Посилання на GitHub:

<https://github.com/pmluxoftlab/ProjectPractice.git>

1. Функціональні вимоги
   1. Формування словників предметної області:

UML – універсальна мова для проектування програмних продуктів.

Діаграма послідовності – граф, який показує життєвий цикл об’єктів та їх взаємодію під час функціонування системи.

Діаграма кооперацій(комунікацій) – граф, який показує відношення між об’єктами.

Діаграма прецедентів – граф, який використовується для визначення бізнес-вимог клієнта.

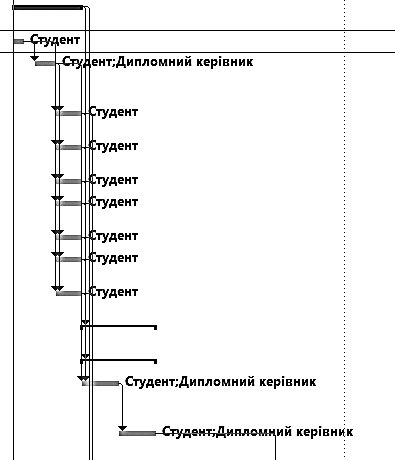
* 1. Формування багаторівневої класифікації функціональних вимог:

Побудування базового шляху проекту виконується на основі цілей проекту. Нижче представлено таблицю базового шляху проекту (таблиця 5), яка враховує робочі години за виключенням вихідних. На модифікацію архітектури ПЗ відведено 9 днів, аби виправити помилки вибору архітектурних стилів та вибрати правильні архітектурні тактики за допомогою методу оцінювання архітектури ATAM. Модифікація архітектури для варіантів використання виконується за 3 дні. На аналіз вимог видається 2 дні, вимоги чітко задані дипломним керівником та потребували незначного уточнення. На організаційні процеси виділено 10 днів. На навчання виділено 10 днів, так як в процес навчання входять розбір технологій, які допомагають пришвидшити процес сам процес розробки, вивчення досвіду інших при вирішенні проблем, пов’язаних с процесом кодування. На процес кодування виділено 25 днів. На документування відведено 3 дні, за цей час йде процес опису роботи системи та визначення технічних вимог до програми. На забезпечення якості виділено 6 днів, так як потрібно перевірити, чи задовольняє система вимогам кінцевого користувача.

