

Державний університет «Одеська Політехніка»

Інститут комп'ютерних систем

Кафедра інформаційних систем

КУРСОВА РОБОТА

з дисципліни «Технології створення ПП »

Тема «Розробка застосунку «МУТС»»

Студента (ки) 3 курсу AI-194 групи
спеціальності 122 – «Комп'ютерні науки»

спеціалізації «Комп'ютерні науки»

Тарасенка Владислава
Олександровича

(прізвище та ініціали)

Керівник Блажко О.А.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала _____

Кількість балів: _____

Оцінка: ECTS _____

Члени комісії

_____	_____
(підпис)	(прізвище та ініціали)
_____	_____
(підпис)	(прізвище та ініціали)
_____	_____
(підпис)	(прізвище та ініціали)

Анотація

В курсовій роботі розглядається процес створення програмного продукту «Manage Your Time Correctly (MYTC)». Робота виконувалась в команді з декількох учасників: Шевченко Даніїл, Тарасенко Владислав. Тому в пояснювальній записці у розділах «Проектування» та «Конструювання» детальніше описано лише одну частину з урахуванням планів проведених робіт з розділу «Планування» з описом особливостей конструювання:

- структур даних моделі MVP в системі керування базами даних PostgreSQL;
- програмних модулів в інструментальному середовищі WebStorm з використанням фреймворку React та мови програмування JavaScript.

Результати роботи розміщено на github-репозиторії за адресою:
<https://github.com/GromZz/mytc>

Перелік скорочень

ОС – операційна система

ІС – інформаційна система

БД – база даних

СКБД – система керування базами даних

ПЗ – програмне забезпечення

ПП– програмний продукт

UML – уніфікована мова моделювання

Зміст

1	Вимоги до програмного продукту.....	6-20
1.1	Визначення потреб споживача.....	6-7
1.1.1	Ієрархія потреб споживача.....	6
1.1.2	Деталізація матеріальної потреби.....	6-7
1.2	Бізнес-вимоги до програмного продукту.....	7-9
1.2.1	Опис проблеми споживача.....	7-8
1.2.1.1	Концептуальний опис проблеми споживача.....	8
1.2.1.2	Метричний опис проблеми споживача.....	8
1.2.2	Мета створення програмного продукту.....	8
1.2.2.1	Проблемний аналіз існуючих програмних продуктів.....	8
1.2.2.2	Мета створення програмного продукту.....	8
1.2.3	Назва програмного продукту.....	8-9
1.2.3.1	Гасло програмного продукту.....	8
1.2.3.2	Логотип програмного продукту.....	9
1.3	Вимоги користувача до програмного продукту.....	9-15
1.3.1	Історія користувача програмного продукту.....	9-10
1.3.2	Діаграма прецедентів програмного продукту.....	10
1.3.3	Сценаріїв використання прецедентів програмного продукту.....	11-15
1.4	Функціональні вимоги до програмного продукту.....	15-17
1.4.1	Багаторівнева класифікація функціональних вимог.....	15-16
1.4.2	Функціональний аналіз існуючих програмних продуктів.....	17
1.5	Нефункціональні вимоги до програмного продукту.....	17-19
1.5.1	Опис зовнішніх інтерфейсів.....	17
1.5.1.1	Опис інтерфейса користувача.....	17
1.5.1.1.1	Опис INPUT-інтерфейса користувача.....	17-18
1.5.1.1.2	Опис OUTPUT-інтерфейса користувача.....	18-19
1.5.1.2	Опис інтерфейсу із зовнішніми пристроями.....	19
1.5.1.3	Опис програмних інтерфейсів.....	19
1.5.1.4	Опис інтерфейсів передачі інформації.....	19
1.5.1.5	Опис атрибутів продуктивності.....	19
2	Планування процесу розробки програмного продукту.....	20-27
2.1	Планування ітерацій розробки програмного продукту.....	20
2.2	Концептуальний опис архітектури програмного продукту.....	20
2.3	План розробки програмного продукту.....	20-27
2.3.1	Оцінка трудомісткості розробки програмного продукту.....	22
2.3.1.1	Вагові коефіцієнти акторів.....	22
2.3.1.2	Вагові коефіцієнти прецедентів.....	22-23
2.3.2	Визначення дерева робіт з розробки програмного продукту.....	24-25
2.3.3	Графік робіт з розробки програмного продукту.....	25-26
2.3.3.1	Таблиця з графіком робіт.....	26
2.3.3.2	Діаграма Ганта.....	26

1. Вимоги до програмного продукту

1.1 Визначення потреб споживача

1. Ієрархія потреб споживача

Відомо, що в теорії маркетингу потреби людини можуть бути представлені у вигляді ієрархії потреб ідей американського психолога Абрахама Маслоу включають рівні:

- фізіологія (вода, їжа, житло, сон);
- безпека (особиста, здоров'я, стабільність),
- приналежність (спілкування, дружба, любов),
- визнання (повага оточуючих, самооцінка),
- самовираження (вдосконалення, персональний розвиток).

