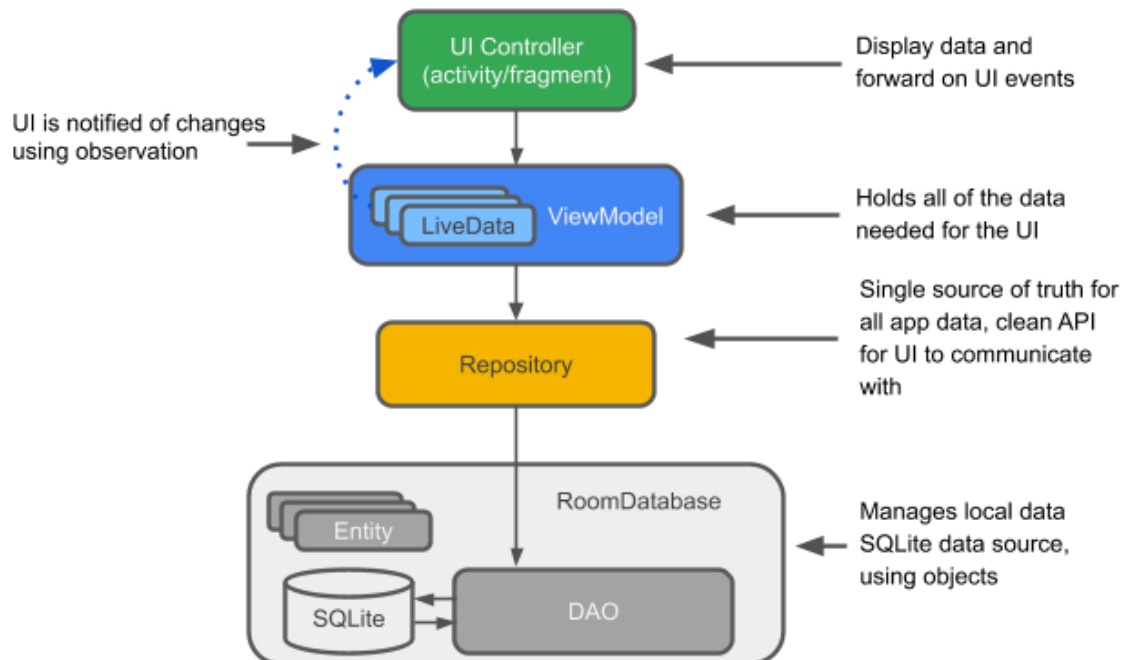


# FridgeManager

1. Wstęp i motywacja: Projektem będzie stworzenie aplikacji na Androida w której użytkownik będzie wprowadzał po zakupach w sklepie. Wprowadzane będą ilości oraz przede wszystkim daty ważności. Sam stykam się z problemem kiedy kupuję produkty i zapominam o ich datach ważności i po jakimś czasie okazuje się, że zapominam o nich i wyciągam już zepsute. Więc aplikacja przede wszystkim ma listować produkty po kończącej się dacie ważności co jest jej główną funkcjonalnością. Informacja o ilości produktów może być też przydatna przy zakupach, zamiast sporządzać listę zakupów na kartce można chodzić z telefonem. Dodatkowo dodano grupowanie produktów po kategoriach.
2. Funkcjonalności
  - możliwość wprowadzenia produktu i jego daty ważności.
  - usuwanie produktu poprzez przesunięcie w lewo lub prawo po ekranie.
  - listowanie po dacie ważności.
  - walidacja wprowadzanych danych
  - swobodne przeglądanie produktów(responsywne UI)
  - Architektura MVVM oraz specjalizowane metody tworzenia baz danych na aplikacje mobilne(Room)

Aplikacja napisana jest na Androida w Javie. Wykorzystałem zaawansowane komponenty Androida takie jak LiveData, Room, ViewModel. Pozwoliło mi to oddzielić frontend od backendu i uzyskać dodatkowe warstwy abstrakcji.



```

+ AddProductActivity extends AppCompatActivity...
- fields
  + EXTRA_NAME: String
  + EXTRA_NAME: String
  + EXTRA_CATEGORY: String
  + EXTRA_DESCRIPTION: String
  + EXTRA_ID: String
  + EXTRA_MONTH: String
  + EXTRA_YEAR: String
  + editExtra: EditText
  + editExtraDesc: EditText
  + editExtraCateg: EditText
  + dayPic: NumberPicker
  + monthPic: NumberPicker
  + yearPicker: NumberPicker
- constructors
- methods
  + onCreate(savedInstanceState: Bundle?): void
  + saveProd(): void
  + onCreateOptionsMenu(): boolean
  + onOptionsItemSelected(): boolean

