Google Developer Student Clubs

Android 1주차

지난주 과제 확인

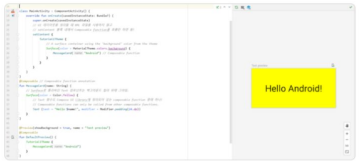
- 이하은

이하은 오후 11:05

1. Composable functions

(UI를) 구성할 수 있는 함수?

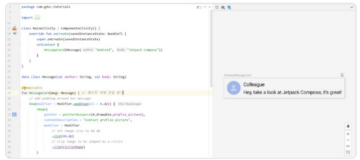
- **XML 파일을 편집하지 않고도, composable functions만 호출해주면 Compose 컴파일러가 알아서 UI 작업을 처리해준다.**
- **@Composable**를 함수 이름 앞에 붙이면, composable function을 만들 수 있다.
- @Composable 앞에 **@Preview**를 붙이면, 안드로이드 장치나 에뮬레이터에 앱을 설치할 필요 없이, **IDE 내에서 (composable function에 의해 생성된) UI 요소를 볼 수 있다.** (편집됨)



이하은 오후 11:06

2. Layout

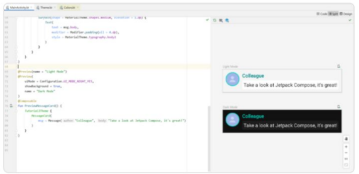
- composable function이 또다른 composable function을 호출하면, UI 요소들이 중첩되면서 **UI 계층 구조**가 만들어진다.
- 이러한 UI 요소들은 Row, Column composable로 **위치 관계를 설정**할 수 있다.
- 이미지를 추가할 때는 **Image** composable을 사용하며, **Modifier**로 이미지의 크기나 레이아웃 등을 설정할 수 있다. (편집됨)



이하은 오후 11:10

3. Material Design

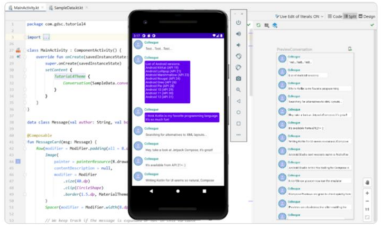
- Compose는 material design 원칙을 지원하기 위해 만들어졌고, 많은 UI 요소가 기본적으로 머티리얼 디자인을 구현한다. 이러한 머티리얼 위젯으로 앱을 스타일링 할 수 있다.
- 머티리얼 디자인은 크게 **색상, 타이포그라피 (텍스트 디자인), 모양** 이 세가지로 구성된다.
- 동일한 함수에 **여러 개의 @Preview annotation**을 붙여주면, 그에 따른 미리보기를 한번에 확인할 수 있다. (라이트 모드와 다크 모드 동시에 미리보기 가능) (편집됨)



이하은 오전 12:19

4. Lists and animations

- XML 레이아웃에서 RecyclerView를 사용하는 대신에, **Conversation 함수의 LazyColumn/LazyRow를 사용**하면 훨씬 더 간단하게 리스트를 보여줄 수 있다.
- Composable functions는 **remember**를 사용해서 메모리에 로컬 상태를 저장하고, **mutableStateOf**에 전달된 값의 변화를 추적할 수 있다. **Composables는 이러한 상태 변화에 따라 자동으로 UI가 업데이트 될 수 있도록 한다.** (편집됨)



지난주 과제 확인

- 오승언

a

오승언 오후 1:49

1. Compasable functions

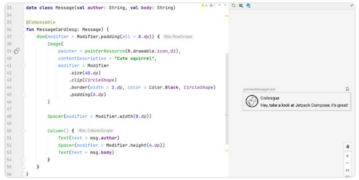
- @Composable 키워드를 이용하여 composable 함수를 구현할 수 있음
- @Preview 키워드를 이용하여 composable 함수로 만들어진 UI elements를 볼 수 있음
- $ 키워드를 이용하여 텍스트를 쉽게 만들 수 있음 (기존에 사용하던 " " + 변수명 + " " 이 아닌 "~~ $parameter ~~" 와 같이 사용할 수 있음)

a

오승언 오후 5:18

2. Layouts

- Column(), Row() 키워드를 이용하여 요소들을 배치할 수 있음 (Column - vertical, Row - horizontal)
- Image() 키워드를 이용하여 drawable의 이미지를 가져와 배치할 수 있음
- Modifier의 함수들을 이용하여 요소들의 속성을 구성할 수 있음(size, clip, border 등)
- Spacer() 함수를 이용하여 요소 사이의 공백을 만들 수 있음

