

Курсовий проект

з дисципліни «Організація баз даних та знань»

на тему **СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ
КОПІРУВАЛЬНОГО ЦЕНТРУ**

Виконав(ла): студент(ка) 3 курсу, групи ПІ-3.2.??

Керівник проф. Малахов Є.В.

Національна шкала _____

Кількість балів _____ Оцінка ECTS _____

ЗМІСТ

ВСТУП	3
1 ПОСТАНОВКА ЗАВДАННЯ	4
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	7
3 ІНФОРМАЦІЙНА МОДЕЛЬ ПРЕДМЕТНОЇ ОБЛАСТІ	8
4 ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	12
5 МОДЕЛЬ ПРОГРАМНОГО ДОДАТКУ	13
6 СТВОРЕННЯ БАЗИ ДАНИХ	16
7 ЗАПИТИ ДО БАЗИ ДАНИХ ДЛЯ РОЗВ’ЯЗАННЯ ПОСТАВЛЕНИХ ЗАДАЧ	19
8 ПРОГРАМНА РЕАЛІЗАЦІЯ КЛІЄНТСЬКОГО ДОДАТКУ	22
8.1 Додаток клієнта копіцентру (веб-сайт)	22
8.2 Додаток співробітника та керуючого копіцентру	23
9 ІНСТРУКЦІЯ КОРИСТУВАЧА З ІЛЮСТРАЦІЯМИ	25
9.1 Клієнт копіцентру	25
9.2 Співробітник копіцентру	27
9.3 Керуючий копіцентром	30
ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ	33
ПЕРЕЛІК ПОСИЛАНЬ	34
Додаток А СХЕМА БАЗИ ДАНИХ	35
Додаток Б ДІАГРАМИ КЛАСІВ	36
Додаток В ЗАПИТИ ДЛЯ СТВОРЕННЯ БАЗИ ДАНИХ	45
Додаток Г ПРОГРАМНИЙ КОД	59

ВСТУП

Сучасна людина майже щодня взаємодіє з різноманітними інформаційними системами: перевіряє залишок коштів на карті, відслідковує місцезнаходження посилок, записується до лікаря, сплачує комунальні послуги та ін. Грамотно спроектована та стабільно працююча інформаційна система здатна спростити та прискорити більшість повсякденних задач. Такі системи стали невід'ємною частиною нашого життя, тому їх проектування, створення та впровадження є розповсюдженим та актуальним завданням.

Предметна область – це копірувальний центр, в якому надаються послуги друку, дизайнерські послуги та здійснюється продаж канцелярських товарів. Метою курсового проекту є проектування та створення інформаційної системи для автоматизації повсякденних процесів (прийом та продаж товару, виконання послуг, управління співробітниками та ін.) у копіцентрі. Для досягнення такої мети потрібно розв'язати наступні задачі:

- проаналізувати предметну область, виокремити користувачів системи;
- виконати проектування бази даних;
- обґрунтувати вибір засобів та технологій розробки;
- розробити базу даних на основі системи керування базами даних PostgreSQL;
- розробити Front-End частину застосунку;
- розробити Back-End частину застосунку;
- протестувати програмний застосунок.

1 ПОСТАНОВКА ЗАВДАННЯ

В інформаційній системі передбачено три типи користувачів: керуючий копіцентром, співробітник та клієнт. У табл. 1.1. наведено перелік задач для кожного з користувачів із зазначенням вхідної та вихідної інформації.

Таблиця 1.1. – Задачі користувачів

Задача	Вхідна інформація	Вихідна інформація
Клієнт		
1. Перегляд цін та асортименту послуг копіцентру		Сторінка з послугами копіцентру
2. Перегляд повної інформації про замовлення	Номер телефону Прізвище	Сторінка з замовленнями клієнта
3. Відгук про роботу копіцентру	Ім'я Телефон Текст відгуку Дата	
Співробітник		
1. Управління клієнтами копіцентру		
1.1 Реєстрація нового клієнта в системі	Прізвище Ім'я Номер телефону Дата реєстрації	
1.2 Зміна інформації про клієнта	Ідентифікатор клієнта	
1.3 Видалення інформації про клієнта	Ідентифікатор клієнта	
2. Оформлення та видача замовлення		
2.1. Додавання інформації про замовлення	Ідентифікатор клієнта Назва послуги Кількість Дата	
2.2. Оновлення статусу замовлення	Номер замовлення	
2.3. Перегляд інформації про замовлення для видачі	Номер замовлення/прізвище	Сторінка з замовленнями даного клієнта
3. Продаж товару		
3.1. Додавання інформації про покупку в бд	Ідентифікатор клієнта Ідентифікатор товару Ідентифікатор співробітника Ціна Кількість	Сторінка з товарними чеками

Продовження таблиці 1.1

Задача	Вхідна інформація	Вихідна інформація
4. Прийом товару та витратних матеріалів		
4.1. Додавання інформації про поставку товару	Постачальник Назва товару Ціна Кількість	Сторінка з накладними на товар
4.2. Додавання інформації про постачання витратних матеріалів	Постачальник Назва витратного матеріалу Ціна Кількість	Сторінка з накладними на витратні матеріали
5. Формування замовлення постачальникам		
5.1. Формування замовлення постачальникам на витратні матеріали і канцелярію	Назва Кількість	
6. Перегляд списку постачальників		
6.1. Перегляд списку постачальників		Сторінка з переліком постачальників копіцентру
7. Перегляд залишків продукції та витратних матеріалів		
7.1 Перегляд залишків продукції		Сторінка з інформацією про залишки продукції
7.2 Перегляд залишків витратних матеріалів		Сторінка з інформацією про залишки витратних матеріалів
8. Перегляд відгуків про роботу копіцентру		
8.1 Перегляд відгуків про роботу копіцентру		Сторінка з відгуками
9. Перегляд відповідностей між витратними матеріалами та послугами		
9.1 Перегляд відповідностей між витратними матеріалами та послугами		Сторінка зі встановленими відповідностями між матеріалами та послугами
Директор		
1. Установка цін і асортименту послуг копіцентра		
1.1 Додавання інформації про послугу	Назва послуги Опис Ціна за одиницю	Сторінка з послугами
1.2 Редагування інформації про послугу	Назва послуги	
2. Управління співробітниками копіцентра		
2.1 Редагування інформації про співробітника	Ідентифікатор співробітника	
2.2 Видалення інформації про співробітника	Ідентифікатор співробітника	

Продовження таблиці 1.1.

Задача	Вхідна інформація	Вихідна інформація
2.3 Додавання інформації про співробітника в БД	Прізвище Ім'я По батькові Мобільний номер Сімейний стан Освіта Адреса Дата найму Дата народження Логін Пароль	
3. Контроль за поставками		
3.1 Перегляд накладних на товари		Сторінка з накладними на товари
3.2 Перегляд накладних на витратні матеріали		Сторінка з накладними на витратні матеріали
3.3 Перегляд наявності товару		Сторінка з інформацією про залишки товарів
3.4 Перегляд наявності витратних матеріалів		Сторінка з інформацією про залишки витратних матеріалів
3.5 Перегляд списку товарів, які потрібно замовити		Сторінка з запитом на замовлення постачальникам
4. Робота з постачальниками		
4.1 Додавання інформації про постачальників	Назва компанії Контактна особа Номер телефону	
4.2 Редагування інформації про постачальників	Назва компанії	
5. Перегляд товарних чеків		Сторінка з товарними чеками
6. Перегляд замовлень клієнтів		Сторінка з клієнтськими замовленнями
7. Встановлення відповідності між витратними матеріалами та послугами	Назва послуги Назва витратного матеріалу	
8. Перегляд клієнтів копіцентру		Сторінка з інформацією про клієнтів
9. Перегляд відгуків про копіцентр		Сторінка з відгуками про копіцентр

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Для розв'язання поставлених задач серед багатьох видів системних архітектур було обрано триланкову архітектуру, що складається з серверу бази даних (рівень управління ресурсами), серверу додатків (рівень прикладного компоненту) та клієнтів (рівень представлення даних). Взаємодія між ними відбувається наступним чином: клієнт надсилає запит на перегляд сторінки який обробляється сервером додатків, після чого сервер додатків за потреби звертається до серверу бази даних та надсилає відповідь у вигляді запитуваної сторінки клієнту. Схема цієї взаємодії наведена на рис. 2.1.



Рисунок 2.1 – Схема триланкової архітектури

В якості шаблону проектування обрано MVC (Model, View, Controller), де модель представляє дані та методи їх обробки, контролер забезпечує зв'язок користувача з додатком, а вид отримує дані з моделі та відображає в інтерфейсі користувача. Схему взаємодії між моделлю, контролером та видом наведено на рис. 2.2.

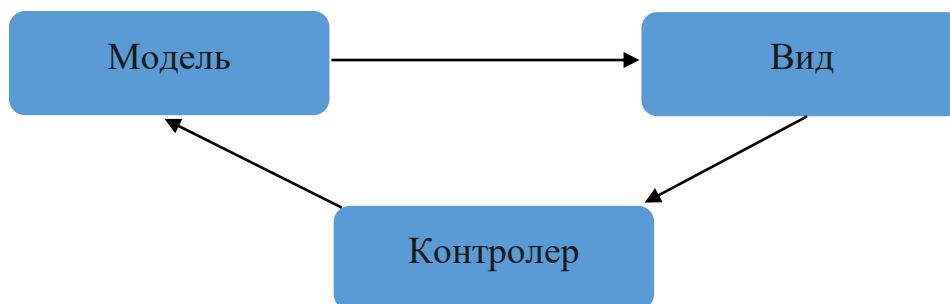


Рисунок 2.2 – Схема взаємодії між моделлю, контролером та видом

3 ІНФОРМАЦІЙНА МОДЕЛЬ ПРЕДМЕТНОЇ ОБЛАСТІ

Сутності та їх властивості з описом обмежень, що потрібні для розв'язання поставлених задач, наведені в таблиці 3.1.

Таблиця 3.1 – Опис сутностей та їх властивостей

Властивість	Опис	Обмеження
Об'єкт «Service»		
service_name	Назва послуги	PRIMARY KEY (PK)
price_per_one	Ціна за одиницю	NOT NULL
desc	Опис послуги	-
Об'єкт «Emp_main»		
e_id	Ідентифікатор	PK
last_name	Прізвище	до 30 симв, NOT NULL
first_name	Ім'я	до 30 симв, NOT NULL
mobile_number	Номер телефону	10 симв, NOT NULL, UNIQUE
username	Ім'я користувача для входу	NOT NULL, UNIQUE
password	Пароль для входу	>5 симв, NOT NULL
Об'єкт «Emp_add»		
e_id	Ідентифікатор	PK
birth_date	Дата народження	NOT NULL
emp_date	Дата прийому на роботу	NOT NULL
patronymic	По-батькові	NOT NULL
family_status	Сімейний стан	-(не)женат, (не)замужем NOT NULL
education	Освіта	-средн, ср-спец, неок-высш, высшее NOT NULL
address	Адреса	NOT NULL
Об'єкт «Product»		
p_id	Ідентифікатор	PK
product_name	Назва товару	до 40 симв, NOT NULL
Об'єкт «Goods_sup»		
company_name	Назва компанії	PK
contact_person	Контактна особа	NOT NULL
tel_num	Телефон	10 симв, NOT NULL, UNIQUE
Об'єкт «Cons_mat»		
cm_id	Ідентифікатор	PK
cm_name	Назва	NOT NULL, UNIQUE
Об'єкт «Client»		
c_id	Ідентифікатор	PK

Продовження таблиці 3.1

Властивість	Опис	Обмеження
last_name	Прізвище	до 30 симв, NOT NULL
first_name	Ім'я	до 30 симв, NOT NULL
mobile_number	Номер телефону	до 10 симв, NOT NULL, UNIQUE
reg_date	Дата реєстрації	NOT NULL
Об'єкт «C_order»		
o_id	Ідентифікатор замовлення	PK
ord_status	Статус замовлення	до 25 симв, NOT NULL
o_date	Дата оформлення	NOT NULL
Об'єкт «Ord_receipt»		
service_name	Послуга	PK
c_id	Ідентифікатор клієнта	PK, FOREIGN KEY (FK)
o_id	Ідентифікатор замовлення	PK, FK
amount	К-сть	NOT NULL
Об'єкт «Cons_invoice»		
inv_id	Ідентифікатор	PK
i_date	Дата	NOT NULL
company_name	Компанія	NOT NULL
Об'єкт «Cons_invoice_add»		
inv_id	Ідентифікатор	PK, FK
cm_id	Витратний матеріал	NOT NULL, FK
amount	К-сть	NOT NULL
price	Ціна	NOT NULL
Об'єкт «Goods_invoice»		
inv_id	Ідентифікатор	PK
i_date	Дата	NOT NULL
company_name	Компанія	NOT NULL
Об'єкт «Goods_invoice_add»		
inv_id	Ідентифікатор	PK, FK
cm_id	Товар	NOT NULL, FK
amount	К-сть	NOT NULL
purchase_price	Ціна при закупівлі	NOT NULL
Об'єкт «Goods_receipt»		
rec_id	Ідентифікатор	PK
r_date	Дата	NOT NULL
e_id	Співробітник	NOT NULL, FK
c_id	Клієнт	NOT NULL, FK
Об'єкт «Goods_receipt_add»		
rec_id	Ідентифікатор	PK, FK

Продовження таблиці 3.1

Властивість	Опис	Обмеження
p_id	Товар	NOT NULL, FK
amount	К-сть	NOT NULL
price	Ціна при продажу	NOT NULL
Об'єкт «Contact»		
id	Ідентифікатор	PK
person	Ім'я	NOT NULL
phone	Телефон	NOT NULL
text	Текст відгуку	NOT NULL
date	Дата	NOT NULL
Об'єкт «Cons_serv»		
cm_id	Ідентифікатор витратного матеріалу	PK, FK
service_name	Назва послуги	PK, FK
Об'єкт «Buy_request»		
id	Ідентифікатор	PK
name	Товар	NOT NULL
amount	Кількість	NOT NULL

Між сутностями наявні 4 типи зв'язків:

- «один-до-одного»;
- «один-до-багатьох» ;
- «багато-до-багатьох»;
- «n-арний зв'язок»

Зв'язок «один-до-одного» реалізований між таблицями emp_main та emp_add. Для формалізації цього типу зв'язку до таблиці emp_add внесено первинний ключ таблиці emp_main у якості зовнішнього ключа [1].

Зв'язок «один-до-багатьох» наявний між таблицями:

- client та goods_receipt, ord_receipt;
- c_order та ord_receipt;
- emp_main та goods_receipt;
- product та goods_receipt_add, goods_invoice_add;
- service та ord_receipt, cons_serv;
- cons_mat та cons_invoice_add;
- goods_sup та cons_invoice, goods_invoice;
- goods_invoice та goods_invoice_add;
- cons_invoice та cons_invoice_add;
- goods_receipt та goods_receipt_add.

Для формалізації даного типу зв'язку первинний ключ однозв'язної сутності додається до схеми N-зв'язної сутності у якості зовнішнього ключа [1].

