



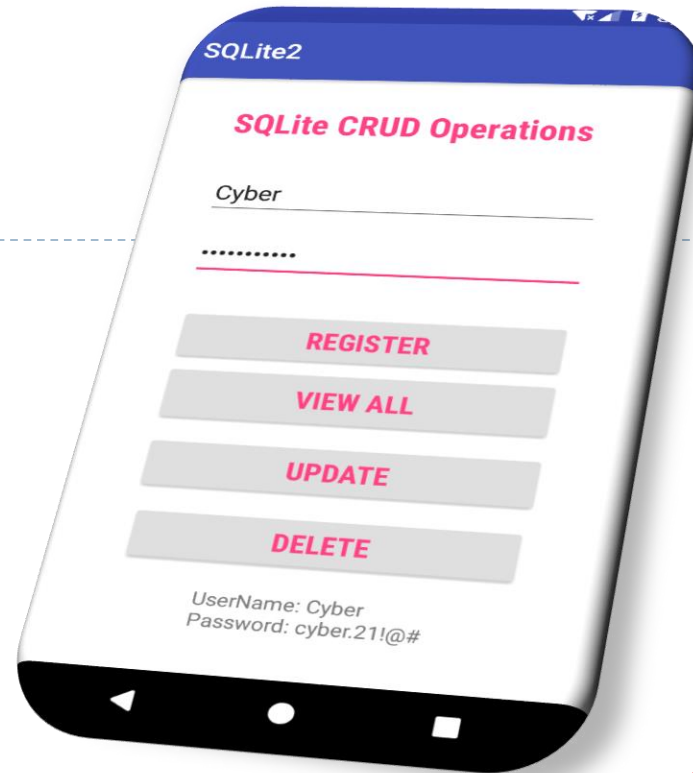
Assosa University
College of Computing and Informatics
Department of Computer Science

Wireless Communication and Mobile Computing
Mobile Application *Development*
Practical Lab Sessions

Android SQLite Database Connectivity

Prepared by: Gebreigziabher A.

