

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра програмних систем і технологій

Дисципліна
«АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ»

Лабораторна робота № 6
«СПЕЦИФІКАЦІЯ ТА ВІДСТЕЖЕННЯ ВИМОГ»
на тему:
«Автоматизована система безготівкових електронних платежів»

Виконав:	Гоша Д.О	Перевірів:	Юрчук І. А.
Група	ІПЗ-23	Дата перевірки	
Форма навчання	Денна	Бали	
Спеціальність	121		
2021			

Мета: Дослідити процес аналізу вимог за допомогою діаграм потоків даних та набути практичних навичок у використанні інструментальних засобів Enterprise Architect і Microsoft Visio при їх побудові.

Завдання:

1. На основі технічного завдання з лабораторної роботи № 2.1 виконати аналіз функціональних та експлуатаційних вимог до програмного продукту;
2. Визначити основні технічні рішення (вибір мови програмування, структура програмного продукту, склад функцій ПП, режими функціонування) і занести результати в документ, який називається «Ескізний проект»;
3. Побудувати діаграми прецедентів (варіантів використання) системи, що задана згідно з варіантом (додаток 1) .
4. Навести вербальний опис усіх варіантів використання системи в текстовому редакторі MS Word.
5. Визначити та побудувати діаграми потоків даних для розв'язуваної завдання;
6. Визначити та побудувати діаграми «сутність-зв'язок», якщо програмний продукт містить базу даних;
7. Визначити та побудувати функціональні діаграми;
8. Визначити та побудувати діаграми переходів станів;
9. Визначити та побудувати специфікації процесів;
10. Додати словник термінів;
11. Оформити результати, використовуючи MS Office або MS Visio в вигляді ескізного проекту.

Виконання завдання:

1. Аналіз функціональних та експлуатаційних вимог до програмного продукту:

Функціональні вимоги - це перелік функцій або сервісів, які повинна надавати система (ISO/IEC ДСТУ 9126 гост), а також обмежень на дані і поведінку системи при їхньому виконанні. Специфікація функціональних вимог - опис функцій та їхніх властивостей, які не містять у собі протиріч і виключень. Серед них визначимо такі вимоги:

- 1.1. Створення транзакції.
- 1.2. Прийом транзакції.
- 1.3. Внутрішній обмін валют.
- 1.4. Створення облікового запису.
- 1.5. Друк чеку.

Експлуатаційні вимоги визначають характеристики програмного забезпечення, які проявляються в процесі його використання. До таких характеристик належать:

- 1.6. Правильність – функціонування у відповідності з технічним завданням. Ця вимога є обов'язковою для всякого програмного продукту, але оскільки ніяке тестування не дає гарантії 100%-ої правильності, мова може йти про деяку ймовірність наявності помилок. Ймовірність збою системи управління космічними

польотами повинна бути близька до нуля;