Зв'язок «багато-до-багатьох» виникає між таблицями `service` та `cons_mat`. Для формалізації даного зв'язку була створена допоміжна таблиця `cons_serv`, яка містить первинні ключі цих двох таблиць. Ця множина атрибутів є первинним ключем таблиці, оскільки вона не містить інших атрибутів [1].

N-арний зв'язок пов'язує три таблиці: `client`, `c_order` та `service` для формування інформації про замовлення. Для його формалізації було створено допоміжну таблицю `ord_receipt`, первинним ключем якої є множина первинних ключів всіх трьох таблиць, що беруть участь у зв'язку [1]. До цієї таблиці додано атрибут `amount` для позначення кількості одиниць замовлення. Усі таблиці знаходяться у нормальній формі Бойса-Кодда. Схему бази даних, що ілюструє сутності та зв'язки між ними, наведено в додатку А (рис. А.1).

4 ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Інформаційна система розроблена у вигляді трьох веб-додатків. Для роботи з базою даних обрано СКБД PostgreSQL, для Back-End частини фреймворки Laravel та Yii2, для Front-End фреймворки Bootstrap 3 та 4.

Сьогодні існує багато систем керування базами даних (MySQL, Oracle, MariaDB, PostgreSQL та ін). Для реалізації бази даних цієї системи обрано діалект PostgreSQL з наступних причин:

- реалізує реляційну модель даних, яка є зрозумілою для кінцевого користувача;
- має гнучкий механізм управління правами користувачів БД (ролі);
- є можливість створювати схеми для відокремлення даних користувачів;
- підтримує мову plpgsql як розширення стандарту SQL для створення ефективних збережених процедур;
- підтримує курсорні цикли у збережених процедурах, що спрощує програмування;
- постійно підтримується розробниками.

У якості веб-серверу обрано зв'язку Apache та Nginx (в пакеті OpenServer) яка на сьогодні є дуже розповсюдженою та ефективною. Вона знижує навантаження на сервер, тому що Nginx обробляє статичний контент, а Apache-динамічний. Тому статичні сторінки одразу завантажує Nginx, що прискорює роботу застосунку.

Front-End частина розроблена на фреймворку Bootstrap (версії 3 та 4) для забезпечення адаптивної, кросбраузерної та швидкої розробки інтерфейсу користувача. Back-End частина розроблена на фреймворках Laravel та Yii2. Використання PHP-фреймворків дозволяє оптимізувати обробку користувацьких запитів та роботу з базою даних. Фреймворк Yii2 дозволяє швидко розробляти CRUD-додатки для маніпулювання базами даних [7], має наочну та якісно перекладену документацію та гнучку структуру проекту. Laravel має кращий механізм маршрутизації та на відміну від Yii2 частіше оновлюється [2]. Обидва фреймворки реалізують MVC архітектуру.

Для розробки використовувалися IDE PhpStorm та Sublime Text.

5 МОДЕЛЬ ПРОГРАМНОГО ДОДАТКУ

Клієнти, співробітники та керуючий коісцентром користуються трьома окремими додатками. Модель рівня представлення даних побудована у вигляді ієрархії сторінок, що відображає послідовність переходів по сторінках додатку. Ієрархія сторінок для застосунку для клієнта копісцентру наведена на рис. 5.1.

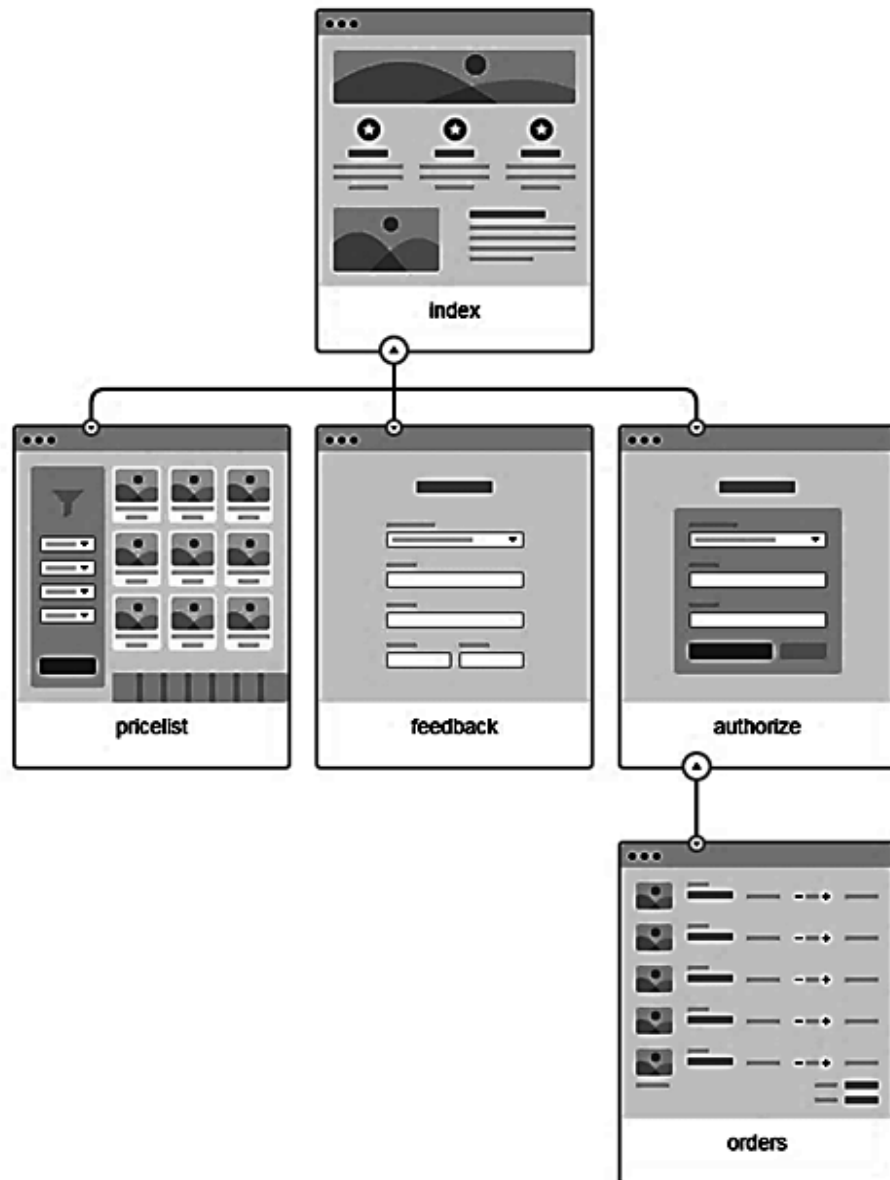


Рисунок 5.1 – Ієрархія сторінок додатку для клієнтів копісцентру

Ієрархія сторінок для додатку для співробітників та керуючого наведена в додатку Б (рис. Б.5, рис. Б.6, рис.Б.7, рис.Б.8).

Моделі рівня прикладного компоненту та управління ресурсами створюються у вигляді діаграм класів для демонстрації загальної структури ієрархії класів системи, їх атрибутів, методів та взаємозв'язків. На рисунку 5.2 наведена діаграма рівня прикладного компоненту для застосунку клієнта копіцентру.

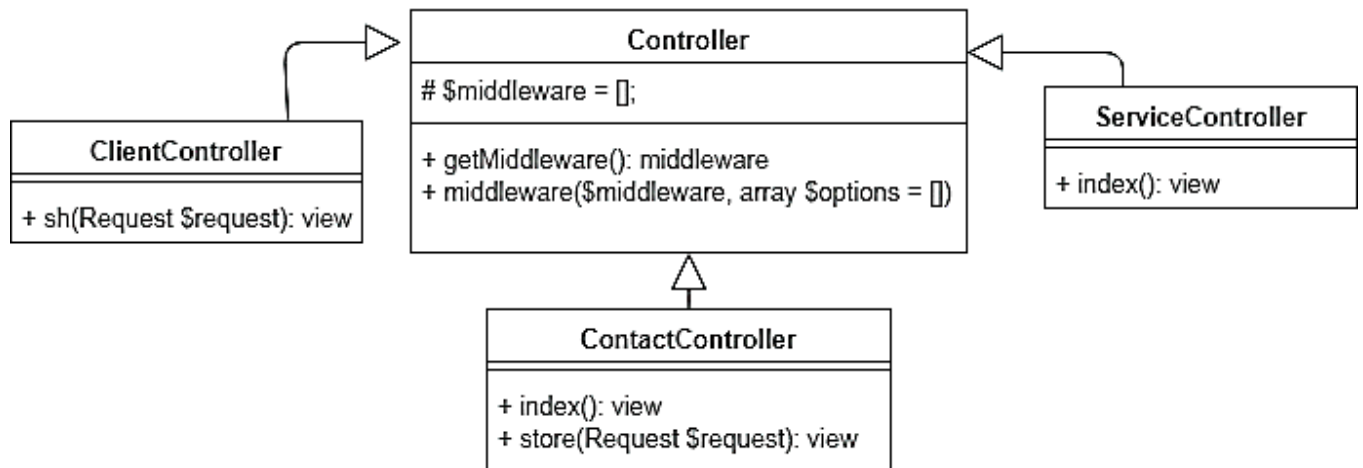


Рисунок 5.2 – Діаграма класів застосунку для клієнта копіцентру – прикладний компонент

Контролери відповідають за відображення видів, авторизацію користувача в системі, обробку даних перед відправкою запитів до сервера БД або після отримання відповіді від нього. Діаграми класів рівня прикладного компоненту для застосунків клієнта та співробітників наведено у додатку Б (рис. Б.3 та рис. Б.4).

Модель рівня управління ресурсами для застосунку клієнта копіцентру наведена на рисунку 5.3. Класи на цьому рівні забезпечують доступ до бази даних та обмін даними із сервером додатків.

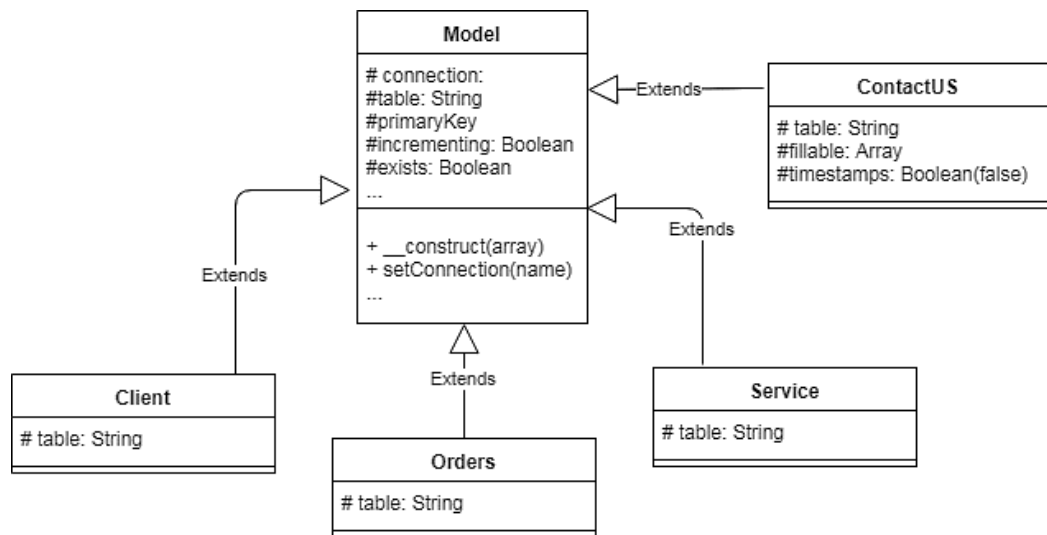


Рисунок 5.3 - Діаграма класів застосунку для клієнта копіцентру – рівень управління ресурсами

Класи Client, Orders, Service та ContactUS розширюють базовий клас Model [2]. Базовий клас Model містить велику кількість стандартних методів та властивостей, тому на діаграмі відображені лише найчастіше використовувані. Застосунок для клієнта копіцентру має найпростішу структуру з усіх трьох та враховує особливості фреймворку Laravel.

Застосунки для співробітника та керуючого копіцентром мають іншу структуру. Всі класи розширюють базовий клас ActiveRecord [3] для роботи з базою даних, а клас User реалізує інтерфейс IdentityInterface [4] для авторизації користувача в системі. Між класами наявні відношення асоціації потужності 1..1 та 1..*. Діаграми класів для програмного застосунку співробітника та керуючого рівня управління ресурсами наведені у додатку Б (рис. Б.1 та рис. Б.2).

6 СТВОРЕННЯ БАЗИ ДАНИХ

В цьому розділі описані основні об'єкти бази даних та права доступу користувачів до неї. Повний код створення бази даних наведено у додатку В.

Домени:

- 1) valid_edu: допустимі значення для рівня освіти співробітників
- 2) family_status: допустимі значення для сімейного статусу співробітників

Таблиці:

- 1) Client – інформація про клієнтів
- 2) C_order – номер та статус замовлення
- 3) Ord_receipt – дані про замовлення (послуга, клієнт..)
- 4) Emp_main – основна інформація про співробітника
- 5) Emp_add – додаткова інформація про співробітника
- 6) Service – перелік послуг копієцентра
- 7) Cons_serv – відповідність витратних матеріалів та послуг
- 8) Cons_mat – витратні матеріали
- 9) Product – товари на продаж
- 10) Goods_receipt – основна інформація про чек на продаж
- 11) Goods_receipt_add – додаткова інформація про чек на продаж
- 12) Cons_invoice – основна інформація про накладну на витратні матеріали
- 13) Cons_invoice_add – дод. інформація про накладну на витратні матеріали
- 14) Goods_invoice - основна інформація про накладну на товари
- 15) Goods_invoice_add - додаткова інформація про накладну на товари
- 16) Goods_sup – постачальники товару
- 17) Buy_request – запити на замовлення товару чи матеріалів
- 18) Contact – таблиця відгуків

Представлення:

- 1) Consn – накладні на витратні матеріали
- 2) Goodsn – накладні на товари
- 3) Sh_ord – чеки замовлення послуг
- 4) Sh_rec – чеки продажу товару
- 5) Show_consmat_remains – залишки витратних матеріалів

- 6) Show_goods_remains – залишки товару
- 7) Client_spent – витрати клієнтів на послуги
- 8) Client_spent_p – витрати клієнтів на товари
- 9) Cons_spent – витрати копієнтра на закупівлю витратних матеріалів
- 10) Prod_spent – витрати копієнтра на товари
- 11) Sell_e_income – сума продажу товарів співробітниками
- 12) Serv_rank – рейтинг послуг за затребуваністю, принесеним доходом та сумарними витратами на витратні матеріали.

Розподіл прав доступу для користувачів бази даних наведено в таблиці 6.1.

