HAVI REGISTER

SUBJECT

Sr.No.	Date	Topic
		Version Control System (Gittlub)
	3/3/21	Android Studio IDE Introduction
		Constraint Layout
		Linear layout
		View & View Groups - Layouts details
		On Click Listener (3 types)
		Activity, Intents
		Activity Lifecycle (log.d used)
		Adapters, List View binding
		Passing of data b/w activities (.putExtra, .getStringExtra)
		Seekbar
		Media: Images, Animations, Videos, Audio
		Font Customization
		SQLite (Toast) (Dialog)
		Drawer Navigation (Vector Asset
		RecyclerView
		Material Design
		Notifications
		Fragments

HAVI PHOTO STATE & STATIONERS
Ghang Road Street #1, Sheikhupura. 0300-4548755-0321-4208016

Scanned with CamScanner

good links:

> flaticon.com 64 x 64
> Android Asset Studio (on Google: Launcher icon generator) don't change icon name

lecture #2

Lecture Outline:

> Template-based approach to Systematically Perform a Real-world Task.
> Version Control System
> Create account on Github
> Clone a Repository
> Git Commit
> Git Status
> Git Push
> Git Pull, log, online editing
> Branching, Merge, delete

dp - density pixels

sp - scale independent pixels ← use for text

px - pixel

ctrl + D to duplicate a line of code

ctrl + / to comment code

ctrl + ␣ space to display tag options

o VCS

> Keep track of changes to code

- Create a file - Added Code - Modified Code
- Deleted Code

> Code Synchronizing between different groups

① Central Repository int a = 1; [Same Code Available to all members]

A's Repository int a = 1; int b = 2;

B's Repository int a = 1; String B = "hello";

C's Repository int a = 1; int x = a;

[Diff. people working on diff. files]

Scanned with CamScanner

(2)

Central Repository

int a = 1;
int b = 2;
String B = "hello";
int x = a;

Same as prev. diagram.

(3)

Central Repository

int a = 1;
int b = 2;
int B = "hello";
int x = a;

A's Repository | B's Repository | C's Repository

int a = 1;
int b = 2;
int B = "hello";
int x = a;

> Code Testing: (without losing original)

> Revert back to old versions

O Github

> Create Account

*> Create Repository

- Provide details
- Url of repository

> Git Clone

We want to link up or make clone of a code present on Central repository to a local repository

git clone [url]

<- don't write [ ]

This will create a new folder. Everything in that folder will be on github.

> Git Add

Added to local repository
Not available in central yet

git add [File Name]

code file.txt

to open a .txt doc. for editing

Scanned with CamScanner

> Git Commit : Changes made with messages to track changes

```
git commit -m "Message"
git commit -m "First Commit"
```

> Git Status: Tells status of repository

```
git status
```

[whether up to date or not.]

origin/master

> Git Push: Push changes to server. [synchronize with central repository]

```
git push
```

> Create Repository using command line

```
echo "# MC-1" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M master
git remote add origin https://github.com/haqmuaz99/MC-1.git
git push -u origin master
```

> Commiting Changes to repository:

← more files then seperate with space of "File names with space"

```
git add [fileName]
git commit -m "_____"
git push
```

Track of changes

To do together:
```
git Commit -am "_____"
```

> Edit Online

web version will be ahead of local version

- Commit directly to master branch

Scanned with CamScanner

> View Changes

> Version Hierarchy

> Git Pull : To get changes from Github to local repository

git pull

Lecture # 3

> Merge Conflicts: same code being edited by IDE

> Git log

> Remove files from Central Repository

git rm file3.txt

git add . — selects all files

git commit -m "Filesdeleted"

git push

> Branching

– Create a branch

- git branch — git branch — will give info about branches
- git check — git checkout -b newbranch — name of new branch — creates as well as navigates branches
- git merge — git merge newbranch ← Move to the other branch i.e Master before this

– Delete a branch

git branch -D NewBranch

Scanned with CamScanner

- Mobile Computing
- Goals of Mobile Computing
- Benefits
  - > Portability > Miniaturization > Convergence
  - > Digital Ecosystems > Applications

[In slides] - Theoretical

---

lecture 4
3 Mar 2021

Lecture Outline

- > Android IDE Introduction
- > Setting of new project in Android Studio
- > Android Studio file system
- > Android virtual device (AVD)
- > USB debugging
- > Android profiler
- > Virtual Devices, Real Devices
- > Layout

How to share app?
> Make APK
Build ⇒ Build Bundle/APK
After building click on locate
we can rename it

APK - Android Application Package

- Start a new Project
- Choose Project (Template)
- Android Project Configuration

Playstore pe unique

Package Name - will uniquely identify ur app's APK file

https://play.google.com/store/apps/details?id= [package name]

better practice: com domain name app name

API – Application Programming Interface

Minimum API level FINISH

- Background tasks (building...)
- Android studio File Structure
- Files for Coding

Java → MainActivity.java activity_main.xml

MainActivity — coding GUI stuff

xml is a markup language

- Files for GUI

  we design the layout here.

- Component Tree It shows the structure of design.
- Project Resources

res

drawable (images) Gradient, custom text view, custom edit view, image shape, etc

layout

activity_main.xml make more files for more layouts

mipmap

Alt + ↵

to open a menu which will allow Extract String Resource

icon according to resolution requirements (more 'x' better resolution)

Define primary, accent etc ← values colors.xml here for eg we can name a colour, & whenever we need it we can use that name.

Whenever using some text define here (better practice) ← strings.xml

style.s.xml

v. imp

- build.gradle (Project) dependencies, 3rd party tools Required files All info about our app.

Gradle Script
- make APK
- The work done to run app. on device

- build.gradle (module app)

  which SDK
  version code
  version name
  etc.

  update stuff

  change version no. & it will sync the project

  Add dependencies etc

SDK – S/W Development Kit

- TODO — thru this we can see in a glance what stuff we have to do or are pending

  e.g. //TODO : Have to implement reverse indexing

- Terminal

- Project build — ▷ - on top right

- Project logCat — > all the details of what's happening
  > debug

- AVD (Android Virtual Devices) Manager

  TOOLS → AVD Manager

  can download virtual devices thru this

  + create Virtual Device — select device — Clone Device

- Run on Virtual Device

- App launched

- Setting for real Device

  In phone → settings → Developer Options → turn on

  Integerate for debugging

  [Different for different phones]

  this will show your phone for running the app.

  In Android Studio.

- Running on real device.

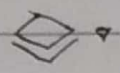
Scanned with CamScanner

- Manifests
  All our app is managed here
  > Intent filter Tells us which activity is to be watched on launch — Main Launcher and many more

- Android Profiler — Gives us details of resources (CPU usage, memory, network etc).

  view → tool windows → Profiler

- Design and Blueprint



- XML Design View
  (GUI type thing to design)
- XML Text View — activity_main.xml — text (bottom left)
  (Coding to design)

- Constraint Layout [In Android Studio playlist on Youtube]
  Top Right Left Bottom Constraints

  Remove bottom spring to make the text go up

> Click on the circles on edges to delete a constraint or to attach anchor points.

