

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ДОСЛІДЖЕННЯ ФУНКЦІОНАЛУ ОБ'ЄКТНОЇ СКБД

Текстова частина до курсової роботи
за спеціальністю «Інженерія програмного забезпечення» - 121

Керівник курсової роботи,
к.ф.-м.н., доцент
Ющенко Ю.О.
(прізвище та ініціали)

(підпис)

“ ___ ” _____ 2020 р.

Виконав студент _____
Романенко А.І.
(прізвище та ініціали)

“ ___ ” _____ 2020 р.

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ
Зав.кафедри інформатики,
к.ф.-м.н., доцент
Гроховський С.С.
(прізвище та ініціали)

(підпис)
“ ____ ” _____ 2020 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу

студенту Романенко Антону Ігоровичу факультету інформатики 3-го курсу
ТЕМА Дослідження функціоналу об'єктної СКБД

Зміст ТЧ до курсової роботи:

- Індивідуальне завдання
- Календарний план виконання роботи
- Анотації
- Вступ
- 1 Постановка задачі
- 2 Проектування бази даних
- 3 Конструювання запитів
- Висновки
- Список літератури

Дата видачі “ ____ ” _____ 2020 р. Ющенко Ю.О. _____
(підпис)

Завдання отримав _____
(підпис)

Тема: Дослідження функціоналу об'єктної СКБД
Календарний план виконання роботи:

№	Назва етапу	Термін виконання	Примітка
1.	Отримання теми курсової роботи	27.10.2019	
2.	Визначення інструментів реалізації	16.11.2019	
3.	Пошук тематичної літератури	12.12.2019	
4.	Ознайомлення з літературою	22.12.2019	
5.	Ознайомлення з мовою Swift	10.01.2020	
6.	Ознайомлення з СКБД Realm	24.02.2020	
7.	Дослідження предметної області	04.03.2020	
8.	Проектування бази даних	12.03.2020	
9.	Конструювання запитів	14.04.2020	
10.	Реалізація додатку	02.05.2020	
11.	Написання текстової частини	08.05.2020	

Студент Романенко А.І.

Керівник Ющенко Ю.О.

“ ”

Зміст

Анотація	5
Вступ.....	6
Основна частина.....	7
1 Постановка задачі	7
1.1 Функціональні вимоги.....	7
1.2 Технічні вимоги.....	7
1.3 Програмні вимоги	8
2 Проектування бази даних.....	8
2.1 Опис предметних областей	9
3 Конструювання запитів	11
4 Опис додатку	17
Висновки	19
Список літератури.....	20

Анотація

Головною метою визначення особливостей та набуття навичок роботи з об'єктною базою даних та розробка мобільного додатку задля реалізації певних запитів обробки інформації.

Основна частина включає в себе такі розділи: постановка задачі, проектування бази даних, конструювання запитів та опис додатку. В розділі постановки задачі описані основні задачі, функціональні, технічні та програмні вимоги. В розділі проектування бази даних описане даталогічне проектування з наведенням певних таблиць та моделей. В розділі конструювання запитів надаються коди запитів різних видів з демонстрацією вихідних даних бази. В останньому розділі описується додаток з додаванням ілюстрацій.

Ключові слова: об'єктна система керування базою даних, Realm, Swift, запити.

Вступ

Складно уявити собі якусь сферу життя без правильно упорядкованої структури в наш час. Тільки так можливо побудувати якнайбільш ефективну і працездатну модель. Без правильного подання інформації в чітко упорядкованій структурі, вибору головних факторів впливу в певній сфері і правильного облаштування функціоналу, будь-яка робоча сфера була б, якнайменш, незручною в експлуатації або взагалі непридатною для будь-якої роботи. Саме задля уникнення таких казусів у багатьох середовищах, де проводяться певні операції з інформацією, використовують бази даних.

Метою курсової роботи було визначення особливостей та набуття навичок роботи з об'єктною базою даних та розробка мобільного додатку задля реалізації певних запитів обробки інформації.

В даному проекті було створено базу даних для обліку комунальних платежів. Була проаналізована предметна область та зіставлена робоча модель, також створено мобільний додаток для смартфонів під керуванням IOS. Для проекту була обрана база даних Realm.