Таблиця 6.1. – Права доступу користувачів до бази даних

Таблиця	Користувачі БД		
	client	head	employee1 (2,3,4)
Client	S	S	ISUD
C_order	S	S	ISU
Ord_receipt	S	S	IS
Emp_main		ISUD	S
Emp_add		ISUD	
Service	S	ISU	S
Cons_serv		ISUD	S
Cons_mat		S	IS
Product		S	ISU(product_name)
Goods_receipt		US (e_id)	ISU (c_id)
Goods_receipt_add			IS
Cons_invoice			IS
Cons_invoice_add			IS
Goods_invoice			IS
Goods_invoice_add			IS
Goods_sup		ISUD	S
Buy_request		S	IS
Contact	IS	S	S
Представлення	Користувачі БД		
	client	head	employee1 (2,3,4)
Consn		S	
Goodsn		S	
Sh_ord	S	S	S
Sh_rec		S	

Продовження таблиці 6.1

Представлення	Користувачі БД		
	client	head	employee1 (2,3,4)
Show_consmat_remains		S	S
Show_goods_remains		S	S
client_spent		S	
client_spent_p		S	
cons_spent		S	
prod_spent		S	
sell_e_income		S	
serv_rank		S	
Збережена процедура	Користувачі БД		
	client	head	employee1 (2,3,4)
check_if_c_exists			EXEC
check_if_g_exists			EXEC

Тут літери означають:

- S: перегляд вмісту таблиці;
- I: додавання нової інформації до таблиці;
- U: редагування існуючої інформації (якщо в дужках вказана назва поля таблиці, то дозволяється редагувати лише це поле);
- D: видалення інформації з таблиці;
- EXEC: право на виклик збереженої процедури

Код запитів до бази даних для налаштування доступу до системи наведено у додатку В.

7 ЗАПИТИ ДО БАЗИ ДАНИХ ДЛЯ РОЗВ'ЯЗАННЯ ПОСТАВЛЕНИХ ЗАДАЧ

Керуючий копіцентром:

- Створення нової послуги, постачальника, відповідності між товарами та послугами:

```
INSERT INTO service VALUES ('Печать на чашках',150, В наличии  
чашки разных цветов по 400 мл.);
```

```
INSERT INTO goods_sup VALUES ('ООО ЛидерПринт','Виталий  
Демидович','0955567899');
```

```
INSERT INTO cons_serv VALUES (1,'Ксерокс ч\б А4');
```

- Зміна інформації про послугу, співробітника, постачальника:

```
UPDATE      service      SET      price_per_one=400      WHERE  
service_name='Фотокнига';
```

```
UPDATE emp_main SET mobile_number='0668495317' WHERE e_id=1;
```

```
UPDATE emp_add SET family_status='женат' WHERE e_id=2;
```

```
UPDATE goods_sup SET tel_num='0994067999' WHERE company_name='ООО  
ЛидерПринт';
```

- Прийом співробітника на роботу

```
BEGIN ;
```

```
INSERT INTO emp_main (first_name, last_name,  
mobile_number, username, password) VALUES  
('Ухова','Маргарита','0678416422','employee3','employee3');
```

```
SELECT MAX (e_id) FROM emp_main;
```

```
INSERT INTO emp_add VALUES (3,'Владимировна','замужем',  
'высшее','г.Теплодар, Беляевский район,  
ул.Энергетиков,9,13' '2017-02-10','1996-12-23');
```

```
COMMIT;
```

- Звільнення співробітника

```
DELETE FROM emp_main WHERE e_id=2;
```

Для цієї задачі створено тригер [6], який при звільненні співробітника перепризначає всі його продажі на керуючого, аби не втрачати значення статистичних показників. Код даного тригера (delEmployee) наведено в додатку В.

- Контроль за поставками

Для всіх задач, що входять в контроль за поставками (див. табл. 1.1), використовується команда SELECT * FROM:

```
SELECT * FROM goodsn;  
SELECT * FROM consn;  
SELECT * FROM buy_request;  
SELECT * FROM show_goods_remains;  
SELECT * FROM show_consmat_remains;
```

- Перегляд товарних чеків, замовлень клієнтів, переліку клієнтів копіцентру

```
SELECT * FROM sh_ord;  
SELECT * FROM sh_rec;  
SELECT * FROM client;
```

Клієнт копіцентру:

- Перегляд послуг копіцентру, повної інформації про замовлення

```
SELECT * FROM service;  
  
SELECT * FROM sh_ord where mobile_number = '0957427950' AND  
last_name = 'Руденко';,
```

- Відгук про роботу копіцентру

```
INSERT INTO contact(person,text,date,phone) VALUES ('Андрей',  
'Очень приветливый персонал', '2019-05-04', '0957427955');
```

Співробітник:

- Реєстрація нового клієнта в системі, додавання запитів на замовлення постачальникам

```
INSERT INTO client (last_name, first_name, mobile_number,  
reg_date) VALUES (Самошин', 'Глеб', '0672391762', '2018-10-12');  
  
INSERT INTO buy_request(name, amount) VALUES ('Бумага А4', 1);
```

- Додавання інформації про замовлення

```
BEGIN;  
  
INSERT INTO c_order (ord_status, o_date) VALUES ('Новый', 2019-  
05-21);  
  
SELECT MAX o_id FROM c_order;  
  
INSERT INTO ord_receipt VALUES (7, 'Прошивка дипломов', 25, 1);  
  
COMMIT;
```

При додаванні інформації про замовлення треба перевіряти, чи наявні матеріали для його виконання (тригер T_BLOCK_CONS, додаток В).

- Зміна інформації про клієнта, оновлення статусу замовлення

```
UPDATE client SET mobile_number='0666787878' WHERE c_id=13;
```

```
UPDATE c_order set ord_status ='Готов' where o_id=23;
```

- Видалення інформації про клієнта

```
DELETE FROM client WHERE c_id = 12;
```

При видаленні інформації про клієнта треба не втратити його замовлення та придбання. За допомогою тригеру delClient (додаток В) відбувається перепризначення придбань та замовлень на роздрібного клієнта.

- Перегляд інформації про замовлення для його видачі, списку постачальників, залишків продукції та витратних матеріалів

```
SELECT * FROM sh_ord; SELECT * FROM goods_sup;
```

```
SELECT * FROM show_consmat_remains;
```

```
SELECT * FROM show_goods_remains;
```

- Продаж товару

```
BEGIN ;
```

```
INSERT INTO goods_receipt VALUES (1,1, CURRENT_DATE,1);
```

```
SELECT MAX (rec_id) FROM goods_receipt;
```

```
INSERT INTO goods_receipt_add VALUES (1,1,100,65); COMMIT;
```

Під час продажу товару слід перевіряти, скільки одиниць даного товару є в копіїцентрі, і якщо кількість недостатня, заборонити здійснення продажу. Для цього створено тригер T_BLOCK_GOODS (додаток В).

- Прийом товару або витратного матеріалу

Перед прийомом товару або витратного матеріалу слід перевірити, чи був такий товар вже поставлений раніше. Якщо так, то отримати його ідентифікатор для заповнення накладної, якщо ні, то додати його до переліку товарів та сгенерувати ідентифікатор. Ця перевірка відбувається за допомогою збережених процедур check_if_c_exists та check_if_g_exists (додаток В).

```
BEGIN ;
```

```
INSERT INTO goods_invoice /cons_invoice (company_name, i_date)  
VALUES ('ООО Канцмаркет', CURRENT_DATE);
```

```
SELECT MAX (rec_id) FROM goods_invoice /cons_invoice;
```

```
SELECT check_if_g_exists ('Точилка железная', 1, 100,20) /  
check_if_c_exists('Картридж черный', 1, 100,20); COMMIT;
```

8 ПРОГРАМНА РЕАЛІЗАЦІЯ КЛІЄНТСЬКОГО ДОДАТКУ

8.1 Додаток клієнта копієцентру (веб-сайт)

Моделі застосунку для клієнта наведені у лістингу 8.1.

```
class Client extends Model
{
    protected $table = 'client';
}
class Orders extends Model
{
    protected $table = 'sh_ord';
}
class ContactUS extends Model
{
    public $table = "contact";
    protected $fillable = ["namesurname", "phone" , "message"];
    public $timestamps = false;
}
class Service extends Model
{
    protected $table = 'service';
}
```

Лістинг 8.1 – Моделі застосунку для клієнта

Контролер, що відповідає за відображення замовлень клієнтів наведений у лістингу 8.2.

```
class ClientController extends Controller
{
    public function sh(Request $request)
    {
        $user = Client::where([[ 'mobile_number', '=', $request->input('num') ], [ 'last_name', '=', $request->input('fam') ],
        ]->first();
        if(empty($user))
        {
            session()->flash('notif','1');
            return view('authorize');
        }

        $orders=DB::table('ord_receipt')->join('service','ord_receipt.service_name','=','service.service_name')->join('c_order','ord_receipt.o_id','=','c_order.o_id')
```

```

->select('ord_receipt.*', 'service.price_per_one','c_order.*')
->where('ord_receipt.c_id','=', $user->c_id)->get();
if(empty($orders)){ session()->flash('notif','2');}

return view('orders',['orders'=>$orders],['users'=>$user-
>first_name]);}}

```

Лістинг 8.2 – Контролер для відображень замовлень клієнта

8.2 Додаток співробітника та керуючого копії центру

Модель «Клієнт» для застосунку співробітника наведена у лістингу 8.3.

```

class Client extends \yii\db\ActiveRecord
{
    public static function tableName()
    {
        return 'client';
    }
    public static function getDb()
    {
        return Yii::$app->userDb;
    }
    public function rules()
    {
        return [
            [['last_name', 'first_name', 'mobile_number'], 'required'],
            [['reg_date'], 'safe'],
            [['last_name', 'first_name'], 'string', 'max' => 30],
            [['mobile_number'], 'string', 'max' => 10],
            [['mobile_number'], 'unique'],];
    }
    public function attributeLabels()
    {
        return [
            'c_id' => 'Ид',
            'last_name' => 'Фамилия',
            'first_name' => 'Имя',
            'mobile_number' => 'Моб.ном',
            'reg_date' => 'Дата рег.',];
    }
    public function getOrdReceipts()

```

```
{return $this->hasMany(OrdReceipt::className(), ['c_id' =>
'c_id']);}}
```

Лістинг 8.3 – Модель «Клієнт» для застосунку співробітника

Кожна модель містить методи `tableName()`, що визначає ім'я таблиці, `rules()`, що визначає правила валідації, `attributeLabels()` що відповідає за відображувані назви стовпців таблиці та `get (Назва_зв'язаної_моделі)` якщо наявні зв'язки. Моделі можуть мати й інші методи. Код основних контролерів та моделей наведено у додатку Г.

Види (або сторінки, які бачить користувач) побудовані за принципом шаблонізації. Розмітка виконана мовою HTML. Усі сторінки вбудовуються у головний шаблон (або `layout`), який містить верхнє меню та контейнер для виводу інформації. Приклад частини виду наведений у лістингу 8.4.

```
<div class="container" style="padding-top:5%">
    <?= Breadcrumbs::widget([
        'links' => isset($this->params['breadcrumbs']) ?
$this->params['breadcrumbs'] : [],
    ]) ?>
    <?= Alert::widget() ?>
    <?= $content ?>
</div>
</div>
```

Лістинг 8.4 – Приклад виду для відображення інформації

Змінна `$content` містить контент поточної сторінки, і щоразу її вміст змінюється, але в цілому сторінки мають схожу структуру.

9 ІНСТРУКЦІЯ КОРИСТУВАЧА З ІЛЮСТРАЦІЯМИ

В даному розділі розглядається розв'язання користувацьких задач за допомогою розробленого інтерфейсу.

9.1 Клієнт копіцентру

При переході за посиланням sscv.com завантажується головна сторінка сайту (рис.9.1)

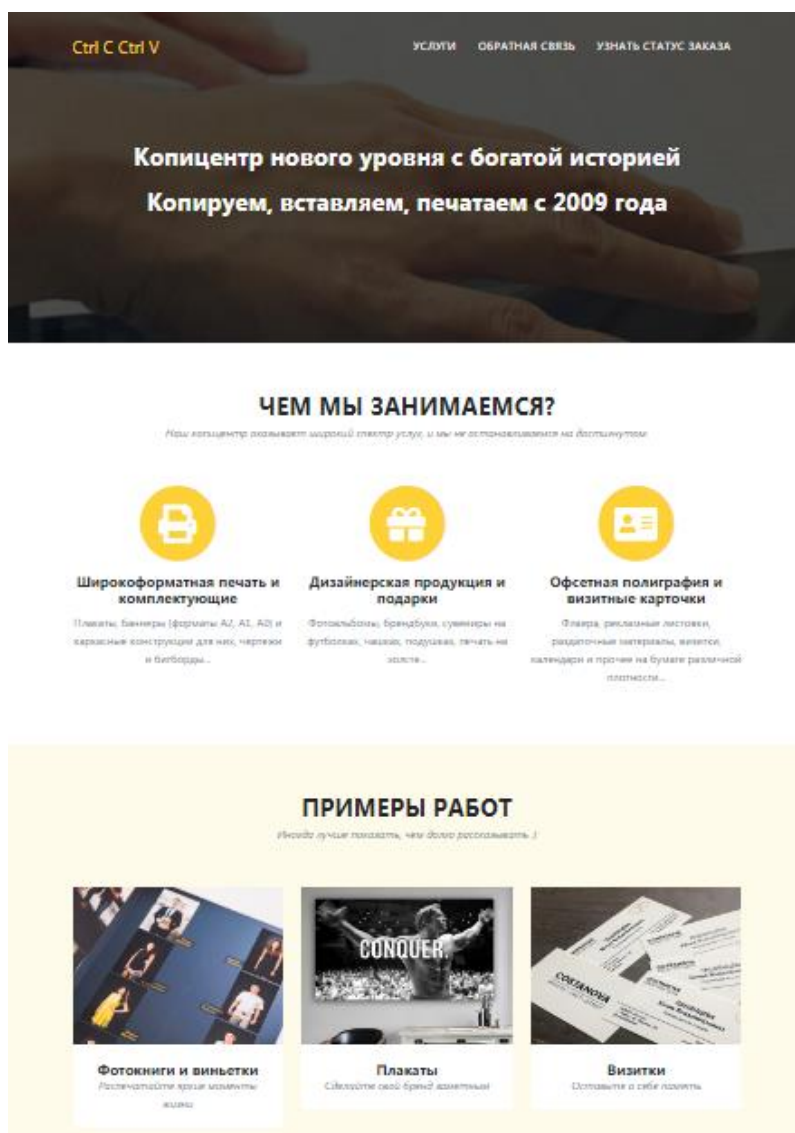


Рисунок 9.1 – Головна сторінка веб-сайту

Для перегляду переліку послуг у верхньому меню треба обрати пункт «Услуги». На рис. 9.2 зображена сторінка з послугами копіцентру.

The image shows two service cards side-by-side. The left card is titled 'Ксерокс цветной' (Color Xerox) with a price of '1.40 ₴' and a note 'Сходство 1:1 с оригиналом' (1:1 similarity to original). The right card is titled 'Фото на холсте A4' (A4 photo on canvas) with a price of '200 ₴' and a detailed description: 'Фото на плотном художественном холсте с любых носителей. Обращаем Ваше внимание, что оператор имеет право отказать в печати, если фото недостаточно качественное.' (Photo on thick artistic canvas from any carriers. We draw your attention that the operator has the right to refuse printing if the photo is of insufficient quality.)

