

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ



к.т.н, доц. Михелєв І.Л.

" ____ " _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "бакалавр"

на тему:

“Розроблення підсистеми обліку розрахунків за житлово-комунальні послуги
АІС об'єднання співвласників багатоквартирного будинку”

Виконала: студентка групи 4141

Анайко А. Д.



(підпис)

Керівник роботи:

к.т.н., доц. Гайда А. Ю.

(посада, науковий ступінь вчене звання)



підпис)

м. Миколаїв – 2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій
Спеціальність 122 "Комп'ютерні науки"
Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми

к.т.н, доц. Гайда А.Ю.

"___" _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти "бакалавр"

Студенту Анайко Анастасія Дмитрівна

(прізвище, ім'я, по батькові)

1. Тема роботи: "Розроблення підсистеми обліку розрахунків за житлово-комунальні послуги АІС об'єднання співвласників багатоквартирного будинку"

Керівник роботи к.т.н., доц. Гайда А.Ю.

Затверджені наказом ректора № 295-уч від "27" березня 2025 року.

2. Термін подання роботи: 20 червня 2025 р.

3. Вихідні дані до проєкту (роботи) (ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем):
ДСТУ 3008:2015 – "Інформація та документація. Звіти у сфері науки і техніки";
ДСТУ ISO/IEC 15910:2012 "Інформаційні технології. Документація для програмного забезпечення"; Закон України "Про об'єднання співвласників багатоквартирного будинку"

4. Перелік питань, що належать до розробки (найменування розділів):

- Аналіз предметної області, розробка концепції підсистеми обліку розрахунків та постановка задачі.
- Проектні рішення по підсистемі обліку розрахунків АІС ОСББ.
- Модель розгортання підсистеми обліку розрахунків АІС ОСББ.

5. Перелік презентаційних матеріалів

Задачі операційної діяльності ОСББ; Технологічні системи на балансі ОСББ; Мета та задачі роботи; Типова модель кооперації в житлово-комунальному господарстві; Концепція підсистеми обліку розрахунків за надані послуги АІС ОСББ; Модель прецедентів підсистеми обліку розрахунків АІС ОСББ; Модель прецедентів ведення розрахунків за виконані роботи; Логічна модель даних (фрагмент); Модель кооперації підсистеми обліку розрахунків АІС ОСББ; Інтерфейс програмного забезпечення підсистеми обліку розрахунків; Висновки по роботі

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1		03.02.2025	22.02.2025
2		22.02.2025	21.03.2025
3		21.03.2025	25.04.2025
4		25.04.2025	07.06.2025

7. Дата видачі завдання 03 лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

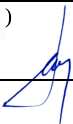
№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	13.02.2025	
2	Постановка задачі	15.02.2025	
3	Розробка концепції	22.02.2025	
4	Розробка проекту ІС	21.03.2025	
5	Реалізація проекту	25.04.2025	
6	Розробка питань з охорони праці	07.06.2025	
7	Розробка графічних матеріалів	12.06.2025	
8	Подання роботи на перевірку на плагіат	10.06.2025	
9	Подання роботи рецензенту	15.06.2025	
10	Подання роботи на захист	20.06.2025	

Студент


(підпис)

Анайко А. Д.
(ПІБ)

Керівник роботи


(підпис)

к.т.н., доц. Гайда А. Ю.
(ПІБ)

Анотація

Робота присвячена розробленню проєкту підсистеми обліку розрахунків за житлово-комунальні послуги автоматизованої інформаційної системи об'єднання співвласників багатоквартирного житлового будинку (ОСББ). В роботі зроблено акцент на підвищенні ефективності документального забезпеченні процесів ведення обліку розрахунків за житлово-комунальні послуги в частині їх надання та виконання.

В роботі проведено аналіз предметної області, розроблено концепцію підсистеми обліку розрахунків автоматизованої інформаційної системи ОСББ, сформульовано постановку задачі, розроблені проєктні рішення по підсистемі.

Робота викладена на ____ сторінках друкованого тексту, містить ____ рисунків, ____ таблиць, ____ посилань на джерела та ____ додатків.

Ключові слова: корпоративна інформаційна система; об'єднання співвласників; документальне забезпечення проєкту; база даних; житлово-комунальні послуги.

Abstract

The work is devoted to the development of a project for the accounting subsystem for settlements for housing and communal services of the automated information system of the association of co-owners of an apartment building (OSBB).

The work focuses on increasing the efficiency of documentary support for the processes of accounting for settlements for housing and communal services in terms of their provision and execution. The work analyzes the subject area, develops the concept of the accounting subsystem for settlements of the automated information system of the OSBB, formulates the problem statement, and develops design solutions for the subsystem.

The work is presented on ____ pages of printed text, contains ____ figures, ____ tables, ____ references to sources and ____ appendices.

Key words: corporate information system; association of co-owners; documentary support for the project; database; housing and communal services.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

AIC – автоматизована інформаційна система

АРМ – автоматизоване робоче місце

БД – база даних

ІС – інформаційна система

ІТ – інформаційні технології

ОСББ – об'єднання співвласників багатоквартирного будинку

ПЗ – програмне забезпечення

СКБД – система керування базами даних

UML – уніфікована мова моделювання

ЗМІСТ

ВСТУП

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ, РОЗРОБКА КОНЦЕПЦІЇ ПІДСИСТЕМИ ОБЛІКУ РОЗРАХУНКІВ ОСББ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Напрямки діяльності ОСББ

1.2. Визначення недоліків наявної автоматизованої інформаційної системи

ОСББ

1.3. Аналіз наявних рішень автоматизації діяльності об'єднань співвласників багатоквартирних будинків

1.4. Розробка концепції підсистеми обліку розрахунків АІС ОСББ

1.5. Функціонально-вартісний аналіз варіантів концепції підсистеми

1.5.1. Функціональний аналіз

1.5.2. Вартісний аналіз

1.6. Висновки до розділу 1

2. ПРОЕКТНІ РІШЕННЯ ПО ПІДСИСТЕМІ ОБЛІКУ РОЗРАХУНКІВ АІС ОСББ

2.1. Загальносистемні рішення по підсистемі обліку розрахунків АІС ОСББ

2.1.1. Функції підсистеми як автоматизованої системи

2.1.2. Контекстна модель підсистеми обліку розрахунків у частині

«сервер баз даних і сервери додатків»

2.2. Рішення з інформаційного забезпечення

2.3. Рішення з програмного забезпечення

2.3.1. Вибір технології організації інформаційного сховища

підсистеми

2.3.2. Вибір мови програмування

2.3.3. Визначення вимог щодо форматів файлів даних підсистеми

обліку розрахунків АІС ОСББ

2.3.4. Фізична модель даних

2.4. Рішення з математичного забезпечення

2.5. Рішення з технічного забезпечення

2.6. Рішення з організаційного забезпечення

2.7. Висновки до розділу 2

3. МОДЕЛЬ РОЗГОРТАННЯ ПІДСИСТЕМИ ОБЛІКУ РОЗРАХУНКІВ АІС ОСББ

3.1. Особливості розгортання підсистеми обліку розрахунків

3.2. Висновки до розділу 3

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

Список використаних джерел

Додаток А. Технічне завдання на розробку проєкту підсистеми обліку розрахунків АІС ОСББ

Додаток Б. Код моделей даних сервера (фрагмент)

ВСТУП

Об'єднання співвласників багатоквартирного будинку (ОСББ) в Україні - це неприбуткова юридична особа, створена власниками квартир у багатоквартирному будинку для спільного утримання та управління спільним майном будинку та прибудинковою територією. ОСББ дає змогу мешканцям самостійно вирішувати питання, пов'язані з управлінням та обслуговуванням будинку, збирати кошти за послуги, що надаються співвласникам будинку, здійснювати закупівлю матеріалів і послуг і розрахунки за виконані для ОСББ роботи, контролювати якість послуг.

ОСББ – це самостійний господарюючий статутний суб'єкт, що може здійснювати господарську та фінансову діяльність із метою забезпечення належних умов життя та утримання будинку та прибудинкової території. ОСББ забезпечують підтримання зелених насаджень, дитячих майданчиків, сигналізації, впорядкування та прибирання прибудинкових територій, експлуатацію кабельних мереж, мереж теплопостачання, водопостачання та каналізаційного відведення, а також задоволення інших потреб громади будинку, забезпеченням ефективного використання комунального майна та збереження навколишнього природного середовища.

ОСББ в своїй операційній діяльності забезпечують вирішення наступних головних задач:

- забезпечують експлуатацію технічних засобів, житлового фонду з проведенням поточних і капітальних ремонтів, реконструкцій і модернізацій кабельних, теплових і водопровідно-каналізаційних мереж;
- здійснюють облік наявних, отриманих та витрачених ресурсів: обладнання, комплектуючих, теплової та електроенергії, холодної води тощо;
- проводять облік стану оплати послуг, що надаються ОСББ по фізичним та юридичним особам;
- ведуть облік і фінансування робіт з підтримки зелених насаджень, впорядкування та прибирання прибудинкових територій, обслуговування лічильників тощо.

Соціальна спрямованість діяльності ОСББ та суттєве з точки зору забезпечення життєвих потреб громади значення результатів діяльності

організації визначає актуальність вирішення задач щодо підвищення ефективності роботи організації шляхом автоматизації його діяльності в частині проведення необхідних розрахунків за надані клієнтам послуги, а також розрахунків організації за отримані послуги та ресурси.

У своїй діяльності персонал ОСББ оперує значними обсягами документів різного характеру, що охоплюють усі сфери їх діяльності, в тому числі: кадровий та бухгалтерський облік, ведення інформації про заплановані до виконання та виконані роботи, складання звітів про надходження, зберігання та витрати комплектуючих та матеріалів, розрахунки для нарахування сплати за послуги тощо.

Документи, що застосовуються при здійсненні діяльності ОСББ повинні зберігатись тривалий час, а це значить, що з кожним роком зростає їх обсяг і кількість. Так, в Україні для населених пунктів з числом мешканців понад 10-20 тисяч, є типовим зведення багатоповерхових багатоквартирних будинків з числом квартир понад 50-100, а кількість видів отриманих послуг на кожную квартиру може сягати понад 10. Здебільшого, ці документи зберігаються в паперовому вигляді, в архівах, а також в електронному вигляді на робочих місцях повноважних співробітників ОСББ. При цьому, із збільшенням кількості документів їх використання стає дедалі складнішим, зменшується оперативність фінансування окремих (зазвичай, незапланованих) робіт та послуг. Все це разом обумовлює потребу впровадження в ОСББ використання спеціалізованих інформаційних систем. Такі системи надають можливість мешканцям будинку визначати терміни та суми сплати за послуги, контролювати рух коштів, планувати додаткові потреби. У свою чергу, керівництво ОСББ отримує можливість активізувати збори коштів на поточні планові та незаплановані потреби, визначати можливості і терміни виконання окремих робіт тощо. Крім того такі системи забезпечують доступ та обробку документів, їх збереження, а в окремих випадках – управління доступом, контроль версій та доступ через мережу Інтернет. Таким чином, робота з документами щодо їх розміщення, зберігання, виконання пошуку, ведення розрахунків тощо, що потребує значних витрат часу співробітників ОСББ, може виконуватись з використанням засобів автоматизованих інформаційних систем (АІС) на основі відповідних технічних та програмних засобів.

Завданням автоматизованих інформаційних систем зі збору та оброблення платежів за житлово-комунальні послуги є інтеграція комунікаційних засобів (комп'ютерів, планшетів, смартфонів) працівників організації і мешканців ОСББ в автоматизовану інформаційну систему та розподіл інформаційних та фінансових потоків між працівниками житлово-комунальної організації, пунктами прийому комунальних платежів тощо.

У більшості міст України створення таких систем мало на меті вирішення наступних наявних проблем:

- збільшити швидкість проходження комунальних платежів та забезпечити їх надходження на рахунки об'єднань співвласників багатоквартирних будинків у повному обсязі;
- знизити собівартість та трудомісткість робіт з формування платіжних документів, їх прийому, обробки, обліку та контролю;
- стимулювати громадян до оплати спожитих послуг та зменшення рівня заборгованості – споживачам забезпечено можливість оплачувати всі комунальні послуги з найменшими зусиллями і в одному пункті прийому платежів;
- підвищити оперативність, повноту та достовірність інформації про комунальні платежі, пільги та житлові субсидії.

При створенні і впровадженні автоматизованої інформаційної системи необхідно вирішити ряд проблем, пов'язаних з адаптацією наявних програмних засобів автоматизації абонентських служб об'єднань співвласників багатоквартирних будинків та конвертації даних про нарахування та оплату послуг. На першому етапі впровадження автоматизованої інформаційної системи має бути вирішене завдання синхронізації даних про всіх споживачів з різних об'єднань співвласників багатоквартирних будинків. Оскільки такі системи багатофункціональні і значна роль належить зовнішнім інформаційним зв'язкам, тому мають бути вироблені організаційні та адміністративні заходи по реалізації автоматизованої інформаційної системи.

Мета роботи: підвищення ефективності процесів управління ОСББ в частині обліку розрахунків за житлово-комунальні послуги шляхом автоматизації процесів ведення обліку надходжень за надані послуги та витрат на поточні

заплановані або незаплановані роботи, визначення боржників і планування фінансових потоків.

Для запровадження автоматизованої інформаційної системи обліку розрахунків за надані ОСББ послуги мають бути вирішені наступні основні задачі:

- проаналізована діяльність організації та пов'язані з цією діяльністю бізнес-процеси та інформаційні ресурси організації;
- визначені проблеми, що виникають в наслідок неналежного інформаційного забезпечення поточної діяльності організації;
- визначені напрямки, технології та механізми вирішення нагальних проблем інформаційного забезпечення діяльності організації;
- запропонована концепція автоматизованої інформаційної системи організації;
- змодельовані основні бізнес-процеси організації в частині ведення обліку розрахунків;
- розроблені рішення з проєктування автоматизованої інформаційної системи;
- реалізоване програмне та інші види забезпечення автоматизованої інформаційної системи.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ, РОЗРОБКА КОНЦЕПЦІЇ ПІДСИСТЕМИ ОБЛІКУ РОЗРАХУНКІВ ОСББ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Напрямки діяльності ОСББ

Одне із найважливіших завдань будь-якого органу місцевого самоврядування полягає в забезпеченні комфортного проживання мешканців населеного пункту, раціональному використанні рухомого й нерухомого майна, земельних ділянок та інших ресурсів, що перебувають у комунальній (тобто загальній, спільній) власності. Для забезпечення належних умов проживання мешканців населених пунктів або територіальних підрозділів населених пунктів створюють комунальні (муніципальні) організації та підрозділи. Одним з різновидів таких організацій в Україні є ОСББ.

Зважаючи на процеси роздержавлення та приватизації у комунальному секторі, традиційні комунальні організації – житлово-експлуатаційні контори, ЖЕКи – виявились достатньо ефективними у наданні таких послуг, як збір відходів, розподіл води та послуг транзиту, але у частині утримання будинків та прибудинкових територій ці організації виявились неефективними, тому їм на заміну прийшли ОСББ. При цьому виникла потреба забезпечення збору коштів за отримані послуги в межах ОСББ та ведення обліку і розрахунків з муніципальними корпораціями (тепло- та енерго- компаніями, компаніями з водопостачання та водовідведення тощо), що виявилось життєздатним засобом також і для надання деяких місцевих організаціям державних послуг [7].

При здійсненні своєї діяльності ОСББ керується Законом України “Про об'єднання співвласників багатоквартирного будинку” і вирішує наступні задачі:

- Забезпечує ведення обліку та організацію управління об'єктами житлово-комунального господарства, що перебувають у власності або спільному користуванні громади.
- Забезпечує реалізацію власних і делегованих повноважень щодо комплексного розвитку житлово-комунального господарства з питань впорядкування та прибирання прибудинкових територій, експлуатації та ремонту житла, водо- та теплопостачання, водовідведення, зеленого господарства, дитячих майданчиків, технічних приміщень тощо.

- Бере участь у розробленні проєктів програм технічного розвитку комунального господарства і цільових програм, метою яких є підвищення рівня забезпеченості населення, соціальної сфери в усіх видах житлово-комунальних послуг, поліпшення їх якості.
- Забезпечує облік та аналіз споживання житлово-комунальних послуг на території та надає інформаційні послуги з окремих видів своєї діяльності.
- Забезпечує участь в проведенні зборів мешканців для розв'язання наявних проблем об'єднань співвласників багатоквартирних будинків.

В межах своїх повноважень ОСББ надають мешканцям населених пунктів послуги з постачання електроенергії, води, газу, опалення тощо; забезпечують відведення каналізаційних стоків, вивіз сміття тощо (рисуюнок 1.1).



Рисуюнок 1.1. Види та обсяги розрахунків за надані комунальними організаціями послугами [5]

В умовах повномасштабної російсько-української війни ОСББ також активно впроваджують нові технології, у тому числі [13]:

- автономні системи опалення (газові котельні та/або теплові насоси, зазвичай встановлюються у спеціальних технічних приміщеннях на даху будівлі);
- аварійні автономні системи електроживлення (на основі двигунів внутрішнього згорання);

- автономні системи генерації електроенергії (на основі сонячних панелей та акумуляторних батарей для накопичення електроенергії);
- автономні системи очищення питної води (на основі напівпромислових систем фільтрування води);
- системи відеоспостереження та сигналізації (на основі веб-камер, у тому числі — з підключенням до мережі Інтернет);
- системи керування і управління доступом (СКУД, на основі електронних замків і ключів);
- загальнодомові лічильники обсягів спожитого газу, води, тепла, електроенергії тощо.

Зазначені системи є складними і вимагають безперервного і кваліфікованого обслуговування. Вирішення цих задач вимагає від керівництва ОСББ не тільки залучення відповідних фахівців, але і здійснення своєчасних розрахунків за виконані роботи, планування робіт з обслуговування і ремонту, закупівлі додаткового обладнання тощо.

Для виконання розрахунків за надані послуги при комунальних організаціях створюють спеціалізовані центри розрахунків. Особлива роль АІС ОСББ полягає в тому, що такі АІС відкривають нові можливості для підвищення дисципліни розрахунків та зменшення заборгованості за отримані комунальні послуги. Також з'являються можливості оперативного формування звітів про рівень збору платежів за комунальні послуги і стан заборгованості за послуги, аналізу, прогнозування і планування розвитку житлово-комунального господарства. За накопиченими даними та відповідно до наявної моделі їх представлення в перспективі можна буде аналізувати вплив соціальних, економічних та інших чинників і факторів на стан оплати житлово-комунальних послуг.

В процесі ведення обліку виконаних робіт з забезпечення експлуатації технологічного обладнання та наданих клієнтам послуг, ведення розрахунків з постачальниками, планування потреб в ресурсах та забезпечення бухгалтерського і оперативного обліку, а також забезпечення інших видів управлінської діяльності в ОСББ склалась інформаційна система підтримки управлінської діяльності, в якій забезпечується вирішення наступних основних задач:

- інформаційне забезпечення процесів експлуатації технічних засобів, житлового фонду з проведенням поточних і капітальних ремонтів, реконструкцій і модернізацій кабельних, теплових і водопровідно-каналізаційних мереж;
- інформаційне забезпечення процесів підтримки зелених насаджень, впорядкування та прибирання прибудинкових територій;
- ведення обліку по отриманих організацією ресурсах: комплектуючих, теплової та електроенергії, холодної та гарячої води тощо;
- ведення обліку стану оплати послуг, що надаються комунальною організацією, по фізичним та юридичним особам;
- ведення обліку звернень громадян;
- ведення обліку робіт з обслуговування загальнодомових приладів обліку (лічильників);
- надання довідкової інформації та інформації про стан рахунків.