Realm – open-source об'єктна база даних, яка підтримується IOS/Android, Xamarin, React Native, Windows та багатьма іншими платформами, що і стало основною причиною її вибору. Також перевагами є те, що Realm швидше та зручніше ніж її аналоги, такі як CoreData або SQLite. Данна система керування базами даних(далі: СКБД) не модифікація якоїсь іншої, а є новою, побудованою з нуля, СКБД.

Джерелами дослідження стали офіційне керівництво розробника Swift (Apple) та офіційна документація Realm.

Основна частина

1 Постановка задачі

Завданням курсової роботи є створення бази даних обліку комунальних платежів. База повинна містити данні із таких предметних областей, як: власник, приміщення, платіжка. Наступним завданням є створення додатку задля надання змоги перевірити та відобразити коректну роботу існуючого функціоналу. Додаток повинен мати мінімальний для роботи з даними функціонал, мати зрозумілий інтерфейс. Програма повинна мати мінімальні системні вимоги для забезпечення швидкості та не займати багато місця.

1.1 Функціональні вимоги

Головною вимогою до бази даних є наявність необхідної інформації для роботи системи обліку комунальних платежів, таку як данні про власників, приміщення та платіжки. Точну інформацію про власника: повне ім'я, номер паспорту задля точної ідентифікації і всі приміщення, які йому належать. Інформацію про приміщення: адресу, власника, якому належить, особовий рахунок, та всі майбутні або поточні платіжки. Данні про платіжки: дату, приміщення, якому була виписана та рахунки за все види комунальних послуг. Програмний додаток повинен включати в себе функціонал для виконання обраних запитів та відображення оброблених даних з бази.

1.2 Технічні вимоги

Характеристики для стабільної роботи додатку:

частота процесора – 2.4GHz;

оперативна пам'ять – 8 Гб;

операційна система – MacOS 10.15 Catalina або новіше.

1.3 Програмні вимоги

Для коректної роботи додатку та користування ним необхідно мати систему під керуванням MacOS або IOS. Для внесення змін до бази даних (вручну) необхідні встановлені Realm Browser та XCode 11.3.

2 Проектування бази даних

База даних обліку комунальних платежів включає в себе три предметні області: Owner, Premise, Bill.

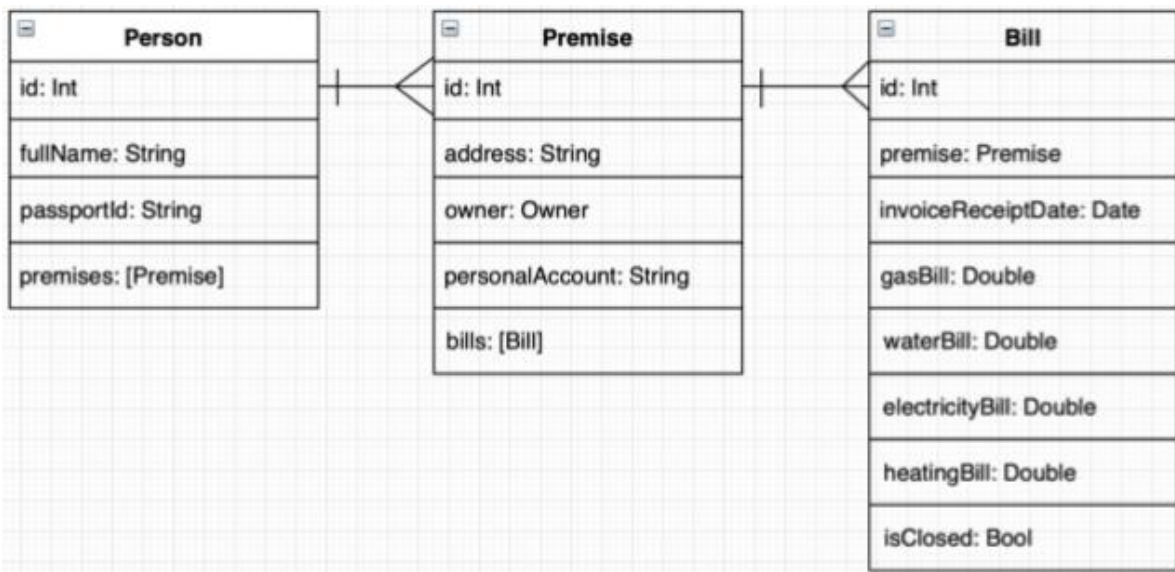


Рисунок 2.1 – модель бази даних

2.1 Опис предметних областей

Owner

Ця предметна область містить дані про власника приміщення. Кожен власник повинен мати повне ім'я, номер паспорту і приміщення, що йому належать.

