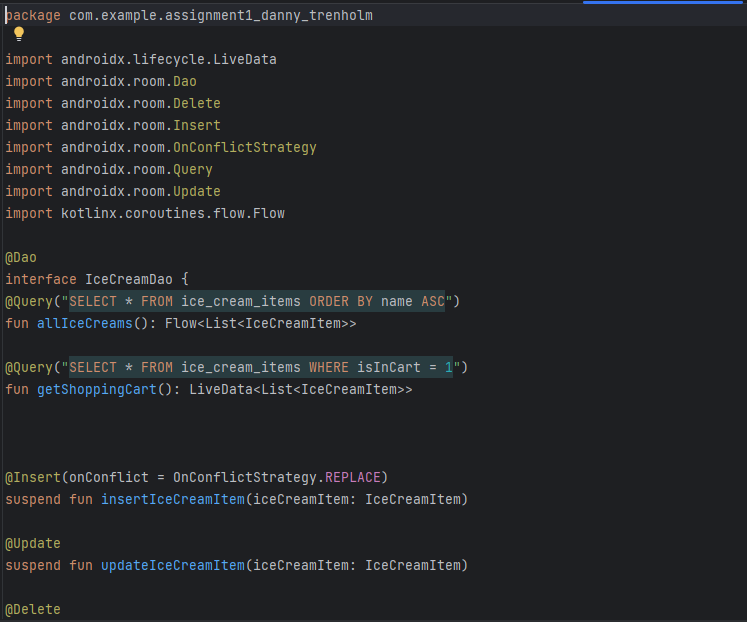
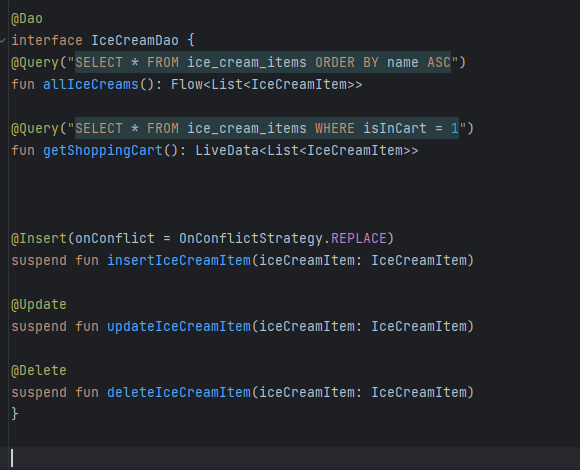
Report- Assignment #3

Mobile Application Development

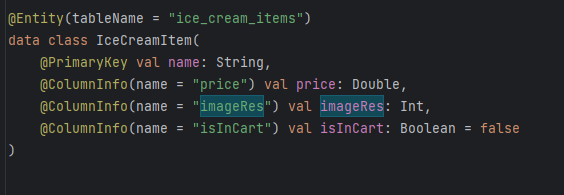
Danny Trenholm

201202966



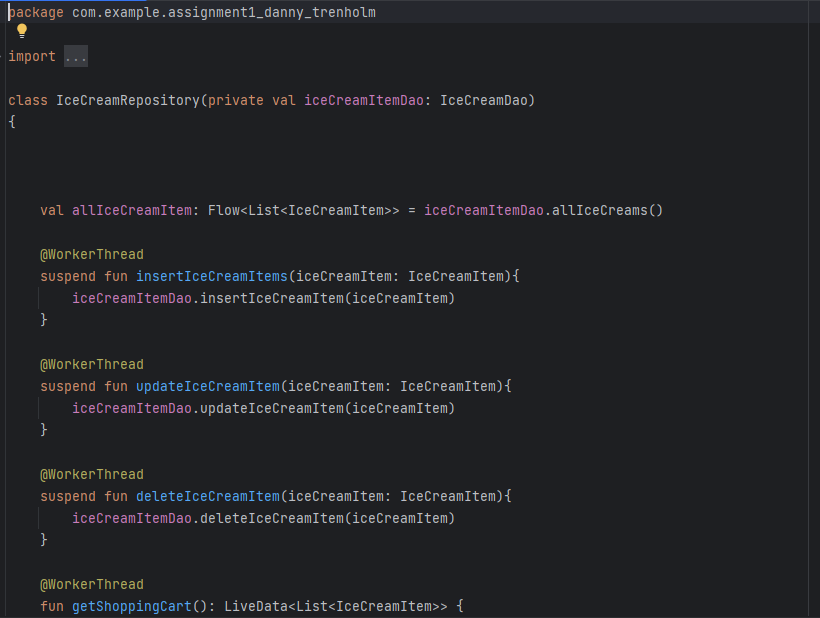
  
  
First, we create the IceCreamDao file, and for the purpose of this exercise, we are using a Boolean called isInCart to set the shopping cart items in the database. allIceCream is just purely for conjecture.

