

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Кафедра прикладної математики

ЗВІТ

ПРО ВИКОНАННЯ КУРСОВОЇ РОБОТИ

з дисципліни “Бази даних та інформаційні системи”

на тему: Поповнення мобільного рахунку

Студента IV курсу, групи КМ-32
напряму підготовки 6.040301 –
прикладна математика
КАЦАЛАПА Р.В.

Викладач
ТЕРЕЩЕНКО І.О.

Оцінка: ____ балів

ЗМІСТ

ВСТУП.....	2
1 ПОСТАНОВКА ЗАДАЧІ	3
2 ОПИС РЕЗУЛЬТАТІВ	4
2.1 Визначення підгруп користувачів.....	4
2.2 Визначення прецедентів використання. Побудова use case діаграм	6
2.3 Життєвий цикл та взаємодія об'єктів ІС. Діаграми послідовності	10
2.4 Моделювання бізнес-процесів. Діаграми IEDF3	13
2.5 Вибір моделі життєвого циклу	19
2.6 Діаграми потоків даних.....	21
2.7 Діаграми відношень сутностей	25
ВИСНОВКИ.....	27
ДОДАТКИ.....	32
Додаток Д. Діаграма послідовності для групи неавторизованих користувачів	36
Додаток Е. Діаграма послідовності для групи авторизованих користувачів	37
Додаток Є. Діаграма послідовності для групи адміністраторів даних.....	39
Додаток Ж. Діаграма послідовності для групи адміністраторів системи	40
Додаток З. Концептуальна ER-діаграма	41
Додаток И. Логічна ER-діаграма	42
Додаток І. Фізична ER-діаграма	43
Додаток Ї. Sql-скрипт генерації бази для СУБД Oracle 10g	44

ВСТУП

Інформаційна система – це сукупність організаційних і технічних засобів для збереження та обробки інформації з метою забезпечення інформаційних потреб користувачів.

Метою роботи є інформаційна система поповнення мобільного рахунку.

ІС в першу чергу створюється для кінцевого користувача, для бізнесу. Тому основою при розробці системи є задоволення бізнес-потреб. Важливо вміти показати на будь-якому етапі проекту, як ті чи інші рішення відповідають бізнес-потреbam замовника. Вдалі рішення полегшують взаємодію і обмін інформацією, допомагають досягати бізнес-цілей.

Початком проектування є визначення границь ІС. Визначивши границі можна приступати до проектування, визначення властивостей, функцій, необхідних замовнику. У ІС, що прямо пов'язані з грошовими операціями, велика увага має бути приділена питанням безпеки: правилам обробки, зберігання конфіденційної інформації, паролів, технологіям взаємодії користувача з даними для уникнення порушень цілісності, конфіденційності та доступності даних. Наступним кроком є аналіз швидкодії системи. Правильне проектування дозволяє зекономити кошти на апаратній частині системи та підтримці проекту, а також впливає на рівень обслуговування кінцевого користувача. Бізнес-системи рідко функціонують самі по собі, вони поєднують багато підсистем, інтегруються з оточуючим інформаційним середовищем. В контексті цієї інтеграції необхідно приділяти увагу:

- взаємодії з існуючими додатками, системами;
- конвертації та перетворенні даних.

Системи поповнення мобільного рахунку взаємодіють із банківськими системами та системами операторів телефонного зв'язку, останні визначають власні програмні інтерфейси, через які проходить обмін інформацією.

1 ПОСТАНОВКА ЗАДАЧІ

Визначити та описати групи користувачів та їхні ролі в ІС, що призначена для поповнення мобільного рахунку. Для всіх видів користувачів створити Use Case діаграму, що описує процеси всіх користувачів (без адміністратора). Включити окремо опис для адміністраторів, що вказує управління процесами, побудувати діаграми послідовностей для визначених груп користувачів, визначити основні процеси системи, побудувати діаграми нотації станів всіх об'єктів системи, які беруть участь в основних процесах системи, побудувати діаграми потоків основних процесів, аргументувати вибір моделі життєвого циклу для всіх основних процесів інформаційної системи, створити діаграми потоків даних для основних процесів до другого рівня, побудувати діаграми відношень сутностей, описати концептуальну, логічну та фізичну модуль даних.

2 ОПИС РЕЗУЛЬТАТІВ

2.1 Визначення підгруп користувачів

Для інформаційної системи поповнення мобільного рахунку визначено наступні підгрупи користувачів:

- неавторизовані користувачі;
- авторизовані користувачі;
- адміністратори.

Неавторизованим користувачам надається доступ до наступних операцій:

- перегляд основної інформації, що стосується умов надання послуг поповнення мобільного рахунку – терміни нарахування коштів, розміри комісії;
- авторизація \ реєстрація в системі, відновлення пароля;

Авторизовані користувачі мають власний Особистий Кабінет (ОК) і їм надаються права виконувати наступні операції:

- перегляд основної інформації, що стосується умов надання послуг поповнення мобільного рахунку – терміни нарахування коштів, розміри комісії;
- вихід з ОК;
- додавання \ видалення номеру мобільного телефону для поповнення;
- додавання \ видалення даних банківської картки;
- редагування профілю (зміна паролю, прив'язка електронної пошти);
- здійснення оплати послуги;
- перегляд історії поповнень та перевірка статусів платежів даного акаунту;
- перевірка статусу операції поповнення рахунку за номером платежу.
- підписка на регулярний платіж.

Адміністратори інформаційної системи поділяються на два типи:

- адміністратор даних;
- адміністратор ІС.

Для адміністратора даних визначені операції звичайного користувача, а також наступні операції:

- перегляд зареєстрованих користувачів, обмеження доступу користувачів;
- надсилання листів на пошту користувача;
- перегляд історії платежів.

Для адміністратора системи:

- резервне копіювання та відновлення інформації;
- контроль росту інформаційної системи (перегляд кількості користувачів, кількості запитів за проміжки часу тощо);
- контроль доступності зовнішніх сервісів для переводу коштів.

Банківські дані конкретного користувача — номер його картки, термін придатності картки, секретний код не зберігаються. Зберігається лише останні 4 цифри номеру картки для користувацької ідентифікації.

2.2 Визначення прецедентів використання. Побудова use case діаграм

Система, що проектується, представляється у вигляді множини сутностей (акторів), що взаємодіють з системою через варіанти використання. При цьому актором називають будь-яку сутність, яка взаємодіє з системою ззовні. Це може бути людина, технічний пристрій, програма чи інша система, що може виступати джерелом впливу на модельовану систему, як це визначить сам розробник. Прецедент слугує для опису сервісів, які система надає актору. Діаграма варіантів використання може доповнюватися пояснювальним текстом, який розкриває сенс чи семантику компонентів.

Для побудови діаграми потрібно визначитись з наступними елементами:

- функціонал (сервіси), що визначають варіант використання;
- актори;
- зв'язки (переходи) між акторами та процесами.

Тобто діаграма прецедентів – це актори, прецеденти та зв'язки між ними.

2.2.1 Опис діаграми для неавторизованого користувача

Діаграма прецедентів (Use case diagram) неавторизованого користувача наведена у додатку А. Неавторизованому користувачу доступний процес авторизації, який має вкладені процеси – реєстрація, логін та авторизація у якості тимчасового користувача. Останній виконується системою автоматично, але користувач може залогінитися або зареєструватися, а також, система може залогінити користувача автоматично, якщо термін дії сесії не сплив. Процес реєстрації супроводжується підтвердженням реєстрації, після чого користувач повертається до інтерфейсу авторизації і не може змінити дані реєстрації, тобто

повернутися безпосередньо до попереднього процесу. Інтерфейс входу в акаунт “Login” має розширення – відновлення паролю “Restore password”. За умови входу в особистий кабінет користувач стає авторизованим для інформаційної системи як постійний користувач.

Тимчасовий користувач не вважається авторизованим в ІС в повному сенсі цього слова і йому надаються права виконувати наступні операції:

- перегляд основної інформації, що включає в себе (includes) опис процедури поповнення мобільного рахунку, правила користування сервісом та основні положення, згідно з якими працює система, контактну інформацію;
- перевірка статусу платежу за його унікальним номером;
- зв'язок зі спеціалістом, що надає технічну допомогу.

2.2.2 Опис діаграми для авторизованого користувача

Авторизованому користувачу (див. Додаток Б) інтерфейс авторизації «Auth» надає можливість у увійти в особистий кабінет, у разі виходу з особистого кабінету також зареєструватися під новим ім'ям користувача або переглядати лише інформацію, що доступна неавторизованим користувачам, у якості тимчасового користувача.

Особистий кабінет надає панель керування, взаємодію з якою визначає користувацький процес «Manage Control panel», який включає наступні вкладені процеси:

- процеси, спільні з неавторизованими користувачами (перегляд основної інформації, правил та положень функціонування системи, контактних даних, звертання за технічною допомогою);
- редагування даних акаунту;
- перегляд історії платежів, розширеннями для якого є застосування фільтрів пошуку;

- управління збереженими номерами мобільних телефонів – перегляд доданих номерів, додавання нового номеру, видалення номеру;
- управління банківськими картками акаунту – перегляд доданих банківських рахунків, додавання нової картки, видалення картки. Додавання картки супроводжується подальшою верифікацією власника картки шляхом виконання відповідної банківської транзакції. Картка додається лише у випадку успішної верифікації;
- створення нового платежу. Цей процес включає вибір номеру телефону для поповнення з-поміж збережених або введення його вручну, вибір банківської картки серед збережених або введення банківських реквізитів вручну, вибору суми поповнення та можливість зробити платіж регулярним, процес обробки платежу (підготовки транзакції). Як тільки транзакція може бути виконана, редагування платежу стає неможливим і дані відправляються на обробку до банківської системи.
- вихід з панелі керування та особистого кабінету (Logging out).

2.2.3 Опис діаграми для адміністратора даних

Адміністратор даних (див. Додаток В) може виконувати всі операції, що доступні звичайним авторизованим і неавторизованим користувачам – вихід з особового кабінету із подальшою можливістю створення нового акаунту, користування системою у якості тимчасового користувача, входом під іншим ім'ям користувача і усі операції, що вкладені до цих або сліднують після них, а також додаткові, вкладені відносно процесу панелі керування (Manage Control Panel):

- надавати технічну підтримку за допомогою онлайн чату або електронною поштою в рамках доступних засобів;
- керувати акаунтами: створювати, відновлювати на видаляти акаунти, переглядати акаунти;
- переглядати інформацію про тимчасових користувачів;

- переглядати стани платежів користувачів системи, інформацію про дату та час операції, суму платежу, номер мобільного, що поповнювався, номери транзакцій.

Банківські реквізити користувача системи обробляються виключно системою, зберігаються у зашифрованому вигляді і недоступні для перегляду та модифікації нікому.

2.2.4 Опис діаграми для адміністратора інформаційної системи

Адміністратор інформаційної системи, діаграма прецедентів для якого наведена у додатку Г, може виконувати всі операції, що доступні авторизованим і неавторизованим користувачам, а також спеціальні, необхідні для підтримки та відстеження функціонування інформаційної системи:

- переглядати дані про навантаження на систему, відслідковувати можливий її ріст;
- управляти резервним копіюванням, відновленням інформації;
- налаштовувати конфігурацію системи;
- перевіряти доступність зовнішніх сервісів (банківських систем, сервісів SMS-сповіщення тощо).

2.3 Життєвий цикл та взаємодія об'єктів ІС. Діаграми послідовності

Діаграми послідовності – діаграма, на якій для деякої сукупності об'єктів на єдиній часовій осі показаний їх життєвий цикл та взаємодія (відправлення запитів та отримання відповідей). У якості об'єктів можуть виступати користувачі, класи, програмні компоненти та системи в цілому.

Для групи неавторизованих користувачів діаграма послідовності наведена у додатку Д. Зліва направо перерахуємо об'єкти в системі, що задіяні у даному випадку:

- user – неавторизований користувач;
- session processing – підсистема керування сесіями (Login sessions);
- login – підсистема авторизації, що включає як видиму користувачам форму авторизації, так і невидиму частину – код авторизації при дійсній сесії;
- сервер з даними про об'єкти, що створюються, змінюються та зникають у рамках системи у ході її функціонування. Це такі об'єкти як сесія, користувач (акаунт, профіль), платіж, транзакція, банківська картка, телефонний рахунок;
- online support – підсистема он-лайн підтримки користувачів;
- register – підсистема реєстрації, що включає форми реєстрації, скриптові файли обробки форм та підтвердження реєстрації;
- check payment – частина підсистеми payments, що доступна у більш ширшому варіанті для авторизованих користувачів та адміністраторів, дозволяє переглядати статуси зроблених платежів;
- mail – сервер електронної пошти – використовується для відправки листів користувачам з підтвердженням реєстрації;
- bank – банк, що надає інструменти розробника для здійснення платіжних операцій зі стороннього ресурсу (API, SDK).
- admin – адміністратор, що надає он-лайн підтримку користувачам.