Таблиця базового шляху проекту

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Назва задачі | Тривалість | Початок | Кінець | Попередник | Назви ресурсів | СДР |
| Модифікація архітетутри ПЗ | 9 днів | Чт 29.10.15 | Вт 10.11.15 |  | Студент | DW1 |
| Аналіз вимог | 2 днів | Чт 29.10.15 | Пт 30.10.15 |  | Студент | DW1.AA |
| Реконструкція програмної архітектури | 4 днів | Пн 02.11.15 | Чт 05.11.15 | 2 | Студент;Дипломний керівник | DW1.AB |
| ВВ: Create new Diagramm | 3 днів | Пт 06.11.15 | Вт 10.11.15 | 3;2 | Студент | DW1.AC |
| ВВ: Open Diagramm | 3 днів | Пт 06.11.15 | Вт 10.11.15 | 3;2 | Студент | DW1.AD |
| ВВ: Add Block | 3 днів | Пт 06.11.15 | Вт 10.11.15 | 3;2 | Студент | DW1.AE |
| ВВ: Add Connection | 3 днів | Пт 06.11.15 | Вт 10.11.15 | 3;2 | Студент | DW1.AG |
| ВВ: Save to File | 3 днів | Пт 06.11.15 | Вт 10.11.15 | 3;2 | Студент | DW1.AH |
| ВВ: Convert to XML | 3 днів | Пт 06.11.15 | Вт 10.11.15 | 3;2 | Студент | DW1.AI |
| ВВ: Init Draw Scene | 3 днів | Пт 06.11.15 | Вт 10.11.15 | 2;3 | Студент | DW1.AJ |
| Організаційні процеси | 10 днів | Ср 11.11.15 | Вт 24.11.15 |  | Студент | DW3 |
| Навчання | 10 днів | Ср 11.11.15 | Вт 24.11.15 | 3 | Студент | DW3.AC |
| Вивчення існуючої реалізації | 5 днів | Ср 11.11.15 | Вт 17.11.15 | 3;6;7;9;4;10;5;8 | Студент;Дипломний керівник | DW3.AC.1 |
| Вивчення архітектури MVC | 5 днів | Ср 18.11.15 | Вт 24.11.15 | 13 | Студент;Дипломний керівник | DW3.AC.2 |
| Розробка | 25 днів | Пт 13.11.15 | Чт 17.12.15 |  |  | DW5 |
| ВВ: Create new Diagramm | 4 днів | Пт 13.11.15 | Ср 18.11.15 | 4 | Студент | DW5.AA |
| ВВ: Open Diagramm | 4 днів | Пт 13.11.15 | Ср 18.11.15 | 5 | Студент | DW5.AK |
| ВВ: Add Block | 4 днів | Пт 13.11.15 | Ср 18.11.15 | 6 | Студент | DW5.AJ |
| ВВ: Add Connection | 4 днів | Пт 13.11.15 | Ср 18.11.15 | 7 | Студент | DW5.AI |
| ВВ: Save to File | 4 днів | Пт 13.11.15 | Ср 18.11.15 | 8 | Студент | DW5.AH |
| ВВ: Convert to XML | 4 днів | Пт 13.11.15 | Ср 18.11.15 | 9 | Студент | DW5.AG |
| ВВ: Init Draw Scene | 4 днів | Пт 13.11.15 | Ср 18.11.15 | 10 | Студент | DW5.AF |
| Допоміжні процеси | 9 днів | Ср 25.11.15 | Пн 07.12.15 | 14 |  | DW6 |
| Документування | 3 днів | Ср 25.11.15 | Пт 27.11.15 | 14;16;17;18;19;20;21;22 | Студент | DW6.AA |
| Забезпечення якості | 6 днів | Пн 30.11.15 | Пн 07.12.15 | 24;16;17;18;19;20;21;22 | Студент | DW6.AB |

Нижче на рисунках 1 та 2 представлені базовий шлях проекту та критичний шлях проекту(критичний шлях виділено жирним).



