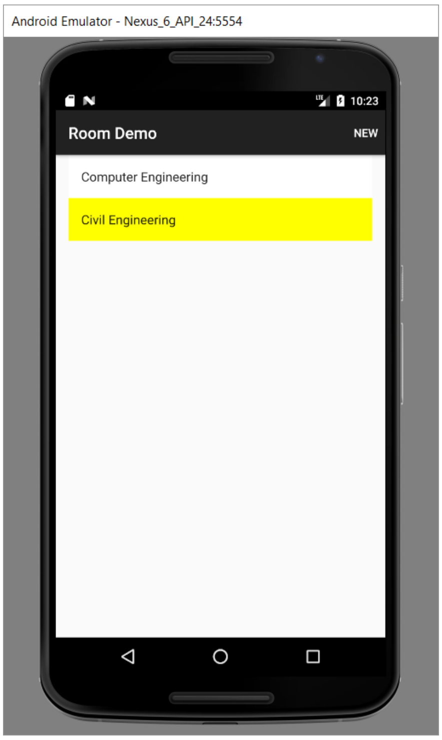
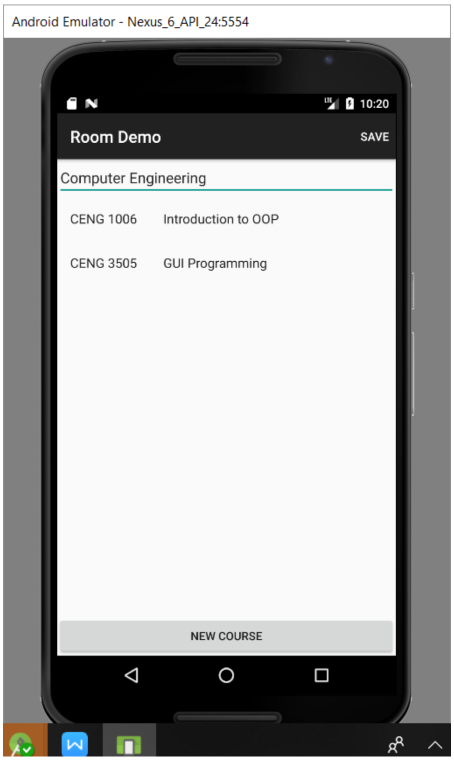
Lab 11: Room Demo

In this lab we will be practicing to save data locally using room.



Computer Engineering

Civil Engineering



Computer Engineering

CENG 1006 Introduction to OOP

CENG 3505 GUI Programming

1. Create new project in Android Studio with an empty activity.
2. Select defaults for minimum SDK
3. Add the following dependency to gradle

```
def room_version = "2.2.6"

  implementation "androidx.room:room-runtime:$room_version"
  annotationProcessor "androidx.room:room-compiler:$room_version"
```

4. Create the following entities

```
@Entity
public class Department implements Serializable{

    @PrimaryKey(autoGenerate = true)
    public long id;

    public String name;
}
```

```
@Entity(foreignKeys = @ForeignKey(entity = Department.class,
parentColumns = "id", childColumns = "deptId"))
public class Course {

   @PrimaryKey(autoGenerate = true)
   public long id;

   public String code;

   public String name;

   public long deptId;
}
```

5. Create the following Dao

```
@Dao
public interface UniversityDAO {

   @Query("Select * from department")
   public List<Department> getAllDepartments();

   @Query("Select * from course where deptId =  :deptId")
   public List<Course> getCourses(long deptId);

   @Insert (onConflict = OnConflictStrategy.REPLACE)
   public long insertDepartment(Department dept);

   @Insert (onConflict = OnConflictStrategy.REPLACE)
   public void insertCourses(Course... dept);

   @Update
   public void updateDepartment(Department dept);

   @Update
   public void updateCourses(Course... dept);

   @Delete
   public void deleteDepartment(Department dept);

   @Delete
   public void deleteCourses(Course... dept);

}
```

6. Create the following Room Database

```
@Database(entities = {Department.class, Course.class},
version = 1)
public abstract class UniversityDB extends RoomDatabase {
    public abstract UniversityDAO getDao();
}
```

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

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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>

</androidx.constraintlayout.widget.ConstraintLayout>
```

8. Create a fragment for listing departments based on Fragment (List) as shown below

9. Modify the fragment_department 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/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="@dimen/text_margin"
        android:textAppearance="?android:attr/textAppearanceListItem"
/>
</LinearLayout>
```