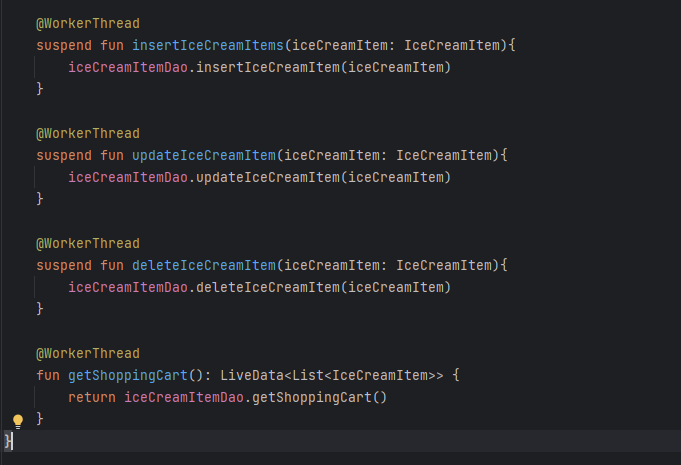
Here is our data class, located in MainActivity



The purpose is to use the price, and image\_res for the picture and calculation components. isInCart creates our custom table.

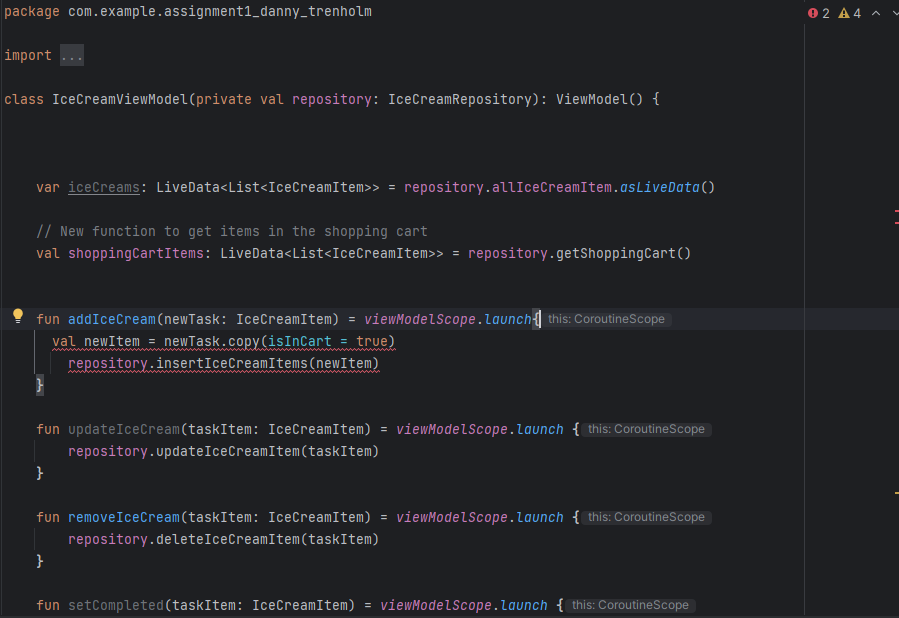
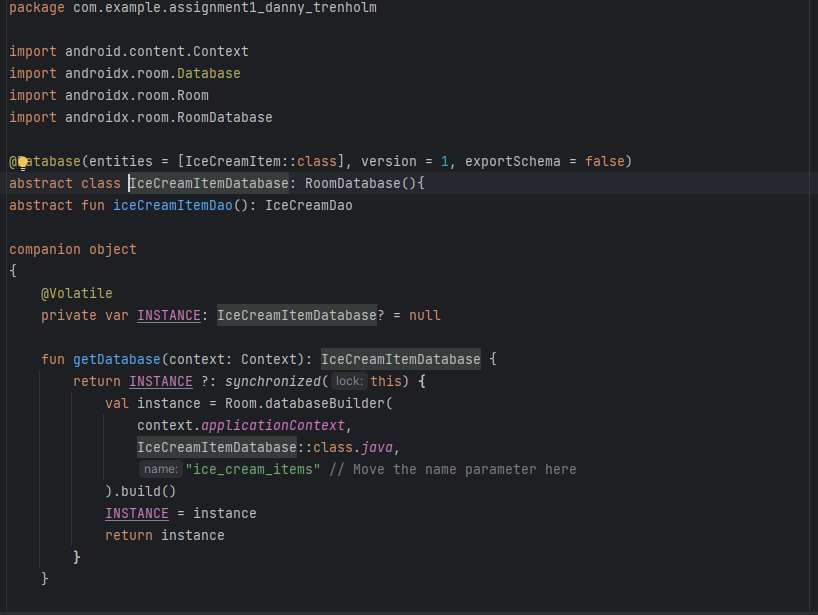
Now, to the Repository to implement the Daos