Email: gebrishab@gmail.com

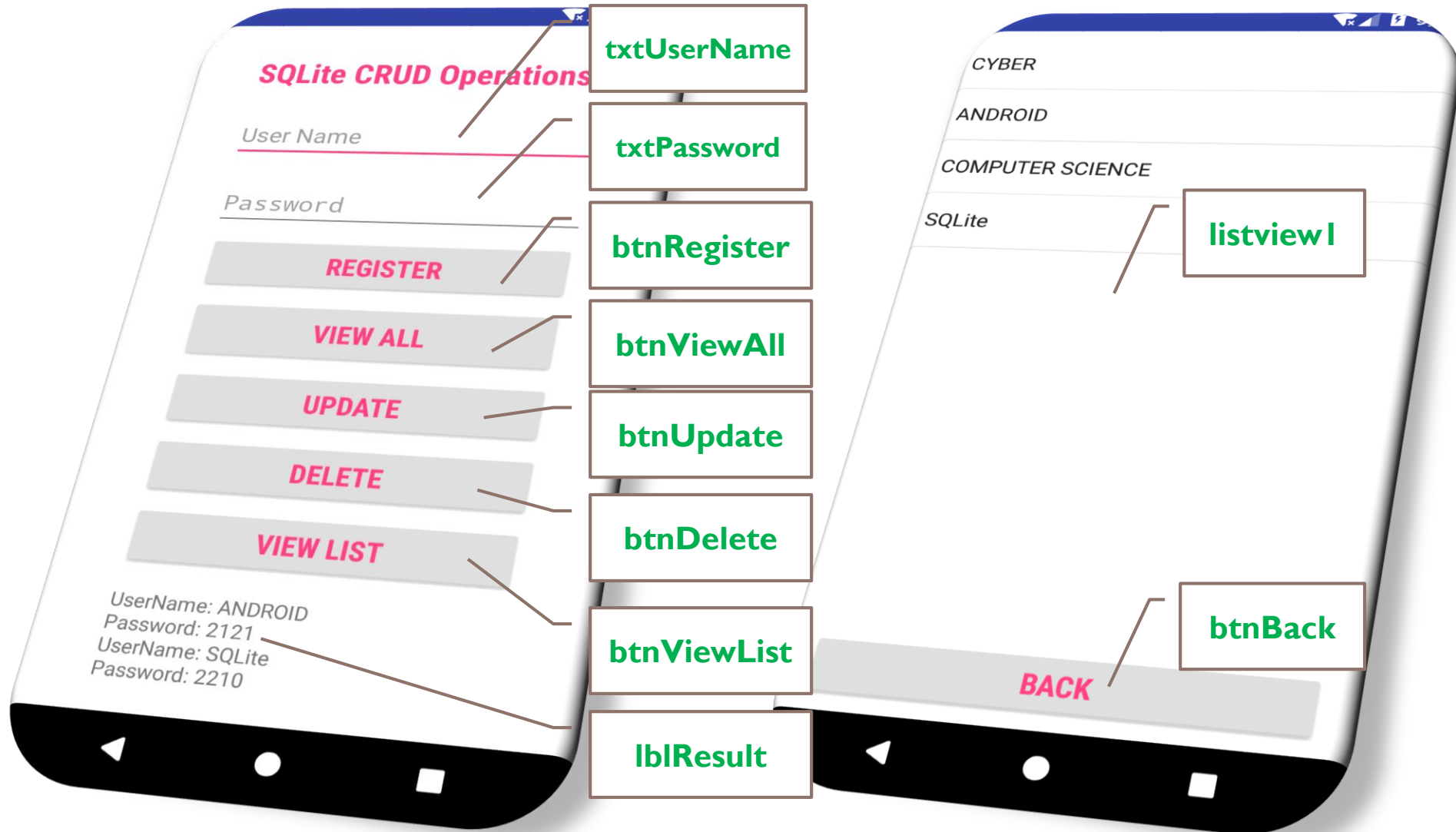


Android SQLite Database Connectivity

Prepared by: Gebreigziabher A.

Email: gebrishab@gmail.com

Step 1: UI Design



<TextView

```
    android:id="@+id/textView"  
    android:fontFamily="sans-serif-condensed"  
    android:text="SQLite CRUD Operations"  
    android:textColor="@color/colorAccent"  
    android:textSize="25dp"  
    android:textStyle="bold" />
```

<ListView

```
    android:id="@+id/listview1" />
```

...Cont'd

<EditText

```
    android:id="@+id/txtUserName"  
    android:hint="User Name"  
    android:fontFamily="sans-serif-condensed"  
    android:inputType="textPersonName" />
```

<EditText

```
    android:id="@+id/txtPassword"  
    android:hint="Password"  
    android:inputType="textPassword"  
    android:fontFamily="sans-serif-condensed" />
```

<TextView

```
    android:id="@+id/txtResult"  
    android:hint="Result" />
```

...Cont'd

<Button

```
    android:id="@+id/btnRegister"
    android:text="REGISTER"
    android:textSize="20dp"
    android:textColor="@color/colorAccent"
    android:fontFamily="sans-serif-condensed" />
```

<Button

```
    android:id="@+id/btnViewAll"
    android:text="VIEW ALL"
    android:textSize="20dp"
    android:textColor="@color/colorAccent"
    android:fontFamily="sans-serif-condensed" />
```

<Button

```
    android:id="@+id/btnViewAllList"
    android:text="VIEW LIST"
    android:textSize="20dp"
    android:textColor="@color/colorAccent"
    android:fontFamily="sans-serif-condensed" />
```

...Cont'd

<Button

```
    android:id="@+id/btnUpdate"  
    android:text="UPDATE"  
    android:textSize="20dp"  
    android:textColor="@color/colorAccent"  
    android:fontFamily="sans-serif-condensed" />
```

<Button

```
    android:id="@+id/btnDelete"  
    android:text="DELETE"  
    android:textSize="20dp"  
    android:textColor="@color/colorAccent"  
    android:fontFamily="sans-serif-condensed" />
```

<Button

```
    android:id="@+id/btnBack"  
    android:text="BACK"  
    android:textSize="20dp"  
    android:textColor="@color/colorAccent"  
    android:fontFamily="sans-serif-condensed" />
```

Step 2: DatabaseHelper.java

```
public class DatabaseHelper extends SQLiteOpenHelper {  
    public static final String DATABASE_NAME = "Student.db";  
    public static final String TABLE_NAME = "tblStudent";  
  
    public static final String COLUMN_1 = "userName";  
    public static final String COLUMN_2 = "password";  
  
    public DatabaseHelper(Context context) {  
        super(context, DATABASE_NAME, null, 1);  
    }  
  
    public void onCreate(SQLiteDatabase db) {  
        db.execSQL("CREATE TABLE "+TABLE_NAME+  
            "(userName TEXT PRIMARY KEY, password TEXT)");  
    }  
  
    public void onUpgrade(SQLiteDatabase db, int oldVersion,  
        int newVersion) {  
        db.execSQL("DROP TABLE IF EXISTS "+TABLE_NAME);  
    }  
}
```



Database Version

Insert Record

```
public boolean insertData(String userName, String password){

    SQLiteDatabase db = this.getWritableDatabase();

    ContentValues contentValues = new ContentValues();

    contentValues.put(COLUMN_1, userName);
    contentValues.put(COLUMN_2, password);

    long result = db.insert(TABLE_NAME, null, contentValues);
    db.close();
    if(result==-1) {
        return false;
    }else {
        return true;
    }
}

public Cursor getAllData(){ //Retrieve Record
    SQLiteDatabase db =this.getWritableDatabase();
    Cursor result = db.rawQuery("SELECT *FROM "+TABLE_NAME, null);
    return result;
}
```