Ведення обліку оплати комунальних послуг ОСББ включає вирішення наступних основних задач:

- формування звітів по наданих послугах по організації, що надає послуги, і споживачам послуг;
- формування звітів по оплачених послугах у розрізі за споживачами послуг та послугами.

Облік оплати комунальних послуг ОСББ також напряду пов'язаний з процесами розрахунків заробітної платні працівникам організації і з процесами розрахунків з контрагентами.

Зв'язок з процесами розрахунків заробітної платні працівникам організації обумовлений тим, що наймані працівники організації так само сплачують за комунальні послуги, таким чином, що сплата може здійснюватись автоматично про нарахуванні заробітної платні шляхом перерахування коштів за отримані робітником комунальні послуги на рахунок ОСББ. Така процедура визначена тим, що частина робітників проживають у житловому фонді, який знаходиться у підпорядкуванні ОСББ.

Зв'язок з процесами розрахунків з контрагентами організації обумовлений тим, що для окремих контрагентів є зручним здійснювати взаєморозрахунки з комунальним організацією, оскільки контрагенти не тільки надають послуги і

матеріали ОСББ, але іноді і самі отримують послуги від ОСББ, наприклад, через оренду окремих технічних приміщень або частини прибудинкової території.

Отже, такі зв'язки підвищують дисципліну розрахунків за комунальні послуги, але, у свою чергу, призводять до ускладнень процесів нарахування заробітної платні та розрахунків з контрагентами.

Технології обробки електронних документів та засоби мережі Інтернет дозволять забезпечити різноманітні види інформаційного обміну між центром розрахунків та споживачами послуг, у тому числі про тарифи, надані послуги, заборгованість, поточний стан розрахунків тощо. Як наслідок, в межах інформаційної системи населеного пункту часто функціують кілька автоматизованих інформаційних систем, для доступу до яких загалом використовуються автоматизовані робочі місця, розподілені між структурними підрозділами органів управління населеного пункту і об'єднаних в локальну мережу. При цьому АІС ОСББ може бути підключена до АІС населеного пункту, що дозволить забезпечити взаємодію з наступними складовими:

- автоматизована інформаційна система ведення обліку підприємства зелених насаджень (ЗеленБуд);
- автоматизована інформаційна система підприємств з впорядкування та прибирання прибудинкових територій;
- автоматизовані інформаційні системи підприємств з енергопостачання, тепlopостачання, водопостачання, каналізаційного відведення тощо.

Інші автоматизовані інформаційні системи, з якими може взаємодіяти АІС ОСББ, забезпечують часткове вирішення задач з виконання інженерних розрахунків при плануванні, ведення кадрового обліку та обліку матеріалів, виконаних земляних робіт, ведення звернень громадян, здійснення розрахунків з банками тощо.

1.2. Визначення недоліків наявної автоматизованої інформаційної системи ОСББ

В частині використання спільних інформаційних ресурсів, форматів представлення даних та інших аспектів діяльності при взаємодії інформаційної системи організації з наявними у населеному пункті підприємствами та їх автоматизованими системами, засоби взаємодії є недостатньо інтегрованими, що

стало наслідком різноспрямованих підходів при їх створенні. Слабка горизонтальна інтеграція виникла як в наслідок застосування різно направлених процесно-орієнтованих підходів, що застосовувались при автоматизації в окремих підрозділах комунальних служб населеного пункту, так і з причини використання в окремих автоматизованих системах різних форматів представлення даних, а також різного інструментального програмного забезпечення, серед якого у тому числі є вже морально застарілі наступні []:

- пакет *Visual FoxPro* – для ведення обліку з зеленого будівництва, впорядкування та прибирання прибудинкових територій, теплопостачання, водопостачання та каналізаційного відведення та частково інших задач;
- програмні модулі на базі 1С – для ведення бухгалтерського, кадрового і складського обліку;
- програмні модулі на базі додатків *Microsoft Office* – для вирішення інженерних та інших задач.

Іншим суттєвим недоліком інтеграції наявної на підприємстві автоматизованої інформаційної системи та систем населеного пункту є те, що вирішення центральних задач з ведення обліку про стан розрахунків на рівні населеного пункту виконується окремими засобами автоматизованої системи, які є закритими в частині архітектури, протоколів, математичного та програмного забезпечення, структур даних тощо, що унеможливорює ефективний розвиток та розширення наявної автоматизованої системи.

Закритість структур даних АІС населеного пункту не тільки унеможливорює внесення змін в інформаційну модель наявної інформаційної системи з метою розширення її функціональності, але й не дозволяє забезпечувати повноцінне, таке, що виходить за межі передбачених в системі механізмів, адміністрування баз даних системи. Загалом, це призводить до того, що в системі з часом експлуатації накопичуються окремі службові дані, які не можуть бути видалені через закритість системи. Як наслідок, виконання окремих типових операцій в системі може займати до декілька хвилин, а генерація окремих звітів – декілька десятків хвилин, що є неприйнятним для розв'язання актуальних задач організації з точки зору поточного стану інформаційних технологій.

Крім того, використання в якості базового програмного забезпечення комерційного власницького продукту 1С не відповідає вимогам та рекомендаціям Кабінету міністрів України щодо використання окремих продуктів виробництва російської федерації як країни-агресора, а використання застарілої версії Visual FoxPro не дозволяє застосовувати розроблені на цій платформі рішення в сучасних операційних системах та на сучасному технічному забезпеченні. Це також ускладнює можливості реінженірингу структур даних та бази даних у цілому.

Наявність зазначених обмежень також не дозволяє забезпечити повноцінну вертикальну інтеграцію інформаційної ОСББ з центрами розрахунків в інтегровану інформаційну систему.

Таким чином, спираючись на результати аналізу застосованих в наявній інформаційній системі організації підходів та технологій, можна зробити висновок про недоцільність або неможливість подальшого розвитку цих систем в частині розширення їх функціональності, взаємної інтеграції та інтеграції з іншими загальноміськими інформаційними системами.

Як наслідок, можна визначити, що розробка проєкту нової відкритої інтегрованої автоматизованої інформаційної системи ОСББ на базі нових технологій проєктування та розробки програмного забезпечення має сенс та є доцільною.

1.3. Аналіз наявних рішень автоматизації діяльності об'єднань співвласників багатоквартирних будинків

Виходячи зі складності процесів обліку розрахунків за надані комунальні послуги та платежів за отримані комунальними організаціями від контрагентів товари та послуги у ряді міст України органи місцевої влади почали шукати більш ефективні шляхи організації роботи об'єднань співвласників багатоквартирних будинків, що призвело до розширення структури управління населеного пункту і, як наслідок, до зростання складності процесів управління та контролю.

З метою забезпечення інтеграції АІС ОСББ в інформаційні системи населеного пункту та враховуючи спорідненість задач розглянемо типові для населених пунктів рішення, що застосовуються у сфері житлово-комунального

комплексу. Як правило, автоматизовані системи збору та обробки платежів за житлово-комунальні послуги створювались на основі спеціалізованих інформаційно-обчислювальних центрів при органах міської влади з використанням єдиних загальноміських баз даних усіх споживачів комунальних послуг та впровадженням сучасних методів розрахунку, збору та обліку розрахунків. При цьому, залежно від потреб та можливостей, в Україні і за кордоном зазвичай використовують:

- спеціалізовані програмні продукти у складі ERP-систем з високим рівнем інтеграції з іншими підсистемами [6];
- самостійні продукти з розвинутим інтерфейсом (рисунки 1.3, 1.4);
- “самописні” рішення, зазвичай на основі електронних таблиць офісних пакетів (Open Office, Libre Office, MS Office тощо) з вкрай простим інтерфейсом (рисунок 1.5).

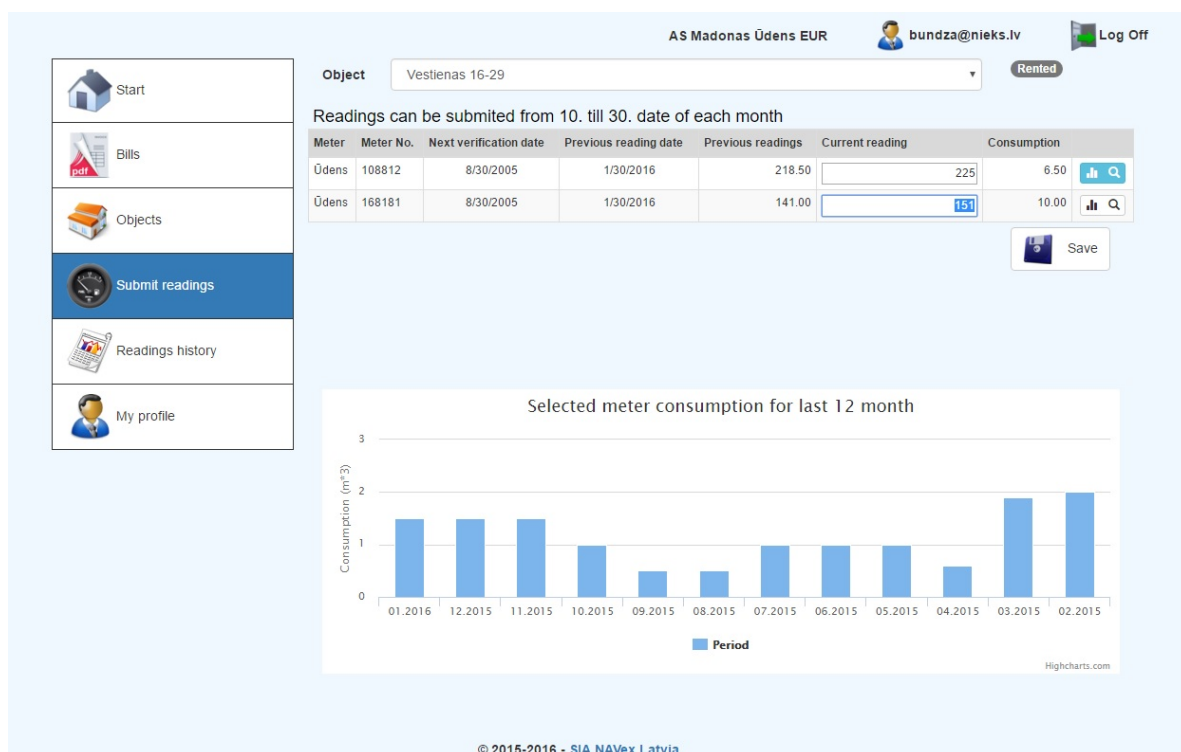


Рисунок 1.3. Приклад інтерфейсу програми обліку комунальних послуг з розвинутим діалоговим інтерфейсом та візуалізацією

Лише найбільш розвинуті з таких програмних продуктів інтегровані з іншими інформаційними системами міста, а інші лише частково використовують механізми реалізації систем автоматизації збору платежів за житлово-комунальні

послуги, які надають розробникам широкі можливості для подальшого розширення електронних засобів отримання даних з інших автоматизованих систем.

Edit - Service Address Card - C00050 · SA00005 · 01.05.2016

HOME NAVIGATE

View Edit New Delete Notes Links Refresh Clear Filter Go to Previous Next

Manage Show Attached Page

C00050 · SA00005 · 01.05.2016

General

Customer No.: C00050 Flat No.:
No.: SA00005 Blocked:
Customer Name: John Source:
Description: Service Address Statistic Grp:
Street Code: Headcount: 0
Street Name: Group Code: AP
Building No.: Department Code:
Project Code:

Service

Service Filter Clear Filter

Card
Related Utilities
Ledger Entries Ctrl+F7
Dimensions Ctrl+Shift+D
Meter Change Log
Change/Install Meter Ctrl+Shift+N
Connected subscriber Ledger Entries
Warnings Ctrl+B
Transfer to another Service Address

Code	Accounting Method	Resource No.	Description
COLD	Meter	CW	Cold Water

Free Water: ☐
Hot Water: ☐
Rain Water: ☐
Own Water: ☐

Communication
Invoices

OK

Рисунок 1.4. Приклад інтерфейсу програми обліку комунальних послуг з діалоговим текстовим інтерфейсом

Комплексна автоматизована система розрахунків населення за житло та комунальні послуги у м. Києві. Система розрахунків населення за житло та комунальні послуги є частиною автоматизованої технології управління житлово-комунальним господарством міста і яскравим прикладом організованого і централізованого варіанту вирішення проблеми збору платежів за надані комунальні послуги та аналізу фінансової діяльності об'єднань співвласників багатоквартирних будинків.

City of Bingham			
Water Utility Fund			
Statement of Revenue, Expenses and Changes in Net Position			
For the Year Ended December 31, 2017			
Operating Revenue			
Sales of Water (Net of \$4,328 Provision for Doubtful Accounts)			
Operating Expenses			
Operations			
Maintenance			
Depreciation Expense			
Operating Income			
Non-Operating Expenses			
Interest on Long-Term Debt			
Change in Net Position			
Total Net Position January 1			
Total Net Position December 31			

Рисунок 1.5. Приклад інтерфейсу програми обліку розрахунків на базі MS Office

Головною метою створення системи було прискорення обігу грошових коштів в організаціях житлово-комунального господарства при збереженні єдиної розрахункової квитанції, єдиної загальноміської бази даних квартиронаймачів та збереження можливості сплати по квитанції в будь-якій банківській організації м. Києва. Обслуговує понад 800 тис. особистих рахунків платників за житло та комунальні послуги. База даних підприємств – постачальників комунальних послуг охоплює понад 420 підприємств.

Система використовує високопродуктивні мережеві програмно-технічні та телекомунікаційні засоби на базі оптоволоконного зв'язку. Впровадження та експлуатація системи забезпечили можливість:

- одержувати оперативну, своєчасну й достовірну інформацію про стан нарахування та сплати населенням міста за житло та спожиті комунальні послуги;
- спростити процедуру призначення пільг та субсидій;
- спростити розрахункові операції населення з організаціями – постачальниками послуг та ЖЕО;

отримувати точну й своєчасну інформацію про надходження коштів для відшкодування пільг та субсидій та про заборгованість населення по сплаті за житло та комунальні послуги.

Інформаційна мережа для автоматизації процедури оплати комунальних послуг і нарахування субсидій у м. Хмельницькому. Створена і впроваджена у м. Хмельницькому інформаційна мережа спрощує процедуру оформлення житлових субсидій і процедуру збирання платежів за надані житлово-комунальні послуги. Усі бази даних формуються і наповнюються в комунальних організаціях, а Інформаційний центр забезпечує тільки обмін даними між організацією і пунктами приймання. Це сприяє підвищенню відповідальності організації при формуванні бази даних, прискорює виявлення неточностей, які історично прижилися в межах одного організації.

Ядром системи є інформаційно-обчислювальний центр міської ради. Тут з баз даних об'єднань співвласників багатоквартирних будинків щомісячно формується єдина база даних нарахувань за надані послуги і передається в усі пункти приймання платежів.

Характерною особливістю і відмінною рисою цієї мережі є те, що вона забезпечує можливість платити за вибором та незалежно від місця проживання. Пошук організований за адресою, ідентифікаційним кодом або прізвищем платника. На пункті приймання оплати за комунальні послуги мешканець отримує повну інформацію про стан сплати за комунальні послуги та наявну заборгованість. Квитанції про сплату роздруковуються під час оплати в пункті приймання платежів, що дозволяє уникнути операцій доставки квитанцій громадянам за місцем проживання та оброблення корінців оплачених квитанцій. Мешканці мають можливість платити за послуги по декількох лічильниках (теплової та електричної енергії, води тощо).

Ще однією перевагою такої мережі є те, що вона забезпечує обмін інформацією між пунктами приймання оплати за комунальні послуги і комунальними організаціями. Обмін відбувається щоденно в автоматичному режимі за допомогою каналів зв'язку загального доступу.

Система також значно спрощує роботу працівників ЖЕКів при підготовці різноманітної звітності інформаційного плану. Втричі скорочуються затрати часу для видачі довідок про склад сім'ї та стан оплати за комунальні послуги. За лічені хвилини можна отримати інформацію про рівень заборгованості за квартплату, список боржників та пільговиків, зробити перерахунки квартплати, скласти списки квартир, де квартплата з тих чи інших причин не нараховується тощо.

Централізована автоматизована система збору житлово-комунальних платежів у м. Луцьку. Луцька система збирання житлово-комунальних платежів (система збирання та обліку даних про оплату послуг) бере свій відлік з травня 1995 р. Її суть полягає в об'єднанні всіх інформаційних потоків, від яких залежать нарахування громадянам за житлово-комунальні послуги. Запровадження цієї системи дало змогу:

Ядром системи є міський обчислювальний центр, який забезпечує створення і супровід загальноміської бази даних про споживачів комунальних послуг; збір інформації про нарахування субсидій, про переміщення громадян, зміни в умовах проживання; проведення розрахунків вартості комунальних послуг; нарахування субсидій і їх перерахування при будь-яких змінах; реєстрація та облік пільг з оплати житлово-комунальних послуг; забезпечення пунктів приймання платежів інформацією про нарахування комунальних послуг і розмір субсидій; оброблення даних про фактичну оплату послуг; формування звітів про стан оплати комунальних послуг.

Автоматизована система фірми "ВАС" у м. Кременчук. Система обслуговує понад 120 тис. абонентів, понад 50 різних комунальних служб. До системи підключились 7 банків, у яких створено 50 автоматизованих робочих місць по прийманню платежів.

Система працює в режимі реального часу, коли дані про нарахування, сплату, надходять до відповідних баз даних через кілька секунд. Учасниками автоматизованої системи є банки, комунальні служби, відділи субсидій та

розрахунковий центр. Автоматизовано процедуру обміну інформацією між усіма учасниками системи.

Уся інформація про нарахування, пільги, субсидії передається розрахунковим центром до автоматизованих пунктів приймання платежів. Розрахунковий центр отримує всю інформацію про оплату послуг. До розрахункового центру також передається інформація про зміни в нарахуваннях за будь-яким видом послуг від конкретного виконавця послуг. Всі комунальні служби мають можливість переглянути в системі картки з нарахуваннями по кожному абоненту та отримати будь-які необхідні дані та звіти, але тільки за "свою" послугою.

Для отримання субсидій абонентам необхідно звернутися тільки до відділу субсидій, які мають інформацію про нарахування та платежі даного абонента. Будь-які зміни, внесені відділом субсидій, відразу надходять до розрахункового центру та до пунктів приймання. Внесена в систему інформація про грошові кошти рухається безпосередньо від платника (споживача) через банк на рахунок організації-постачальника послуг.

Єдина система обліку оплати населенням спожитих житлово-комунальних послуг та енергоносіїв у м. Харків.. Увесь масив платежів (70% населення) обробляється вручну: квитанції про сплату доставляються до єдиного розрахункового центру, де оператори вносять їх до бази даних про нарахування і сплату послуг.

Єдиний розрахунковий центр виконує лише функцію "розщеплення" платежів та забезпечує перерахування на рахунки постачальників послуг, використовуючи бази даних нарахувань житлово-комунальних підприємств та бази даних субсидій. Всі організації-постачальники послуг зобов'язані регулярно передавати до "Мегабанку" бази даних про нарахування платежів до сплати. Установи, які приймають платежі від населення, повинні перераховувати кошти на рахунок "Мегабанку" і передавати до "Мегабанку" квитанції про сплату.

Система обліку комунальних послуг та бухгалтерія об'єднань співвласників багатоквартирних будинків, Молдова, реалізована на базі відкритої ERP-системи ODOO. Забезпечує вирішення наступних задач [6]:

- Облік абонентів - введення абонентів і облік по вулицях, ділянках, населеним пунктам;

- Введення залишків - введення залишків по абонентам і послуг;
- Послуги і тарифи - можливість впровадження широкого спектру послуг і розрахунку тарифів по водоміра, за кількістю і за іншими параметрами;
- Друк квитанції - друк квитанції та редагування шаблонів відповідно до моделей організації;
- Облік водомірів - установка, показання лічильника, деінсталяція і зміна лічильників.

