



**ŽILINSKÁ UNIVERZITA V ŽILINE**

Fakulta riadenia  
a informatiky

Semestrálna práca z predmetu  
**VÝVOJ APLIKÁCIÍ PRE MOBILNÉ ZARIADENIA**

**vypracoval:** Quang Phung Viet

**študijná skupina:** 5ZYR21

**cvičiaci:** doc. Ing. Patrik Hrkút, PhD.

**termín cvičenia:** streda Blok 1-3

v Žiline dňa 01.05.2024



## Špecifikácia zadania:

Za semestrálnu prácu som si vybral vypracovať aplikáciu na nákupný zoznam. Výber tejto témy bol z dôvodu malého výskytu na app store Obchod Play kde som videl potenciálnu medzeru, ktorú by som mohol doplniť. Zamerlal som sa viac na finančnú stránku nakupovania. Teda prehľad návykov nakupovania. Za koľko bol zaplatený nákup, a aj krivka priemeru nákupov. A pre doplnok som pridal aj hlasový vstup pre niektoré polia, ktoré musí užívateľ vyplniť.

## Podobné aplikácie a popis:

Nezmenilo sa moc od poslaného checkpointu, takže tu skopírujem požadované veci na dokumentáciu.

Listonic:

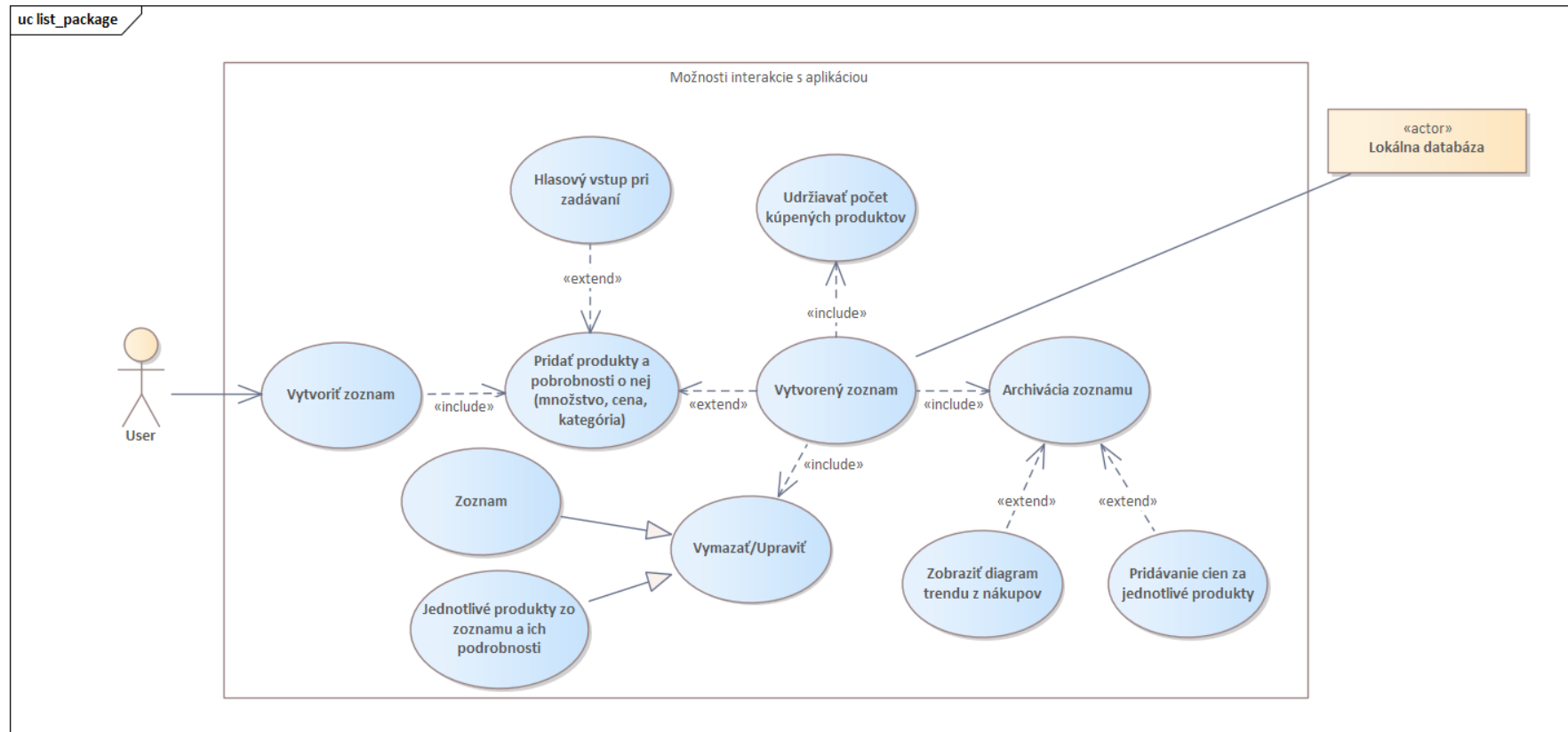
- ✚ Táto aplikácia má najbližšie k mojej verzii shopping listu.
- ✚ Vyznačuje sa možnosťou prístupu cez webovú stránku listonic.com a zdieľanie zoznamov s ostatnými (musia mať tiež aplikáciu).
- ✚ Hlavným publikom sú ľudia, ktorí chcú mať zorganizovaný list na nákup.
- ✚ Moja aplikácia je viac orientovaná na financie, čiže bude ukazovať cenové trendy v nákupoch (diagram).
- ✚ Mínusy mojej apky oproti tejto bude hlavne prístup k zoznamu cez web a zdieľanie zoznamu s ostatnými.

