Лабораторийн тайлан №4

Апплекэйшны архитектуртай танилцан котлин программчлалын хэлэнд активитиг хэрэгжүүлэх

МТЭС, МКУТ, Программ хангамж

Г.Батням, 22B1NUM5578

1.Ажлын зорилго

Энэ лабораторийн ажлаар апплекэйшны архитектурын талаарх ойлголттой болон зөв зохицуулж activity lifecycle – ийг мөн дэлгэц хоорондын шилжилтийг зөв тодорхойлж хэрэгжүүлнэ.

2.Онолын судалгаа

**Activity lifecycle** – **activity үүсэхээс эхлээд устаж, систем тухайн activity-ийн нөөцийг сэргээх хүртэл үргэлжилдэг ба эхлээд устах хүртэлх** янз бүрийн төлөвүүдээс бүрдэнэ. Хэрэглэгч нэг activity рүү орж, гарч байх үед тус activity нь амьдралын циклийн өөр өөр төлөвүүд рүү шилждэг. **Activity устах үед** үйлдлийн систем (OS) түүний санах ойг сэргээдэг.

**RememberSaveable** – ашиглан **төхөөрөмжийн тохиргоо өөрчлөгдөх үед алдагдаж болзошгүй өгөгдлийг хадгалах**

**Activity** – Бүх **activity** нь **амьдралын цикл (lifecycle)** гэж нэрлэгдэх үйл явцтай байдаг. Энэ ойлголт нь **амьтан, ургамлын амьдралын мөчлөгтэй төстэй** бөгөөд үүнийг эрвээхэйний амьдралын мөчлөгөөр дүрслэж болно.Эрвээхэйний амьдралын мөчлөгөөс ялгаатай нь activity нь төлвүүдийн **хооронд урагш, хойш шилжиж** болдог. Android нь activity төлвийн шилжилт болгоны үед эдгээр callback-уудыг автоматаар дууддаг.Хөгжүүлэгчийн хувьд та эдгээр **callback-уудыг override** хийж, activity lifecycle-ийн өөрчлөлтөнд хариу үйлдэл үзүүлэх боломжтой.

**Logcat** бол **Android Studio-ийн логийн консол** бөгөөд Android систем болон таны апп-аас гарсан **мэдээллийг харуулдаг**.

🔹 **Log.d() хэрхэн ажилладаг вэ?**

* d = **debug**, энэ нь **алдаа засах зориулалттай**.
* **Эхний параметр** → TAG (таны аппын ангиллыг тодорхойлно).
* **Хоёр дахь параметр** → msg (жинхэнэ лог мессеж).

***Зураг2 – ийн тайлбар:***

**Апп эхлэхэд дараах дарааллаар ажиллана:**  
✅ onCreate() → **Activity үүснэ**  
✅ onStart() → **Activity харагдана**  
✅ onResume() → **Activity foreground-д орж, хэрэглэгчтэй харилцах боломжтой болно**

**Хэрэв хэрэглэгч Home товч дараад буцаад орвол:**  
✅ onPause() → **Activity түр зогсоно**  
✅ onStop() → **Activity харагдахгүй болно**  
✅ onRestart() → **Activity дахин эхэлнэ**  
✅ onStart() → **Дахин харагдана**  
✅ onResume() → **Дахин хэрэглэгчтэй харилцана**

**Апп хаагдах үед:**  
✅ onPause() → **Түр зогсоно**  
✅ onStop() → **Болих төлөвт орно**  
✅ onDestroy() → **Устаж, санах ойг чөлөөлнө**

**Тохиргооны өөрчлөлт(*configuration change*)** – нь төхөөрөмжийн төлөв байдал маш их өөрчлөгдөж, систем хамгийн хялбар аргаар асуудлыг шийдэхийн тулд үйл ажиллагааг бүрэн зогсоож, дахин бүтээх үед тохиолддог. **Тохиргооны өөрчлөлтөөс өгөгдөл хадгалахыг хүсвэл**, хувьсагчийг rememberSaveable ашиглан зарлана.

**Тохиргооны өөрчлөлт тохиолдоход**:

* Android нь **бүх lifecycle-ийн зогсоох callback-уудыг** ажиллуулна.
* **Дараа нь үйл ажиллагааг шинээр эхлүүлж**, бүх lifecycle-ийн эхлүүлэх callback-уудыг ажиллуулна.

**Тохиргооны өөрчлөлтийн дараа Android үйл ажиллагааг onCreate()-с эхлүүлнэ.**

**Architecture –** Мобайл апп нь дотроо **ui layer , data layer** гэсэн 2 хэсгээс бүрддэг.Ui layer нь ямар нэгэн event дэлгэцэнд өөрчлөлт ороход data layer руу мэдэгддэг харин data layer нь хэрэгтэй мэдээллийг ui layer руу дамжуулдаг мөн өгөгдөл өөрчлөгдөх тоолонд ui layer өөрчлөгдөнө.

➕ **(Нэмэлт) Домэйн давхарга (Domain Layer)**

* **UI болон Data давхаргын хооронд гүүр болж өгдөг.**
* **Энэ давхарга заавал байх шаардлагагүй.**

**UI Layer Pipeline:  
 1.App Data -> UI Data  
 2.UI Data -> UI Element  
 3.User Events -> UI Changes  
 4.Repeat**

**1 урсгалтай architecture зохион байгуулалтын хувьд:**

* **View model   
  1.**Ui – д ашиглах төлөвийг харуулдаг.  
  **3.**Хэрэглэгчийн үйлдлийг зохицуулж төлөвийг шинэчилдэг.
* **UI elements   
  2.**Хэрэглэгчийн ямар нэгэн үйл явдал event – ийн талаар view model – д мэдэгддэг.  
  **4.**Төлөвт тохирох өгөгдлийг дамжуулна.

**ViewModel –** нь үйл ажиллагаа дахин үүсгэгдэх үед өгөгдлийг алдахгүй хадгалдаг.Программ дуусах үед өгөгдөл устдаг ба хурдан дахин үүсэх процессын үед өгөгдлийг хадгалдаг. **UI-д шаардлагатай өгөгдлийг хадгалж, UI-д хүргэдэг.**

**Хамгийн нийтлэг архитектурын зарчмууд:**

1. **"Separation of Concerns" (Үүргийн тусгаарлалт)**

 **Аппын функцийг тусдаа үүрэгтэй ангиудад хуваах хэрэгтэй.**

 Ингэснээр **код цэвэрхэн, ойлгомжтой, дахин ашиглахад хялбар** болно.