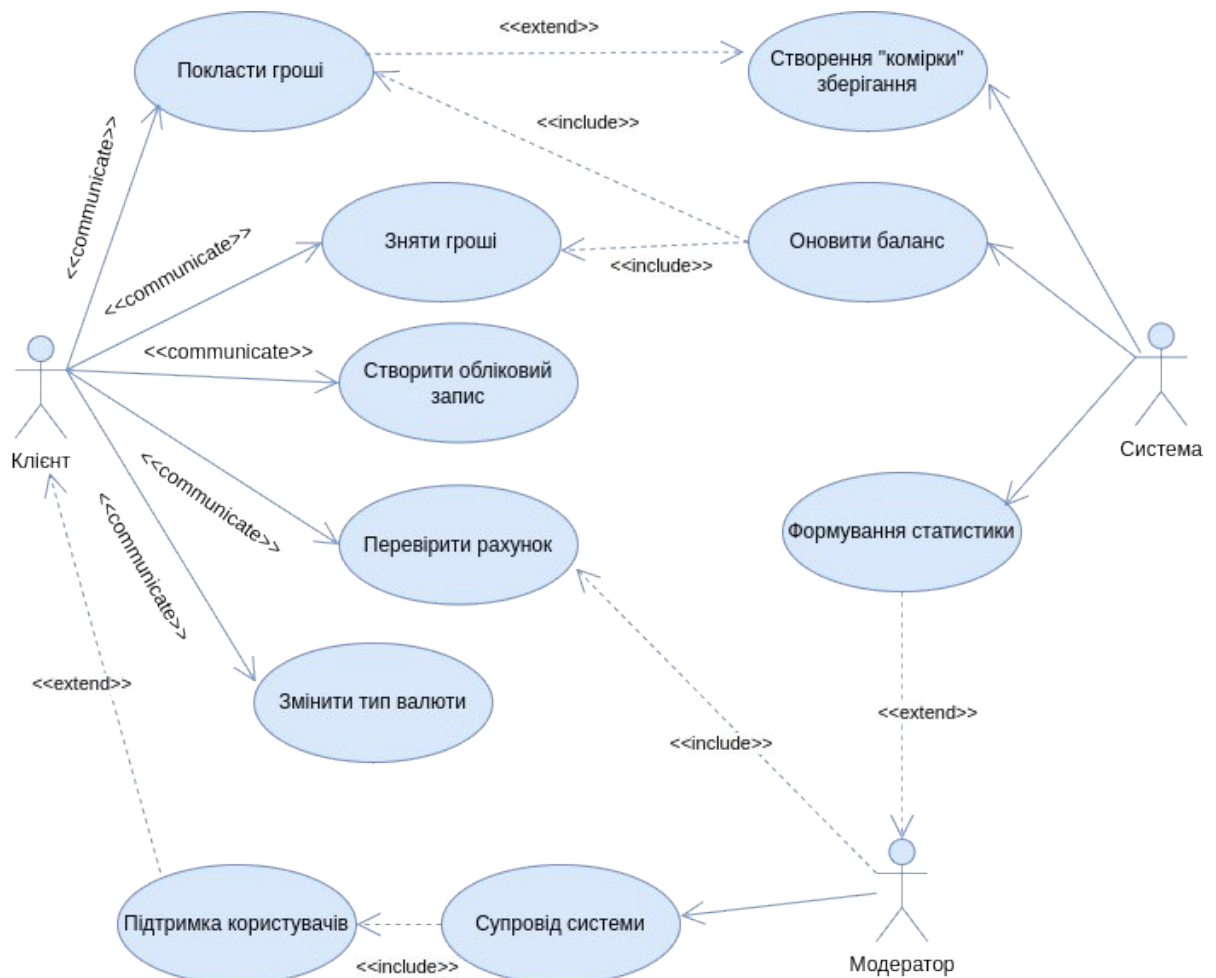
- 1.7. Універсальність – забезпечення правильності роботи при довільних допустимих даних і захист від неправильних даних. Так як в попередньому випадку, довести універсальність програми неможливо, тому є зміст говорити про степінь її універсальності;
- 1.8. Надійність – забезпечення повної повторюваності результатів, тобто забезпечення їх правильності при наявності різного роду збою. Джерелами завад можуть бути технічні і програмні засоби, а також люди, які працюють з цими засобами. Зараз існує достатня кількість способів запобігти втрат інформації при збоях. Наприклад, прийом «створення контрольних точок», при якому зберігаються проміжні результати, що дозволяє після збою програми продовжити роботу з даними, записаними в останній контрольній точці. Можливо також зменшити кількість помилок, використовуючи дублювання систем чи введення надлишкової інформації;
- 1.9. Перевірюваність – можливість перевірки результатів. Для цього необхідно документально фіксувати початкові дані, встановлені режими та іншу інформацію, яка впливає на результати. Особливо це проявляється, коли сигнали поступають безпосередньо від датчиків;
- 1.10. Точність результатів – забезпечення похибки результатів не вище заданої. Величина похибки залежить від точності початкових даних, степені адекватності моделі, точності вибраного методу і похибки виконання операцій на комп'ютері. Жорсткі вимоги до точності пред'являють системи навігації (наприклад, система стиковки космічних апаратів) і системи управління технологічними процесами;
- 1.11. Захищеність – забезпечення конфіденційності інформації. Найбільш жорсткі вимоги пред'являються до систем, в яких зберігається інформація, пов'язана з державною і комерційною таємницею. Для забезпечення захисту інформації використовують програмні, криптографічні, правові та інші методи;
- 1.12. Програмна сумісність – можливість сумісного функціонування з іншим програмним забезпеченням. Частіше всього в даному випадку мова йде про функціонування програми під управлінням заданої операційної системи. Однак може бути потрібний обмін даними з деякою іншою програмою. У цьому випадку точно обговорюється формат даних.
- 1.13. Апаратна сумісність – можливість сумісного функціонування з деяким обладнанням. Цю вимогу формують у вигляді мінімально можливої конфігурації обладнання, на якому буде працювати дане програмне забезпечення. Якщо припускається використання нестандартного обладнання, то для нього повинні бути описані інтерфейси;
- 1.14. Ефективність – використання мінімально можливої кількості ресурсів технічних засобів (наприклад, часу мікропроцесора, об'єму оперативної пам'яті, об'єму зовнішньої пам'яті, кількості зовнішніх