Для групи авторизованих користувачів діаграма послідовності наведена у додатку Е. Для цієї групи мають місце об'єкти з діаграми для неавторизованих користувачів, а також:

- profile – керування профілем користувача;
- payments – дозволяє переглядати історію платежів та створювати нові;
- phone numbers – дозволяє переглядати збережені номери мобільних телефонів та додавати нові для подальшого поповнення;
- bank cards – дозволяє переглядати збережені дані власних банківських карток та додавати нові банківські картки для збереження їх у системі;

Власне поповнення мобільного рахунку відбувається шляхом створення нового платежу з вказаним номером телефону, сумою поповнення, реквізитами банківської картки, відправки його до банку та підтвердження платежу користувачем.

Для групи адміністраторів даних діаграма послідовності наведена у додатку Є. Для цієї групи передбачені всі процеси авторизованих і, як наслідок, неавторизованих користувачів, а на діаграмі зображені притаманні лише адміністраторам даних маніпуляції з об'єктами:

- online support – читання повідомлень від користувачів, відправка відповідей;
- accounts – перегляд даних про зареєстрованих користувачів, реєстрація нового користувача і відправка на пошту даних для входу, деактивація акаунту, видалення користувача;
- payments – перегляд інформації про платежі – дата, сума, номер платежу, дані користувача, статус платежу;
- session processing – перегляд активних сесій (авторизованих та тимчасових користувачів), ініціювання переривання сесій.

Для групи адміністраторів системи діаграма послідовності наведена у додатку Ж. Для цієї групи передбачені процеси також і авторизованих користувачів, а на діаграмі зображені лише специфічні для даної групи операції над об'єктами:

- server load statistics – відслідковування навантаження на систему – кількість активних сесій, статистика сесій, кількість користувачів в системі, кількість платежів;
- configure server – налаштування резервного копіювання, конфігурацій, планувальника перезавантаження, вмикання та вимикання серверу, відновлення резервних даних;
- test tools – засоби для перевірки та відслідковування доступності зовнішніх ресурсів (банківської системи, операторів мобільних телефонів).

2.4 Моделювання бізнес-процесів. Діаграми IEDF3

Основними процесами системи є:

- реєстрація, авторизація – створення сеансів, вхід у рамках сесії, обробка форм реєстрації та входу;
- керування платежами – створення, редагування, відправка, перевірка;
- ініціювання транзакції, процеси, що обмінюються даними з банком.

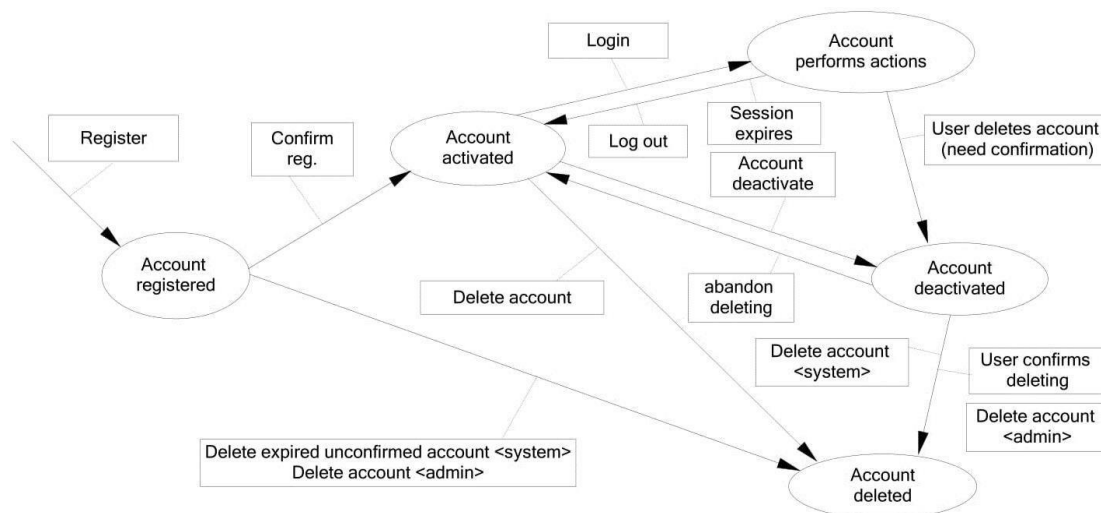
Для ефективного керування будь-яким процесом, необхідно мати уявлення про його сценарії та іншу документацію проекту.

Існує два типи діаграм в стандарті IDEF3, що представляють опис технологічного процесу в різних ракурсах. Діаграми першого типу називаються діаграмами опису послідовності етапів процесу (PFDD), другого – діаграмами станів об'єкту і його трансформацій (OSTN). Діаграми PFDD описують технологічний процес з точки зору спостерігача, а OSTN – з точки зору самого об'єкту системи.

2.4.1 Діаграми станів об'єктів системи

Для відображення станів на діаграмах виокремлено об'єкти: акаунт та платіж.

На рисунку 2.4.1.1 зображено діаграму OSTN для об'єкту «акаунт».



(рисунк 2.4.1.1 – Діаграма станів об'єкту «акаунт»)

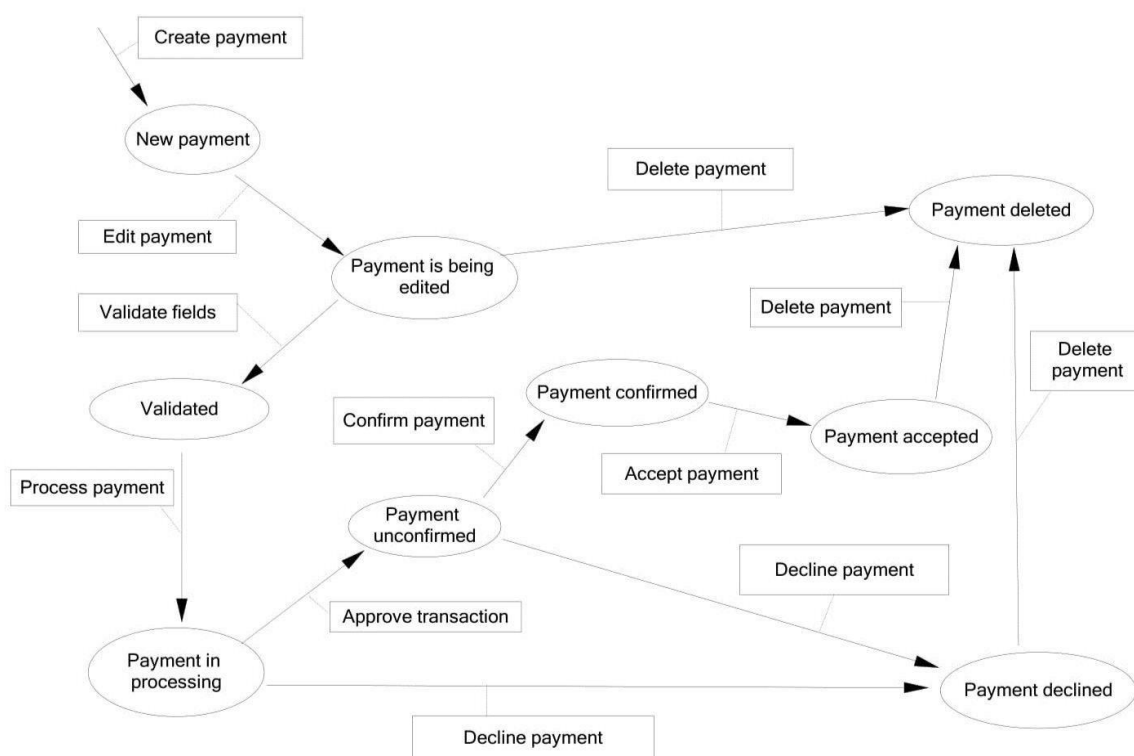
Появу нового об'єкту «акаунт» ініціює процес реєстрації. Якщо новим користувачем не надано підтвердження дії, то акаунт може бути видалений адміністратором або системою. Інакше, якщо підтвердження відбулося, акаунт стає активованим, тобто за його даними можна здійснити вхід до системи. Акаунт, що не проявляв активність протягом деякого часу, встановленому в налаштуваннях системи, може бути деактивований з повідомленням власника електронною поштою, а потім видалений, або одразу видалений на розсуд адміністраторів.

Акаунт має стан такого, що виконує операції після процесу входу в систему. Акаунт з цього стану може перейти в деактивований, якщо його власник обере опцію видалення акаунту, або назад в активований, якщо термін сесії сплине, або користувач власноруч вийде із системи. Акаунт може бути переведений зі стану деактивованого у стан активованого, якщо користувач або адміністратор відмінить видалення. Через деякий час, встановлений у конфігурації системи, або після

підтвердження користувачем видалення, або діями адміністратора, акаунт може бути видалений.

Платіж – об’єкт, що створює користувач для здійснення оплати послуги поповнення мобільного рахунку. Цей об’єкт характеризується унікальним номером і містить в собі дані про користувача, номери мобільних рахунків, дані банківської картки, суму поповнення, інформацію про мерчанта. Об’єкт проходить через декілька процесів системи і навіть поза системою, а тому його стан зазнає змін.

На рисунку 2.4.1.2 зображено діаграму OSTN для об’єкту «платіж».

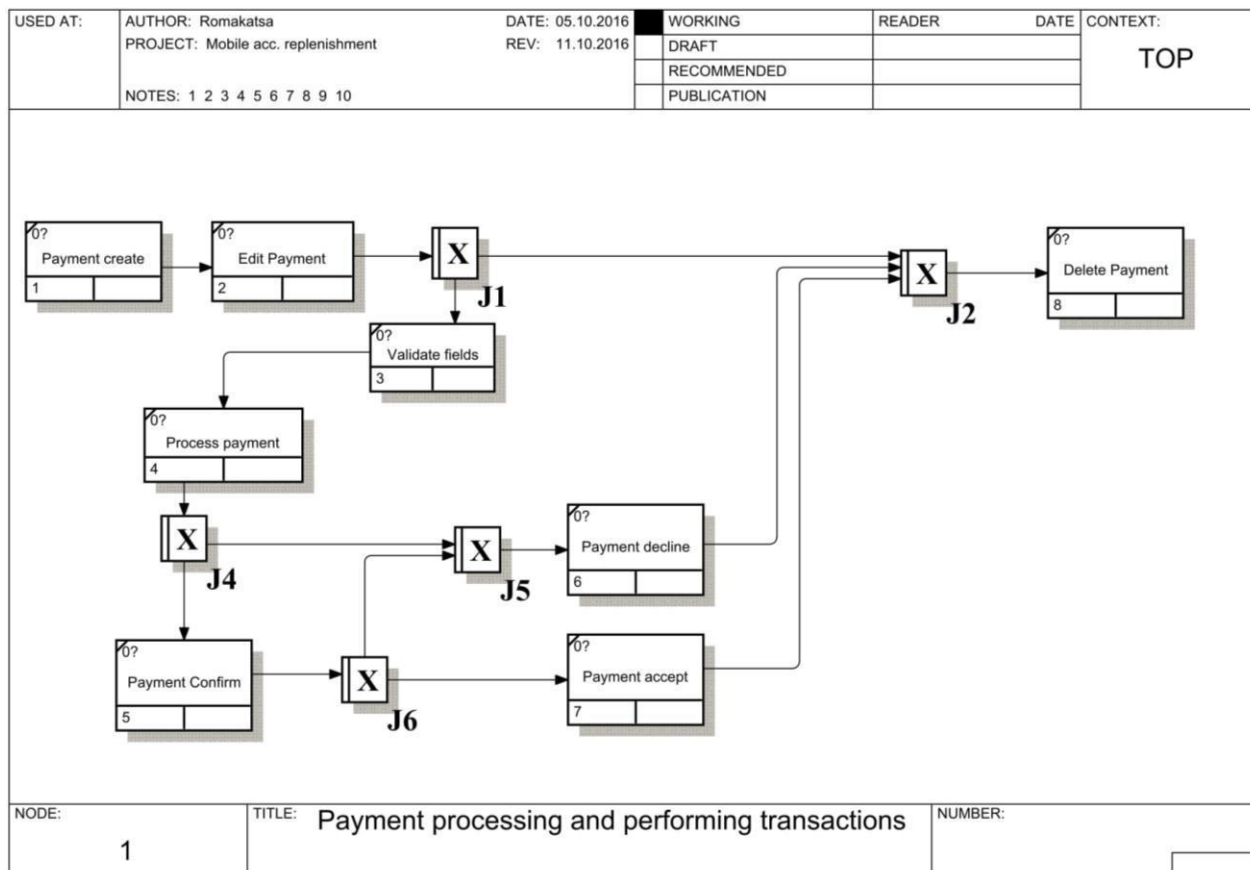


(рисунок 2.4.1.2 – Діаграма станів об’єкту «платіж»)

Об’єкт «платіж» з’являється в системі у процесі створення платежу. Після редагування платіж має стан редагованого. Правильність редагування перевіряється процесом валідації, після чого процес може бути переведений у стан валідного, або редагований платіж може бути видалений користувачем. Валідний платіж може

бути відправлений на обробку і перейти у стан такого, що обробляється сумісно системою для поповнення мобільного рахунку та банком. Якщо банк дає можливість здійснювати операції з карткою, то платіж стає непідтвердженим, якщо ж операція неможлива, то банк відхиляє транзакцію і платіж переходить у стан відхиленого. Непідтверджений платіж може перейти у стан підтвердженого після вдалого проходження процесу верифікації власника картки, або у стан відхиленого – за невдалого (коли власник не ввів підтверджувальні дані або ввів їх некоректно). Підтверджений платіж закінчує оброблятися і переходить у стан здійсненого, зберігається певний час у системі, після чого, як і відхилений платіж видаляється.