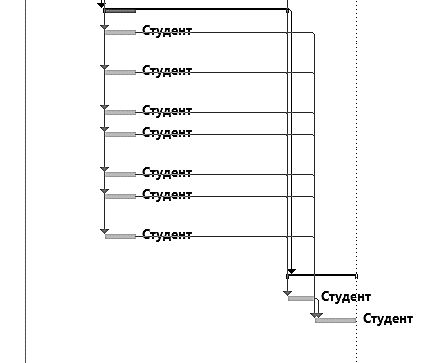


Рисунок 1.1 – Базовий шлях проекту

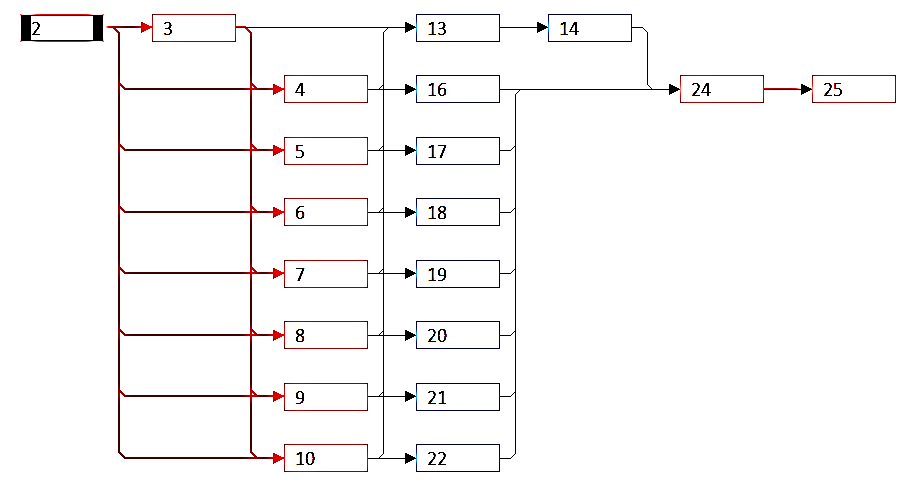


Рисунок 1.2 – Критичний шлях проекту

1.3 Функціональний аналіз існуючих продуктів:

|  |  |  |  |
| --- | --- | --- | --- |
| Параметри | this.UML | StarUML | Visio |
| Створення діаграмм UML | + | + | + |
| Збереження файлу у файл формату XML | + | - | - |
| Конвертація діаграм UML у граф специфікацій вимог | + | - | - |
| Платний продукт | - | +/-(умовно безкоштовний) | - |
| Зручність інтерфейсу користувача | + | + | + |
| Популярність | - | + | + |
| Вивід документу на друкування | + | + | + |
| Можливість запуску на різних платформах | + | +/-(немає в наявності реалізації для Linux) | - |

+ Так

* Ні
  1. MoSCoW принцип розташування пріорітетів:

M: Збереження файлу у файл формату XML

S: Конвертація діаграм UML у граф специфікацій вимог

C: Можливість запуску на різних платформах

W: Платний продукт

1. Нефункціональні вимоги:
   1. Інтерфейс користувача:

Дана програма дозволяє створювати наступні види діаграм:

Діаграми варіантів використання(рисунок 2.1):

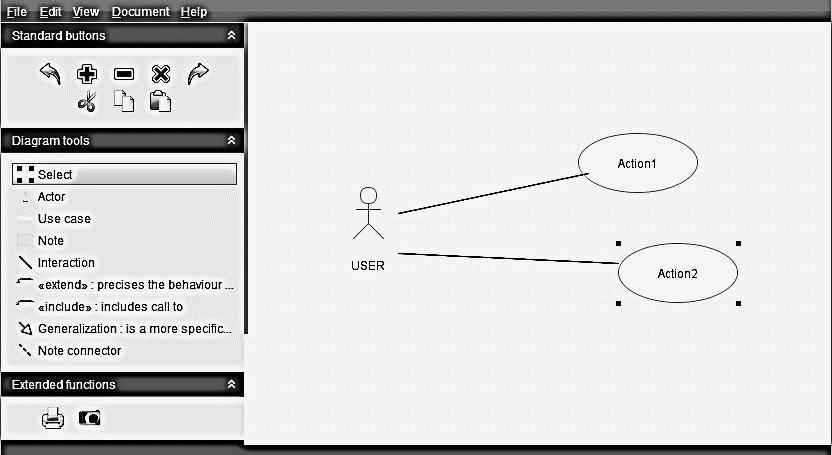


Рисунок 2.1 – Діаграма варіантів використання

Діаграми класів(рисунок 2.2):

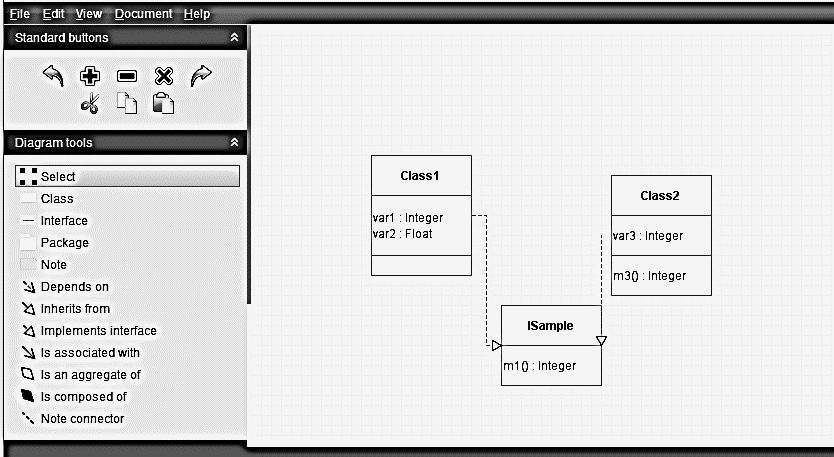


Рисунок 2.2 – Діаграма класів

Діаграми послідовностей(рисунок 2.3):

