

Міністерство освіти і науки України
Державний університет «Одеська політехніка»
Інститут комп'ютерних систем
Кафедра інформаційних систем

КУРСОВА РОБОТА

з дисципліни «Технології створення програмних продуктів»

за темою

«Розробка застосунку “Polar”»

Пояснювальна записка до етапів визначення вимог до програмного продукту та
планування процесів розробки програмного продукту

Виконали:
студенти 3-го курсу
групи УІ-191
Ігнатюк А.Г,
Кисельов Д.С.
Фаюк К.С.

Перевірив:
Блажко О. А.

Одеса - 2021

Анотація

В курсовій роботі розглядається процес створення програмного продукту «Polar» на етапах визначення вимог до програмного продукту та планування процесів розробки.

Робота виконувалась в команді з декількох учасників. Роботу виконували студенти групи УІ-191 Ігнатюк Андрій Геннадійович, Кисельов Даніїл Сергійович, Фаюк Костянтин Сергійович.

В робота пов'язана з такими матеріальними потребами споживача, як самовираження, а саме із необхідністю персонального розвитку. Аналіз вказаних потреб визначив інформаційну потребу - нестачу доступних та цікавих джерел вивчення історії власного університету, що призводить до загальної байдужості до цієї теми з боку викладачів і студентів.

При визначенні ступеня готовності існуючих програмних продуктів до вирішення інформаційної потреби проаналізовано наступні програмні продукти: Florence City Guide, Questo, Franklin Institute AR, Kennedy Space Center AR, National Museum of Singapore AR.

Поточну версію пояснювальної записки до результатів роботи розміщено на *GitHub*-репозиторії за адресою: <https://github.com/andrii-ihnatiuk/polar-proj>.

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

AR – доповнена реальність

Зміст

	стор.
1. Вимоги до програмного продукту	6
1.1 Визначення потреб споживача	7
1.1.1 Ієрархія потреб споживача	
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	
1.2.1.2 Опис цільової групи споживача	
1.2.1.3 Метричний опис проблеми споживача	
1.2.2 Мета створення програмного продукту	
1.2.2.1 Проблемний аналіз існуючих програмних продуктів	
1.2.2.2 Мета створення програмного продукту	
1.2.3 Назва програмного продукту	
1.2.3.1 Гасло програмного продукту	
1.2.3.2 Логотип програмного продукту	
1.3 Вимоги користувача до програмного продукту	
1.3.1 Історія користувача програмного продукту	
1.3.2 Діаграма прецедентів програмного продукту	
1.3.3 Сценаріїв використання прецедентів програмного продукту	
1.4 Функціональні вимоги до програмного продукту	
1.4.1. Багаторівнева класифікація функціональних вимог	

1.4.2 Функціональний аналіз існуючих програмних продуктів

1.5 Нефункціональні вимоги до програмного продукту

1.5.1 Опис зовнішніх інтерфейсів

1.5.1.1 Опис інтерфейсів користувача

1.5.1.1.1 Опис INPUT-інтерфейсів користувача

1.5.1.1.2 Опис OUTPUT-інтерфейсів користувача

1.5.1.2 Опис інтерфейсу із зовнішніми пристроями

1.5.1.3 Опис програмних інтерфейсів

1.5.1.4 Опис інтерфейсів передачі інформації

1.5.1.5 Опис атрибутів продуктивності

2. Планування процесу розробки програмного продукту

2.1 Планування ітерацій розробки програмного продукту

2.2 Концептуальний опис архітектури програмного продукту

2.3 План розробки програмного продукту

2.3.1 Оцінка трудомісткості розробки програмного продукту

2.3.2 Визначення дерева робіт з розробки програмного продукту

2.3.3 Графік робіт з розробки програмного продукту

2.3.3.1 Таблиця з графіком робіт

2.3.3.2 Діаграма Ганта

1. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

1.1 Визначення потреб споживача

1.1.1 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 1.1.1 представлено одну ієрархію потреби споживача, яку хотілося б задовольнити, використовуючи майбутній програмний продукт.