Bring!

- ✚ Viac menej je obsah aplikácie rovnaký ako Listonic, ale líši sa funkciami ako:
- ✚ Možnosť pridávať zákaznícke karty
- ✚ Pridávať recepty s potrebnými ingredienciami

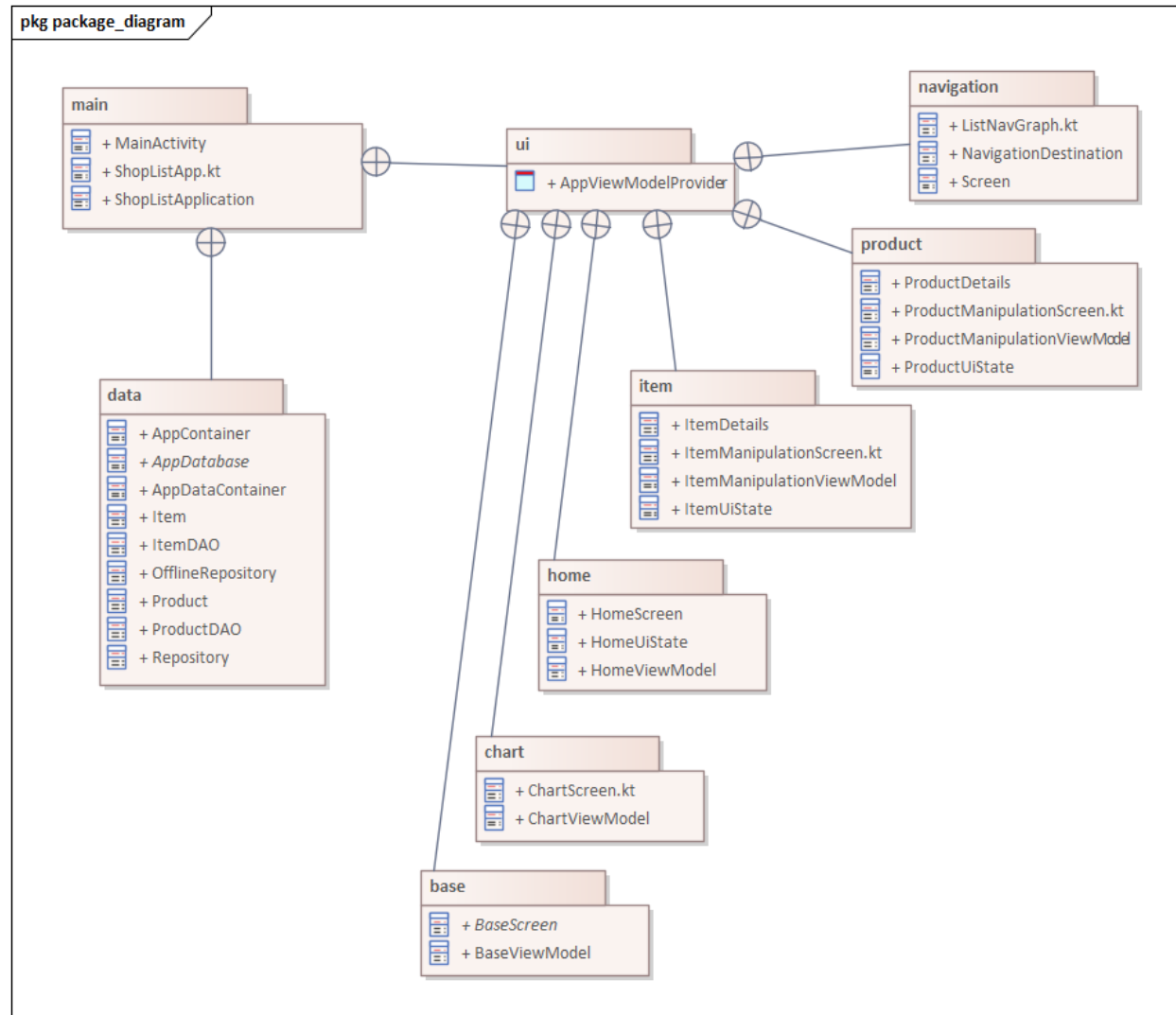
## Návrh riešenia problému:

Krátka analýza:





Návrh aplikácie:



Podrobnejšie ku jednotlivým triedam bude k dispozícii v repozitári súbor .eap.