1. **"Drive UI from a Model" (UI-г Model-оос удирдах)**

 **UI-г Model-оос удирдах нь хамгийн зөв арга.**

 **Model** нь **өгөгдөл хадгалах, боловсруулах үүрэгтэй**.

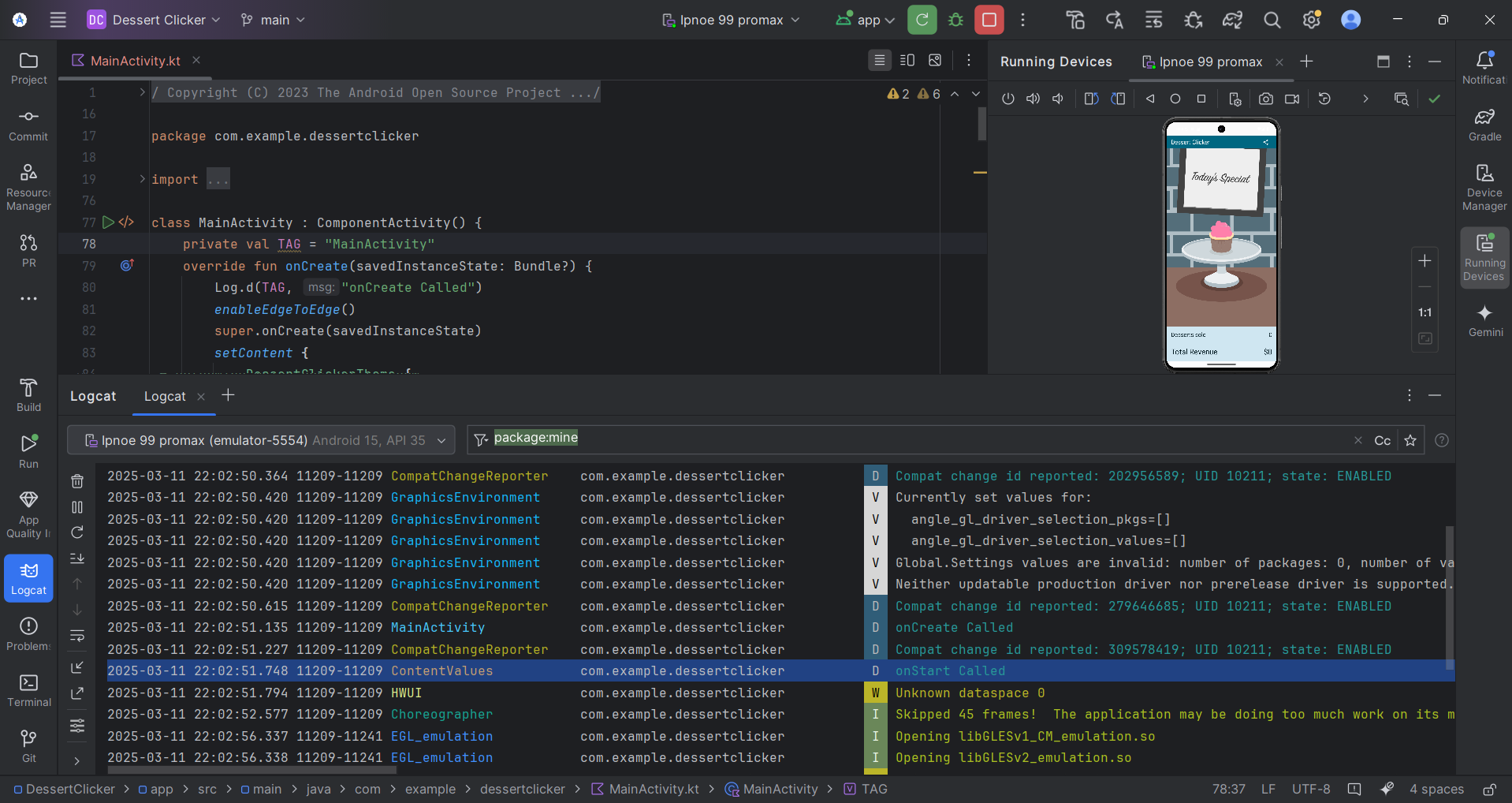
 **Model нь UI-гээс хамааралгүй** тул **аппын амьдралын мөчлөгөөс (Lifecycle) үл хамааран** ажилладаг.

**Навигац –** Шилжих шилжилтийн хүрэх газрыг тодорхой зааж өгөх ёстой.  
**NavHost –** Зорьсон газруудын хооронд шилжих үед тэдгээрийг дахин бүрдүүлж харуулдаг хоосон сав юм.  
**Route –** Шилжих шилжилтийн замыг зааж өгдөг.  
**Back Stack –** Энэ ньөмнөх дэлгэцүүдийг дарааллын дагуу хадгалаад буцсан тохиолдолд тухайн стекд байгаа өмнөх дэлгэцэнд очиж түүнийг харуулдаг.  
**Back –** Back button, up button уудыг ашиглан буцах боломжийг олгодог.