На рисунку 1.1 представлено рівень потреби споживача, який хотілося б задовольнити, використовуючи майбутній програмний продукт.



Рисунок 1.1.1 – Ієрархія потреби споживача

Був обраний рівень «Самовираження», тому що, використовуючи програмний продукт «МУТС», споживач задовольняє такі потреби, як вдосконалення та персональний розвиток.

1.1.2 Деталізація матеріальної потреби

Для деталізації матеріальної потреби можна скористатися ментальними картами (MindMap). При створенні ментальних карт матеріальна потреба розташовується в центрі карти. Асоціативні гілки можна швидко створити, припускаючи, що в загальному вигляді з об'єктом пов'язані три потоки даних /

інформації: вхідний, внутрішній, вихідний. Кожен потік - це асоціативна група, що включає можливі п'ять гілок, що відповідають на п'ять питань: Хто? Що? Де? Коли? Як?

Відповідно до рекомендацій по створенню ментальних карт кожна гілка-асоціація може бути розділена на додаткові асоціативні гілки, які деталізують відповіді на поставлені питання.

Потреба, яка була визначена при аналізі матеріальних проблем споживача, основні та додаткові асоціативні гілки зображені на Рис 1.2

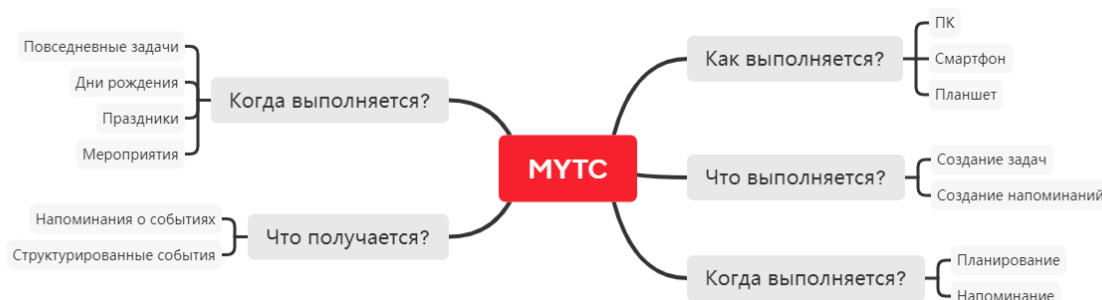


Рисунок 1.1.2 – Деталізація матеріальної потреби

1.2 Бізнес-вимоги до програмного продукту

1.2.1 Опис проблеми споживача

1.2.1.1 Концептуальний опис проблеми споживача

Концептуальний опис проблеми споживача відображений у табл. 1.1

Загальний опис проблеми	Метричні показники незадоволеності споживача
Неможливо тримати всі справи в голові або скрізь носити свій паперовий щоденник.	Можливість внесення як однієї, так і N-кількості подій

Таблиця – 1.2.1.1

1.2.1.2 Метричний опис проблеми споживача

Рівень доступності AL (AL – Access Level) можна визначити як

$$AL = NA / N,$$

Де NA – кількість справ, які користувач може тримати в голові; N – загальна кількість справ.

1.2.2 Мета створення програмного продукту

1.2.2.1 Проблемний аналіз існуючих програмних продуктів

Проблемний аналіз існуючих програмних продуктів відображений у табл.2.1

0	Назва	Вартість	Ступінь готовності	Примітка
1	Google Calendar	Безкоштовно	1	Неможливість вибору категорії подій
2	Samsung Calendar	Безкоштовно	1	Застарілий інтерфейс
3	Evernote	Безкоштовно	1	Неможливість ділитися подіями

Таблиця 1.2.2.1

1.2.2.2 Мета створення програмного продукту

Мета створення МҮТС - вирішення проблеми сучасної людини у питанні планування свого часу, важливих зустрічей, тощо. Поспішаючи, людина може забути одну з безлічі щоденних подій, додаток дозволить їм правильно розпорядитися своїм часом.

1.2.3 Назва програмного продукту

Назвою програмного продукту було вирішено взяти «МҮТС (Manage your time correctly)», що в перекладі з англійської означає «правильно розпоряджайтеся своїм часом». Назва «МҮТС» відображає мету та гасло програмного продукту.