Update Record

```
public boolean updateData(String userName, String password) {
    SQLiteDatabase db = this.getWritableDatabase();

    ContentValues contentValues = new ContentValues();
    contentValues.put(COLUMN_1, userName);
    contentValues.put(COLUMN_2, password);

    int result = db.update(TABLE_NAME, contentValues,
        "userName=?", new String[]{userName});
    if (result > 0) {
        return true;
    }
    else {
        return false;
    }
}

public Integer deleteData(String userName){ //Delete Record
    SQLiteDatabase db = this.getWritableDatabase();
    int result = db.delete(TABLE_NAME, "userName=?",
        new String[]{userName});
    return result;
}
```

View All - List View

```
public Cursor  viewAllList() { //Displaying to List View

    SQLiteDatabase db = this.getReadableDatabase();

    String[] columns = new String[]{COLUMN_1, COLUMN_2};

    Cursor cursor = db.query(TABLE_NAME, columns,
null, null, null, null, null);

    return cursor;

}
```

Step 3: MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    EditText txtUserName, txtPassword;
    TextView txtResult;
    Button btnRegister, btnViewAll, btnUpdate, btnDelete, btnViewList;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        DatabaseHelper db = new DatabaseHelper(this);
        txtUserName = (EditText) findViewById(R.id.txtUserName);
        txtPassword = (EditText) findViewById(R.id.txtPassword);
        txtResult = (TextView) findViewById(R.id.txtResult);
        btnRegister = (Button) findViewById(R.id.btnRegister);
        btnViewAll = (Button) findViewById(R.id.btnViewAll);
        btnUpdate = (Button) findViewById(R.id.btnUpdate);
        btnDelete = (Button) findViewById(R.id.btnDelete);
        btnViewList = (Button) findViewById(R.id.btnViewList);
    }
}
```

Register

```
btnRegister.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        String userName = txtUserName.getText().toString();  
        String password = txtPassword.getText().toString();  
  
        boolean result = db.insertData(userName, password);  
        if(result==true) {  
            Toast.makeText(getApplicationContext(),  
                "Registered!", Toast.LENGTH_LONG).show();  
        }  
        else {  
            Toast.makeText(getApplicationContext(),  
                "Failed to Register!", Toast.LENGTH_LONG).show();  
        }  
    }  
});
```

View All – Text View(Label)

```
btnViewAll.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Cursor result = db.getAllData();

        StringBuffer stringBuffer = new StringBuffer();
        if(result!=null && result.getCount()>0) {
            while (result.moveToNext()) {
                stringBuffer.append("UserName: " + result.getString(0) + "\n");
                stringBuffer.append("Password: " + result.getString(1) + "\n");
            }
            txtResult.setText(stringBuffer.toString());
        }
        else {
            Toast.makeText(getApplicationContext(), "No Data to
            Retrieve!", Toast.LENGTH_LONG).show();
        }
    }
});
```

Update

```
btnUpdate.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        String userName = txtUserName.getText().toString();  
        String password = txtPassword.getText().toString();  
  
        boolean result = db.updateData(userName, password);  
        if(result==true){  
            Toast.makeText(getApplicationContext(),  
                "Data Updated!", Toast.LENGTH_LONG).show();  
        }  
        else {  
            Toast.makeText(getApplicationContext(),  
                "No Data to Update!", Toast.LENGTH_LONG).show();  
        }  
    }  
});
```

Delete

```
btnDelete.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        String userName = txtUserName.getText().toString();  
        int result = db.deleteData(userName);  
        if(result==1){  
            Toast.makeText(getApplicationContext(),  
                "Data Deleted!", Toast.LENGTH_LONG).show();  
        }  
        else {  
            Toast.makeText(getApplicationContext(),  
                "Data Not Deleted!", Toast.LENGTH_LONG).show();  
        }  
    }  
});
```


View List

```
btnViewAllList.setOnClickListener(new
View.OnClickListener() { //Displaying to List View

    @Override
    public void onClick(View v) {

        Intent i = new Intent(getApplicationContext(),
        ViewAllList.class);

        startActivity(i);
    }
});
}
```

View All - List View

```
public class ViewAllList extends Activity {
    //Displaying to List View (ViewAllList.java)
    ArrayList<String> users = new ArrayList<String>();
    ArrayAdapter<String> adapter;
    ListView listView;
    Button btnBack;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view);

        DatabaseHelper db = new DatabaseHelper(this);

        adapter = new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_1, users);

        btnBack = (Button) findViewById(R.id.btnBack);
        listView = (ListView) findViewById(R.id.listview1);
    }
}
```

...Cont'd

```
Cursor cursor = db.viewAllList();
if(cursor!=null && cursor.getCount()>0) {
    while (cursor.moveToNext()) {
        String userName = cursor.getString(0);
        users.add(userName);
    }
    db.close();
    listView.setAdapter(adapter);
} else {
    Toast.makeText(getApplicationContext(), "No Data Found!",
    Toast.LENGTH_LONG).show();
}
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int
    position, long id) {
        Toast.makeText(getApplicationContext(), users.get(position),
        Toast.LENGTH_SHORT).show();
    }
});
```

```
btnBack.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Intent in = new Intent(getApplicationContext(),  
            MainActivity.class);  
        startActivity(in);  
    }  
});  
}  
}
```