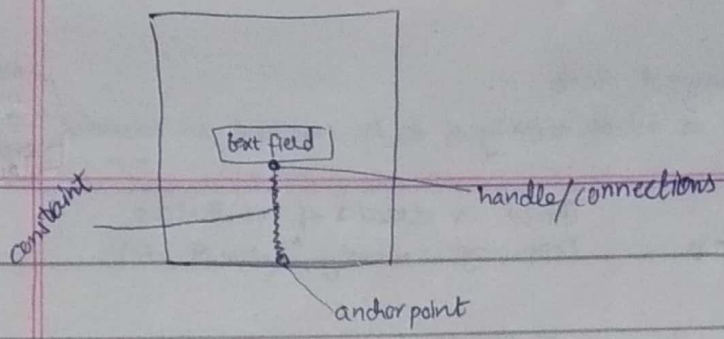
> Horizontal handle will only attach to horizontal anchor points & vice versa for vertical

> One handle can have only one constraint, but one anchor point can have multiple constraints.

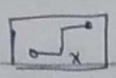
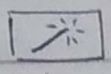
> Atleast 1 horizontal & 1 vertical constraint is necessary for each view.

Instead of running again & again, click ⚡ symbol to apply changes, to save time

Scanned with CamScanner



> By default i.e., without constraints : top left corner

>  

Removes all constraints — Infer constraints = Automatically sets constraints to where text field is

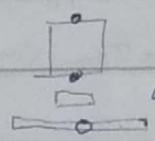
> Constraint layout allows creation of large & complex layouts with a flat view hierarchy (no nested view groups)

increases performance

> Match Constraint

Remove & use wrap content if want to use biasness

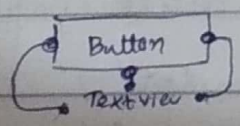
will edit width / height of the element acc. to constraints.



use Margins as less as possible as for different screens they might not work well [cz CONSTANT]

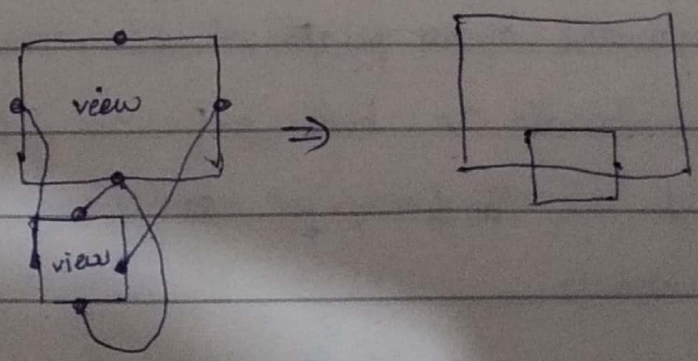
>

To place text below button



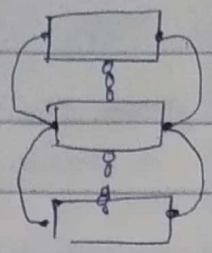
This will always keep the text below the button in centre.

To make one element be ½ above the bottom line of another element



Scanned with CamScanner

> Chains



component tree

↳ Constraint layout

- Button 1
- Button 2
- Button 3

select all

⇒ Right Click

⇒ Chains ⇒ create vertical chains

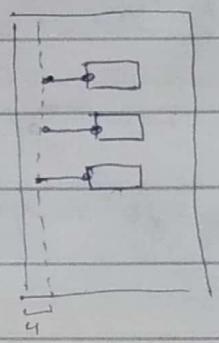
↳ ⇒ Packed (will bring the buttons together)

↳ ⇒

↳ ⇒

> Guidlines

on left top.

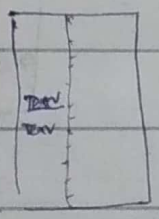


will keep the button 4 _ away from left

> Barriers

In guidlines

keep two (say) textViews in it [thru component tree (highlight + place in barrier)]



Now whatever we place on RHS of barrier will remain on RHS of it.

split

top Right corner
- opens code

[Some times design ~~tab~~ becomes irresponsive]

• Change barrier direction thru split code.

<android.constraintlayout.widget.Barrier

app:barrierDirection="left"

Component Tree

▾ I barrier

reference IDs [textView1, textView2

font family
(Roboto -good)

> Groups [from guidelines as well]

Suppose we have 3 buttons on screen and we want them all to be gone on click of any one of them.

visibility: [ ]
gone

put the 3 buttons in the group. [□] group

> Constraint Baseline

Name:

It will arrange the input field with the text properly.

Image View
⇒ scaleType
⇒ centerCrop
to fit an image horizontally

to change
Colour of icons
select them
⇒ tint
(type colour w and autofill will help)
⇒ @ android:color/white

View width
2dp
to make line

Circle image view
in split code
change image view to circle image view (autofill) & replace
app: srcCompat
with
android: src
⇒ Build

Libraries available on google
⇒ paste in build grade (Module)
↳ dependencies
⇒ sync

similarly
Rounded image view

Scanned with CamScanner

Lecture 5
8 March 2021
video 7

Lecture Outline

- View, ViewGroup
- Layout
- Layout Parameters
- Constraint layout
- Constraint baseline
- Constraint centre
- Chains
- Constraint bias
- Infer constraints

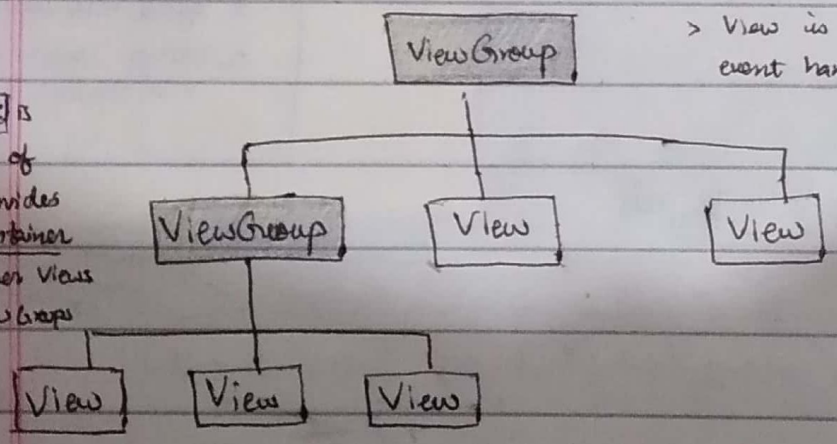
★ Layouts:
- Constraint Layout
- Linear Layout
- Relative Layout
- Frame Layout
- Grid Layout (Gallery)
- Table Layout (calculator)
- Tab Layout (In whatsapp we can swipe chats tab, status tab)

★ Responsive Layout: A layout that fits on both portrait & landscape view of a mobile

View & ViewGroup

View: Basic building block for UI
- Created from View Class & occupies a rectangular area on screen
- View is responsible for drawing & event handling
- View is the base class for widgets, which are used to create interactive UI components like buttons, text fields etc.

View Group: is a subclass of View & provides invisible container that hold other views or other ViewGroups & define their layout properties



- A View usually draws something the user can see and interact with.
- A ViewGroup is an invisible container.
- The View objects are usually called "widgets" and can be one of many subclasses, such as Button or TextView
- The ViewGroup objects are usually called "layouts" can be one of many types that provide a different layout

Scanned with CamScanner

structure, such as LinearLayout or ConstraintLayout.