1.2.3.1 Гасло програмного продукту

Гасло — лаконічна фраза, що впадає в око, добре запам'ятовується та

висловлює суть продукту. На початковому етапі розробки було вигадано гасло, яке найкращим чином відображає мету та суть роботи веб-сервісу та пов'язано з логотипом та назвою.

Гасло - «Час, витрачений на гарне планування, має властивість компенсуватися!»

1.2.3.2 Логотип програмного продукту

Відмінним способом представлення назви програмного продукту є його логотип, що поєднує зорові образи. На рис 1.2.3.2 можна побачити розроблений командою логотип, який відображає назву веб-сервісу.

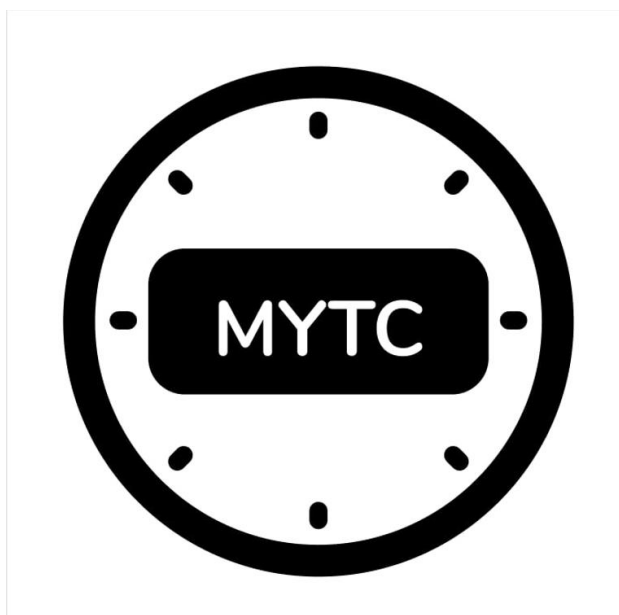


Рисунок - 1.2.3.2 – Логотип

1.3 Вимоги користувача до програмного продукту

1.3.1 Історія користувача програмного продукту

Гість може:

- переглядати загальнодоступні події;
- отримувати сповіщення про них.

Зареєстрований користувач може:

- додавати власні події;
- редагувати події;
- видаляти події з календаря;
- сортувати події по їх типу;

- отримувати оповіщення тільки про певні типи подій;
- встановлювати час і дату сповіщень;
- створювати сім'ю;
- ділитися нагадуваннями з учасниками сім'ї;
- розподіляти привілеї в сім'ї.

1.3.2 Діаграма прецедентів програмного продукту

Діаграма прецедентів (Use Case UML-діаграма) включає:

- актори (зацікавлені особи і зовнішні системи зі своїм API);
- прецеденти як основні функції ПП;
- зв'язки між прецедентами і акторами як множиною зацікавлених осіб;
- можливі зв'язки-узагальнення між акторами.

Діаграма прецедентів програмного продукту зображена на рис. 1.3.1



Рисунок - 1.3.2. Діаграма прецедентів програмного продукту

1.3.3 Сценарії використання прецедентів програмного продукту

Для кожного прецедента описаний сценарій використання з урахуванням пунктів:

- назва прецеденту;
- передумови початку виконання прецеденту;
- актори як зацікавлені особи у виконанні прецеденту;
- актор-основна зацікавлена особа як ініціатор початку прецеденту;
- гарантії успіху (що отримують актори у разі успішного завершення прецеденту);
- основний успішний сценарій;
- альтернативні сценарії, прив'язані до кроків основного успішного сценарію.

Реєстрація

Актори: користувач (неавторизований)

Передумови: відкрита сторінка реєстрації.

Постумови: користувача зареєстровано.

Основний сценарій:

1. Користувач відкриває сторінку реєстрації.
2. Система відображає форму реєстрації.
3. Користувач вводить ім'я та пароль для реєстрації.
4. Користувача зареєстровано.

Альтернативний сценарій:

1. Введено некоректні дані.
2. Користувач заповнює форму ще раз.

Перегляд подій

Актори: користувач (авторизований, неавторизований)

Передумови: відкрита сторінка подій.

Постумови: події відсортовані за категоріями.

Основний сценарій:

1. Користувач відкриває сторінку подій.

2. Система відображає список усіх загальнодоступних подій.
3. Користувач обирає тип події.
4. Виведені обрані події.

Альтернативний сценарій:

1. Не виведено жодної події
2. Користувач обирає інший тип подій

Авторизація

Актори: користувач (неавторизований)

Передумови: відкрита сторінка авторизації.

Постумови: користувача авторизовано.

