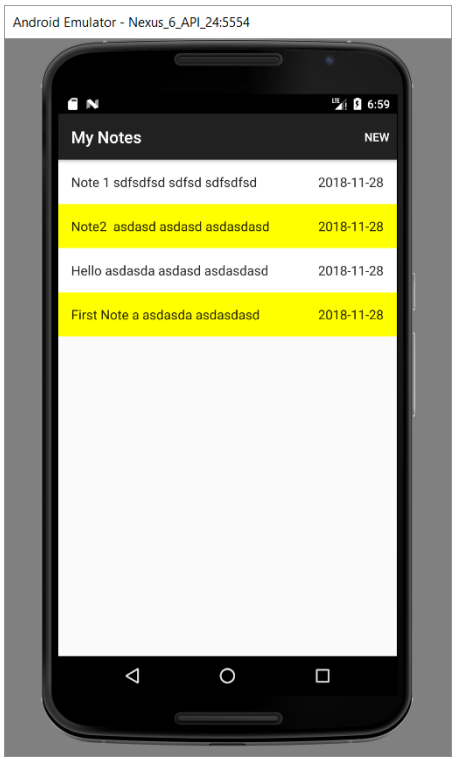
Lab 10: My Notes

In this lab we will be implementing the below application which provides note taking.



1. Create new project in Android Studio with an Empty Activity.

2. Select defaults for minimum SDK
3. Create a Cloud Firestore database as described in the lectures
4. Register your application and add the configuration file to your project
5. Update gradle files and as described in lecture notes
6. Add the firestore library to your project.

7. Open the activity_main.xml and modify it as shown below.

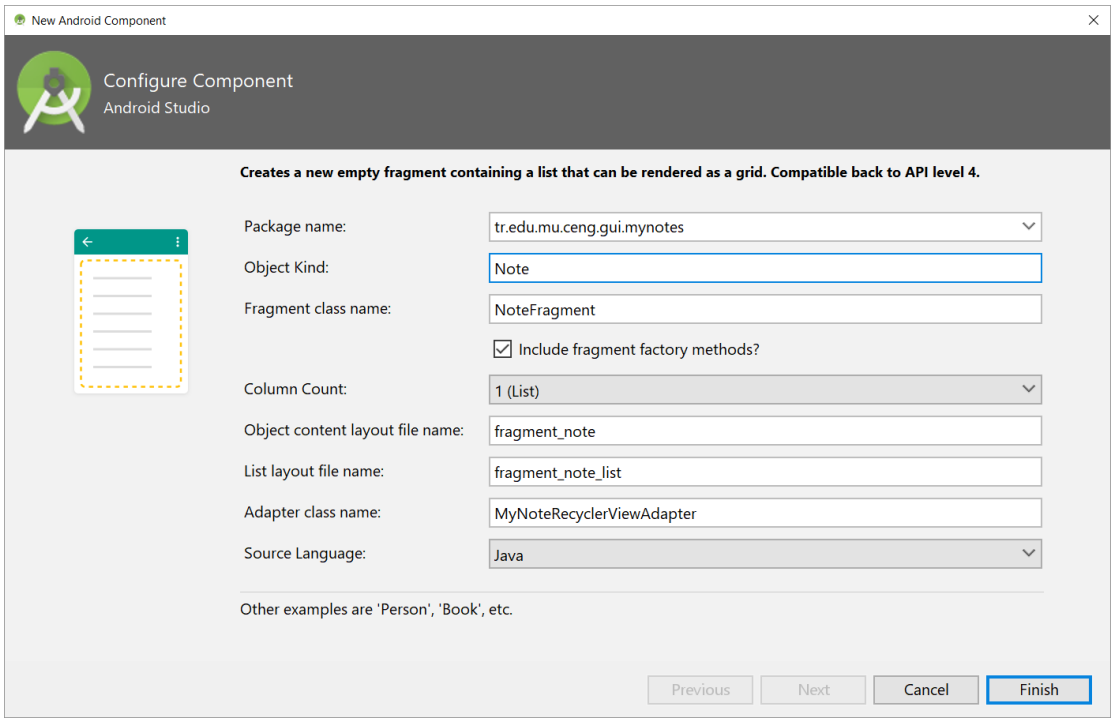
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
android:layout_height="match_parent" tools:context=".MainActivity">
<FrameLayout
    android:id="@+id/container"
    android:layout_width="0dp"
    android:layout_height="0dp"
    app:layout_constraintBottom_toBottomOf="parent"
```