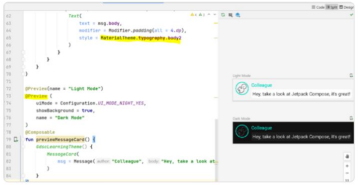


a

오승언 오후 2:18

3. Material design

- UI위젯 스타일을 쉽게 구현할 수 있음
- MaterialTheme을 활용해서 색상, 텍스트 스타일, 모양 등을 지원받을 수 있음
- 하나의 composable 함수에 두 가지의 preview를 적용할 수 있음(Light mode, Dark mode)
- 튜토리얼에는 나오지 않지만, 터치 피드백, 곡선 모션, 뷰 상태 변경 등을 간편하게 구현할 수 있다고 함

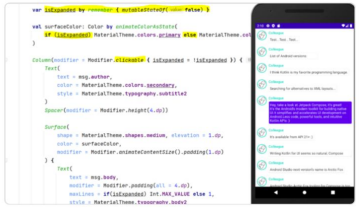


a

오승언 오후 2:18

4. Lists and animations

- LazyColumn 함수를 이용해서 MessageCard 요소들을 수직으로 배치하고 스크롤 할 수 있음 (LazyRow는 가로방향) (일반적인 Column함수는 많은 수의 항목이나 길이를 알 수 없는 목록을 표시할 경우 적절하지 않다고 함)
- by remember 키워드를 사용한 isExpanded 변수를 추가해서 현재 상태를 로컬에 저장하고 isExpanded의 값이 바뀌면 자동으로 원래 상태로 돌아올 수 있음



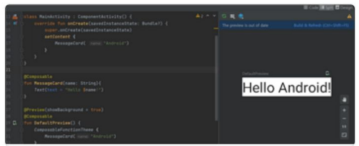
지난주 과제 확인

- 한윤재

import 주의!!
ui 관련 라이브러리는 “compose”를 확인
import문 지우고 다시 import 해보기

한윤재 오후 11:13

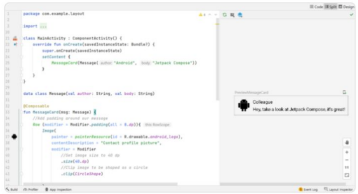
1. Composable functions
   - @Composable을 붙여서 Composable 함수를 만들 수 있다.
   - @Preview를 붙여서Composable 함수를 미리 볼 수 있다.



한윤재 오전 1:01

2. Layouts
   - Column, Row 함수로 element를 수직, 수평으로 정렬한다.
   - modifier를 사용해 composable을 구성하거나 장식할 수 있고, 상호작용을 추가할 수 있다.

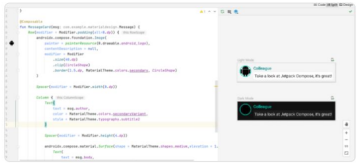
+ Modifier 작성에서 ‘Unsolved reference : padding’ 오류가 나타남.
→‘import java.lang.reflect.Modifier’를 ‘import androidx.compose.ui.Modifier’로 변경
→stack overflow에서 찾음.



한윤재 오후 11:56

3. Material design
   - Compose는 Material 디자인 및 UI요소를 즉시 사용 가능하도록 구현한다.
   - Material 디자인은 색상, 글씨체, 도형의 세 가지 핵심 요소를 중심으로 이루어진다.
   - Material 디자인 지원으로 야간 모드가 자동으로 조정된다.