Стовпці таблиці Owner:

id - Первинний ключ таблиці;

fullName - повне ім'я власника;

passportId - номер паспорту власника;

premises - усі його приміщення.

Всі вони є обов'язковими.

id Int, Primary Key	fullName String	passportId String	premises [Premise]
1	Name Surname	123456789	Premise 2
2	Alexey Alexeev	834343428	Premise 1
3	Alexandr Alexa...	210414041	Premise 1
4	Tim Cook	121212121	Premise 2
5	John Doe	111111111	Premise 1

Рисунок 2.1.1 – таблиця Owner

Premise

У цій предметній області зберігаються інформація про приміщення. Кожне приміщення повинне мати власника, адресу та особовий рахунок комунальних платежів та платіжки.

Стовпці таблиці Premise:

id - Первинний ключ таблиці;

owner – власник;

address - адреса приміщення;

personalAccount - особовий рахунок;

bills - усі платіжки цього приміщення.

Всі вони є обов'язковими.

id	owner	address	personalAccount	bills	
int	<Owner>	String	String	[Bill]	
1	(1. Name	City, street, flat 1	1234567891	Bill	1
2	(1. Name	City, street, flat2	1234567892	Bill	1
3	(3. Alexandr	California, SF, 3...	1230896781	Bill	1
4	(4. Tim Cook,	California, LA, 1...	678213456	Bill	2
5	(2. Alexey	Georgia, Atlant...	657489300	Bill	1
6	(4. Tim Cook,	California, SF, 1...	657489300	Bill	1
7	(5. John.Doe,	California, SF, 2...	657489300	Bill	5

Рисунок 2.1.2 – таблиця Premise

Bill

Данна предметна область несе платіжну інформацію, що стосується приміщення. Кожна платіжка повинна мати приміщення, до якого відноситься, дату видачі та суму до сплати за певні види послуг.

Стовпці таблиці Bill:

id - Первинний ключ таблиці;

premise - приміщення, якому належить;

invoiceReceiptDate - дата видачі;

gasBill - сума за газопостачання;

waterBill – сума за водопостачання;

electricityBill - сума за електроенергію;

heatingBill - сума за опалення;

isClosed - чи сплачена.

Всі вони є обов'язковими.

id	premise	invoiceReceiptDate	gasBill	waterBill	electricityBill	heatingBill	isClosed
int	<Premise>	Date	Double	Double	Double	Double	Boolean
1	(1. Owner...) City, street, flat1.	29 Apr 2020, 06:13:24	140,5	0	70	190,2	<input type="checkbox"/>
2	(2. Owner...) City, street, flat2.	29 Apr 2020, 06:14:22	32	49,1	77	341	<input type="checkbox"/>
3	(3. Owner...) California, SF, 3...	29 Apr 2020, 06:15:08	334	77	213	343	<input type="checkbox"/>
4	(4. Owner...) California, LA, 1...	29 Apr 2020, 06:15:35	33	37	199	202	<input type="checkbox"/>
5	(4. Owner...) California, LA, 1...	29 Apr 2020, 06:15:56	32	37	98	313	<input type="checkbox"/>
6	(5. Owner...) Georgia, Atlanta...	29 Apr 2020, 06:16:12	323	332	312	32	<input type="checkbox"/>
7	(6. Owner...) California, SF, 1...	29 Apr 2020, 06:16:51	122	89	310	12	<input type="checkbox"/>
8	(7. Owner...) California, SF, 2...	29 Apr 2020, 06:17:28	323	89,2	303	99,8	<input type="checkbox"/>
9	(7. Owner...) California, SF, 2...	30 Apr 2020, 15:55:00	323	89,2	303	99,8	<input type="checkbox"/>
10	(7. Owner...) California, SF, 2...	30 Apr 2020, 16:01:32	323	89,2	303	99,8	<input type="checkbox"/>
11	(7. Owner...) California, SF, 2...	30 Apr 2020, 16:01:50	323	89,2	303	99,8	<input type="checkbox"/>
12	(7. Owner...) California, SF, 2...	30 Apr 2020, 16:24:39	323	89,2	303	99,8	<input type="checkbox"/>