пристроїв та ін.). Ефективність оцінюється по кожному ресурсу окремо, тому вимоги ефективності часто суперечать одна одній. Наприклад, щоб зменшити час виконання програми, необхідно збільшити об'єм оперативної пам'яті;

- 1.15. Адаптованість – можливість швидкої модифікації з ціллю пристосування до змінних умов функціонування. Оцінити цю характеристику кількісно практично неможливо. Можливо тільки констатувати, що при розробці даного ПЗ використовувались прийоми, які полегшують його модернізацію;
- 1.16. Повторне входження – можливість повторного виконання без пере загрузки з диску. Дана вимога зазвичай пред'являється до програмного забезпечення, резидентно завантаженому в оперативну пам'ять (наприклад, драйвери);
- 1.17. Реєнтерабельність – можливість «паралельного» використання декількома процесами. Щоб задовольнити цю вимогу, необхідно створювати копію даних, що змінюються програмою, для кожного процесу.

2. Визначення основних технічних рішень:

- 2.1. Середовище розробки Visual Studio 2022
- 2.2. Сервер бд MS SQL Server 2019
- 2.3. Мова програмування C#
- 2.4. Entity Framework Core + Migrations + Identity
- 2.5. Тип застосунку ASP.NET Core MVC 3.1
- 2.6. HTML5 + SASS (CSS)
- 2.7. JavaScript + jQuery



3.

Діаграма прецедентів 1

4. Вербальні специфікації прецедентів:

1. Перевірка рахунку:

4.1.Короткий опис - Користувач має змогу перевірити свій баланс заощаджень;

4.2.Суб'єкт - Користувач;

4.3.Передумова - Користувач хоче дізнатися залишок на балансі;

4.4.Основний потік:

4.4.1. Користувач переходить на сайт:

4.4.1.1. Браузер відображає головну сторінку сайту;

4.4.2. Якщо нічого не відбувається виконується А1.

4.5.Альтернативні потоки: А1. Не надійне з'єднання, вірус, або технічні роботи. В перших 2 випадках браузер сам повідомляє про помилку, в останньому - проблема системи (технічні роботи).

4.6.Постумова - Користувач має змогу переглядати баланс, дізнатися відправника, суми транзакцій, тощо;

2. Створення транзакції:

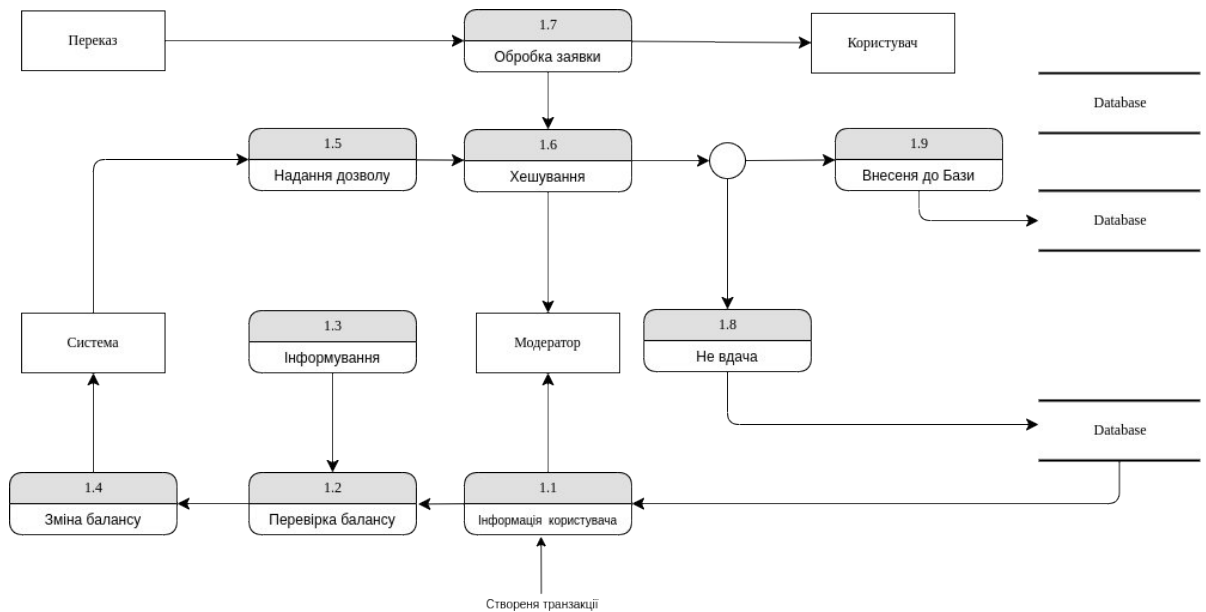
4.7.Короткий опис - Користувач має змогу створити переказ іншому користувачу;