попередньо зареєструватися. Сесія авторизованого користувача може бути видалена за неактивності або після виходу з акаунту. На рисунку 2.4.2.2 зображено діаграму послідовності етапів процесу здійснення платіжу.



(рисунку 2.4.2.2 – Діаграма послідовності етапів процесу здійснення транзакції та обробки платежу)

2.5 Вибір моделі життєвого циклу

Весь програмний продукт в цілому розроблюється згідно інкрементної моделі життєвого циклу. За такої моделі увесь необхідний функціонал розділено і реалізовується у різних збірках – мають місце декілька циклів відповідно до кожної підсистеми чи процесу, розробка кожного з яких відбувається за каскадною моделлю. Кожний новий модуль проходить стадії визначення вимог, дизайну, впровадження та тестування.

Варто зазначити, що зазначені вище збірки в даній моделі не є прототипами продукту чи версіями, готовими до використання.

Дана модель обрана з тих міркувань, що задовольняє наступні особливості проекту:

- вимоги до системи в цілому визначені з достатньою точністю;
- абсолютна більшість функціоналу визначена наперед і вноситимуться лише незначні зміни, тому система може бути розділена на частини;
- чітко визначені терміни здачі проекту, необхідно буде мати реалізованим найбільш важливий функціонал, а розробку деяких компонентів можна залишити на етапі аналізу;
- використовуються нові технології та знання, необхідні навички ще не здобуті на даному етапі, а тому аналіз та дизайн деяких компонентів відкладається, в той час коли інші функції можна вдало реалізовувати.

Найбільш пріоритетним є впровадження наступних модулів:

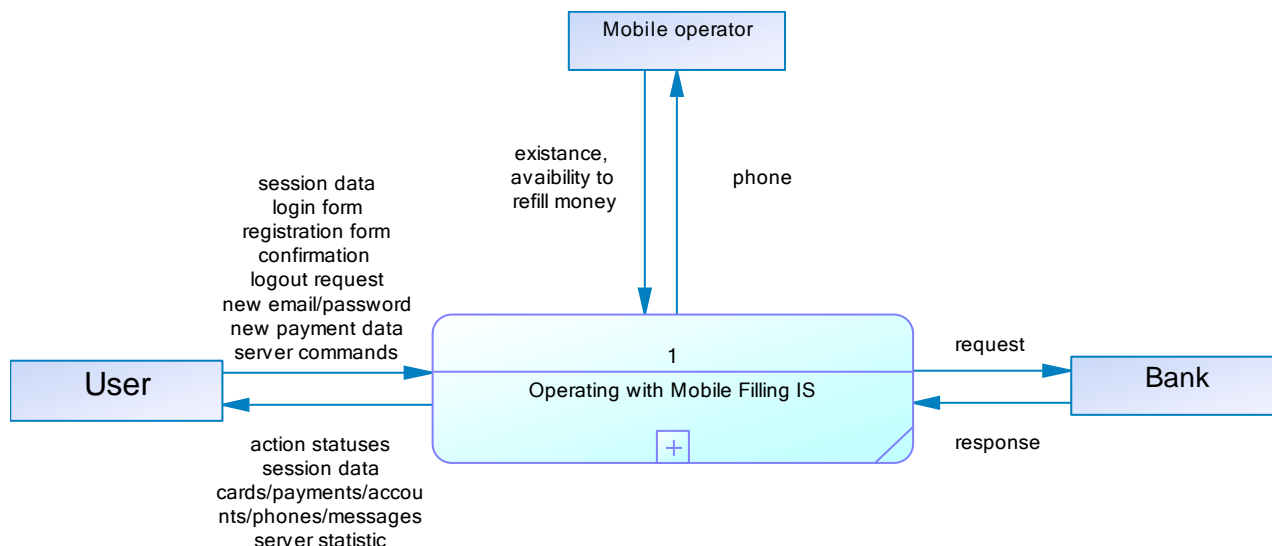
- реєстрація, авторизація (процеси створення сесій, авторизації у рамках сеансу, обробка форм реєстрації та входу);
- особистий кабінет користувача (надання централізованого доступу до інших функцій);
- керування платежами (процеси створення, редагування, відправки платежів та перевірки їх статусів);

- комунікації з банком (процеси формування та відправки запитів до банку та обробки відповідей за допомогою API, що надається банком);
- засоби адміністрування.

Дані модулі розроблюватимуться за каскадною моделлю. Для кожного процесу окремо додатково визначатимуться вимоги, що доповнюватимуть загальні вимоги до системи, виходячи з того, що вже буде реалізовано на той момент і відповідно до можливих нових обставин.

2.6 Діаграми потоків даних

Створення діаграм потоків даних починається з діаграми нульового рівня, що представлена на рисунку 2.6.1.



(рисунок 2.6.1 – Діаграма потоків даних нульового рівня, DFD0)

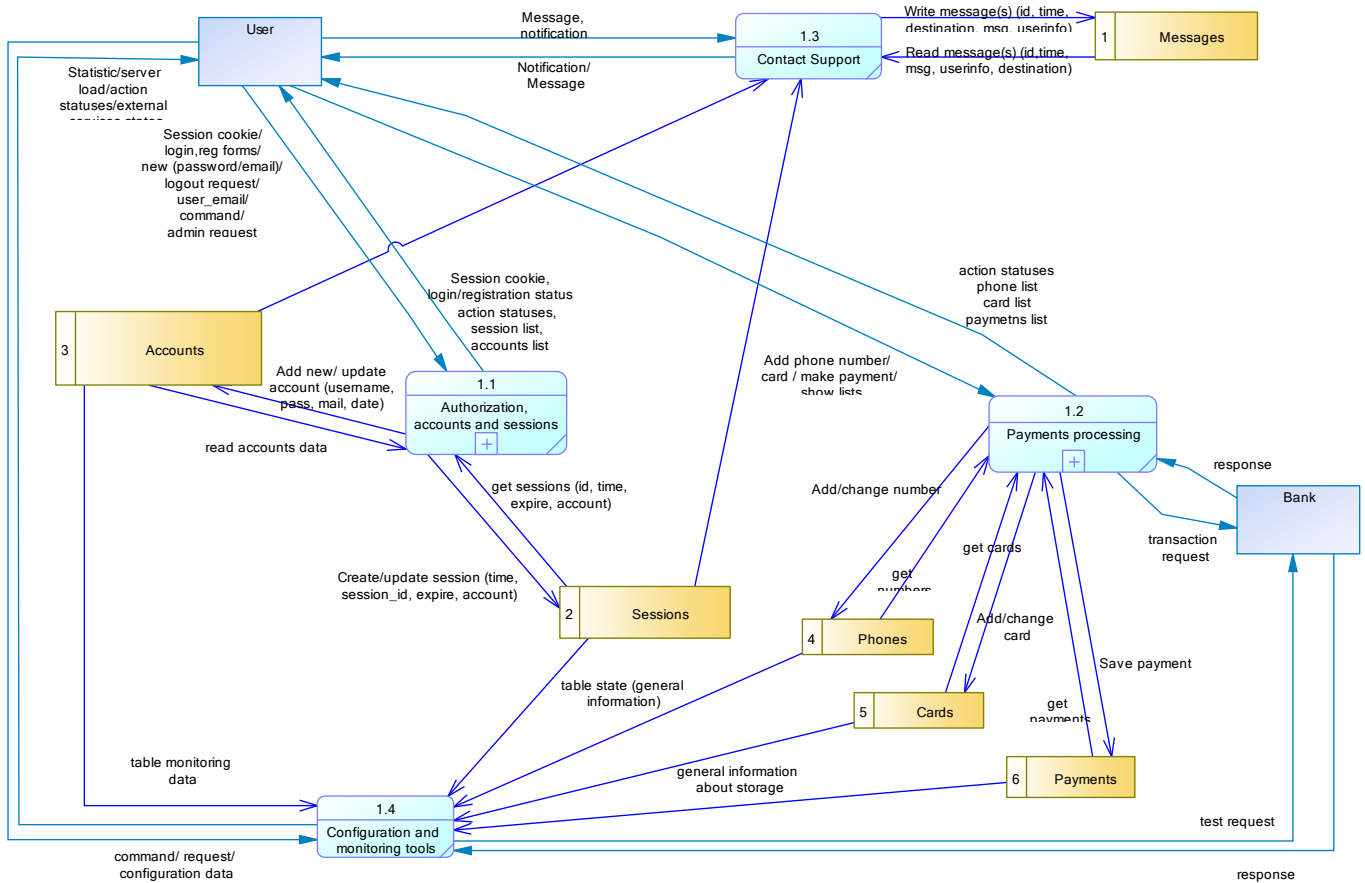
На діаграмі зображено зовнішні сутності: користувач та банк та оператор телефонного зв'язку.

Робота з інформаційною системою (процес 1 на DFD0) на діаграмі першого рівня (рисунок 2.6.2) представлена наступними підсистемами:

- авторизації, реєстрації та обробки сесій;
- створення та обробки платежів;
- надання онлайн підтримки користувачам;
- налаштування та моніторингу інформаційної системи.

На діаграмі також визначено шість внутрішніх накопичувачів:

- акаунти;
- сесії;
- номери мобільних телефонів;
- банківські картки;
- платежі;
- повідомлення.

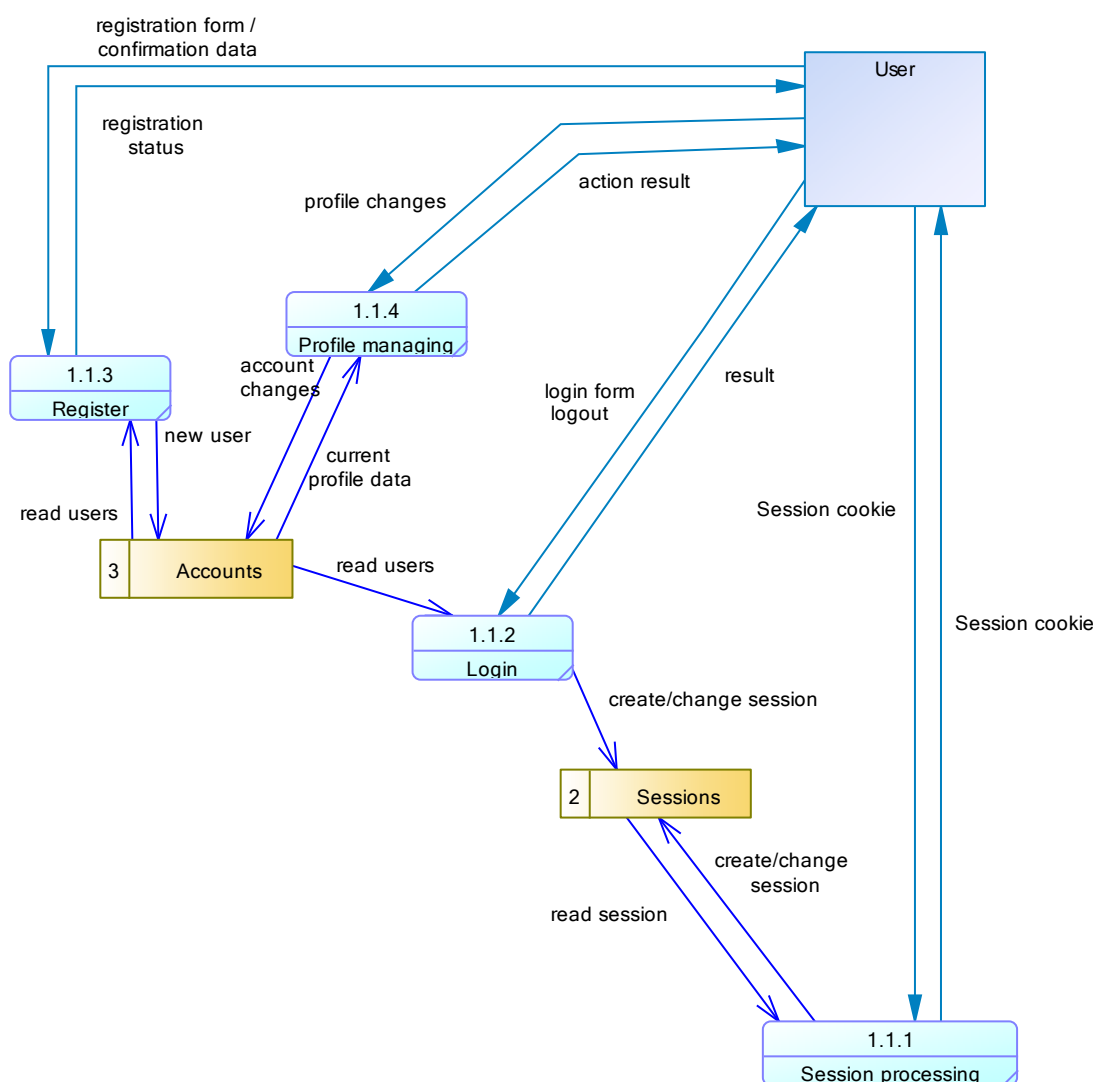


(рисунок 2.6.2 – Діаграма потоків даних першого рівня DFD1).