Назва послуги	Ціна	Додаткова інформація
Ксерокс цветной	1.40 ₴	Сходство 1:1 с оригиналом
Фото на холсте A4	200 ₴	Фото на плотном художественном холсте с любых носителей. Обращаем Ваше внимание, что оператор имеет право отказать в печати, если фото недостаточно качественное.

Рисунок 9.2 – Услуги копіцентру

Для перегляду статусу замовлень у верхньому меню треба обрати пункт «Узнать статус заказа». На рис. 9.3 зображена форма, у яку користувач вводить власні дані для отримання доступу до замовлень.

The image shows a form for entering user data. It has two input fields: 'Номер телефона' (Phone number) with the value '0957427950' and 'Фамилия' (Surname) with the value 'Руденко'. Below the phone number field is a note: 'Ввод в формате 0950000000 без пробелов и разделителей' (Input in format 0950000000 without spaces and separators). At the bottom is a yellow button labeled 'Показать' (Show).

Номер телефона

0957427950

Ввод в формате 0950000000 без пробелов и разделителей

Фамилия

Руденко

Показать

Рисунок 9.3 – Форма для ввода данных користувача

Якщо введена інформація коректна, користувач побачить перелік своїх замовлень зі статусами (рис.9.4). Якщо ні, то виведеться повідомлення про помилку.

Здравствуйте, Алиса

Поиск по любому атрибуту заказа

Номер заказа	Название услуги	Количество	Статус	Дата	Стоимость
20	Печать фото 10x15	12	Новый	2019-05-06 00:00:00	50.4
25	Прошивка дипломов	1	Новый	2019-05-06 00:00:00	200
26	Ксерокс ч\б А4	1	Новый	2019-05-06 00:00:00	0.5
48	Печать на чашках	1	Готов	2019-05-07 00:00:00	150
48	Фотокнига	1	Готов	2019-05-07 00:00:00	300
8	Фотокнига	1	Отказ	2019-05-06 00:00:00	300
19	Печать фото 10x15	12	В работе	2019-05-06 00:00:00	50.4

Рисунок 9.4 – Перелік замовлень користувача

Для того, щоб залишити відгук про роботу копіцентру, у верхньому меню треба обрати пункт «Обратная связь». На рис. 9.5 зображена форма, у яку користувач вводить власні дані та текст відгуку.

Номер телефона

0999573425

Имя

Владислав

Расскажите, что было не так

Порядовали цены на канцтовары, оперативность печати и приветливость персонала. Обязательно приду к вам еще!

Отправить

Рисунок 9.5 – Форма для відправлення відгуку про роботу копіцентру

9.2 Співробітник копіцентру

Перед початком роботи із застосунком співробітник має здійснити вхід в систему. При переході за посиланням sscv-employee.com якщо вхід раніше не виконано завантажується форма для введення даних (рис. 9.6), якщо виконано, то головна сторінка з короткою інструкцією для співробітника (рис 9.7).

Рисунок 9.6 – Форма для авторизації в системі

Инструкция

Всего главное меню содержит **5 разделов**:

Работа с клиентами, где Вы можете управлять клиентами копицентра (а именно просмотр, регистрация нового клиента или изменение информации о нем), сообщить клиенту статус заказа или выдать его, предварительно спросив у клиента номер заказа и фамилию а так же продать клиенту товар.

Работа с заказами, где можно создать новый заказ, выслушав все пожелания клиента и изменить статус заказа в процессе его выполнения, чтобы клиент мог следить за ходом выполнения работы в личном кабинете.

Работа с поставщиками, в котором можно просматривать информацию о поставщиках, с которыми мы работаем, для того чтобы делать у них заказы по поручению начальства. Если Вы видите, что тот или иной товар или расходный материал на исходе, в этом же разделе можно создать запрос на заказ поставщикам, чтобы проинформировать начальство.

Прием товара, в котором создаются накладные на товар или расходные материалы для фиксирования факта получения товара от поставщика.

Наличие товара, где можно увидеть, сколько единиц товаров и расходных материалов есть на данный момент в копицентре. Рекомендуем перед выполнением услуг или продажами просматривать этот раздел. Если же для выполнения услуги или продажи товара будет чего-то не хватать, система выдаст соответствующее сообщение при попытке оформления чека.



Чтобы завершить работу с системой, нажмите кнопку "Выход" в правом верхнем углу. Если вы приступаете к работе, а на этом компьютере вход уже выполнен другим сотрудником, то нужно выполнить выход из системы и войти под своей учетной записью. Вы же не хотите приписать свои активные продажи другому человеку, верно? :)

Рисунок 9.7 – Інструкція для співробітника

Головне меню має 5 розділів, організованих у вигляді випадаючих списків з посиланнями на інші сторінки. У розділі «Робота з клієнтами» можна зареєструвати нового клієнта, видати клієнту замовлення або повідомити статус, здійснити продаж товару та переглянути відгуки. У розділі «Робота з замовленнями» можна створити нове замовлення, оновити статус та переглянути відповідність між витратними матеріалами та послугами, у розділі «Робота з постачальниками» можна переглянути список постачальників та створити запит на замовлення постачальникам. У «Прийом товару» здійснюється створення накладних на прийом товару або витратних матеріалів, а у розділі «Наличие на складе» - перегляд залишків товару та витратних матеріалів. Завершення роботи з системою здійснюється натисканням кнопки «Выход».

Для всіх користувацьких задач, що передбачають перегляд певної інформації з бази даних, інтерфейс має вигляд стовпців з полями для пошуку по кожному з них. Типовий вигляд такого інтерфейсу наведено на рис. 9.8.

Новый клиент

Для изменения информации о клиенте нажмите на кнопку . Редактировать можно имя, фамилию и номер телефона. Если же клиент по ряду причин больше не желает пользоваться нашими услугами, его можно удалить кнопкой .

Показано 1-6 из 6 элементов.













Ид	Фамилия	Имя	Моб.ном	Дата рег.	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
2	Старцев	Валентин	0679018992	2018-03-26 00:00:00	 
3	Самошин	Глеб	0672391762	2018-10-12 00:00:00	 
4	Ястребова	Елизавета	0934946866	2016-01-11 00:00:00	 
5	Богрова	Елена	0935817704	2018-09-12 00:00:00	 
7	Руденко	Алиса	0957427950	2018-06-12 00:00:00	 
1	Клиент	Розница	0000000000	1900-01-01 00:00:00	 

Рисунок 9.8 – Перегляд інформації про клієнтів

На таких сторінках можуть бути наявні кнопки для додавання нової інформації, редагування вже існуючої (має вигляд олівця) та видалення (має вигляд кошику для сміття), а також поля для пошуку. Приклад результату пошуку наведено на рис. 9.9.

Заказы

У клиентов, которые зарегистрированы, нужно спрашивать номер заказа и фамилию, у незарегистрированных (без карты) только номер заказа.

Изменить статус заказа

Показано 1-5 из 5 элементов.

Ид заказа	Статус	Дата	Фамилия	Имя	Услуга	Кол-во	Стоимость
<input type="text"/>	<input type="text"/>	<input type="text"/>	Ястребова	Елизавета	Фотокнига	<input type="text"/>	<input type="text"/>
21	Новый	2019-05-06 00:00:00	Ястребова	Елизавета	Фотокнига	1	300
24	Новый	2019-05-06 00:00:00	Ястребова	Елизавета	Фотокнига	1	300
27	Новый	2019-05-06 00:00:00	Ястребова	Елизавета	Фотокнига	1	300
49	Новый	2019-05-07 00:00:00	Ястребова	Елизавета	Фотокнига	1	300
53	Новый	2019-05-12 00:00:00	Ястребова	Елизавета	Фотокнига	1	300

Рисунок 9.9 – Результаты поиска за прізвищем, ім'ям та послугою

Форми для введення нової інформації (прийом співробітника на роботу, реєстрація нового клієнта, прийом товару ті ін.) або редагування існуючої інформації (зміна статусу замовлення, оновлення клієнтських даних тощо) мають

вигляд полів для введення тексту або випадаючі списки, що містять інформацію з бази даних. Типовий вигляд таких форм наведено на рис 9.10 та 9.11.

Изменить статус заказа: 2

Статус

Готов

Дата

2019-05-06 00:00:00

Сохранить

Рисунок 9.10 – Оновлення статусу замовлення

Новый заказ

Клиент

Самошин Глеб

Услуга

Ксерокс цветной

Количество

12

Сохранить и завершить

Сохранить и ввести еще 1 услугу в заказ

Рисунок 9.11 – Додавання інформації про нове замовлення

При видаленні інформації система виведе повідомлення для підтвердження видалення для запобігання випадкових дій. Вигляд такого повідомлення наведено на рис. 9.12.

Подтвердите действие на странице sscv-employee.com

Вы точно хотите удалить этот элемент?

ОК

Отмена

Рисунок 9.12 – Підтвердження видалення інформації

9.3 Керуючий копіцентром

Інтерфейс для керуючого повністю ідентичний інтерфейсу для співробітників. При переході за посиланням sscv-head.com керуючий здійснює вхід (див. рис. 9.6), але замість інструкції (див рис. 9.7) на головній сторінці застосунку відображені статистичні дані копіцентру (рис. 9.13).

Рейтинг услуг копицентра

Услуга	По востребованности	По доходности	По стоимости расхмат
Фотокнига	1	1	1
Ксерокс ч\б А4	2	4	2
Печать фото 10х15	3	3	3
Печать баннеров	4	2	4

Общие затраты клиентов на услуги

Фамилия	Имя	Сумма
Самошин	Глеб	5535.90
Ястребова	Елизавета	2045.20
Руденко	Алиса	1051.30
Клиент	Розница	1050.00
Богрова	Елена	417.10
Старцев	Валентин	300

Общие затраты клиентов на товары

Фамилия	Имя	Сумма
Клиент	Розница	40
Ястребова	Елизавета	90
Самошин	Глеб	71
Руденко	Алиса	22
Богрова	Елена	55
Старцев	Валентин	274

Рисунок 9.13 – Статистика копiцентру: рейтинг послуг та витрати клiєнтiв

Два статистичних елементи, а саме обсяг продажу по співробітниках за весь час та за місяць представлені у вигляді стовпчикових діаграм для наочності (рис 9.14) за допомогою бібліотеки FusionCharts [5].

Затраты копицентра на расходные материалы

За все время За текущий месяц

234642.15



Затраты копицентра на товары

За все время За текущий месяц

17350

17350



Рисунок 9.14 – Статистика копiцентру: витрати та обсяги продажу товарiв

Ці діаграми будуються використовуючи значення з бази даних, а отже є динамічними. За бажанням можна змінювати їх зовнішній вигляд або набір даних, на основі яких вони будуються.

У верхньому меню 8 елементів: «Сотрудники», «Клиенты», «Услуги», «Поставщики», «Заказы», «Товарные чеки», «Накладные» (випадаючий список з посиланнями «Накладные на товары» та «Накладные на услуги») та «Отзывы». За допомогою меню здійснюється перехід по сторінках додатку та завершення роботи з ним (кнопка «Вихід» у правому верхньому куті). Вигляд частини головного меню наведений на рис. 9.15.

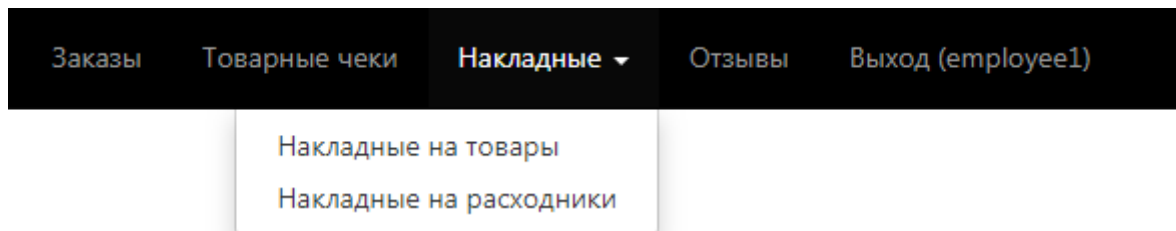


Рисунок 9.15 – Головне меню з випадаючим списком

ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ

В даному курсовому проекті виконано аналіз предметної області, в результаті якого визначено користувачів системи та їх задачі, спроектовано і розроблено базу даних на основі СКБД PostgreSQL яка реалізує реляційну модель даних; розроблено та протестовано Front-End частину застосунку з використанням фреймворку Bootstrap та Back-End частину на основі фреймворків Laravel та Yii2.

Даний курсовий проект є працюючим прототипом, що розв'язує усі поставлені задачі, проте застосунок має перспективи розвитку:

- реалізація замовлення послуг через веб-сайт з можливістю внесення оплати за допомогою білінгових систем;
- впровадження чату між менеджером та клієнтом, де клієнт може ставити запитання щодо послуг та особливих замовлень та отримувати відповідь протягом 1-2 хвилин чи раніше;
- оптимізація інтерфейсу, що вирішує завдання прийому товарів в копіцентр з метою прискорення цього процесу (той, що реалізований на сьогоднішній день, не є оптимальним);
- формування та реалізація більшої кількості статистичних показників роботи копіцентру ;
- розробка та впровадження системи знижок в копіцентрі для певних груп клієнтів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Малахов Є.В. Основи проектування БД: Конспект лекцій. [Електронний ресурс] – Режим доступу: <http://library.opu.ua>.
2. Дронов В. Laravel. Быстрая разработка современных динамических Web-сайтов на PHP, MySQL, HTML и CSS — БХВ-Петербург, 2018. — 768 с. — 978-5-9775-3845-9.
3. Полное руководство (v2): ActiveRecord [Електронний ресурс] – Режим доступу: <https://yiiframework.com.ua/ru/doc/guide/2/db-active-record/>
4. Полное руководство (v2): Аутентификация [Електронний ресурс] – Режим доступу: <https://yiiframework.com.ua/ru/doc/guide/2/security-authentication/>
5. FusionCharts: JavaScript charts for web & mobile [Електронний ресурс] – Режим доступу: <https://www.fusioncharts.com>
6. PostgreSQL : Документация: 9.6: CREATE TRIGGER [Електронний ресурс] – Режим доступу: <https://postgrespro.ru/docs/postgresql/9.6/sql-createtrigger>
7. Mark Safronov, Jeffrey Winesett: Web Application Development with Yii 2 and PHP — PAKKT Publishing, 2015. — 377 с.