Отже, кожна з наявних автоматизованих інформаційних систем має як спільні риси, обумовлені вирішенням тих самих завдань, так і ряд відмінностей, зумовлених конкретними місцевими умовами та особливостями реалізації. Однозначно говорити про переваги чи недоліки тієї чи іншої системи не можна, оскільки схема їх реалізації залежить як від фінансових та технічних можливостей окремого населеного пункту, так і від рівня автоматизації об'єднань співвласників багатоквартирних будинків, розміру населеного пункту тощо.

Також слід враховувати, що на відміну від АІС комунальних підприємств і служб населених пунктів, що оперують інформацією про десятки або сотні тисяч абонентів, АІС ОСББ повинна оперувати інформацією про сотні абонентів. Ця особливість визначає пріоритети при проектуванні і розробці АІС: головне не швидкість обробки, а можливість інтеграції з окремими абонентами – мешканцями ОСББ з одного боку, і інтеграція з АІС комунальних служб міста – з іншого. Ці особливості мають бути враховані при розробці концепції підсистеми обліку розрахунків АІС ОСББ.

1.4. Розробка концепції підсистеми обліку розрахунків АІС ОСББ

З метою забезпечення можливостей горизонтальної інтеграції підсистем проєктованої автоматизованої системи між собою та вертикальної інтеграції системи в інформаційну систему населеного пункту в проєкті підсистеми обліку розрахунків автоматизованої інформаційної системи ОСББ повинні бути передбачені відповідні механізми.

Забезпечення можливостей горизонтальної інтеграції підсистем між собою має передбачати такі інструменти реалізації системи, що дозволять досить просто нарощувати функціональність системи шляхом розширення інформаційної бази системи та введення в систему нових модулів.

Вертикальна інтеграція системи в інформаційну систему міста має передбачати забезпечення таких можливостей доступу до системи з боку зацікавлених осіб, що не повинні вимагати суттєвої модифікації їх систем шляхом надання віддаленого доступу до ресурсів системи для інших клієнтів системи – через локальну мережу або мережу Інтернет.

Локальний доступ повинен забезпечуватись через автоматизовані робочі місця (АРМ), встановлені на робочих місцях уповноваженого персоналу організації. Для цього на клієнтських робочих місцях має бути встановлене відповідне клієнтське програмне забезпечення, а клієнтам мають бути надані необхідні права та повноваження з доступу до інформаційних ресурсів сервера автоматизованої інформаційної системи відповідно до функціональних обов'язків кожного з користувачів – клієнтів системи.

Глобальний (віддалений) доступ для інших клієнтів системи має забезпечуватись через мережу Інтернет.

В результаті забезпечення означених механізмів доступу, доступ до інформаційних ресурсів повинен надаватися для таких видів користувачів системи:

- авторизованих користувачів – співробітників розрахункового центру та / або комунальних служб (надання прав і повноважень доступу до інформаційних ресурсів системи забезпечується як для режиму локального доступу);
- авторизованих користувачів – отримувачів послуг (авторизація виконується на основі відомої користувачам приватної облікової інформації: адреса, номер розрахункової книжки тощо);
- інших користувачів – інформація надається без авторизації (оголошення, тарифи, таблиці-графіки ремонтних робіт тощо).

Важливим, з точки зору організації автоматизованої системи, також є вибір інструментального забезпечення інтеграції підсистем в інформаційну систему. У відносно невеликих проєктах така інтеграція забезпечується безпосередньо на рівні програмного забезпечення системи таким чином, що зв'язки між підсистемами визначаються через безпосереднє звернення до об'єктів даних системи та виклику їх функцій. Наявність таких зв'язків спрощує розробку системи на початкових етапах її реалізації, але, на жаль, не дозволяє простими

засобами нарощувати систему в завершальних стадіях її розробки та на етапі експлуатації – коли будь яка модифікація системи на рівні одного модуля неминуче призводить до зміни встановлених зв'язків між цим модулем та іншими модулями.

Для подолання цієї проблеми в складних системах реалізують більш гнучкі технології взаємодії підсистем шляхом реалізації механізмів створення зв'язків між модулями, в тому числі – користувачем системи на етапі її експлуатації. Так, в системах, подібних до MS Office або 1С, та в більшості інших складних систем реалізовані і надаються користувачам механізми, що дозволяють встановлювати нові правила ведення розрахунків, змінювати структуру бази даних шляхом встановлення нових зв'язків, модифікації наявних або створення нових таблиць бази даних тощо. Реалізація таких механізмів вимагає забезпечення специфічних технологій доступу до даних, а в окремих випадках – ще й використання спеціалізованих мов програмування з розширеною підтримкою механізмів метапрограмування.

З врахуванням вищенаведеного, може бути запропонована концепція підсистеми обліку розрахунків автоматизованої інформаційної системи ОСББ з АРМ бухгалтера та адміністратора підсистеми обліку розрахунків. Адміністратор автоматизованої інформаційної системи ОСББ також повинен забезпечити подальше підключення інших АРМ для інших користувачів системи, можливість передбачена у технічному завданні проєкту.

Концепція представлена на рисунку 1.6 в нотації моделі кооперації UML.

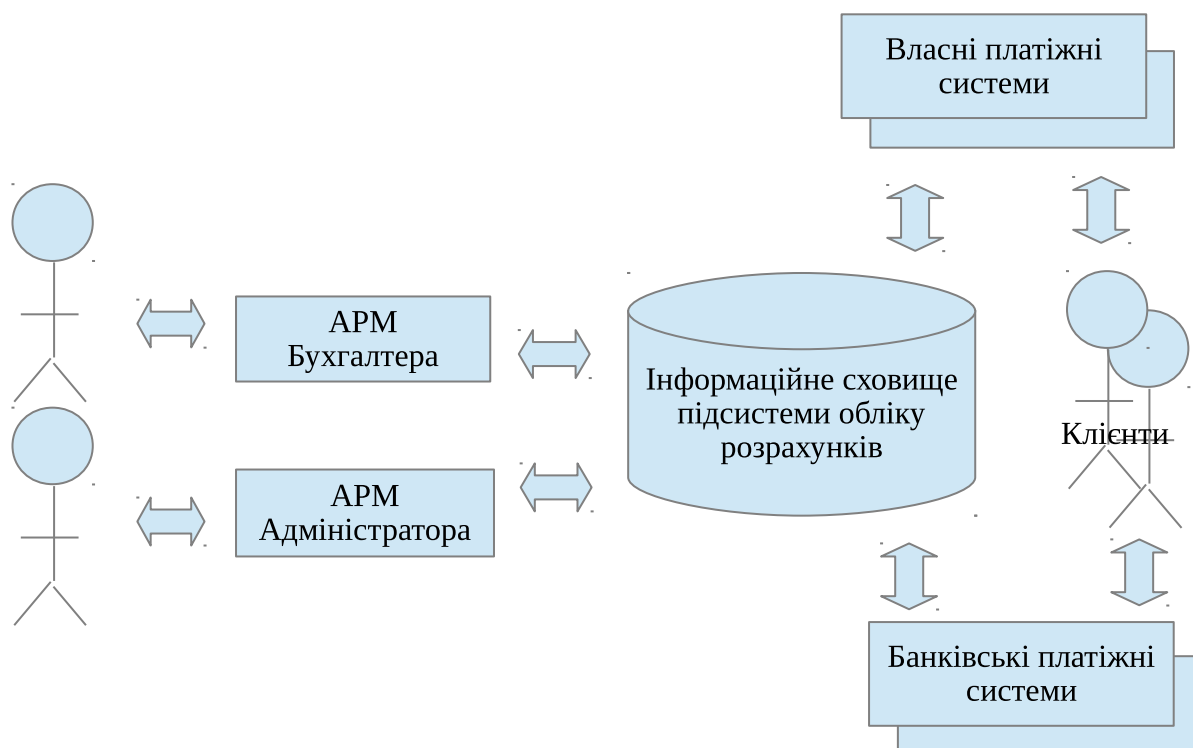


Рисунок 1.6. Концепція підсистеми обліку розрахунків автоматизованої інформаційної системи ОСББ

1.5. Функціонально-вартісний аналіз варіантів концепції підсистеми

1.5.1. Функціональний аналіз

На основі вимог до функцій та якостей автоматизованої системи визначимо параметри системи:

- захищеність програмного забезпечення;
- швидкість розробки ПЗ проєкту;
- наочність коду;
- швидкість обробки запитів;
- валідність коду.

Результати експертних оцінок вагових параметрів методом попарного порівняння і визначення вагових коефіцієнтів φ_i наведені в таблиці 1. Кількість експертів має бути непарною. В якості експертів можуть виступати спеціалісти з експлуатації, використання та тестування інформаційного забезпечення. Усереднені оцінки експертів (максимальне значення параметра дорівнює 10 балів, мінімальне - 0) занесені в таблицю 1.2.

Таблиця 1.1. Оцінки параметрів варіантів АІС (варіант 1 – Python, варіант 2 – Java)

Параметр	Варіант 1	Варіант 2
Захищеність програмного забезпечення	9	8
Швидкість розробки ПЗ проєкту	7	8
Наочність коду	9	10
Швидкість обробки запитів	8	9
Підтримка технологій метапрограмування	9	3
Валідність коду	9	9

Визначимо також коефіцієнти значущості φ_i і бальні оцінки B_i інформаційної системи (представлені в таблицях 1.2 та 1.3). Крім того, визначимо коефіцієнт технічного рівня $K_{mp} = \sum \varphi_i * B_i$, який визначає технічний рівень застосованих у системі технологій.

Результати розрахунків за варіантами (варіант 1 – Python, варіант 2 – Java) концепції автоматизованої інформаційної системи представлені в таблиці 1.4.

Таблиця 1.2. Визначення коефіцієнта значущості φ_i

Параметр	φ_i
Захищеність програмного забезпечення	0,40
Швидкість розробки ПЗ проєкту	0,35
Наочність коду	0,4
Швидкість обробки запитів	0,35
Підтримка технологій метапрограмування	0,70
Валідність коду	0,35

1.5.2. Вартісний аналіз

Аналіз функціональних властивостей системи та її функцій має бути доповнений вартісним аналізом. Для цього розглянемо зведені витрати по кожному з варіантів, запропонованих у проєкті. Результати вартісного аналізу зведені в таблиці 1.3 та 1.4.

Зведені витрати Z_j розраховуються у сфері виробництва (у розробника) або у сфері експлуатації (у виконавця). У нашому випадку, досить розрахувати у сфері виробництва.

Поточні витрати при розробці інформаційної системи Z – це собівартість створення системи. Для усіх варіантів вважатимемо, що собівартість системи однакова.

Капітальні вкладення – це вкладення, які мають одноразовий характер. В даному випадку, в капітальні вкладення варто включити купівлю ліцензійного програмного забезпечення, технічного і організаційного забезпечення, необхідного для створення системи, інші капітальні вкладення в проєкті відсутні – вони будуть здійснюватись цільовим коштом громади за її статтями розвитку.

Зважаючи, що робота виконується як дипломна, а в проєкті застосовується здебільшого безкоштовне вільне системне та інструментальне програмне забезпечення, вартість програмного забезпечення за обома варіантами на даний момент будемо вважати однаковою.

Таблиця 1.3. Визначення коефіцієнтів ІС (варіант 1 – Python, варіант 2 – Java)

Параметр	Варіант	Бальна	Ваговий	K_{mp}
----------	---------	--------	---------	----------

		оцінка B_i	коефіцієнт φ_i	
Захищеність програмного забезпечення	1	9	0,40	3,6
	2	8		3,2
Швидкість розробки ПЗ проєкту	1	7	0,35	2,45
	2	8		2,8
Наочність коду	1	9	0,4	3,6
	2	10		4
Швидкість обробки запитів	1	8	0,35	2,8
	2	9		3,15
Підтримка технологій метапрограмування	1	9	0,70	6,3
	2	3		2,1
Валідність коду	1	9	0,35	3,15
	2	9		3,15

Таблиця 1.4. Коефіцієнти технічного рівня (варіант 1 – Python, варіант 2 – Java)

Варіант 1	21,9
Варіант 2	18,4

1.6. Висновки до розділу 1

У роботі був виконаний аналіз функцій підсистеми обліку розрахунків автоматизованої інформаційної системи ОСББ, визначений склад загальних вимог та специфічні до діяльності вимоги до інформаційного забезпечення, форматування та збереження документів, визначені проблеми наявної автоматизованої інформаційної системи.

На основі результатів аналізу наявних рішень для систем автоматизованої обробки документів об'єднань співвласників багатоквартирних будинків України та зарубіжжя, та з врахуванням переліку задач, вирішення яких повинна забезпечувати підсистема, була розроблена концепція підсистеми обліку розрахунків автоматизованої інформаційної системи ОСББ у складі АРМ бухгалтера (основний) та АРМ адміністратора системи (допоміжний).

2. ПРОЕКТНІ РІШЕННЯ ПО ПІДСИСТЕМІ ОБЛІКУ РОЗРАХУНКІВ АІС ОСББ

2.1. Загальносистемні рішення по підсистемі обліку розрахунків АІС ОСББ

2.1.1. Функції підсистеми як автоматизованої системи

Виходячи з мети створення проекту підсистеми обліку розрахунків як складової автоматизованої інформаційної системи ОСББ, яка полягає в підвищенні ефективності управління комунальним господарством і організацією в цілому, підвищенні дисципліни розрахунків шляхом розробки та впровадження проекту відкритої інтегрованої автоматизованої інформаційної системи організації на базі нових технологій проектування та програмування, визначимо ключові елементи та функції підсистеми обліку розрахунків автоматизованої інформаційної системи ОСББ.

Основними складовими будь якої автоматизованої системи є об'єкт та суб'єкт управління, структура управління, інформаційне забезпечення системи та потоки даних (рисунок 2.1).



Рисунок 2.1. Узагальнена модель структури підсистеми обліку розрахунків АІС ОСББ та інформаційних потоків організації

Відповідно до вимог технічного завдання проєкт підсистеми обліку розрахунків автоматизованої інформаційної системи ОСББ повинен враховувати наступну функціональність підсистеми обліку розрахунків організації:

- Ведення обліку наданих послуг з тепло-, водо-, каналізаційного господарства (Абонентський відділ):
 - ведення нормативно-довідкової інформації (види послуг, база постачальників і одержувачів послуг);
 - ведення особистих рахунків для фізичних та юридичних осіб;
 - ведення даних лічильників води та тепла;
 - ведення пільг;
 - ведення платежів;
 - формування квитанцій;
 - розрахунок та планування теплових навантажень;
 - формування звітів.
- Ведення бухгалтерського обліку (Бухгалтерія):
 - нарахування заробітної плати;
 - ведення каси та проведення платежів через касу;
 - складання звітів (пенсійний, податковий, листів непрацездатності).
- Ведення кадрового обліку (Бухгалтерія):
 - ведення списків особового складу організації;
- Ведення обліку матеріалів (Бухгалтерія):
 - ведення складу та списань;
 - ведення за матеріально-відповідальними особами;
 - ведення по банках;
 - ведення основних фондів;
 - розрахунки з податку на додану вартість.
- Надання довідкової інформації (Абонентський відділ):
 - надання інформації про стан рахунків;
 - надання інформації про спожиті послуги.

- Адміністрування системи (Адміністратор):
 - створення резервних копій даних системи;
 - відновлення даних системи з резервних копій даних системи;
 - створення, видалення та блокування користувачів системи;
 - редагування облікової інформації користувачів системи;
 - управління правами доступу до даних системи;
 - управління правами доступу до програмних модулів системи.

Зазначені функції дозволяють визначити основні процеси, що підлягають автоматизації, та вхідні та вихідні структури даних, необхідні для інформаційного забезпечення цих процесів (рисунки 2.2). На основі переліку процесів та структур даних надалі можуть бути визначені вимоги до інструментального, математичного, інформаційного, програмного, організаційного та інших видів забезпечення системи.

В частині ведення обліку з електро-, тепло-, водо-, та каналізаційного господарства, інших видів робіт, визначених вище, Абонентський відділ отримує від Клієнтів – фізичних та юридичних осіб, споживачів послуг – необхідну інформацію: про наявність пільг та субсидій; спожиті обсяги енергії, води, тепла та інших ресурсів; виконані платежі. На основі цієї інформації та результатів планування тарифів та навантажень Абонентський відділ виконує для Клієнтів розрахунок сум до сплати за комунальні послуги та формує звіти, які надаються в Бухгалтерію та адміністратору з розрахунків за надані ОСББ послуги. Окрім цього, Абонентський відділ також підтримує базу клієнтів, веде облік пільговиків, встановлених пристроїв обліку спожитих електроенергії, води, теплової енергії тощо.

На основі опису процесів ведення обліку спожитих електроенергії, тепло-, водо-, витрат каналізаційного господарства та інших послуг можуть бути виділені зайняті в процесах компоненти системи – структурні підрозділи організації та Клієнти – а також визначені інформаційні потоки взаємодії підрозділів організації як елементів автоматизованої системи між собою та з Клієнтами підсистеми обліку розрахунків АІС ОСББ.

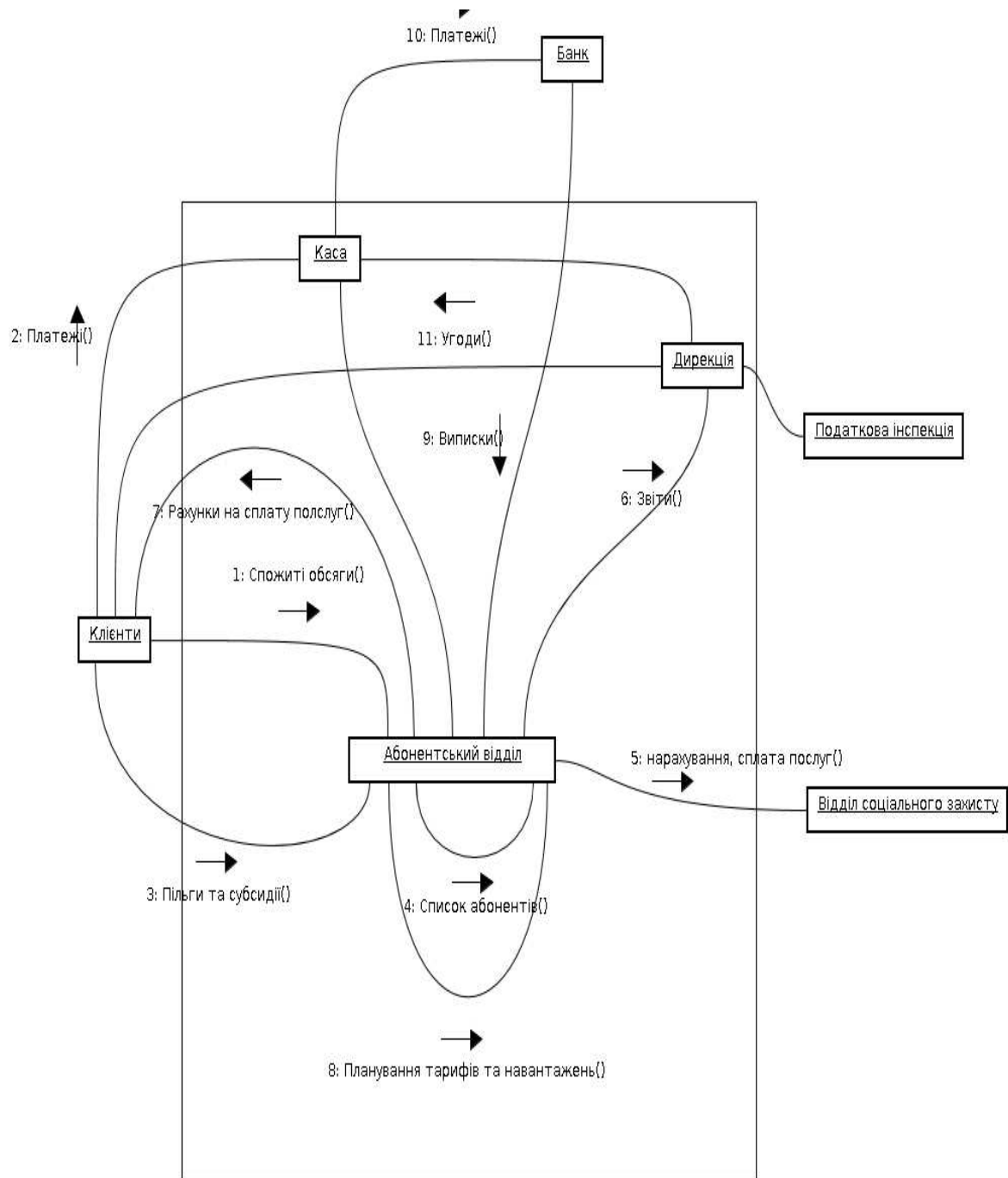


Рисунок 2.2. Модель кооперації системи в частині ведення обліку з тепло- водо- каналізаційного господарства

Діаграма взаємодії (діаграма кооперації) належить до видів діаграм, застосовуваних для моделювання динамічних аспектів системи. На цих діаграмах показують зв'язки, що включають зовнішні об'єкти і відносин між ними, у тому числі повідомлення, якими об'єкти обмінюються. Таким чином

діаграма кооперації відображає структурну організацію активних елементів системи – відправників та отримувачів повідомлень – та використовуються для моделювання функціональних аспектів системи. З точки зору управління системою діаграми взаємодій можуть використовуватися для моделювання окремих потоків управління у складі прецеденту визначених процесів системи – її прецедентів.