Як видно з діаграми процесу авторизації, реєстрації та обробки сесії оперують cookie-файлами з ідентифікаторами сесій користувача, login- та реєстраційними формами, зберігаючи дані у накопичувачах акаунтів та сесій.

Процеси створення та обробки платежів маніпулюють даними про картки, номери телефонів, власне платежі, встановлюють зв'язок з банком шляхом запитів та відповідей (request and response).

На рисунку 2.6.3 зображено діаграму другого рівня для процесів авторизації.



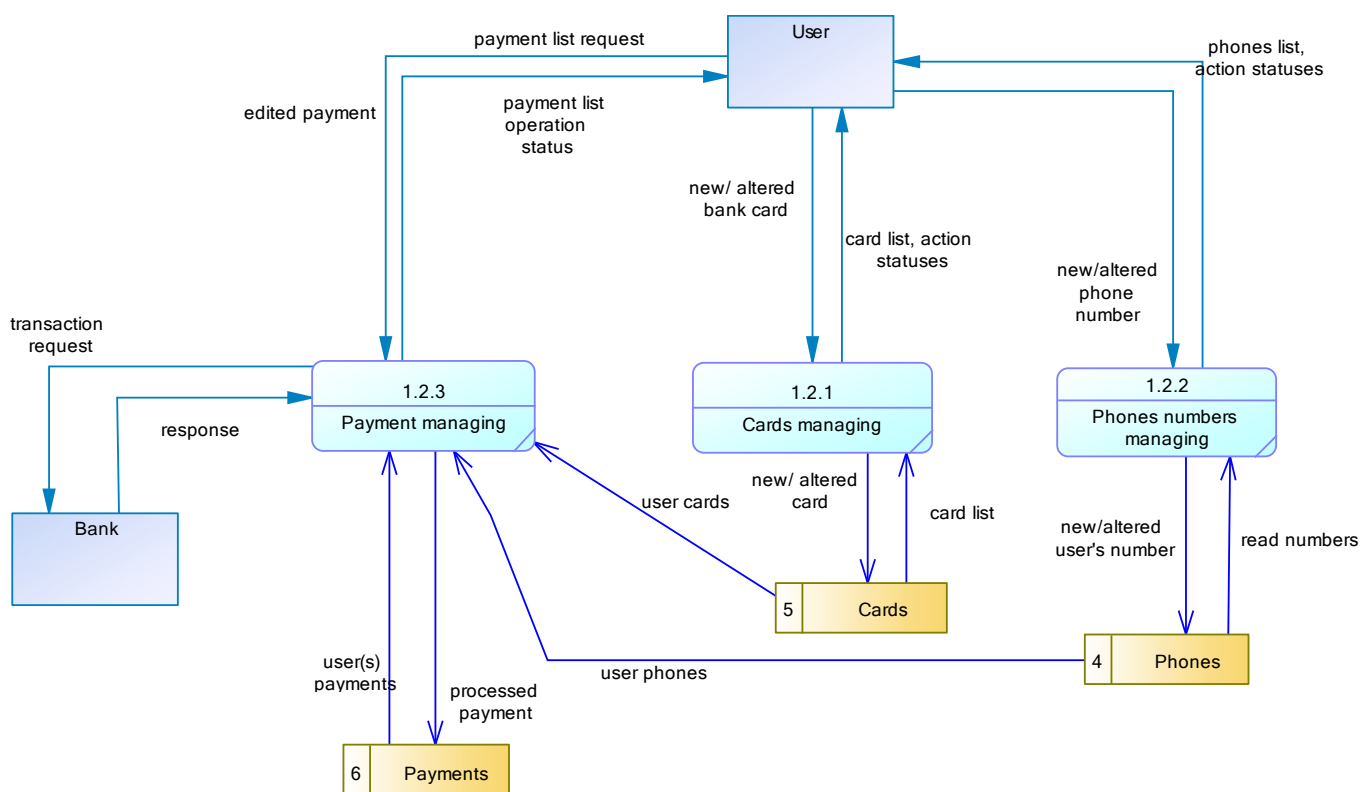
(рисунок 2.6.3 – Діаграма потоків даних другого рівня для процесів авторизації).

Користувач надсилає дані реєстрації та підтвердження, а відповідна підсистема створює акаунт у базі даних, якщо це можливо, та надає відповідь

користувачу. Підсистема входу в систему звіряє надіслані користувачем дані із даними акаунтів, створює сесію авторизованого користувача, надає відповідь користувачу про статус авторизації та сесійний cookie-файл.

Підсистема обробки сесій може авторизувати користувача на основі надісланого ним cookie-файлу з ідентифікатором сесії, якщо така існує у відповідному внутрішньому накопичувачі і термін її дії не сплив.

Підсистема створення та обробки платежів включає в себе підпроцеси, пов'язані з оперуванням даними номерів мобільних, банківських карток, платежів – створення, перегляд, редагування, видалення; створення банківських транзакцій. На рисунку 2.6.4 зображено діаграму другого рівня для процесів обробки платежів.



2.7 Діаграми відношень сутностей

Згідно з попередніми розділами курсової роботи, в системі присутні такі сутності:

- користувач (його акаунт);
- банківська картка;
- платіж;
- номер мобільного телефону;

Відповідно, ці сутності, а також допоміжні, присутні на концептуальній ER-діаграмі (додаток 3).

Користувач (акаунт) має наступні поля у відповідній таблиці БД:

- логін для входу, є головним ключем;
- адреса електронної пошти, може змінюватися, тому не застосовується для ідентифікації користувача;
- ім'я та прізвище;
- пароль, зберігається у хешованому вигляді;
- роль користувача. Таблиця ролей та допоміжні до неї на цій діаграмі відсутні в силу невизначеності у шляхах реалізації даного функціоналу;
- статус акаунту, перелік яких визначений на діаграмах трансформацій станів об'єктів ІС і зберігається в окремій таблиці "Account statuses". Акаунт та статус пов'язані відношенням «багато до багатьох», адже акаунти можуть мати різні статуси впродовж існування, хоча не одночасно. Введена проміжна таблиця, що зберігає відповідність акаунтів статусам та містить історію їх змін.

Номери мобільного телефону користувача має наступні поля:

- логін користувача – частина складеного головного ключа;
- номер мобільного телефону – частина складеного головного ключа;
- назва (підпис) номеру у даного користувача в особистому кабінеті.

Декілька користувачів можуть мати збереженими однакові номери мобільних телефонів для їх поповнення. Таблиця номерів телефонів відсутня, оскільки номер

телефону не має інших характеристик у системі, окрім самого номеру, тому номер є просто символьним полем, як у даній таблиці, так і у таблиці платежів.

Банківська картка має наступні поля:

- номер картки – частина первісного ключа сутності (зберігаються лише останні 4 символи для користувацької ідентифікації картки);
- токен, що дозволяє здійснювати оплату без реквізитів, надається банком після першої оплати або верифікації власника картки;
- назва картки у особистому кабінеті;
- логін користувача – зовнішній ключ та частина первісного ключа.

Платіж має наступні поля:

- ідентифікатор платежу – первісний ключ;
- дата та час обробки платежу;
- сума платежу;
- номер мобільного телефону;
- номер картки, з якої здійснено оплату – зовнішній ключ, визначає зв'язок «одна картка – багато платежів»;
- логін користувача – зовнішній ключ, визначає зв'язок «один користувач – багато платежів».

У додатку II наведено логічну модель даних.

У додатку I наведено фізичну модель даних.

У додатку III наведено sql-скрипт, згенерований з фізичної моделі для СУБД Oracle 10g.

ВИСНОВКИ

У системі визначено наступних користувачів:

- адміністратор ІС;
- адміністратор даних;
- неавторизований користувач;
- авторизований користувач.

Користувачів кожного виду може бути більше ніж один, тобто маємо 4 базові групи користувачів.

Неавторизованим користувачам надано можливості виконувати операції з найнижчим рівнем безпеки, що не модифікують більше трьох зв'язків між даними та не вимагають зберігання даних про користувача (операції, здебільшого пов'язані із переглядом інформації);

Авторизовані користувачі можуть виконувати операції, пов'язані з модифікуванням багатьох зв'язків між даними та зберіганням інформації про користувача, а також платіжні операції.

Адміністратори контролюють доступність сервісу поповнення мобільного рахунку, а також сторонніх сервісів, необхідних для надання послуг, забезпечують технічну підтримку інших користувачів. Адміністратори мають найвищі права і можуть звужувати права інших користувачів за потреби, але самі діють у рамках, визначених розробниками інформаційної системи.

Розроблені Use Case UML діаграми складаються з наступних елементів:

- а) актор;
- б) процес;
- в) перехід.

Актор – це об'єкт ІС, що виконує певні процеси і поділяється на:

- master (адміністратори);
- slave (решта).

Процес описує інтерфейс, що надає користувачу певний функціонал.

Перехід вказує перехід користувача між інтерфейсами та взаємозв'язок між інтерфейсами.

Діаграми послідовностей уточнюють можливості поведінки користувача і його взаємодію з системою, а саме з наступними групами процесів:

- авторизації та реєстрації;
- керування банківськими картками;
- здійснення платежів;
- технічної підтримки.

Різниця між авторизованими, а також адміністраторами, та неавторизованими користувачами для системи полягає у відсутності активної сесії у останніх. Це перевіряється при будь-якому запиті клієнта до сервера і якщо сесія існує і активна, термін її дії продовжується.

Неавторизований користувач після створення (або продовження, що не вказано на діаграмі) сесії тимчасового користувача може виконувати операції перевірки платежу, звертання до підтримки, заповнення та відправки форм входу та реєстрації у будь-якій послідовності. Авторизація та реєстрація можуть бути не вдалими, що відображено на діаграмі. Після успішної авторизації користувач стає авторизованим, що відображається в його сесії, і реєструватися не може.

Авторизований користувач після перевірки активності сесії та її продовження може виконувати одні з наступних груп дій (альтернативно):

- переглядати та змінювати профіль, перевіряти та створювати платежі, редагувати банківські картки та номери телефонів у будь-якій послідовності;
- видаляти акаунт;
- виходити із системи.

Після видалення акаунту чи виходу із системи користувач стає неавторизованим.

Адміністратори є авторизованими користувачами із додатковими правами. Дії, зазначені на діграмах у додатках Є та Ж, відокремлені пунктирними лініями можуть виконуватися у будь-якій послідовності.

Основні процеси системи поділені на дві ключові групи:

- авторизаційні;

- ті, що відповідають безпосередньо за створення та обробку платежів.

До неосновних, але важливих для повного функціонування системи, належать засоби для комунікації між адміністраторами та користувачами. Вони мають бути визначені пізніше, після етапів впровадження основних процесів. Відповідно до визначених груп процесів, визначені два основних об'єкта, стани яких змінюються під час виконання основних процесів:

- акаунт користувача;
- платіж для поповнення мобільного рахунку.

Акаунт користувача має стани:

- зареєстрований;
- активований;
- активний (існує активна сесія, пов'язана з цим акаунтом);
- деактивований;
- видалений.

Платіж має стани:

- новий;
- редагований;
- валідний;
- в обробці;
- непідтверджений;
- підтверджений;
- відхилений;
- зарахований;
- видалений.

Модель життєвого циклу розроблюваного програмного забезпечення та його підсистем була визначена як інкрементна. Ця модель поєднує у собі багаторазове використання каскадної – для поетапного впровадження функціоналу. Найбільш пріоритетними для реалізації є модулі авторизації, особистого кабінету, обробки

платежів, обміну даних з банком, адміністрування, розробка яких ведеться окремо згідно з водоспадною моделлю.

В ході виконання даної роботи були створені діаграми потоків даних нульового та першого рівня для системи поповнення мобільного рахунку та діаграми другого рівня для основних процесів – авторизації, створення та обробки платежів, а саме:

а) визначені зовнішні сутності:

- користувач;
- банк;
- оператор телефонного зв'язку;

б) визначені внутрішні накопичувачі:

- акаунти;
- сесії;
- номери мобільних телефонів;
- банківські картки;
- платежі;
- повідомлення (для неосновних процесів);

в) визначені підсистеми та відповідні підпроцеси:

- реєстрації;
- авторизації;
- обробки сесій;
- редагування профілю;
- створення та перегляду платежів;
- керування картками;
- керування номерами телефонів;
- надсилання повідомлень (неосновний, може бути використана стороння система)
- конфігурації системи (для адміністраторів інформаційної системи);

г) визначенні зв'язки між вищезазначеними об'єктами діаграми:

- надсилання реєстраційних форм, форм авторизації, запитів виходу;

- надсилання cookie-файлів;
- дані карток, телефонних номерів;
- запити на перегляд даних, їх представлення;
- відповіді системи на дії користувача – повідомлення, статуси;
- тощо.