> Each layout file must contain exactly one root element, which must be a View or ViewGroup object. Once you've defined the root element, you can add additional layout objects or widgets as child elements to gradually build a View hierarchy that defines your layout. For example, here's an XML layout that uses a vertical linearLayout to hold a TextView & a Button.

In Slides

- Load XML Resources

Add new resource

▽ res

▷ drawable

right click ▽ layout

⇒ New

⇒ Layout resource file

⇒ (settings: LinearLayout (say) root element)

In MainActivity.java

the name we gave to the resource

setContentView (R.layout.Linear_layout);

previously activity_main: would have been written here. We've replaced that to implement this layout

- Layout Parameters

> XML layout attributes named Layout_something define layout parameters for the View that are appropriate for

Scanned with CamScanner

> Layouts are subclasses of View Group class, & a typical layout defines the visual structure for an Android UI.

> The layout defines the visual structure of your app.

> Android Studio uses this structure to display the view elements on the screen.

> Layouts are also called view containers

> A layout may contain any type of views such as buttons, labels etc

View & View Group

View Group is a subclass of

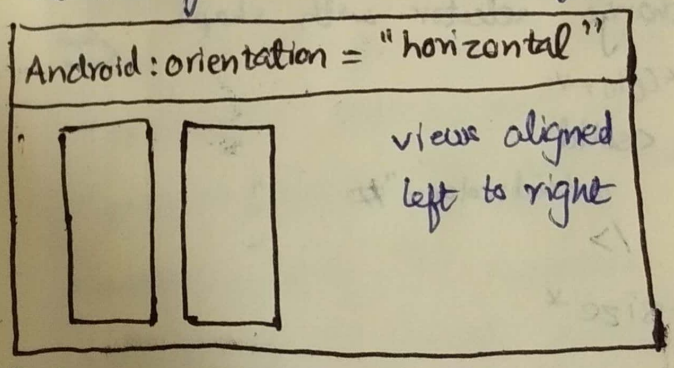
3. Constraint Baseline

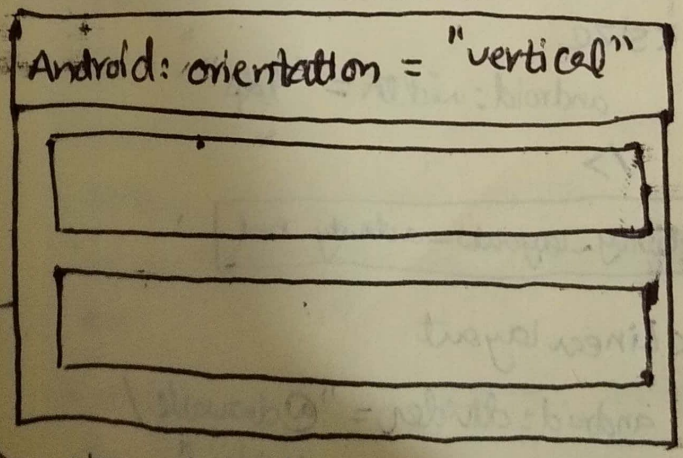
Name :

It will arrang ... ut field with properly.

Linear Layout

Android Linear layout is a View Group that aligns all its children in either vertically or horizontally.





Scanned with CamScanner

Custom divider < width, colour

res ⇒ drawable

* right click *

New ⇒ Resource File

Filename : divider (should be lowercase)

divider.xml

change selector with shape

Color

```
<solid
    android:color="#    "
/>
```

size

```
<size
    android:width="4dp"
/>
```

activity_layouts_activity.xml

```
<LinearLayout
    android:divider="@drawable/divider"
    android:showDividers="__"
```

Constraint Baseline

Name :

it will arrange

properties

Linear Layout Attributes

- android:id This is the ID which uniquely identifies the layout.
- android:divider This is drawable to use as a vertical divider b/w buttons. We can use a colour value in the form of "#RGB", "#ARGB", "#RRGGBB", or "#AARRGGBB" (Don't have to call for each view)
- android:gravity This specifies how an object should position its content, on both X & Y axes. Possible values are top, bottom, left, right, center, center_vertical, center_horizontal etc
- android:orientation This specifies the direction of arrangement. Default is horizontal.

Scanned with CamScanner

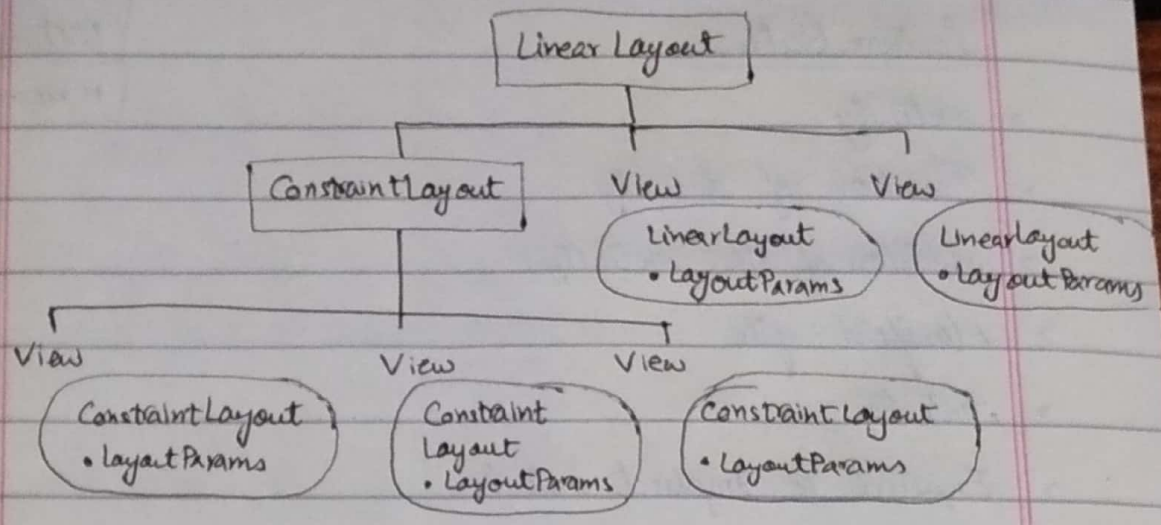
- android:layout_weight

This assigns importance on how much screen a View can relatively occupy compare to others.
(for responsive layout)

Scanned with CamScanner

the ViewGroup in which it resides.

Every ViewGroup class implements a nested class that extends ViewGroup.LayoutParams. This subclass contains property types that define the size and position for each child view, as appropriate for the view group.