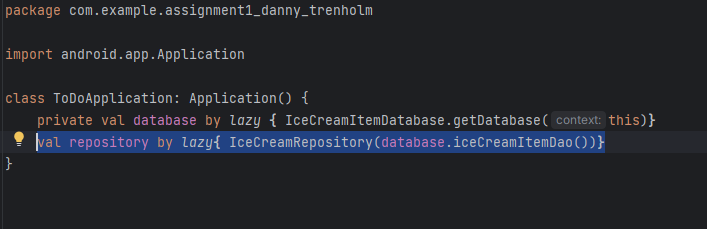




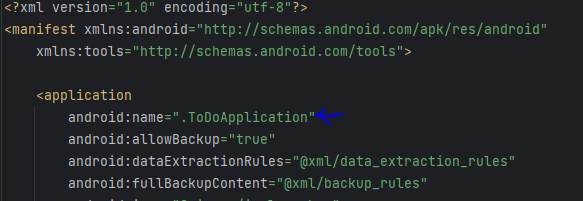
We implement our Dao as WorkerThread, and we are using three main threads here.

1. insertIceCreamItems – used for inserting into our custom table
2. deleteIceCreamItem – used for deleting our item from the custom table
3. getShoppingCart() – retrieve the shopping cart database

I then implement the ViewModel  
  
  
  
The iceCreamView Model adds, deletes, and gets the shoppingCartItems. Do not mind the red underline, that is just due to suspicious indent. We then implement the database  
  
  
then we add a toDoApplication to create our custom room for our shopping cart



This is implemented into our xml file, located in manifests



All the action happens in the MainActivity with this code. The styledText, RepeatImage, and CustomTopAppBar are all UI. IceCreamApp takes in the viewModel, and displays it through the StylizedFinishedApplication.

Here is that code for the main activity  
  
package com.example.assignment1\_danny\_trenholm  
  
import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.activity.viewModels  
import androidx.compose.foundation.Image  
import androidx.compose.foundation.layout.\*  
import androidx.compose.foundation.lazy.LazyColumn  
import androidx.compose.foundation.shape.*CircleShape*import androidx.compose.material3.Button  
import androidx.compose.material3.Surface  
import androidx.compose.material3.Text  
import androidx.compose.runtime.Composable  
import androidx.compose.runtime.getValue  
import androidx.compose.runtime.livedata.observeAsState  
import androidx.compose.runtime.mutableStateOf  
import androidx.compose.runtime.remember  
import androidx.compose.runtime.setValue  
import androidx.compose.ui.Alignment  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.graphics.Color.Companion.*Blue*import androidx.compose.ui.graphics.Color.Companion.*Green*import androidx.compose.ui.graphics.Color.Companion.*Red*import androidx.compose.ui.graphics.Shadow  
import androidx.compose.ui.res.painterResource  
import androidx.compose.ui.text.TextStyle  
import androidx.compose.ui.text.font.FontFamily  
import androidx.compose.ui.text.style.TextAlign  
import androidx.compose.ui.tooling.preview.Preview  
import androidx.compose.ui.unit.dp  
import com.example.assignment1\_danny\_trenholm.ui.theme.Assignment1\_Danny\_TrenholmTheme  
import androidx.compose.ui.unit.sp  
import androidx.compose.runtime.mutableIntStateOf  
import androidx.compose.ui.graphics.Color  
import androidx.compose.ui.graphics.Color.Companion.*Yellow*import androidx.compose.ui.text.font.FontWeight  
import androidx.lifecycle.ViewModelProvider  
import androidx.room.ColumnInfo  
import androidx.room.Entity  
import androidx.room.PrimaryKey  
  