Складена відповідно до опису системи модель взаємодії компонентів цієї системи (модель кооперації) представлена на рисунку 2.2.

На рисунках 2.3 та 2.4 представлені деталізовані моделі кооперації відносно кожного з структурних підрозділів організації, відповідно: на рисунку 2.3 – деталізація потоків даних щодо дирекції, на рисунку 2.4 – деталізація щодо абонентського відділу.

2.1.2 Контекстна модель підсистеми обліку розрахунків у частині «сервер баз даних і сервери додатків»

З метою забезпечення можливостей ведення автоматизованих розрахунків та зберігання даних в модель взаємодії компонентів системи (рисунки 2.2, 2.3, 2.4) повинні бути введені додаткові елементи: сервер додатків, на який має бути покладене ведення розрахунків відповідно до існуючих правил; та сервер баз даних, на який має бути покладене забезпечення зберігання даних, ведення резервних копій тощо.

Виокремлення сервера баз даних дозволить забезпечити можливості подальшого розширення системи шляхом введення в систему додаткових модулів, що зможуть використовувати наявну базу даних як інформаційне сховище системи. Завдяки цьому буде забезпечуватись горизонтальна інтеграція цих модулів в систему, а в результаті надання доступу до бази даних системи іншим системам, з'явиться можливість не тільки забезпечити вертикальну інтеграцію системи в загальноміську інформаційну систему, але й надати доступ до публічної інформації будь яким зацікавленим стороннім користувачам системи.

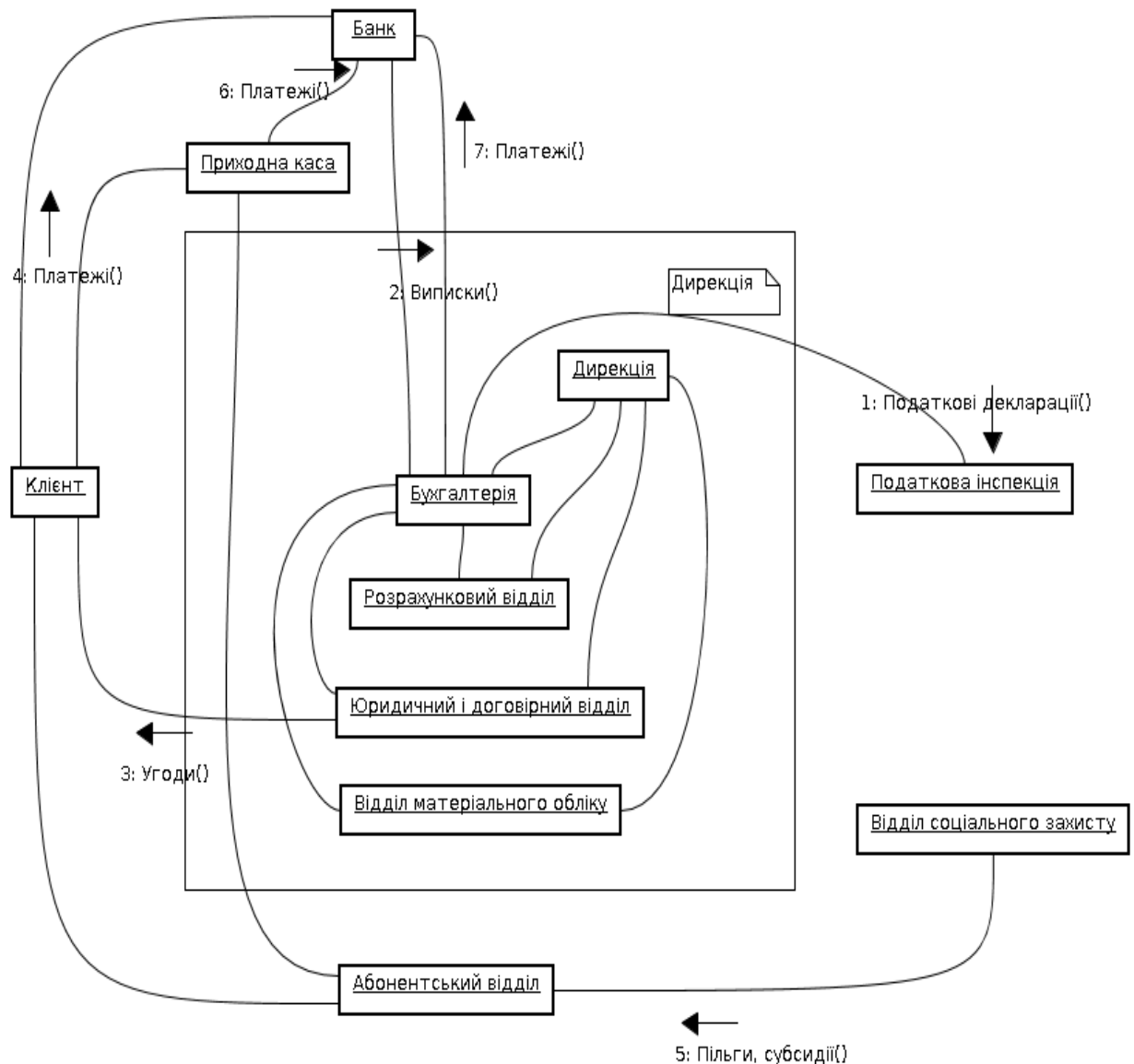


Рисунок 2.3. Деталізована модель кооперації системи в частині роботи підрозділу “Дирекція”

Виокремлення сервера додатків, в свою чергу, дозволить розширювати функціональність системи за рахунок доробки або встановлення додаткових модулів обробки даних незалежно від сервера баз даних. Це дозволить забезпечити мобільність розроблених рішень.

Наявність серверів додатків та баз даних також дозволить отримувати оперативну інформацію про стан розрахунків та виробничих потужностей

напрямую з інформаційного сховища системи. Разом ці два компоненти утворюють підсистему збереження та обробки даних, на яку також має бути покладено задачі надійного збереження даних та їх відновлення при виникненні нештатних ситуацій, обмеження доступу до даних.

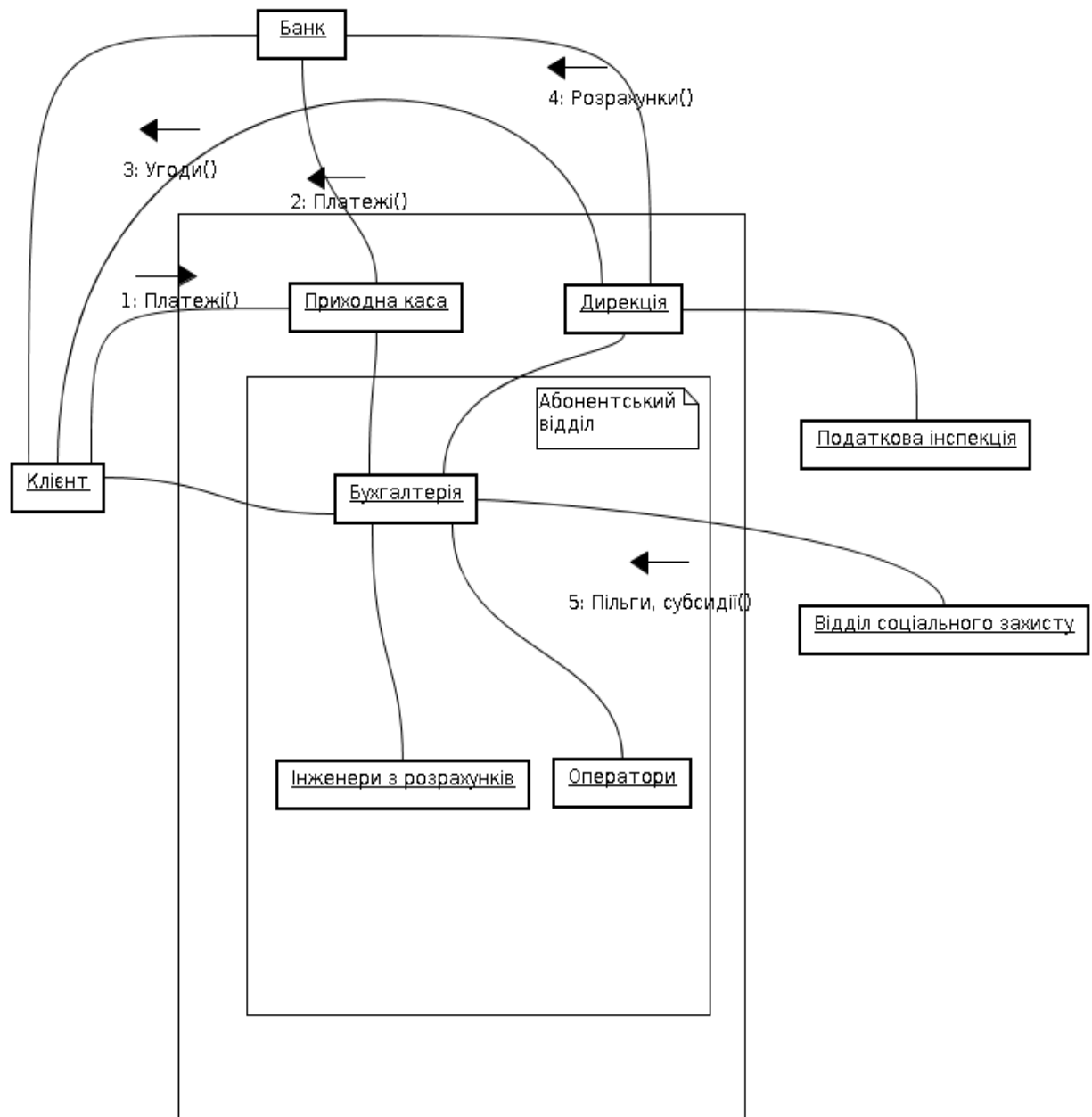


Рисунок 2.4. Деталізована модель кооперації системи в частині роботи підрозділу "Абонентський відділ"

Виходячи з опису функцій системи, викладеному в Пункті 2: “Загальна концепція відкритої автоматизованої інформаційної системи організації”, перелік функцій повинен включати:

- ведення нормативно-довідкової інформації (види послуг, база постачальників і одержувачів послуг);
- ведення особистих рахунків для фізичних та юридичних осіб;
- ведення даних лічильників води та тепла;
- ведення пільг;
- ведення платежів;
- формування квитанцій;
- розрахунок та планування теплових навантажень;
- формування звітів.

На основі розробленої моделі кооперації компонентів системи та з метою формалізації вимог до функцій автоматизованої системи та структур даних системи визначимо перелік функцій системи, що повинні бути автоматизовані.

Модель потоків даних (функціональна модель) являє собою опис потоків даних, які описують окремі операції. Модель потоків даних відображає функціональні залежності значень, що обчислюються в системі, включаючи вхідні та вихідні значення і внутрішні сховища даних. Модель потоків даних звичайно представляють як граф, на якому показано рух даних від джерел через процеси, які забезпечують перетворення даних до споживачів даних в інших об'єктах системи. Модель потоків даних містить процеси, які забезпечують необхідні перетворення даних, потоки даних, які забезпечують перенесення даних, активні об'єкти, які виробляють і споживають дані, і сховища даних, які пасивно зберігають дані.

Для складних ІС будується ієрархія контекстних діаграм. При цьому контекстна діаграма верхнього рівня містить не єдиний головний процес, а набір підсистем, з'єднаних потоками даних. Контекстні діаграми наступного рівня деталізують контекст і структуру підсистем. Ієрархія контекстних діаграм визначає взаємодію основних функціональних підсистем

проектованої ІС як між собою, так і з зовнішніми вхідними і вихідними потоками даних і зовнішніми об'єктами (джерелами і приймачами інформації).

Модель потоків даних автоматизованої системи в частині ведення обліку з електро-, тепло-, водо- і каналізаційного господарства представлена на рисунку 2.5. Модель представлена в нотації моделі варіантів використання UML.

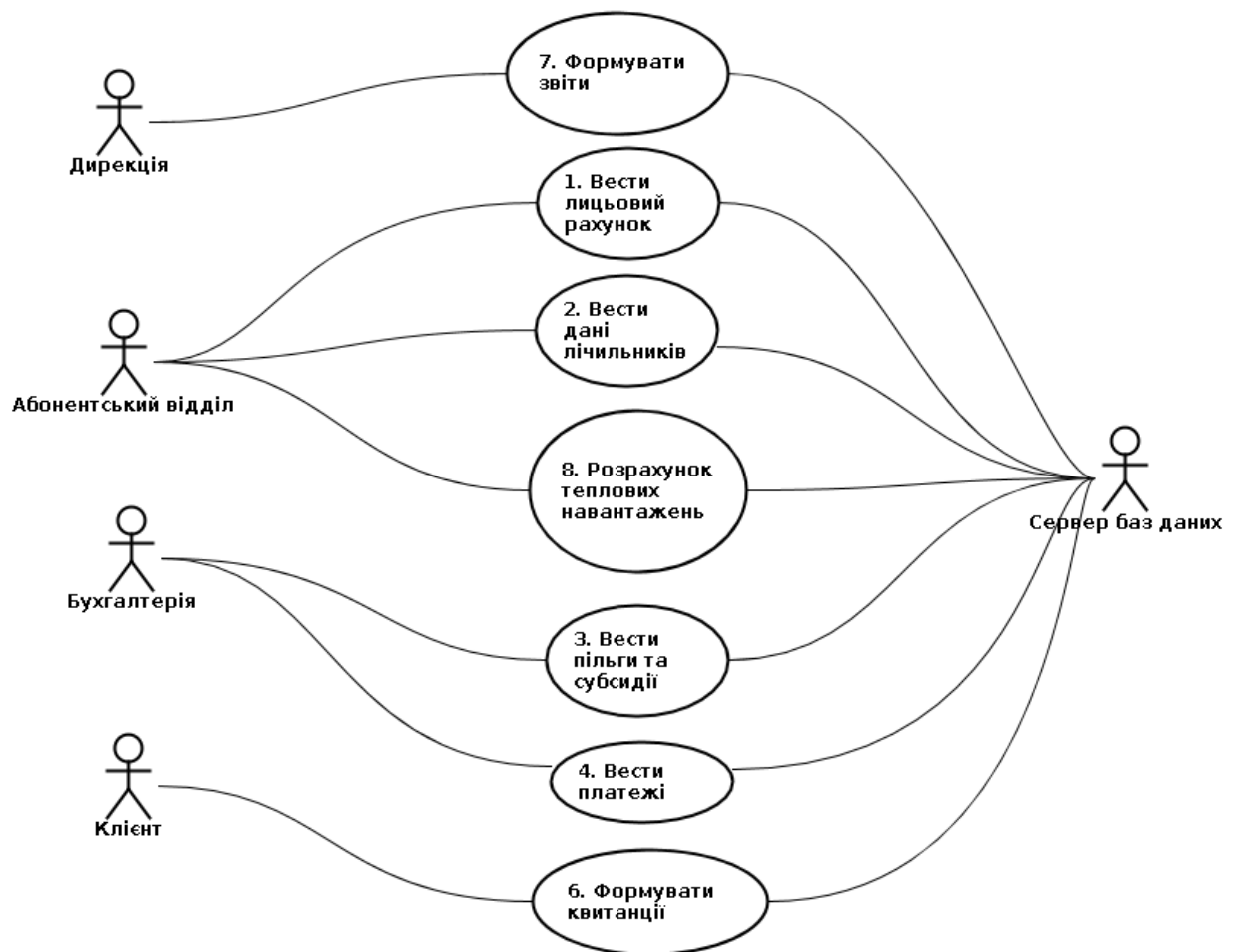


Рисунок 2.5. Модель потоків даних автоматизованої системи в частині ведення обліку з тепло- водо- каналізаційного господарства

Модель потоків даних підсистеми представлена на рисунку 2.8. Сервер баз даних та сервер додатків на моделі потоків даних є джерелами файлів для

створення резервних копій та відновлення резервних копій. Файловий архів для збереження копій на моделі не представлений – він має вестись окремо від сховища файлів системи. Це дозволить відновити робочу базу даних або копію архіву при пошкодженні одного з носіїв інформації – носія бази даних або архіву даних системи.

Як було зазначено вище, підсистема обліку розрахунків АІС ОСББ також має включати процеси з розрахунків з заробітної плати та ведення матеріального обліку, що є наслідком наявних на підприємстві механізмів розрахунків за комунальні послуги з працівниками організації, що винаймають службове житло, та взаєморозрахунків з юридичними особами. Ці процеси зазначені на моделях прецедентів бухгалтера (Рисунки 2.6, 2.7).

2.2 Рішення з інформаційного забезпечення

Підсистема «сервер баз даних і сервери додатків» повинна забезпечувати зберігання та обробку даних, що визначаються відповідно до функцій та структур даних системи в цілому. Не зважаючи на це, в частині забезпечення доступу до даних та вирішення інших задач адміністрування бази даних та програмного забезпечення сервера додатків підсистема «сервер баз даних і сервери додатків» повинна спиратись на власні дані, що визначають діяльність з забезпечення адміністрування.

Виходячи з аналізу предметної області та результатів розробки моделей випадків використання системи, а також після визначення вимог до змісту і структури даних підсистеми, які необхідно зберігати, можливо перейти до організації даних у інформаційному сховищі підсистеми.

Концептуальна (змістовна) модель – це абстрактна модель, що визначає структуру даних модельованої системи, властивості її елементів і причинно-наслідкові зв'язки, властиві системі і істотні з точки зору досягнення мети моделювання.