Required Dependency in build.gradle (Module)

Layout Conversion

Images to be used
- paste in drawable

Load XML Resources

Main Activity.java

```
Button myButton = findViewById
                  (R.id.button4);
```

res folder

import android.widget.Button

will automatically come.

```
TextView textView = findViewById (R.id.textView3);
```

activity_main.xml

```
<Button
    android:id="@+id/button4"

    android:onClick="SomeOneIsThere"
```

function name

In big projects, don't apply onClickListener in XM
It is not a good practi

Main Activity .java

```
public void someOneIsThere (View view)
{
    textView.setText ("Some one new is there");
}
```

Entry Points of Activity

We can enter in an application from multiple points.

lecture 6

10 March 2021

Scanned with CamScanner

the ViewGroup in which it resides.
Every ViewGroup class implements a
extends ViewGroup.LayoutParams.
property types that
child view
the size
the view

OnClickListener

on any view
3 ways
MainActivity.java

① public class MainActivity
extends AppCompatActivity {

ⓘ TextView tvfirst;

— onCreate(——) {

ⓘⓘ tvfirst = findViewById
(R.id.tvfirst);

ⓘⓘⓘ tvfirst.setOnClickListener
(new View.OnClickListener() {
@Override
public void onClick(View v) {

}
});

Linear

Button myButton = findVie
(R.i
res
folder

Main
Activity.
java

TextView textView = findView

the ViewGroup in which it ...
Every ViewGroup class implemen...
extends ViewGroup.LayoutPara...
property types that define the ...
child view, as ... for the ...

(2) public class MainActivity
extends CompActivity
(i) implements View.OnClickListener
{
(ii) TextView tvSecond;
onCreate(———){
———
(iii) tvSecond = findViewById
(R.id.second);
(iv) } → second.setOnClickListener
(this);
@Override
(v) public void onClick(View view)
{
switch(view.getId())
{
case R.id.second:
———
break;
}

this function will automatically come after (i)

Main Activity.java

activity_main.xml

...ton = find...
(R...
res folder
TextView textView = find...

<Button
android:id="@+id/button...
———
android:onClick="...

Scanned with CamScanner

yout.
element,
e

the ViewGroup in which i
Every ViewGroup class impl
extends ViewGroup.Layout
property types that define
child view, as appropriate for

In big projects don't apply onClickListener in XML.
It's not a good approach and can result in errors.

3) Activity_main.xml

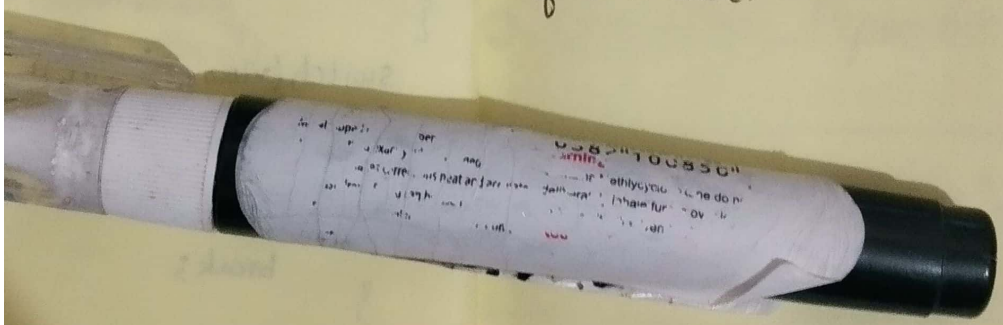
<TextView

android:OnClick "third"

(any name)
name of func.
which will automatically
get created in
MainActivity.java

MainActivity.java

1 method would have been automatically created out of OnCreate() method.



load XML resou

Main Activity.java

Button myButton =

TextView textView =

activity

<Button

Lecture 6
10 March 2021

Lecture Outline

- Activity
- Function of Activity
- Addition of New Activity
- Manifest file
- Intents
- Explicit & Implicit intents
- Working of Activities
- Activity Lifecycle

- Activity (Jo screen pe nazar aa raha hai at a point.)
- An activity is an application component
- Represents one window, one hierarchy of views.
- Typically fills the screen, but can be embedded in other activity or appear as floating window.
- Java class, typically one Activity in one file

Functions of Activity

- Reciting "surah", sending email, dialing number ← Activities
- Handles user interactions, such as button clicks, text entry, or login verification.

Scanned with CamScanner

Main Activity .java

```
public void someOneIsThere (View
{
    textView.setText ("Some
}
```

Entry Points of Activity

We can enter in an application from multiple points.

For e.g.

> When we click on FB notification, it opens that.

If we click on friend req notif, it opens that.

> If we minimize the app. we continue from the same place

> By clicking on the app. icon

Exit Points of Activity

For e.g.

> When we minimize the app.

> When we minimize, then swipe it out

> Moving to an external app from within app.

> Rep.

one window, one

> fills the screen,

d in other activity

ting window.

ava class, typically one Activi

Scanned with CamScanner

> Can start other activities in the same or other apps.

> Has a life cycle

- created - started - runs - is paused
- resumed - stopped - destroyed

- Examples of Activities – in slides

RegEx – Regular Expression

A regular expression is a sequence of characters that specifies a search pattern.
Usually such patterns are used by string-searching algorithms for "find" or "find & replace" operations on strings, or for input validation.
It is a technique developed in CS & formal language theory.

Used to
1. Validate Text
2. Search thru text

- Activity manifest

If there are 2 activities with intent in manifest. It will start the 1st one & ignore the second.

declare starting activity

intent-filter to start from launcher

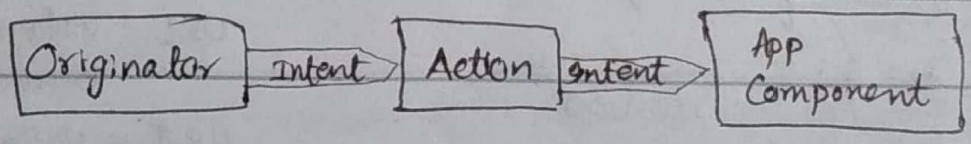
- Add New Activity

In Slides

tick launcher activity, if you want this activity to be ur starting activity

- Intent (Jo kaam ham android system se karwaana chahte hain)

An intent is a description of an operation to be performed. An intent is an object used to request an action from another app component via the Android System.



- Intents can do:

> Start an activity

- A button click starts a new Activity for text entry.

Scanned with CamScanner

- Clicking Share opens an app that allows you to post a photo.

> Start a service

- Fetching data in the background.

> Deliver Broadcast

- The system informs everybody that the phone is now charging.

○ Explicit & Implicit intents

> Explicit Intent (We tell what to do)

- Starts a specific activity
  - Main activity starts Surah Activity
  - Main activity starts Parah Activity

> Implicit Intent (We don't define)

- Asks system to find an Activity that can handle this request
  - Multiple options for sharing.

○ Examples of Implicit Intents

* In Slides *

URL - Uniform Resource locator

URI - Uniform Resource Identifier

○ Working of Activities

> All activity instances are managed by the

Scanned with CamScanner

Android runtime

> Started by an "Intent", a message to the Android runtime to run an activity.

main2Activity.Java

```
Public void MoveActivity (View view)
{
  Intent intent = new Intent(packageContext: this, MainActivity.class);
  StartActivity (intent);
}
```

current activity

the activity we want to move to

activity_main2.xml

```
<Button
android: onClick="MoveActivity"
/>
```

8 - Intent video on YouTube

for complete code

---

counter app on rotation screen resets.

