4. КОНСТРУЮВАННЯ ПРОГРАМНОГО ПРОДУКТУ

4. 1 Особливості конструювання структур даних

4.1.1 Особливості інсталяції та роботи з СУБД

Розглянуто створення бази даних для роботи веб-системи подій з використанням СУБД PostgreSQL.

Для тестування програмних модулів використовується СУБД PostgreSQL версії 13.0.

4.1.2 Особливості створення структур даних

До створеної бази даних були створені тригери та функції для маніпулювання даними, занесена демонстраційна інформація. Також були розроблені запити для демонстрації можливостей створеної бази даних. Архітектура бази даних передбачає підключення її до комп’ютерного додатку чи веб-сайту через використання користувачів. Така система дозволяє обмежувати привілеї для захисту даних.

Розглянемо SQL-запити для створення таблиць:

1. Таблиця “Користувач”

CREATE SEQUENCE s\_user;

CREATE TABLE user

(id\_user INT PRIMARY KEY DEFAULT NEXTVAL('s\_user'),

name VARCHAR,

surname VARCHAR,

middlename VARCHAR,

group VARCHAR,

age INT,

sex BOOLEAN,

login VARCHAR,

password VARCHAR,

rights BOOLEAN );

1. Таблиця “Афіша”

CREATE TABLE posters

(id\_poster INT PRIMARY KEY DEFAULT NEXTVAL('s\_user'),

name VARCHAR,

date DATE,

description VARCHAR,

publisher VARCHAR,

id\_keyword INT REFERENCES keyword(id),

id\_user INT REFERENCES user(id) ));

1. Таблиця “Кейворд”

CREATE TABLE keyword

(id\_keyword INT PRIMARY KEY DEFAULT NEXTVAL('s\_keyword'),

name VARCHAR,

);

1. Таблиця «Обрані кейворд»

CREATE TABLE favposter

(

id\_poster INT,

id\_user INT,

PRIMARY KEY (id\_user,id\_poster)

);

1. Таблиця « Обрані афіши»

CREATE TABLE favkeyword

(

id\_user INT,

id\_keyword INT,

PRIMARY KEY (id\_user, id\_keyword)

);

Для зберігання первинного ключа події використовується тип даних INT, бо операції порівняння та пошуку швидше за все працюють з цифровими типами.

База даних складається з наступних таблиць: Користувачі, Афіша, Кейворд, Обрані афіши, Обрані кейворд.

Таблиця «Користувач» містить інформацію про усіх користувачів системи. У ній присутні такі поля, як унікальний ідентифікатор (первинний ключ), ім’я, прізвище, по батькові, групу, вік, стать, пароль та логін для входу у систему, права (звичайний користувач чи адміністратор).

Таблиця «Афіша» містить інформацію про афіши. У ній присутні такі поля, як унікальний ідентифікатор події, назва афіши, її дата та час створення.

Таблиця «Обрані кейворд» містить інформацію про усі ключові слова, що були обрані користувачем. Це також таблиця, яка зв’язує таблиці «Кейворд» и «Користувач» у св’язок багато-до-багатьох. У ній присутні такі поля, як унікальні ідентифікатори (первинні ключі) таблиць «Кейворд» и «Користувач»

Таблиця «Обрані афіши» містить інформацію про усі афіші, що що були обрані користувачем. Це також таблиця, яка зв’язує таблиці «Афіші» и «Користувач» у св’язок багато-до-багатьох. У ній присутні такі поля, як унікальні ідентифікатори (первинні ключі) таблиць «Афіші» и «Користувач».

4.2 Особливості конструювання програмних модулів

4.2.1 Особливості роботи з інтегрованим середовищем розробки

Інформаційна система «Study Posters» представляє собою андроід-додаток.

Розробка системи виконується в Android Studio. WebStorm знаходить застосування всіх можливостей сучасної андроід-розробки. Включає в себе розумне автодоповнення коду, перевірку помилок на льоту, швидку навігацію по коду і рефакторинг для Java та Kotlin.

