КОМИТЕТ ПО ОБРАЗОВАНИЮ ПРАВИТЕЛЬСТВА САНКТ-ПЕТЕРБУРГА

Санкт-Петербургское государственное

бюджетное профессиональное образование учреждение

«Колледж информационных технологий»

**ОТЧЕТ**

**по практической работе №8**

**«Приложение заметки с использованием базы данных»**

**по МДК01.03.**

Работу выполнил студент 393 гр.:

Кашицын Артем Андреевич

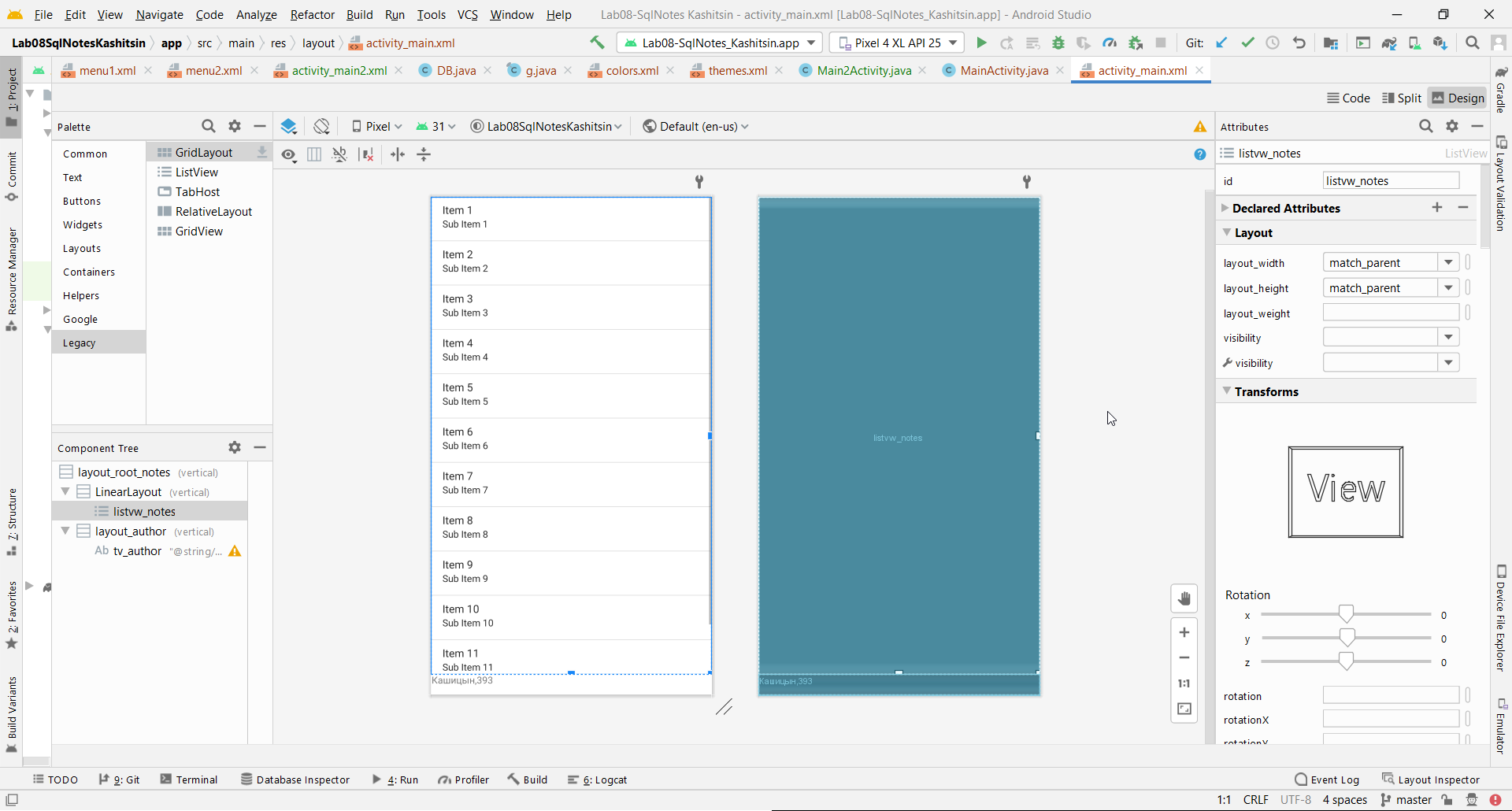
Преподаватель:

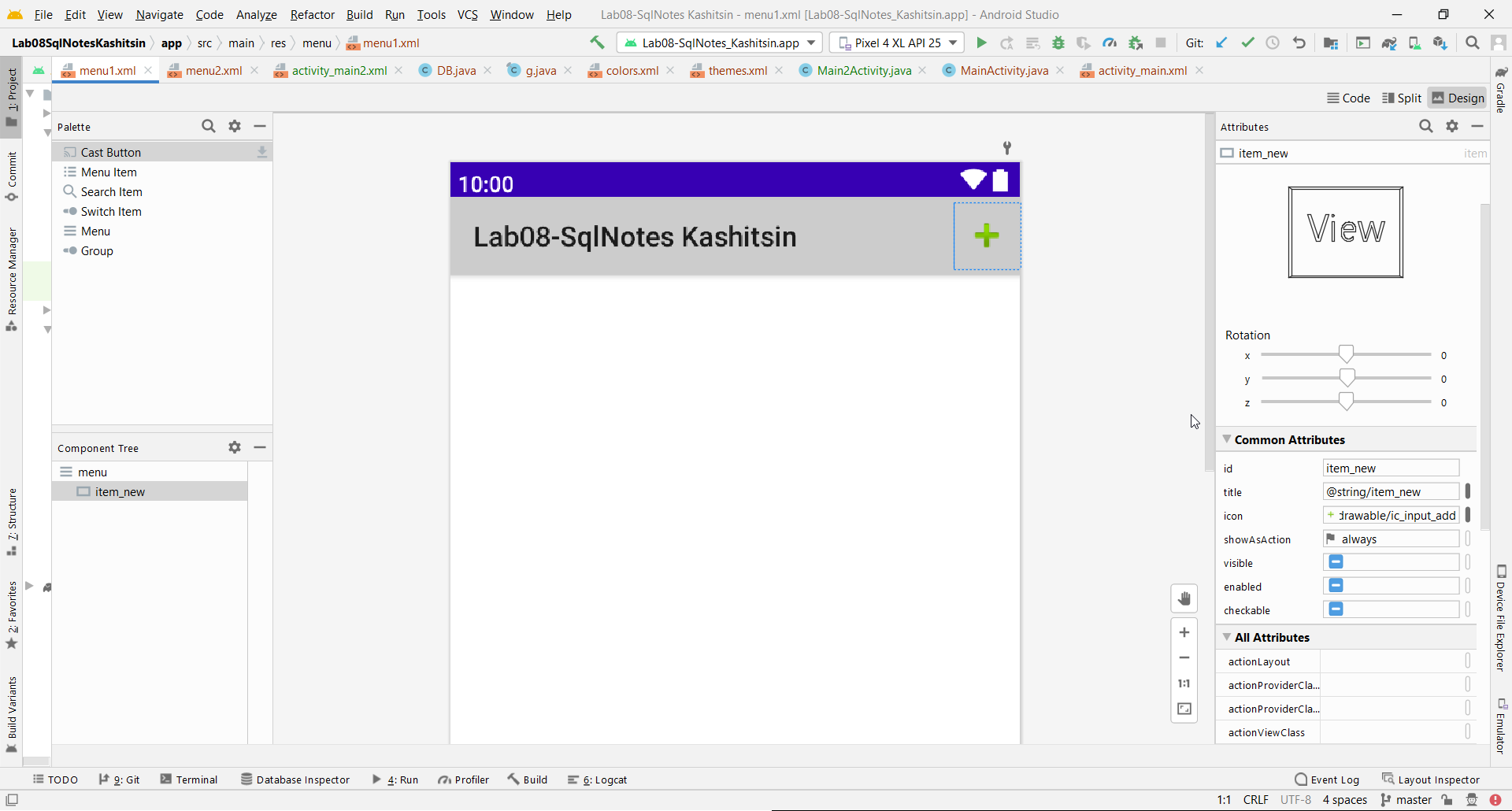
Фомин Александр Валерьевич

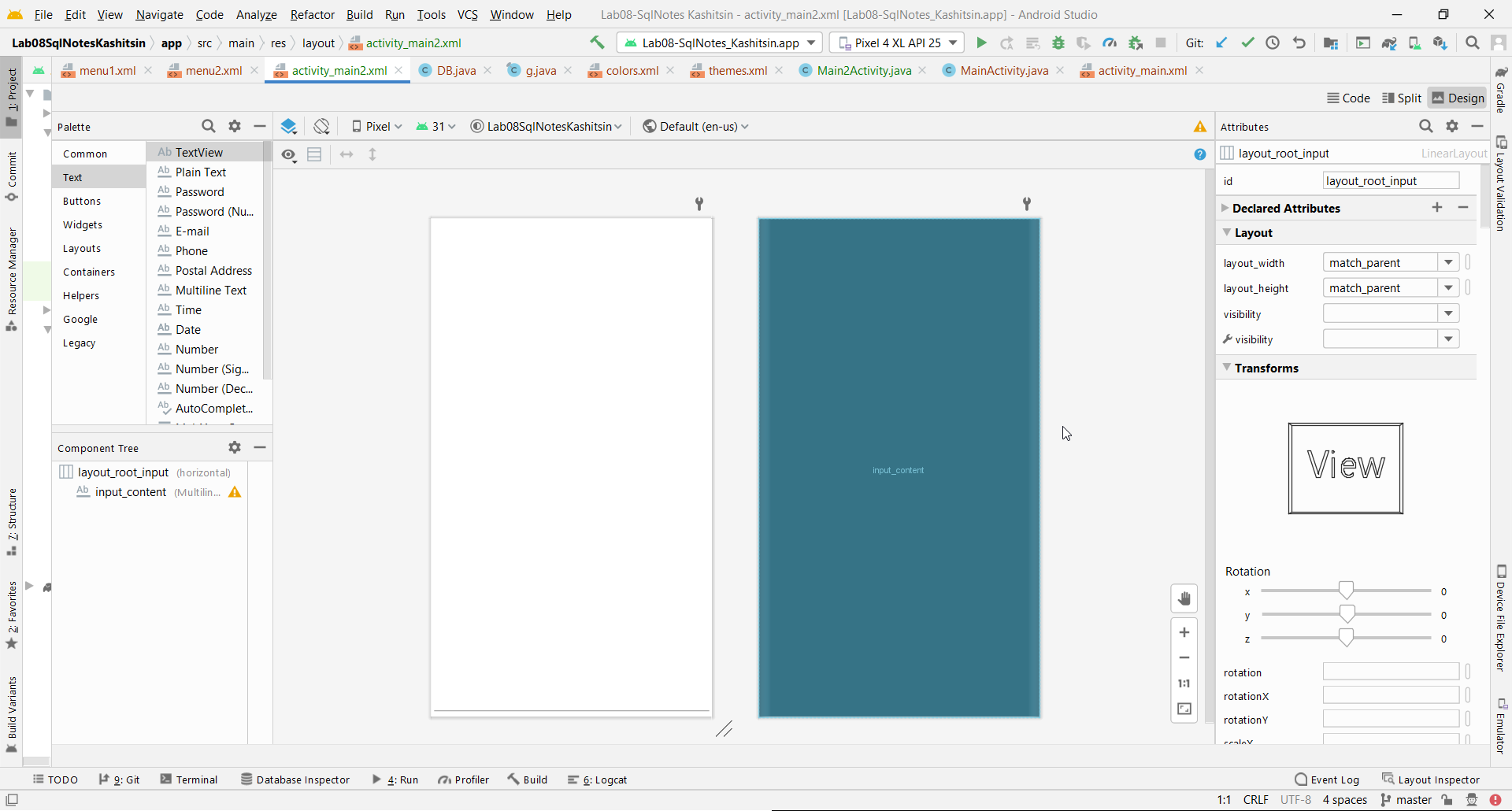
Санкт-Петербург 2021

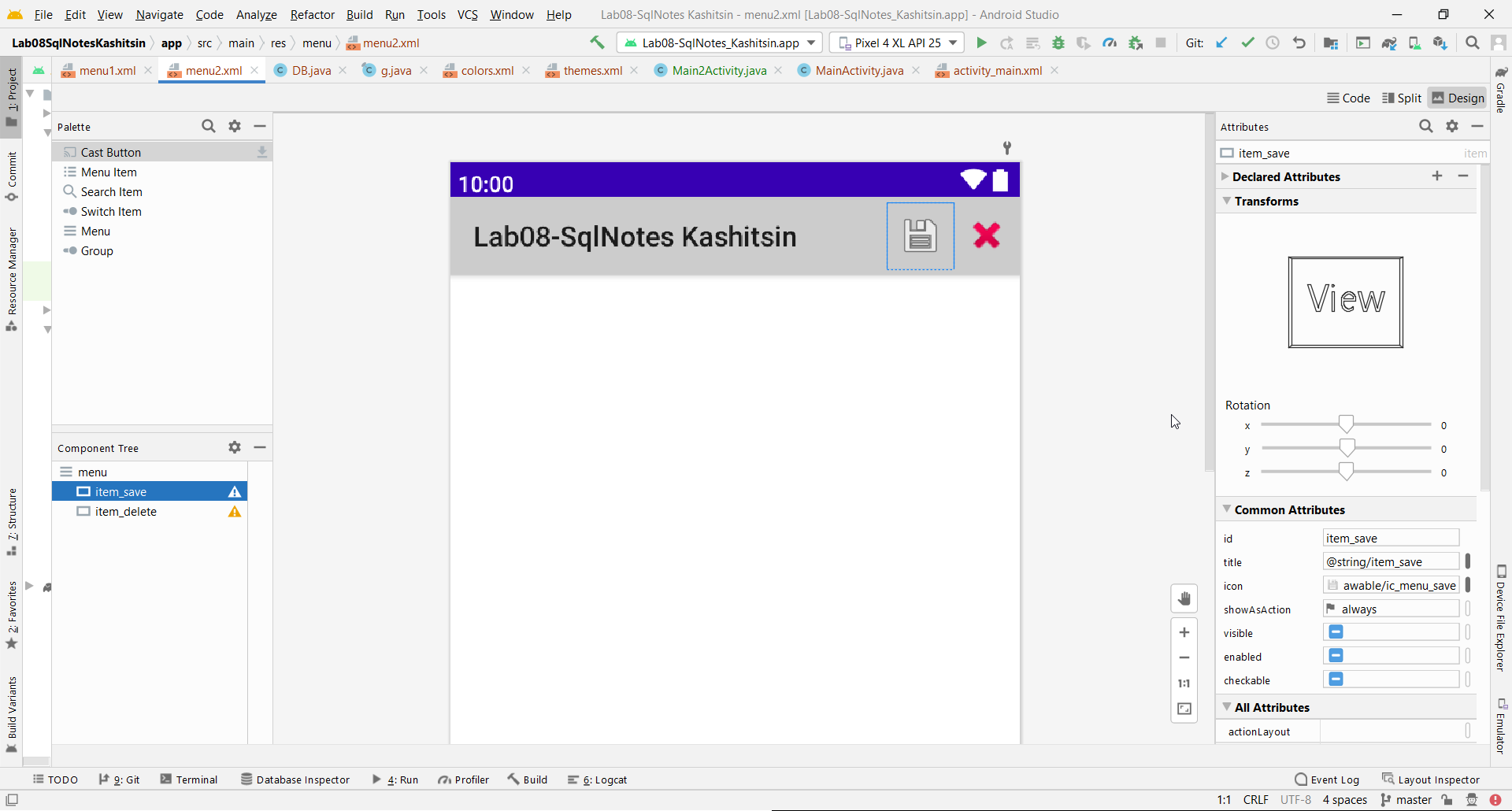
**ХОД РАБОТЫ**

На первой форме расположен ListView, в который последовательно будут добавляться заметки. К первой форме добавлено меню menu1, элемент которого непосредственно добавляет заметку. На второй форме расположен многострочный ввод текста. Ко второй форме относится меню menu2, на нем два элемента: сохранение и удаление заметки.









Для управления заметкой был создан класс Note с двумя полями: id заметки и txt – текст заметки. При попытке привести экземпляр класса к строковому типу методом toString() выводится id с текстом (содержанием) заметки.

Класс DB создан для взаимодействия с базой данных SQLite встроенным в android. Класс DB наследует класс SQLiteOpenHelper.

Процедура onCreate создает таблицу с полями id (идентификатор заметки) и txt (текст заметки).

Функция getMaxId определяет и возвращает максимальное значение идентификатора в БД – последней добавленной, на тот момент времени, заметки. Если заметок нет, по-умолчанию возвращает 0 как последний идентификатор.