Основний сценарій:

1. Користувач відкриває сторінку авторизації.
2. Система відображає форму авторизації.
3. Користувач вводить ім'я та пароль для авторизації.
4. Користувача авторизовано.

Альтернативний сценарій:

1. Невірний логін або пароль.
2. Користувач ще раз вводить дані.

Додавання подій

Актори: користувач (авторизований)

Передумови: відкрита сторінка «Додати подію».

Постумови: подія/нагадування додані.

Основний сценарій:

1. Користувач відкриває сторінку «Додати подію».
2. Система відображає форму додавання події.
3. Користувач вводить дату/час, назву і по бажанню опис події у форму.
4. Подію додано.

Альтернативний сценарій:

1. Користувач некоректно заповнює поля.
2. Подію не додано.

Редагування подій

Актори: користувач (авторизований)

Передумови: відкрита сторінка «Редагувати подію».

Постумови: подія/нагадування відредаговані.

Основний сценарій:

1. Користувач відкриває сторінку «Редагувати подію».
2. Система відображає форму редагування події.
3. Користувач змінює дату/час, назву і по бажанню опис події у форму.
4. Подію відредаговано.

Альтернативний сценарій:

1. Користувач некоректно змінює поля.
2. Подію не змінено.

Видалення подій

Актори: користувач (авторизований)

Передумови: відкрита сторінка «Редагувати подію».

Постумови: подія/нагадування видалені.

Основний сценарій:

1. Користувач відкриває сторінку «Редагувати подію».
2. Система відображає меню видалення події.
3. Користувач обирає «Видалити подію».
4. Подію видалено.

Встановлення часу і дати нагадувань

Актори: користувач (авторизований)

Передумови: відкрита сторінка «Редагувати подію».

Постумови: встановлені час та дата нагадування про подію.

Основний сценарій:

1. Користувач відкриває сторінку «Редагувати подію».
2. Система відображає форму редагування події.
3. Користувач встановлює дату/час нагадування.
4. Час і дата нагадування встановлені.

Альтернативний сценарій:

1. Користувач некоректно встановлює дату/час нагадування.
2. Час і дата нагадування не встановлені.

Створення родини

Актори: користувач (авторизований)

Передумови: відкрита сторінка «Створення родини».

Постумови: родина створена.

Основний сценарій:

1. Користувач відкриває сторінку «Створення родини».
2. Система відображає форму створення та налаштування родини.
3. Користувач обирає зареєстрованого користувача та додає його у родину.
4. Родину створено.

Альтернативний сценарій:

1. Користувач обирає незареєстрованого користувача та додає його у родину.
2. Родину не створено.

Поділитися подіями

Актори: користувач (авторизований)

Передумови: відкрита сторінка «Поділитися подією».

Постумови: успішний поділ подією.

Основний сценарій:

1. Користувач відкриває сторінку «Поділитися подією».
2. Система відображає форму поділу подією.
3. Користувач обирає зареєстрованого користувача та відправляє йому свою подію.
4. Подія відправлена.

Розподіл привілей у родині

Актори: користувач (авторизований)

Передумови: відкрита сторінка «Налаштування родини».

Постумови: привілеї розподілені.

Основний сценарій:

1. Користувач відкриває сторінку «Налаштування родини».
2. Система відображає форму налаштування родини.
3. Користувач обирає зареєстрованого користувача та події, до яких він матиме доступ.
4. Родину створено.

1.4 Функціональні вимоги до програмного продукту

1.4.1. Багаторівнева класифікація функціональних вимог

Ідентифікатор функції	Назва функції
FR 1	Авторизація користувача
FR 1.1	Створення запиту у користувача на отримання його параметрів ідентифікації та аутентифікації
FR 1.2	Передача від користувача його параметрів ідентифікації та аутентифікації
FR 1.3	Пошук інформації у базі даних користувачів
FR 1.4	Створення сесії для користувача
FR 2	Створення події
FR 2.1	Додавання події до бази даних
FR 3	Видалення події
FR 3.1	Видалення події з бази даних
FR 4	Редагування події
FR 4.1	Редагування події у базі даних
FR 5	Встановлення часу і дати нагадувань
FR 5.1	Створення запиту у користувача на отримання дати та часу нагадування

FR 5.2	Передача від користувача параметрів дати та часу нагадування
FR 5.3	Додавання нагадування до існуючої події у базі даних
FR 6	Перегляд подій по категоріям
FR 6.1	Користувач обирає категорії подій, які він хоче подивитись
FR 6.2	З бази даних відображаються обрані події
FR 7	Створення сім'ї
FR 7.1	Додавання сім'ї до бази даних
FR 8	Поділ подіями
FR 8.1	Створення запиту для того щоб поділитися подією
FR 8.2	Копіювання інформації про подію з бази даних
FR 8.3	Відправка інформації про подію
FR 9	Розподіл привілеїв
FR 9.1	Створення запиту у користувача на отримання форми налаштування сім'ї
FR 9.2	Передача від користувача його параметрів налаштування привілеїв у сім'ї