4.8.Суб'єкт – Користувач;

- 4.9. Передумова - Користувач хоче перерахувати певну кількість коштів отримувачу;
 - 4.10. Основний потік:
 - 4.10.1. Користувач переходить на сайт:
 - 4.10.1.1. Браузер відображає сторінку переказу;
 - 4.10.1.2. Користувач заповнює дані та суму переказу;
 - 4.10.1.3. Користувач натискає кнопку оформити замовлення, система зберігає інформацію в бд;
 - 4.10.2. Якщо форма з неправильними, або не повними даними, виконується А1. Якщо нічого не відбувається виконується А2.
 - 4.11. Альтернативні потоки:
 - А1. Не валідна, або не повна інформація. В разі не валідних даних або неповної інформації, система повідомляє це користувачеві повідомленнями, та дає змогу виправити дані;
 - А2. Не надійне з'єднання, вірус, або технічні роботи. В перших 2 випадках браузер сам повідомляє про помилку, в останньому - проблема системи (технічні роботи).
 - 4.12. Постумова - Користувач має змогу створити транзакцію перерахувати певну кількість коштів отримувачу;
3. Супровід системи:
- 4.13. Короткий опис - Модератор підтримує життєдіяльність системи, в разі проблем знешкоджує їх;
 - 4.14. Суб'єкт - Модератор ;
 - 4.15. Передумова - Модератор вирішує проблеми всіх працівників, супроводжує систему, та гарантує надійність;
 - 4.16. Основний потік:
 - 4.16.1. Забезпечує надійність системи, та піклується про працездатність працівників та системи;
 - 4.16.2. Супроводжує систему;
 - 4.16.3. Забезпечує надійність системи, в разі помилок виконується А1;
 - 4.16.4. Забезпечує працездатність працівників.
 - 4.17. Альтернативні потоки:
 - А1. Надійність системи. В разі помилок, системний адміністратор повинен проаналізувати та знайти помилку і якнайшвидше її виправити.
 - 4.18. Постумова - Модератор успішно супроводжує систему, та забезпечує її надійність;
4. Зміна типу валюти;
- 4.19. Короткий опис – Користувач змінює системний баланс у розмірі ішої валюти;
 - 4.20. Суб'єкт - Користувач;
 - 4.21. Передумова – користувач хоче змінити тип валюти , наприклад для оплати за кородоном;
 - 4.22. Основний потік:

