button) view.findViewById(R.id.job_submit_button);
        confirmButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                addJobToFirebase();
                PostedJobFragment postedJobFragment = new PostedJobFragment();
                FragmentManager manager = getFragmentManager();
```

```
77                  FragmentTransaction transaction = manager.beginTransaction();
78                  transaction.remove(AddNewJobFragment.this);
79                  transaction.replace(R.id.fragment_area, postedJobFragment);
80                  transaction.commit();
81                  Toast.makeText(getContext(), "Successful !",
82                          Toast.LENGTH_SHORT).show();
83              }
84          });
85      }
86
87      public void addJobToFirebase() {
88          String key = mDatabase.child("Jobs").push().getKey();
89
90          Job job = new Job();
91          job.setJobId(key);
92          job.setJobName(jobNameET.getText().toString());
93          job.setSymptomType(symptomTypeET.getText().toString());
94          job.setRequiredEducation(requiredEducationET.getText().toString());
95          job.setRequiredSkill(requiredSkillET.getText().toString());
96          job.setWorkplace(workplaceET.getText().toString());
97          job.setPayment(paymentET.getText().toString());
98          job.setOwnerUID(mFirebaseUser.getUid());
99
100         mDatabase.child("Jobs").child(key).setValue(job);
101     }
102
103     @Override
104     public void onDetach() {
105         super.onDetach();
106         fab = (ImageButton) getActivity().findViewById(R.id.fab);
107         fab.setVisibility(View.VISIBLE);
108     }
109 }
110
```

```java
package com.marktrs.macapp.Fragment.Recruiter;

import android.content.Context;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v7.widget.GridLayoutManager;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageButton;
import android.widget.TextView;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.marktrs.macapp.Model.Job;
import com.marktrs.macapp.Model.JobApplication;
import com.marktrs.macapp.R;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

public class PostedJobFragment extends Fragment {

    // TODO: Customize parameter argument names
    private static final String ARG_COLUMN_COUNT = "column-count";
    // TODO: Customize parameters
    private int mColumnCount = 2;

    private OnPostedFragmentInteractionListener mListener;

    private FirebaseDatabase database;
    private DatabaseReference jobsRef;
    private DatabaseReference jobApplicationRef;

    private ValueEventListener getAllJobListener;
    private ValueEventListener getAllApplicationListener;

    private ArrayList<Job> jobs;

    private FirebaseAuth mFirebaseAuth;
    private FirebaseUser mFirebaseUser;

    private RecyclerView recyclerView;
    private Map<String, Integer> jobApplicationCount;
    private TextView noContentText;
    private ImageButton fab;

    /**
     * Mandatory empty constructor for the fragment manager to instantiate the
     * fragment (e.g. upon screen orientation changes).
     */
    public PostedJobFragment() {
    }

    // TODO: Customize parameter initialization
    @SuppressWarnings("unused")
    public static PostedJobFragment newInstance(int columnCount) {
        PostedJobFragment fragment = new PostedJobFragment();
        Bundle args = new Bundle();
        args.putInt(ARG_COLUMN_COUNT, columnCount);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
77
78          database = FirebaseDatabase.getInstance();
79          mFirebaseAuth = FirebaseAuth.getInstance();
80          mFirebaseUser = mFirebaseAuth.getCurrentUser();
81
82          if (getArguments() != null) {
83              mColumnCount = getArguments().getInt(ARG_COLUMN_COUNT);
84          }
85      }
86
87      @Override
88      public View onCreateView(LayoutInflater inflater, ViewGroup container,
89                               Bundle savedInstanceState) {
90          View view = inflater.inflate(R.layout.fragment_postedjob_list, container, false);
91
92          // Set the adapter
93          if (view instanceof RecyclerView) {
94              Context context = view.getContext();
95              recyclerView = (RecyclerView) view;
96              if (mColumnCount <= 1) {
97                  recyclerView.setLayoutManager(new LinearLayoutManager(context));
98              } else {
99                  recyclerView.setLayoutManager(new GridLayoutManager(context, mColumnCount));
100             }
101             //TODO: Add posted job here
102
103
104             jobsRef = database.getReference("Jobs");
105             jobApplicationRef = database.getReference("JobApplications");
106
107             mFirebaseUser = mFirebaseAuth.getCurrentUser();
108
109             getAllApplicationListener = new ValueEventListener() {
110                 @Override
111                 public void onDataChange(DataSnapshot dataSnapshot) {
112                     jobApplicationCount = new HashMap<>();
113                     if (dataSnapshot.getChildrenCount() > 0) {
114                         for (DataSnapshot dtSnapshot : dataSnapshot.getChildren()) {
115                             JobApplication application = dtSnapshot.getValue(JobApplication.class);
116                             String id = application.getJobId();
117                             if (jobApplicationCount.containsKey(id)) {
118                                 jobApplicationCount.put(id, jobApplicationCount.get(id) + 1);
119                             } else {
120                                 jobApplicationCount.put(id, 1);
121                             }
122                         }
123                         jobsRef.addListenerForSingleValueEvent(getAllJobListener);
124                     }
125                 }
126
127                 @Override
128                 public void onCancelled(DatabaseError databaseError) {
129
130                 }
131             };
132
133             getAllJobListener = new ValueEventListener() {
134                 @Override
135                 public void onDataChange(DataSnapshot dataSnapshot) {
136                     jobs = new ArrayList<>();
137                     for (DataSnapshot dtSnapshot : dataSnapshot.getChildren()) {
138                         Job job = dtSnapshot.getValue(Job.class);
139                         if (job.getOwnerUID().equals(mFirebaseUser.getUid())) {
140                             jobs.add(job);
141                         }
142                     }
143                     if (jobs.isEmpty()) {
144                         if (jobs.isEmpty()){
145                             noContentText = (TextView) getActivity().findViewById(R.id.no_content);
146                             noContentText.setText("You never announce any job \n\n Try to tap (+) button to add new
    job");
147                             noContentText.setVisibility(View.VISIBLE);
148                         }
149                     } else {
150                         recyclerView.setAdapter(new MyPostedJobRecyclerViewAdapter(jobs, jobApplicationCount,
    mListener));
```