Функция getNote, получая идентификатор заметки, возвращает соответствующий текст заметки. По-умолчанию возвращает «».

Процедура getAllNotes, получая ArrayList, собирает все id и txt с таблицы и каждую такую запись, отделяя поля номерами столбцов (0 – id и 1 - txt) создает экземпляр класса Notes и добавляет его в массив ArrayList.

Процедура addNote – добавляет новую запись в таблицу с переданными id и txt

Процедура saveNote – ищет заметку по переданному id и заменяет значение txt на переданное содержание заметки.

Процедура deleteNote – ищет заметку по переданному id и удаляет её из таблицы.

Первая форма:

OnCreate: В конструкторе класса DB передаём базу данных notes.db. После этого все действия с таблицей будут выполняться именно в указанной базе данных. При нажатии на элемент массива (выбрать заметку) осуществляется переход на вторую форму с переносом id и txt выбранной заметки. Выполняется процедура updateList()

onActivityResult: При возвращении на первую форму вызывается updateList()

updateList: Перезаписываем массив заметок, получив все заметки таблицы с помощью notes.getAllNotes.

onCreateOptionsMenu: Связываем menu1 с первой формой.

onOptionsItemSelected: При клике на значок добавления, создаём заметку с идентификатором на 1 больше чем максимальный идентификатор и с текстом «Hello, world!»

Вторая форма:

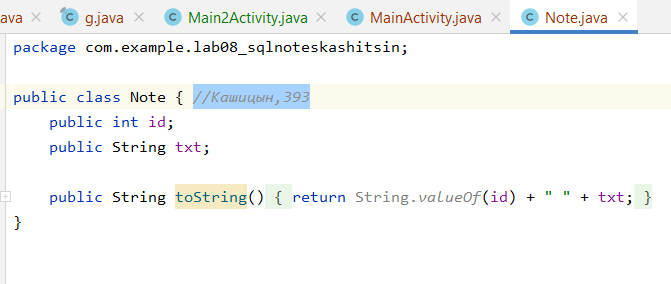
onCreateOptionsMenu: Связываем menu2 со второй формой.

onOptionsItemSelected:

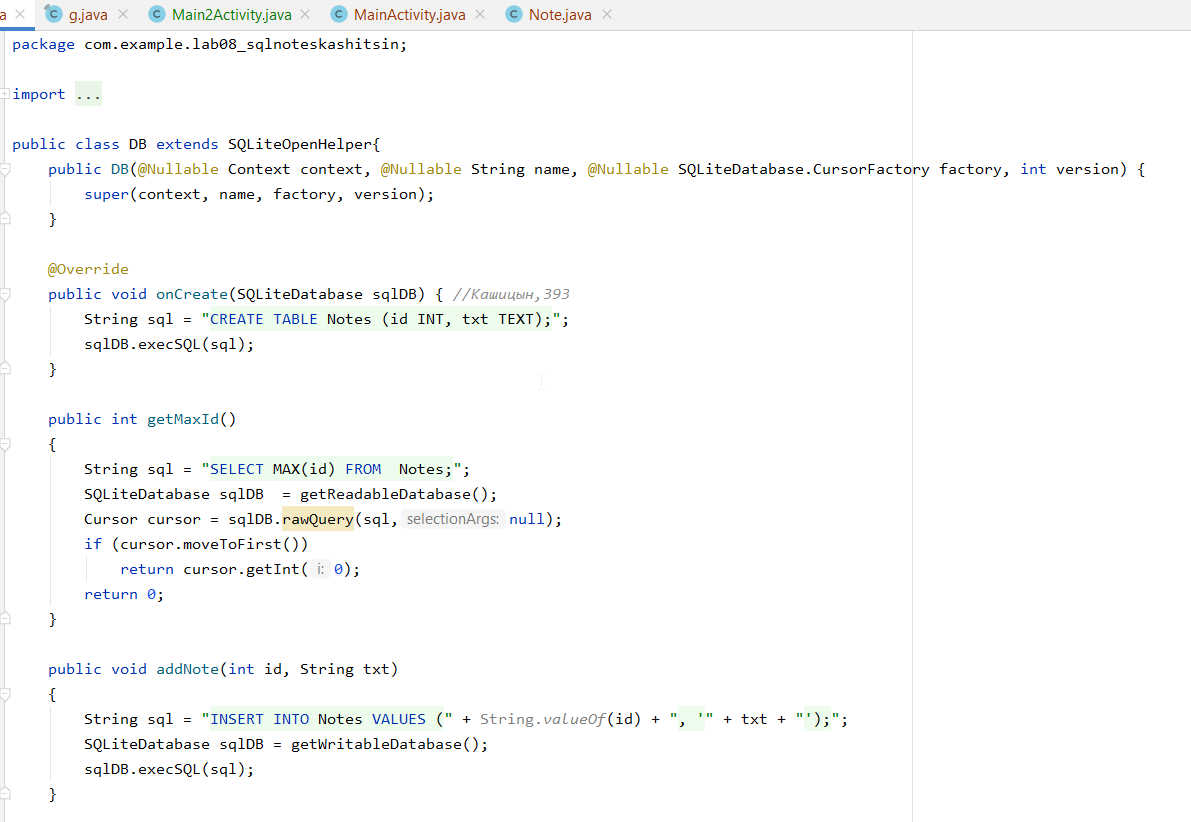
Если нажата кнопка сохранения, то отредактированный текст с переданным id заметки посылаем в notes.saveNote