Додаток А

СХЕМА БАЗИ ДАНИХ



Рисунок А.1 – Схема бази даних

Додаток Б ДІАГРАМИ КЛАСІВ

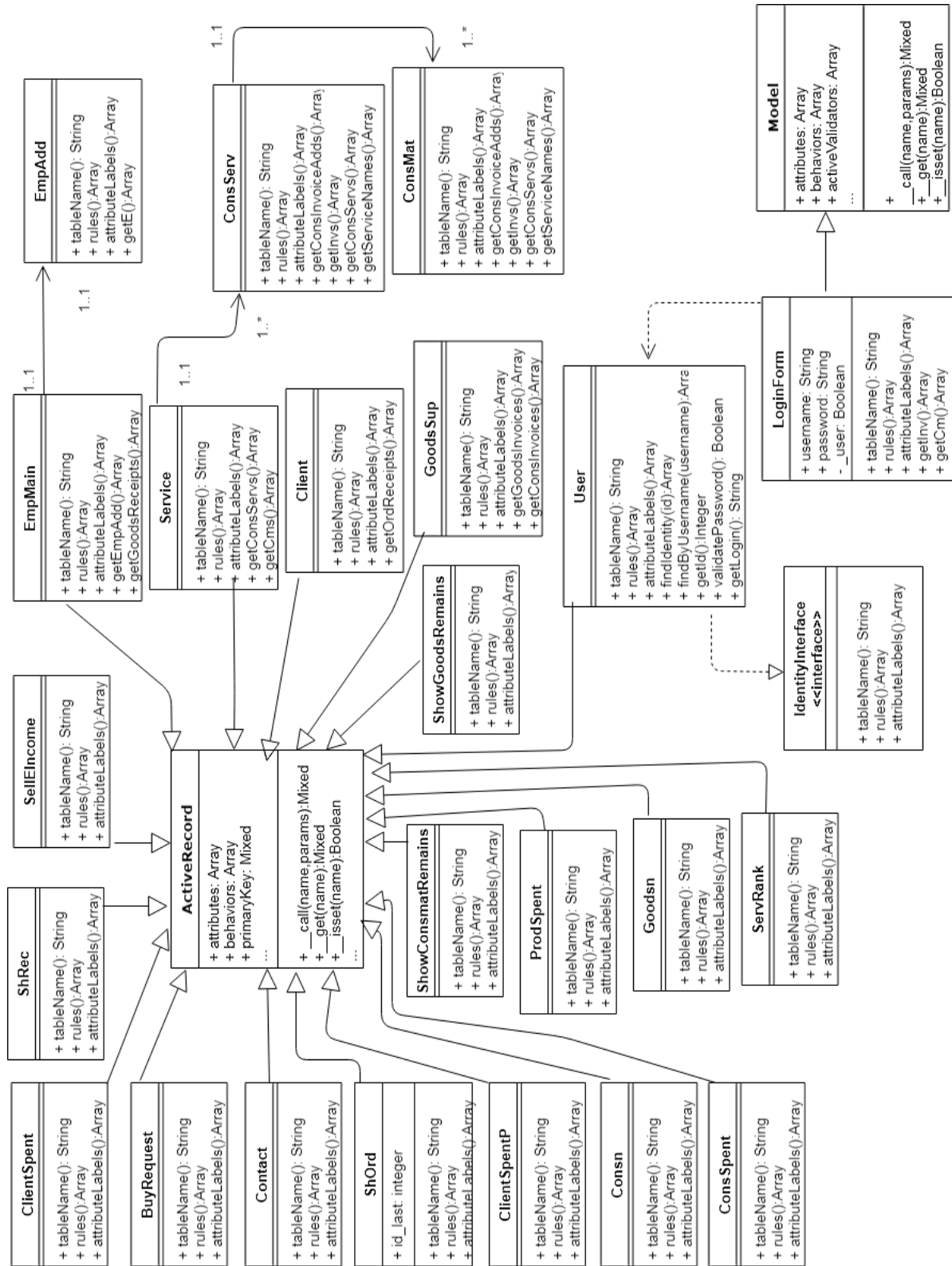


Рисунок Б.1 – Діаграма класів для застосунку керуючого (моделі)

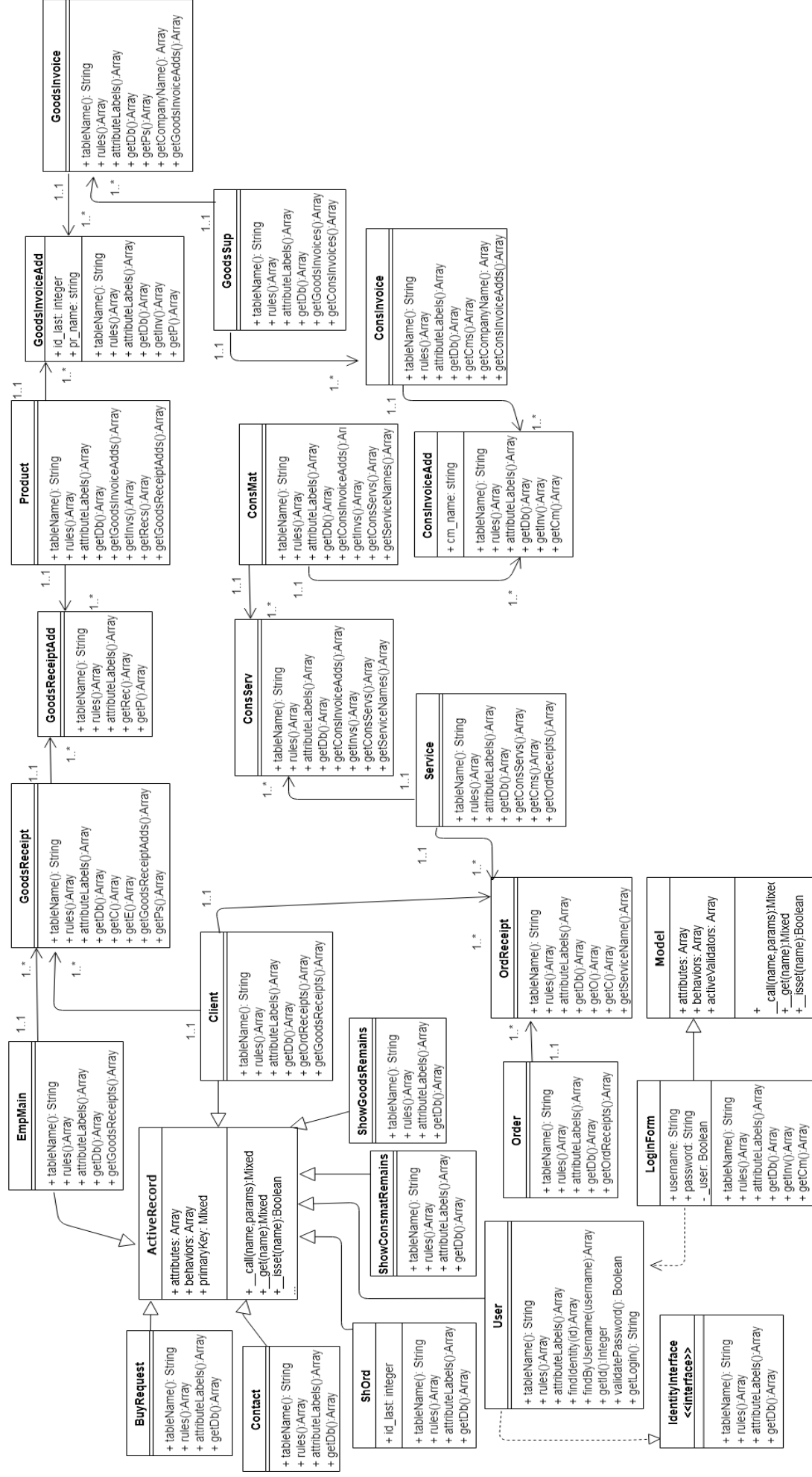


Рисунок Б.2 – Діаграма класів для застосунку співробітника (моделі)

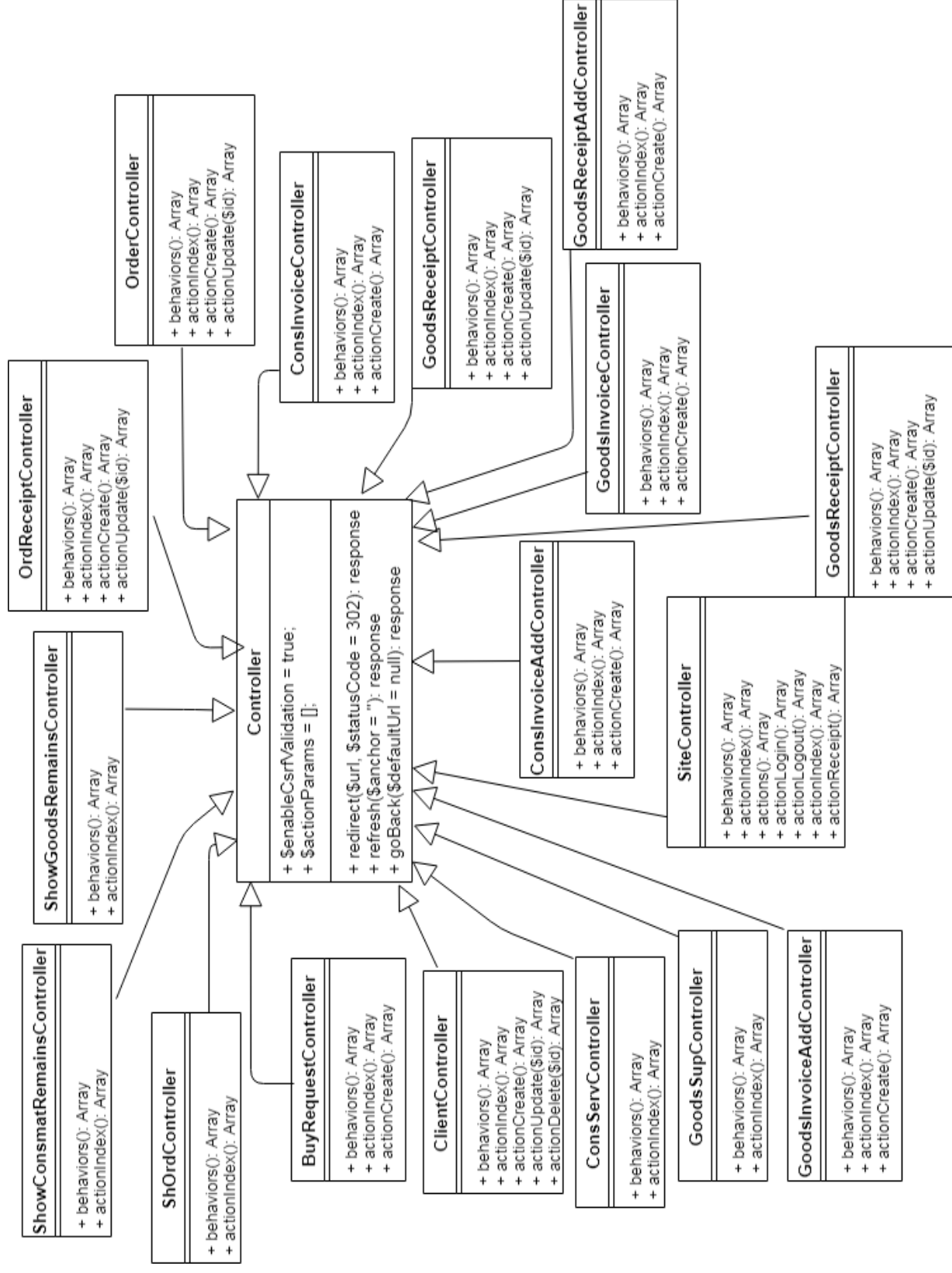


Рисунок Б.3 – Діаграма класів для застосунку співробітника (контролери)

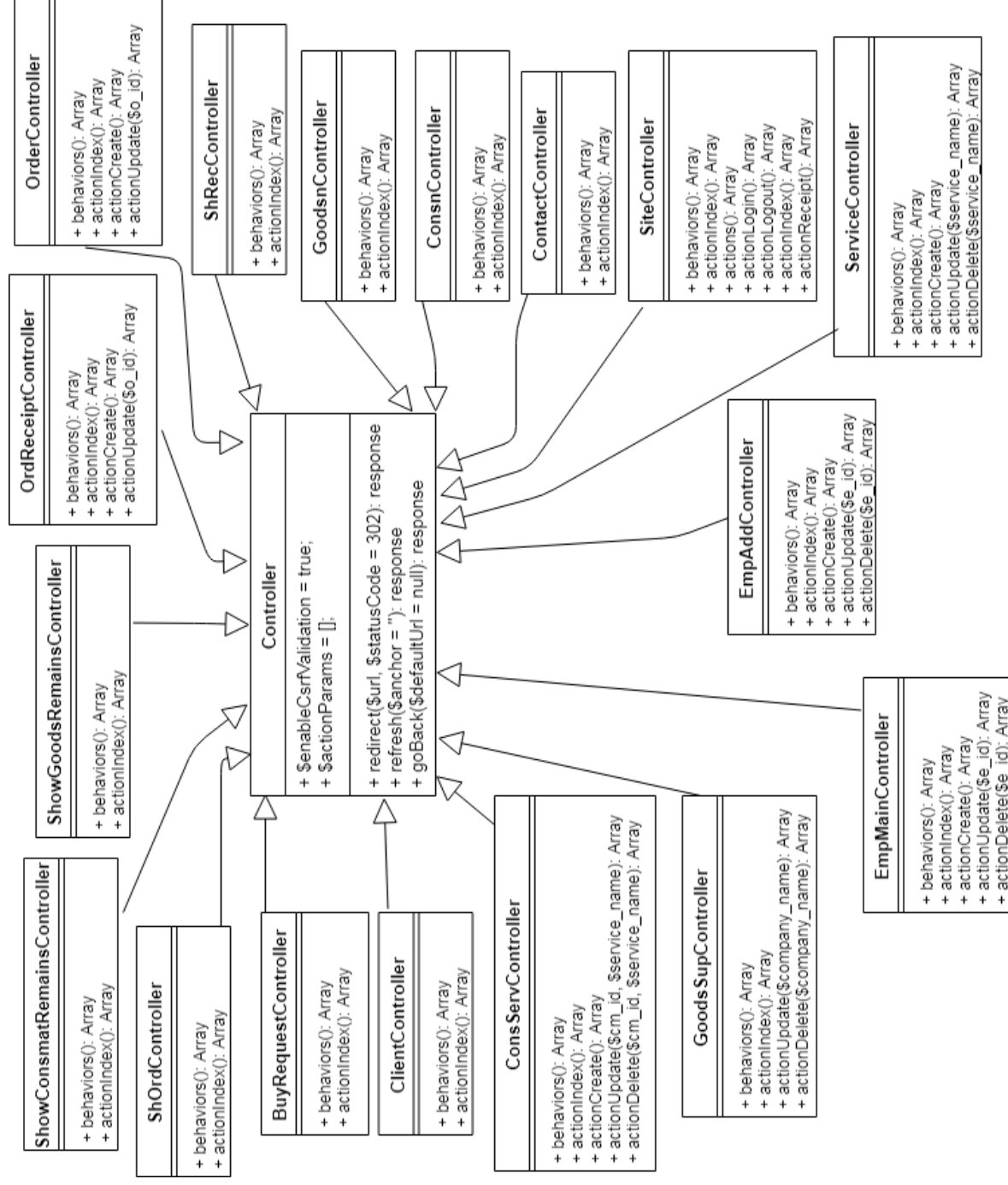


Рисунок Б.4 – Діаграма класів для застосунку керуючого(контролери)