Рисунок 2.1.3 – таблиця Bill

3 Конструювання запитів

Запит 1: простий запит на вибірку

```
var ownersList: Results<Owner> {
    get {
        return realm.objects(Owner.self)
    }
}
```

Повертає список власників (їх повне ім'я та номер паспорту).

Requests	Data
	Full name:Name Surname Passport number:123456789
	Full name:Alexey Alexeev Passport number:834343428
	Full name:Alexandr Alexandrov Passport number:210414041
	Full name:Tim Cook Passport number:121212121
	Full name:John Doe Passport number:111111111

Запит 2: вибірка з використанням BETWEEN AND

```
var res = self.realm.objects(Premise.self).filter("id BETWEEN {0,2}")
premises = res
```

Повертає список приміщень, id яких знаходяться між 0 і 2.

Requests	Data
	Owner:Vanya Ivanov Address:California, SF, 23131 Personal account:657489300
	Owner:Alexey Alexeev Address:California, AYE, 11111 Personal account:1111111

Запит 3: вибірка з використанням IN

```
var res = self.realm.objects(Premise.self).filter("id IN {1,4}")
premises = res
```

Повертає список приміщень, id яких належать масиву {1,4}.

Requests	Data
	Owner:Vanya Ivanov Address:California, SF, 23131 Personal account:657489300
	Owner:Tim Cook Address:California, SA, 09385123 Personal account:12858291

Запит 4: вибірка з використанням DISTINCT

```
var res = self.realm.objects(Owner.self).distinct(by: ["fullName"])
owners = res
```

Повертає список власників, який не включає повних тезок (людей з однаковими прізвищем та іменем).

Requests	Data
	Full name:Name Surname
	Full name:Alexey Alexeev
	Full name:Alexandr Alexandrov
	Full name:Tim Cook
	Full name:John Doe

Запит 5: запит виду MIN або MAX

```
let maxValue = realm.objects(Bill.self).max(ofProperty: "gasBill") as
Double?
```

```
self.max = maxValue
```

Повертає найбільше значення платіжки за газ.

Requests	Data
	334.0

Запит 6: запит виду SUM або AVG

```
let sumValue = realm.objects(Bill.self).sum(ofProperty: "gasBill") as
Double
```

```
self.sum = sumValue
```

Повертає суму всіх платіжок за газ.

Requests	Data
	2631.5

Запит 7: запит з функцією COUNT

```
let billsC = realm.objects(Premise.self).filter("bills.@count > 0")
self.count = billsC.count
```

Повертає кількість приміщень з кількістю платіжок більше 0.

Requests	Data
	7

Запит 8: вибірка з використанням агрегатної функції і додаванням полів

```
let maxWaterBill = realm.objects(Bill.self).sorted(byKeyPath:
"waterBill").last
self.maxBill = maxWaterBill
```

Повертає платіжку з найбільшою сумою за водопостачання.

Requests	Data
	ID:6 Water bill:332.0 Gas bill:323.0 Electricity bill:312.0 Heating bill:32.0 Date:29-Apr-20...

Запит 9: вибірка з використанням агрегатної функції і умовою на вибірку поля

```
let billRes = realm.objects(Bill.self).filter("id BETWEEN
{0,3}").value(forKeyPath: "gasBill") as! [Double]
let avgGasBill = billRes.reduce(0, +) / Double(max(billRes.count, 1))
self.avg = avgGasBill
```

Повертає середнє значення суми за газопостачання серед платіжок з id від 0 до 3.