```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">


</FrameLayout>
</android.support.constraint.ConstraintLayout>
```

8. Create the Note class shown below and generate getters and setters

```
public class Note {

    @Exclude
    private String id;
    private Timestamp date;
    private String content;
}
```

9. Create a fragment for listing notes based on Fragment (List) as shown below



10. Modify the fragment_note as shown below

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">


    <TextView
        android:id="@+id/note_header"
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_margin="@dimen/text_margin"
        android:textAppearance="?android:attr/textAppearanceListItem" />

    <TextView
        android:id="@+id/note_date"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="@dimen/text_margin"
        android:textAppearance="?android:attr/textAppearanceListItem" />
</LinearLayout>
```

11. Modify the NoteFragment as shown below

```
public class NoteFragment extends Fragment {
    private OnNoteListInteractionListener mListener;
    RecyclerView recyclerView;
    public NoteFragment() {}
    public static NoteFragment newInstance() {
        NoteFragment fragment = new NoteFragment();
        return fragment;
    }
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_note_list, container, false);
        if (view instanceof RecyclerView) {
            recyclerView = (RecyclerView) view;
        }
        Log.d("Fragment" , "OnCreateView");
        return view;
    }
    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        if (context instanceof OnNoteListInteractionListener) {
            mListener = (OnNoteListInteractionListener) context;
        } else {
            throw new RuntimeException(context.toString()
                    + " must implement OnNoteListInteractionListener");
        }
    }
    @Override
    public void onDetach() {
```

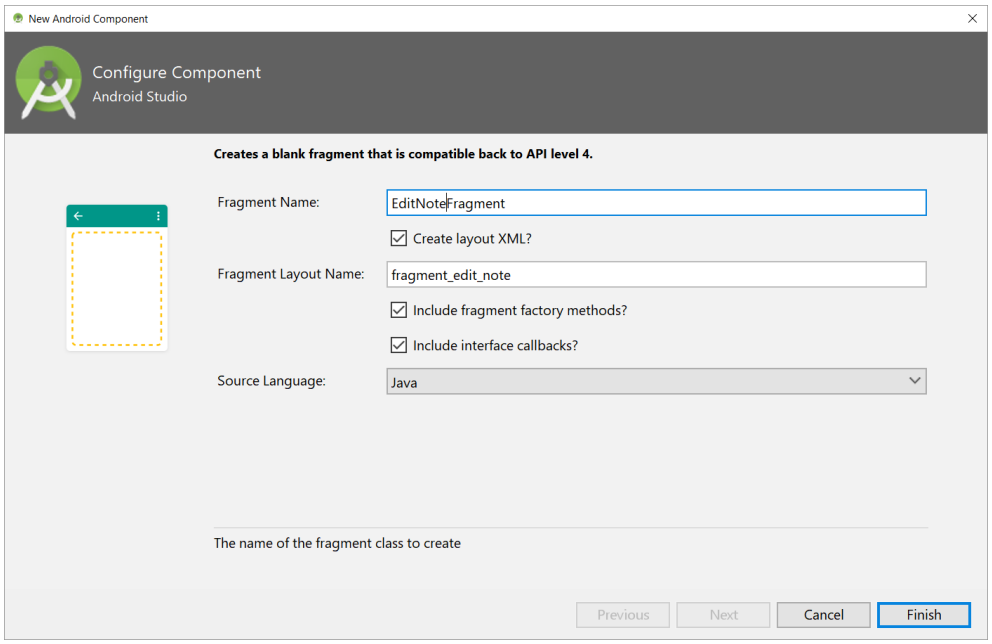
```
        super.onDetach();
        mListener = null;
    }
    /**
     * Interface for listing note operations in the list
     */
    public interface OnNoteListInteractionListener {
        void onNoteSelected(Note item);
    }
    public void updateNotes(List<Note> notes){
        Log.d("Fragment" , "updateNotes");
        recyclerView.setLayoutManager(new
LinearLayoutManager(recyclerView.getContext()));
        recyclerView.setAdapter(new MyNoteRecyclerViewAdapter(notes, mListener));
    }
}
```

12. Modify the Adapter as shown below

```
public class MyNoteRecyclerViewAdapter extends
RecyclerView.Adapter<MyNoteRecyclerViewAdapter.ViewHolder> {
    private final List<Note> mValues;
    private final NoteFragment.OnNoteListInteractionListener mListener;
    public MyNoteRecyclerViewAdapter(List<Note> notes,
NoteFragment.OnNoteListInteractionListener listener) {
        mValues = notes;
        mListener = listener;
    }
    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.fragment_note, parent, false);
        return new ViewHolder(view);
    }
    @Override
    public void onBindViewHolder(final ViewHolder holder, int position) {
        holder.mItem = mValues.get(position);
        String content = mValues.get(position).getContent();
        String header = content.length() < 30 ? content : content.substring(0,30);
        holder.mHeaderView.setText(header.replaceAll("\n", " "));
        holder.mDateView.setText((new SimpleDateFormat("yyyy-MM-dd")).
                format(mValues.get(position).getDate().toDate()));
        holder.mView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