Таблиця – 1.4.1. Багаторівнева класифікація функціональних вимог

1.4.2 Функціональний аналіз існуючих програмних продуктів

Ідентифікатор функції	Google Calendar	MYTC
FR 1	+	+
FR 2	+	+
FR 3	+	+
FR 4	+	+
FR 5	+	+
FR 6	-	+
FR 7	-	-
FR 8	+	+
FR 9	-	-

Таблиця – 4.2. Функціональний аналіз існуючих програмних продуктів

1.5 Нефункціональні вимоги до програмного продукту

1.5.1 Опис зовнішніх інтерфейсів

1.5.1.1 Опис інтерфейса користувача

1.5.1.1.1 Опис INPUT-інтерфейса користувача

Ідентифікатор функції	Засіб INPUT-поток	Особливості використання
FR 1.1	Стандартна комп'ютерна клавіатура	
FR 4.1	2/3-кнопочний маніпулятор типу "миша", стандартна комп'ютерна клавіатура	Використання лівої кнопки миші для завершення процесу вводу даних

FR 5.1	2/3-кнопочний маніпулятор типу "миша" , стандартна комп'ютерна клавіатура	Використання лівої кнопки миші для завершення процесу вводу даних
FR 6.1	2/3-кнопочний маніпулятор типу "миша"	Використання лівої кнопки миші для вибору виведення даних

Таблиця – 1.5.1.1.1. Опис INPUT-інтерфейса користувача

1.5.1.1.2 Опис OUTPUT-інтерфейса користувача

Для функції «FR 1.1» OUTPUT-інтерфейса користувача.

The image shows a login interface titled "Войти" (Login). It contains two input fields: "Ваш Email" (Your Email) and "Ваш пароль" (Your password). Below these fields is a large button labeled "Войти". At the bottom, there are two links: "Забыли пароль?" (Forgot password?) and "Зарегистрироваться" (Register).

Рисунок 1.5.1.1.2.1 – Дизайн функції FR1

Для функції «FR 5.1» OUTPUT-інтерфейса користувача.

The image shows a task creation interface. It includes a text input field for "Название задачи:" (Task name:), a date picker for "Дата:" (Date:), and a time picker for "Время:" (Time:). Below these is a section "Выберите категорию:" (Select a category:) with four radio button options: "Мероприятие" (Event), "Напоминание" (Reminder), "Дни рождения" (Birthdays), and "Праздники" (Holidays). At the bottom is a large button labeled "Добавить" (Add).

Рисунок 1.5.1.1.2.2 – Дизайн функції FR5.1

1.5.1.2 Опис інтерфейсу із зовнішніми пристроями

FR1.1 – Desktop, Notebook, смартфон

FR4.1 – Desktop, Notebook, смартфон

FR5.1 – Desktop, Notebook, смартфон

FR6.1 – Desktop, Notebook, смартфон

1.5.1.3 Опис програмних інтерфейсів

Даний ПП працює на будь-яких операційних системах, певна версія програмного забезпечення не потрібна, бо ПП адаптований під усі. Взаємодія ПП із зовнішніми ПП не передбачається.

1.5.1.4 Опис інтерфейсів передачі інформації

Інтерфейси передачі інформації для реалізації більшості функцій ПП:

Провідні інтерфейси:

- Ethernet

Безпроводні інтерфейси:

- Wi-Fi

1.5.1.5 Опис атрибутів продуктивності

Ідентифікатор функції	Максимальний час реакції ПП на дії користувача, секунди
FR 1.1	4
FR 4.1	3
FR 5.1	3
FR 6.1	3

Таблиця 1.5.1.5.1 – Часи максимальної реакції для функції

2 Планування процесу розробки програмного продукту

2.1 Планування ітерацій розробки програмного продукту

З метою забезпечення для вимог таких рекомендацій IEEE-стандарту, як необхідність, корисність при експлуатації, здійсненність функціональних вимог до ПП, були визначені функціональні пріоритети, які будуть використані при плануванні ітерацій розробки ПП.

2.2 Концептуальний опис архітектури програмного продукту

Концептуальний опис архітектури програмного продукту зображено на рис.

2.2. Архітектурний тип – Rich Web Application (RWA).