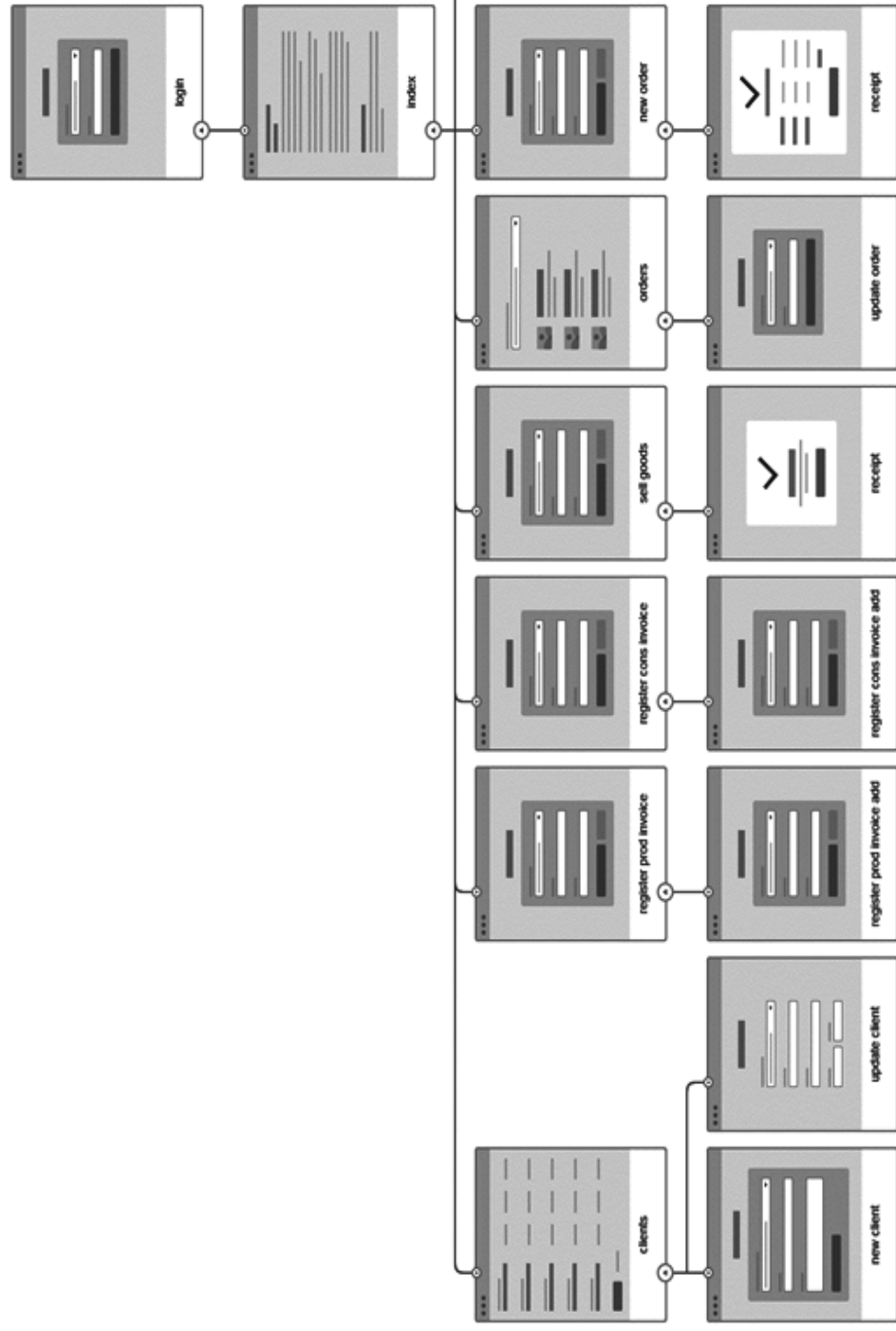


Рисунок Б.5 – Ієрархія сторінок застосунку для співробітника (частина 1)

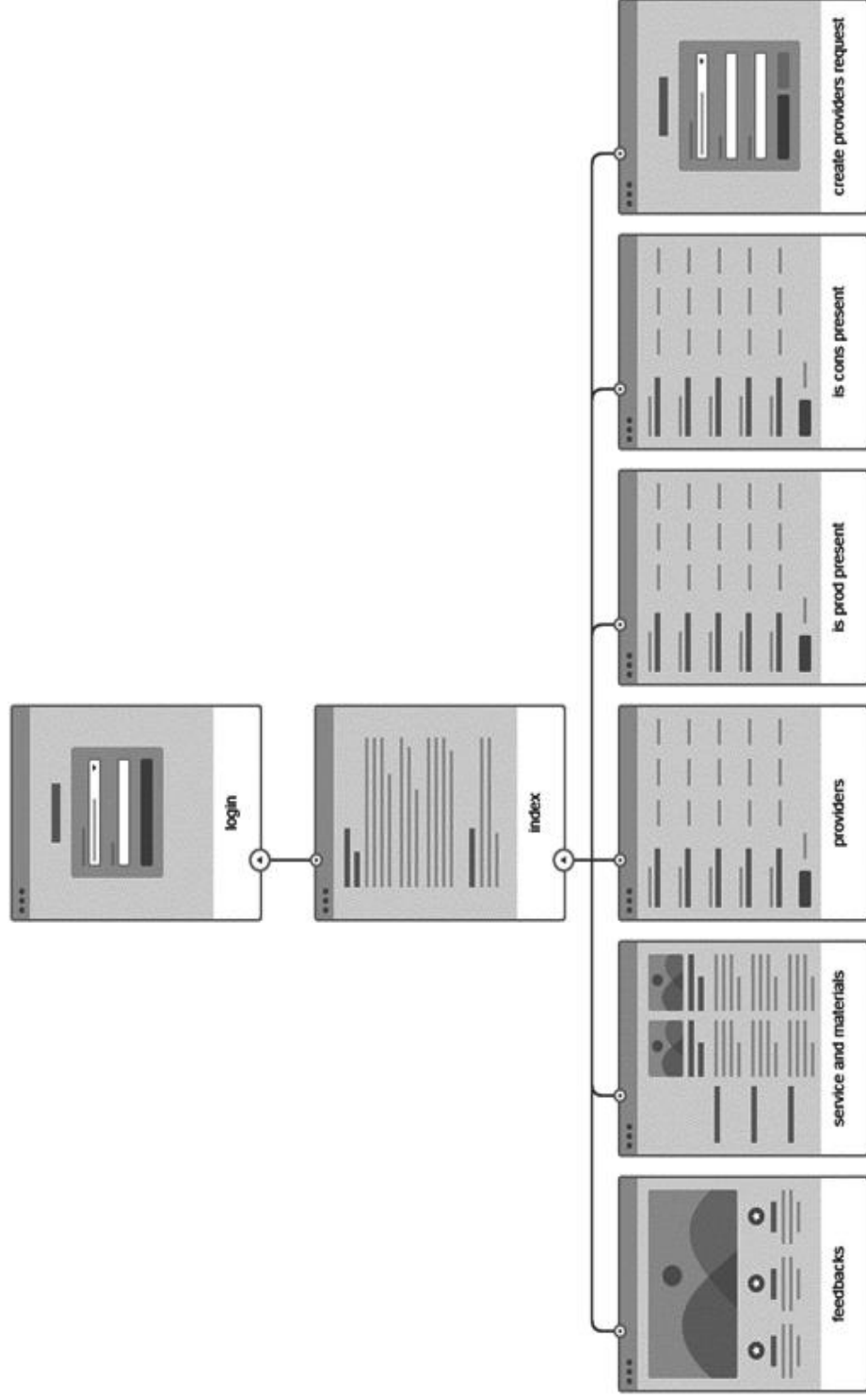


Рисунок Б.6 – Ієрархія сторінок застосунку для співробітника (частина 2)

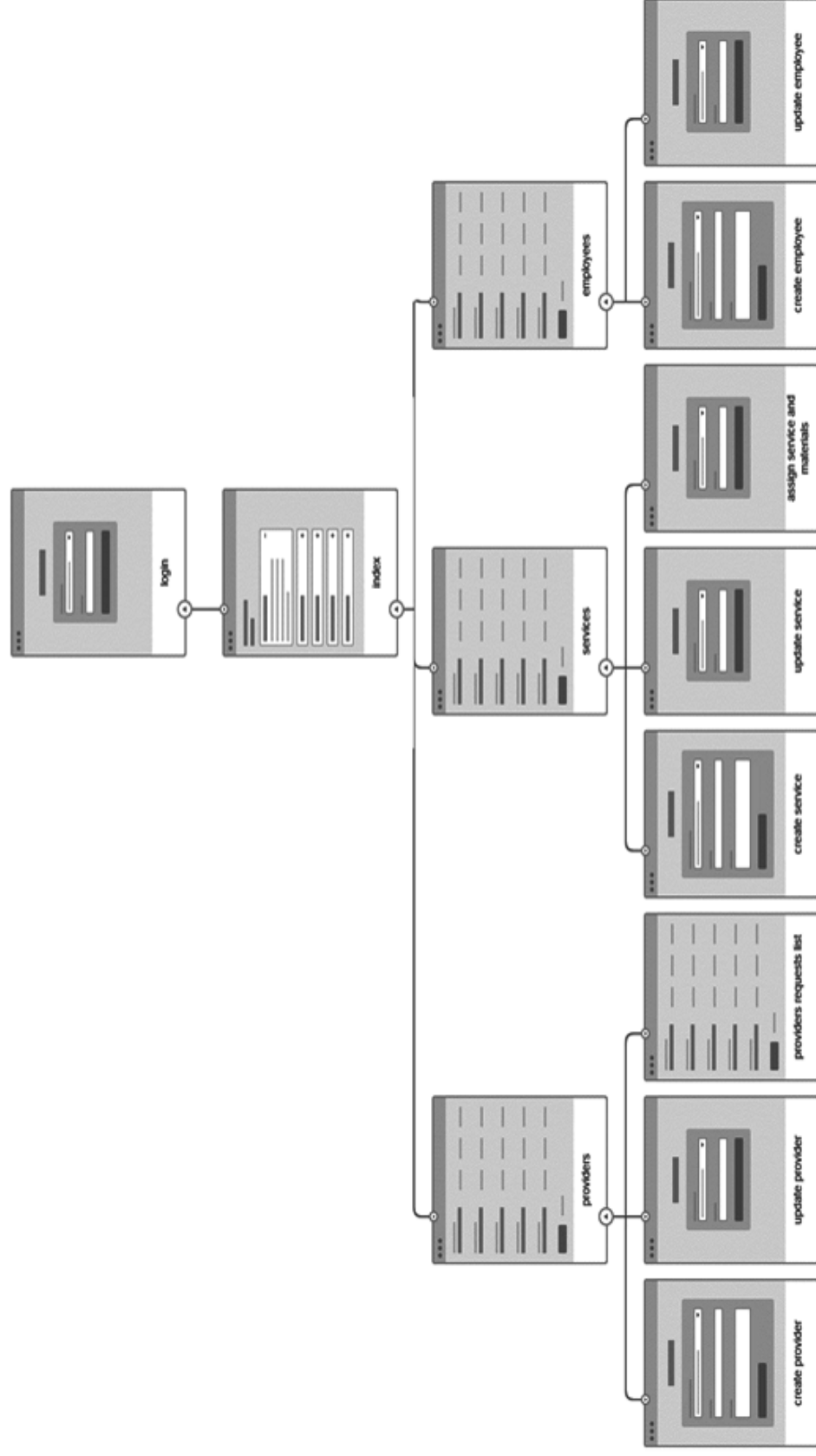


Рисунок Б.7 – Ієрархія сторінок застосунку для керуючого (частина

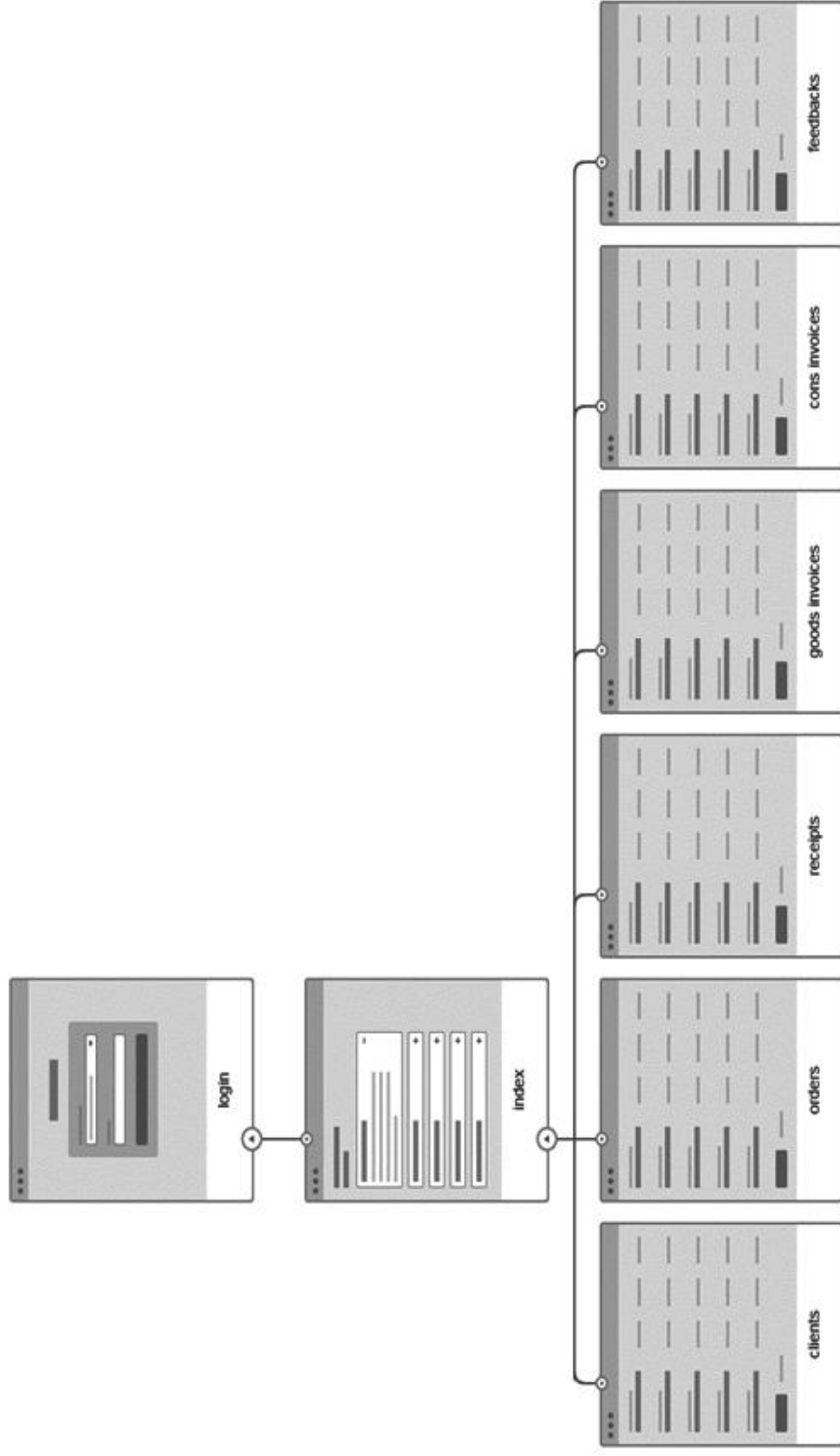


Рисунок Б.8 – Ієрархія сторінок застосунку для керуючого(частина 2)