Solution

onSaveInstanceState (Bundle outState)
{
It will save the values in bundle onStop state

outState.putInt ("value", count);

In onCreate

if (savedInstanceState != null)
{
count = savedInstanceState.getInt (key:"value");

textView.setText ...
}

Lecture 7

18 March 2021

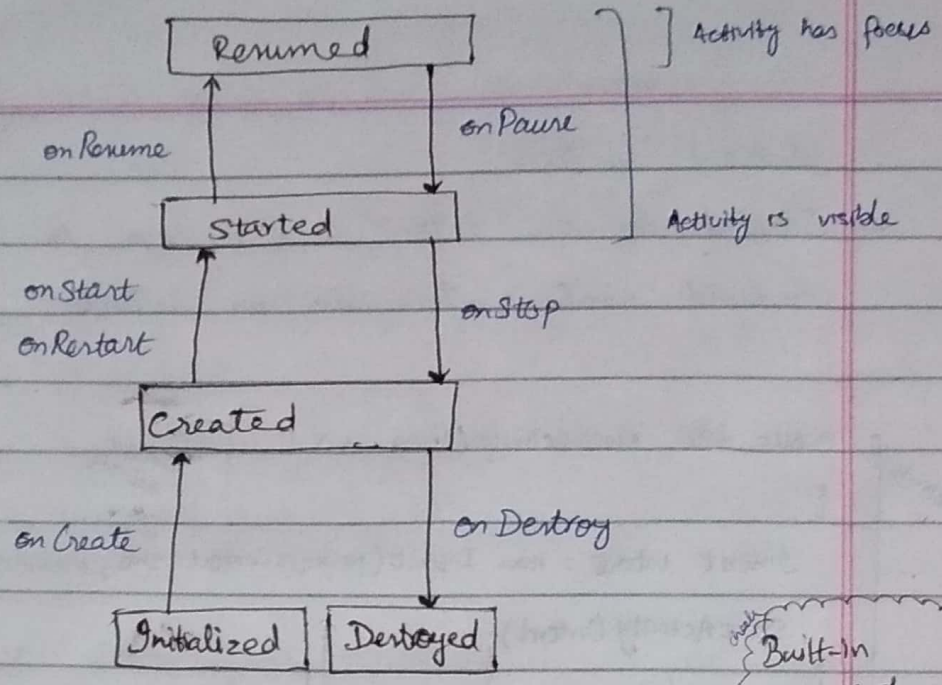
Activity Lifecycle

To navigate transitions b/w stages of the activity lifecycle, the Activity class provides a core set of 6 callbacks

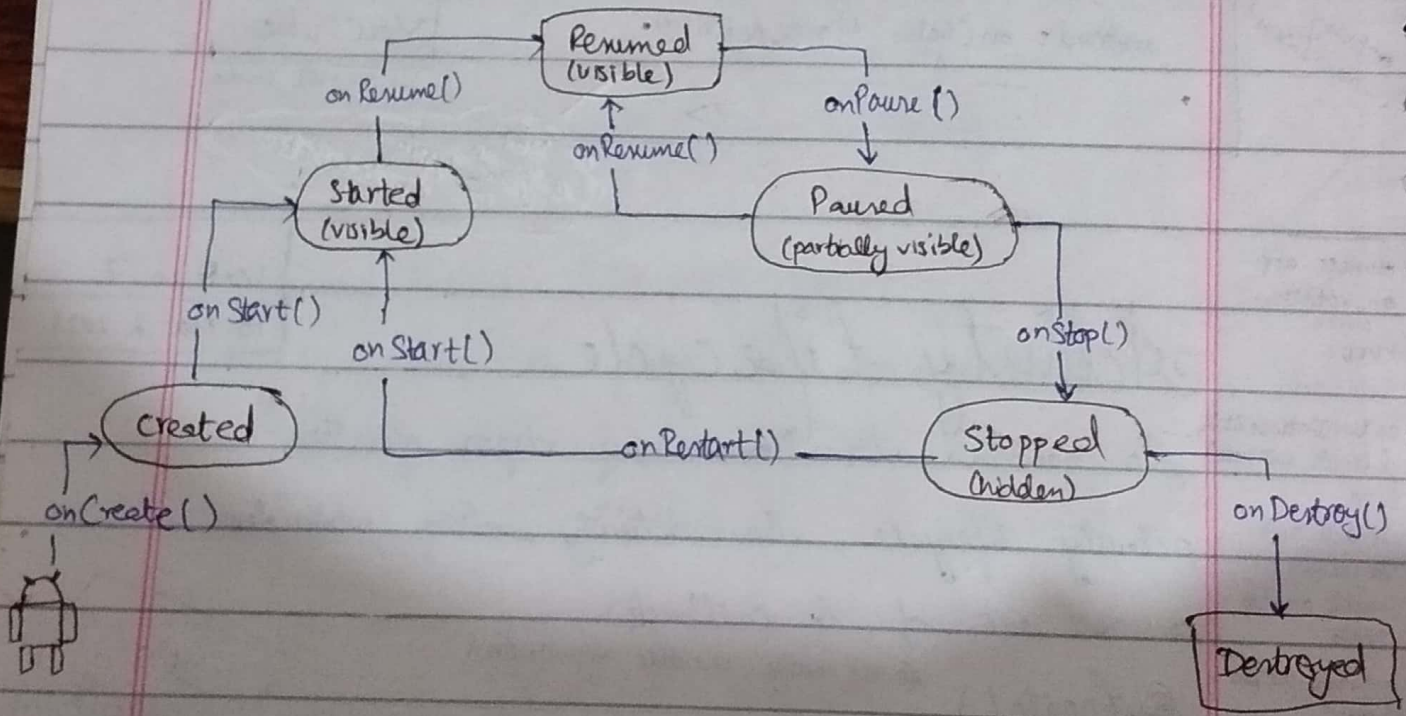
- onCreate() — All the coding, variable initialisation
- onStart()
- onResume() — After calling of this, we see the UI
- onPause() — - partially visible
- onStop() — - hidden — on minimization of the app. etc
- onRestart()
- onDestroy()

super.On__();
to work from any of these methods

Scanned with CamScanner



Built-in apps are of highest priority



* WATCH VIDEO FOR DEMO *

log.d(tag: " ", msg: " ");

to know from logcat, when is this happening.

{When going into activity 2 from activity 1

ALC1 onCreate	ALC2 onStart
	ALC2 onResume
ALC1 onStart	ALC1 onStop
ALC1 onResume	ALC2 onPause (exit activity 2 by going back)
ALC1 onPause	ALC1 onStart
ALC2 onCreate	ALC1 onResume
	ALC2 onStop
	ALC2 onDestroy