Рис. 1.1.1 – Ієрархія потреб споживача

1.1.2 Деталізація матеріальної потреби

При деталізації матеріальної потреби в даній курсовій роботі було використано так звані ментальні карти (Mind Map). Під час створення ментальних карт, матеріальна потреба була розташована в центрі карти. Для даної роботи матеріальна потреба була охарактеризована як “Процес пізнання”.

Асоціативні гілки були створені, 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 була розділена на додаткові асоціативні гілки, які деталізують відповіді на поставлені питання.

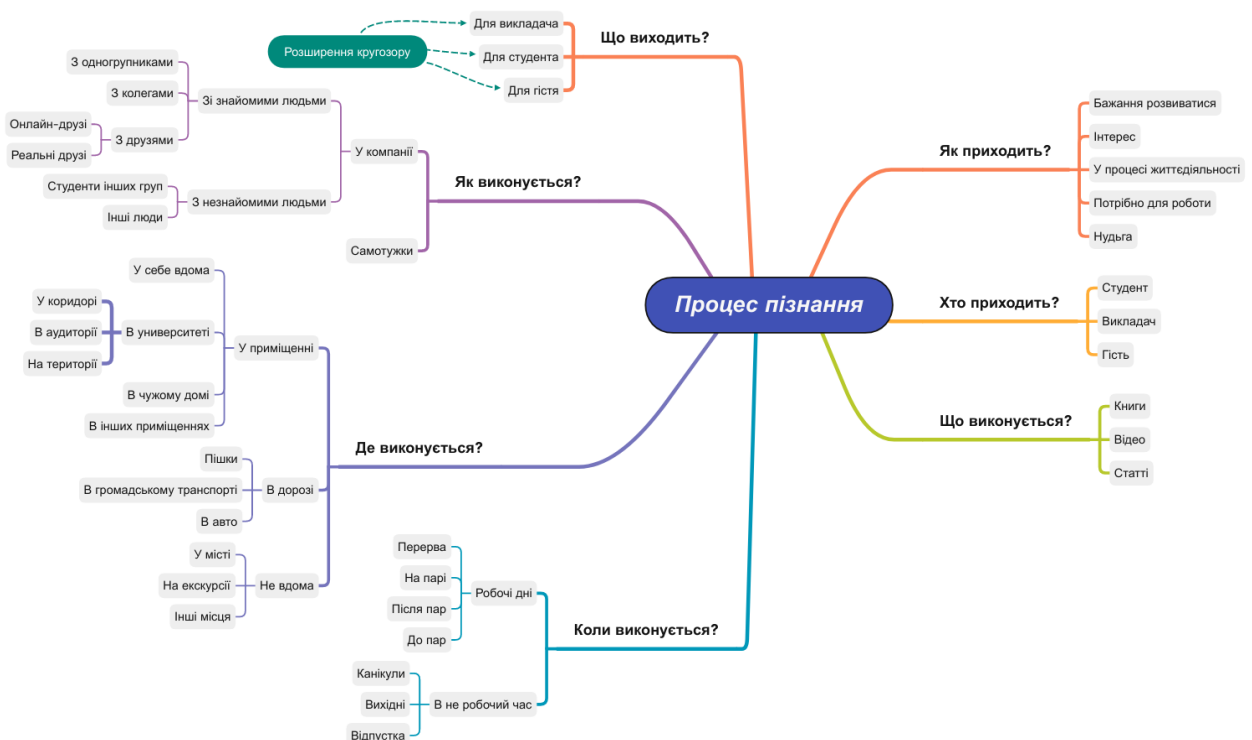


Рисунок 1.1.2 - Деталізація матеріальної потреби користувача

1.2 Бізнес-вимоги до програмного продукту

1.2.1 Опис проблеми споживача

1.2.1.1 Концептуальний опис проблеми споживача

Головна проблема, яка була вилучена під час розробки бізнес вимог до програмного продукту - нестача доступних та представлених мовою споживача джерел з інформацією, із яких можна дізнатися про історію Державного університету “Одеська політехніка”.