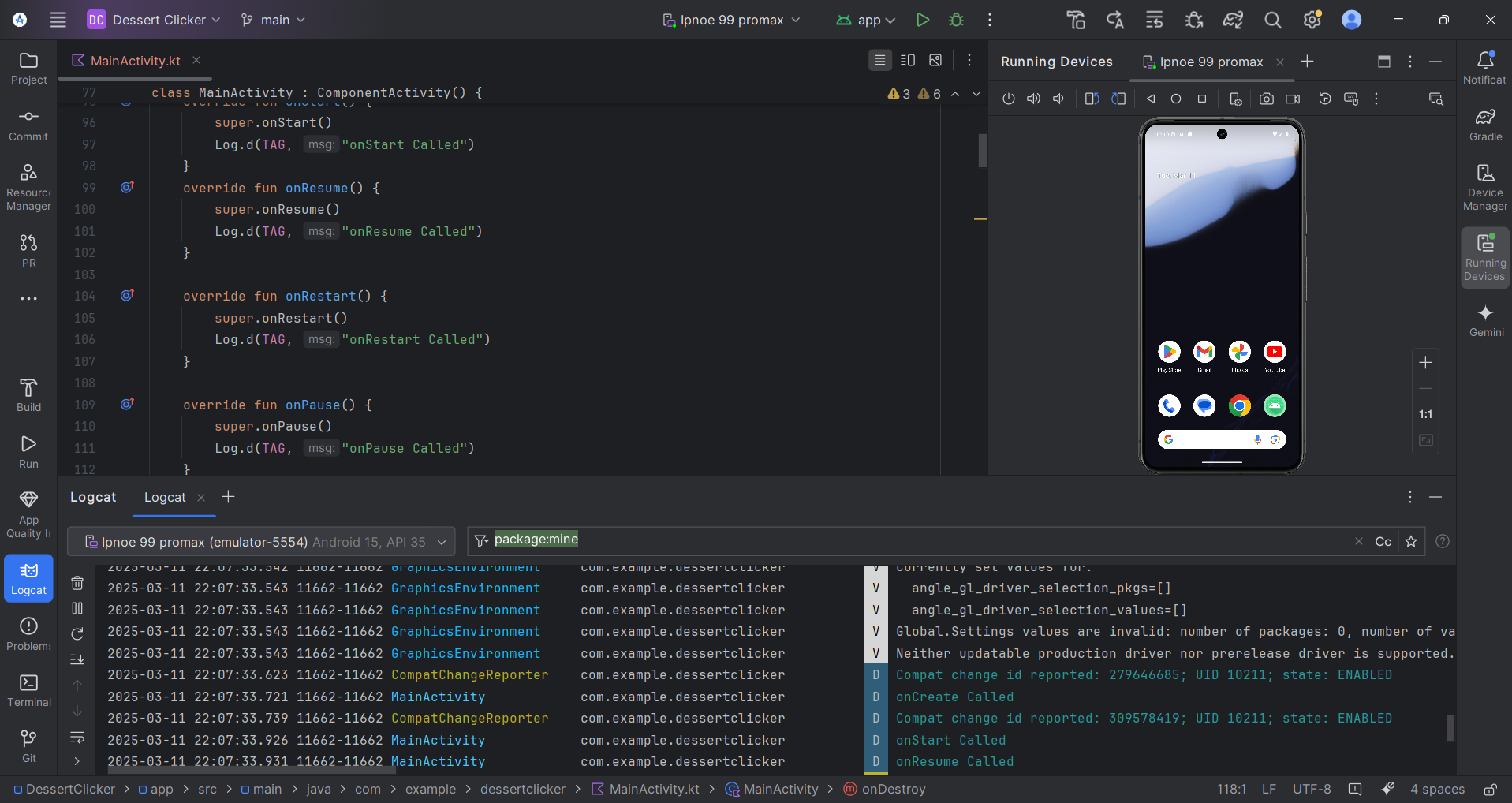
**Хийх ажилын хувьд:**🔹 NavHost composable ашиглан **замууд (routes)** болон **дэлгэцүүд (screens)** тодорхойлох  
🔹 NavHostController ашиглан дэлгэц хооронд шилжих  
🔹 **Back Stack** удирдаж, өмнөх дэлгэц рүү буцах  
🔹 Intents ашиглан мэдээллийг бусад апп руу хуваалцах  
🔹 **AppBar**-ийг өөрчлөх (гарчиг солих, буцах товч нэмэх гэх мэт)

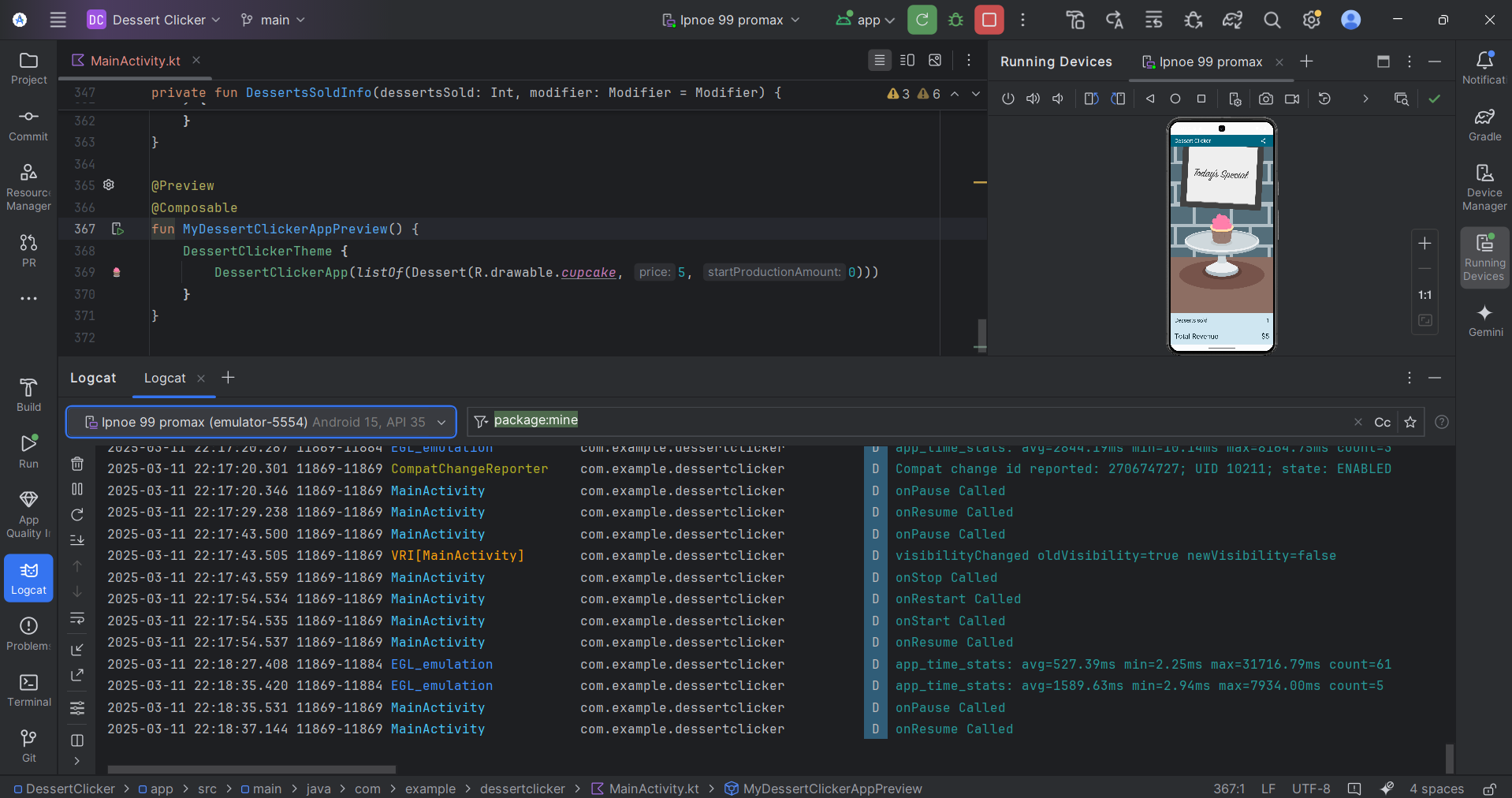
Дэлгэцийн мэдээллийн дизайны 3 нийтлэг загвар байдаг.  
1.Canonical Layouts(List – Detail) – Дэлгэцийг хуваан мэдээллийг харуулах  
2.Supporting panel(Focus – Supporting Panle) – Видео бичлэг тоглуулж байгаа тохиолдолд дэлгэцээ хуваан байршуулах  
3.Feed – Мэдээ нийтлэлд ихэвчлэн ашигладаг.  
**BreakPoint –** Дэлгэц ямар ч хэмжээтэй байсан апп – д байгаа зайны хэмжээнд тохируулан зохион байгуулахад тусална.