```
151                     }
152                 }
153
154                 @Override
155                 public void onCancelled(DatabaseError databaseError) {
156
157                 }
158             };
159
160             jobApplicationRef.addValueEventListener(getAllApplicationListener);
161
162         }
163         return view;
164     }
165
166     @Override
167     public void onAttach(Context context) {
168         super.onAttach(context);
169         if (context instanceof OnPostedFragmentInteractionListener) {
170             mListener = (OnPostedFragmentInteractionListener) context;
171         } else {
172             throw new RuntimeException(context.toString()
173                     + " must implement OnListFragmentInteractionListener");
174         }
175     }
176
177     @Override
178     public void onDetach() {
179         super.onDetach();
180         jobsRef.removeEventListener(getAllJobListener);
181     }
182
183     public interface OnPostedFragmentInteractionListener{
184         void onPressPostedJob(Job job);
185     }
186
187 }
188
```

```java
package com.marktrs.macapp.Fragment.Recruiter;

import android.content.Context;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v7.widget.GridLayoutManager;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.Query;
import com.google.firebase.database.ValueEventListener;
import com.marktrs.macapp.Model.Job;
import com.marktrs.macapp.Model.JobApplication;
import com.marktrs.macapp.Model.User;
import com.marktrs.macapp.R;
import com.marktrs.macapp.Fragment.Recruiter.dummy.DummyContent;
import com.marktrs.macapp.Fragment.Recruiter.dummy.DummyContent.DummyItem;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

/**
 * A fragment representing a list of Items.
 * <p/>
 * Activities containing this fragment MUST implement the {@link OnListFragmentInteractionListener}
 * interface.
 */
public class JobApplicationAchiveFragment extends Fragment {

    // TODO: Customize parameter argument names
    private static final String ARG_COLUMN_COUNT = "column-count";
    private static final String JOB_KEY = "jobKey";
    // TODO: Customize parameters
    private int mColumnCount = 1;
    private OnListFragmentInteractionListener mListener;

    private FirebaseDatabase database;
    private DatabaseReference applicationRef;
    private ValueEventListener getApplication;
    private ArrayList<JobApplication> jobApplications;
    private ArrayList<User> profileMap;

    private RecyclerView recyclerView;

    private TextView noContentText;

    private Job job;

    /**
     * Mandatory empty constructor for the fragment manager to instantiate the
     * fragment (e.g. upon screen orientation changes).
     */
    public JobApplicationAchiveFragment() {
    }

    // TODO: Customize parameter initialization
    @SuppressWarnings("unused")
    public static JobApplicationAchiveFragment newInstance(Job job) {
        JobApplicationAchiveFragment fragment = new JobApplicationAchiveFragment();
        Bundle args = new Bundle();
        args.putSerializable(JOB_KEY, job);
        fragment.setArguments(args);
        return fragment;
    }

```