  
class MainActivity : ComponentActivity() {  
  
  
 private val iceCreamViewModel: IceCreamViewModel by *viewModels* **{** IceCreamItemModelFactory((*application* as ToDoApplication).repository)  
 **}** override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
  
 *setContent* **{** Assignment1\_Danny\_TrenholmTheme **{** // A surface container using the 'background' color from the theme  
 StyledFinishedApplication(iceCreamViewModel)  
 **}  
 }** }  
}  
  
  
  
  
@Composable  
fun StyledFinishedApplication(iceCreamViewModel: IceCreamViewModel){  
 val lightYellow = *Color*(247, 247, 216) // Using RGB values  
 Surface(  
 modifier = Modifier.*fillMaxSize*(),  
 color = lightYellow  
 ) **{** IceCreamOrderApp(iceCreamViewModel)  
 **}**}  
  
@Composable  
fun IceCreamOrderApp(iceCreamViewModel: IceCreamViewModel) {  
 var shoppingCart by remember **{** *mutableStateOf*(*listOf*<IceCreamItem>()) **}** // Observe changes in shopping cart items  
 val shoppingCartItems by iceCreamViewModel.shoppingCartItems.observeAsState(*listOf*())  
  
 shoppingCart = shoppingCartItems  
  
 val iceCreamItems = *listOf*(  
 IceCreamItem("Vanilla", 3.50, R.drawable.*vanilla*),  
 IceCreamItem("Chocolate", 3.50, R.drawable.*chocolate*),  
 IceCreamItem("Strawberry", 3.50, R.drawable.*strawberry*),  
 IceCreamItem("Hoofprints", 3.50, R.drawable.*hoofprints*),  
 IceCreamItem("Dinosaur Bones", 3.50, R.drawable.*dinosaur\_bones*),  
 IceCreamItem("Cookie Dough", 3.50, R.drawable.*chocolate\_chip\_cookie\_dough*),  
 IceCreamItem("Vanilla Cup", 3.50, R.drawable.*vanilla\_cup*),  
 IceCreamItem("Chocolate Cup", 3.50, R.drawable.*chocolate\_cup*),  
 IceCreamItem("Strawberry Cup", 3.50, R.drawable.*strawberry\_cup*),  
 IceCreamItem("Hoofprints Cup", 3.50, R.drawable.*hoofprints*),  
 IceCreamItem("Dinosaur Bones Cup", 3.50, R.drawable.*dinosaur\_bones\_cup*),  
 IceCreamItem("Cookie Dough Cup", 3.50, R.drawable.*chocolate\_chip\_cookie\_dough\_cup*)  
  
 )  
  
 LazyColumn(  
 modifier = Modifier  
 .*fillMaxSize*()  
 .*padding*(16.*dp*)  
 ) **{** item **{** RepeatImage(30)  
 StyledText(text = "Hello")  
 CustomTopAppBar(title = "Ice Cream Order")  
 **}** item **{** IceCreamList(iceCreamItems) **{** iceCreamItem **->** iceCreamViewModel.addIceCream(iceCreamItem)  
 **}  
 }** item **{** Spacer(modifier = Modifier.*height*(16.*dp*))  
 ShoppingCart(shoppingCartItems) **{** removedItem **->** iceCreamViewModel.removeIceCream(removedItem)  
 **}  
 }** item **{** RepeatImage(70)  
 **}  
 }**}  
  