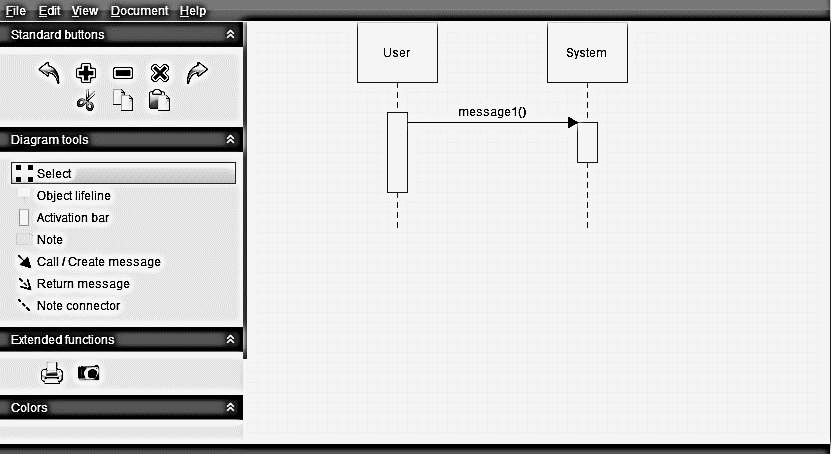


Рисунок 2.3 – Діаграма послідовностей

Діаграма кооперацій(рисунок 2.4):

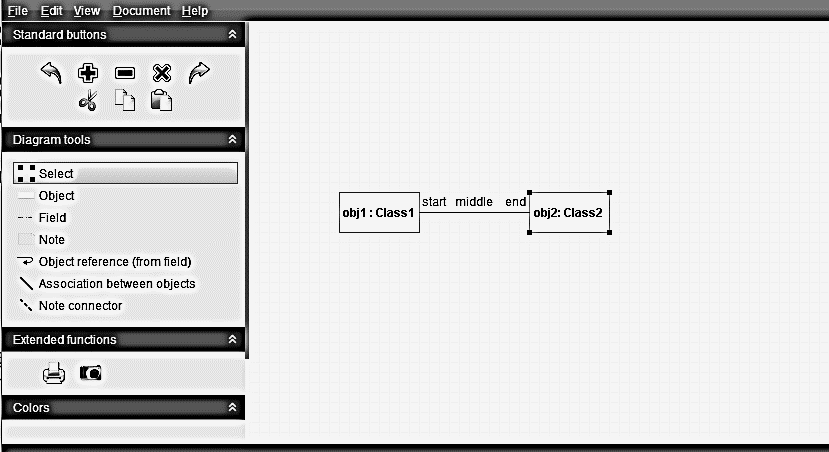


Рисунок 2.4 – Діаграма кооперацій

2.2 Атрибути якості:

Таблиця 2.1 – Атрибути якості

| Набір характеристик (за ISO 9126) | Властивості програмного забезпечення |
| --- | --- |
| Функціональність | * Здатність до взаємодії (interoperability): Програмний продукт повинен взаємодіяти з іншими модулями системи для обміну даними. * Захищеність (security): Дані, які надає система конкретному модулю, недоступні іншим модулям. |
| Ефективність | * Часові характеристики (time behaviour): Збереження даних повинно бути < 15 секунд; Додавання нового елементу на область редагування < 5 секунд * Використання ресурсів (resource utilisation): Програма не повинна використовувати більше 500 МБ. |
| Надійність | * Здатність до відновлення працездатності при відмовах (recoverability): Програма повинна мати можливість автоматично зберігати дані та у разі збоїв запускати останній збережений варіант діаграми. |
| Переносимість (мобільність) | * Адаптованість (adaptability): Платформою для написання програми буде обрано мову програмування Java, програма буде підтримувати усі актуальні платформи. |

