Посилання на GitHub:

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

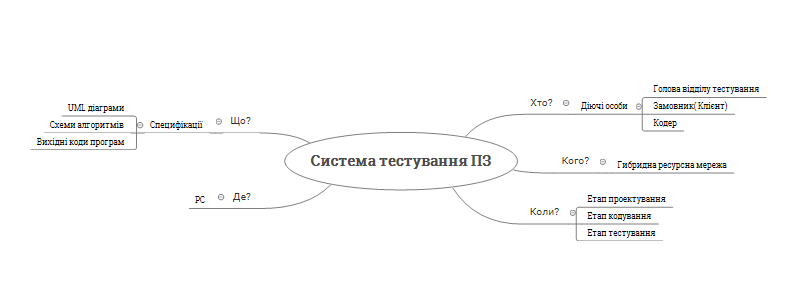
Визначення бізнес-вимог

* 1. Опис предметної області автоматизації:

Предметна область: комплексна система тестування програмного забезпечення.

Суб’єкти автоматизації: модулі комплексної системи тестування програмного забезпечення.

Представлення деталізованого об’єктів та суб’єктів у графічному вигляді:

Визначення проблем та суб’єктів автоматизації при взаємодії з об’єктом:

При проведенні процесу тестування програмного забезпечення діючі продукти орієнтовані на програмний код, а наша система комплексного тестування програмного забезпечення дозволяє проводити тестування вже на стадіях проектування програмного продукту. Діючі аналоги не мають потрібного функціоналу, такого як переведення специфікацій вимог у документ формату XML для створення графу специфікацій вимог.

Проблемний аналіз існуючих програмних 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 | - | + | + |

+ Так

* Ні

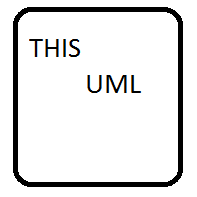
Визначення цілей , яких повинен досягати з урахуванням критеріїв, які показують економічний або інший ефект:

Метою являється створення системи - редактору діаграм UML(модулю системи комплексного тестування ПЗ), яка допоможе скоротити час на опис вимог. Система повинна створювати граф специфікацій вимог для подальшого використання іншими модулями.

Визначення унікальної назви програмного продукту, який включає описання об’єкту, досяжності цілей або вирішення проблем:

Назва програмного продукту – this.UML.

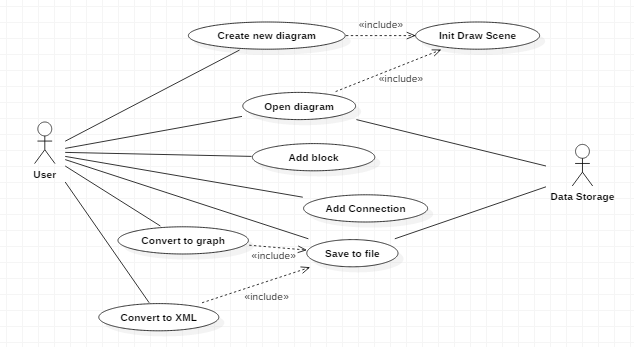
Нижче представлено логотип програмного продукту:



Логотип відображає у собі назву продукту на білому фоні. Квадратний щит з округленими кутами обрамлений чорним кольором.

* 1. Визначення вимог користувача:

Опис діаграми прецедентів:



Нижче представлено опис прецедентів:

**1.1.1 Прецедент: Create new diagram**

Передумова: Користувач запустив редактор

Сценарій:

1. Користувач вибирає опцію зі створення нової діаграми

2. Система відображає діалогове вікно з вибором назви діаграми та її розмірами.

3. Користувач вносить всі дані.

4. Система відображає діалогове вікно з вибором місця збереження файлу.

5. Користувач вибирає потрібне місце.

6. Система підтверджує успішне створення діаграми.

7. Запуск ВВ Init Draw Scene.

**1.1.2 Прецедент: Open Diagram**

Передумова: Користувач запустив редактор

1. Користувач вибирає опцію з відкриття існуючої діаграми.

2. Система відображає діалогове вікно з вибору місця розташування файлу.

3. Користувач вибирає потрібний файл.

4. Система підтверджує успішне завантаження файлу.

5. Запуск ВВ Init Draw Scene.

**1.1.3 Прецедент: Init Draw Scene**

Передумова: Система завантажила файл з діаграмою

1. Система прочитує зміст файлу (тут надо расширение, что файл поврежден)

2. Система відображає графічну сцену редагування та панель інструментів.

3. Система розміщує компоненти, які були прочитані з файлу, на графічну сцену.

**1.1.4 Прецедент: Add Block**

Передумова: Файл з діаграмою відкрито

1. Користувач на панелі інструментів вибирає потрібний вид блоку.

2. Користувач на графічній сцені вибирає місце розташування компоненту.

3. Система відображає блок на графічній сцені.

4. Система відображає активне поле редагування всередині блоку.

5. Користувач заповнює вміст блоку.

6. Система фіксує зміст блоку на графічній сцені.

**1.1.5 Прецедент: Add Connection**

Передумова: Файл з діаграмою відкрито

1. Користувач на панелі інструментів вибирає компонент зв'язку.

2. Користувач на графічній сцені вибирає місце розташування початку стрілки.

3. Система відображає на графічній сцені стрілку, кінець якої закріплений у вибраному користувачем місці.

3а. На місці сцени розташований блок.

3а.1 Система закріплює початок стрілки до блоку та відображає це на графічній сцені.

4. Користувач вибирає місце розтащування кінця стрілки.

5. Система закріплює цінець стрілки на графічній сцені.

5а. На місці сцени розташований блок.

5а.1 Система закріплює цінець стрілки до блоку та відображає це на графічній сцені.

**1.1.6 Прецедент: Save to file**

Передумова: Файл з діаграмою відкрито

1. Користувач вибирає опцію зі збереження файлу.

2. Система відображає діалогове вікно з вибору назви та місця розташування файлу.

3. Користувач заповнює необхідні дані.

4. Система відображає повідомлення про успішне збереження файлу.

**1.1.7 Прецедент: Convert to XML**

Передумова: Файл з діаграмою відкрито

1. Користувач вибирає опцію збереження діаграми у форматі XML.

2. Система виконує переробку діаграми та відображає результат в окремому вікні.

3. Користувач вибирає опцію збереження файлу.

4. Запуск ВВ Save file.

Специфікація вимог користувача( нефункціональних вимог):

| 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. |