```

```

+ AppCompatActivity extends AppCompatActivity...
- implements AppCompatActivity
  + TaskStackBuilder: SupportParent...
  + ActionBarDrawerToggle: DelegatePro...
- fields
- constructors
- methods

```

```

+ MainActivity extends AppCompatActivity...
- fields
  + productViewModel: ProductViewModel
  + ADD_PRODUCT_REQUEST: int
  + SHOW_PRODUCT_REQUEST: int
- constructors
- methods
  + onCreate(savedInstanceState: Bundle?): void
  + onActivityResult(requestCode: int, resultCode: int, data: Intent?): void

```

```

+ ProductAdapter extends RecyclerView.Adapter...
- fields
  + products: List<Product>
  + listener: OnItemClickListener
- constructors
- methods
  + onCreateViewHolder(parent: ViewGroup, viewType: Int): ProductViewHolder
  + onBindViewHolder(holder: ProductViewHolder, position: Int): void
  + getItemCount(): Int
  + selfProducts(products: List<Product>): void
  + selfProduct(position: Int): Product
  + selfOnItemClickListener(listener: OnItemClickListener): void

```

```

+ AndroidViewModel extends ViewModel...
- fields
  + mAppCompatActivity: AppCompatActivity
- constructors
  + AndroidViewModel(appCompatActivity: AppCompatActivity)
- methods
  + getAppCompatActivity(): AppCompatActivity

```

```

+ ProductViewModel extends AndroidViewModel...
- fields
  + repository: ProductRepository
  + allProducts: LiveData<List<Product>>
  + searchByLiveDate: LiveData<List<Product>>
  + filterLiveDate: MutableLiveData<String>
- constructors
  + ProductViewModel(appCompatActivity: AppCompatActivity)
- methods
  + insert(product: Product): void
  + update(product: Product): void
  + delete(product: Product): void
  + getAllProducts(): LiveData<List<Product>>
  + selfCategory(category: String): void

```

```

+ ProductDatabase extends RoomDatabase
- fields
  + instance: ProductDatabase
  + roomCallback: Callback
- constructors
  + ProductDatabase(): ProductDatabase
- methods
  + synchronize()
  + getInstance(context: Context): ProductDatabase

```

```

+ RoomDatabase
- fields
- constructors
- methods

```

```

+ RecyclerView.Adapter<V>
- fields
  + mObservable: AdapterObservable
  + mHasStableIds: boolean
- constructors
  + Adapter()
- methods

```

```

+ ProductDao
- fields
- methods
  + insert(product: Product): void
  + update(product: Product): void
  + delete(product: Product): void
  + deleteAllProduct(): void
  + getAllProduct(): LiveData<List<Product>>
  + getProductsByCategory(selectedCategory: String): LiveData<List<Product>>

```

```

+ ProductRepository
- fields
  + productID: ProductID
  + allProducts: LiveData<List<Product>>
  + category: String
- constructors
  + ProductRepository(appCompatActivity: AppCompatActivity)
- methods
  + insert(product: Product): void
  + update(product: Product): void
  + delete(product: Product): void
  + deleteAllProduct(product: Product): void
  + getProductsByCategory(selectedCategory: String): LiveData<List<Product>>
  + getAllProduct(): LiveData<List<Product>>

```

```

+ Product
- fields
  + id: Int
  + name: String
  + expDate: Date
  + description: String
  + category: String
- constructors
  + Product(name: String, expDate: Date, description: String, category: String)
- methods
  + selfExpDate(expDate: Date): void
  + selfDescription(description: String): void
  + selfCategory(category: String): void
  + selfID(id: Int): void
  + getID(): Int
  + getNA(): String
  + getCategory(): String
  + getExpDate(): Date

```