3.Хэрэгжүүлэлт

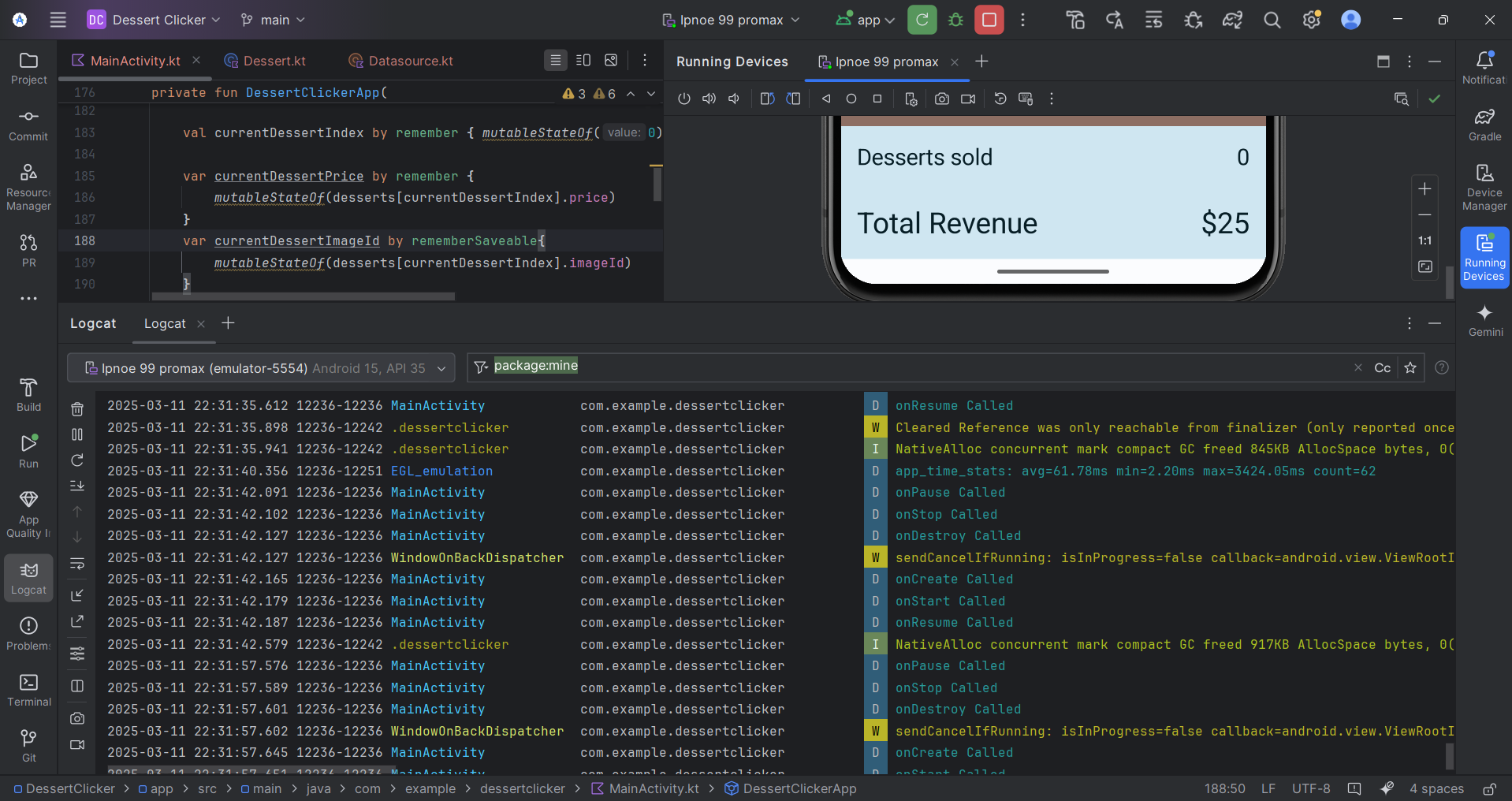


***Зураг 1*** *Аппын дахин тодорхолойгдсон функцууд ямар ямар төлөвт орсон байгааг харсан*

** ***Зураг 2*** *Аппын дахин тодорхолойгдсон функцуудыг нэмсэн*

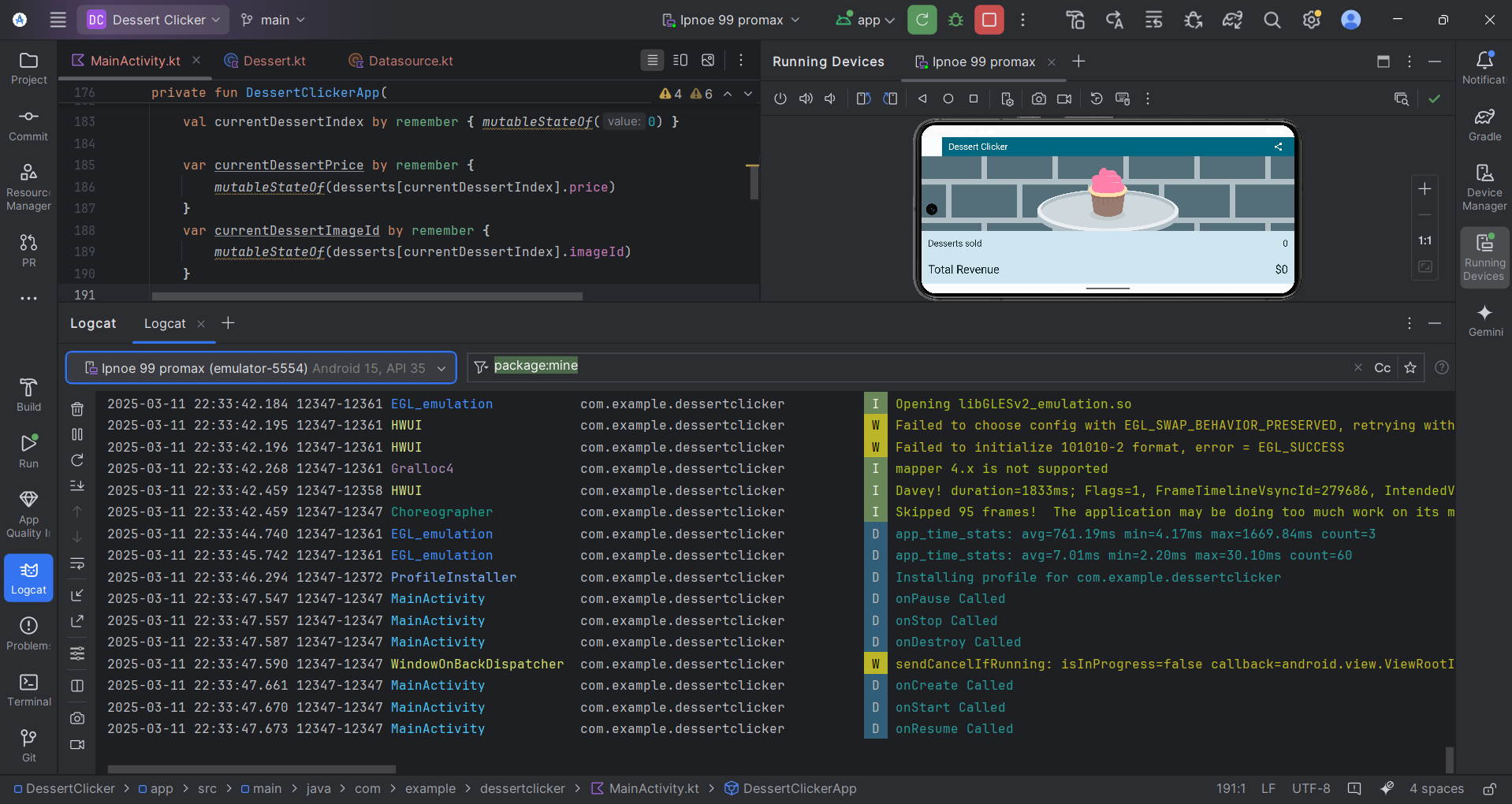
**

***Зураг 3*** *Гараад буцаж орох мөн