В ході даного етапу визначені наступні сутності та їх первісні ключі (у дужках):

- акаунт користувача (логін);
- статус акаунту (назва статусу);
- номери мобільного телефону користувача (логін користувача та номер телефону);
- банківська картка (4 останні цифри номеру картки та логін);
- платіж (ідентифікатор платежу).

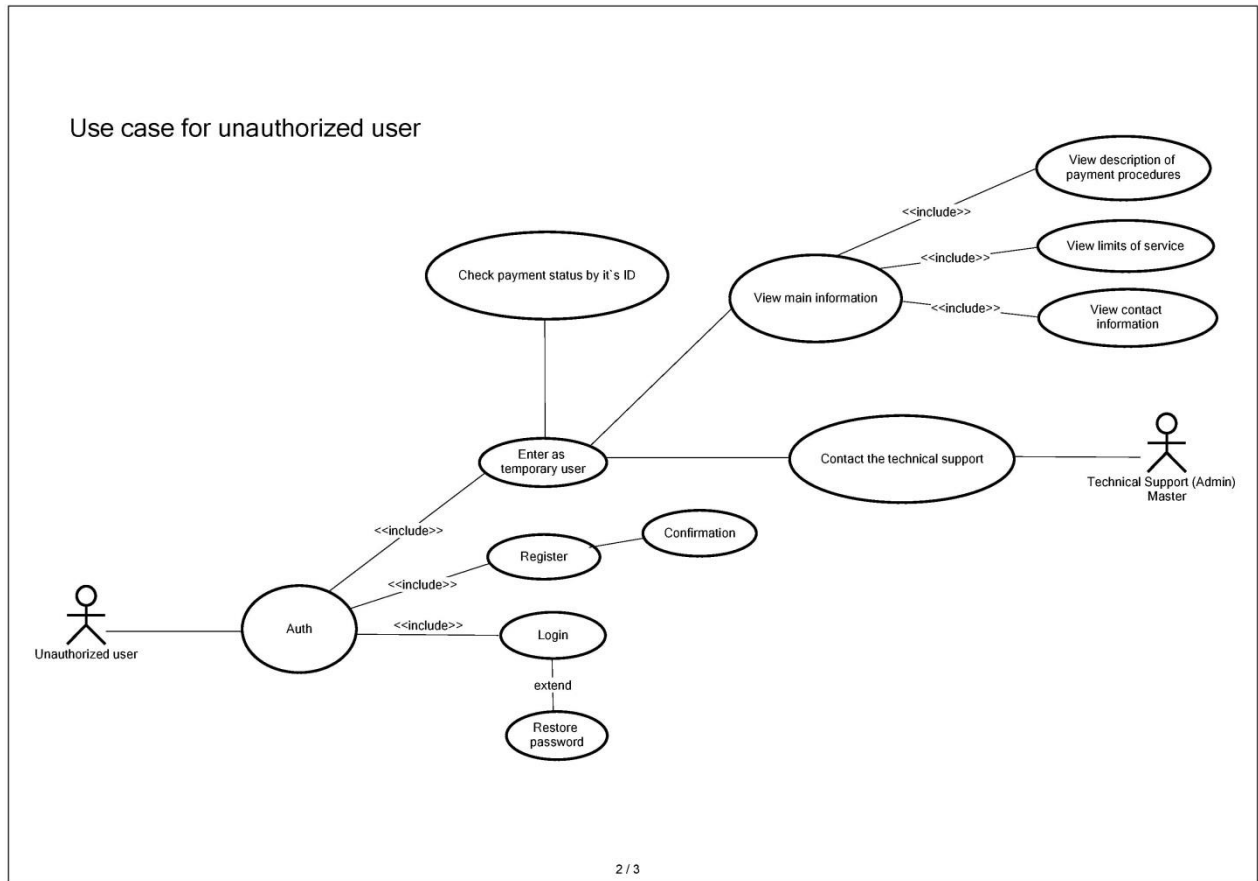
Між наступними сутностями існують зв'язки «один до багатьох»:

- між акаунтом і таблицею відповідності номерів телефонів акаунту;
- між акаунтом та банківськими картками;
- між акаунтом та платежами;
- між акаунтом і таблицею відповідності статусів акаунтам;
- між статусом акаунту та таблицею відповідності статусів акаунтам;
- між банківською карткою та платежами;
- між телефонами та платежами.

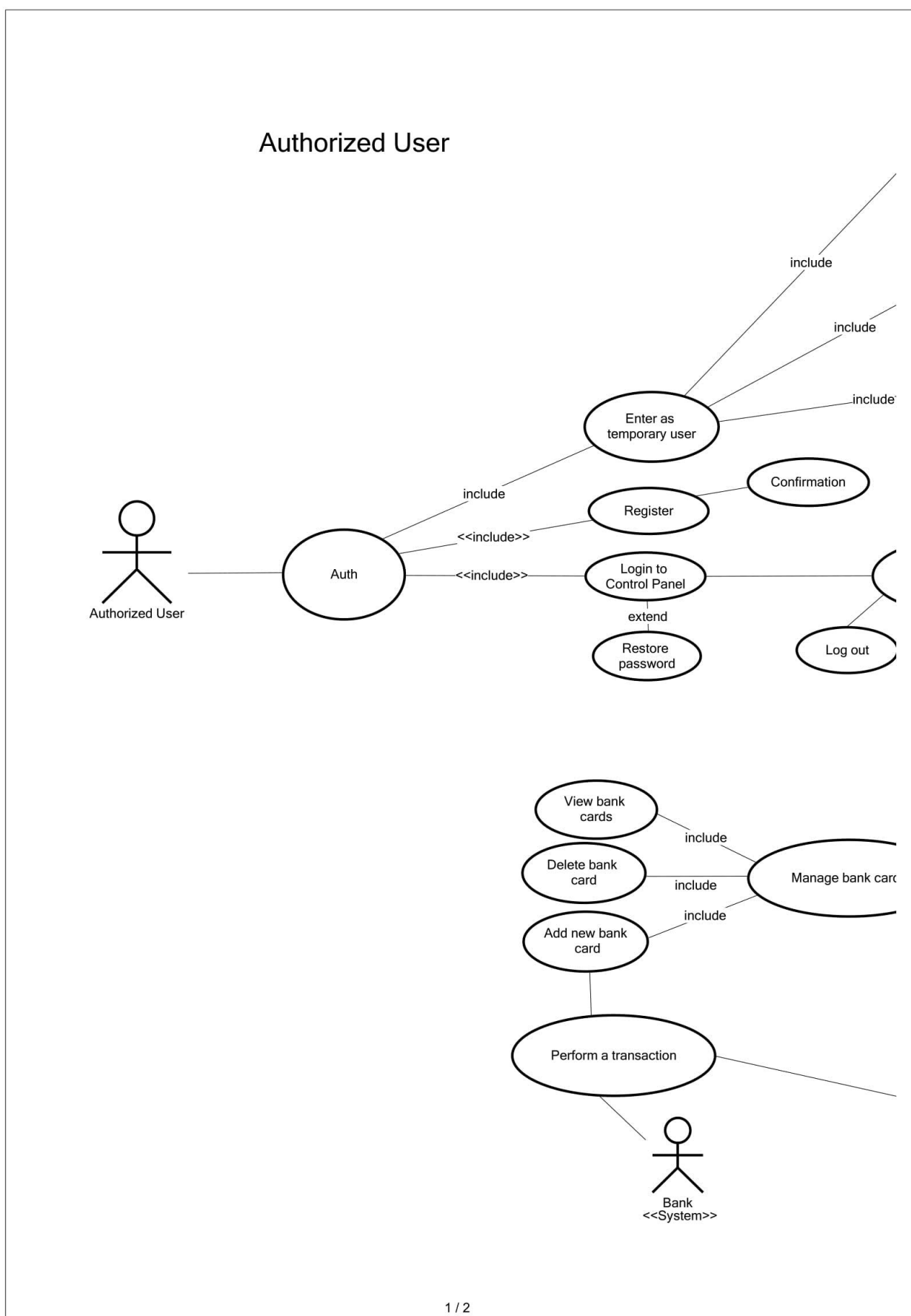
Було створено концептуальну, логічну та фізичну модель даних у системі.