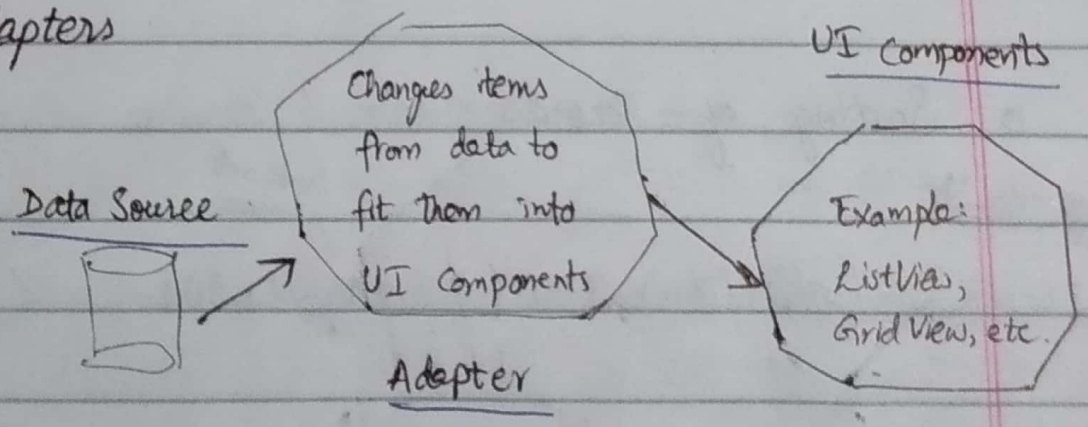
}

Lecture Outline

Lecture 8
21 March 2021

> Adapters
> Android Adapters
> Array List
> Array Adapter
> List View binding and on Item Click Lister
> ~~Spinners~~ Passing data b/w activities
> Resources

Adapters



Name — Text View

Age — Text View

Scanned with CamScanner

- ArrayList

Difference b/w Array + ArrayList (Java)

Array	ArrayList
fixed Length data structure	Variable length Collection Class
can contain both primitives & objects	cannot contain primitives

- ArrayAdaptor (context, Layout, data source)

- List View binding (See complete code from slides)

```
ListView listView = findViewById(R.id.myListView);
ListView.setAdaptor(arrayAdaptor);
```

- Update of Records

```
arrayAdaptor.notifyDataSetChanged();
```

Instead of

- Sorting of Records

Data set name

```
Collections.sort(friendArrayList);
```

- ListView item Click Listner

```
ListView.setOnItemClickListener(new AdaptorView.OnItemClickListner)
{ @override
    public void onItemClick
```

Scanned with CamScanner

(AdapterView<?> adaptorView, View view, int i, long l)

↑ Complete List view

↑ Item

↑ Position of the view in the adapter

↑ Row ID of item

Lecture Outline

Lecture 9

24 March 2021

- Images
- Animations
- Movement, Rotation, Fade
- Videos
- Audios

- Images
  - i) Copy & paste images to drawable folder
  - ii) Use ImageView to insert the image

imageView.animate().alph(0).setDuration(2000);

0 or 1 — ms = 2 sec

imageView.animate().translationYBy(1000).setDuration(2000);

↑ downwards

translationXBy(-500)

rotation(180)

~~total~~ 360°

720° - 2 circles

Scanned with CamScanner

.rotation(720). alpha(0). setDuration(2000);

.rotation(220). scaleX(0.5f). scaleY(0.5f);

float value

rotation along x axis

- Video

From widgets add VideoView
Paste the video in raw folder under res

setVideoPath
MediaController
setAnchorView
setMediaController

```
onCreate
vv.setVideoPath("android.resource://"
        + getPackageName() + "/" + R.raw.video);
vv.start();
MediaController mc = new MediaController(this);
mc.setAnchorView(vv);
vv.setMediaController(mc);
```

name of video file

- Audio

Paste the audio in raw folder

```
MediaPlayer mp = MediaPlayer.create(this, R.raw.audio);
mp.start();
```

name of audio file

* Building Play Pause buttons for it *

```
mp.start();        mp.pause();
```

- SeekBar

Add from widgets

> setting Max value of seekbar

setMax

```
sb.setMax(40);   // will divide into 40
```

any value

(eg:

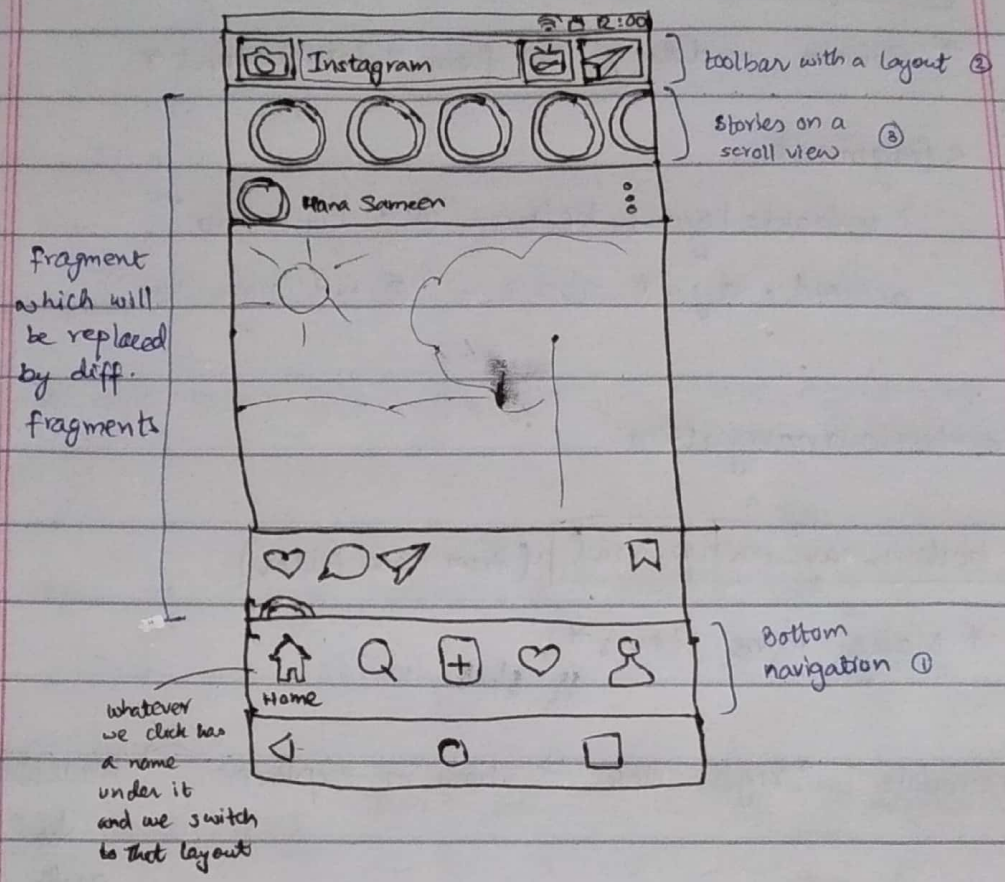
> On SeekBar ChangeListener

- on Progress Changed

Scanned with CamScanner

(Online tutorial)

Instagram Layout Design



Fragment: We use it when we want to have multiple transitions in a single activity

activity_main.xml — Toolbar, Bottom Navigation

Changed main parent to Relative Layout

< —— bottomnavigation ... —— > layout height = 56 dp (default)

* Remove Constraint Layout attributes *

android : layout_alignParentBottom = "true"

* Drop toolbar view on Layout. *

Scanned with CamScanner

colors.xml (from values folder)

Define Colours.

ctrl + tab
to move back to activity_main.xml

activity_main.xml

* Remove padding top from RelativeLayout *

< fragment
android: layout_below = "@+id/toolbar"
android: layout_above = "@+id/nav_view"

< bottomnavigation

bottom_nav_menu.xml (from menu folder)

* Make more items *

drawable → right click → show in explorer → drawable
↓
paste the images

Max size of images should be 520 x 720 (medium sized)

< toolbar

at most places

android: elevation = "6dp"

Scanned with CamScanner

- How to customize/stylize font?