Інформація про історію університету, яка була задокументована, знаходиться здебільшого у традиційних паперових книгах, більша частина з яких навіть не оцифрована. Тому дана інформація не задовольняє таким вимогам користувача, як доступність та представленість мовою споживача.

Дана інформація є важкодоступною, оскільки для її отримання без застосунку користувач повинен іти до бібліотеки та шукати відповідну паперову книгу. Така дія забирає багато часу та не є дуже цікавою, через що більшість користувачів не стане витратити на це особистий час. Також ця інформація не представлена мовою користувача, адже записана у професійній літературі чи архівних записах, які важко зрозуміти людині, незнайомій зі специфікою даної сфери.

1.2.1.2 Опис цільової групи споживачів

Під час аналізу цільової групи споживачів, було виділено ряд характеристик, які в подальшому можуть впливати на його вимогу до інформації.

Нижче приведено перелік цих характеристик:

- 1) Вік - застосунок здебільшого орієнтований на молодих людей, у зв'язку. Орієнтовний вік потенційних користувачів від 17 до 25 років.

- 2) Стаття - в даному випадку, стаття не має значення, так як застосунок підійде як для чоловіків, так і для жінок.
- 3) Освіта - для більшості користувачів потрібна наявність вищої або середньої освіти, бо люди із нижчим рівнем освіти рідко бувають в університетах;
- 4) Професія - застосунок орієнтований на студентів та викладачів університетів. Університети можуть бути різними, адже їх працівникам та студентам також може бути цікава історія Державного університету “Одеська політехніка”;
- 5) Особисті якості - потенційні користувачі застосунку повинні мати такі особисті якості, як цікавість та бажання дізнатися щось нове. Людям без цих особистих якостей не буде цікаво вивчати історію університету.

На основі вищевказаного переліку можна визначити цільову групу програмного продукту.

Застосунок “Polar” - орієнтований на цікавих молодих людей (незалежно від статі) із середньою або неповною вищою освітою віком від 17 до 25 років, які є здобувачами освіти в університеті або молодими викладачами, а також зацікавлені в історії університету.

1.2.1.3 Метричний опис проблеми споживача

Для того, аби проблему споживача можна було описати у метричному вигляді, потрібно виконати її перетворення. Переведення проблеми із споживчої в метричну форму представлено у таблиці 1.2.1.3.1.

Загальний опис проблеми	Метричні показники незадоволеності споживача
Нестача доступних та зрозумілих джерел для вивчення історії університету	Низький рівень зацікавленості до вивчення історії університету.

Таблиця 1.2.1.3.1 - Переведення проблеми із споживчої в метричну форму

Також на даному етапі можна ввести новий параметр - рівень зацікавленості, який вказано у визначенні проблеми. Рівень зацікавленості можна позначити аббревіатурою IL (IL - Interest Level). Його можна визначити за наступною формулою:

$$IL = TI / T$$

Де:

TI (TI - Total Interested) - загальна кількість зацікавлених у вивченні історії університету людей.

T - загальна кількість людей, пов'язаних з університетом.

Тобто, ці параметри є пропорційними - чим більша кількість студентів або викладачів буде зацікавлена історією університету, тим більшим буде рівень зацікавленості.

Також у даному пункті можна представити приклад значення метрики проблеми споживача. Роздивимось ситуацію до виходу застосунку. Будемо вважати, що загальна кількість людей, пов'язаних з університетом - 15000 студентів та викладачів, а кількість зацікавлених людей (які самостійно вивчають історію) - 2000 людей.

Рівень зацікавленості у цьому випадку: $IL = 2000/15000 = 0.13$ (чи 13%).

Роздивимось ситуацію, коли вийде застосунок. Будемо вважати, що у зв'язку з легкою доступністю інформації та легкості її отримання кількість таких людей зросте до 5000.