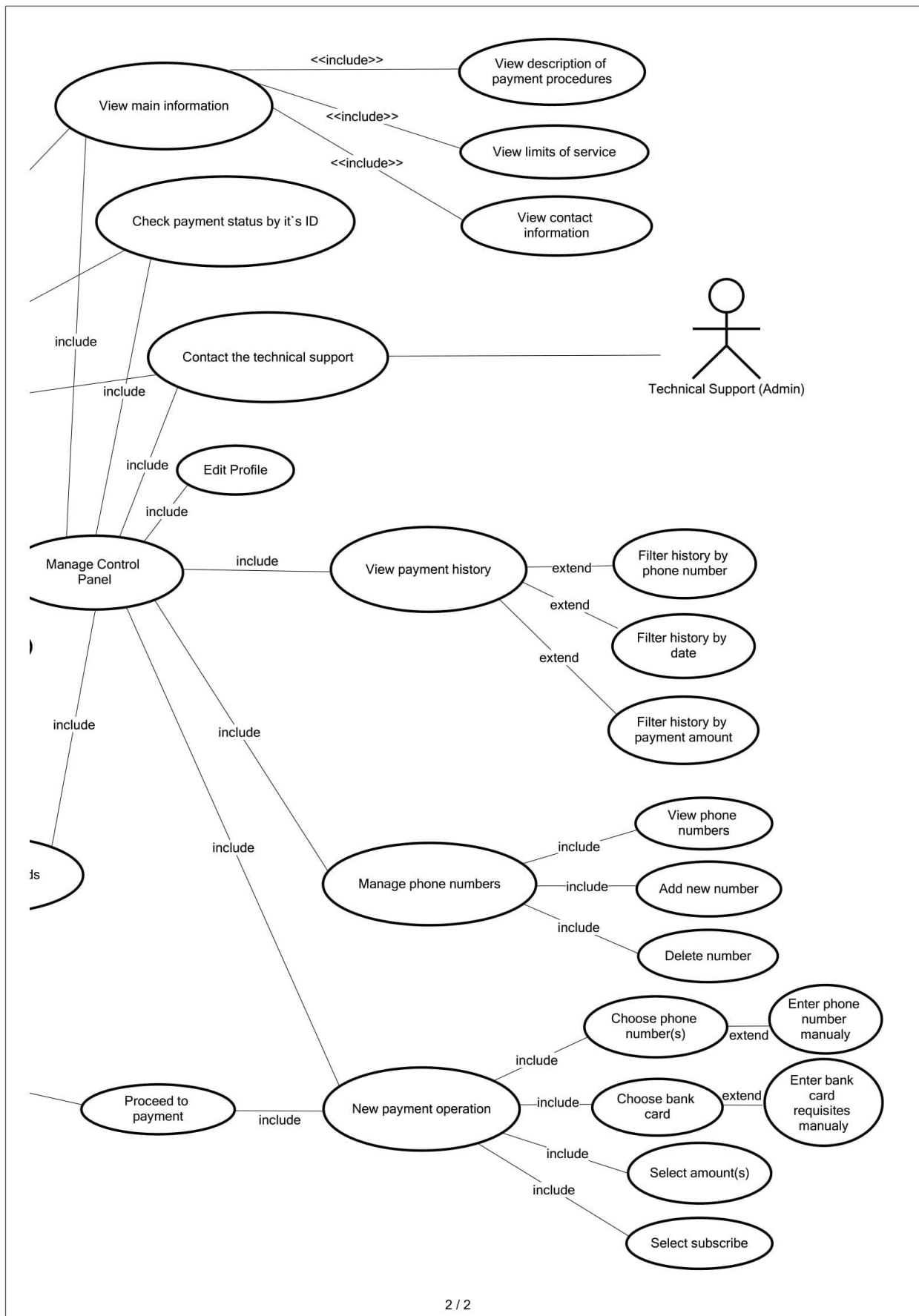
ДОДАТКИ

Додаток А. Діаграма прецедентів для неавторизованого користувача

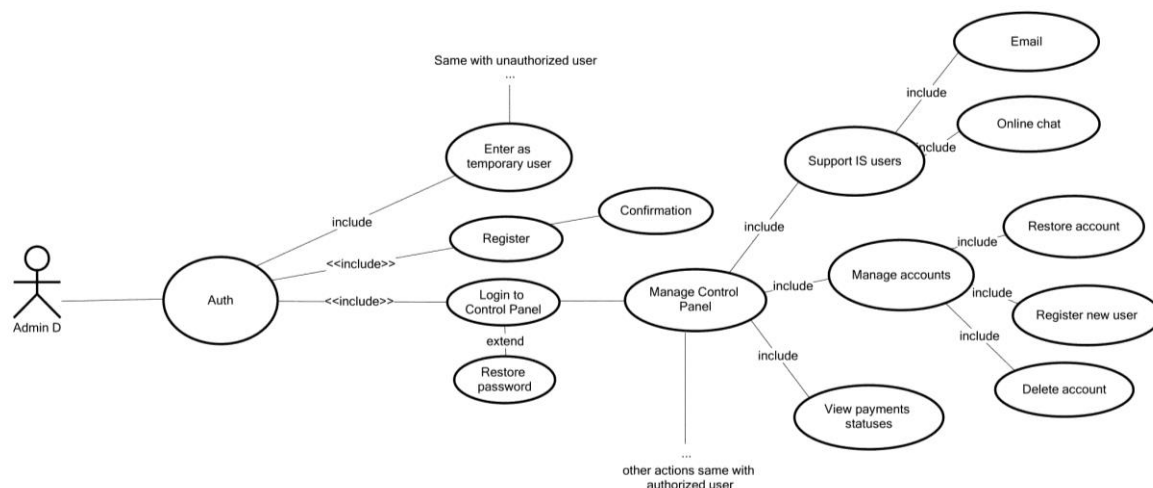


Додаток Б. Діаграма прецедентів для авторизованого користувача

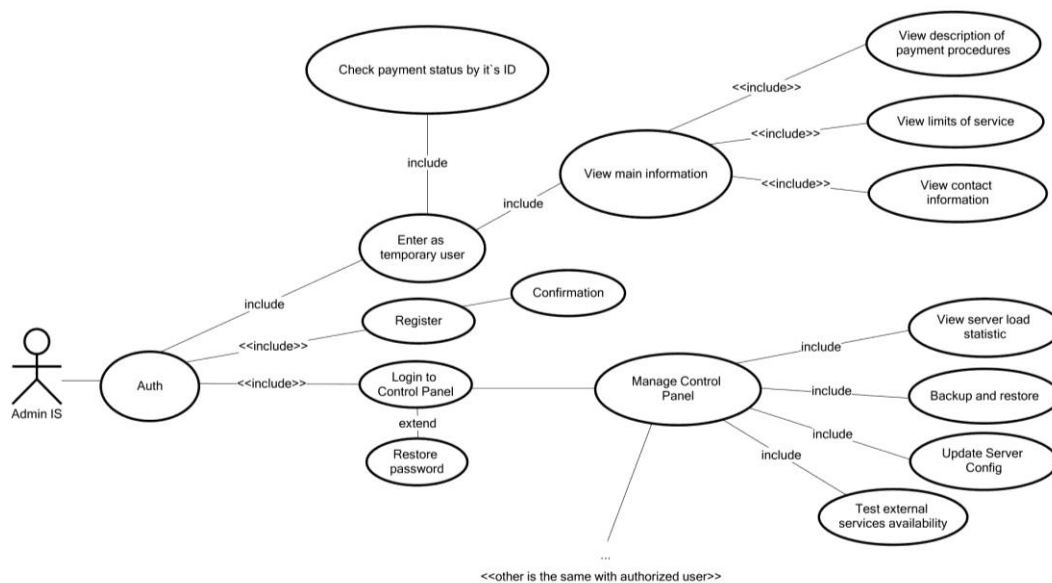




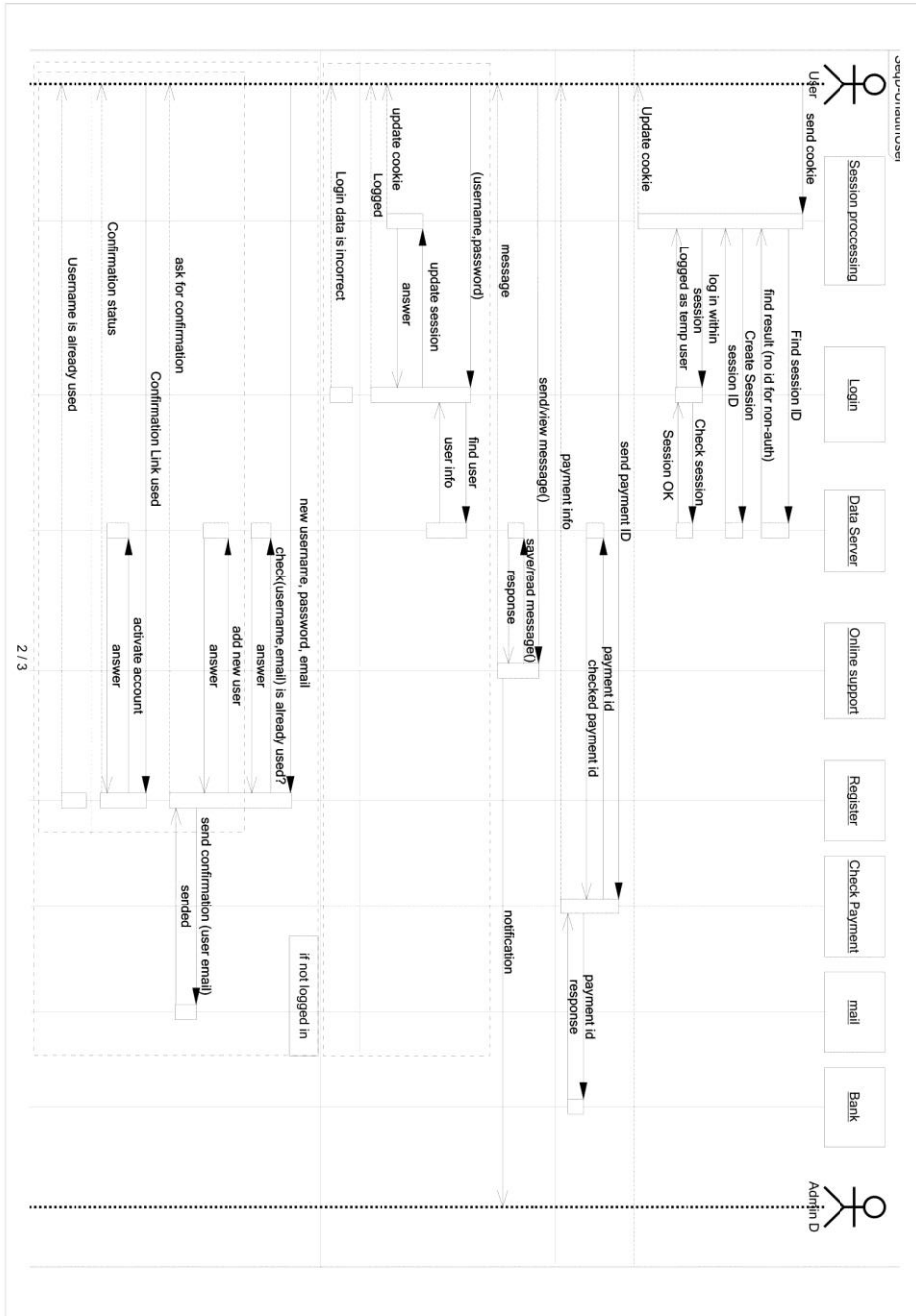
Додаток В. Діаграма прецедентів для адміністратора даних



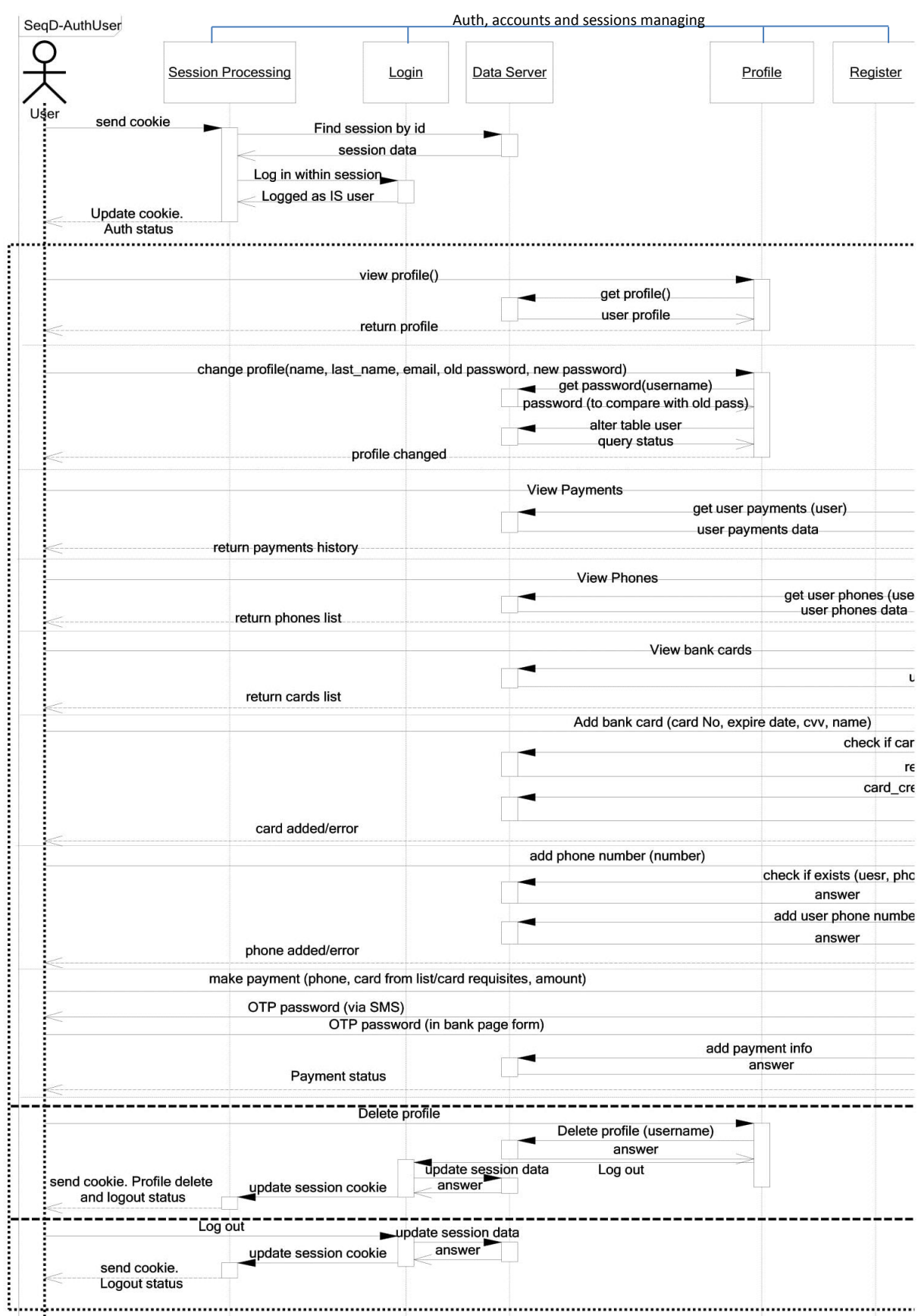
Додаток Г. Діаграма прецедентів для адміністратора ІС



