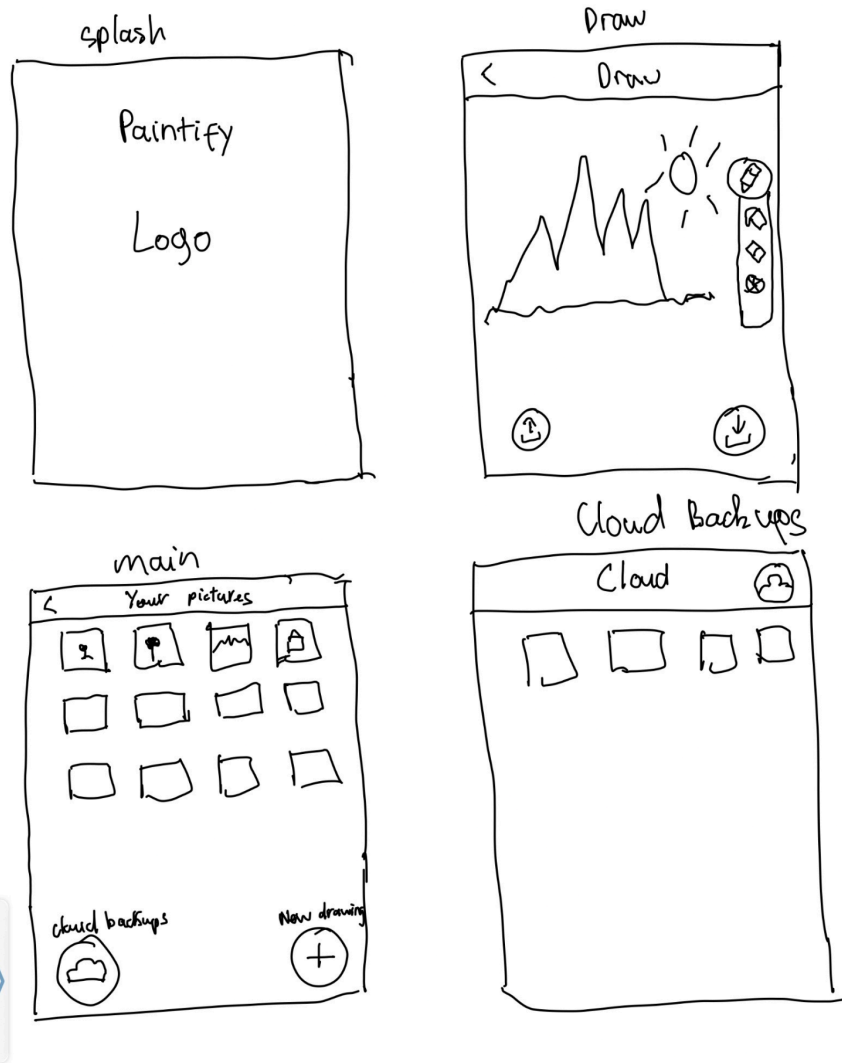
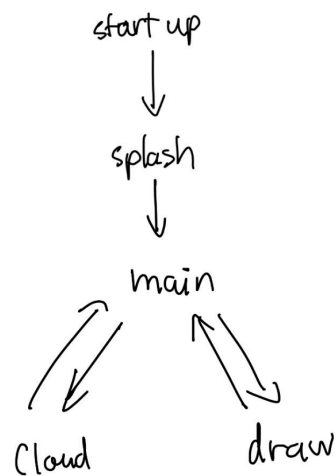


Paintify - Implementation Plan - Dustin, Ian, Nolan



* delete by holding _____

Navigation flow



Layouts to create (Phase 1)

1. `activity_splash.xml` – app logo + timed/instant route
2. `activity_main.xml` – contains a button to navigate to `activity_draw.xml` (
3. `activity_draw.xml` – full-screen canvas + top app bar (Back disabled), bottom tool strip:
 - Color button (opens simple color picker dialog)
 - Size button (opens simple slider dialog)
 - Shape toggle (Freehand / Line / Rect / Circle)
 - Undo / Redo
4. `dialog_color_picker.xml` – small grid of preset swatches (e.g., black, white, red, green, blue, yellow, purple)
5. `dialog_brush_size.xml` – slider + live preview

(No `activity_main.xml` or `activity_cloud.xml` in Phase 1, kept out to avoid scope creep.)