Рівень зацікавленості у цьому випадку: $IL = 5000/15000 = 0.33$ (чи 33%).

Для того, аби перевірити рівень зацікавленості історією університету, було проведено анкетування серед студентів. Питання були пов'язані із тим, наскільки цікаво студентам було вивчати місце свого навчання, та вказанням деяких даних для уточнення цільової групи відповідаючих. Усього в анкетуванні прийняли участь 50 осіб.

Анкета була складена із шести питань:

- 1) “Оберіть ваш курс”.
- 2) “Яку частину корпусів університету ви відвідали?”.
- 3) “Чи хотіли би відвідати усі корпуси?”.
- 4) “Чи знайомі ви з історією політеху?”.
- 5) “Чи хочете ви дізнатися більше про місце, в якому навчатесь?”.
- 6) “Чи подобаються вам квести із пошуком схованих речей?”.

Результати анкетування представлені на рисунках 1.2.1.3.1 - 1.2.1.3.6

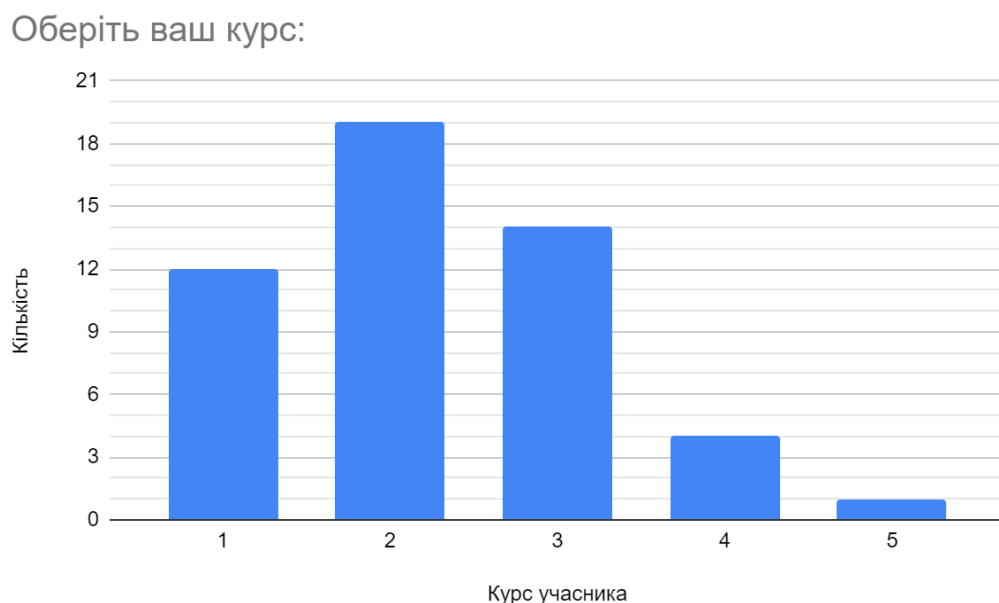


Рисунок 1.2.1.3.1 - Розподіл курсів за респондентами

1) У якій частині корпусів політеха ви бували?