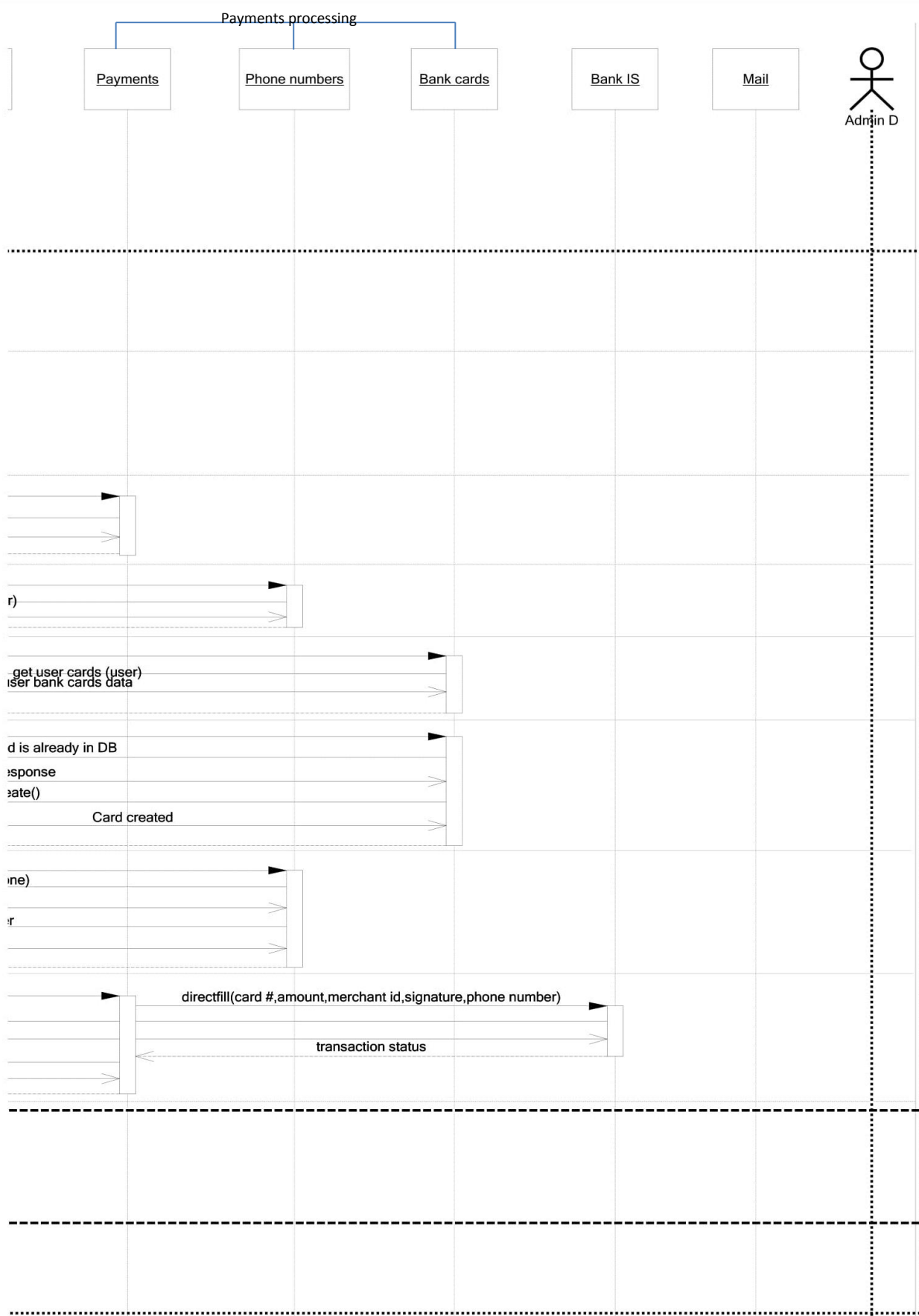
Додаток Д. Діаграма послідовності для групи неавторизованих користувачів



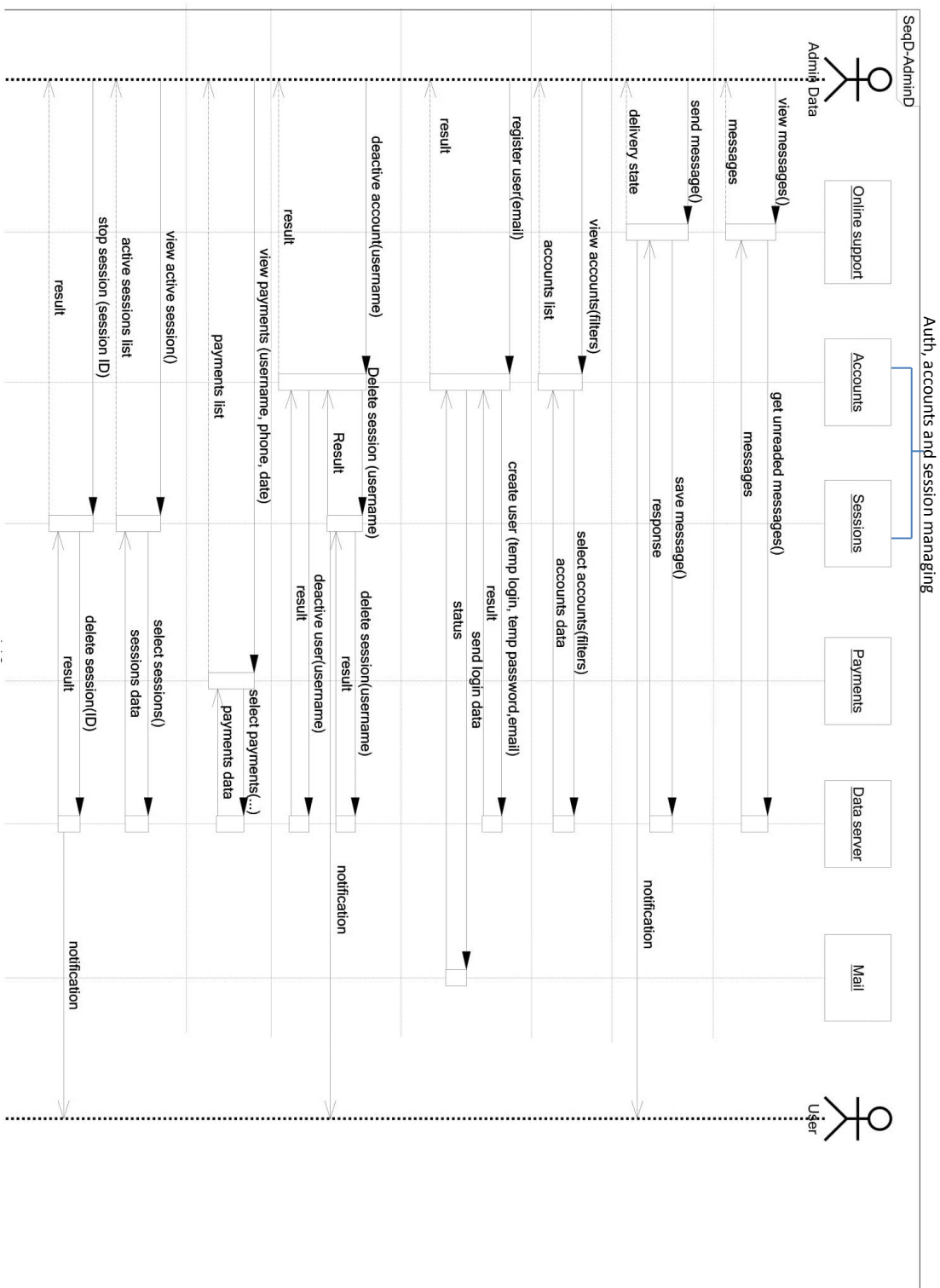
Додаток Е. Діаграма послідовності для групи авторизованих користувачів



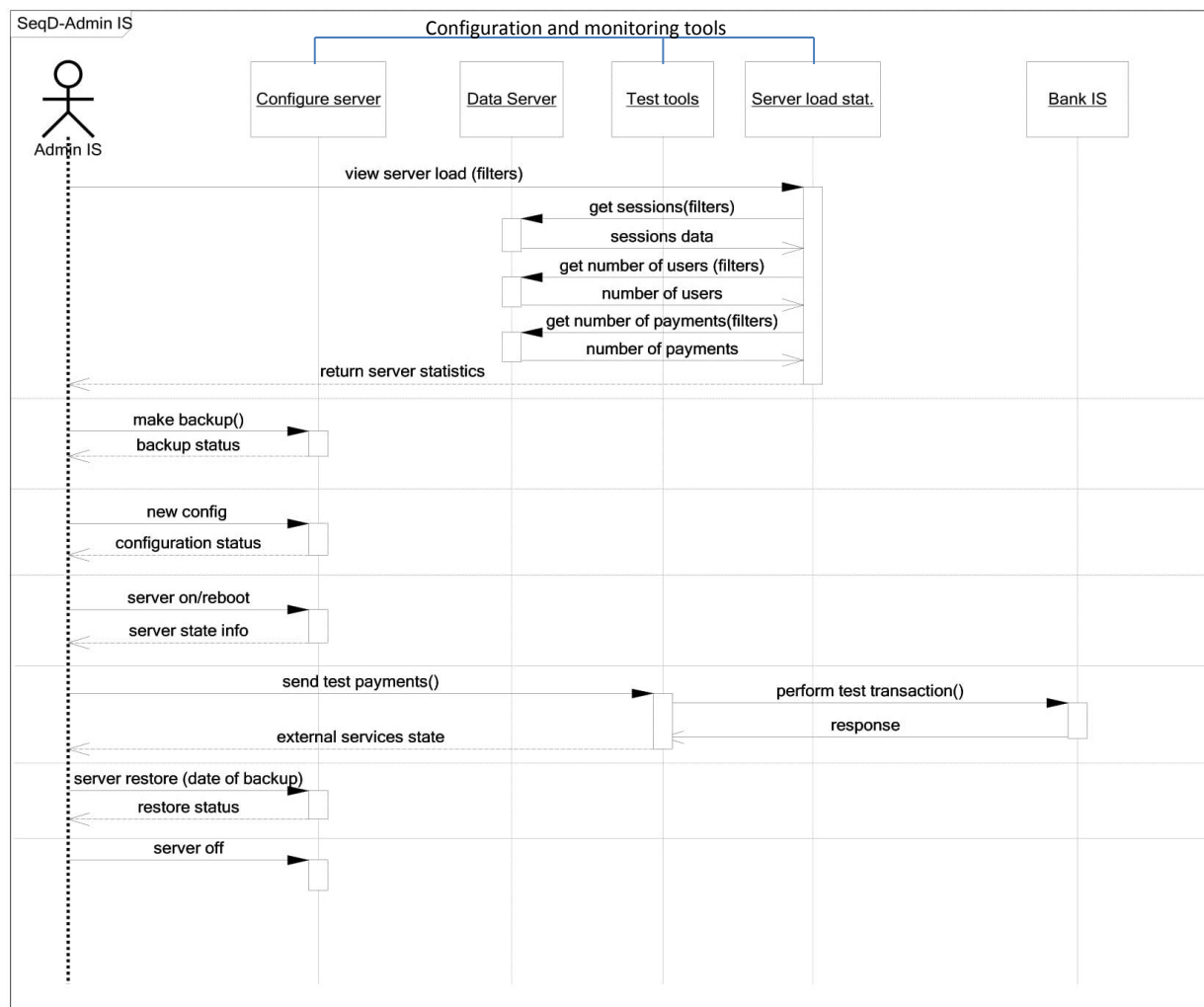
Продовження діаграми Е.



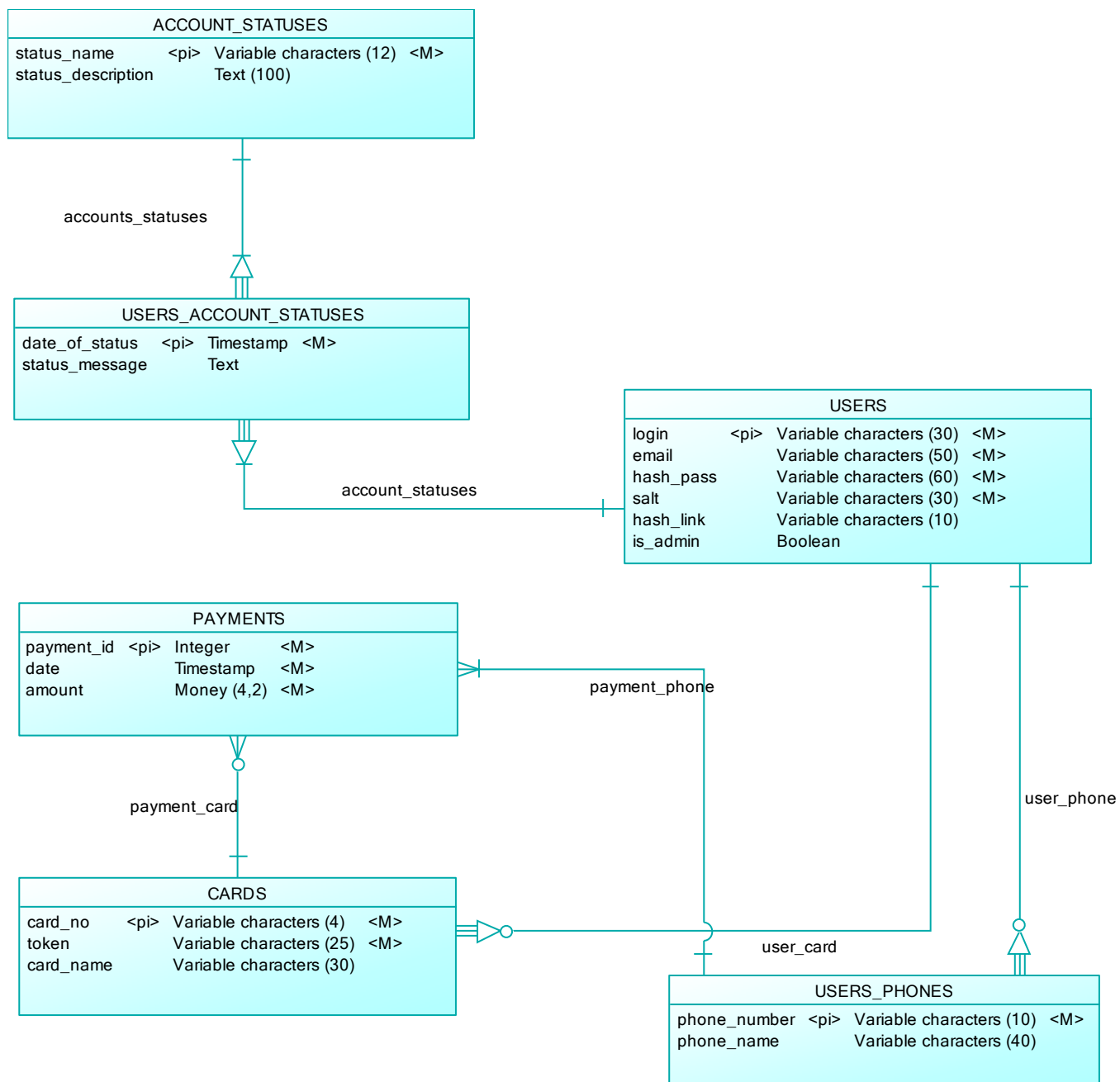
Додаток Є. Діаграма послідовності для групи адміністраторів даних