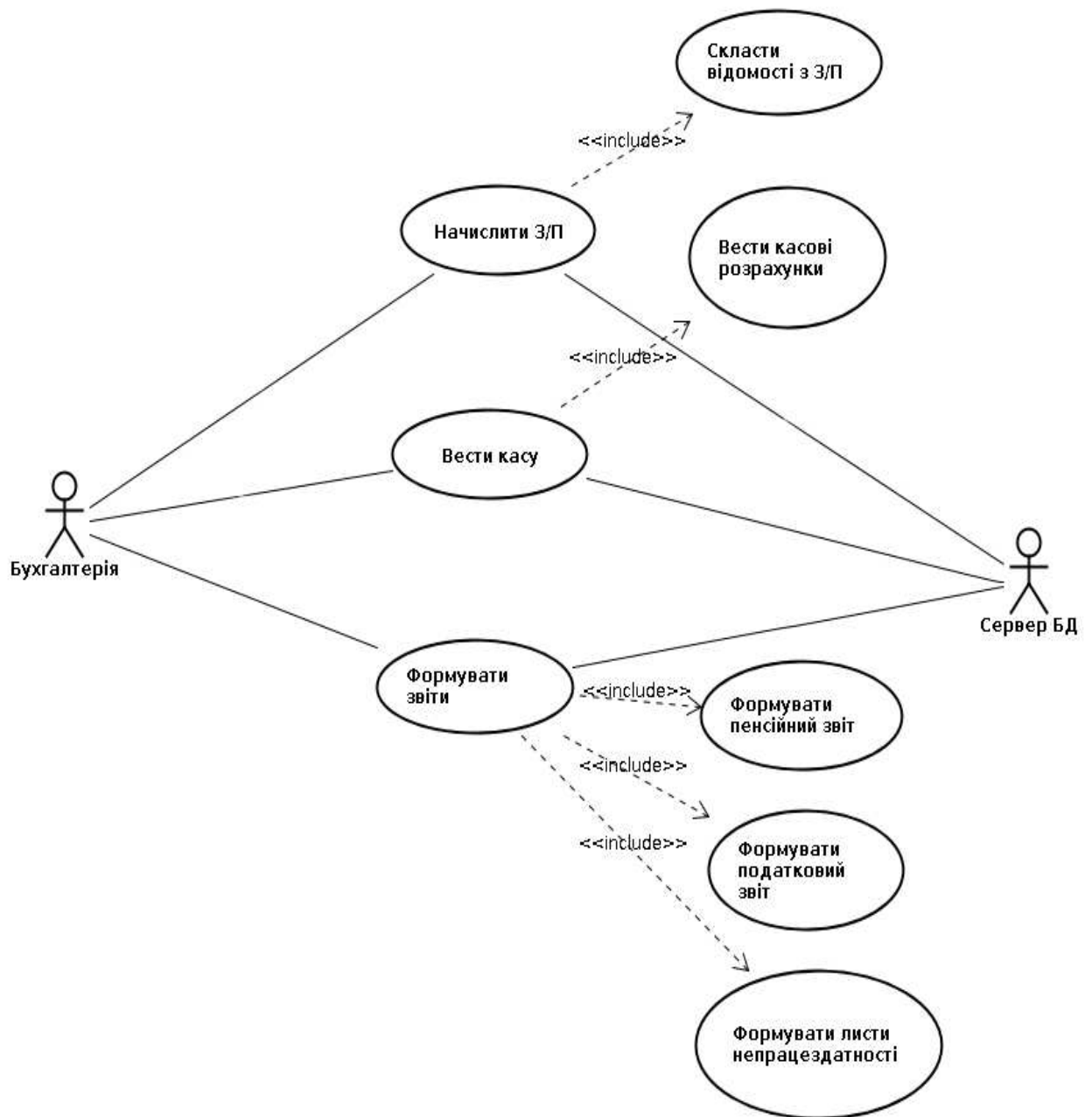


Рисунок 2.6. Модель прецедентів ведення розрахунків з заробітної плати підсистеми обліку розрахунків АІС ОСББ

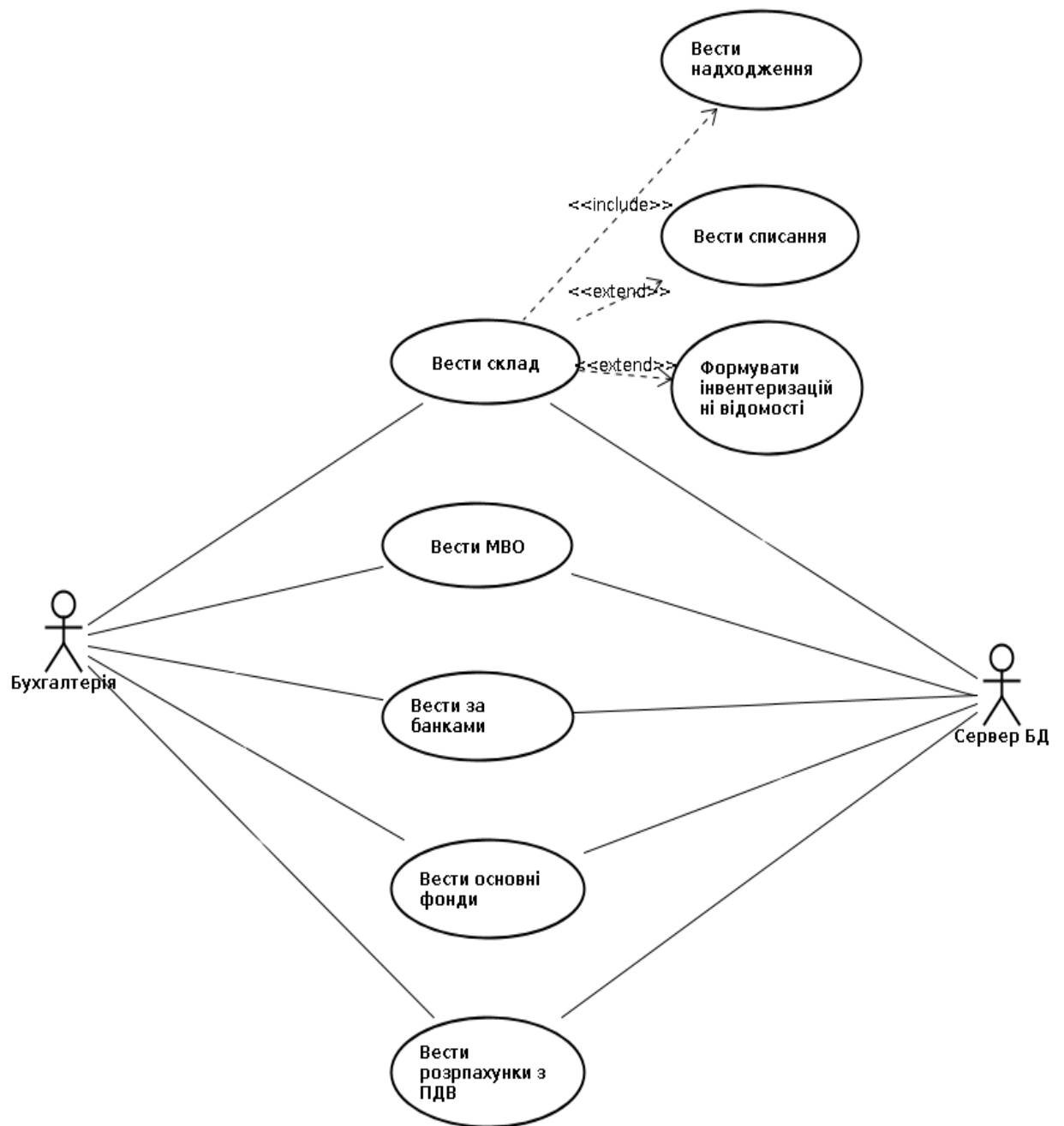


Рисунок 2.7. Модель прецедентів підсистем ведення обліку розрахунків за матеріалами підсистеми обліку розрахунків АІС ОСББ

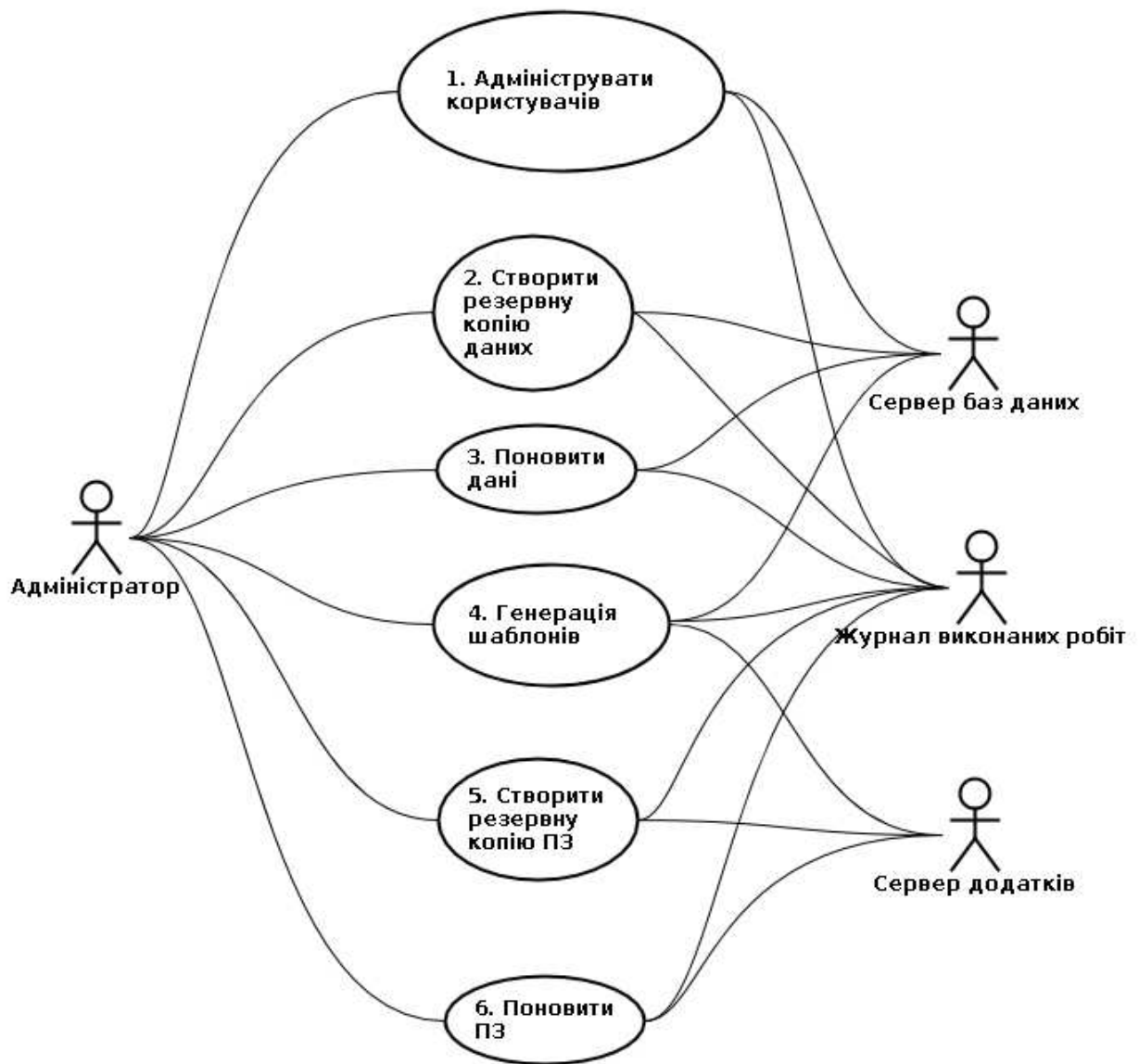


Рисунок 2.8. Модель потоків даних АРМ «Адміністратор системи» центру розрахунків АІС ОСББ

Концептуальна модель – це модель, представлена переліком понять предметної області і зв'язків між ними, що разом визначають семантичну структуру розглянутої предметної області або її конкретного об'єкта. Таким чином, концептуальна модель є моделлю предметної області, що складається з переліку взаємопов'язаних понять, які використовуються для опису цієї предметної області, разом з властивостями і характеристиками, що повністю

класифікують ці поняття за типами, ознаками в даній предметній області і відповідно до процесів в ній.

Відповідно до свого призначення ця підсистема повинна оперувати такими сутностями, як “користувачі системи”, “права доступу”, “види операцій з адміністрування”, “роботи”.

Модель даних підсистем «сервер баз даних і сервери додатків» та «Адміністратор системи» представлена на рисунках 2.9, 2.10, 2.11 відповідно. Атрибут “id” є ідентифікуючим, атрибут “description” – документаційним. Потужність асоціативних зв’язків між сутностями позначена на зв’язках, символ “*” визначає потужність як “багато”.

Враховуючи, що функції адміністратора системи є невід’ємними від підсистеми «сервер баз даних і сервери додатків», модель даних підсистеми Автоматизоване робоче місце «Адміністратор системи» була розглянута вище.

З метою забезпечення механізмів документування виконаних адміністратором операцій при обслуговуванні баз даних системи є доцільним створення окремого, незалежного від сервера баз даних та сервера додатків засобу ведення журналу. Для цього пропонується використання текстового файлу, розміщеного по за межами бази даних, що дозволить вести журналювання виконаних над сховищем операцій з створення резервних копій та поновлення бази даних з копій, видалення тимчасових робочих наборів даних тощо.

Модель даних підсистеми представлена на рисунку 2.11, сутностями моделі є таблиці, що зберігають інформацію про користувачів, види та роботи з адміністрування системи, пов’язані з роботами ресурси системи (файли, комп’ютери, бази даних тощо).

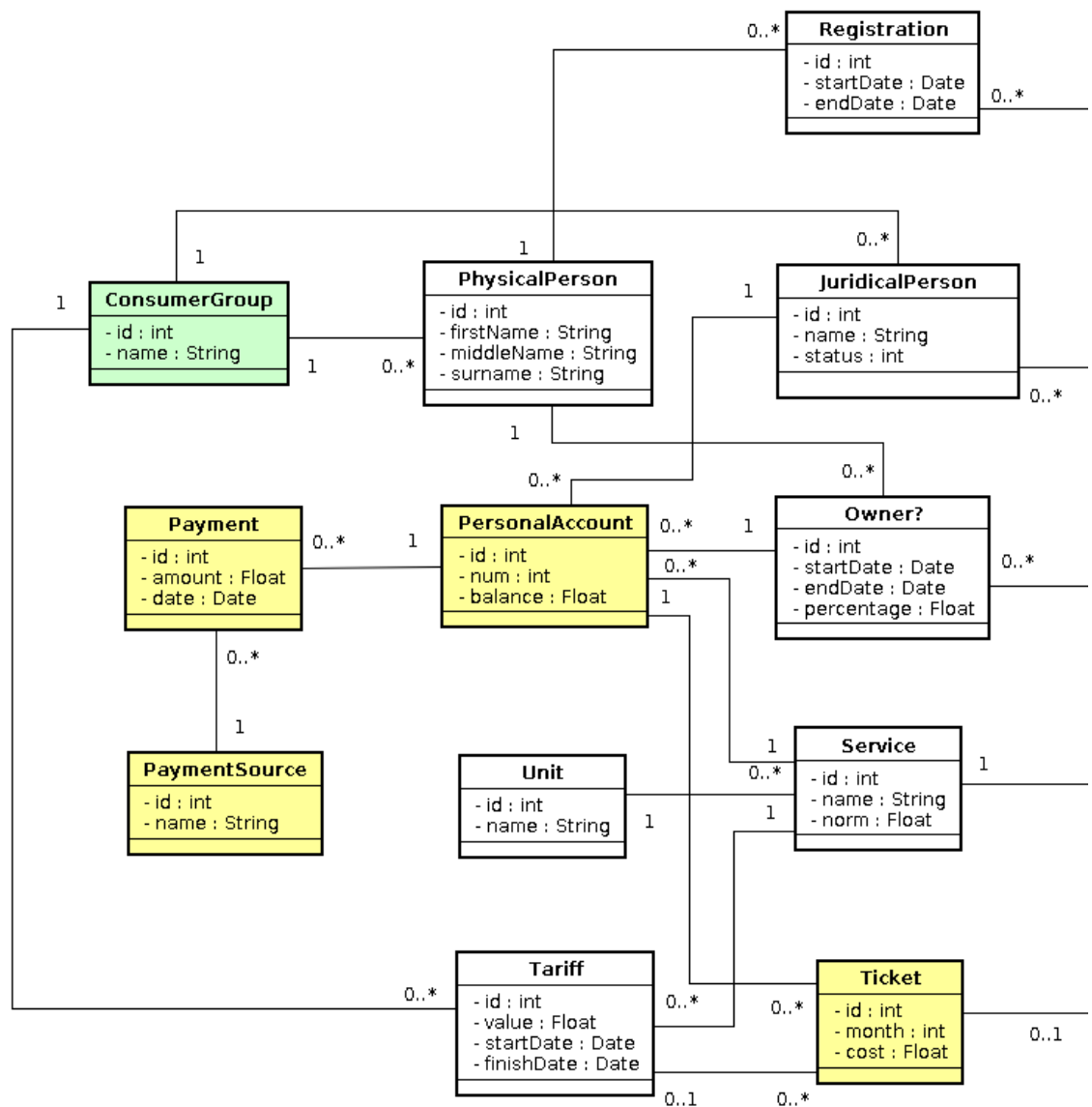


Рисунок 2.9. Модель даних підсистеми обліку розрахунків АІС ОСББ в частині ведення обліку з тепло- водо- каналізаційного господарства (фрагмент загальної моделі)

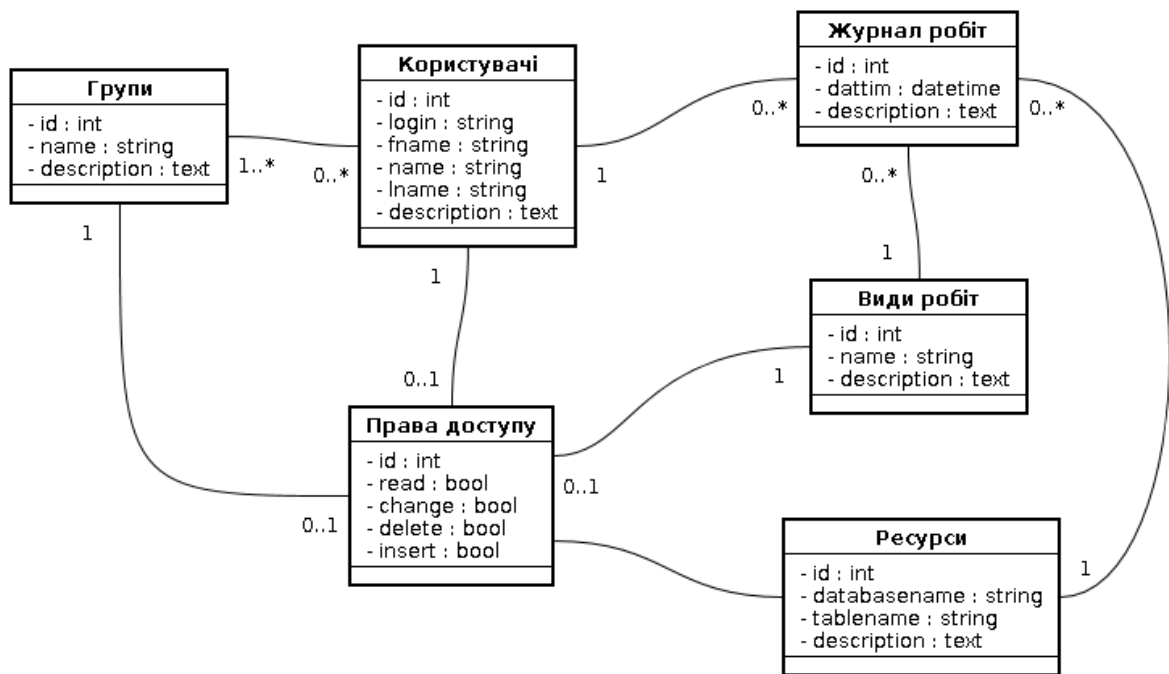


Рисунок 2.11. Модель даних підсистем «сервер баз даних і сервери додатків» та «Адміністратор системи»

2.3 Рішення з програмного забезпечення

2.3.1 Вибір технології організації інформаційного сховища системи

Суттєвою проблемою розробки та експлуатації (в тому числі – розширення) корпоративних систем є складність структур даних, які складаються в результаті розвитку таких систем. Як наслідок, розширення функціональності таких систем часто супроводжується змінами в структурах даних бази даних, що для складних систем є нетривіальною задачею і часто потребує суттєвої переробки моделей систем, які використовують дані, структура яких зазнає змін.