  
@Composable  
fun StyledText(text: String) {  
 Row **{** text.*forEach* **{** character **->** val style = when (character) {  
 'H' -> TextStyle(color = *Red*, fontSize = 80.*sp*)  
 'e' -> TextStyle(color = *Green*, fontWeight = FontWeight.Bold, fontSize = 100.*sp*)  
 'l' -> TextStyle(color = *Blue*, fontSize = 92.*sp*)  
 'o' -> TextStyle(color = *Yellow*, fontWeight = FontWeight.Bold, fontSize = 100.*sp*)  
  
 else -> TextStyle() // Default style  
 }  
 Text(text = character.toString(), style = style)  
 **}  
 }**}  
  
@Composable  
fun CustomTopAppBar(title: String) {  
 Text(  
 text = title,  
 fontSize = 38.*sp*,  
 style = TextStyle(  
 color = *Blue*,  
 shadow = Shadow(  
 color = androidx.compose.ui.graphics.Color.Black,  
 blurRadius = 50f,  
 ),  
  
 ),  
 textAlign = TextAlign.Center,  
 fontFamily = FontFamily.Serif,  
 modifier = Modifier  
 .*fillMaxWidth*()  
 .*padding*(16.*dp*)  
  
  
 )  
}  
  
@Composable  
fun RepeatImage(times: Int) {  
 Row(  
 modifier = Modifier  
 .*heightIn*(max = 150.*dp*)  
 .*height*(100.*dp*),  
 horizontalArrangement = Arrangement.spacedBy(2.*dp*)  
 ) **{** *repeat*(times) **{** val painter = painterResource(id = R.drawable.*parlour*)  
 Image(  
 painter = painter,  
 contentDescription = null,  
 modifier = Modifier  
 .*fillMaxHeight*()  
 .*weight*(  
 weight = painter.intrinsicSize.width / painter.intrinsicSize.height,  
 fill = false  
 )  
 )  
 **}  
 }**}  
  
  
@Composable  
fun IceCreamList(  
 iceCreamItems: List<IceCreamItem>,  
 onItemClick: (IceCreamItem) -> Unit  
) {  
 Column **{** Text("Choose Ice Cream:", fontSize = 20.*sp*)  
  
 iceCreamItems.*forEach* **{** iceCreamItem **->** Row(  
 modifier = Modifier  
 .*padding*(vertical = 8.*dp*)  
 .*fillMaxWidth*(),  
 verticalAlignment = Alignment.CenterVertically  
 ) **{** Image(  
 painter = painterResource(id = iceCreamItem.imageRes),  
 contentDescription = null,  
 modifier = Modifier.*size*(50.*dp*)  
 )  
 Spacer(modifier = Modifier.*width*(8.*dp*))  
 Text(iceCreamItem.name, fontSize = 16.*sp*)  
 Spacer(modifier = Modifier.*weight*(1f))  
 Button(  
 onClick = **{** onItemClick(iceCreamItem) **}**,  
 content = **{** Text(text = "Add") **}** )  
 **}  
 }  
 }**}  
  
@Composable  
fun ShoppingCart(  
 items: List<IceCreamItem>,  
 onRemoveItemClick: (IceCreamItem) -> Unit  
) {  
 Column **{** Text("Shopping Cart:", fontSize = 20.*sp*)  
  
 items.*forEach* **{** item **->** Row(  
 modifier = Modifier  
 .*padding*(vertical = 8.*dp*)  
 .*fillMaxWidth*(),  
 verticalAlignment = Alignment.CenterVertically  
 ) **{** Image(  
 painter = painterResource(id = item.imageRes),  
 contentDescription = null,  
 modifier = Modifier.*size*(50.*dp*)  
 )  
 Spacer(modifier = Modifier.*width*(8.*dp*))  
 Text("${item.name} - $${item.price}", fontSize = 16.*sp*)  
 Spacer(modifier = Modifier.*weight*(1f))  
 Button(  
 onClick = **{** onRemoveItemClick(item) **}**,  
 content = **{** Text(text = "Remove") **}** )  
 **}  
 }** val totalPrice = items.*sumByDouble* **{ it**.price **}** Text("Total Price: $$totalPrice", fontSize = 20.*sp*)  
 **}**}  
  
@Entity(tableName = "ice\_cream\_items")  
data class IceCreamItem(  
 @PrimaryKey val name: String,  
 @ColumnInfo(name = "price") val price: Double,  
 @ColumnInfo(name = "imageRes") val imageRes: Int,  
 @ColumnInfo(name = "isInCart") val isInCart: Boolean = false  
)