хуваалцах товчыг дарах хооронд хэрхэн төлөв солигдож байгааг харуулж байна*

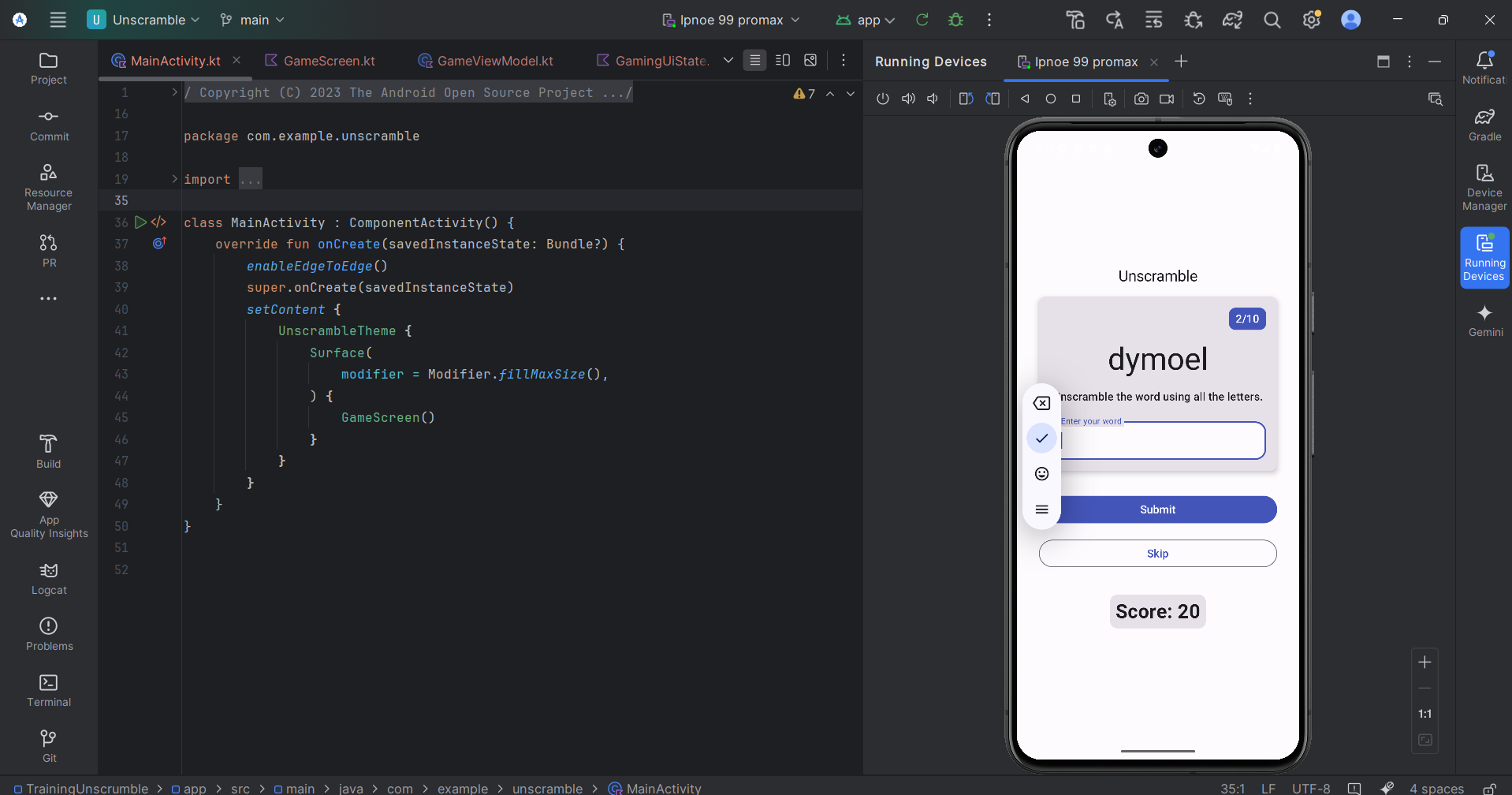
**

***Зураг 4*** *Тухайн үнийн утгыг rememberSaveable**ашиглан хэрэгжүүлсэн*

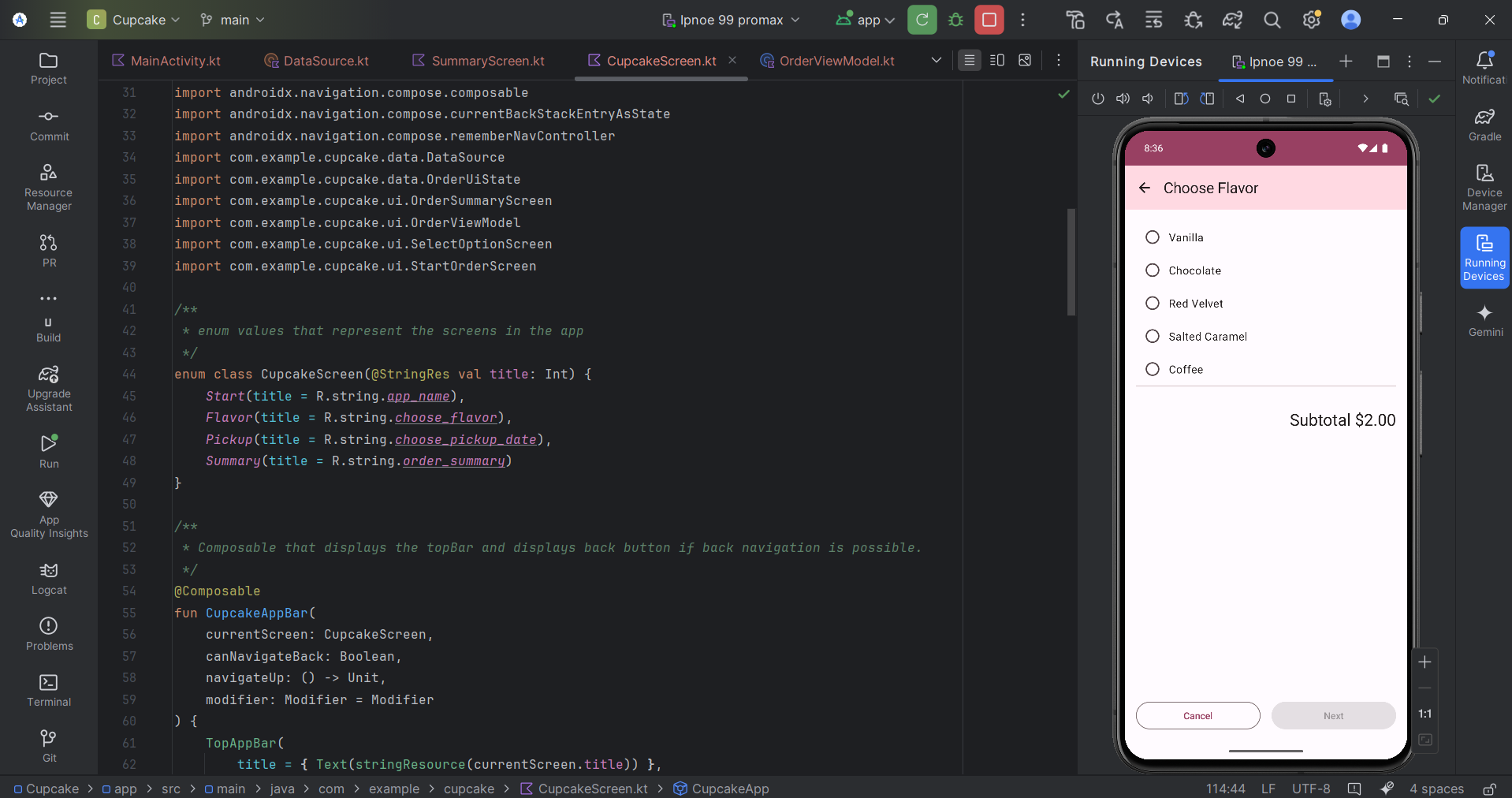
**Тайлбар  
Тухайн төлөв хадгалж буй утгаа rememberSaveable гэж тодорхойлж