Складність вирішення задачі реорганізації даних в системах з реляційною базою даних (яких більшість) тим значніше зростає в наслідок невідповідності методів організації даних в програмах, що використовують базу даних – вони звичайно розробляються відповідно до методологій об’єктно-орієнтованого програмування – та методів організації даних в

необ'єктних сховищах даних, що загалом призводить до необхідності виконання над базами даних вкрай складних запитів. Розробка таких запитів на складних структурах даних є невиправдано трудомісткою задачею та, в окремих випадках, вимагає створення допоміжних таблиць даних (*Temporary tables*) або видів (*Views*), що, в свою чергу, також ускладнює роботу з базою даних.

Зазначена проблема могла б бути подолана за рахунок використання об'єктно-орієнтованих систем управління базами даних (СУБД), які б дозволили повною мірою задовольнити потреби щодо організації даних з боку об'єктно-орієнтованого програмного забезпечення, але, на жаль, об'єктно-орієнтовані СУБД на сьогодні не отримали широкого розповсюдження – найперше, через свою значну складність та надмірну вартість.

З метою подолання означених проблем останнім часом широко використовують об'єктно-орієнтовані надбудови над реляційними базами даних (*ORM, Object Relational Mapping*), які забезпечують відповідність організації даних на рівні програми та на рівні структур даних. Використання *ORM* надає розробнику можливість маніпулювати даними на рівні об'єктів та абстрагуватись від особливостей реалізації структур даних на конкретному сервері баз даних (рисунки 2.12). Це також дозволяє відносно простими засобами отримувати зрозумілий і якісний об'єктно-орієнтований код та спрощує його тестування модифікацію та супровід.

Деякі пакети вирішують цю проблему, надаючи бібліотеки класів, здатних виконувати перетворення даних автоматично. Маючи список таблиць в базі даних і об'єктів у програмі, вони автоматично перетворюють запити з одного виду в інший. У результаті запиту генеруються і виконуються необхідні запити, а результати перетворюються в об'єкти відповідного типу.

З точки зору програміста *ORM*-система повинна виглядати як постійне сховище об'єктів. На практиці всі *ORM*-системи, зазвичай, виявляють себе саме такими в тому чи іншому вигляді, забезпечуючи, таким чином,

можливість ігнорування реляційної природи бази даних як набору таблиць та відношень між ними.

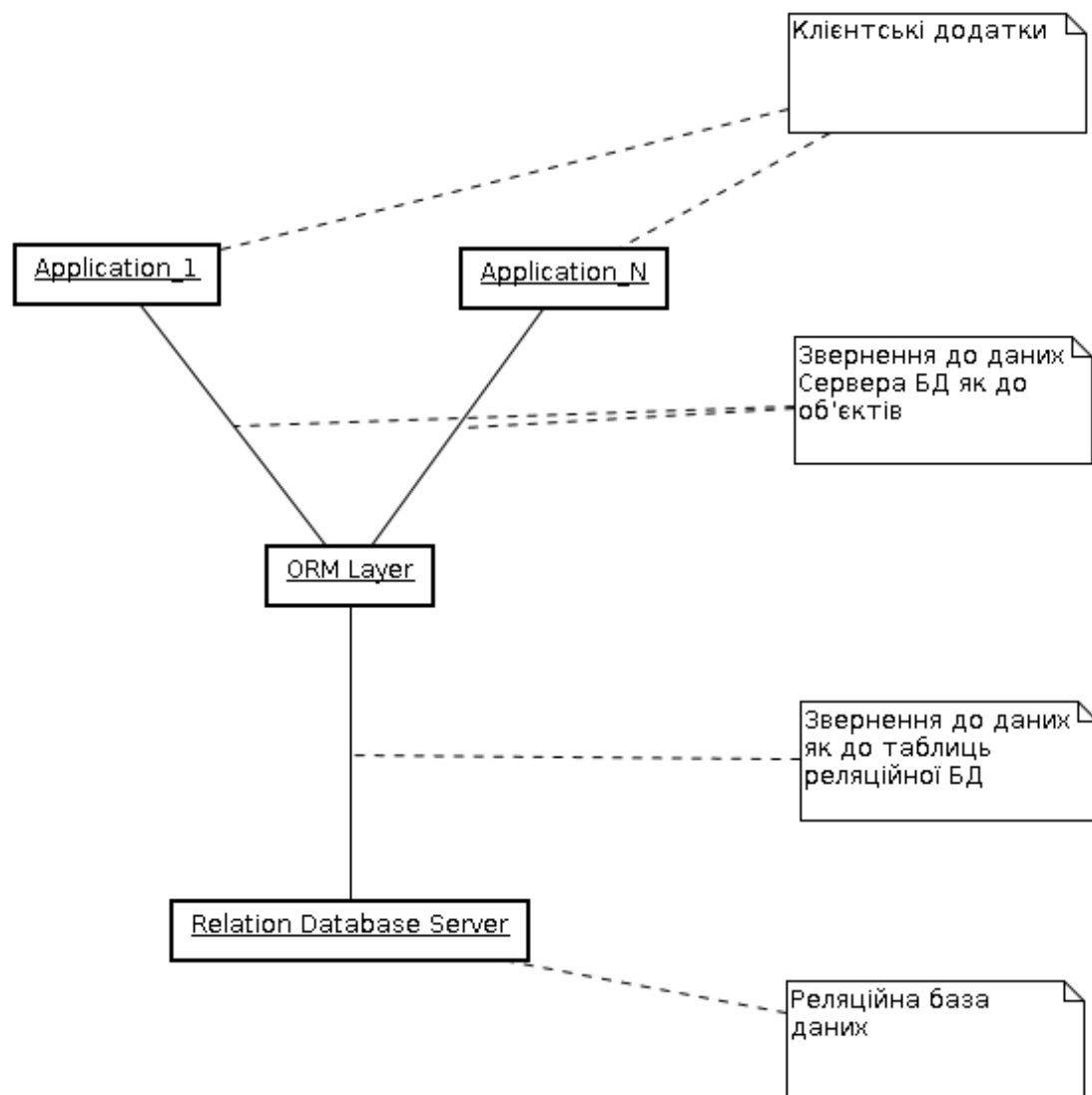


Рисунок 2.12. Компонентна модель доступу до реляційної бази даних з Object Relational Mapping (ORM)

Враховуючи, що програмне забезпечення для складних систем звичайно розробляється відповідно до принципів об'єктно-орієнтованого програмування, необхідність використання *ORM*-систем виникає в наслідок невідповідності способів організації структурованих даних реляційних систем управління базами даних способам організації об'єктів як

структурованих даних та механізмів їх обробки в програмах, які використовують такі бази даних. Ця невідповідність полягає в тому, що об'єкти програми можуть знаходитись у відношеннях агрегації, асоціації та використання, а реляційні бази даних підтримують лише відношення асоціації, причому, відношення асоціації багато-до-багатьох реляційними базами даних не підтримується.

Використання *ORM*-технології має і негативні наслідки, насамперед, через деяке зменшення пропускнуєї спроможності сервера баз даних. Більш того, також і шар транзакцій при цьому виявляється більш повільним і неефективним (особливо в термінах згенерованого запиту, записаного мовою запитів – *SQL*).

Взагалі, для проєктів, які характеризуються високим навантаженням на сховища даних, питання отримання необхідної продуктивності гостро стоїть і без застосування *ORM*. Але для таких систем також мають велике значення і отримання відносно простих можливостей розширення системи, а саме такі можливості і забезпечуються використанням *ORM*.

Таким чином, використання *ORM* може призвести до того, що програми будуть працювати повільніше і використовувати більше пам'яті, ніж програми, написані «вручну», але *ORM* позбавляє розробника необхідності написання великої кількості коду, часто одноманітного і схильного до помилок, тим самим значно підвищуючи швидкість розробки. Крім того, більшість сучасних реалізацій *ORM* дозволяють жорстко визначати код *SQL* запитів, який буде використовуватися при тих чи інших діях (збереження в базу даних, завантаження, пошук тощо) з постійним об'єктом.

Впровадження *ORM* звичайно призводить до деградації продуктивності системи на близько 30%.

Маючи на увазі швидкий прогрес в підвищенні продуктивності обчислювальних систем (близько 30% – 50% на рік) та тенденції до зменшення вартості обладнання, можна розраховувати на те, що протягом року недоліки системи, пов'язані з деградацією обчислень внаслідок використання *ORM* будуть повністю нівельовані за рахунок підвищення

продуктивності апаратного забезпечення. При цьому переваги від впровадження *ORM*, пов'язані з можливостями більш простого розширення структур даних, залишаються в повній мірі і надалі.

2.3.2. Вибір мови програмування

Однією з важливих експлуатаційних характеристик інформаційної системи є її продуктивність. Продуктивність комп'ютерної інформаційної системи визначається, в тому числі, наступними факторами:

- технологіями та інструментальними засобами обробки даних, які використовуються в інформаційній системі;
- обсягом і складністю структур даних, що підлягають обробці;
- продуктивністю апаратного забезпечення системи.

Технології обробки даних напряму пов'язані з задачами, що підлягають автоматизації, та визначають можливість застосування тих чи інших алгоритмів обробки даних та механізмів доступу до даних. Перелік задач встановлюється Замовником відповідно до власних вимог і, з точки зору Виконавця (проектувальника системи), є незмінним. Відповідно до цього, Виконавець може обирати ті чи інші інструментальні засоби, які відповідають вирішуваним задачам та дозволяють досягти необхідних експлуатаційних характеристик системи. До таких засобів належать мова програмування, алгоритми, бібліотеки, системи управління базами даних, протоколи передачі даних тощо. Вибір тих чи інших інструментальних засобів досягається в результаті компромісу між простотою, потужністю, надійністю, швидкістю, захищеністю та іншими характеристиками цих засобів. Взагалі вважається, що складність інструментальних засобів та складність проєктованої системи знаходяться в прямій залежності – чим більш складна проєктована система, тим складнішими виявляються вимоги до інструментальних засобів, і тим складнішими виявляються самі інструментальні засоби.

Одним з ключових аспектів проєктування і експлуатації складних систем є плинність зовнішнього середовища. Стосовно до системи, що

проектується, ця плинність може виявлятися в змінах правил розрахунків – тобто в змінах бізнес-логіки. Одним з суттєвих проявів цих змін є те, що такі зміни можуть потребувати внесення нових правил розрахунків в програмне забезпечення системи – як наслідок, такі системи виявляються залежними від Розробника настільки, що їх експлуатація без постійного втручання Розробника є неможливою. Для запобігання такої залежності подібні системи реалізують в спосіб, що дозволяє відокремити бізнес-логіку від проекрованої системи, що досягається шляхом перетворення правил розрахунку в дані, які можуть вноситись в систему користувачами системи нарівні з іншими даними системи.

Простою ілюстрацією подібних рішень є, наприклад, Microsoft Excel з пакету Microsoft Office, який дозволяє вносити в електронні таблиці формули для розрахунків нарівні з даними, які використовуються в цих формулах. Для забезпечення відносно простої можливості інтерпретації цих формул подібні інструменти виконуються з використанням інтерпретованих мов програмування, або такі мови в такі інструменти вбудовуються. Так, пакет Microsoft Office для забезпечення такої можливості широко використовує *Visual Basic Application (VBA)* – розширення до інтерпретованої мови *Visual Basic*. Серед подібних мов програмування останнім часом особливу популярність отримала високорівнева кросплатформна мова програмування *Python* (Пайтон).

Python – інтерпретована об'єктно-орієнтована мова програмування високого рівня з динамічною типізацією, розроблена в 1990 році. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднання наявних компонентів. *Python* підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор *Python* та стандартні бібліотеки доступні як у скомпільованій так і у вихідній формі на всіх основних платформах. В мові програмування *Python* підтримується декілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована. Наявність

декількох парадигм дозволяє широкою мірою охопити різні аспекти реалізації бізнес-логіки.

Однією з головних причин популярності Python є його простота у вивченні, яка базується на високій наближеності синтаксису цієї мови до інших поширених мов, в тому числі – мови запису формул в математиці. Крім того, ця мова дозволяє вирішувати складні завдання – ця можливість базується на великій кількості додаткових модулів та бібліотек. Наочність коду *Python* добре ілюструє хрестоматійний приклад обміну значень двох змінних a і b – операція, яка в *Python* записується в спосіб, наближений до природних мов:

$$b, a = a, b$$

Враховуючи корисність відокремлення бізнес-логіки від внутрішнього коду системи, така наочність Python видається вкрай важливою. Крім того, Python ліквідує проблеми, пов'язані з супроводом програмного коду і дозволяє викладати складні ідеї реалізації системи простими мовними конструкціями, що має сприяти підвищенню надійності коду та забезпеченню фактору розвитку системи. Простота програмного коду – істотний фактор, який полегшує подальше його супровід – синтаксис і загальні ідіоми цієї мови настільки ясні, короткі і прості, що дозволяють працювати з ним протягом тривалих періодів часу.

Продуктивність технічних засобів хоча і є важливим фактором, але для більшості задач вдалий вибір алгоритмів обробки часто дозволяє отримати вигреш в декілька порядків, що важливо в порівнянні з виграшем за рахунок збільшення потужності обчислювальної системи, яке для рядових систем звичайно складає всього один порядок при еквівалентному зростанні ціни системи.

Наступним фактором, що суттєво впливає на продуктивність інформаційної системи є складність структур даних, що підлягають обробці. Так само, як і вибір алгоритмів обробки даних, це питання значною мірою знаходиться в частині питань Розробника системи і напряду пов'язане з самими алгоритмами. Зважаючи, що обсяг даних визначається Замовником,

Розробник може обирати для обробки такі фрагменти даних, які найкраще відповідають алгоритмам та технічному забезпеченню. Фрагментація даних може виконуватись на рівні їх поділу на окремі файли та/або мережеві пакети. Враховуючи взаємозалежність структур даних та алгоритмів, питання організації, структурування та форматів даних має бути розглянуте більш повно.

2.3.3 Визначення вимог щодо форматів файлів даних підсистеми обліку розрахунків АІС ОСББ

Вимоги щодо форматів даних автоматизованої системи можуть бути визначені виходячи з наступних аспектів: типів, структур та розмірів. Виходячи з того, що з точки зору операційної системи окремий фрагмент даних є файлом, а поняття типів, структур та розмірів цілком охоплюються файловим представленням даних в системі, розглянемо критерії та вимоги до організації даних в файли з метою їх окремої (самостійної) обробки та транспортування. Взагалі, обробка і транспортування даних не можуть відбуватись окремо, тому при визначенні продуктивності систем застосовують комплексні тести, які охоплюють читання даних з файлів, завантаження в пам'ять, передачу тощо. Найбільше простим тестом є проста передача даних між машинами в мережі таким чином, щоб обробці підлягала не швидкість передачі даних, а затримки на їх обробку – латентність технічного і програмного забезпечення системи.

2.3.4. Фізична модель даних

Враховуючи те, що в системі використовується ORM (object-relational mapping, об'єктно-реляційно проєкція), що забезпечує зв'язок бази даних з концепціями об'єктно-орієнтованої мови програмування, фізична модель даних не розроблялась – замість неї були реалізовані ORM-моделі: Документ, Користувач, Група, Персонал, Звання, Посада, Структура документа, Тип документа, Дисципліна.

В зв'язку з тим, що АРМ є лише частиною наявної інформаційної системи ОСББ, а в системі є доцільним використання єдиної бази даних, наведемо опис таблиць бази даних також з врахуванням таблиць, що вже існують в базі даних інформаційної системи (в складі інших АРМ).

Опис кожної з таблиць фізичної моделі представлений в табличному вигляді і наведений в Додатку Б.

2.4 Рішення з математичного забезпечення

З метою визначення вимог щодо вибору структур та фрагментування даних бази даних автоматизованої інформаційної системи та враховуючи необхідність інтеграції розроблюваної підсистеми обліку з іншими АІС розглянемо вимоги до засобів обміну даними між підсистемою та іншими системами. Зважаючи на можливі значні обсяги даних в базі даних системи бажано обрати “прозорий” протокол передачі даних – такий, що з одного боку дозволить вирішувати задачі транспортування, а з іншого – вибору сервісів обробки. Саме такі протоколи часто є основою для корпоративних систем обробки та зберігання даних. Найбільше поширеним серед подібних протоколів є протокол *RPC (Remote Procedure Call – Віддалений Виклик Процедур)*.

Ідея виклику віддалених процедур полягає в розширенні механізму передачі управління і даних в програмі, що виконується на одній машині, на передачу управління і даних між декількома програмами, що виконуються на різних комп'ютерах мережі. Засоби віддаленого виклику процедур призначені для полегшення організації розподілених обчислень і створення розподілених клієнт-серверних інформаційних систем – саме таких, як система, що підлягає проектуванню. Найбільша ефективність використання *RPC* досягається в тих додатках, в яких існує інтерактивний зв'язок між віддаленими компонентами з невеликим часом відповідей і відносно малою кількістю переданих даних. Такі додатки називаються *RPC-орієнтованими*. Як наслідок, на відміну від локального виклику, віддалений виклик процедур обов'язково використовує транспортний рівень мережевої архітектури

(наприклад *TCP* – *Transport Control Protocol*), проте це залишається прихованим від розробника.

Іншими характерними рисами виклику віддалених процедур є асиметричність взаємодії, при якій одна зі сторін взаємодії є ініціатором передачі, а інша – обробником запитів. Крім цього, сама передача даних вимагає відкриття з'єднання і, як наслідок, призводить до додаткових витрат часу та зростання латентності каналу передачі даних.

Існує також ряд проблем, пов'язаних з неоднорідністю мов програмування і операційних систем: структури даних і структури виклику процедур, підтримувані в деякій мові програмування, можуть не підтримуватися в інших мовах. Це призводить до виникнення відомої проблеми сумісності також і на рівні конвертації даних, що передаються між подібними системами.

Як впливає з загального опису технологій застосування *RPC* є орієнтованою на додатки з невеликим часом відповідей і відносно малою кількістю переданих даних. При цьому для визначення меж застосування таких технологій в корпоративних системах важливо визначити значення часу відповіді і кількості переданих даних, при яких можливості застосування цих технологій все ще залишаються прийнятними з точки зору швидкості передачі даних і латентності каналу передачі даних і інформаційної системи в цілому.

Для проведення дослідження засобами мови програмування Python були реалізовані *XML-RPC* клієнт і сервер. Спрощена версія коду клієнта і сервера (з метою демонстрації основної функціональності коду, а не деталей реалізації) наведені в додатках А і Б відповідно.

Інтерфейс сервера визначений як набір функцій простої обробки тексту, яка полягає в конкатенації (об'єднанні) рядків. Такий досить простий тест, насправді, є дещо наближеним до реальних задач генерації документів, оскільки значна частина таких задач полягає не в обчисленнях, а в обробці тексту [5, 6].

Експеримент проводився в конфігурації локальної мережі, яка включала комутатор і три комп'ютера, поєднані каналом з пропускнуою спроможністю 100 МБіт. Отримані результати, що визначають залежність відносного часу передачі одного байта від розміру пакету даних (відносна латентність каналу передачі даних), представлені на графіках нижче (рисунки 2.13 – 2.16).