Requests	Data
	190.0

Запит 10: вибірка з використанням агрегатної функції і умовою з агрегатною функцією

```
let billRes = realm.objects(Bill.self).filter("id BETWEEN {0,3} AND
gasBill <= 300").min(ofProperty: "waterBill") as Double?
self.min = billRes
```

Повертає мінімальну суму за водопостачання серед платіжок з id від 0 до 3 та сумою за газопостачання меншою або рівною 30

Requests	Data
	89.2

Запит 11: вибірка з використанням агрегатної функції, умовою з агрегатною функцією та умовою на вибірку поля з сортуванням

```
let billRes = realm.objects(Bill.self).filter("invoiceReceiptDate < %@ AND
electricityBill >= \(((realm.objects(Bill.self).min(ofProperty: "electricityBill") as
Double?).unsafelyUnwrapped)", Date()).sorted(byKeyPath: "id")
self.bills = billRes
```

Повертає платіжки, які видані раніше поточної дати, сума за електропостачання більше мінімальної і вони відсортовані по id.

Requests	Data
	ID:1 Water bill:95.0 Gas bill:124.0 Electricity bill:303.0 Heating bill:99.8 Date:11-May-2020
	ID:2 Water bill:89.2 Gas bill:123.0 Electricity bill:352.0 Heating bill:251.0 Date:11-May-2020
	ID:3 Water bill:89.2 Gas bill:323.0 Electricity bill:303.0 Heating bill:99.8 Date:11-May-2020
	ID:4 Water bill:111.0 Gas bill:295.0 Electricity bill:172.0 Heating bill:99.8 Date:11-May-2020
	ID:5 Water bill:112.0 Gas bill:323.0 Electricity bill:293.0 Heating bill:108.0 Date:11-May-2020

Запит 12: запит з використанням вкладеного запиту та умовами <, =, >

```
let gasBills = realm.objects(Bill.self).filter("gasBill >
\\((realm.objects(Bill.self).average(ofProperty: "gasBill") as
Double?).unsafelyUnwrapped)")
self.bills = gasBills
```

Повертає платіжки, в яких сума за газ більша за середню.

Requests	Data
ID:3	Water bill:89.2 Gas bill:323.0 Electricity bill:303.0 Heating bill:99.8 Date:11-May-2020
ID:4	Water bill:111.0 Gas bill:295.0 Electricity bill:172.0 Heating bill:99.8 Date:11-May-2020
ID:5	Water bill:112.0 Gas bill:323.0 Electricity bill:293.0 Heating bill:108.0 Date:11-May-2020

Запит 13: запит з використанням вкладеного запиту та агрегатної функції

```
let electricityBills = realm.objects(Bill.self).filter("electricityBill >
\\((realm.objects(Bill.self).min(ofProperty: "electricityBill") as
Double?).unsafelyUnwrapped)")
self.bills = electricityBills
```

Повертає платіжки, в яких сума за електропостачання більша за мінімальну (всі крім мінімальної).

Requests	Data
ID:1	Water bill:95.0 Gas bill:124.0 Electricity bill:303.0 Heating bill:99.8 Date:11-May-2020
ID:2	Water bill:89.2 Gas bill:123.0 Electricity bill:352.0 Heating bill:251.0 Date:11-May-2020
ID:3	Water bill:89.2 Gas bill:323.0 Electricity bill:303.0 Heating bill:99.8 Date:11-May-2020
ID:5	Water bill:112.0 Gas bill:323.0 Electricity bill:293.0 Heating bill:108.0 Date:11-May-2020

Запит 14: запит з використанням вкладеного запиту та умовами ANY або SOME

```
let heatingBills = realm.objects(Bill.self).filter("heatingBill > 400")
if heatingBills.count > 0 {
    anyBool = true
```

```

} else {
    anyBool = false
}

```

Повертає певне значення логічного оператора, умовою якого є існування платіжки з сумою за опалення більше 400.

Requests	Data
	false

Запит 15: запит з використанням вкладеного запиту та функцією IN

```

let ownersWithNames = realm.objects(Owner.self).filter("fullName IN
{'Tim Cook', 'John Doe'}")
self.owners = ownersWithNames

```

Повертає власників, імена яких належать даному масиву.