- How to remove by default toolbar at top?

values → styles.xml

```
<style name="AppTheme" parent="Theme.AppCompat.
                                Light.NoActionBar">
```

- How to change colour of status bar?

```
<color name="statusbarcolor">#CECECE</color>
```

<u>For Fragments</u> Java → com.___.___ → ui → dashboard
→ ⋮
→ ⋮

Scanned with CamScanner

Lecture 10

28 March 2021

SQLite

open source database, RDBMS, not case-sensitive, local database, Android has SQLite built-in.

To create & upgrade a database in our android app, we create a subclass of SQLiteOpenHelper class, then, in the constructor of the subclass call the super() method.

① Create Java Class (named DatabaseHelper) (in the (package) folder of MainActivity)

```
import android.database.sqlite.SQLiteOpenHelper;
```

② public class DatabaseHelper extends SQLiteOpenHelper

③ Implement both the methods & create constructor super method (first option)

```
{
```

for our identification — case don't matter

④ public static final String DATABASE_NAME = "Student.db";

" " " " TABLE_NAME = "Student_table";

" " " " COL_1 = "ID";

" " " " COL_2 = "Name";

" " " " COL_3 = "SURNAME";

" " " " COL_4 = "MARKS";

⑤ public DatabaseHelper (Context context) ← This constructor will create database

```
{
    super( context, DATABASE_NAME, null, 1 )
}
```

(name, factory, version (say))

Scanned with CamScanner

⑥ public void onCreate (SQLiteDatabase db) ← creates a table

```
{
    db.execSQL("create table" + TABLE_NAME +
```

↑ executes whatever query we pass in it.

```
        "(ID INTEGER PRIMARY KEY AUTOINCREMENT,
        NAME TEXT, SURNAME TEXT, MARKS INTEGER)"
        );
}
```

⑦ public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)

```
{
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
    onCreate(db);
}
```

⑧ MainActivity.java

```
public class MainActivity ...
{
    DatabaseHelper myDB;
    onCreate ...
    {
        myDb = new DatabaseHelper(this);
```

view → Tool Windows → Device File Explorer

Scanned with CamScanner

Now,

▷ data

▷ data

All programs run on device/emulators

▷ ——. org. databaseApp (our current program's package name)

⇓

can be found from manifest file

▷ database

Student.db

* To view this database, export it to the local PC *

Select the database (Student.db)

then select "pull a file from device" button

* If don't have SQLite installed on Windows use chrome extension SQLite Manager to view it *

○ How to insert data into the created database?

① * Design UI *

② Database Helper.java

(i) public boolean insertData (String name, String surname, String marks)

{

(Readable & writable)

(ii) SQLiteDatabase db = this.getWritableDatabase();

Scanned with CamScanner

(iii) ContentValues contentValues
= new ContentValues();

(iv) contentValues.put(COL_2, name);
contentValues.put(COL_3, surname);
contentValues.put(COL_4, marks);

long result =
(v) db.insert(TABLE_NAME, null, contentValues);
← this method will return -1 in case of error.

```
if (result == -1)
    return false;
else
    return true;
```

MainActivity.java

```
btnAdd.setOnClickListener(
bool isInserted = myDb.insertData(
        etName.getText().toString(),
        etSurname.getText().toString());

if (isInserted == true)
    Toast.makeText(MainActivity.this, "Data Inserted",
        Toast.LENGTH_LONG).show();
else
    Toast.makeText(MainActivity.this, "Data not inserted",
    Toast.LENGTH_LONG).show();
```

Scanned with CamScanner

[Sir's Lectures]

SQLite (Insertion; Video 1, 2)

Youtube

first video: * Designing UI *

second video: * Developing Model Class *

Add new java class (here named CustomerModel) in the package (folder) of MainActivity)

```
public class CustomerModel
{
    private String name;
    private int age;
    private boolean isActive;
    private int id; // will be automatically assigned
```

[on empty place] → right click → generate... → constructors

" → " → " → Getter & Setters

" → " → " → toString()

Main Activity.java

```
CustomerModel cm;
btnAdd.setOnClick...
{
    cm = new CustomerModel(
            etName.getText().toString(),
            Integer.parseInt(etAge.getText().toString()),
```

Scanned with CamScanner

[Sir's lecture]

sActivityStatus.isChecked(), 1);

[CODE PROVIDED]

Sir's lecture — How to show SQLite Database Values? — lecture # 11, 30 March 2021

video 3 — DatabaseHelper.java

```
public List<CustomerModel> getAllRecords()
{
    List<CustomerModel> myList
            = new ArrayList<>();

    String query = "SELECT * FROM " + CUST_TABLE;
    SQLiteDatabase DB = this.getReadableDatabase();
Cursor c = DB.rawQuery(query, null);

    if (c.moveToFirst())
    {

        do {
            String custName = c.getString(1);
            int custAge = c.getInt(2);
            Boolean isActive = c.getInt(3) == 1 ? true : false;
                cz we dont have for boolean

            CustomerModel newCM
                = new CustomerModel(custName, custAge,
                                    isActive, 1);
```

Scanned with CamScanner

[Sir's lecture]

```
    myList.add(newCM);
  } while (c.moveToNext());

  c.close();
  DB.close();
  return myList;
}
```

MainActivity.java

```
MainActivity
{
    ArrayAdapter<CustomerModel>
        arrayAdapter;

btnViewAll.setOnClick---
{
  DatabaseHelper dbHelper = new DatabaseHelper(MainActivity
                                .this);

  List<CustomerModel> allCust = dbHelper.getAllRecords();

/*Toast.makeText
  (MainActivity.this, allCustomers.toString(),
      Toast.LENGTH_LONG).show();  */

  arrayAdapter = new ArrayAdapter<CustomerModel>
          (MainActivity.this, android.R.layout.
          simple_list_item_1, allCust);

lvCustomer.setAdapter(arrayAdapter);
```

Scanned with CamScanner

[Online lecture]

O How to show SQLite Database values?

DatabaseHelper.java

① public Cursor getAllData()
{
② SQLiteDatabase db = this.getReadableDatabase();
③ Cursor result = db.rawQuery("Select * from " + TABLE_NAME, null);

return result;
}

MainActivity.java

btnViewAll.setOnClick...
{
cursor objects have some properties as shown next
④ Cursor result = myDb.getAllData();

⑤ if (result.getCount() == 0)
{
we make this method
showMessage("Error", "Nothing Found");
return;
}

⑥ StringBuffer buffer = new StringBuffer();
while (result.moveToNext())
{
buffer.append("Id:"+result.getString(0)+"\n");

```
buffer.append("Name : " + result.getString(1) + "\n");
buffer.append("Age : " + result.getString(2) + "\n");
buffer.append("Activity Status:" + result.getString(3) +
                                    + "\n\n");
}
```

(7) `Result.close();`

(8) `DB.close();`

```
showMessage("Data", buffer.toString());
}
```

outside of everything

```
public void showMessage(String title, String Message)
{
    AlertDialog.Builder builder
            = new AlertDialog.Builder(this);

    builder.setCancelable(true);
    builder.setTitle(title);
    builder.setMessage(Message);
    builder.show();
}
```

→ so we can close the dialogue.