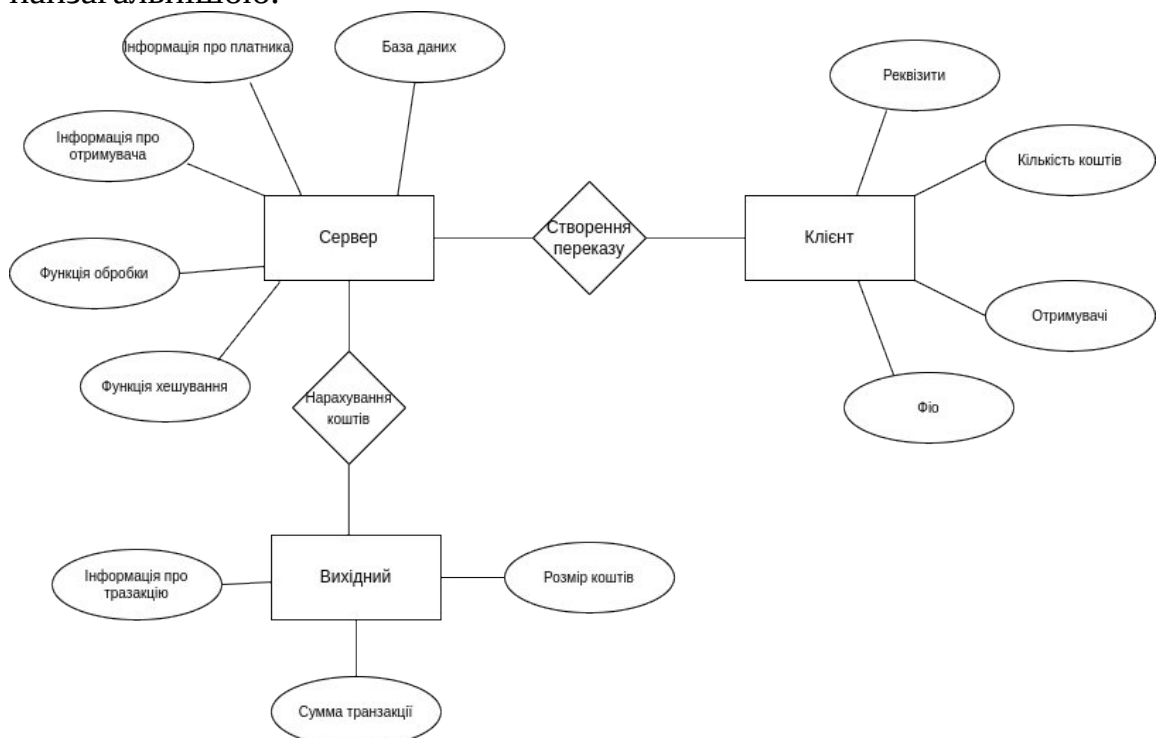
- 4.22.1. Користувач переходить на сторінку застосунку "Налаштування"
- 4.22.2. Змінює тип валюти на один із доступних системи Виконується А1;
- 4.22.3. Альтернативні потоки:
 - А1. Зміна інтерфейсу на ішну валюту, змога перерказів за кордон , подвійна конвертація валюти.
- 4.23. Постумова – Користувач змінив тип валюти;
- 5. Надання статистики:
 - 4.24. Короткий опис - Система має змогу переглядати кількість замовлень на всю продукцію магазину, та надає статистику для формування звіту бухгалтеру;
 - 4.25. Суб'єкт - Система;
 - 4.26. Передумова - Система повинна зробити аналітику замовлень магазину, та передати інформацію бухгалтеру;
 - 4.27. Основний потік:
 - 4.27.1. Система переглядає потрібну інформацію в БД:
 - 4.27.1.1. разі недостовірних даних або їх втрати виконується А1.
 - 4.27.2. Менеджер формує звіт;
 - 4.27.3. Альтернативні потоки:
 - А1. Недостовірні дані або втрата. Інформація передається системному адміністратору який вирішує проблему.
 - 4.27.4. Постумова - Система надає інформацію бухгалтеру, який формує звіт;
- 5. Діаграма потоків:

Така діаграма складається з трьох типів вузлів: вузлів обробки даних, вузлів збереження даних і зовнішніх вузлів, які представляють зовнішні по відношенню до використовуваної діаграми джерела чи користувачі даних. Дуги в діаграмі відповідають потокам даних, які передаються від вузла до вузла. Вони помічені іменами відповідних даних. Описання процесу, функції чи системи обробки даних, які відповідають вузлу діаграми, може бути представлено діаграмою наступного рівня деталізації, якщо процес достатньо складний.