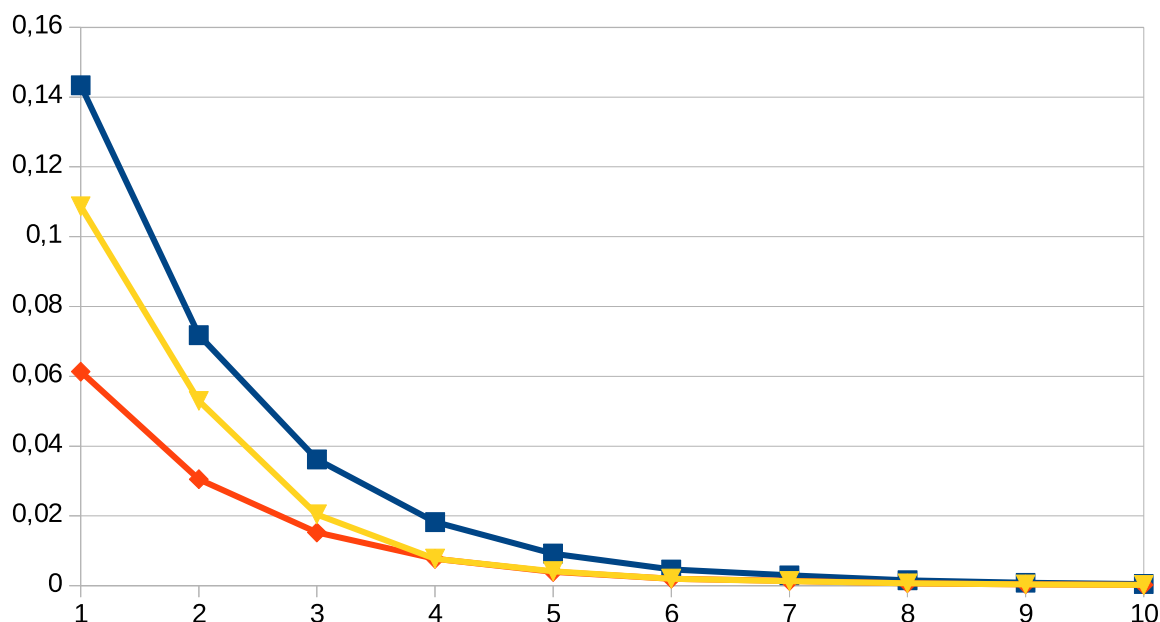
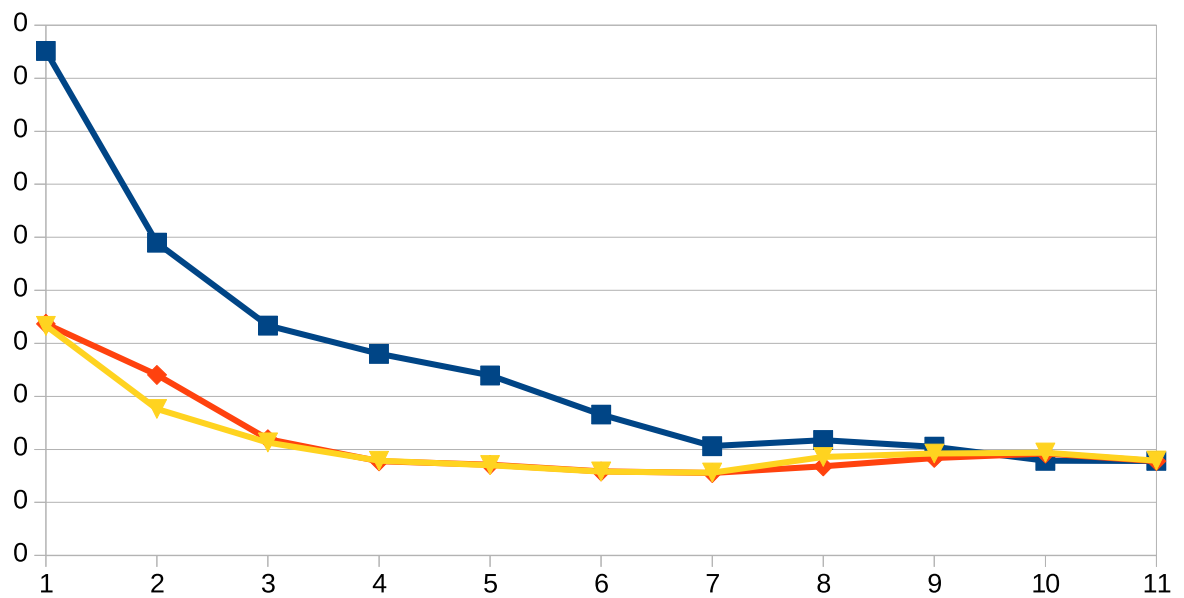


Рисунок 2.13. Відносна латентність каналу передачі даних для малих пакетів даних (розмір пакету $10 \cdot 2^x$)

Як видно з результатів проведеного дослідження, відносна латентність каналу передачі даних для малих пакетів зменшується зі збільшенням розміру пакету і це має місце для пакетів розміром від десятків байт і до 2-4 Кбайт; для пакетів розміром від 10 Кбайт і до 10 Мбайт відносна латентність каналу передачі даних стає практично лінійною з тенденцією до зменшення зі збільшенням розміру пакета. Таким чином, можна визначити, що найбільше прийнятними з точки продуктивності системи є вибір розміру файлів в діапазоні 10 кбайт – 10 Мбайт.



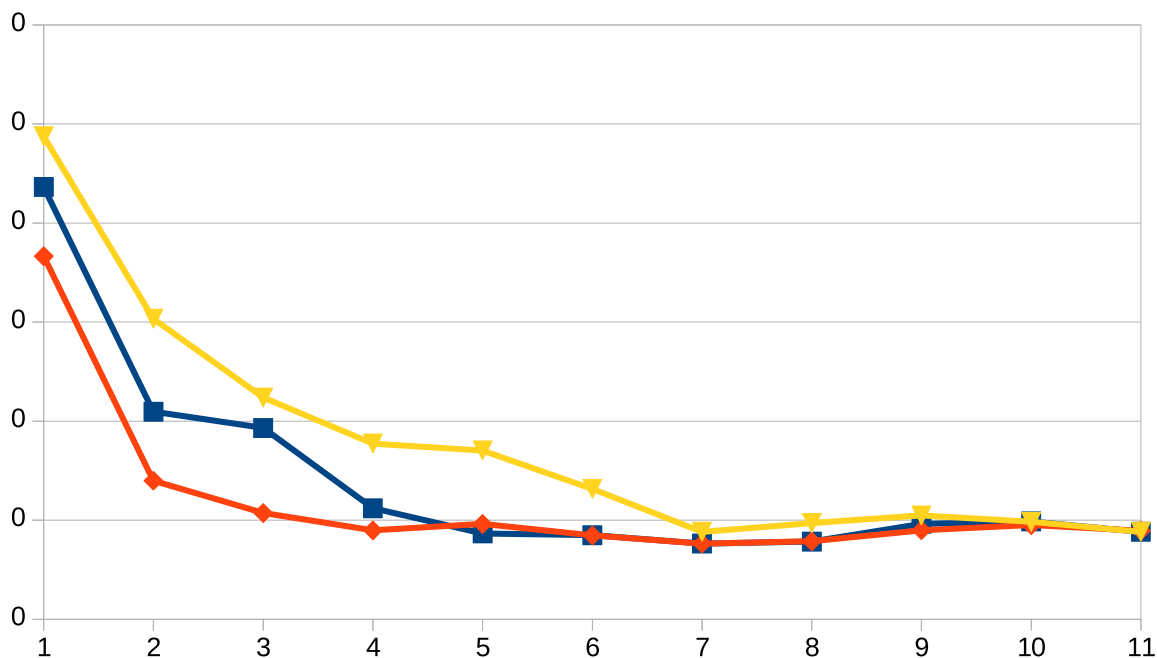


Рисунок 2.16 – Відносна латентність каналу передачі даних для малих пакетів з повторними підключеннями до сервера (розмір пакету $1000 \times 2^{**x}$)

2.5 Рішення з технічного забезпечення

Технічне забезпечення автоматизованої системи організації складається з сервера баз даних, сервера додатків та клієнтських робочих місць, які приєднано до локальної мережі організації. Враховуючи корпоративну важливість забезпечення цілісності інформації в базі даних та безперервності здійснення обчислень з тарифікації послуг, а також з метою забезпечення необхідних параметрів надійності роботи системи, сервер системи повинен працювати під керівництвом операційної системи Linux як більш стійкої до відмов і атак. Окрім власне операційної системи на сервері також має бути встановлене наступне системне програмне забезпечення: сервер баз даних MySQL та веб-сервер Apache.

На ранніх етапах експлуатації системи, коли об'єм накопичених даних буде незначним, система може використовувати побутові персональні комп'ютери, в подальшому за результатами стрес-тестування вимоги до обладнання серверів можуть бути уточнені.

Відповідно до попередньої оцінки інформаційного навантаження на систему, для функціонування серверної частини системи необхідне обладнання з такими мінімальними характеристиками:

- Апаратні, не нижче:
 - ЦП Intel-Core i5;
 - 2Гб ОЗУ;
 - Жорсткий диск з 20 Гб вільного простору.
- Програмні, не нижче:
 - ОС Ubuntu Server 25.04 LTS (як стабільна версія системи);
 - Веб-сервер Apache з підтримкою mod_python;
 - Сервер баз даних MySQL версії 5.

Для функціонування клієнтської сторони необхідно наявність у клієнтів з'єднання з локальною мережею ОСББ і наявність веб-браузера на стаціонарному комп'ютері, ноутбукі або мобільному пристрої. В разі розміщення Веб-сервера підсистеми у глобальній мережі Інтернет клієнтам буде достатньо мати підключення до мережі Інтернет.

2.6. Рішення з організаційного забезпечення

В результаті проєктування підсистеми обліку розрахунків АІС ОСББ можна зробити висновок про те, що структура наявної в організації автоматизованої інформаційної системи зазнає суттєвих змін. Тим не менше, робочі форми підсистеми обліку розрахунків були, в цілому, збережені (їх вміст визначається не функціональністю або технологіями інформаційної системи, а виключно нормативними документами з ведення розрахунків та бухгалтерського обліку). Таким чином, змін в роботі персоналу з обліку розрахунків АІС ОСББ не планується.

Для забезпечення можливостей ефективної роботи системи адміністратор системи має ознайомитись з проєктною документацією на підсистему та пройти підготовку з основ роботи та адміністрування ОС Linux, Веб-сервера Apache та СКБД MySQL (пакет LAMP — Linux, Apache, MySQL, Python).

2.7. Висновки до розділу 2

В розділі представлені проєктні рішення, розроблені відповідно до Технічного завдання на створення підсистеми обліку розрахунків АІС ОСББ. Ці рішення, загалом, містять наступне:

- загальносистемні рішення;
- рішення з організаційного забезпечення;
- рішення з інформаційного забезпечення;
- рішення з технічного забезпечення;
- рішення з програмного забезпечення.

При виборі інструментальних засобів зроблено акцент на використанні вільного системного та прикладного програмного забезпечення на базі ОС GNU Linux і відомого як LAMP (Linux, Apache, MySQL, Python).

3. МОДЕЛЬ РОЗГОРТАННЯ ПІДСИСТЕМИ ОБЛІКУ РОЗРАХУНКІВ АІС ОСББ

3.1. Особливості розгортання підсистеми обліку розрахунків

Відповідно до розробленої концепції підсистеми обліку розрахунків автоматизованої інформаційної системи (рисунок 1.6) в частині ведення розрахунків та обліку з електро, тепло-, водо- та каналізаційного господарства, була розроблена діаграма розгортання, виконана в нотації моделі кооперації UML, яка представлена на рисунку 3.1.

З рисунку видно, що розгортання підсистеми обліку розрахунків АІС ОСББ передбачає створення також додаткових підсистем автоматизованої системи (АРМ Бухгалтера, АРМ адміністратора, підсистеми обробки та збереження інформації). АРМ Бухгалтера забезпечує роботу підсистеми ведення розрахунків “Абонвідділ”. Підсистема збереження даних та АРМ адміністратора (підсистема адміністрування) є допоміжними.

На рисунку 3.2 представлено приклад екранної форми АРМ бухгалтера. У формі відображаються реальні дані про абонентів ОСББ, тому колонки з персональними даними (прізвища та адреси) “стиснуті”. Персональні дані клієнтів ОСББ на рисунку приховані.

3.2. Висновки до розділу 3

В результаті аналізу інформаційної системи організації визначено недоліки наявної системи та обґрунтовано доцільність розробки проєкту нової відкритої інтегрованої автоматизованої інформаційної системи організації на базі нових інформаційних технологій, технологій проєктування та програмування.

Визначено, що провадження автоматизованої системи організації дозволить за рахунок підвищення дисципліни розрахунків забезпечити підвищення якості наданих комунальних послуг та дійти зменшення заборгованості за отримані послуги.

В частині прийняття проєктних рішень розроблено моделі ескізного проєкту підсистем “Сервер баз даних і сервери додатків” та “Адміністратор

системи”. Передбачені можливості ведення журналу операцій над файлами системи.

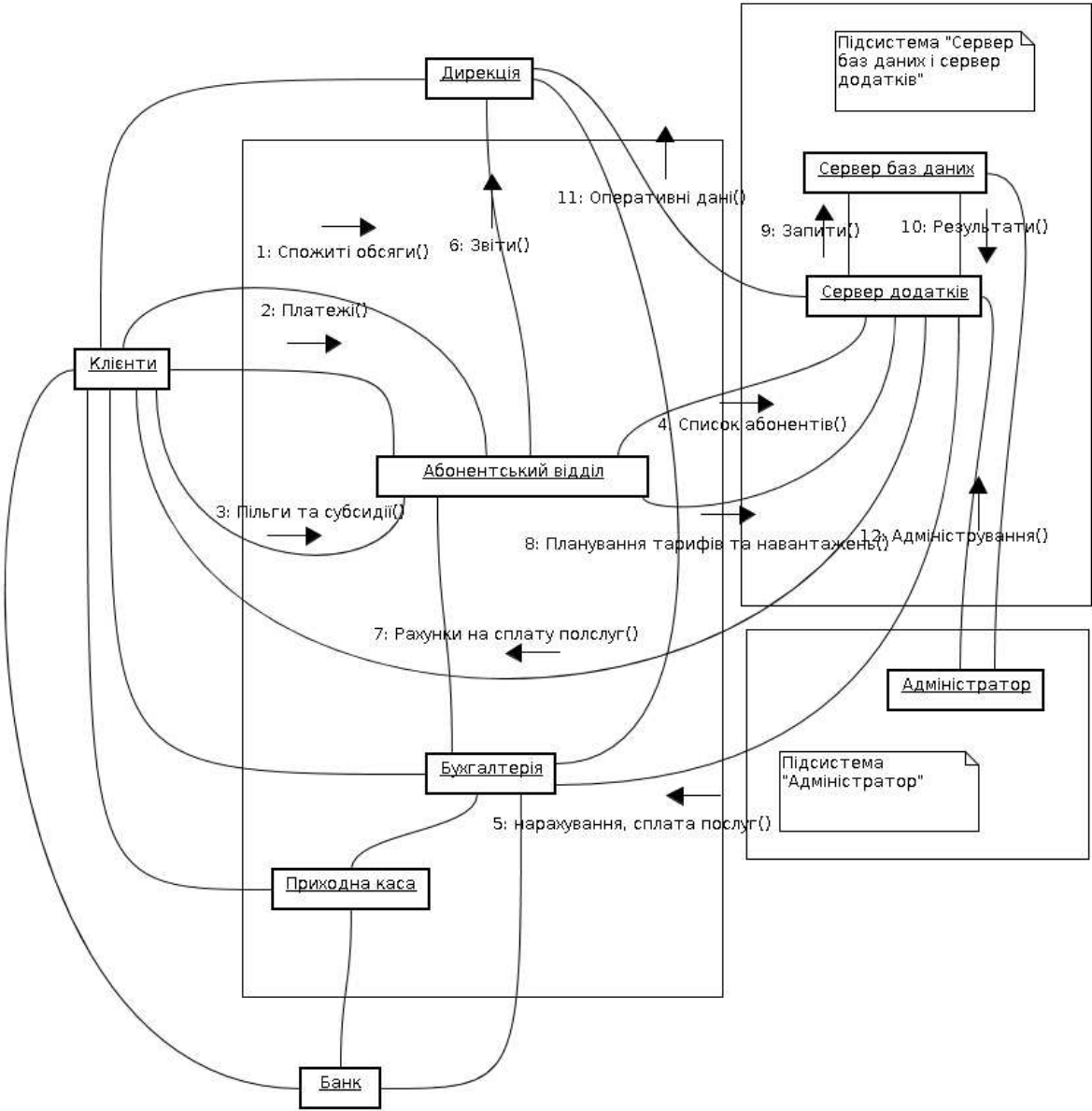


Рисунок 3.1. Модель розгортання підсистеми обліку розрахунків АІС ОСББ

MainWindow

File Tool Help Exit

Фіз. особи Юр. особи Розрахунки Прилади обліку Житловий фонд Послуги

Абоненти (Фіз. особи) Категорії абонентів Фізичні особи Рахунки

Rec: 1 100 / 15886; Fil: None;

	ІНП	ПІБ	Адреса	да	дз	Вода	Отп.	Заг.пл.	Опл.пл.	Прожив.	Пільгов.
1						061242	561242	0.000000	0.000000	1	0
2	2191...					808000	508000	71.300000	69.600000	2	0
3	2191...					810272	510272	68.000000	64.200000	4	0
4	2191...					811629	511629	50.500000	49.600000	2	0
5	2191...					813482	513482	82.400000	80.500000	0	0
6	2191...					011974	511974	68.000000	66.100000	7	4
7	2190...					806845	506845	34.400000	33.400000	1	0
8	2190...					801124	501124	38.500000	37.500000	0	0
9	2191...					811016	511016	60.400000	60.400000	2	0
10	2190...					806678	506678	66.600000	65.200000	1	0
11	2191...					812804	512804	51.000000	49.300000	3	0
12	2191...					812906	512906	68.300000	66.400000	2	0
13	2190...					031106	531106	22.200000	22.000000	1	0
14	2190...					803429	503429	35.500000	34.400000	1	1
15	2190...					801732	501732	51.300000	49.500000	3	0
16	2190...					801801	501801	53.700000	52.500000	2	0
17	2190...					805004	505004	69.500000	67.600000	1	0
18	2190...					801625	501625	49.900000	48.100000	0	0
19	2190...					816037	516037	66.000000	64.200000	5	2
20	2191...					808234	508234	48.900000	47.200000	2	0
21	2190...					816606	516606	45.100000	43.300000	4	0
22	2191...					808036	508036	67.900000	65.200000	4	4

Ready

Рисунок 3.2. Приклад екранної форми в частині ведення абонентів та обліку розрахунків за комунальні послуги АРМ бухгалтера АІС ОСББ

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

В результаті виконання даної дипломної роботи було розроблено АРМ менеджера програмних проєктів АІС ІТ-компанії.

Програмне забезпечення АРМ системи призначене для підвищення ефективності операційних процесів працівників ІТ-компанії та для підвищення якості процесів інформаційного обміну та комунікацій користувачів системи в частині обліку розрахунків за житлово-комунальні послуги шляхом автоматизації процесів ведення обліку надходжень за надані послуги та витрат на поточні заплановані або незаплановані роботи, визначення боржників і планування фінансових потоків.

Програмне забезпечення підсистеми розроблене мовою програмування Python з використанням фреймворку Qt 5.

Розроблене програмне забезпечення задовольняє вимогам, які поставлені у технічному завданні.