---

(2) How to update values in SQLite Database table?

(1) *Designing UI*

- Insert ID textView, so we can specify what is to be updated
- Add update button

Scanned with CamScanner

DatabaseHelper.java

```
(2) public boolean updateData (String id, String name, ...)
{
(3)   SQLiteDatabase db = this.getWritableDatabase();
(4)   ContentValues cv = new ContentValues();

(5)   cv.put (COL_1, id);
      cv.put (COL_2, name);
      cv.put (COL_3, age);
      cv.put (COL_3, activityStatus);

(6)   db.update (TABLE_NAME, cv, "ID=?",
                 new String[]{id});

}
```

returns the number of rows affected

MainActivity.java

```
btnUpdate.setOnClick...
{
  boolean isUpdate = myDb.updateData
        (etId.getText().toString(),
        etName.getText().toString(),
        etAge.getText().toString(),
        sActivityStatus.isChecked().toString()
        )

if (isInserted == true) { Toast ... }
else { Toast ... }
```

Scanned with CamScanner

0 How to delete values in SQLite Database?

1 *Designing UI*

Add delete button.

DatabaseHelper.java

2 public Integer deleteData (String id)

{

3 SQLiteDatabase db = this.getWritableDatabase();

4 return db.Delete(TABLE_NAME, "ID = ?", new String[] {id})

returns no. of rows deleted. 0 if none.

this will be passed here

}

ActivityMain.java

btnDelete.SetOnClick...

{

Integer deletedRows = myDb.deleteData

(etID.getText().toString());

if (deletedRows > 0)

toast...

else

toast...

Scanned with CamScanner

lecture topic:

Drawer Navigation

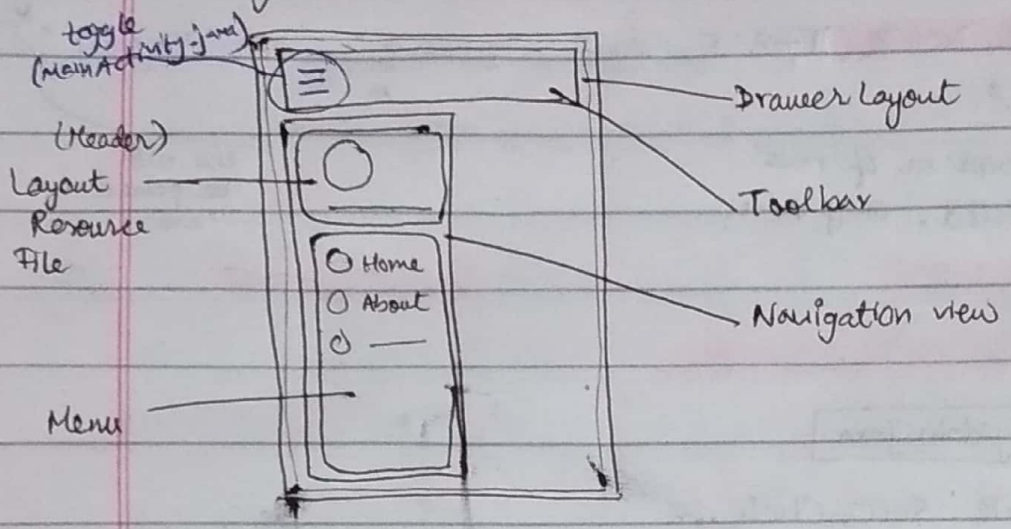
Header
Menu bar

lecture 12
4 April 2021

- Vector Asset

drawable → right click → new → Vector Asset

- Navigation Drawer (Custom)



☆ i) no actionbar (In style.xml theme) ii) Drawer Layout (as base)

iii) Make your own toolbar

iv) Navigation view v) toggle bar (not in design. (xml) we will make it thru code)

vi) Menu in navigation view

vii) (Header of navigation view) Layout Resource File ☆

(1) File → File Structure → Dependencies → All Modules

→ [+] → Library dependency → *search "design"*

→ *select com.android.support* → [OK] → [APPLY]

→ [OK]

OR

build gradle

implementation 'com.google.android.material:material:1.0.0'

(2) ★ Explanation

(i) themes.xml

parent = "Theme. Material Components. DayNight . NoActionBar"

(ii) activity_main.xml

<androidx.drawerlayout.widget.DrawerLayout
android:id="@+/drawer" tools:openDrawer="start"

(iii) activity_main.xml

```
<RelativeLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    androidx.appcompat.widget.toolbar
    <~~Toolbar~~
        android:layout_width="match_parent"
        "      "  height="56dp"
        "      id = "@+id/toolbar"
        "      title = "Nav Drawer Demo"
        "      background = "@color/ ____"
        "      titleTextColor = "@color/ ____"
    />
</RelativeLayout>
```

(iv) res → new → android resource directory →
select Resource type as menu → [ok]

menu → New → Menu Resource File → *File name: iconmenu.xml *

drawable → New → vector asset

iconmenu.xml

```
<menu ...

    <item
        android:id = "@+id/menu_home"
        android:icon = "@drawable/ic_baseline_home_24"
        android:title = "Home"
    />

    :

</menu>
```

(vii) layout → layout resource file → * File name: nav_header ~~NavHeader~~ *

nav_header.xml

```
<androidx.appcompat.widget.LinearLayoutCompat ...
    android:layout_width = "match_parent"
    android:layout_height = "wrap_content"
    android:orientation = "vertical" >

    <ImageView
        android:layout_width = "wrap_content"
```

Scanned with CamScanner

```
android: layout_height = "wrap_content"
android : src = "@drawable/ _____ "
```

```
<TextView
    android : layout_width = "wrap_content"   (size we want or)
    android : layout_height = "wrap_content"
    android : textColor = "@color/ _____ "
    android : text Size = "30dp"
    android : text = "Star Nav App" />
```

(iv) activity_main.xml

outside of relative layout

```
<com.google.android.material.navigation.NavigationView
    android: layout_width = "wrap_content"
    "      "   height = "match_parent"
                id = "@+id/navmenu"
    "
    app : menu = "@menu/iconmenu"
    app: headerLayout = "@layout/nav_header"
    android: layout_gravity = "start" />
```

(v) MainActivity.java

```
MainActivity ... {
```

```
NavigationView navView;
ActionBarDrawerToggle toggle;        ( ≡ )
DrawerLayout drawerlayout;
androidx.
~~Toolbar~~ toolbar;

protected void onCreate ...
{
    toolbar = findViewsById(R.id.toolbar);
    setSupportActionBar(toolbar);
    navView = findViewById(R.id.navMenu);
    drawerLayout = " (" " drawer);

toggle = new ActionBarDrawerToggle
        (this, drawerLayout, toolbar, R.string.open,
                                    R.string.close);

drawerlayout.addDrawerListener(toggle);

toggle.syncState();
navView.setNavigationItemSelectedListener(
  new Navigation ...
{
    switch (item.getItemId())
    {
      case R.id.menu_home:
            drawerLayout.closeDrawer(GravityCompat.START);
            break;
      ⋮
    } return true;
```

Scanned with CamScanner