Діаграма потоків 1

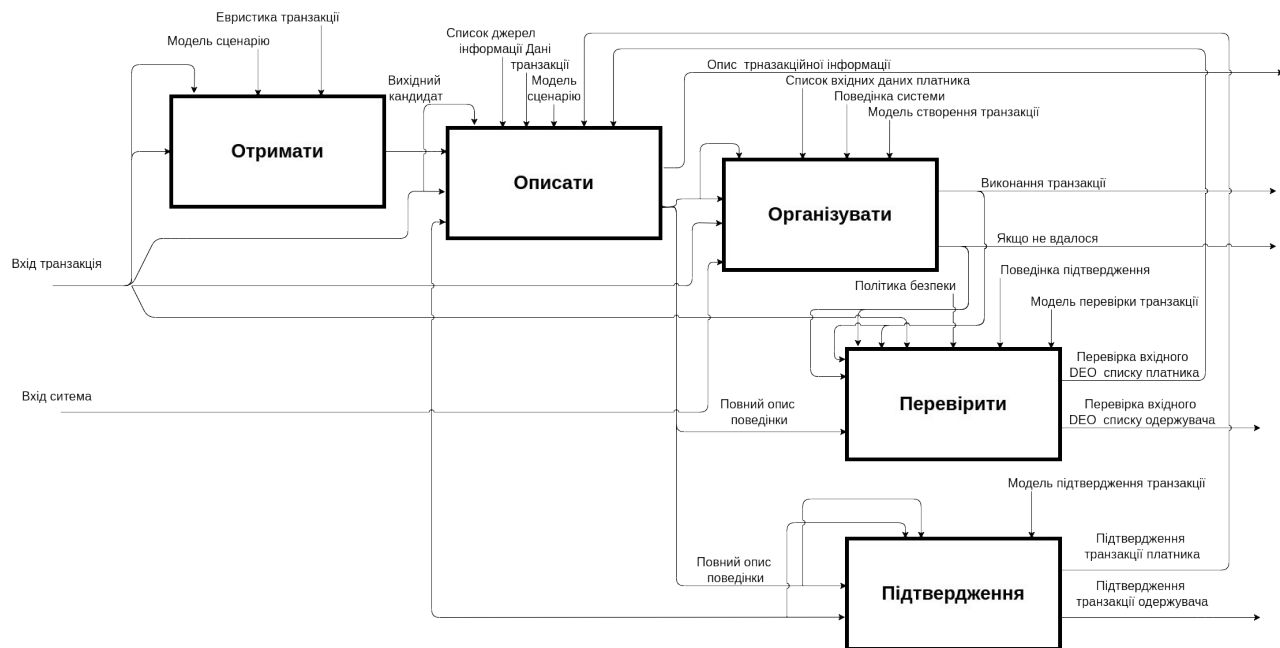
6. Модель «сутність-зв'язок» — модель даних, яка дозволяє описувати концептуальні схеми за допомогою узагальнених конструкцій блоків. ER-модель — це мета-модель даних, тобто засіб опису моделей даних. Існує ряд моделей для представлення знань, але одним з найзручніших інструментів уніфікованого представлення даних, незалежного від програмного забезпечення, що його реалізує, є модель «сутність-зв'язок». Важливим є той факт, що з моделі «сутність-зв'язок» можуть бути породжені всі існуючі моделі даних (ієрархічна, мережева, реляційна, об'єктна), тому вона є найзагальнішою.