өгвөл дэлгэцийн өөрчлөлтлөд анхны утгаа хадгалах боломжтой.**

**

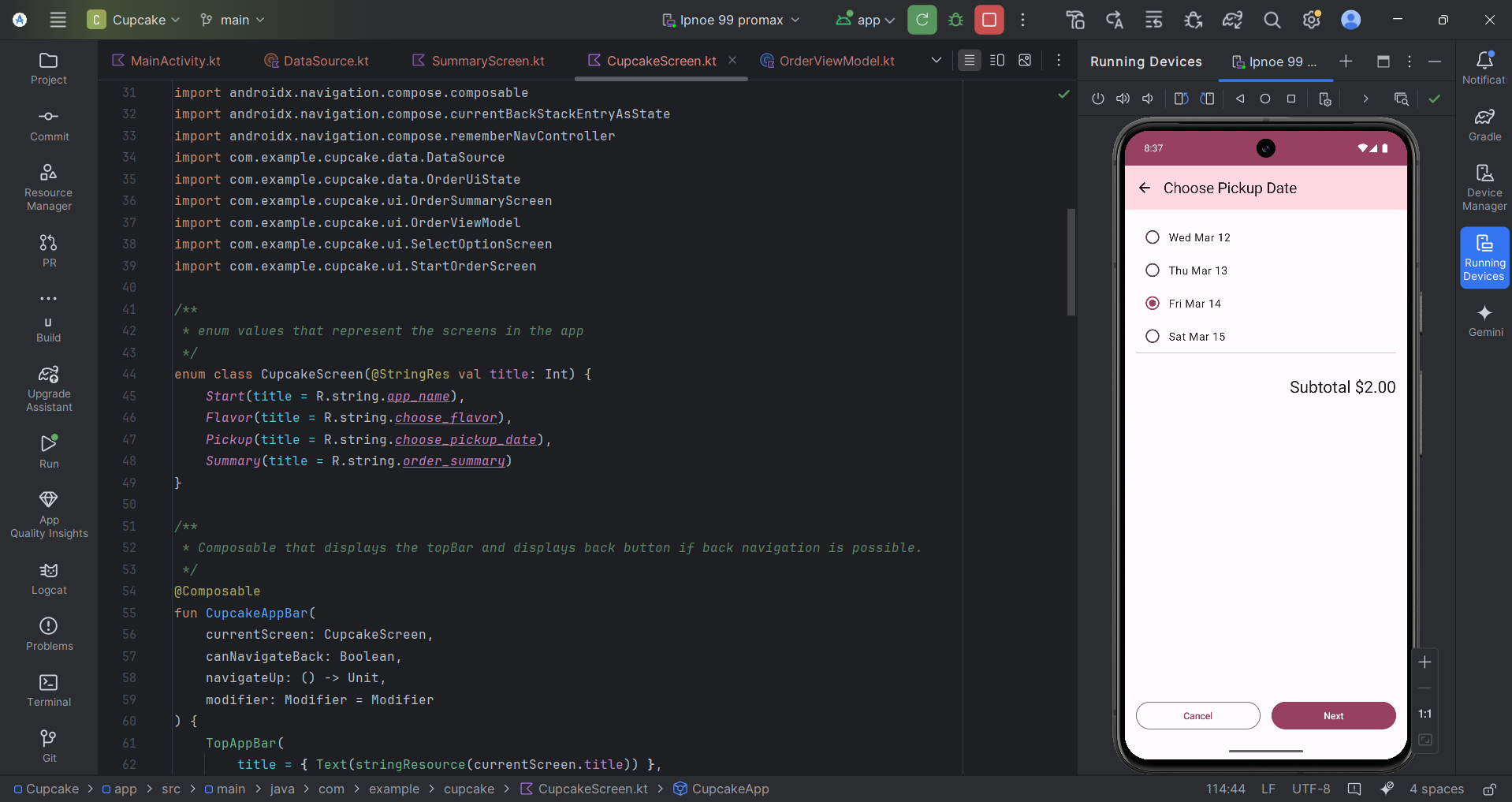
***Зураг 5*** *Тухайн үнийн утгыг remember ашиглан хэрэгжүүлсэн***Тайлбар  
Тухайн төлөв хадгалж буй утгаа remember гэж тодорхойлж өгвөл дэлгэцийн өөрчлөлтлөд анхны утгаа хадгалж чадахгүй.**