Додаток В

ЗАПИТИ ДЛЯ СТВОРЕННЯ БАЗИ ДАНИХ

Домени

```
CREATE DOMAIN education character(9)
[CONSTRAINT valid_edu]
CHECK (VALUE IN('средн', 'ср-спец', 'неок-высш', 'высшее'));

CREATE DOMAIN family_status character(10)
[CONSTRAINT valid_fst]
CHECK (VALUE IN('женат', 'не женат', 'замужем', 'не замужем'));
```

Таблиці

buy_request

```
CREATE TABLE buy_request
(name character varying NOT NULL,
 amount integer NOT NULL,
 id serial NOT NULL PRIMARY KEY )
```

client

```
CREATE TABLE client
(c_id serial NOT NULL PRIMARY KEY
 last_name character(30) NOT NULL,
 first_name character(30) NOT NULL,
 mobile_number character(10) NOT NULL,
 reg_date timestamp without time zone NOT NULL DEFAULT
 CURRENT_DATE,
 UNIQUE (mobile_number))
```

service

```
CREATE TABLE service
(service_name character varying NOT NULL PRIMARY KEY,
 price_per_one numeric NOT NULL,
 desc character varying )
```

c_order

```
CREATE TABLE c_order
(o_id serial NOT NULL PRIMARY KEY,
 ord_status character varying(25) NOT NULL,
 o_date timestamp without time zone DEFAULT CURRENT_DATE NOT
 NULL)
```

product

```
CREATE TABLE product
(p_id serial NOT NULL DEFAULT PRIMARY KEY
 product_name character(40) NOT NULL,
UNIQUE (product_name))
```

cons_mat

```
CREATE TABLE cons_mat
(cm_id serial NOT NULL PRIMARY KEY,
cm_name character varying NOT NULL ,
UNIQUE (cm_name))
```

goods_sup

```
CREATE TABLE goods_sup
(company_name character varying NOT NULL PRIMARY KEY,
contact_person character varying NOT NULL,
tel_num character(10) NOT NULL,
UNIQUE (tel_num))
```

emp_main

```
CREATE TABLE emp_main
(e_id serial NOT NULL PRIMARY KEY,
last_name character(30) NOT NULL,
mobile_number character(10) NOT NULL,
first_name character(30) NOT NULL,
username character varying NOT NULL,
password character varying NOT NULL,
UNIQUE (mobile_number,username)),
CONSTRAINT check_min_length CHECK (length(password::text) >= 5)
```

emp_add

```
CREATE TABLE emp_add
(e_id integer NOT NULL PRIMARY KEY,
patronymic character(30) NOT NULL,
family_status family_status NOT NULL,
education education NOT NULL,
address character varying NOT NULL,
emp_date date NOT NULL,
birth_date date NOT NULL,
UNIQUE (e_id),
FOREIGN KEY (e_id) REFERENCES emp_main (e_id)
ON UPDATE NO ACTION
ON DELETE CASCADE)
```

cons_serv

```
CREATE TABLE cons_serv
(cm_id integer NOT NULL,
 service_name character varying NOT NULL,
PRIMARY KEY (cm_id, service_name),
FOREIGN KEY (cm_id) REFERENCES cons_mat (cm_id)
    ON UPDATE NO ACTION
    ON DELETE RESTRICT,
FOREIGN KEY (service_name) REFERENCES service (service_name)
    ON UPDATE NO ACTION
    ON DELETE RESTRICT)
```

cons_invoice

```
CREATE TABLE cons_invoice
(inv_id serial NOT NULL PRIMARY KEY
 company_name character varying NOT NULL,
 i_date timestamp without time zone NOT NULL DEFAULT
CURRENT_DATE,
FOREIGN KEY (company_name) REFERENCES goods_sup (company_name)
ON UPDATE NO ACTION
ON DELETE RESTRICT)
```

cons_invoice_add

```
CREATE TABLE cons_invoice_add
( inv_id integer NOT NULL,
  cm_id integer NOT NULL,
  amount integer NOT NULL,
  price numeric NOT NULL,
PRIMARY KEY (inv_id, cm_id),
FOREIGN KEY (cm_id)
    REFERENCES cons_mat (cm_id)
    ON UPDATE NO ACTION
    ON DELETE RESTRICT,
FOREIGN KEY (inv_id)
    REFERENCES cons_invoice (inv_id)
    ON UPDATE NO ACTION ON DELETE CASCADE)
```

goods_invoice

```
CREATE TABLE goods_invoice
(inv_id serial NOT NULL PRIMARY KEY,
 company_name character varying NOT NULL,
 i_date date NOT NULL DEFAULT CURRENT_DATE,
FOREIGN KEY (company_name) REFERENCES goods_sup
(company_name)
    ON UPDATE NO ACTION ON DELETE RESTRICT)
```

goods_invoice_add

```
CREATE TABLE goods_invoice_add
(inv_id integer NOT NULL,
 p_id integer NOT NULL,
 amount integer NOT NULL,
 purchase_price numeric NOT NULL,
 PRIMARY KEY (inv_id, p_id),
 FOREIGN KEY (inv_id) REFERENCES goods_invoice (inv_id)
     ON UPDATE NO ACTION
     ON DELETE RESTRICT,
 FOREIGN KEY (p_id) REFERENCES product (p_id)
     ON UPDATE CASCADE
     ON DELETE CASCADE)
```

goods_receipt

```
CREATE TABLE goods_receipt
(rec_id serial NOT NULL PRIMARY KEY,
 e_id integer NOT NULL,
 r_date timestamp without time zone NOT NULL DEFAULT
CURRENT_DATE,
 c_id integer NOT NULL,
 FOREIGN KEY (c_id) REFERENCES client (c_id)
     ON UPDATE NO ACTION
     ON DELETE NO ACTION,
 FOREIGN KEY (e_id) REFERENCES emp_main (e_id)
     ON UPDATE NO ACTION
     ON DELETE RESTRICT)
```

goods_receipt_add

```
CREATE TABLE goods_receipt_add
( rec_id integer NOT NULL,
  p_id integer NOT NULL,
  amount integer NOT NULL,
  price numeric NOT NULL,
  PRIMARY KEY (rec_id, p_id),
  FOREIGN KEY (p_id) REFERENCES product (p_id)
      ON UPDATE NO ACTION
      ON DELETE RESTRICT,
  FOREIGN KEY (rec_id) REFERENCES goods_receipt (rec_id)
      ON UPDATE NO ACTION ON DELETE NO ACTION)
```

ord_receipt

```
CREATE TABLE public.ord_receipt
(c_id integer NOT NULL,
 service_name character varying NOT NULL,
```



```

o_id integer NOT NULL,
amount integer NOT NULL,
PRIMARY KEY (c_id, service_name, o_id),
FOREIGN KEY (c_id) REFERENCES client (c_id)
    ON UPDATE NO ACTION
    ON DELETE RESTRICT,
FOREIGN KEY (o_id) REFERENCES c_order (o_id)
    ON UPDATE NO ACTION
    ON DELETE RESTRICT,
    FOREIGN KEY (service_name) REFERENCES service
(service_name)
    ON UPDATE NO ACTION
    ON DELETE RESTRICT)

```

Представления

show_consmat_remains

```

CREATE VIEW show_consmat_remains (consmat_id, remains) as
with begin_amount as(
SELECT cons_serv.cm_id, sum(cons_invoice_add.amount) as bought
FROM   cons_serv
    inner join cons_invoice_add on
cons_serv.cm_id=cons_invoice_add.cm_id
group by cons_serv.cm_id),
after_amount as
(SELECT cons_serv.cm_id, sum(ord_receipt.amount) as spent
FROM   cons_serv
    inner join ord_receipt on
cons_serv.service_name=ord_receipt.service_name
    inner join cons_invoice_add on
cons_serv.cm_id=cons_invoice_add.cm_id
group by cons_serv.cm_id)
select begin_amount.cm_id, (bought-spent) as остаток from
begin_amount
    inner join after_amount using(cm_id)

```

show_goods_remains

```

CREATE VIEW show_goods_remains AS
WITH begin_amount AS (
    SELECT goods_invoice_add.p_id,
        sum(goods_invoice_add.amount) AS bought
    FROM goods_invoice_add
    GROUP BY goods_invoice_add.p_id
), after_amount AS (
    SELECT goods_receipt_add.p_id,
        sum(goods_receipt_add.amount) AS spent

```

```

        FROM goods_receipt_add
        GROUP BY goods_receipt_add.p_id)
SELECT product.p_id as id, product.product_name AS good_name,
       begin_amount.bought - after_amount.spent AS remains
FROM begin_amount
     JOIN after_amount USING (p_id)
     JOIN product USING (p_id);

```

Consn

```

CREATE OR REPLACE VIEW consn AS
SELECT cons_invoice.inv_id,
       cons_invoice.company_name,
       cons_invoice.i_date, cm_id, amount, price
FROM cons_invoice
     INNER JOIN cons_invoice_add add USING (inv_id);

```

Goodsn

```

CREATE OR REPLACE VIEW goodsn AS
SELECT goods_invoice.inv_id,
       goods_invoice.company_name,
       goods_invoice.i_date, p_id, amount, purchase_price
FROM goods_invoice
     INNER JOIN goods_invoice_add add USING (inv_id);

```

Sh_ord

```

CREATE OR REPLACE VIEW sh_ord AS
SELECT c_order.o_id,
       c_order.ord_status,
       c_order.o_date,
       client.last_name,
       client.first_name,
       ord_receipt.service_name,
       ord_receipt.amount,
       service.price_per_one * ord_receipt.amount AS "Стоимость"
FROM c_order
     INNER JOIN ord_receipt USING (o_id)
     INNER JOIN client USING (c_id)
     INNER JOIN service USING (service_name);

```

Sh_rec

```

CREATE OR REPLACE VIEW sh_rec AS
SELECT goods_receipt.rec_id,
       goods_receipt.e_id,
       goods_receipt.r_date,
       goods_receipt_add.p_id,

```

```

        goods_receipt_add.amount,
        goods_receipt_add.price
FROM goods_receipt
INNER JOIN goods_receipt_add USING (rec_id);

```

client_spent

```

CREATE OR REPLACE VIEW client_spent AS
SELECT ord_receipt.c_id,
       client.last_name,
       client.first_name,
       sum(ord_receipt.amount * service.price_per_one) AS ord_sum
FROM service, client, c_order
INNER JOIN ord_receipt ON c_order.o_id = ord_receipt.o_id
WHERE service.service_name = ord_receipt.service_name AND
client.c_id = ord_receipt.c_id
GROUP BY ord_receipt.c_id, client.last_name,
client.first_name
ORDER BY (sum(ord_receipt.amount * service.price_per_one))
DESC;

```

client_spent_p

```

CREATE OR REPLACE VIEW client_spent_p AS
SELECT goods_receipt.c_id,
       client.last_name,
       client.first_name,
       sum(goods_receipt_add.amount * goods_receipt_add.price) AS
g_sum
FROM goods_receipt_add
INNER JOIN goods_receipt USING (rec_id)
INNER JOIN client USING (c_id)
GROUP BY goods_receipt.c_id, client.last_name,
client.first_name;

```

cons_spent

```

CREATE OR REPLACE VIEW cons_spent AS
WITH all_time_spent AS (
        SELECT sum(cons_invoice_add.amount *
cons_invoice_add.price) AS sall
        FROM cons_invoice_add
    ), curr_month_spent AS (
        SELECT sum(cons_invoice_add.amount *
cons_invoice_add.price) AS s
        FROM cons_invoice_add
        INNER JOIN cons_invoice USING (inv_id)
        WHERE date_part('month', cons_invoice.i_date) =
date_part('month', CURRENT_DATE) AND date_part('year',
cons_invoice.i_date) = date_part('year', CURRENT_DATE))

```

```
SELECT all_time_spent.sall,curr_month_spent.s FROM
all_time_spent,curr_month_spent;
```

prod_spent

```
CREATE OR REPLACE VIEW prod_spent AS
WITH all_time_spent AS (
    SELECT sum(goods_invoice_add.amount *
goods_invoice_add.purchase_price) AS sall
    FROM goods_invoice_add
), curr_month_spent AS (
    SELECT sum(goods_invoice_add.amount *
goods_invoice_add.purchase_price) AS s
    FROM goods_invoice_add
    INNER JOIN goods_invoice USING (inv_id)
    WHERE date_part('month', goods_invoice.i_date) =
date_part('month', CURRENT_DATE) AND date_part('year',
goods_invoice.i_date) = date_part('year', CURRENT_DATE))
SELECT all_time_spent.sall,curr_month_spent.s FROM
all_time_spent,curr_month_spent;
```

sell_e_income

```
CREATE OR REPLACE VIEW sell_e_income AS
WITH emp_main_info AS (
    SELECT emp_main.e_id,emp_main.last_name,emp_main.first_name,
emp_add.patronymic FROM emp_main
    INNER JOIN emp_add USING (e_id)), emp_income AS (
    SELECT goods_receipt.e_id,sum(goods_receipt_add.price -
goods_receipt_add.amount) AS sumall FROM goods_receipt
    INNER JOIN goods_receipt_add USING (rec_id)GROUP BY
goods_receipt.e_id), emp_m_income AS (
    SELECT goods_receipt.e_id,
    sum(goods_receipt_add.price -
goods_receipt_add.amount) AS s
    FROM goods_receipt
    JOIN goods_receipt_add USING (rec_id)
    WHERE date_part('month', goods_receipt.r_date) =
date_part('month', CURRENT_DATE) AND date_part('year',
goods_receipt.r_date) = date_part('year', CURRENT_DATE)
    GROUP BY goods_receipt.e_id)
SELECT emp_main_info.e_id,
    emp_main_info.last_name,
    emp_main_info.first_name,
    emp_main_info.patronymic,
    emp_income.sumall,
    emp_m_income.s
FROM emp_main_info
    INNER JOIN emp_income USING (e_id)
```

```

        INNER JOIN emp_m_income USING (e_id);

serv_rank
CREATE OR REPLACE VIEW serv_rank AS
    SELECT s.service_name,
           rank() OVER (ORDER BY (count(o.service_name)) DESC) AS
vostreb_rank,
           rank() OVER (ORDER BY (sum(s.price_per_one - cia.price))
DESC) AS dohod_rank,
           rank() OVER (ORDER BY (sum(cia.price)) DESC) AS
cons_mat_rank
    FROM service s
        INNER JOIN ord_receipt o USING (service_name)
        INNER JOIN cons_serv cs USING (service_name)
        INNER JOIN cons_invoice_add cia USING (cm_id)
    GROUP BY s.service_name;