```
 77     @Override
 78     public void onCreate(Bundle savedInstanceState) {
 79         super.onCreate(savedInstanceState);
 80
 81         database = FirebaseDatabase.getInstance();
 82         this.job = (Job) getArguments().getSerializable(JOB_KEY);
 83
 84         if (getArguments() != null) {
 85             mColumnCount = getArguments().getInt(ARG_COLUMN_COUNT);
 86         }
 87     }
 88
 89     @Override
 90     public View onCreateView(LayoutInflater inflater, ViewGroup container,
 91                              Bundle savedInstanceState) {
 92         View view = inflater.inflate(R.layout.fragment_jobapplicationachive_list, container, false);
 93
 94         // Set the adapter
 95         if (view instanceof RecyclerView) {
 96             Context context = view.getContext();
 97             recyclerView = (RecyclerView) view;
 98             if (mColumnCount <= 1) {
 99                 recyclerView.setLayoutManager(new LinearLayoutManager(context));
100             } else {
101                 recyclerView.setLayoutManager(new GridLayoutManager(context, mColumnCount));
102             }
103
104             applicationRef = database.getReference("JobApplications");
105             profileMap = new ArrayList<>();
106
107             getApplication = new ValueEventListener() {
108                 @Override
109                 public void onDataChange(DataSnapshot dataSnapshot) {
110                     jobApplications = new ArrayList<>();
111                     if (dataSnapshot.getChildrenCount() > 0) {
112                         for (DataSnapshot snapshot : dataSnapshot.getChildren()) {
113                             JobApplication application = snapshot.getValue(JobApplication.class);
114                             if (application.getJobId().equals(job.getJobId())) {
115                                 jobApplications.add(application);
116                             }
117                         }
118                         recyclerView.setAdapter(new MyJobApplicationAchiveRecyclerViewAdapter(jobApplications,
    mListener));
119                     }
120                     if (jobApplications.isEmpty()){
121                         noContentText = (TextView) getActivity().findViewById(R.id.no_content);
122                         noContentText.setText("There are no application on this time");
123                         noContentText.setVisibility(View.VISIBLE);
124                     }
125                 }
126
127                 @Override
128                 public void onCancelled(DatabaseError databaseError) {
129
130                 }
131             };
132
133             applicationRef.addValueEventListener(getApplication);
134         }
135         return view;
136     }
137
138     @Override
139     public void onAttach(Context context) {
140         super.onAttach(context);
141         if (context instanceof OnListFragmentInteractionListener) {
142             mListener = (OnListFragmentInteractionListener) context;
143         } else {
144             throw new RuntimeException(context.toString()
145                     + " must implement OnListFragmentInteractionListener");
146         }
147     }
148
149     @Override
150     public void onDetach() {
151         super.onDetach();
```

```
152         mListener = null;
153         applicationRef.removeEventListener(getApplication);
154         noContentText.setVisibility(View.GONE);
155         noContentText.setText("");
156     }
157
158     /**
159      * This interface must be implemented by activities that contain this
160      * fragment to allow an interaction in this fragment to be communicated
161      * to the activity and potentially other fragments contained in that
162      * activity.
163      * <p/>
164      * See the Android Training lesson <a href=
165      * "http://developer.android.com/training/basics/fragments/communicating.html"
166      * >Communicating with Other Fragments</a> for more information.
167      */
168     public interface OnListFragmentInteractionListener {
169         // TODO: Update argument type and name
170         void onListFragmentInteraction(JobApplication jobApplication);
171     }
172 }
173
```

```java
package com.marktrs.macapp.Fragment.Recruiter;

import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import android.widget.Toast;
import com.marktrs.macapp.Model.Job;
import com.marktrs.macapp.R;

import java.util.ArrayList;
import java.util.Map;

public class MyPostedJobRecyclerViewAdapter extends RecyclerView.Adapter<MyPostedJobRecyclerViewAdapter.ViewHolder> {

    private ArrayList<Job> jobs;
    private Map<String, Integer> jobApplicationCount;
    private PostedJobFragment.OnPostedFragmentInteractionListener mListener;

    public MyPostedJobRecyclerViewAdapter(ArrayList<Job> jobs, Map<String, Integer> jobApplicationCount,
PostedJobFragment.OnPostedFragmentInteractionListener listener) {
        this.jobs = jobs;
        this.jobApplicationCount = jobApplicationCount;
        this.mListener = listener;
    }