10. Modify the DepartmentFragment as shown below

```
public class DepartmentFragment extends Fragment {

    private static final String ARG_DEPARTMENTS = "departments";
    private OnDepartmentListInteractionListener mListener;
    private ArrayList<Department> departments;
    MyDepartmentRecyclerViewAdapter mAdapter;

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

    public static DepartmentFragment newInstance(ArrayList<Department> notes) {
        DepartmentFragment fragment = new DepartmentFragment();
        Bundle args = new Bundle();
        args.putSerializable(ARG_DEPARTMENTS, notes);
        fragment.setArguments(args);
        return fragment;
    }

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

        if (getArguments() != null) {
            departments =
(ArrayList<Department>)getArguments().getSerializable(ARG_DEPARTMENTS);
        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_department_list, container, false);
```

```java
        // Set the adapter
        if (view instanceof RecyclerView) {
            Context context = view.getContext();
            RecyclerView recyclerView = (RecyclerView) view;

            recyclerView.setLayoutManager(new LinearLayoutManager(context));

            mAdapter = new MyDepartmentRecyclerViewAdapter(departments, mListener);
            recyclerView.setAdapter(mAdapter);
        }
        return view;
    }

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        if (context instanceof OnDepartmentListInteractionListener) {
            mListener = (OnDepartmentListInteractionListener) context;
        } else {
            throw new RuntimeException(context.toString()
                    + " must implement OnNoteListInteractionListener");
        }
    }

    @Override
    public void onDetach() {
        super.onDetach();
        mListener = null;
    }

    public void setDepartments(ArrayList<Department> departments) {
        this.departments.clear();
        this.departments.addAll(departments);
        mAdapter.notifyDataSetChanged();

    }
    /**
     * Interface for listing note operations in the list
     */
    public interface OnDepartmentListInteractionListener {
        void onDepartmentSelected(Department item);
    }
}
```

11. Modify the Department Adapter as shown below

```
public class MyDepartmentRecyclerViewAdapter extends
RecyclerView.Adapter<MyDepartmentRecyclerViewAdapter.ViewHolder> {

    private final List<Department> mValues;
    private final DepartmentFragment.OnDepartmentListInteractionListener mListener;

    public MyDepartmentRecyclerViewAdapter(List<Department> notes,
DepartmentFragment.OnDepartmentListInteractionListener listener) {
        mValues = notes;
        mListener = listener;
    }

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

    @Override
    public void onBindViewHolder(final ViewHolder holder, int position) {
        holder.mItem = mValues.get(position);
        holder.mNameView.setText(mValues.get(position).name);

        holder.mView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (null != mListener) {
                    mListener.onDepartmentSelected(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 mNameView;
        public Department mItem;

        public ViewHolder(View view) {
            super(view);
            mView = view;
            mNameView =  view.findViewById(R.id.name);
        }

        @Override
        public String toString() {
            return super.toString() + " '" + mNameView.getText() + "'";
        }
    }
}
```

12. Create an EditDepartment fragment based on the Blank fragment template

13. Create a CourseAdapter class as shown below

```
public class CourseAdapter extends RecyclerView.Adapter<CourseAdapter.MyViewHolder> {

    private List<Course> courses;

    public class MyViewHolder extends RecyclerView.ViewHolder {
        public TextView code, name;

        public MyViewHolder(View view) {
            super(view);
            code =  view.findViewById(R.id.code);
            name = view.findViewById(R.id.name);
        }
    }


    public CourseAdapter(List<Course> courses) {
        this.courses = courses;
    }

    @Override
    public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View itemView = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.course_row, parent, false);

        return new MyViewHolder(itemView);
    }

    @Override
    public void onBindViewHolder(MyViewHolder holder, int position) {
        Course course = courses.get(position);
        holder.code.setText(course.code);
        holder.name.setText(course.name);
    }

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

14. Create course_row layout 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/code"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="@dimen/text_margin"
        android:textAppearance="?android:attr/textAppearanceListItem"
/>
    <TextView
        android:id="@+id/name"
        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"
/>
</LinearLayout>
```