****

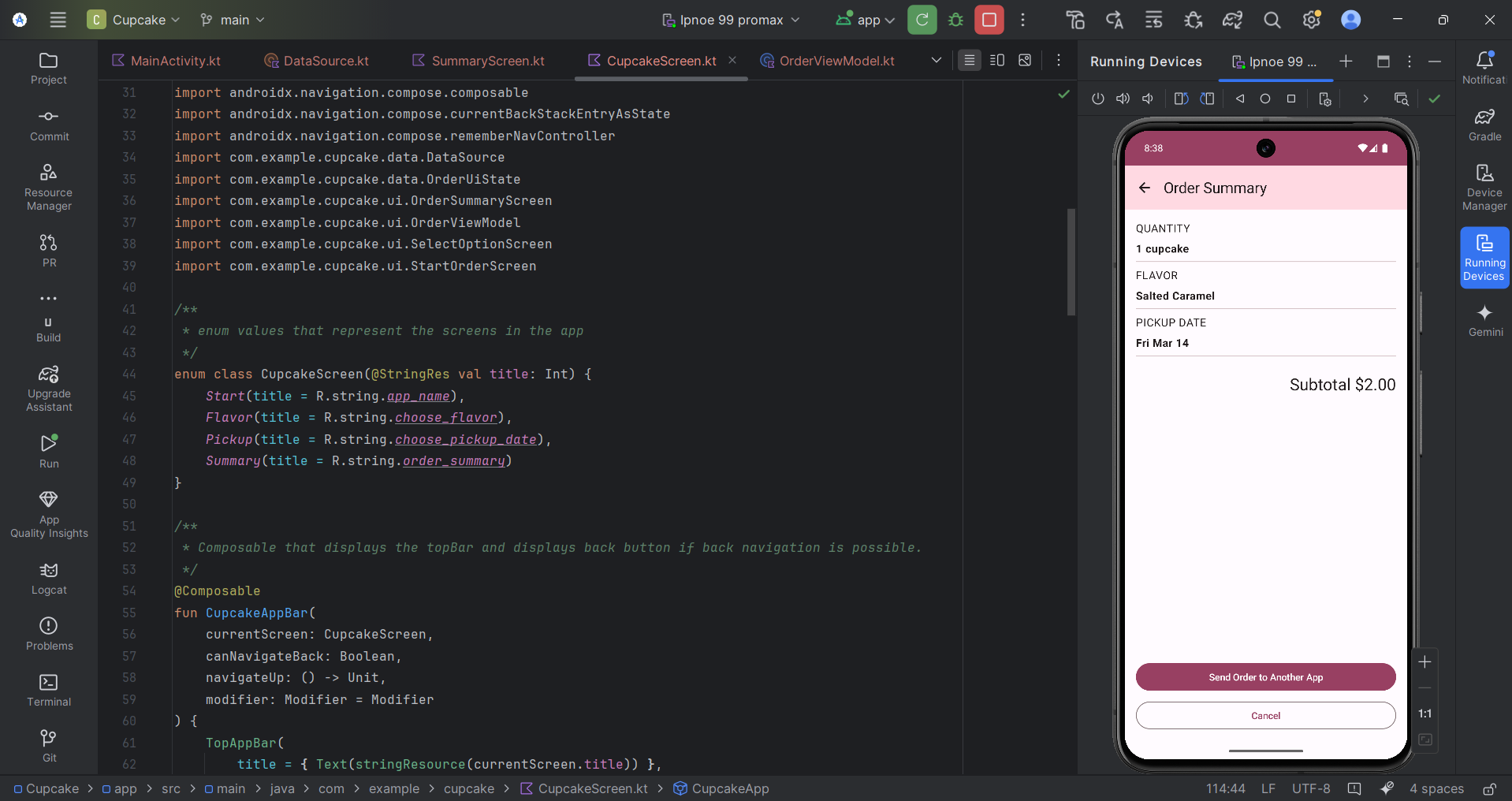
***Зураг 6*** *Uncrumble аппыг хэрэгжүүлсэн*

