**Міністерство Освіти І НАУКИ України**

**Національний університет "Львівська політехніка"**

Інститут **ІКНІ**

Кафедра **ПЗ**

**ЗВІТ**

До лабораторної роботи № 1

**На тему:** *“Формування технічного завдання як результат аналізу та визначення вимог.”*

**З дисципліни:** *“* *Вступ до інженерії програмного забезпечення”*

**Лектор:**

доцент кафедри ПЗ

Левус Є. В.

**Виконав:**

студент групи ПЗ-18

Юшкевич А.І.

**Прийняв:**

асис. кафедри ПЗ

Вишневський О.К.

Львів – 2023

**Тема роботи:** формування технічного завдання як результат аналізу та визначення вимог.

**Мета роботи:** навчитися складати найпростіше технічне завдання до розробки програми.

Теоретичні відомості

*14. Наведіть приклади нефункціональних вимог*

Нефункціональні вимоги - це вимоги, які визначають, наскільки добре повинен функціонувати продукт. Вони включають такі аспекти, як продуктивність, надійність, безпека, зручність використання та ремонтопридатність. Нефункціональні вимоги так само важливі, як і функціональні, оскільки вони допомагають гарантувати, що продукт відповідає потребам користувачів і зацікавлених сторін.

*29. У чому полягає складність отримання результатів аналізу й визначення вимог до ПЗ?*

Основною перепоною під час етапу аналізу та визначення вимог до ПЗ є непорозуміння між програмістом, та замовником, який через свою необізнаність у сфері інженерії програмного забезпечення не завжди може чітко сформулювати вимоги до продукту.

*33.* *Які чинники визначають зручність користування інтерфейсом?*

Існує 10 евристик, що допомагають створити зручний користувацький інтерфейс. На їхній основі можна виділити наступні чинники:

1. Відображення статусу системи – розуміння користувачем чи виконує програма корисну діяльність у кожен момент часу.
2. Інтуїтивне розуміння інтерфейсу на основі знань з реального світу
3. Надання користувачу можливості швидко виправляти помилки.
4. Однозначне трактування будь-якого процесу/вибору/дії програми.
5. Уникання певної реалізації функціоналу програми, що може спричинити помилку з боку користувача.
6. Створення інтерфейсу таким чином, щоб користувач не був змушений запам’ятовувати і тримати в голові багато інформації одночасно.
7. Наявність інструментів, які б пришвидшували процес користування програмою, або підвищували б його ефективність.
8. Чистий інтерфейс, вільний від зайвих, або рідко використовуваних елементів.
9. Повідомлення про помилки допомагають користувачу, не знайомому з програмуванням, зрозуміти і виправити несправність
10. Наявність додаткових матеріалів (зокрема документації), які за потреби допомагають користувачу зорієнтуватися у середовищі.

**Завдання**

**Умова.** Скласти технічне завдання (концептуальний рівень) до програми згідно індивідуального варіанту (№1-30) за поданим вище планом.

У п.4.3. представити зображення UI для демонстрації взаємодії користувача з програмою для кожної функції, зазначеної в п.4.1. Для забезпечення Usability використати евристики (UX) 1-8 (див. Додаток).

У п.4.4. задати дві нефункціональні вимоги.

У п.6 вказати документи, які будуть створені під час лабораторних робіт: блок-схеми алгоритмів, текст програми (за вимогами читабельності та зрозумілості), звіт про тестування.

У п.7 вказати два інформаційні джерела за темою програмування, як з Інтернет-ресурсів, так й друковані.

Варіанти для складання ТЗ

**25**. З клавіатури ввести послідовність записів, які містять дані про книгу : <Автор>, <Назва Книги>, <Рік видання>, <Кількість Сторінок>, <Вартість>. Роздрукувати введені дані у вигляді таблиці, а також подати інформацію згідно варіанту. Передбачити зчитування вхідних даних з файлу та запис результатів у файл. Відсортувати за назвою в алфавітному порядку дані про книги, вартість яких більша середньої в бібліотеці. Вилучити зі списку дані про книги з назвою, що починається на букви П, К, Л.

Технічне завдання

1. **Загальні положення:**

Назва: Library Manager;

Умовне позначення: LibMng;

Замовник: Національний університет “Львівська політехніка”

Розробник: ФОП Юшкевич А.І.

1. **Призначення системи:**

Програма “Library Manager” розробляється для автоматизації операцій пов’язаних зі зберіганням та обробкою інформації про книги, що зберігаються у бібліотеці Львівської політехніки. Система має спростити керування даними про книги, підняти ефективність та швидкість цих процесів.