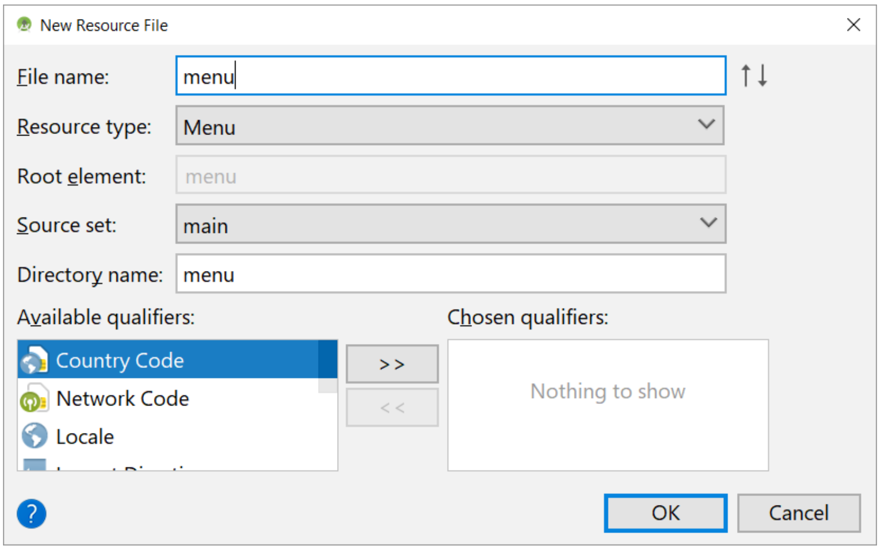
15. Modify fragment_edit_department as shown below

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

    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Type Name Of Department" />

   <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/courses"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"/>

    <Button
        android:id="@+id/new_course"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="New Course" />

</LinearLayout>
```

16. Modify the EditDepartmentFragment as shown below.

```
public class EditDepartmentFragment extends Fragment {
    private static final String ARG_DEPARTMENT = "department";
    private static final String ARG_COURSES = "courses";
    private Department department;
    private ArrayList<Course> courses;
    private EditText txtName;
    private RecyclerView recyclerView;
    private CourseAdapter mAdapter;

    Button btnNewCourse;

    public EditDepartmentFragment() {}

    /**
     * Use this factory method to create a new instance of
     * this fragment using the provided parameters.

     * @return A new instance of fragment EditNoteFragment.
     */
    public static EditDepartmentFragment newInstance(Department department,
ArrayList<Course> courses) {
        EditDepartmentFragment fragment = new EditDepartmentFragment();
        Bundle args = new Bundle();
        args.putSerializable(ARG_DEPARTMENT, department);
        args.putSerializable(ARG_COURSES, courses);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            department = (Department) getArguments().getSerializable(ARG_DEPARTMENT);
            courses = (ArrayList<Course>) getArguments().getSerializable(ARG_COURSES);
        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_edit_department, container, false);
    }
    @Override
    public void onViewCreated(View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        recyclerView = view.findViewById(R.id.courses);
        txtName = view.findViewById(R.id.name);
        btnNewCourse = view.findViewById(R.id.new_course);

        txtName.setText(department.name);

        mAdapter = new CourseAdapter(courses);
        RecyclerView.LayoutManager mLayoutManager = new
LinearLayoutManager(getActivity());
        recyclerView.setLayoutManager(mLayoutManager);
        recyclerView.setAdapter(mAdapter);

        btnNewCourse.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                AlertDialog.Builder builder = new
AlertDialog.Builder(EditDepartmentFragment.this.getActivity());
                builder.setTitle("New Course");

                final EditText txtCode = new
EditText(EditDepartmentFragment.this.getActivity());
                txtCode.setHint("Course Code");

                txtCode.setInputType(InputType.TYPE_CLASS_TEXT );

                final EditText txtName = new
EditText(EditDepartmentFragment.this.getActivity());
                txtName.setHint("Course Name");
                txtName.setInputType(InputType.TYPE_CLASS_TEXT );

                LinearLayout viewGroup = new
LinearLayout(EditDepartmentFragment.this.getActivity());
                viewGroup.setOrientation(LinearLayout.VERTICAL);
                viewGroup.addView(txtCode);
                viewGroup.addView(txtName);



                builder.setView(viewGroup);
                builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {

                        String code = txtCode.getText().toString();
                        Log.d("Dialog", code);