Classes to implement (Phase 1)

UI / View

- `SplashActivity`
- `DrawActivity`
- `DrawingCanvasView` (custom view): handles touch, renders strokes/shapes, supports undo/redo

Dialogs

- `ColorPickerDialogFragment`
- `BrushSizeDialogFragment`

State (MVVM)

- `DrawingViewModel` (uses `SavedStateHandle`):
 - `currentTool` (enum: PEN, LINE, RECT, CIRCLE)
 - `penColor` (int)
 - `penSizePx` (float)
 - `strokes` (immutable list of `Stroke` / `ShapeCommand`)
 - `redoStack`
 - intent methods: `beginStroke()`, `extendStroke()`, `endStroke()`, `undo()`, `redo()`, `setColor()`, `setSize()`, `setTool()`

Domain / Models

- `ToolType` (enum)
- `Stroke` (data: list of points, paint params)
- `Shape` (sealed: `LineShape`, `RectShape`, `CircleShape` with start/end points + paint)
- `PaintParams` (color, width, style)

Utilities (Phase 1)

- `TouchToDrawingMapper` (optional helper to map `MotionEvent`s → model updates)

(No repository, no storage, no share manager in Phase 1.)

Unit tests to write (Phase 1)

Local JVM tests (fast):

1. `DrawingViewModelTest`
 - **Tool selection** updates correctly
 - **Color/size** setters update state
 - **Stroke lifecycle**: begin→extend→end yields a stored stroke with expected points/params
 - **Shapes**: for tool LINE/RECT/CIRCLE, begin→end creates the correct shape with derived bounds
 - **Undo/Redo**: stacks behave correctly (no crash on empty undo/redo)
 - **Rotation-safety**: simulate `SavedStateHandle` restore (construct with prefilled state) and verify state persists

2. `PaintParamsTest`

- Width and color propagate to new strokes/shapes

3. (Optional) `TouchToDrawingMapperTest`

- MOVE throttling / point accumulation behavior

Instrumented (optional if time tight):

4) `SplashNavigationTest` – app launches, shows splash, navigates to `DrawActivity`

5) `DrawUiSmokeTest` – tap color/size/shape controls don't crash; basic draw gesture shows something (pixel diff or count of strokes via ViewModel)

Task assignment (Phase 1)

Member A (Nolan) – Canvas & Interaction

- `DrawingCanvasView` (touch → ViewModel intents, rendering)
- Shape drawing (line/rect/circle preview while dragging)
- Undo/Redo wiring from UI to VM
- Unit tests: `DrawingViewModelTest` (stroke lifecycle, shapes)

Member B (Dustin) – State & Dialogs

- `DrawingViewModel` + `SavedStateHandle` integration
- `ColorPickerDialogFragment`, `BrushSizeDialogFragment`
- Hook dialogs to update VM; observe VM to reflect current settings
- Unit tests: tool/color/size changes, undo/redo edge cases

Member C (Ian) – App Shell & Splash

- `SplashActivity` (1–1.5s delay or immediate route), theming
 - `DrawActivity` layout (top app bar + bottom tool strip)
 - Optional instrumented tests (splash nav, smoke)
 - Project scaffolding: ktlint/detekt config (optional if time)
-

Implementation order (Phase 1)

Day 0–1: Setup

1. Repo + branch protections, all members & TAs added
2. Project skeleton (single `app` module), base theme
3. `SplashActivity` → `DrawActivity` navigation

Day 1–2: MVVM skeleton

4. Create `DrawingViewModel` with state fields + basic setters
5. `activity_draw.xml` with controls (no behavior yet)
6. Wire controls to VM (observe LiveData/StateFlow)

Day 2–4: Drawing core

7. Implement `DrawingCanvasView`:
 - PEN: record points on MOVE
 - Shapes: show preview on drag, commit on UP
 - Render strokes/shapes from VM
8. Add Undo/Redo plumbing

Day 4–5: Customization

9. Implement `ColorPickerDialogFragment` + `BrushSizeDialogFragment`
10. Connect to VM (`setColor`, `setSize`)

Day 5–6: Tests & polish

11. Write/finish JVM unit tests listed above
 12. (If time) 1–2 instrumented tests (splash nav + UI smoke)
 13. Accessibility labels on buttons; min touch sizes
-

Minimal file list (Phase 1)

app/

src/main/java/.../ui/

SplashActivity.kt

DrawActivity.kt

DrawingCanvasView.kt

dialogs/ColorPickerDialogFragment.kt

dialogs/BrushSizeDialogFragment.kt

src/main/java/.../vm/

DrawingViewModel.kt

src/main/java/.../model/

ToolType.kt

PaintParams.kt

Stroke.kt

Shape.kt (sealed: LineShape, RectShape, CircleShape)

src/test/java/.../

DrawingViewModelTest.kt

PaintParamsTest.kt

src/androidTest/java/.../ (optional)

SplashNavigationTest.kt

DrawUiSmokeTest.kt

src/main/res/layout/

activity_splash.xml

activity_draw.xml

dialog_color_picker.xml

dialog_brush_size.xml

README.md (how to run, what's tested)