## Popis implementácie:

Aktivity a fragmenty:

Podľa projektu mám aktivitu len pri MainActivity.kt.

Využitie AndroidX komponentov:

Room (celý priečinok „data“ je implementácia databázy), ViewModel (jednotlivé ViewModely popri „screen“ súboroch), Navigation(priečinok navigation)

Dynamické rozloženie (dynamic layout) – napr. zoznamy, mriežky, tabuľky:

LazyColumn

```
LazyColumn(modifier = modifier) {  
    items(items = itemList, key = {it.itemId}) {  
        item -> ListCard(item = item,  
            modifier = Modifier  
                .padding(dimensionResource(id = R.dimen.padding_small))  
                .clickable { onItemClick(item) },  
            products =  
homeViewModel.getProductsFromItem(item.itemId).collectAsState(initial=  
emptyList()).value  
        )  
    }  
}
```

, ModalBottomSheet

```
ModalBottomSheet(  
    onDismissRequest = { coroutineScope.launch {  
        bottomSheetState.hide()  
    }  
    openBottomSheet = false  
    onDismissRequest() },  
    sheetState = bottomSheetState,  
    windowInsets = windowInsets  
) ...
```

, OutlinedTextField

```
OutlinedTextField(modifier = Modifier.fillMaxWidth(),  
    value = textFieldValue,  
    onValueChange = {  
        textFieldValue = it  
        onItemValueChange(itemUiState.itemDetails.copy(name = it))  
    },  
    singleLine = true,  
    label = { Text(text = stringResource(id = R.string.entry_name)) }, ...  
)
```

, TextField

```
TextField(  
    modifier = Modifier.menuAnchor(),  
    value = selectedCategory,  
    onValueChange = onSelectedCategoryChange,  
    singleLine = true,  
    label = { Text(text = stringResource(id = R.string.category)) },  
    trailingIcon = { ExposedDropdownMenuDefaults.TrailingIcon(expanded =  
allowExpanded) },  
    colors = ExposedDropdownMenuDefaults.textFieldColors(),  
    readOnly = true  
)
```



## Použitie externého frameworku / knižnice:

### Voice Input

```
private val _voiceInput = MutableStateFlow("")
val voiceInput = _voiceInput.asStateFlow()

fun handleVoiceInputResult(data: Intent?) {
    val matches =
    data?.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS)
    if (!matches.isNullOrEmpty()) {
        val voiceInput = matches[0]
        viewModelScope.launch {
            _voiceInput.emit(voiceInput)
        }
    }
}
```

## Dodatočné body za použitie technológií mimo rozsahu predmetu:

### Chart API (<https://github.com/patrykandpatrick/vico>)

```
Chart(
    chart = composedChart,
    chartModelProducer = composedChartEntryModelProducer,
    modifier = modifier,
    startAxis = rememberStartAxis(valueFormatter =
integerAxisValueFormatter,
    title = "Product No."),
    bottomAxis = rememberBottomAxis(),
    endAxis = rememberEndAxis(title = "y = Price $", titleComponent =
textComponent { this.textSizeSp = 16f },
    label = null, tick = null, itemPlacer =
AxisItemPlacer.Vertical.default(maxItemCount = 0)),
    topAxis = rememberTopAxis(title = "x = item No.", titleComponent =
textComponent { this.textSizeSp = 16f },
    label = null, tick = null)
)
```

## Zoznam použitých zdrojov:

<https://github.com/patrykandpatrick/vico/tree/master/sample/src/main/java/com/patrykandpatrick/vico/sample/showcase/charts>

[https://developer.android.com/reference/kotlin/androidx/compose/material3/package-summary#ModalBottomSheet\(kotlin.Function0,androidx.compose.ui.Modifier,androidx.compose.material3.SheetState,androidx.compose.ui.unit.Dp,androidx.compose.ui.graphics.Shape,androidx.compose.ui.graphics.Color,androidx.compose.ui.graphics.Color,androidx.compose.ui.unit.Dp,androidx.compose.ui.graphics.Color,kotlin.Function0,androidx.compose.foundation.layout.WindowInsets,androidx.compose.material3.ModalBottomSheetProperties,kotlin.Function1\)](https://developer.android.com/reference/kotlin/androidx/compose/material3/package-summary#ModalBottomSheet(kotlin.Function0,androidx.compose.ui.Modifier,androidx.compose.material3.SheetState,androidx.compose.ui.unit.Dp,androidx.compose.ui.graphics.Shape,androidx.compose.ui.graphics.Color,androidx.compose.ui.graphics.Color,androidx.compose.ui.unit.Dp,androidx.compose.ui.graphics.Color,kotlin.Function0,androidx.compose.foundation.layout.WindowInsets,androidx.compose.material3.ModalBottomSheetProperties,kotlin.Function1))