Діаграма сутність-зв'язок 1

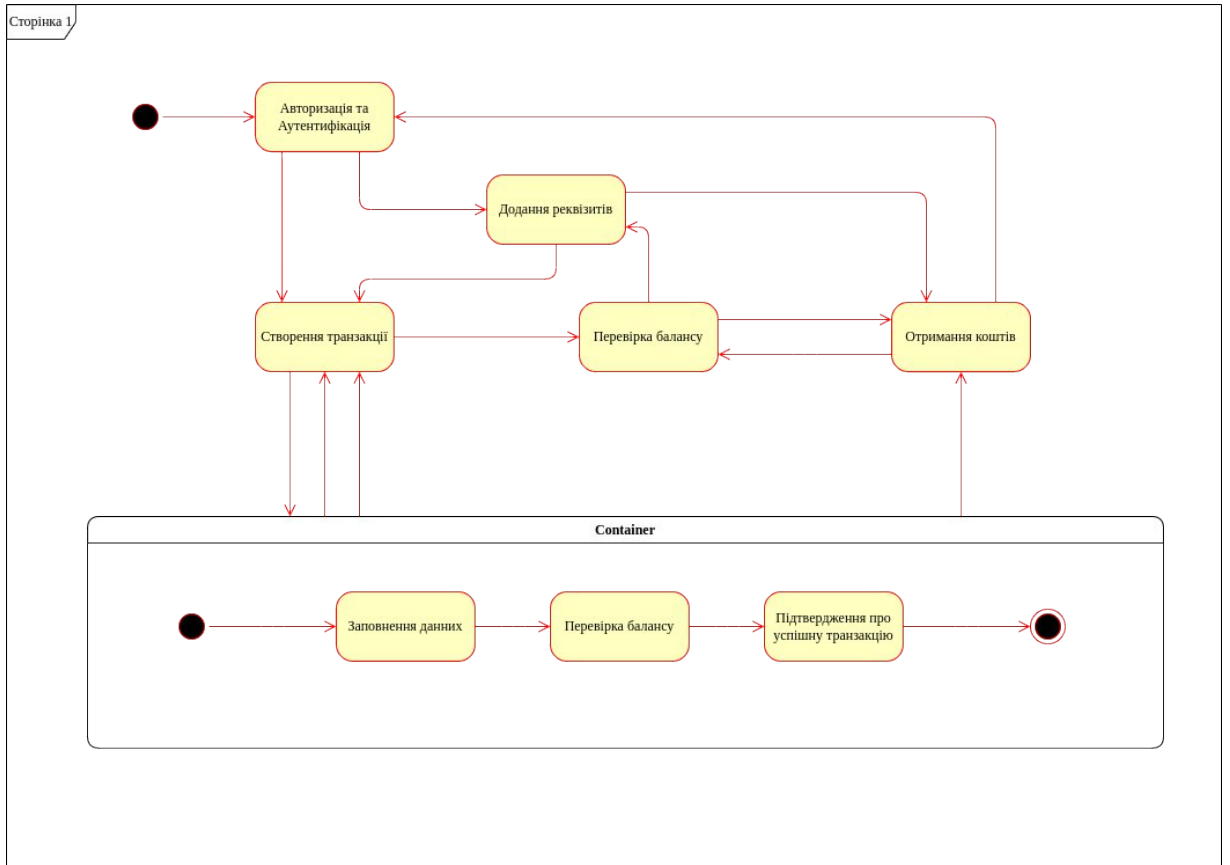
7. SADT - методологія структурного аналізу та проектування, що інтегрує процес моделювання, управління конфігурацією проекту, використання

додаткових мовних засобів та керівництво проектом зі своєю графічною мовою. Процес моделювання може бути поділений на кілька етапів: опитування експертів, створення діаграм та моделей, поширення документації, оцінка адекватності моделей та прийняття їх для подальшого використання. Цей процес добре налагоджений, тому що під час розробки проекту фахівці виконують конкретні обов'язки, а бібліотекар забезпечує своєчасний обмін інформацією.