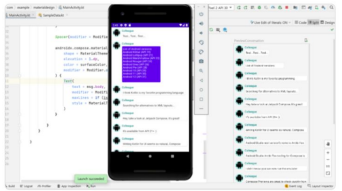
+ massage 클래스를 만드는 부분이 빠져 오류가 생김.
→ 강의 2에서 찾아 적용함.



한윤재 오전 2:48

4. Lists and animations
   - LazyColumn과 LazyRow.는 화면에 표시되는 요소만 렌더링해서 긴 목록에 매우 효율적이고, XML 레이아웃으로 RecyclerView의 복잡성을 피한다.
   - remember 및 mutableStateOf와 같은 상태 API를 사용하여 상태를 변경하면 UI가 자동으로 업데이트된다.

+ 색을 바꾸는 부분의 코드에서 오류가 생김.
→ import android.graphics.Color 를 androidx.compose.ui.graphics.Color로 변경함.



지난주 과제 확인

- 양용수

by ~가 안 될 때 (delegate)
= 으로 선언한 뒤에
surfaceColor.value로 값을 사용

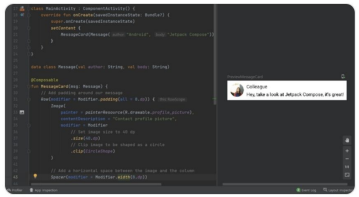
양용수 2:09 AM

1. Composable functions
   - @Composable 주석을 추가하면 함수를 구성할 수 있음.
   - @Composable 앞에 @Preview 주석을 추가하면 UI 요소의 미리 볼 수 있음. 미리보기 창 상단의 새로고침 버튼을 누르면 새로고침 할 수 있음.

(편집됨)

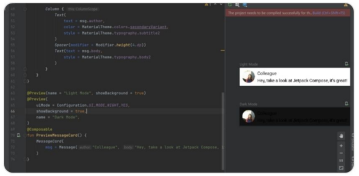
양용수 2:09 AM

2. Layout
   - Column 함수를 사용하면 요소를 수직으로 정렬할 수 있음.
   - Resource Manager를 통해서 이미지를 불러올 수 있음.(여기서 시간 엄청 오래 걸렸음)
   - 이미지와 함께 쓰기 위해서 Row 컴포저블을 추가했음.
   - 레이아웃을 구성할 때 크기와 모양을 조정해봤음. (편집됨)



양용수 2:09 AM

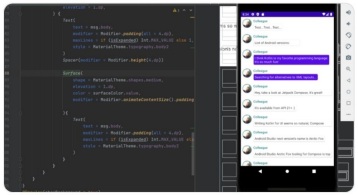
3: Material design
   - 2번 단계에서 좀 꾸미기 단계라고 결론지음
   - 글씨에 색, 도형, 서체, 스타일을 꾸미는 기능을 연습함. (편집됨)