```
            if (null != mListener) {
                mListener.onNoteSelected(holder.mItem);
            }
        }
    });
    if(position %2 == 1) {
        holder.itemView.setBackgroundColor(Color.YELLOW);
    }
    else{
        holder.itemView.setBackgroundColor(Color.WHITE);
    }
}
@Override
public int getItemCount() {
    return mValues.size();
}
public class ViewHolder extends RecyclerView.ViewHolder {
    public final View mView;
    public final TextView mHeaderView;
    public final TextView mDateView;
    public Note mItem;
    public ViewHolder(View view) {
        super(view);
        mView = view;
        mHeaderView =  view.findViewById(R.id.note_header);
        mDateView =  view.findViewById(R.id.note_date);
    }
    @Override
    public String toString() {
        return super.toString() + " '" + mHeaderView.getText() + "'";
    }
}
}
```

Create an EditNote fragment based on the Blank fragment template



13. Modify the EditNoteFragment as shown below.

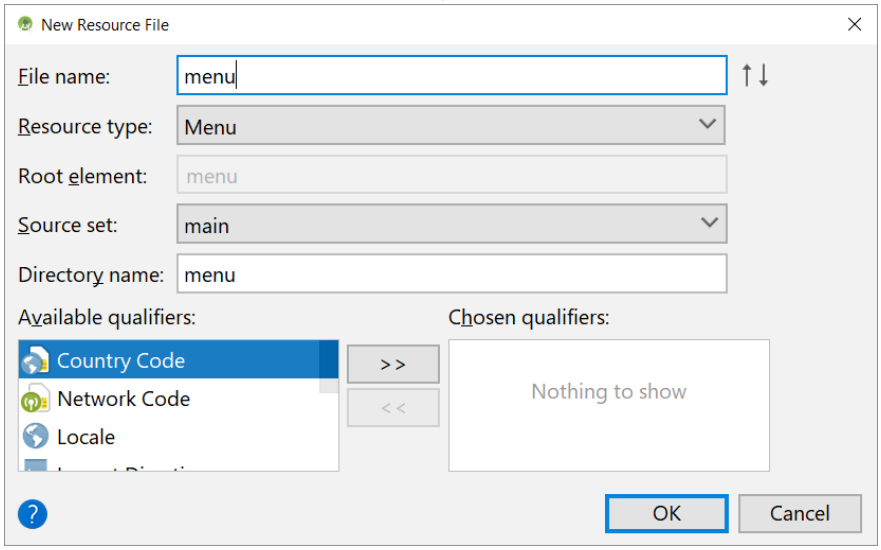
```
public class EditNoteFragment extends Fragment {


    private static final String ARG_NOTE = "content";
    private String content;
    private EditText txtContent;
    public EditNoteFragment() {}
    public static EditNoteFragment newInstance(String content) {
        EditNoteFragment fragment = new EditNoteFragment();
        Bundle args = new Bundle();
        args.putString(ARG_NOTE, content);
        fragment.setArguments(args);
        return fragment;
    }
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            content = getArguments().getString(ARG_NOTE);
        }
    }
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_edit_note, container, false);
    }
```

```
    @Override
    public void onViewCreated(View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        txtContent = view.findViewById(R.id.note_content);
        if (content != null) {
            txtContent.setText(content);
        }
    }
    public String getContent(){
        return txtContent.getText().toString();
    }
}
```

14. Modify the fragment_edit_note.xml as shown below

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".EditNoteFragment">

    <EditText
        android:id="@+id/note_content"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:inputType="textMultiLine"
        android:gravity="top"
        android:hint="Note..." />
</FrameLayout>
```

15. Create a new menu for the Activity



16. Modify the menu.xml as shown below

```
<?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/action_new"
        android:title="New"
        app:showAsAction="always" />
    <item
        android:id="@+id/action_close"
        app:showAsAction="always"
        android:title="Close"
        android:visible="false" />

</menu>
```