Діаграма SADT 1

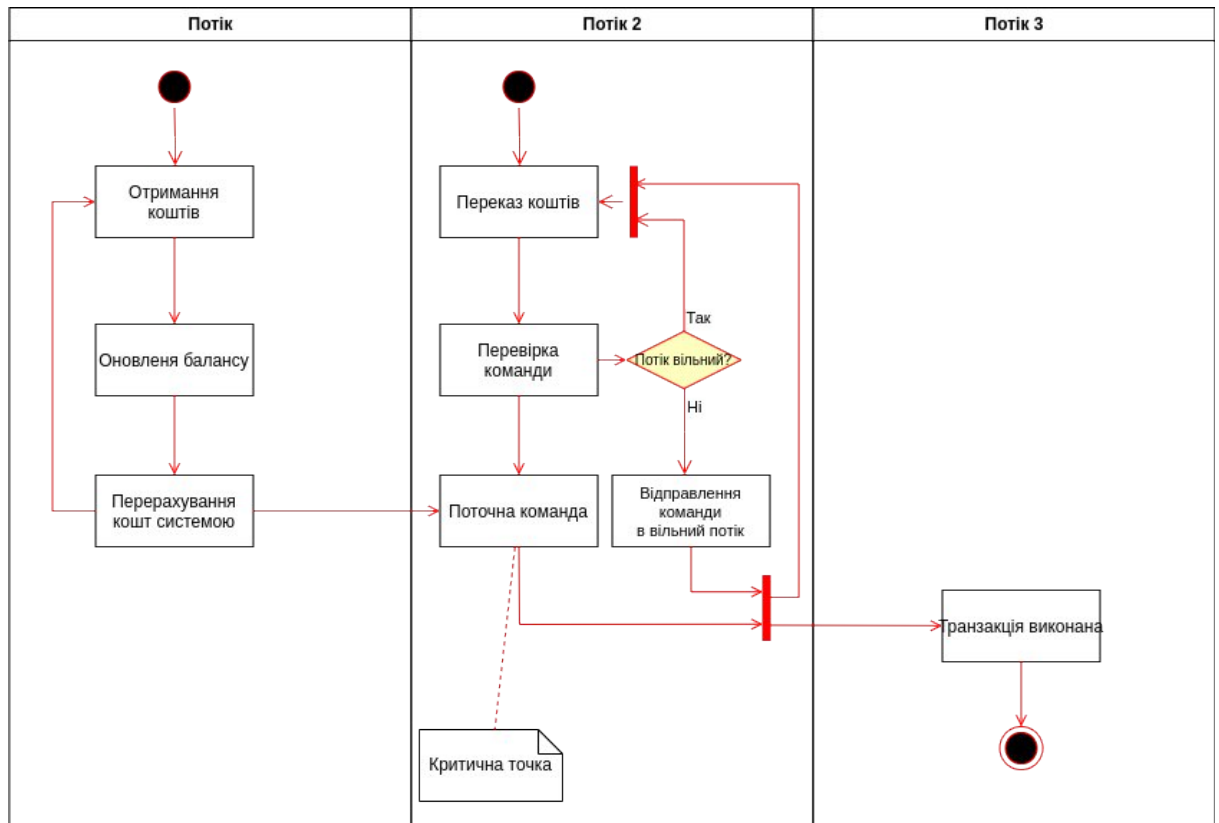
8. Діаграма станів — діаграма, що визначає зміну станів об'єкту у часі, одна з діаграм моделювання поведінки в UML. Подас об'єкт як автомат з теорії автоматів зі стандартизованими умовними позначеннями.



Діаграма станів 1

9. Специфікація процесу - це метод, який використовується для документування, аналізу та пояснення логіки та формул прийняття рішень, що використовуються для створення вихідних даних із вхідних даних процесу. Її метою є скорочення та конкретизація нормативних / інженерних вимог та процедур.

На прикладі нижче специфікація процесів потоків програми. Взаємодія процесів в багато поточній програмі для уникнення колізій та помилок.



Діаграма специфікація процесу 1

- 10.** Словник термінів являє собою короткий опис основних понять, що використовуються при складанні специфікацій. Він повинен включати визначення основних понять предметної області, опис структур елементів даних, їх типів і форматів, а також всіх скорочень і умовних позначень.
- 10.1. Функція – блок коду, який визначається один раз і може викликатись багаторазово
 - 10.2. Валідатор – програма, яка перевіряє правильність написання програмного коду і його відповідність стандартам.
 - 10.3. Сервер – це комп'ютер, який обслуговує всі підключені до нього персональні комп'ютери. При цьому сервер виконує свої функції і завдання без участі в цьому процесі людини, тобто самостійно.
 - 10.4. Бази даних - сукупність відомостей, об'єднаних за певною ознакою. Головне завдання бази даних — гарантоване збереження значних обсягів інформації (так звані записи даних) та надання доступу до неї користувачеві або ж прикладній програмі.
 - 10.5. Транзакція - банківська операція, яка полягає в переведенні коштів з одного рахунку на інший
 - 10.6. Потік — це джерело або призначення даних, зазвичай індивідуальних байтів або знаків. Потоки — це абстракція, що використовується наприклад при читанні або записі файлів, або при зв'язку з вузлами мережі. Три стандартні потоки передвизначені і доступні для всіх програм. В мові C++ концепція потоків реалізована у бібліотеці `iostream` і низці похідних від неї.

11. Висновки

У цій лабораторній роботі я здобув навички у постановки завдання.

Створенні документа «Ескізний проект», що містить:

- 11.1. Вибір методу рішення і мови програмування;
- 11.2. Специфікації процесів;
- 11.3. Всі отримані діаграми;
- 11.4. Словник термінів.

Навчився надати визначення аналізу та пояснити його призначення. Методи та засоби аналізу. Особливості застосування методів аналізу. Існуючі методи збору та виявлення вимог. Призначення діаграми прецедентів.