Список використаних джерел

1. Бардус, І.О., Лазарев, М.І., Ніценко, А.О. Бази даних у схемах – Харків: Видавництво ”Діса плюс”, 2017. – 146 с.
2. Вайсман, В.О. Моделі, методи та механізми створення і функціонування проєктно-керованої організації: Монографія [Текст] / В.О. Вайсман – К. : Наук. світ, 2009. – 146 с.
3. Гайна, Г.А. Основи проєктування баз даних / Навчальний посібник – К. : Кондор, 2024. – 280 с.
4. Заверач, М.М. Бази даних. Інформаційні системи [Текст]: навч. посіб. / М. М. Заверач,, В. В Третько. – Хмельницький : ХНУ, 2007, – 303 с.
5. Закон України “Про об’єднання співвласників багатоквартирного будинку” / Відомості Верховної Ради України, 2002, № 10, с. 78
6. Зарицька О. Л. Бази даних та інформаційні системи: методичний посібник. – Житомир: Вид-во ЖДУ ім. І. Франка, 2010. – 132 с.
7. Комунальне_підприємство // Wikipedia // Електронний ресурс. URL: https://uk.wikipedia.org/wiki/Комунальне_підприємство
8. Організація реляційних баз даних : навч. посіб. / Я.Г. Куваєв, О.А. Жукова, І.А. Сечкін – 2-ге вид., допов. та переробл. – Дніпро : НГУ, 2017. – 157 с.
9. Пасічник, В.В., Литвин, В.В., Шаховська, Н.Б. Проектування інформаційних систем. Навчальний посібник (затв. МОН України) – Львів: 2013. – 380 с.
10. Прохоров, Б. П’ять міфів про комунальні організації // Електронний ресурс. URL: <https://ces.org.ua/myths-about-utilities/>
11. Томашевський О.М. Інформаційні технології та моделювання бізнес-процесів / Навчальний посібник / О.М.Томашевський, Цегелик М.Б., Вітер Г.Г., В.І.Дубук. К.: Центр учбової літератури, 2005.– 296 с.
12. Цибко, Г. Ю., Рамський, Ю. С. Проектування й опрацювання баз даних : Посібник для вчителів. 2-ге вид., перероб. і доп. Тернопіль : Навчальна книга. Богдан, 2005.

13. Держава дає гроші ОСББ на сонячні панелі та теплові насоси. КИЇВ24 | Телеканал Київ. URL: <https://www.youtube.com/watch?v=Qqh-98ydkeU>
14. Bart Voorn, Marieke L. van Genugten, Sandra van Thiel. The efficiency and effectiveness of municipally owned corporations: a systematic review // Електронний ресурс. URL: <https://www.tandfonline.com/doi/full/10.1080/03003930.2017.1319360>
15. Unified Modeling Language User Guide, The (2 ed.). Addison-Wesley. 2005. p. 496.

ТЕХНІЧНЕ ЗАВДАННЯ

Розробка проєкту підсистеми обліку розрахунків АІС ОСББ

А.1. Назва та призначення автоматизованої системи

Підсистема обліку розрахунків АІС ОСББ (далі – “автоматизована система”, "система обліку", “автоматизована система обліку”) призначена для вирішення наступних задач:

- ведення обліку оплати фізичними та юридичними особами комунальних послуг, що надаються ОСББ;
- формування звітів по наданих та/або отриманих послугах по фізичним та юридичним особам ;
- формування звітів по оплачених послугах по споживачам послуг;
- формування звітів по виконаних роботах за роботами та виконавцями робіт;
- надання споживачам послуг довідкової інформації та інформації про стан рахунку.

А.2. Мета створення проєкту підсистеми

Мета створення проєкту підсистеми обліку розрахунків автоматизованої системи ОСББ: підвищення ефективності процесів управління ОСББ в частині обліку розрахунків за житлово-комунальні послуги шляхом автоматизації процесів обліку розрахунків за надані послуги або заплановані роботи, визначення боржників і планування витрат на придбання обладнання і виконання робіт з утримання житлового комплексу у належному для проживання стані.

Впровадження автоматизованої системи має сприяти поліпшенню якості наданих комунальних послуг, зменшення заборгованості за отримані послуги, організації і виконання робіт з обслуговування та ремонту майна ОСББ та спеціального обладнання.

А.3. Доступ до системи і функції системи

А.3.1. Доступ до системи

В проєкті автоматизованої системи повинні бути враховані можливості реалізації двох режимів роботи системи обліку:

- режим локального доступу – через автоматизовані робочі місця (АРМ) персоналу організації Замовника;
- режим віддаленого доступу – через мережу Інтернет для інших клієнтів системи.

Режим локального доступу повинен забезпечуватися шляхом установки на клієнтські робочі місця клієнтського програмного забезпечення та надання прав та повноважень з доступу до інформаційних ресурсів сервера системи відповідно до функціональних обов'язків клієнтів системи. Клієнтські робочі місця та сервер (сервери) системи повинні бути підключені до локальної мережі організації Замовника з швидкістю передачі даних не менше 100 мБіт/сек.

В проєкті системи повинні бути передбачені можливості подальшого розширення системи шляхом надання віддаленого доступу до ресурсів системи для інших клієнтів системи – через локальну мережу або мережу Інтернет. Режим віддаленого доступу повинен забезпечуватися через Веб-браузер і не має вимагати установки на клієнтське робоче місце додаткового до засобів операційної системи програмного забезпечення. При цьому доступ до інформаційних ресурсів повинен надаватися для таких видів користувачів системи:

- авторизованих користувачів – співробітників розрахункового центру та / або комунальних служб (надання прав і повноважень з доступу до інформаційних ресурсів системи забезпечується як для режиму локального доступу);
- авторизованих користувачів – отримувачів послуг (авторизація виконується на основі відомої користувачам приватної облікової інформації: адреса, номер розрахункової книжки тощо);

- інших користувачів – інформація надається без авторизації (оголошення, тарифи, таблиці-графіки виконання ремонтних робіт тощо).

А.3.2. Функції системи

Проект автоматизованої системи повинен враховувати наступну функціональність системи обліку:

- Ведення обліку з тепло- водо- каналізаційного господарства (Абонентський відділ):
 - ведення нормативно-довідкової інформації (види послуг, база постачальників і одержувачів послуг);
 - ведення особистих рахунків для фізичних та юридичних осіб;
 - ведення даних лічильників води та тепла;
 - ведення пільг;
 - ведення платежів;
 - формування квитанцій;
 - розрахунок теплових навантажень та планування теплових навантажень;
 - формування звітів.
- Ведення бухгалтерського обліку (Бухгалтерія);
 - нарахування заробітної плати;
 - ведення каси та проведення платежів через касу;
 - звіти (пенсійний, податковий, листів непрацездатності).
- Ведення кадрового обліку (Бухгалтерія):
 - ведення списків особового складу організації;
- Ведення обліку матеріалів (Бухгалтерія):
 - ведення складу та списань;
 - ведення по МВО;
 - ведення по банках;
 - ведення основних фондів;
 - розрахунки з ПДВ.

- **Адміністрування системи (Адміністратор)**
 - створення резервних копій даних системи;
 - відновлення даних системи з резервних копій даних системи;
 - створення, видалення та блокування користувачів системи;
 - редагування облікової інформації користувачів системи;
 - управління правами доступу до даних системи;
 - управління правами доступу до програмних модулів системи.

А.3.3. Вимоги до вхідних та вихідних даних

За способом представлення в автоматизованій системі вхідні дані повинні поділятися на текстові дані та штрихове кодування.

Текстові дані повинні вводитись оператором з клавіатури в діалоговому режимі в форми вводу. Перелік та розташування елементів форм вводу мають відповідати переліку та взаємному розташуванню елементів бланків паперового носія вхідних документів, що використовуються на центрі розрахунків АІС ОСББ та постановка задачі.

Штрихове кодування повинне вводитись через сканер штрих-коду та виконуватись відповідно до стандарту EAN-13 або сумісність коду із сканером штрих-коду повинен забезпечити Замовник.

Вихідні дані автоматизованої системи повинні бути представлені екранними формами та вихідними документами на паперовому носії відповідно до прийнятих на підприємстві Замовника. Для друку документів має використовуватись лазерний принтер формату А4. Перелік та розташування елементів екранних форм та вихідних документів мають відповідати переліку та взаємному розташуванню елементів бланків паперового носія вихідних документів організації Замовника.

При формуванні вихідних документів має бути передбачена можливість автоматичного занесення в поля документа умовно-постійної інформації, як то: ПІБ відповідальних виконавців, номери рахунків, поточні дата та час тощо.

А.4. Склад автоматизованої інформаційної системи та інтеграція з іншими автоматизованими системами

Проект автоматизованої системи повинен визначати систему як таку, що складається з наступних елементів:

- сервер баз даних і сервери додатків (вимоги до серверів та перелік серверів визначається Виконавцем);
- автоматизовані робочі місця (АРМ) підсистеми: Бухгалтерія, Абонвідділ, Адміністратор системи;
- клієнтське програмне забезпечення (перелік клієнтських робочих місць визначається Замовником відповідно до переліку АРМ системи, функцій системи та штатного комплектування персоналу організації Замовника).

Вимоги до системного та прикладного програмного забезпечення сервера (серверів) системи встановлює Виконавець.

Сервер (сервери) баз даних автоматизованої системи повинен (повинні) забезпечувати можливість доступу до даних відповідно до стандарту на мову запитів SQL. Шаблони документів, інша інформація, що може мати самостійне, окреме від автоматизованої системи значення може зберігатись по за межами сервера (серверів) баз даних автоматизованої системи у вигляді звичайних файлів. Вимоги до форматів файлів встановлює Розробник за погодженням з Замовником.

Вибір програмного забезпечення та конфігурацію клієнтських робочих місць, конфігурацію власної локальної мережі організації забезпечує Замовник.

Кількість клієнтських місць Розробником не обмежується, але може обмежуватись потужністю встановленого Замовником обладнання.

Комп'ютерна мережа повинна підтримувати сімейство протоколів TCP/IP. Вимоги до переліку протоколів та системного програмного забезпечення клієнтських робочих місць в частині функціонування автоматизованої системи обліку встановлює Виконавець.

В проєкті має бути врахована можливість інтеграції системи обліку з іншими автоматизованими системами шляхом надання спільного доступу до бази (баз) даних автоматизованої системи обліку. При цьому Розробник не несе жодних обов'язків щодо внесення будь яких змін в інші автоматизовані системи, дослідження архітектури, принципів роботи або інших особливостей цих систем, що виходять за межі офіційної та в законний спосіб отриманої документації до таких систем.

А.5. Документація, що підлягає передачі замовнику, порядок приймання роботи

А.5.1. Документація, що підлягає передачі

Замовнику передається:

- по закінченні робіт кожного етапу – звіт по етапу (2 екземпляри на паперових носіях та 1 екземпляр в форматі PDF на електронному носії);
- по закінченні робіт – заключний звіт (2 екземпляри на паперових носіях та 1 екземпляр в форматі PDF на електронному носії).

А.6. Порядок контролю та прийому

Для контролю і прийому повинен бути наданий опис програмного забезпечення підсистеми обліку розрахунків АІС ОСББ замовлень торговельної компанії, а також програма і методика випробування.

Порядок контролю і прийому даної розробки здійснюється представником замовника у присутності представника розробника згідно з програмою та методикою випробувань.

Якщо програмне забезпечення підсистеми обліку розрахунків АІС ОСББ замовлень торговельної компанії не пройшло випробування, виконавець зобов'язаний виправити помилки та недоліки у строк не більш ніж 1 місяць зі дня випробування.

За результатами прийому складається акт, який підписується представником замовника та представником розробника та утверджується керівниками організації-замовника і організації-розробника.

Перелік та терміни виконання робіт наведені в табл. А.1.

Таблиця А.1 – Стадії та етапи розробки підсистеми обліку розрахунків АІС ОСББ

Стадії	Етапи роботи	Початок робіт	Закінчення робіт
1. Ескізний проект	Розроблення ескізного проекту підсистеми обліку розрахунків АІС ОСББ	14.03.2025	04.03.2025
	Затвердження ескізного проекту підсистеми обліку розрахунків АІС ОСББ	04.04.2025	25.04.2025
2. Технічний проект	Розроблення технічного проекту підсистеми обліку розрахунків АІС ОСББ	25.04.2025	14.05.2025
	Затвердження технічного проекту підсистеми обліку розрахунків АІС ОСББ	14.05.2025	18.05.2025
3. Робочий проект	Розроблення проектних рішень підсистеми обліку розрахунків АІС ОСББ	18.05.2025	25.05.2025
	Розроблення програмної документації підсистеми обліку розрахунків АІС ОСББ	25.05.2025	01.06.2025
	Випробування програмного забезпечення підсистеми обліку розрахунків АІС ОСББ	01.06.2025	11.06.2025

Код моделей даних сервера (фрагмент)

```

class Abonent(_Base, Model):
    __verbose__ = u'Абонент'
    __ordering__ = ('lastname', 'firstname', 'middlename')
    _type = (
        ("physical", u"Фізична особа"),
        ("legal", u"Юридична особа")
    )
    _type_ = dict(_type)

    idcode = Field(unicode, max_length=40, default="#####",
verbose=u'ІНПП')
    firstname = Field(unicode, max_length=100, verbose=u'Ім_я')
    middlename = Field(unicode, max_length=100, verbose=u'по Батькові')
    lastname = Field(unicode, max_length=200, required=True,
verbose=u'Прізвище')
    typ = Field(unicode, required=True, choices=_type, verbose=u'Тип особи')
    consumergroup = Reference('ConsumerGroup', required=True)
    abonentcategory = Reference('AbonentCategory', required=True)
    lastname_idx = Index('lastname', False)
    lastname_middle_idx = Index('lastname, middlename', False)
    lastname_middle_firstname_idx = Index('lastname, middlename,
firstname', False)
    typ_idx = Index('typ', False)

    def __unicode__(cls):
        return "%s %s.%s." % (cls.lastname, cls.firstname, cls.middlename)

    @attribute(unicode, u'ПІБ')
    def short_name(self):
        if self.typ == 'physical':
            ## після видалення записів з Null-полями - код видалити
            lastname, firstname, middlename = self.lastname,
self.firstname, self.middlename
            firstname = firstname or ' '
            middlename = middlename or ' '
            return u'%s %s. %s.' % (lastname, firstname[0],
middlename[0])
        return self.lastname

    @attribute(int, u'Житло')
    def apartment_count(self):
        return Placement.count(abonent=self.id)

```



```

    """@attribute(unicode, u'Група споживачів')
    def consumergroup_name(self):
        return self.consumergroup.__unicode__()"""

    @attribute(unicode, u'Категорія абонентів')
    def abonentcategory_name(self):
        return self.abonentcategory.__unicode__()

class Apartment(_Base, Model):
    __verbose__ = u'Житло'
    __ordering__ = 'num'

    num = Field(unicode, required=True, verbose=u'Номер', max_length=50)
    floor = Field(unicode, verbose=u'Поверх')
    roomcount = Field(int, required=True, default=0, verbose=u'Кількість
кімнат')
    area = Field(float, verbose=u'Загальна площа')
    heatedarea = Field(float, verbose=u'Опалювана площа')
    building = Reference('Building', required=True)
    equipments = ManyReference('Equipment')
    apartmenttype = Reference('ApartmentType', required=True)
    num_building_idx = Index('num, building', True)

    def __unicode__(cls):
        return unicode(cls.num)

    @attribute(unicode, u'Адреса')
    def address(self):
        return u'%s, д.%s, кв.%s' % (self.building.street.name,
self.building.num, self.num)

    @attribute(unicode, u'Абоненти')
    def abonent_list(self):
        placement_set = self.placement_set
        result = [placement.abonent.__unicode__() for placement in
placement_set]
        return ', '.join(result)

    @attribute(int, u'Вулиць')
    def street_count(self):
        return Street.count(location=self.id)

    @attribute(float, u'Загальна площа')
    def area(self):

```

```

        return Placement.sum("area", lambda p : p.apartment == self.id)

    @attribute(float, u'Опалювана площа')
    def heatedarea(self):
        return Placement.sum("heatedarea", lambda p : p.apartment ==
self.id)

class ApartmentType(_Base, Model):
    __verbose__ = u'Тип житла'
    __ordering__ = 'name'

    name = Field(unicode, max_length=200, required=True, verbose='Назва',
unique=True)

    def __unicode__(cls):
        return cls.name

    @attribute(int, u'Кількість')
    def apartment_count(self):
        return Apartment.count(apartmenttype=self.id)

class Building(_Base, Model):
    __verbose__ = u'Будівля'
    __ordering__ = ('street', 'num')

    num = Field(unicode, required=True, max_length=50, verbose=u'Номер')
    code = Field(unicode, max_length=100, verbose=u'Код') # todo : we need
some samples. Required = ?
    mono = Field(bool, verbose=u'Одноквартирний', required=True)
    route = Field(unicode, required=True, verbose=u'Маршрут')
    freestanding = Field(bool, verbose=u'Окремий', required=True,
default=False)
    street = Reference('Street', required=True, verbose=u'Вулиця')
    district = Reference('District', required=True)
    housingservice = Reference('HousingService', required=True)
    buildingtype = Reference('BuildingType', required=False)
    num_street_idx = Index('num, street', True)
    district_idx = Index('district', False)
    housingservice_idx = Index('housingservice', False)

    def __unicode__(cls):
        return cls.num

    @attribute(int, u'Квартир', ('Apartment', lambda x: {'building': x}))

```

```

    def apartment_count(self):
        return Apartment.count(building=self.id)

    @attribute(unicode, u'Поселення')
    def location_name(self):
        return self.street.location.__unicode__()

    @attribute(unicode, u'Служба')
    def housingservice_name(self):
        return self.housingservice.__unicode__()

class BuildingType(_Base, Model):
    __verbose__ = u'Тип будинку'
    __ordering__ = 'name'

    name = Field(unicode, required=True, verbose=u'Назва', max_length=200,
unique=True)

    def __unicode__(cls):
        return cls.name

    @attribute(int, u'Будівель')
    def building_count(self):
        return Building.count(buildingtype=self.id)

"""class ConsumerGroup(_Base, Model):
    __verbose__ = u'Група споживачів'
    __ordering__ = 'name'

    name = Field(unicode, required=True, verbose=u'Назва', max_length=500)#,
unique=True
    #typ = Field(unicode, required=True, choices=Abonent._type,
verbose=u'Тип особи')

    def __unicode__(cls):
        return cls.name

    @attribute(int, u'Кількість')
    def abonent_count(self):
        return Abonent.count(consumergroup=self.id)
"""

class AbonentCategory(_Base, Model):
    __verbose__ = u'Категорія абонентів'

```

```

__ordering__ = 'name'

name = Field(unicode, required=True, verbose=u'Назва', max_length=500)#,
unique=True
shortname = Field(unicode, required=False, verbose=u'Скорочено',
max_length=100, unique=True)
typ = Field(unicode, required=True, choices=Abonent._type, verbose=u'Тип
особи')

def __unicode__(cls):
    return cls.name

@attribute(int, u'Кількість')
def abonent_count(self):
    return Abonent.count(abonentcategory=self.id)

class District(_Base, Model):
    __verbose__ = u'Район'
    __ordering__ = 'name'

    name = Field(unicode, required=True, verbose=u'Назва', max_length=200)
    location = Reference('Location', required=True)
    name_location_indx = Index('name, location', True)

    def __unicode__(cls):
        return cls.name

    @attribute(int, u'Будівель')
    def building_count(self):
        return Building.count(district=self.id)

    @attribute(unicode, u'Нас. пункт')
    def location_name(self):
        return self.location.name

    @attribute(unicode, u'Тип')
    def location_type(self):
        return self.location.locationtype.name

```