17. Modify the MainActivity as below

```
public class MainActivity extends AppCompatActivity implements
NoteFragment.OnNoteListInteractionListener {
   private static final String TAG = "Firebase Demo";
   boolean displayingEditor = false;
   Note editingNote;
   ListenerRegistration listenerRegistration;
   ArrayList<Note> notes = new ArrayList<>();
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);
       if (!displayingEditor){
           FragmentTransaction ft = getSupportFragmentManager().beginTransaction();
           ft.add(R.id.container,NoteFragment.newInstance(),"list_note");
           ft.commit();
       }else{
           FragmentTransaction ft = getSupportFragmentManager().beginTransaction();

ft.replace(R.id.container,EditNoteFragment.newInstance(editingNote.getContent()));
           ft.addToBackStack(null);
           ft.commit();
       }
       FirebaseFirestore db = FirebaseFirestore.getInstance();
       listenerRegistration = db.collection("notes").orderBy("date",
```

```
Query.Direction.DESCENDING).addSnapshotListener(new EventListener<QuerySnapshot>() {
          @Override
          public void onEvent(@Nullable QuerySnapshot queryDocumentSnapshots, @Nullable
FirebaseFirestoreException e) {

              if (e != null) {
                  Log.e(TAG, "Error retrieving notes", e);
                  return;
              }
              notes.clear();
              for (QueryDocumentSnapshot doc : queryDocumentSnapshots) {

                  Log.d(TAG, doc.getData().toString());

                  Note note = doc.toObject(Note.class);
                  notes.add(note);
              }
              NoteFragment listFragment = (NoteFragment)
getSupportFragmentManager().findFragmentByTag("list_note");
              listFragment.updateNotes(notes);
          }
      });
   }
   @Override
   public boolean onCreateOptionsMenu(Menu menu) {
      MenuInflater inflater = getMenuInflater();
      inflater.inflate(R.menu.menu, menu);
      return super.onCreateOptionsMenu(menu);
   }
   @Override
   public boolean onOptionsItemSelected(MenuItem item) {
      Log.d("onOptionsItemSelected", item.getTitle().toString());
      displayingEditor = !displayingEditor;
      invalidateOptionsMenu();
      switch (item.getItemId()) {
         case R.id.action_new:
            editingNote = createNote();
            FragmentTransaction ft = getSupportFragmentManager().beginTransaction();
            ft.replace(R.id.container,EditNoteFragment.newInstance(""),"edit_note");
            ft.addToBackStack(null);
            ft.commit();
            return true;
         case R.id.action_close:
            onBackPressed();
```

```
            return true;
        default:
            return super.onOptionsItemSelected(item);
        }
    }

    public boolean onPrepareOptionsMenu(Menu menu){
        Log.d("onPrepareOptionsMenu new visible",
menu.findItem(R.id.action_new).isVisible() + "");
        menu.findItem(R.id.action_new).setVisible(!displayingEditor);
        menu.findItem(R.id.action_close).setVisible(displayingEditor);

        return super.onPrepareOptionsMenu(menu);
    }

    @Override
    public void onBackPressed() {
        EditNoteFragment editFragment = (EditNoteFragment)
getSupportFragmentManager().findFragmentByTag("edit_note");
        String content = null;
        if (editFragment != null){
            content = editFragment.getContent();

        }
        super.onBackPressed();
        if (content !=null) {
            saveContent(editingNote, content);
        }
    }
    @Override
    public void onNoteSelected(Note note) {
        editingNote =note;
        FragmentTransaction ft = getSupportFragmentManager().beginTransaction();

ft.replace(R.id.container,EditNoteFragment.newInstance(editingNote.getContent()),"edit
_note");
        ft.addToBackStack(null);
        ft.commit();
        displayingEditor = !displayingEditor;
        invalidateOptionsMenu();
    }



    private Note createNote() {
```

```
        FirebaseFirestore db = FirebaseFirestore.getInstance();
        Note note = new Note();
        note.setId(db.collection("notes").document().getId());
        return  note;

    }

    private void saveContent(Note note, String content) {
        if (note.getContent() == null || !note.getContent().equals(content)) {
            FirebaseFirestore db = FirebaseFirestore.getInstance();
            note.setDate(new Timestamp(new Date()));
            note.setContent(content);
            db.collection("notes").document(note.getId()).set(note);
        }else{
            Log.d(TAG, "notes: " + notes);
            NoteFragment listFragment = (NoteFragment)
getSupportFragmentManager().findFragmentByTag("list_note");
            listFragment.updateNotes(notes);
        }

    }

    @Override
    protected void onStop() {
        super.onStop();
        listenerRegistration.remove();
    }
}
```

18. Run the application