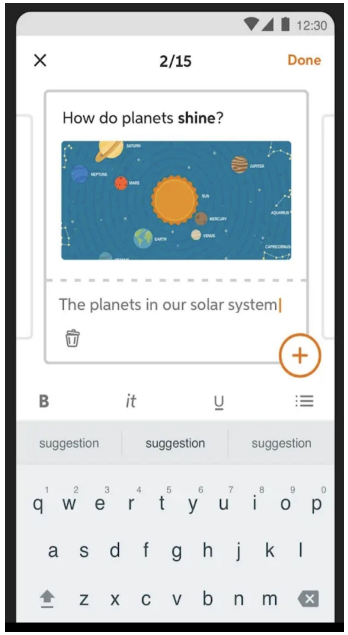
양용수 5:39 PM

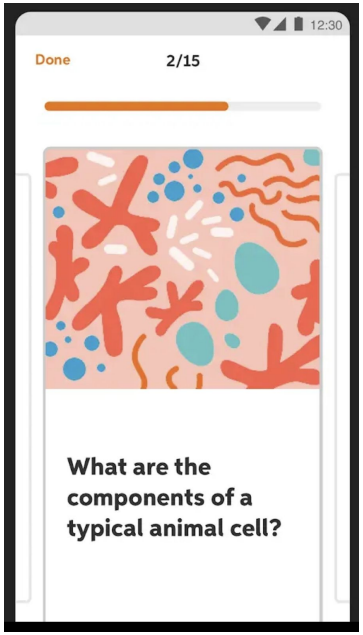
4. List and Animations
   - 데이터 세트를 갖고 오는 법을 알게 됨.
   - 애니메이션을 추가하기 위해서 remember / mutableStateOf 함수를 사용해서 이전 값을 추적함.
   - 색을 입히기 위해 animateColorAsState 함수를 사용함. 예상치 못한 오류가 나서
     `val surfaceColor: Color by animateColorAsState`
   - 를 val surfaceColor = animateColorAsState << 로 수정함.
   - 마지막으로 animateContentsSize 수정자를 사용하여 메시지 컨테이너 크기에 부드럽게 애니메이션을 적용함.

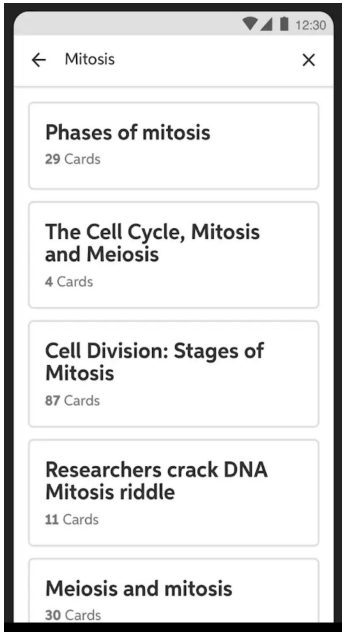
(편집됨)



Goal







Chegg Prep - Study Flashcard

필수적인 UI부터 단계적 학습 가능하다는 장점

진행방식

세션 수업

- Chegg 앱 주요 기능 파악 및 구현 실습
- 수업 진행 후 실습 시간
- 과제 안내

과제

- 2개의 필수 과제 & 1개의 선택 과제
- 코딩 과제는 무조건 GitHub을 통해 기록하기
- 최종적으로 과제가 끝나면 Slack 댓글을 통해 인증

*진행방식은 매주 유동적으로 변할 수 있습니다.

오늘 진행할 내용

- 단축키 소개

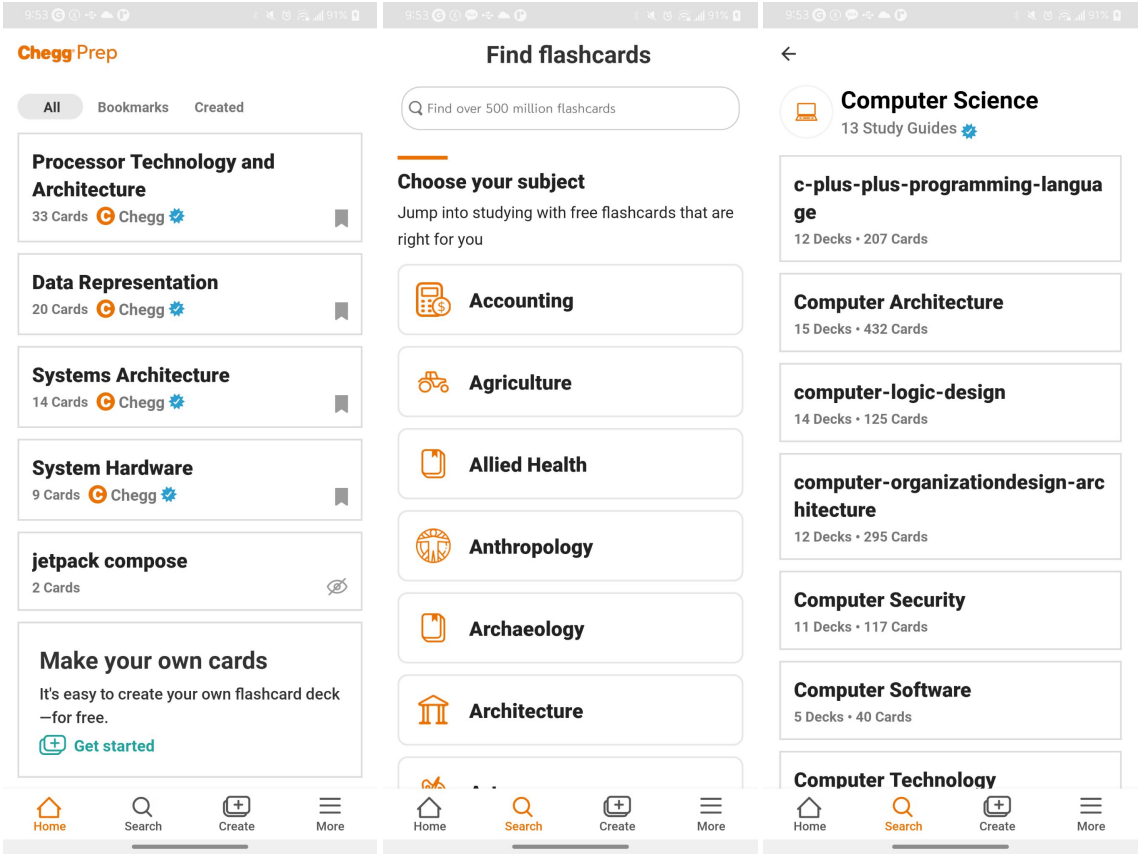
1. comp + Enter : Composable 문자를 자동 완성해서 적어준다.