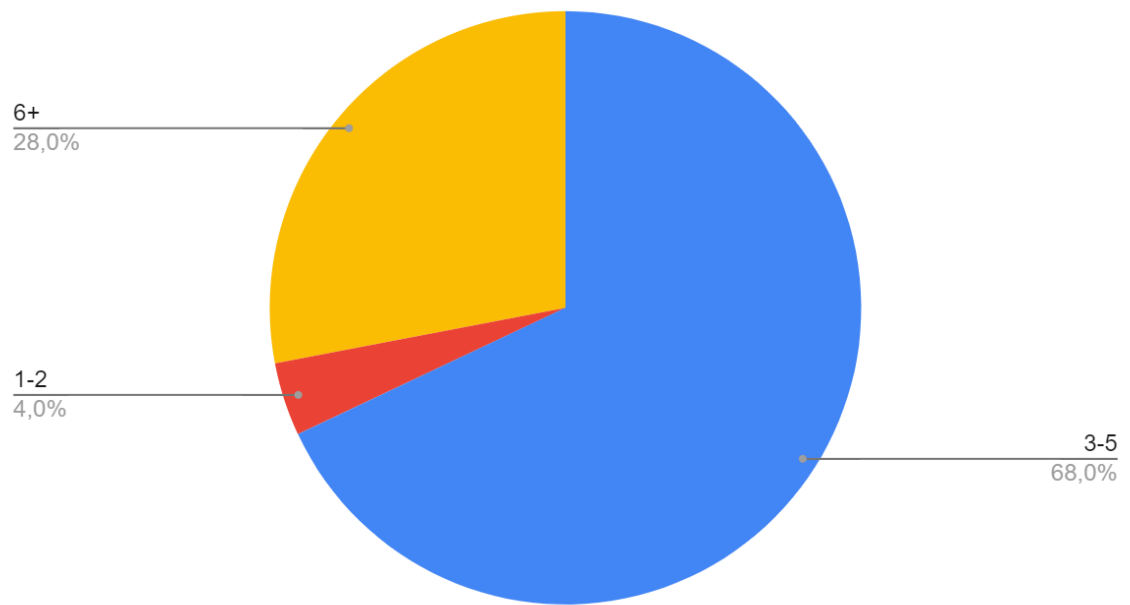


Рисунок 1.2.1.3.2 - Кількість відвіданих корпусів за кількістю респондентів

2) У вас є бажання побувати в усіх корпусах? – количество

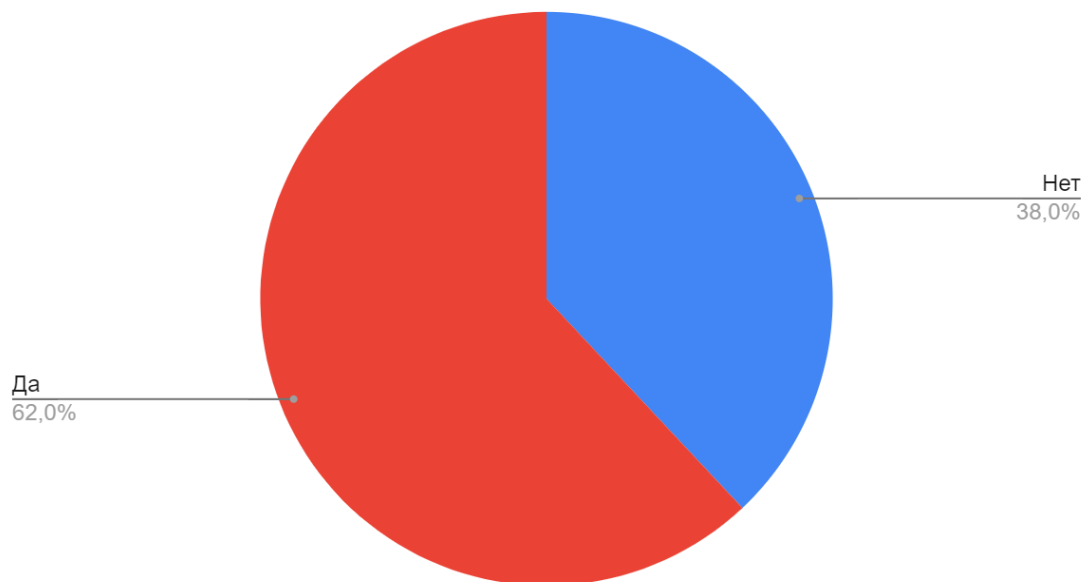


Рисунок 1.2.1.3.3 - Зацікавленість респондентів у відвідуванні усіх корпусів

3) Ви знайомі з історією політеху? – кількість

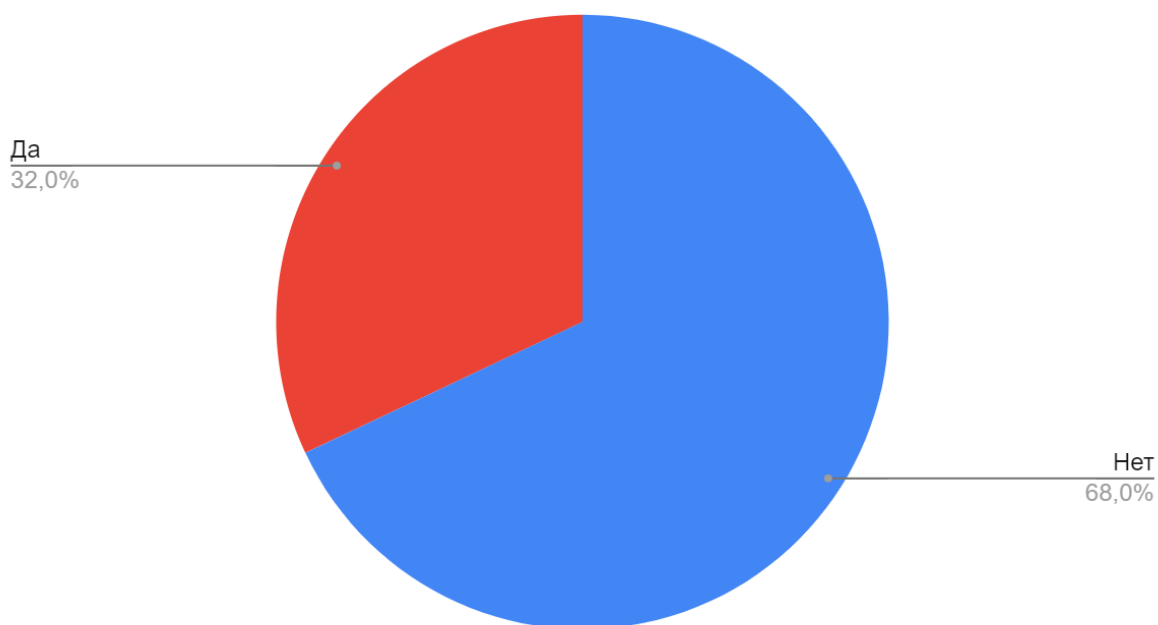


Рисунок 1.2.1.3.4 - Процент студентів, знайомих з історією політеху