                        String name = txtName.getText().toString();
                        Log.d("Dialog", name);

                        Course course = new Course();
                        course.code = code;
                        course.name = name;
                        addCourse(course);
                    }
                });
                builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener()
{
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        dialog.cancel();
                    }
                });

                builder.show();
            }
        });

    }

    public void addCourse(Course course){
        courses.add(course);
        mAdapter.notifyDataSetChanged();
    }

    public Department getDepartment(){
        department.name = txtName.getText().toString();
        return department;
    }

    public ArrayList<Course> getCourses() {
        return courses;
    }
}
```

17. Create a new menu for the Activity



18. 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_save"
        android:title="Save"
        android:visible="false"
        app:showAsAction="always" />
</menu>
```

19. Modify the MainActivity as below

```
public class MainActivity extends Activity implements
DepartmentFragment.OnDepartmentListInteractionListener{


   boolean displayingDepartment = false;

   Department selectedDepartment;

   ArrayList<Department> departments;

   UniversityDB db;


   @Override

   protected void onCreate(Bundle savedInstanceState) {

      super.onCreate(savedInstanceState);

      setContentView(R.layout.activity_main);

      db = Room.databaseBuilder(this,UniversityDB.class,"university").build();
```

```
        new Thread(){
            public void run(){
                departments = retrieveDepartments();
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        FragmentTransaction ft =getFragmentManager().beginTransaction();
                        ft.add(R.id.container,DepartmentFragment.newInstance(departments),
"departments");
                        ft.commit();
                    }
                });
            }

        }.start();
    }

    @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());

        switch (item.getItemId()) {
            case R.id.action_new:
                displayingDepartment = !displayingDepartment;
                invalidateOptionsMenu();
                selectedDepartment = new Department();
                departments.add(selectedDepartment);
                FragmentTransaction ft = getFragmentManager().beginTransaction();

ft.replace(R.id.container,EditDepartmentFragment.newInstance(selectedDepartment,  new
ArrayList<Course>()),"edit_department");
                ft.addToBackStack(null);
                ft.commit();
                return true;
            case R.id.action_save:
                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(!displayingDepartment);
        menu.findItem(R.id.action_save).setVisible(displayingDepartment);
        return super.onPrepareOptionsMenu(menu);
    }



    @Override
    public void onBackPressed() {
        displayingDepartment = !displayingDepartment;
        invalidateOptionsMenu();
        EditDepartmentFragment editFragment = (EditDepartmentFragment)
getFragmentManager().findFragmentByTag("edit_department");
        if (editFragment != null){
            final Department department = editFragment.getDepartment();
            final ArrayList<Course> courses = editFragment.getCourses();
            new Thread(){
                public void run(){
                    saveDepartment(department, courses);
                    departments = retrieveDepartments();
                    runOnUiThread(new Runnable() {
                        @Override
                        public void run() {
                            DepartmentFragment departmentFragment = (DepartmentFragment)
getFragmentManager().findFragmentByTag("departments");
                            departmentFragment.setDepartments(departments);
                        }
                    });

                }
            }.start();

        }
        super.onBackPressed();
    }



    @Override
    public void onDepartmentSelected(Department department) {
        selectedDepartment =department;
        new Thread(){
            public void run(){
                final ArrayList<Course> courses = getCourses(selectedDepartment);
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        FragmentTransaction ft = getFragmentManager().beginTransaction();

ft.replace(R.id.container,EditDepartmentFragment.newInstance(selectedDepartment,
courses),"edit_department");
                        ft.addToBackStack(null);
                        ft.commit();
                        displayingDepartment = !displayingDepartment;
                        invalidateOptionsMenu();
                    }
                });
            }
        }.start();

    }

    public ArrayList<Department> retrieveDepartments(){
        UniversityDAO dao = db.getDao();
        return new ArrayList<>( dao.getAllDepartments());
    }

    private void saveDepartment(Department department, List<Course> courses) {
        UniversityDAO dao = db.getDao();
        long deptId = dao.insertDepartment(department);
        for (Course course : courses){
            course.deptId = deptId;
        }
        dao.insertCourses(courses.toArray(new Course[0]));

    }
    private ArrayList<Course> getCourses(Department department) {
        UniversityDAO dao = db.getDao();
        return new ArrayList<>(dao.getCourses(department.id));
    }
}
```

20. Run the application