```

Тригери

```

--тригер для перевірки кількості певного товару
T_BLOCK_GOODS
--тригерна функція
CREATE OR REPLACE FUNCTION check_goods_amount() RETURNS TRIGGER
AS $$
DECLARE
    ostatok show_goods_remains.remains%TYPE;
    tovar_name varchar;
BEGIN
    SELECT product_name INTO tovar_name FROM product where
p_id=NEW.p_id;
    SELECT remains INTO ostatok FROM show_goods_remains where
good_name=tovar_name;
    IF (NEW.amount>ostatok)
        --якщо введена кількість товару є більшою за наявну, продаж
здійснено не буде
        THEN RAISE EXCEPTION
            'На складі недостаточное количество "%". Всего доступно %,
можно предложить клиенту приобрести данное
количество.', tovar_name, ostatok;
        --буде виведена актуальна кількість даного товару в копівентрі
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE PLPGSQL;
--подія тригера
CREATE TRIGGER T_BLOCK_GOODS BEFORE INSERT ON goods_receipt_add
FOR EACH ROW

```

```
EXECUTE PROCEDURE check_goods_amount();
```

T_BLOCK_CONS

```
--цей тригер схожий на попередній, але спрацьовує при
недостатній кількості витратних матеріалів для виконання послуги
--тригерна функція
CREATE OR REPLACE FUNCTION check_consmat_amount() RETURNS
TRIGGER
AS $$
DECLARE
    ostatok show_goods_remains.remains%TYPE;
    nazvanie cons_mat.cm_name%TYPE;
    cCons cursor is --визначення курсору
        select cm_id from cons_serv
        where service_name=new.service_name;
begin
    for C_rec in cCons loop -курсорний цикл
        select remains into ostatok from show_consmat_remains where
consmat_id=C_rec.cm_id;
        select cm_name into nazvanie from cons_mat where
cm_id=C_rec.cm_id;
        if (ostatok<new.amount)then
            raise exception
                'Недостаточно расходных материалов для выполнения
этой услуги. Расходного материала % осталось %, можно предложить
клиенту выполнить это количество.',nazvanie,ostatok;
        end if;
    end loop;
    return new;
end;
$$ LANGUAGE PLPGSQL;

--тригерна подія
CREATE TRIGGER T_BLOCK_CONS BEFORE INSERT ON ord_receipt FOR
EACH ROW
EXECUTE PROCEDURE check_consmat_amount();
```

bef_empdel

```
--цей тригер перепризначає продаж співробітника на керуючого
копціентром при звільненні
--тригерна функція
create or replace function bef_empdel()
returns trigger
as $$
begin
    if old.e_id=1 then --запобігання видалення керуючого
```

```

raise exception 'Нельзя удалить начальника!';
end if;
update goods_receipt set e_id=1 where e_id=old.e_id;
--перепризначення чеків
return old;
end;
$$ language 'plpgsql'
--тригерна подія
create trigger delEmployee
before delete on emp_main
for each row execute procedure bef_empdel()

```

bef_clidel

--цей тригер перепризначає замовлення та покупки клієнта що видаляється на роздрібного клієнта

```

--тригерна функція
create or replace function bef_clidel()
returns trigger
as $$
begin
--запобігання видалення роздрібного клієнта
if old.c_id=1
then raise exception 'Нельзя удалить розничного клиента!';
end if;
--перепризначення замовлень
update ord_receipt set c_id=1 where c_id=old.c_id;
--перепризначення покупок
update goods_receipt set c_id=1 where c_id=old.c_id;
return old;
end;
$$ language 'plpgsql'

```

```

--тригерна подія
create trigger delClient
before delete on client
for each row execute procedure bef_clidel()

```

Збережені процедури

check_if_c_exists

--перевіряє, чи був такий витратний матеріал вже поставлений, якщо так, то його ідентифікатор запам'ятовується для реєстрації накладної, якщо ні, то він додається до списку та ідентифікатор генерується

```

CREATE OR REPLACE FUNCTION check_if_c_exists(
    cmm_name character varying,

```

```

        i_id integer,
        amount integer,
        price numeric)
    RETURNS integer
AS $$
DECLARE ex_cm INTEGER;
ttemp integer;
tt integer;
BEGIN
    select cons_mat.cm_id into tt from cons_mat where
cm_name=cmm_name;
    if (found=false) then -перевірка, чи був матеріал поставлений
        INSERT INTO cons_mat(cm_name) values(cmm_name) returning
cm_id into ttemp;
        INSERT INTO cons_invoice_add
values(i_id,ttemp,amount,price);
    else
        INSERT INTO cons_invoice_add values(i_id,tt,amount,price);
    end if;
RETURN tt;
END;
$$ LANGUAGE 'plpgsql'

```

check_if_g_exists

```

--функція аналогічна попередній, але працює з товарами
CREATE OR REPLACE FUNCTION check_if_g_exists(
    gg_name character varying,
    i_id integer,
    amount integer,
    price numeric)
    RETURNS integer
AS $$
DECLARE ex_gg INTEGER;
ttemp integer;
tt integer;
BEGIN
    select product.p_id into tt from product where
product_name=gg_name;
    if (found=false) then
        INSERT INTO product(product_name) values(gg_name) returning
p_id into ttemp;
        INSERT INTO goods_invoice_add
values(i_id,ttemp,amount,price);
    RETURN ttemp
    else

```



```

        INSERT INTO
goods_invoice_add(inv_id,p_id,amount,purchase_price)
values(i_id,tt,amount,price);
        return tt;
        end if; END;
$$ LANGUAGE 'plpgsql'

```

```

--Управління доступом до елементів бази даних
CREATE ROLE head WITH PASSWORD 'headcopyshop';
CREATE ROLE client WITH PASSWORD 'cccvclients';
CREATE ROLE employee1 WITH PASSWORD 'employee1';
CREATE ROLE employee2 WITH PASSWORD 'employee2';
CREATE ROLE employee3 WITH PASSWORD 'employee3';
CREATE ROLE employee4 WITH PASSWORD 'employee4';
GRANT INSERT, SELECT ON TABLE buy_request TO employee1;
GRANT SELECT ON TABLE buy_request TO head;
GRANT SELECT ON TABLE c_order TO client;
GRANT INSERT, SELECT, UPDATE ON TABLE c_order TO employee1;
GRANT SELECT ON TABLE c_order TO head;
GRANT SELECT ON TABLE client TO client;
GRANT DELETE, UPDATE, SELECT, INSERT ON TABLE client TO
employee1;
GRANT SELECT ON TABLE client TO head;
GRANT INSERT, SELECT ON TABLE cons_invoice TO employee1;
GRANT INSERT, SELECT ON TABLE cons_invoice_add TO employee1;
GRANT INSERT, SELECT ON TABLE cons_mat TO employee1;
GRANT SELECT ON TABLE cons_mat TO head;
GRANT SELECT ON TABLE cons_serv TO employee1;
GRANT INSERT, SELECT, UPDATE, DELETE ON TABLE cons_serv TO
head;
GRANT INSERT, SELECT, UPDATE, DELETE ON TABLE emp_add TO head;
GRANT SELECT ON TABLE emp_main TO employee1;
GRANT INSERT, SELECT, UPDATE, DELETE ON TABLE emp_main TO head;
GRANT INSERT, SELECT ON TABLE goods_invoice TO employee1;
GRANT INSERT, SELECT ON TABLE goods_invoice_add TO employee1;
GRANT SELECT,UPDATE(e_id) ON TABLE goods_receipt TO head;

```

```

GRANT INSERT, SELECT, UPDATE(c_id) ON TABLE goods_receipt TO
employee1;
GRANT INSERT, SELECT ON TABLE goods_receipt_add TO employee1;
GRANT SELECT ON TABLE goods_sup TO employee1;
GRANT INSERT, SELECT, UPDATE, DELETE ON TABLE goods_sup TO
head;
GRANT INSERT, SELECT, UPDATE ON TABLE ord_receipt TO employee1;
GRANT SELECT ON TABLE ord_receipt TO head;
GRANT SELECT ON TABLE ord_receipt TO client;
GRANT INSERT, SELECT, UPDATE(product_name) ON TABLE product TO
employee1;
GRANT SELECT ON TABLE product TO head;
GRANT SELECT ON TABLE service TO client;
GRANT SELECT ON TABLE service TO employee1;
GRANT INSERT, SELECT, UPDATE ON TABLE service TO head;
GRANT SELECT ON TABLE client_spent TO head;
GRANT SELECT ON TABLE client_spent_p TO head;
GRANT SELECT ON TABLE cons_spent TO head;
GRANT SELECT ON TABLE consn TO head;
GRANT SELECT ON TABLE goodsn TO head;
GRANT SELECT ON TABLE prod_spent TO head;
GRANT SELECT ON TABLE sell_e_income TO head;
GRANT SELECT ON TABLE serv_rank TO head;
GRANT SELECT ON TABLE sh_ord TO employee1;
GRANT SELECT ON TABLE sh_ord TO head;
GRANT SELECT ON TABLE sh_ord TO client;
GRANT SELECT ON TABLE sh_rec TO head;
GRANT SELECT ON TABLE show_consmat_remains TO employee1;
GRANT SELECT ON TABLE show_consmat_remains TO head;
GRANT SELECT ON TABLE show_goods_remains TO employee1;
GRANT SELECT ON TABLE show_goods_remains TO head;
GRANT EXECUTE ON FUNCTION check_if_g_exists() TO employee1;
GRANT EXECUTE ON FUNCTION check_if_c_exists() TO employee1;
GRANT employee1 to employee2; (employee3, employee4);

```

Додаток Г

ПРОГРАМНИЙ КОД

//головний контроллер додатку клієнта

```
<?php
namespace App\Http\Controllers;
use Illuminate\Foundation\Bus\DispatchesJobs;
use Illuminate\Routing\Controller as BaseController;
use Illuminate\Foundation\Validation\ValidatesRequests;
use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
class Controller extends BaseController
{use AuthorizesRequests, DispatchesJobs, ValidatesRequests;}
```

//головний контролер додатку співробітника

```
<?php
namespace app\controllers;
use Yii;
use yii\filters\AccessControl;
use yii\web\Controller;
use yii\web\Response;
use yii\filters\VerbFilter;
use app\models\LoginForm;
use app\models\ContactForm;
use app\models\ClientSpent;
class SiteController extends Controller
{public function behaviors()
{
    return [
        'access' => [
            'class' => AccessControl::className(),
            'only' => ['logout'],
            'rules' => [
                [
                    'actions' => ['logout'],
```

```

        'allow' => true,
        'roles' => ['@'],[],[],],
    'verbs' => [
        'class' => VerbFilter::className(),
        'actions' => [
            'logout' => ['post'],[],[],];}
public function actions()
{
    return [
        'error' => [
            'class' => 'yii\web\ErrorAction',
        ],
        'captcha' => [
            'class' => 'yii\captcha\CaptchaAction',
            'fixedVerifyCode' => YII_ENV_TEST ? 'testme' :
null,[],];}

    //якщо вхід виконано, то завантажується головна сторінка, якщо
ні - сторінка для входу в систему

    public function actionIndex()
    {
        if (!\Yii::$app->user->isGuest) {
            return $this->render('index');
        }
        else {
            $this->layout = 'notlogged';
            return $this->redirect(['login']);}}

    public function actionLogin()
    {
        if (!\Yii::$app->user->isGuest) {
            return $this->goHome();
        }

        $model = new LoginForm();

        if ($model->load(\Yii::$app->request->post()) && $model->login()) {

```

```

        return $this->render('index', [
            'model' => $model,
        ]);
    } else {
        $this->layout = 'notlogged';
        return $this->render('login', [
            'model' => $model,]);}}
//вихід з системи
    public function actionLogout()
    {
        Yii::$app->user->logout();
        return $this->goHome();
    }
//чек на послуги
    public function actionReceipt()
    {
        return $this->render('receipt', [
            'model' => $model
        ]);}}
// головний контролер для додатку керуючого
<?php
namespace app\controllers;
use Yii;
use yii\filters\AccessControl;
use yii\web\Controller;
use yii\web\Response;
use yii\filters\VerbFilter;
//використання моделей для статистики на головній сторінці
use app\models\LoginForm;
use app\models\ContactForm;
use app\models\ClientSpent;
use app\models\ConsSpent;
use app\models\ProdSpent;
use app\models\SelleIncome;

```

```

use app\models\ClientSpentP;
use app\models\ServRank;

class SiteController extends Controller
{
    public function behaviors()
    {
        return [
            'access' => [
                'class' => AccessControl::className(),
                'only' => ['logout'],
                'rules' => [
                    [
                        'actions' => ['logout'],
                        'allow' => true,
                        'roles' => ['@'],
                    ],
                ],
            ],
            'verbs' => [
                'class' => VerbFilter::className(),
                'actions' => [
                    'logout' => ['post'],
                ],
            ],
        ];
    }

    public function actions()
    {
        return [
            'error' => [
                'class' => 'yii\web\ErrorAction',
            ],
            'captcha' => [
                'class' => 'yii\captcha\CaptchaAction',
                'fixedVerifyCode' => YII_ENV_TEST ? 'testme' : null,
            ],
        ];
    }

    // відображення головної сторінки зі статистикою
    public function actionIndex()
    {
        //отримання даних зі статистичних моделей
        $clientSpent=ClientSpent::find()->all();
        $clientSpentP=ClientSpentP::find()->all();
    }
}

```

```

        $consSpent=ConsSpent::find()->all();
        $prodSpent=ProdSpent::find()->all();
        $sellEIncome=SellEIncome::find()->all();
        $srank=ServRank::find()->all();

        //передача цих даних у вид
        if (!\Yii::$app->user->isGuest) {return $this-
>render('index',array('spends'=>$clientSpent,'conspends'=>$consSpent
,'prodspends'=>$prodSpent,'e incomes'=>$sellEIncome,'spendsp'=>$clien
tSpentP,'rrrs'=>$srank));}

        else {$this->layout = 'notlogged';
            return $this->redirect(['login'])->send();}}

        public function actionClients(){
            return $this->render('clients/index')->send();}

        //вхід у систему з переходом на головну сторінку
        public function actionLogin()
        {$clientSpent=ClientSpent::find()->all();
            $clientSpentP=ClientSpentP::find()->all();
            $consSpent=ConsSpent::find()->all();
            $prodSpent=ProdSpent::find()->all();
            $sellEIncome=SellEIncome::find()->all();
            $srank=ServRank::find()->all();

            if (!\Yii::$app->user->isGuest) {
                return $this->goHome();
            }

            $model = new LoginForm();

            if ($model->load(Yii::$app->request->post()) && $model-
>login()) {
                return $this-
>render('index',array('spends'=>$clientSpent,'conspends'=>$consSpent
,'prodspends'=>$prodSpent,'e incomes'=>$sellEIncome,'spendsp'=>$clien
tSpentP,'rrrs'=>$srank));
            } else {
                $this->layout = 'notlogged';
                return $this->render('login', [
                    'model' => $model,]);}

```