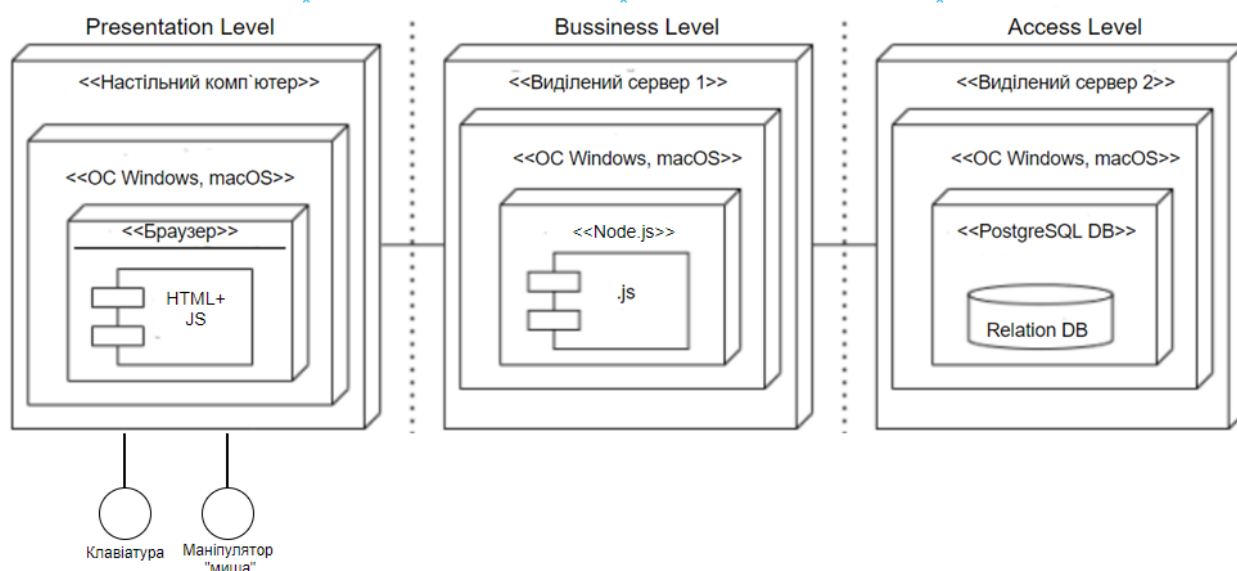


Рисунок – 2.2. Концептуальний опис архітектури програмного продукту

2.3 План розробки програмного продукту

З метою забезпечення для вимог таких рекомендацій IEEE-стандарту, як необхідність, корисність при експлуатації, здійсненність функціональних вимог до ПП, було визначено функціональні пріоритети, які будуть використані при плануванні ітерацій розробки ПП.

При створенні пріоритетів необхідно врахувати:

- сценарні залежності між прецедентами, до яких належать функції, на основі аналізу пунктів передумов початку роботи прецедентів, вказаних в описі сценаріїв роботи прецедентів;

– вплив роботи прецеденту, до якого належить функція, на досягнення мети ПП, наприклад у відсотках, на основі аналізу пунктів гарантій успіху, вказаних в описі сценаріїв роботи прецедентів.

Сценарні залежності будуть перетворені у відповідні функціональні залежності. Вплив роботи прецеденту буде поширено на всі підлеглі функції ієрархії. При визначенні пріоритетів рекомендується використовувати наступні позначки:

- M (Must) – функція повинна бути реалізованою у перших ітераціях за будь-яких обставин;
- S (Should) – функція повинна бути реалізованою у перших ітераціях, якщо це взагалі можливо;
- C (Could) – функція може бути реалізованою, якщо це не вплине негативно на строки розробки;
- W (Want) – функція може бути реалізованою у наступних ітераціях.

Опис представлено в таблиці 2.3

Ідентифікатор функції	Функціональні залежності	Вплив на досягнення мети, %	Пріоритет функції
FR 1		0	M
FR 2		30	M
FR 3	FR 2	0	S
FR 4	FR 2	0	S
FR 5	FR 2	15	C
FR 6		30	M
FR 7		10	W
FR 8		10	W
FR 9	FR 7	5	W

Таблиця 2.3 – Опис функціональних пріоритетів

2.3.1 Оцінка трудомісткості розробки програмного продукту

Визначення нескорегованого показника UUCP (Unadjusted Use Case Points)

1. Визначення вагових показників акторів A

Всі актори діляться на три типи: прості, середні і складні.

Простий актор представляє зовнішню систему з чітко визначеним Програмним інтерфейсом.

Середній актор представляє або зовнішню систему, що взаємодіє з ПП за допомогою мережевих протоколів, або особистість, що користується текстовим інтерфейсом (наприклад, алфавітно-цифровим терміналом).