Requests	Data
	Full name:Tim Cook Passport number:121212121
	Full name:John Doe Passport number:111111111

Запит 16: запит з використанням вкладеного запиту та умовою LIKE

```

var res = self.realm.objects(Owner.self).filter("passportId LIKE '*111?")
print(res)
owners = res

```

Повертає власників, номер паспорту яких закінчується на 111N, де N – одна будь-яка цифра.

Requests	Data
	Full name:John Doe Passport number:111111111

Запит 17: запит з кон'юнкцією двох умов

```
var res = self.realm.objects(Owner.self).filter("passportId LIKE '*4*' AND
fullName CONTAINS 'Alex'")
```

```
print(res)
```

```
owners = res
```

Повертає власників, в номері паспорту яких є цифра 4 і ім'я Alex.

Requests	Data
	Full name:Alexey Alexeev Passport number:834343428
	Full name:Alexandr Alexandrov Passport number:210414041

Запит 18: запит з диз'юнкцією двох умов

```
var res = self.realm.objects(Owner.self).filter("id == 2 or id == 3")
```

```
print(res)
```

```
owners = res
```

Повертає власників, id яких 2 або 3.

Requests	Data
	Full name:Alexey Alexeev Passport number:834343428
	Full name:Alexandr Alexandrov Passport number:210414041

4 Опис додатку

Додаток створений для виконання запитів та відображення таблиць даних. Головне меню складається зі списку запитів, які можна обрати. Після натискання на будь який із варіантів, відбувається виведення результатів виконання.



Рисунок 4.1 – меню запитів

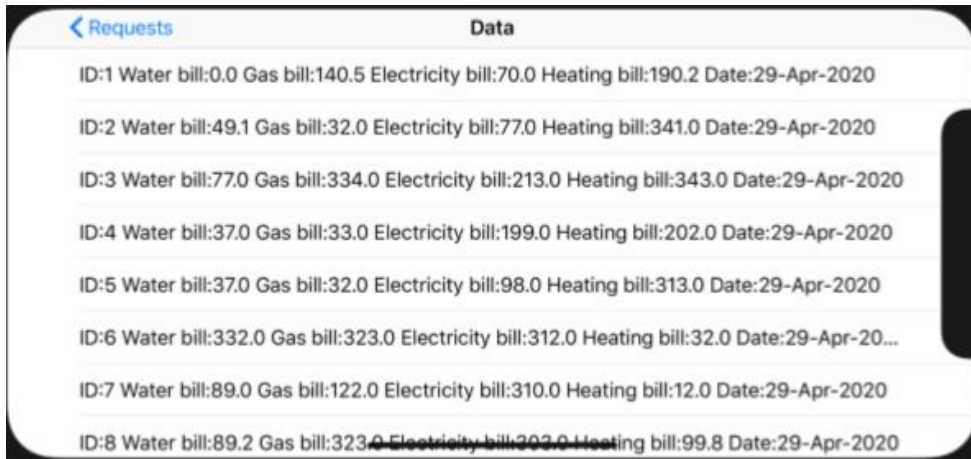


Рисунок 4.2 – результат виконання запиту

Висновки

Під час виконання завдання курсової роботи була досліджена предметна область, для якої застосовний даний проект. Взявши за основу отриману інформацію, була побудована модель бази даних та розроблена сама база даних, що відповідає потребам предметної області. Для розробки бази був використаний сервер Realm. База складається з 3 таблиць, що зв'язані між собою таким чином, що на їх основі можна було побудувати реальну, працюючу бізнес модель для сфери надання комунальних послуг.

Також було розроблено клієнтський додаток з використанням мов, технологій та плагінів таких як: XCode 11.3, UIKit, RealmSwift.

Як результат виконання завдання курсового проекту, були отримані певні навички та досвід роботи з перерахованими вище технологіями, а також розроблений додаток для тестування функціоналу реальних запитів.

Список літератури

1. Realm.io, офіційний сайт Realm: <https://realm.io>
2. Realm.io, Документація Realm для мови Swift: <https://realm.io/docs/swift/latest/>
3. Developer.apple.com, керівництво по Swift: <https://developer.apple.com/swift/>
4. Developer.apple.com, Документація по UIKit: <https://developer.apple.com/documentation/uikit>