2. Alt + Enter : 글씨가 빨간 색인 상태(오류)에 사용하며, import class를 해주어 코드가 정상적으로 작동하도록 해준다. + arguments들을 정리해준다. (자동 줄바꿈, '이름 = ' 붙이기)

3. Ctrl + Alt + L : 코드를 정렬할 수 있다.
   +) Settings -> Code Style 설정을 통해 정렬 방식을 수정할 수도 있다.

오늘 진행할 내용

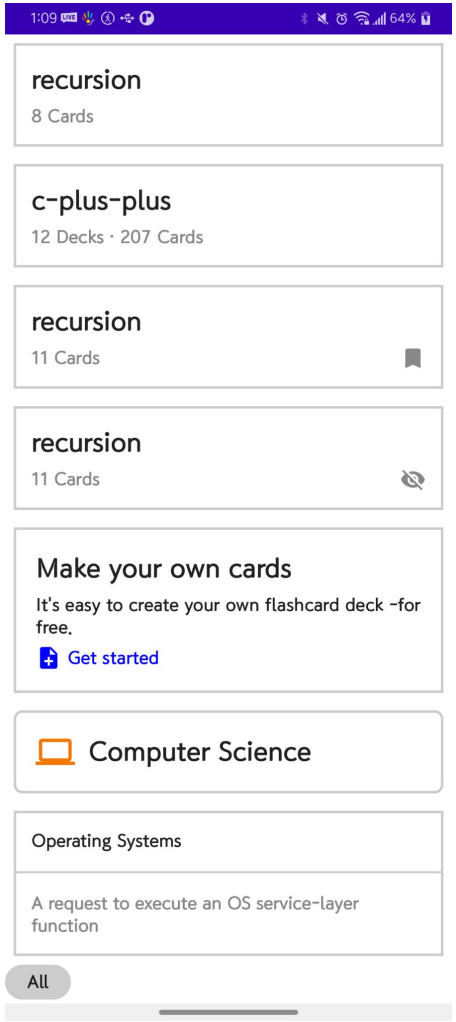
- Chegg Prep 화면



오늘 진행할 내용

- **새 프로젝트 만들기!!**
  - 이름 Chegg Prep
- 만들어 볼 composables

1. DeckInSubject
2. StudyGuide
3. DeckItem
4. MyDeckItem
5. MakeMyDeck
6. SubjectItem
7. CardItem
8. FilterText



Row / Column

html table에서의 row, column처럼 생각하면 X

안에 들어가는 요소들을 배열하는 컨테이너 (LinearLayout)