* 1. Планування розробки:

2.1 Архітектура програмної системи:

На рисунку 5.1 зображена діаграма класів системи:

За основу був взятий архітектурний шаблон MVC. У процесі розробки архітектури була виявлена необхідність додати клас, який відповідає за збереження діаграми у файл формату XML.

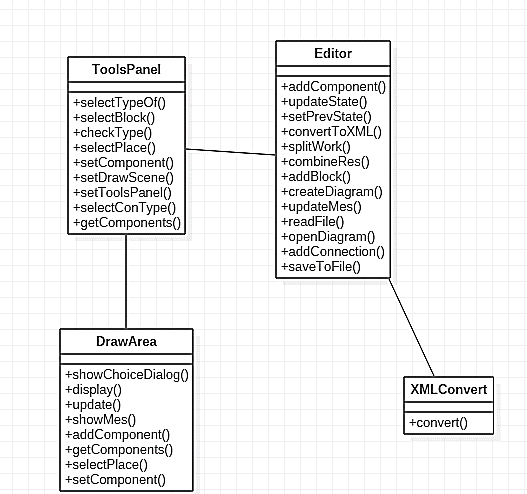


Рисунок 2.5 – Діаграма класів системи

Нижче описані зміни у порівнянні з базовим проектом:

* Додано метод splitWork() та combineRes() до класу Editor згідно застосуванню тактики Введення паралелізму.
* Додано компонент XMLConvert, який містить у собі метод convert(), згідно застосуванню тактики Введення паралелізму.
* До класу Editor додано метод displayErMes() згідно застосування тактики Транзакція.
* До класу Editor додано метод updateState() згідно застосування тактики Транзакція.
* До класу Editor додано метод setPrevSrate() згідно застосування тактики Транзакція.
  1. Технології розробки:
* Операційні системи: Windows, Linux, Mac OS( але обов’язковою умовою являється наявність встановленої Java версії не нижче Java 1.7).
* Структури даних: файли формату XML для діаграм, бінарні файли для збереження діаграм та для переведення діаграм у файл формату XML.
* Інструменти для розробки: IntelliJ IDEA Community Edition 15.0.2 для самого програмного продукту, Sublime Text 3 для створення DTD опису файлу XML, git та github для системи контролю версій.
  1. План розробки:

На рисунку 2.6 зображена діаграма пакетів програмних модулів системи:

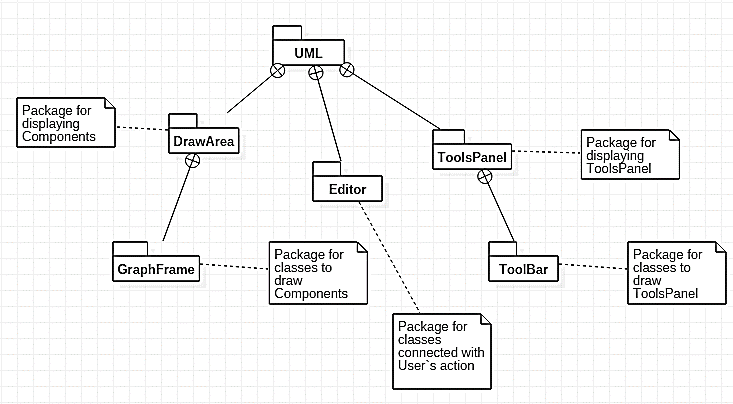


Рисунок 2.6 – Діаграма пакетів програмних модулів системи