**

***Зураг 7*** *Cupcake эхлэл*

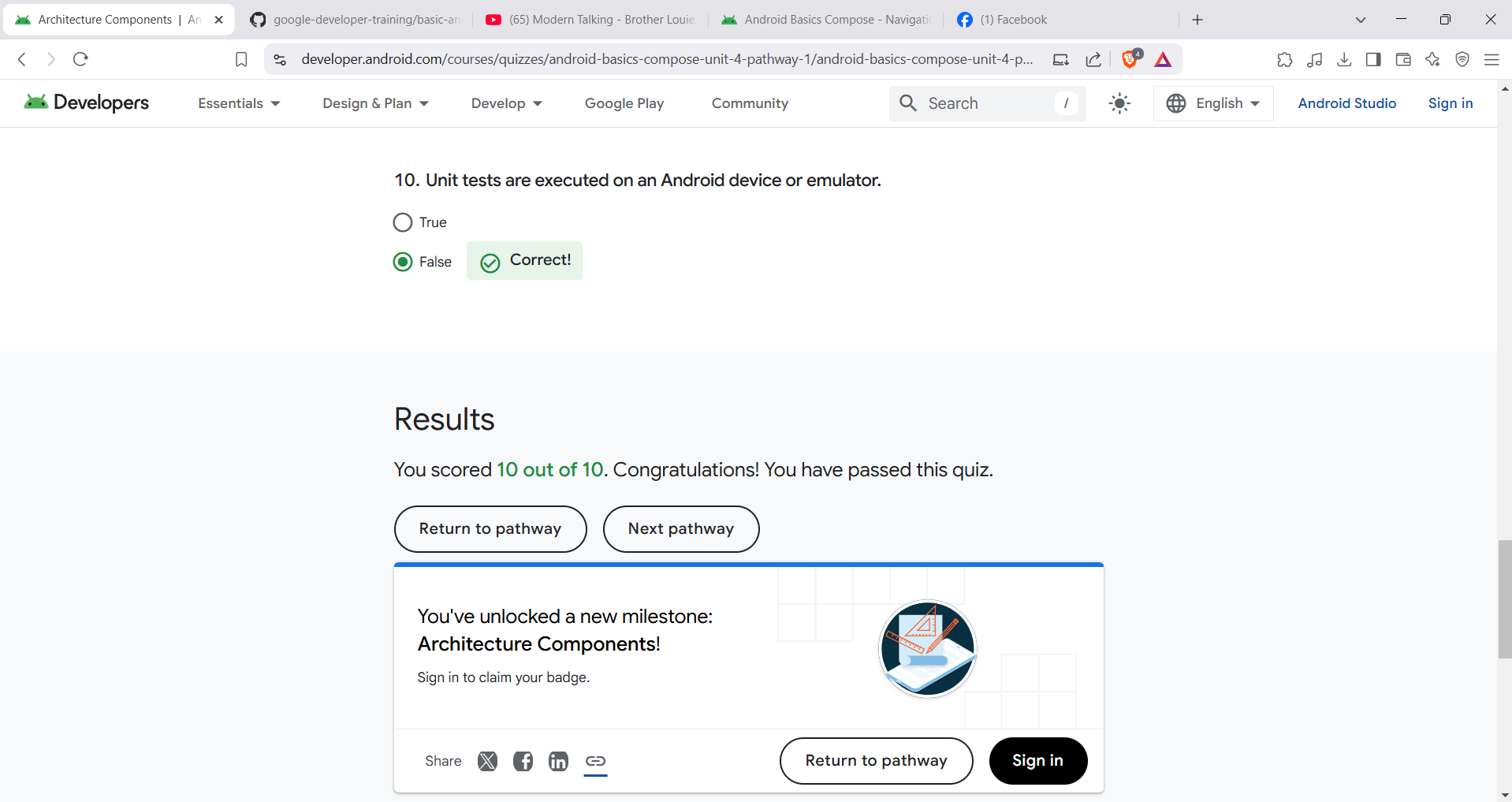
**

***Зураг 8*** *Cupcake дараагийн хэсэг*

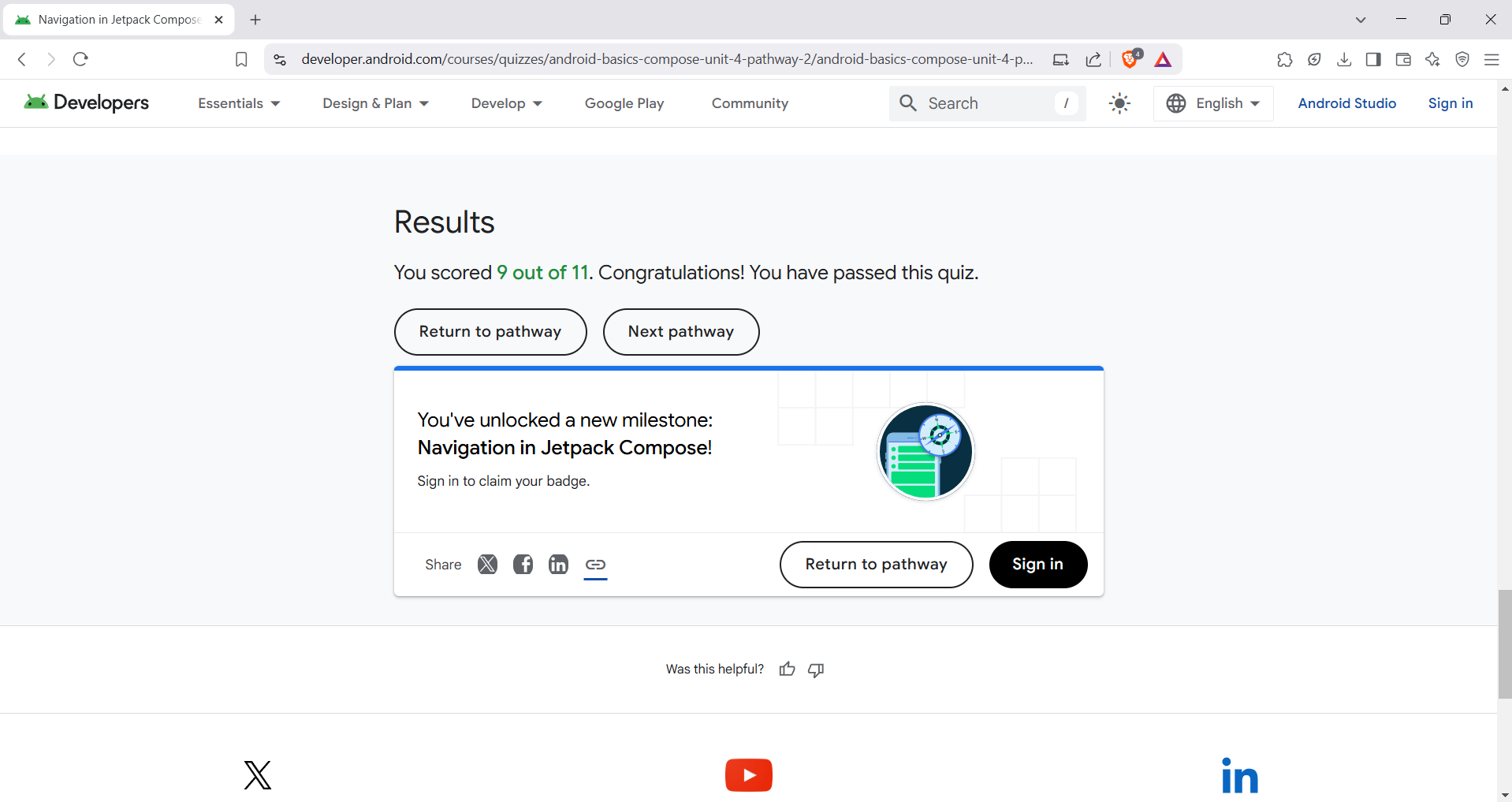
**

***Зураг 9*** *Cupcake сүүлийн хэсэг*

4. Бүлгийн асуулт



***Зураг 1*** *Эхний бүлгийн асуултуудад хариулсан*

  
 ***Зураг 2*** *Дараагийн бүлгийн асуултуудад хариулсан*

5.Дүгнэлт

Энэ лабораторийн ажлаар апплекэйшний дотоод architecture – ийн дотоод зохион байгуулалт зөв зохион байгуулж тухайн апплекэйшний дэлгэц өөрчлөгдөх эсвэл ямар нэгэн шинэ төлөвт шилжихэд хэрхэн анхны утгаа хадгаладаг байх аргатай танилцан түүнийгээ хэрэгжүүллээ.

6. Ашигласан материал

(https://developer.android.com/courses/pathways/android-basics-compose-unit-3-pathway-1, n.d.)

(https://developer.android.com/courses/pathways/android-basics-compose-unit-3-pathway-2?\_gl=1\*4nx32f\*\_up\*MQ.., n.d.)

(https://developer.android.com/courses/pathways/android-basics-compose-unit-3-pathway-3, n.d.)