- Row는 수평으로 배열 (옆으로 추가됨) = horizontal
- Column은 수직으로 배열 (아래로 추가됨) = vertical

Row

- horizontalArrangement
- verticalAlignment

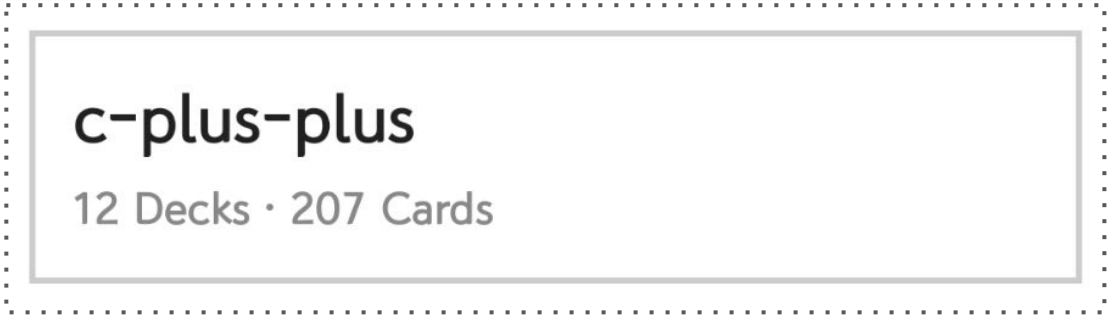
Column

- verticalArrangement
- horizontalAlignment

Column, Text, Spacer, padding, border, clickable

1. DeckInSubject

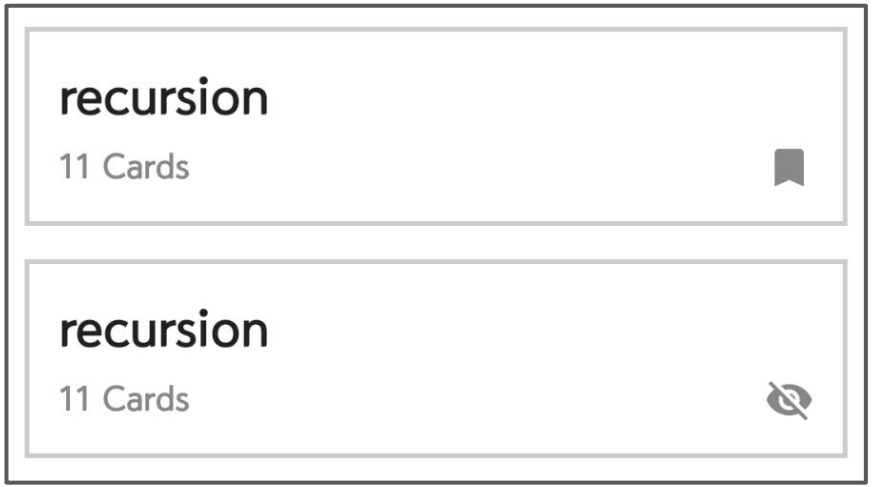
2. StudyGuide



Row, Icon

3. DeckItem

4. MyDeckItem



horizontalArrangement

Start

Center

End

SpaceAround

SpaceEvenly

SpaceBetween

aligned

spaceBy

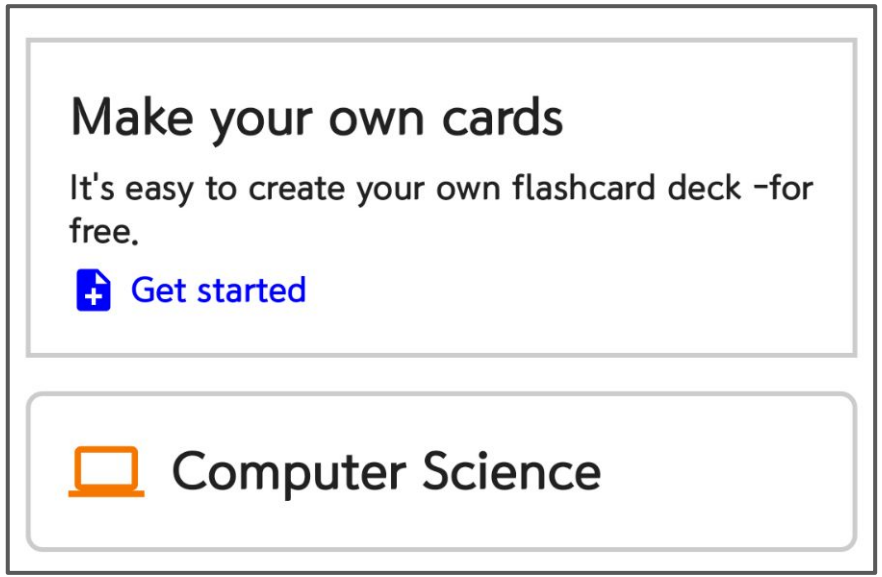
build.gradle (:app)

```
// Icons
    implementation "androidx.compose.material:material-icons-core:$compose_version"
    implementation "androidx.compose.material:material-icons-extended:$compose_version"
```

색 추가

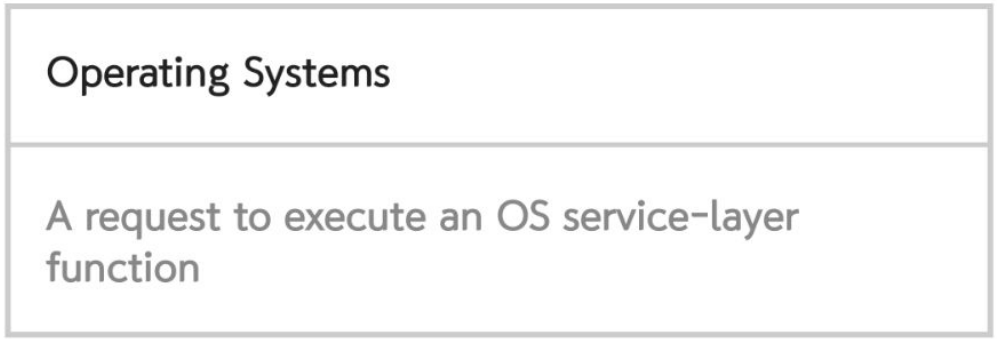
5. MakeMyDeck