    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.fragment_postedjob, parent, false);
        return new ViewHolder(view);
    }

    @Override
    public void onBindViewHolder(final ViewHolder holder, int position) {
        holder.mItem = jobs.get(position);
        holder.mIdView.setText(jobs.get(position).getJobName());
        holder.mContentView.setText(jobs.get(position).getWorkplace());
        holder.jobSymptom.setText(jobs.get(position).getSymptomType());
        if(null != jobApplicationCount.get(jobs.get(position).getJobId())){
            holder.applicationCounter.setText(jobApplicationCount.get(jobs.get(position).getJobId()).toString());
        }else {
            holder.applicationCounter.setText("0");
        }
        holder.mView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (null != mListener) {
                    mListener.onPressPostedJob(holder.mItem);
                }

            }
        });
    }

    @Override
    public int getItemCount() {
        return jobs.size();
    }

    public class ViewHolder extends RecyclerView.ViewHolder {
        public final View mView;
        public final TextView mIdView;
        public final TextView mContentView;
        public final TextView jobSymptom;
        public final TextView applicationCounter;
        public Job mItem;

        public ViewHolder(View view) {
            super(view);
            mView = view;
            mIdView = (TextView) view.findViewById(R.id.id);
            mContentView = (TextView) view.findViewById(R.id.content);
            jobSymptom = (TextView) view.findViewById(R.id.job_symptom);
            applicationCounter = (TextView) view.findViewById(R.id.application_count);
```

```
76          }
77
78          @Override
79          public String toString() {
80              return super.toString() + " '" + mContentView.getText() + "'";
81          }
82      }
83  }
84
```

```java
package com.marktrs.macapp.Fragment.Recruiter;

import android.content.res.ColorStateList;
import android.graphics.Color;
import android.graphics.Paint;
import android.support.v7.widget.RecyclerView;
import android.text.Layout;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.marktrs.macapp.Fragment.Recruiter.JobApplicationAchiveFragment.OnListFragmentInteractionListener;
import com.marktrs.macapp.Fragment.Recruiter.dummy.DummyContent.DummyItem;
import com.marktrs.macapp.Model.JobApplication;
import com.marktrs.macapp.Model.User;
import com.marktrs.macapp.R;

import java.util.ArrayList;

/**
 * {@link RecyclerView.Adapter} that can display a {@link DummyItem} and makes a call to the
 * specified {@link OnListFragmentInteractionListener}.
 * TODO: Replace the implementation with code for your data type.
 */
public class MyJobApplicationAchiveRecyclerViewAdapter extends RecyclerView.Adapter<
MyJobApplicationAchiveRecyclerViewAdapter.ViewHolder> {

    private final ArrayList<JobApplication> mValues;
    private final OnListFragmentInteractionListener mListener;
    private final String ACCEPTED = "Accepted";
    private final String DENIED = "Denied";
    private final String WAITING = "waiting";
    DatabaseReference profileRef;
    FirebaseDatabase database;
    ValueEventListener getProfileListener;

    public MyJobApplicationAchiveRecyclerViewAdapter(ArrayList<JobApplication> items, OnListFragmentInteractionListener
 listener) {
        mValues = items;
        mListener = listener;
    }

    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.fragment_jobapplicationachive, parent, false);
        return new ViewHolder(view);
    }

    @Override
    public void onBindViewHolder(final ViewHolder holder, int position) {
        holder.mItem = mValues.get(position);
        database = FirebaseDatabase.getInstance();
        profileRef = database.getReference("User").child(mValues.get(position).getWorkerId());
        if (!holder.mItem.getStatus().equals(DENIED)) {
            if (holder.mItem.getStatus().equals(ACCEPTED)) {
                holder.lineView.setVisibility(View.VISIBLE);
                holder.facebookView.setVisibility(View.VISIBLE);
                holder.buttonPanel.setVisibility(View.GONE);
                getProfileListener = new ValueEventListener() {
                    @Override
                    public void onDataChange(DataSnapshot dataSnapshot) {
                        User user = dataSnapshot.getValue(User.class);
                        holder.workerName.setText(user.getFirstName() + " " + user.getLastName());
                        holder.workerName.setTextColor(Color.parseColor("#4CAF50"));
                        holder.workerLocation.setText(user.getWorker().getLocation());
                        holder.workerSymptom.setText(user.getWorker().getSymptom());
                        holder.workerSkill.setText(user.getWorker().getSkill());
```