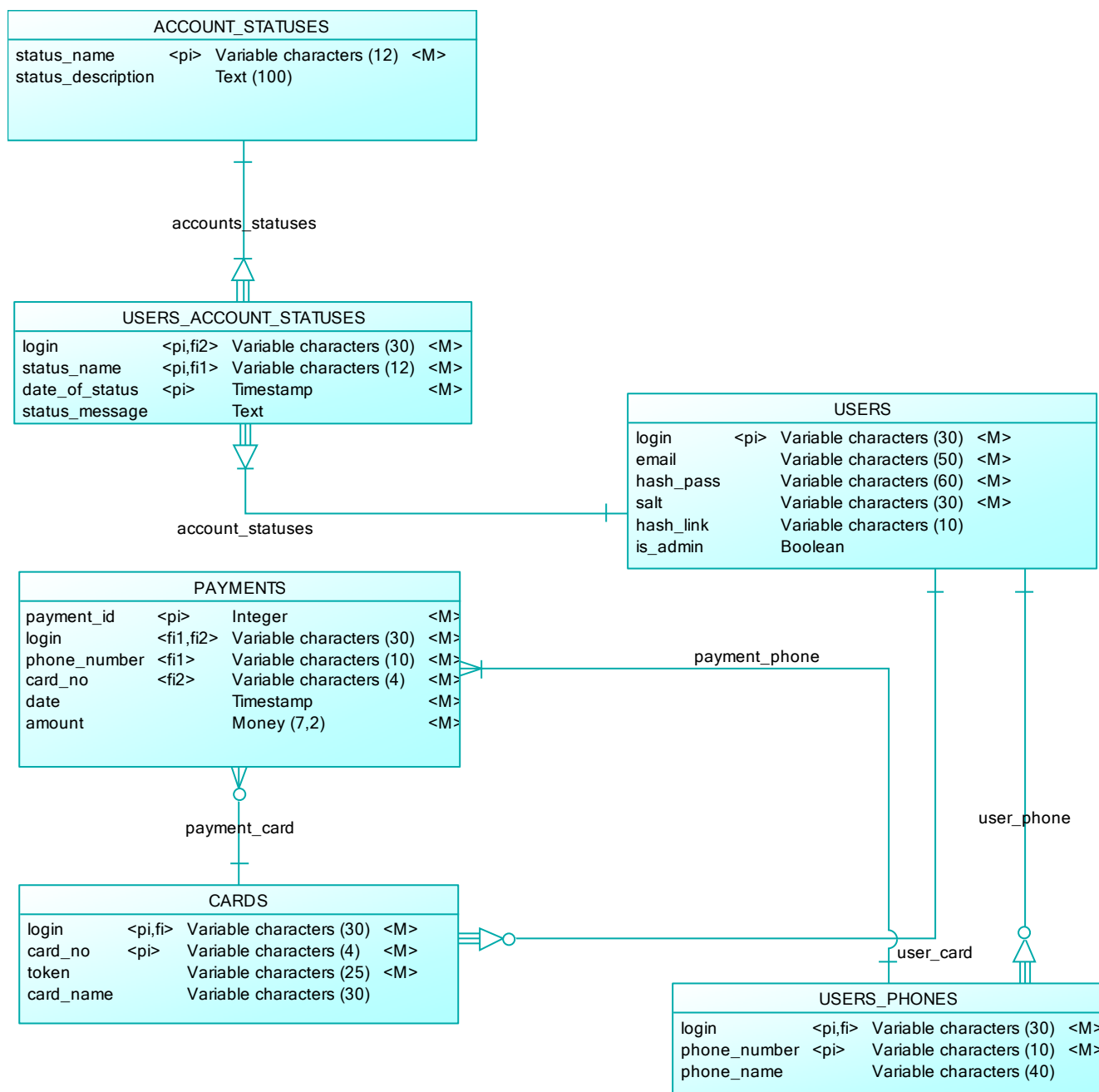
Додаток Ж. Діаграма послідовності для групи адміністраторів системи



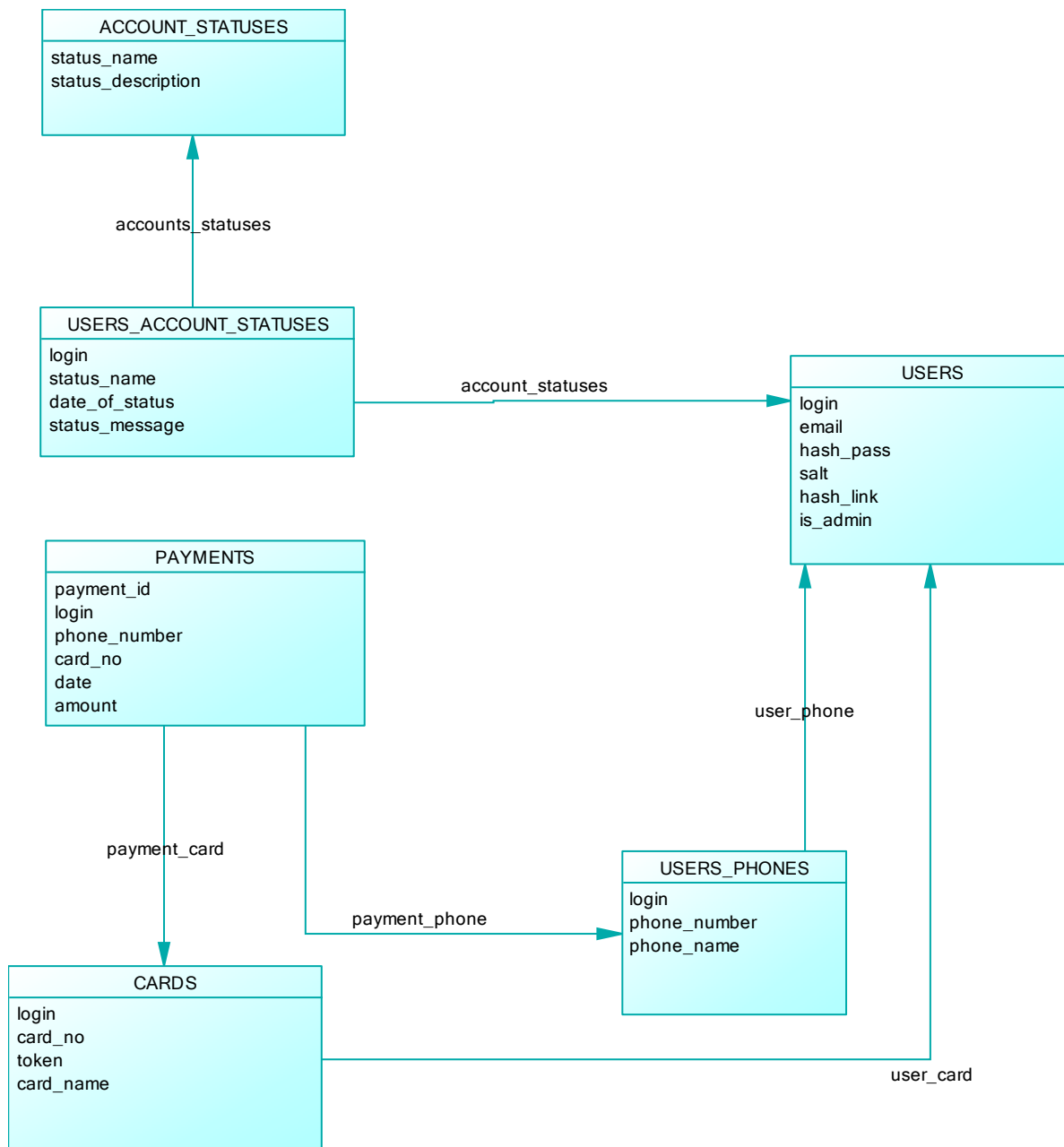
Додаток 3. Концептуальна ER-діаграма



Додаток И. Логічна ER-діаграма



Додаток І. Фізична ER-діаграма



Додаток І. Sql-скрипт генерації бази для СУБД Oracle 10g

```

alter table CARDS
  drop constraint FK_CARDS_USER_CARD_USERS;

alter table PAYMENTS
  drop constraint FK_PAYMENTS_PAYMENT_C_CARDS;

alter table PAYMENTS
  drop constraint FK_PAYMENTS_PAYMENT_P_USERS_PH;

alter table USERS_ACCOUNT_STATUSES
  drop constraint FK_USERS_AC_ACCOUNT_S_USERS;

alter table USERS_ACCOUNT_STATUSES
  drop constraint FK_USERS_AC_ACCOUNTS__ACCOUNT_;

alter table USERS_PHONES
  drop constraint FK_USERS_PH_USER_PHON_USERS;

drop table ACCOUNT_STATUSES cascade constraints;

drop index user_card_FK;

drop table CARDS cascade constraints;

drop index payment_card_FK;

drop index payment_phone_FK;

drop table PAYMENTS cascade constraints;

drop table USERS cascade constraints;

drop index account_statuses_FK;

drop index accounts_statuses_FK;

drop table USERS_ACCOUNT_STATUSES cascade constraints;

drop index user_phone_FK;

drop table USERS_PHONES cascade constraints;

create table ACCOUNT_STATUSES
(
  status_name          VARCHAR2(12)          not null,
  status_description   CLOB,
  constraint PK_ACCOUNT_STATUSES primary key (status_name)
);

create table CARDS
(
  login                VARCHAR2(30)          not null,
  card_no              VARCHAR2(4)           not null,
  token                VARCHAR2(25)          not null,
  card_name            VARCHAR2(30),
  constraint PK_CARDS primary key (login, card_no)

```

```

);

create index user_card_FK on CARDS (
    login ASC
);

create table PAYMENTS
(
    payment_id          INTEGER          not null,
    login               VARCHAR2(30)      not null,
    phone_number        VARCHAR2(10)      not null,
    card_no             VARCHAR2(4)       not null,
    paydate             TIMESTAMP         default current_timestamp,
    amount              NUMBER(7,2)       not null,
    constraint PK_PAYMENTS primary key (payment_id)
);

create index payment_phone_FK on PAYMENTS (
    login ASC,
    phone_number ASC
);

create index payment_card_FK on PAYMENTS (
    login ASC,
    card_no ASC
);

create table USERS
(
    login               VARCHAR2(30)      not null,
    email               VARCHAR2(50)      not null,
    hash_pass           VARCHAR2(60)      not null,
    salt                VARCHAR2(30)      not null,
    hash_link           VARCHAR2(10),
    is_admin             SMALLINT,
    constraint PK_USERS primary key (login)
);

create table USERS_ACCOUNT_STATUSES
(
    login               VARCHAR2(30)      not null,
    status_name         VARCHAR2(12)      not null,
    date_of_status      TIMESTAMP         default current_timestamp,
    status_message      CLOB,
    constraint PK_USERS_ACCOUNT_STATUSES primary key (login, status_name,
date_of_status)
);

create index accounts_statuses_FK on USERS_ACCOUNT_STATUSES (
    status_name ASC
);

create index account_statuses_FK on USERS_ACCOUNT_STATUSES (
    login ASC
);

create table USERS_PHONES
(
    login               VARCHAR2(30)      not null,

```

```

    phone_number          VARCHAR2(10)          not null,
    phone_name            VARCHAR2(40),
    constraint PK_USERS_PHONES primary key (login, phone_number)
);

create index user_phone_FK on USERS_PHONES (
    login ASC
);

alter table CARDS
    add constraint FK_CARDS_USER_CARD_USERS foreign key (login)
        references USERS (login);

alter table PAYMENTS
    add constraint FK_PAYMENTS_PAYMENT_C_CARDS foreign key (login, card_no)
        references CARDS (login, card_no);

alter table PAYMENTS
    add constraint FK_PAYMENTS_PAYMENT_P_USERS_PH foreign key (login, phone_number)
        references USERS_PHONES (login, phone_number);

alter table USERS_ACCOUNT_STATUSES
    add constraint FK_USERS_AC_ACCOUNT_S_USERS foreign key (login)
        references USERS (login);

alter table USERS_ACCOUNT_STATUSES
    add constraint FK_USERS_AC_ACCOUNTS__ACCOUNT_ foreign key (status_name)
        references ACCOUNT_STATUSES (status_name);

alter table USERS_PHONES
    add constraint FK_USERS_PH_USER_PHON_USERS foreign key (login)
        references USERS (login);

```