6. SubjectItem



Divider, clickable enabled

7. CardItem



8. FilterText

All All All

과제

1. 세션 수업 내용 및 실습 복습

중요! GitHub을 설치하고 매주 코딩한 것은 commit & push 하는 것을 목표로 합니다.

commit 메세지에 복습을 하고 나서 가장 기억에 남는 개념 or 꼭 기억에 하고 싶은 내용 기록하기

+ 본인의 티스토리 또는 블로그에 내용을 추가로 정리해두는 것도 좋겠죠?!

GitHub 연동 방법은 Slack에 링크 첨부 예정

-> GitHub 연동 후 복습 기록 또는 화면 캡쳐 Slack에 업로드

과제

2. 하드 코딩으로 UI 만들기 실습

링크: Instagram UI [Instagram Profile UI in Jetpack Compose - Android Studio Tutorial]

영상을 보고 코드를 따라 해보면서 공부해봅시다!
당장은 이해가 안될지라도 조금씩 천천히 따라해봅시다
어려운 부분은 곧 설명 추가할게요~

-> 과제 완료 후에 commit & push 및 기록 화면 캡쳐 후 Slack에 업로드

과제

선택 과제) Jetpack Compose 유튜브 강의

링크: https://youtube.com/playlist?list=PLQkwcJG4YTCSpJ2NLhDTHhi6XBNfk9WiC

재생목록에서 1, 2, 3, 5, 8번 강의들을 세션 수업 복습 겸 추가 실습으로 공부

(2번 과제에 앞서 듣는 것을 추천)

-> 과제 완료 후에 commit & push 화면 캡쳐 또는 학습 내용 Slack에 업로드

끝.