Складний актор представляє особистість, що користується графічним інтерфейсом. Загальна кількість акторів кожного типу помножується на відповідний ваговий коефіцієнт, потім обчислюється загальний ваговий показник (табл.2.3.1.1)

Тип актора	Ваговий коефіцієнт	Кількість
Простий	1	0
Середній	2	0
Складний	3	2

Таблица - 2.3.1.1.Вагові коефіцієнти акторів

2. Визначення вагових показників прецедентів UC

Всі прецеденти діляться на три типи; прості, середні і складні в залежності від кількості кроків успішних сценаріїв (основних і альтернативних). Загальна кількість прецедентів кожного типу помножується на відповідний ваговий коефіцієнт, потім обчислюється загальний ваговий показник.

Тип прецедента	Кількість кроків сценарію	Ваговий коефіцієнт	Кількість
Простий	≤ 3	5	0

Середній	4-7	10	10
Складний	>7	15	0

Таблиця - 2.3.1.2.Вагові коефіцієнти прецендентів

Визначення UUCP:

$$A = 6$$

$$UC = 100$$

$$UUCP = A + UC = 106$$

Технічна складність проекту (TCF - Technical Complexity Factor) обчислюється з урахуванням показників технічної складності.

Кожному показнику присвоюється значення STi в діапазоні від 0 до 5:

- 1) 0 означає відсутність значимості показника для даного проекту;
- 2) 5 - високу значимість.

Показники технічної складності проекту TCF представлено в таблиці

2.3.1.2

Показник	Опис	Вага	ST
T1	Розподілена система	2	2
T2	Висока продуктивність	1	3
T3	Робота кінцевих користувачів on-line	1	1
T4	Складна обробка даних	-1	2
T5	Повторне використання коду	1	2
T6	Простота установки	0,5	0
T7	Простота використання	0,5	1
T8	Переносимість	2	2
T9	Простота внесення змін	1	2
T10	Паралелізм	1	2
T11	Спеціальні вимоги до безпеки	1	1
T12	Доступ зовнішніх користувачів	1	2
T13	Спеціальні вимоги до навчання користувачів	1	1

Таблиця 2.3.1.3 - Показники технічної складності проекту

Значення TCF

$$TCF = 0,6 + (0,01 * (ST_i * Вага_i)) = 0,805$$

Рівень кваліфікації розробників (EF - Environmental Factor) обчислюється з урахуванням наступних показників (табл. 2.3.1.3).

Показник	Опис	Вага	SF
F1	Знайомство з технологією	1,5	1
F2	Досвід розробки додатків	0,5	3
F3	Досвід застосування ООП	1	3
F4	Наявність ведучого аналітика	0,5	1
F5	Мотивація	1	5
F6	Стабільність вимог	2	2
F7	Часткова зайнятість	-1	1
F8	Складні мови програмування	-1	1

Таблиця – 2.3.1.4. Показники рівня кваліфікації розробників

Обчислення значення EF:

$$EF = 1,4 + (- 0,03 * (SF_i * Вага_i)) = 0,94$$

Остаточне значення UCP (Use Case Points)

$$UCP = UUCP * TCF * EF = 80,21$$

Оцінка трудомісткості проекту:

Кількість значень менше 3 між F1 – F6 : 3

Кількість значень більше 3 між F7 - F8 : 0

Загальна кількість – 3

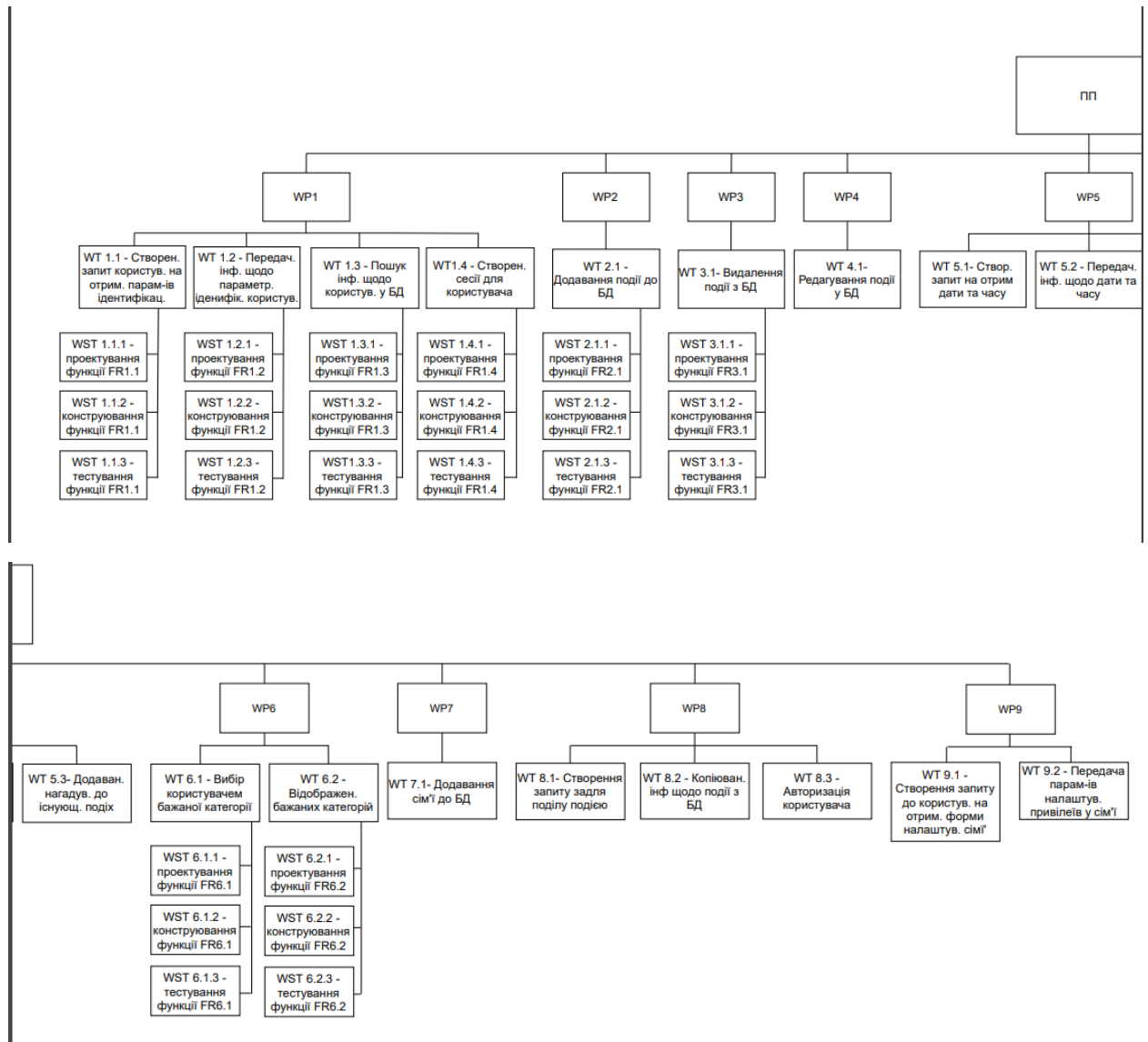
Тому слід використовувати 28 люд.-год на одну UCP:

$$28 * 80,21 = 2245 \text{ люд.-год.}$$

2.3.2 Визначення дерева робіт з розробки програмного продукту

При створенні дерева робіт (Work BreakDown Structure- WBS) використовується дерево функцій, яке було створено раніше.

Для кожної задачі визначаються підзадачі - Work SubTask (WST) з урахуванням базових процесів розробки програмних модулів: проектування, конструювання, модульне тестування, збірка та системне тестування.



Для кожної підзадачі визначається виконавець, що фіксується у вигляді таблиці, яка представлена в таблиці 2.3.3.1

WST	Дата початку	Дні	Дата завершення	Виконувач
WST1.1.1	1.10.2021	3	3.10.2021	Шевченко
WST1.1.2	3.10.2021	3	5.10.2021	Шевченко
WST1.1.3	5.10.2021	2	6.10.2021	Шевченко
WST1.2.1	1.10.2021	3	3.10.2021	Тарасенко
WST1.2.2	3.10.2021	3	5.10.2021	Тарасенко
WST1.2.3	5.10.2021	2	6.10.2021	Тарасенко
WST1.3.1	6.10.2021	3	8.10.2021	Тарасенко
WST1.3.2	8.10.2021	3	10.10.2021	Тарасенко
WST1.3.3	10.10.2021	2	11.10.2021	Тарасенко
WST1.4.1	6.10.2021	3	8.10.2021	Шевченко
WST1.4.2	8.10.2021	3	10.10.2021	Шевченко
WST1.4.3	10.10.2021	2	11.10.2021	Шевченко
WST2.1.1	11.10.2021	4	14.10.2021	Тарасенко
WST2.1.2	14.10.2021	3	16.10.2021	Тарасенко
WST2.1.3	16.10.2021	2	17.10.2021	Тарасенко
WST3.1.1	11.10.2021	4	14.10.2021	Шевченко
WST3.1.2	14.10.2021	3	16.10.2021	Шевченко