1. **Об’єкти даних:**

Програма працює з текстовим файлом, створеним попередньо користувачем. Файл містить послідовність стрічок наступної структури:

<Автор>\t<Назва книги>\t<Рік видання>\t<Кількість сторінок>\t<Ціна книги>

<Автор> - ПІБ автора або його псевдонім

<Назва книги> - Назва книги

<Рік видання> - Рік видання вказаний у форматі “рррр”

<Кількість сторінок> - Кількість сторінок

<Ціна> - Ціна у грн. ₴, ціле число без додаткових позначень

\t – довільна кількість символів табуляції.

Дані повинні бути вказані у строгій відповідності з поданим зразком

Файл може бути порожнім.

1. **Вимоги до програмного забезпечення:**
2. Програма складається з наступних модулів:

* Модуль “IO” – відповідає за роботу з базою даних, (зчитування/запис)
* Модуль “Front” – відповідає за UI
* Модуль “Data Processing” – відповідає за основний функціонал програми (див. наступний пункт).

1. Програма реалізує наступний функціонал:

* Перегляд книг, що занесені в базу даних (текстовий файл). Для цього користувач обирає файл на диску. Список постійно відображається в основному вікні
* Сортування книг за назвою у спадному порядку, ціна яких менша за середню ціну усіх книг, що занесені в базу даних (текстовий файл)
* Видалення книги з бази даних
* Додавання книги до бази даних.
* Видалення усіх книжок, назви яких починаються з літер “P”, “K”, “L”.

1. Мінімальні системні вимоги:

* ОС: Windows 7 або новіше
* Процесор: Intel CPU Core i3
* Оперативна пам’ять: 2 GB ОП
* Відеокарта: Nvidia GTX 460
* Місце на диску: 2 GB доступного місця

1. З прототипом інтерфейсу можете ознайомитися за посиланням:

<https://cutt.ly/n4TOIwC>

1. Не функціональні вимоги:
   * + Система повинна бути простою у використанні та зрозумілою.
     + Система повинна відгукуватися на дії користувача зі швидкістю не меншою за 100 ms.
2. **Стадії розробки**

* **Початок роботи:** 05.04.2023
* **Закінчення роботи:** 17.05.2023

1. Визначитися зі стеком технологій, який буде використовуватися для розробки ПЗ (3 дні).
2. Сформувати дві команди розробників – Frontend та Backend developers, що працюватимуть паралельно.
3. Розробити прототип інтерфейсу, узгодити його із замовником (3 дні)
4. Опис модулів в програмному коді (3 тижні)
   * Розробка модуля “IO” (1 тиждень) (Backend developers)
   * Розробка модуля “Front” (18 днів) (Frontend developers)
   * Розробка модуля “Data Processing” (2 тижні) (Backend developers)
5. Тестування (1 тиждень)

* Тестування усіх функціональних вимог. Повна відповідність ПЗ до запитів замовника.
* Тестування UI на зручність. Перевірка на обов’язкове дотримання усіх 10-ти евристик.

1. Створення документацій (1 тиждень)
   * Інструкція по використанню програми для користувача(4 дні). Оригінал написати українською, перекласти на англійську та польську.
   * Документація до програмного коду (7 днів). Написати англійською мовою.
2. **Посилання**
3. Левус Є. В. Вступ до інженерії програмного забезпечення / Є. В. Левус, Н. Б. Мельник. – Львів: Видавництво Львівської Політехніки, 2018. – 246 с. ;
4. Левус Є. В. Життєвий цикл програмного забезпечення / Є. В. Левус, Т. А. Марусенкова – Львів: Видавництво Львівської Політехніки, 2017. – 207 с.
5. <https://www.geeksforgeeks.org/software-engineering-classification-of-software-requirements/>

Висновки

Аналіз і визначення вимог - один з основних етапів ЖЦ ПЗ. Складність цього етапу полягає у поєднанні формальних й неформальних методів ІПЗ, врахуванні знань предметної області, необхідності спілкування із замовником та майбутніми користувачами. Існують різні типи вимог: функціональні й нефункціональні. Основним результатом цього етапу є задокументовані узгоджені із замовником вимоги у формі спеціального документа. Аналізують і визначають вимоги досвідчені фахівці: аналітики, керівники проектів. Участь замовника, майбутніх користувачів обов’язкова на етапі аналізу й специфікації вимог.