```
75                          holder.workerEducation.setText(user.getWorker().getEducaton());
76                          holder.workerLine.setText(user.getWorker().getLineId());
77                          holder.workerFacebook.setText(user.getWorker().getFacebookId());
78                      }
79
80                      @Override
81                      public void onCancelled(DatabaseError databaseError) {
82
83                      }
84                  };
85
86              } else if (holder.mItem.getStatus().equals(WAITING)) {
87                  holder.buttonPanel.setVisibility(View.VISIBLE);
88                  getProfileListener = new ValueEventListener() {
89                      @Override
90                      public void onDataChange(DataSnapshot dataSnapshot) {
91                          User user = dataSnapshot.getValue(User.class);
92                          holder.workerName.setText(user.getFirstName() + " " + user.getLastName());
93                          holder.workerLocation.setText(user.getWorker().getLocation());
94                          holder.workerSymptom.setText(user.getWorker().getSymptom());
95                          holder.workerSkill.setText(user.getWorker().getSkill());
96                          holder.workerEducation.setText(user.getWorker().getEducaton());
97                          holder.workerLine.setText(user.getWorker().getLineId());
98                          holder.workerFacebook.setText(user.getWorker().getFacebookId());
99
100                         holder.denyButton.setOnClickListener(new View.OnClickListener() {
101                             @Override
102                             public void onClick(View view) {
103                                 holder.workerName.setText(DENIED);
104                                 holder.workerName.setTextColor(Color.parseColor("#BDBDBD"));
105                                 holder.workerName.setPaintFlags(holder.workerName.getPaintFlags() | Paint.
    STRIKE_THRU_TEXT_FLAG);
106                                 holder.buttonPanel.setVisibility(View.GONE);
107                                 if (null != mListener) {
108                                     holder.mItem.setStatus(DENIED);
109                                     mListener.onListFragmentInteraction(holder.mItem);
110                                 }
111                             }
112                         });
113
114                         holder.acceptButton.setOnClickListener(new View.OnClickListener() {
115                             @Override
116                             public void onClick(View view) {
117                                 holder.lineView.setVisibility(View.VISIBLE);
118                                 holder.facebookView.setVisibility(View.VISIBLE);
119                                 holder.acceptButton.setVisibility(View.GONE);
120                                 holder.denyButton.setVisibility(View.GONE);
121                                 if (null != mListener) {
122                                     holder.mItem.setStatus(ACCEPTED);
123                                     mListener.onListFragmentInteraction(holder.mItem);
124                                 }
125
126                             }
127                         });
128                     }
129
130                     @Override
131                     public void onCancelled(DatabaseError databaseError) {
132
133                     }
134                 };
135             }
136             profileRef.addListenerForSingleValueEvent(getProfileListener);
137         }else {
138             holder.workerName.setTextColor(Color.parseColor("#BDBDBD"));
139             holder.workerName.setPaintFlags(holder.workerName.getPaintFlags() | Paint.STRIKE_THRU_TEXT_FLAG);
140         }
141     }
142
143     @Override
144     public int getItemCount() {
145         return mValues.size();
146     }
147
148     public class ViewHolder extends RecyclerView.ViewHolder {
149         public final View mView;
```

```
150         public final TextView workerName;
151         public final TextView workerLocation;
152         public TextView workerSkill;
153         public TextView workerEducation;
154         public TextView workerSymptom;
155         public TextView workerLine;
156         public TextView workerFacebook;
157         public Button acceptButton;
158         public Button denyButton;
159         public View lineView;
160         public View facebookView;
161         public JobApplication mItem;
162         public View buttonPanel;
163 
164         public ViewHolder(View view) {
165             super(view);
166             mView = view;
167             workerName = (TextView) view.findViewById(R.id.worker_name);
168             workerLocation = (TextView) view.findViewById(R.id.worker_location);
169             workerSymptom = (TextView) view.findViewById(R.id.worker_symptoms);
170             workerSkill = (TextView) view.findViewById(R.id.worker_skill);
171             workerEducation = (TextView) view.findViewById(R.id.worker_education);
172             acceptButton = (Button) view.findViewById(R.id.accept_button);
173             denyButton = (Button) view.findViewById(R.id.deny_button);
174             lineView = view.findViewById(R.id.line_view);
175             facebookView = view.findViewById(R.id.facebook_view);
176             workerLine = (TextView) view.findViewById(R.id.worker_line_id);
177             workerFacebook = (TextView) view.findViewById(R.id.worker_facebook_id);
178             buttonPanel = view.findViewById(R.id.button_panel);
179         }
180     }
181 }
182 
```