Если нажата кнопка удаления, то после подтверждения, вызывается notes.deleteNote с переданным id заметки от первой формы. Форма закрывается.

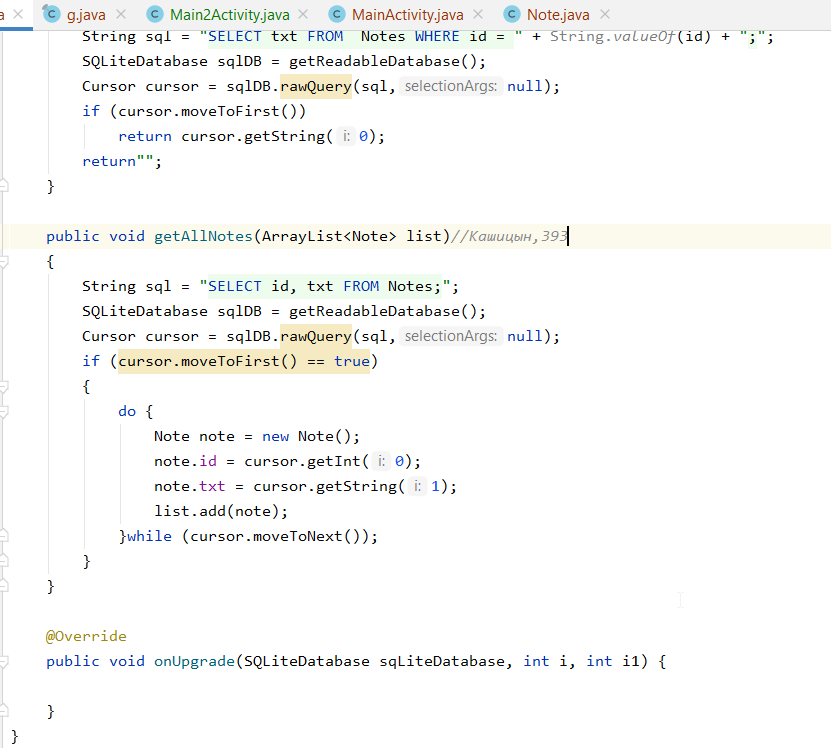
Note.java:



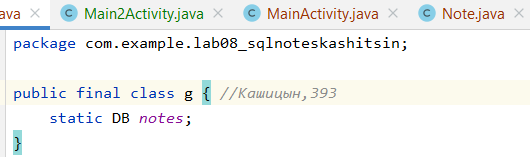
DB.java:



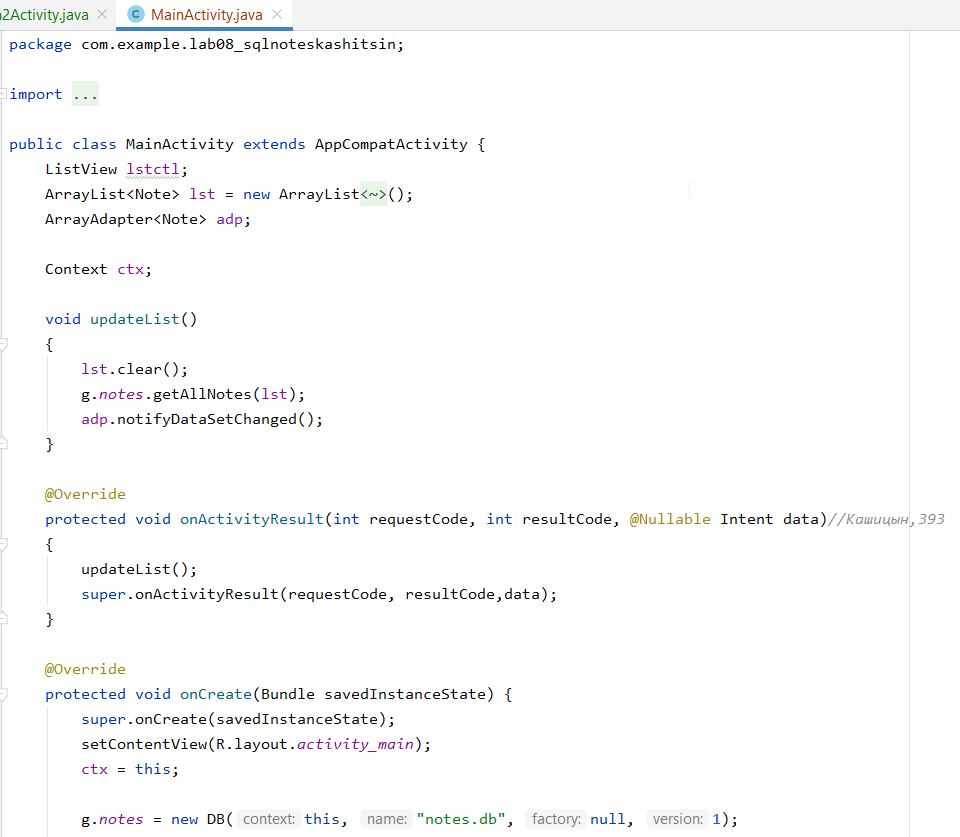




g.java:



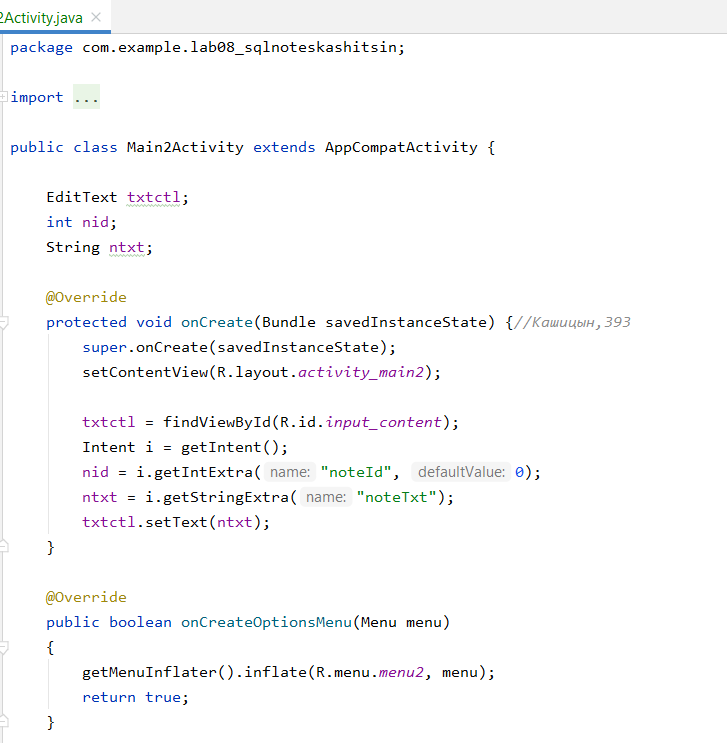
MainActivity.java:

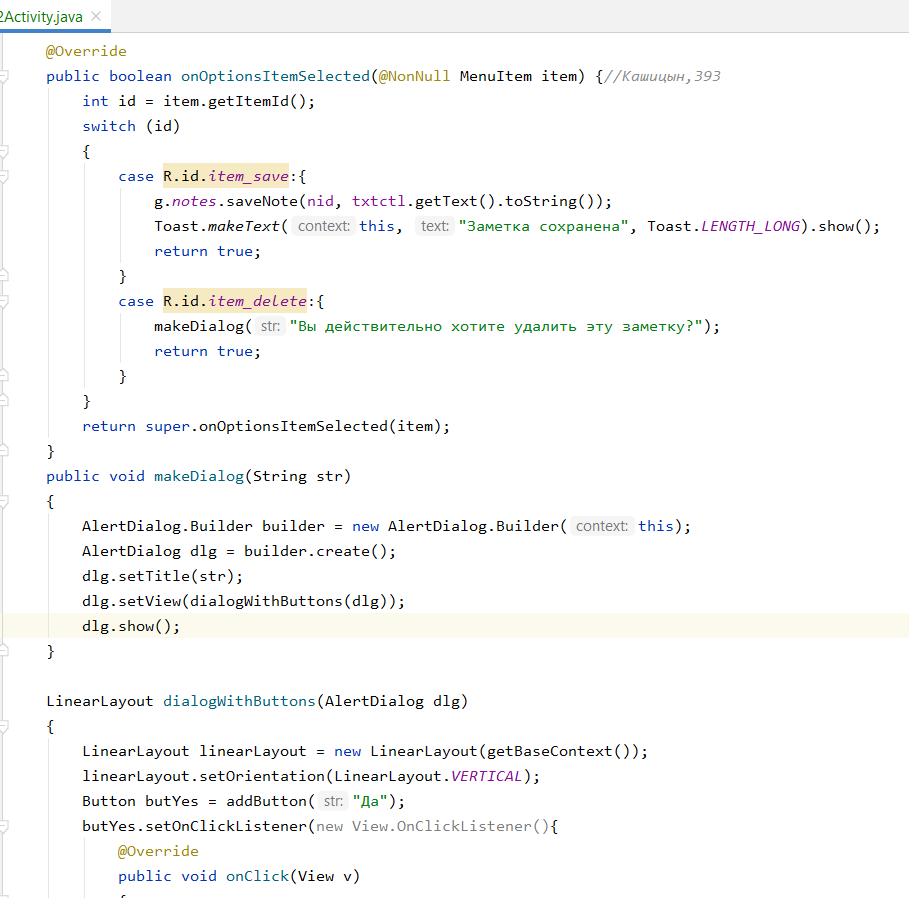


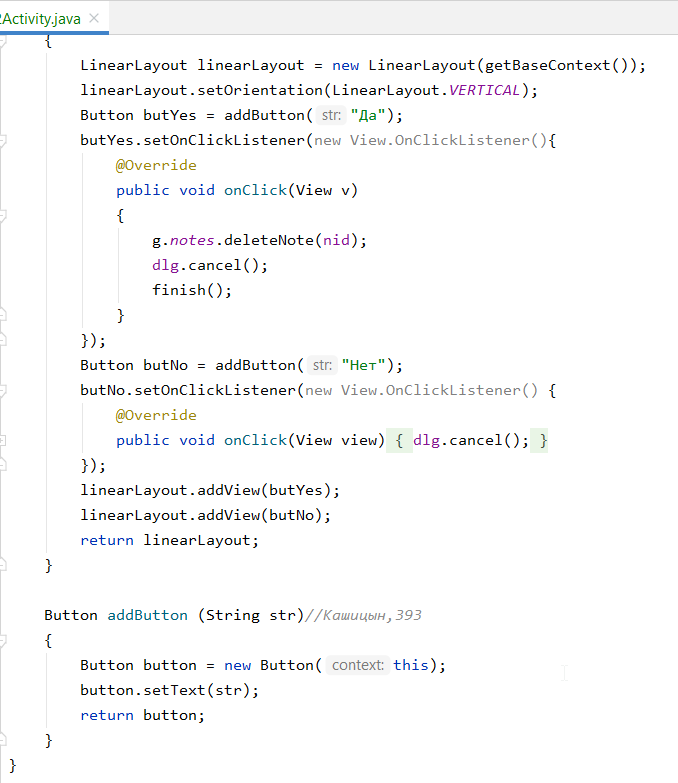




Main2Activity.java:







**ИТОГОВОЕ ПРИЛОЖЕНИЕ**



Добавление новой заметки после удаления

Удаление заметки

Отредактированные заметки

Общий вид

Добавление заметок

Сохранение заметки