4.2.2 Особливості створення програмної структури

Для реалізації системи була обрана мови об’єктно-орієнтованого програмування Kotlin та Java .

Наведемо як приклад клас створення ленти користувача на сайті. Використовуємо мову программування Kotlin.



Створення сторінок у андроід-додадку відбувається за допомогою мови розмітки XML. Приклад використання XML-файлів:

Файл activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <android.support.v7.widget.Toolbar  
 android:id="@+id/main\_toolbar"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:background="@color/colorPrimary"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="@color/white"  
 android:text="@string/app\_name"/>  
   
 <ImageView  
 android:id="@+id/main\_filterIcon"  
 android:layout\_width="24dp"  
 android:layout\_height="24dp"  
 android:layout\_gravity="right|center\_vertical"  
 android:layout\_marginRight="16dp"  
 android:src="@drawable/ic\_filter"/>  
  
 </android.support.v7.widget.Toolbar>  
  
 <fragment  
 android:id="@+id/fragment"  
 android:name="com.sumera.argallery.ui.feature.picturelist.PictureListFragment"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/main\_tabs"/>  
  
 <android.support.design.widget.TabLayout  
 android:id="@+id/main\_tabs"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/main\_toolbar">  
  
 <android.support.design.widget.TabItem  
 android:id="@+id/main\_allTab"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/main\_all\_tab"/>  
  
 <android.support.design.widget.TabItem  
 android:id="@+id/main\_favouriteTab"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/main\_favourites\_tab"/>  
  
 <android.support.design.widget.TabItem  
 android:id="@+id/main\_filteredTab"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/main\_filter\_tab"/>  
  
 </android.support.design.widget.TabLayout>  
  
  
 <FrameLayout  
 android:id="@+id/main\_filterTabContainer"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"/>  
</android.support.constraint.ConstraintLayout>

4.2.4. Особливості розробки алгоритмів методів програмних класів або

процедур/функцій

У програмному середовищі були розроблені алгоритми методів програмних класів. Як приклад, наведемо алгоритм методу додавання афиш у обранне.

class FavouritePostersDataSourceStore @Inject constructor(  
 private val appDatabase: AppDatabase,  
 private val mapper: PictureMapper,  
 private val errorLogger: ErrorLogger  
) : AbstractDataSource() {  
  
 override val dataSourceType = DataSourceType.FAVOURITES  
 init {  
 subscribeToDatabase()  
 }  
  
 override fun loadMore() {  
 // Not required  
 }  
  
 override fun reload() {  
 // Not required  
 }  
  
 private fun subscribeToDatabase() {  
 appDatabase.favouritePicturesDao().getAll()  
 .map { mapper.toPictures(it) }  
 .map { PicturesWithLoadingState(pictures = it, loadingState = LoadingState.COMPLETED)}  
 .subscribeOn(Schedulers.io())  
 .subscribe { picturesWithLoadingState ->  
 changeState(picturesWithLoadingState)  
 }  
 }  
}

data class SetDataSourceType(  
 private val dataSourceType: DataSourceType

) : MainReducers() {  
 override fun reduce(oldState: MainState): MainState {  
 return oldState.copy(dataSourceType = dataSourceType)  
 }  
}

data class SetFavourite(val isFavourite: Boolean) : PictureDetailReducers() {  
 override fun reduce(oldState: PictureDetailState): PictureDetailState {  
 return oldState.copy(isFavourite = isFavourite)  
 }  
}

4.2.5 Особливості використання спеціалізованих бібліотек та АРІ

Для спрощення розробки програмного продукту було використано готове API календаря – React Calendar. Оскільки API має відкритий вихідний код, .календар було кастомізовано та налагоджено відповідно до потреб програмного продукту.

RxKotlin - бібліотека до Kotlin, яка може допомогти вам створювати високореактівние додатки в меншій кількості кодів.

Android KTX - набір розширень Kotlin для розробки додатків під Android. Додає переваги в розробці: менше коду, більше задоволення і полегшення розуміння коду проекту.