4) Бажаєте дізнатися більше про місце навчання? – кількість

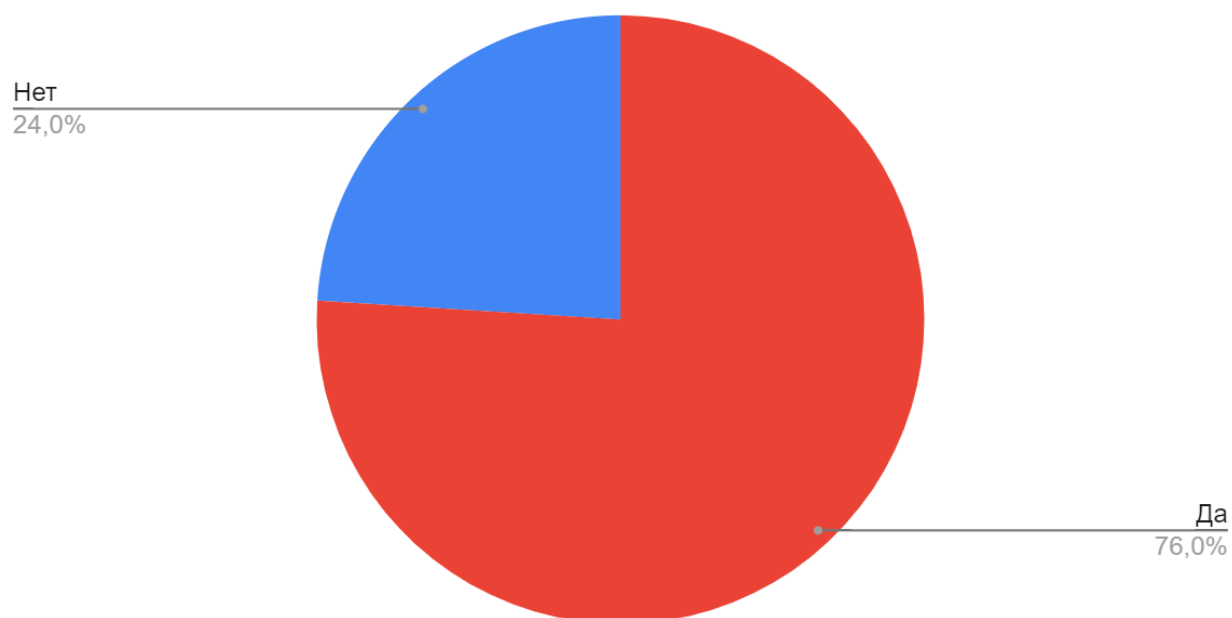


Рисунок 1.2.1.3.5 - Процент бажаючих дізнатися більше про університет

5) Вам подобаються квести з пошукам захованих речей? –
количество

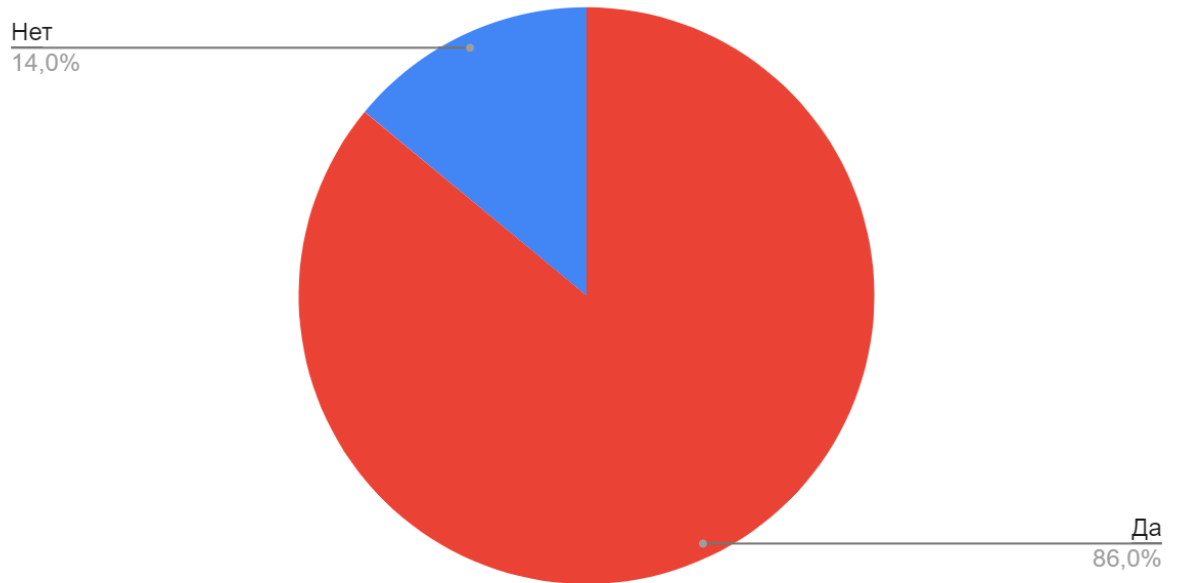


Рисунок 1.2.1.3.6 - Процент респондентів, яким подобаються квести із пошуком схованих речей

На основі проведеного анкетування можна зробити висновок, що вони дійсно підтверджують вказаний приклад значення метрики проблеми споживача - низького рівню зацікавленості до вивчення історії університету. Варто відмітити, що при анкетуванні було забезпечено репрезентативність результатів, оскільки в ньому приймали участь споживачі, які мають всі характеристики цільової групи (див. пункт 1.2.1.2).

1.2.2 Мета створення програмного продукту

1.2.2.1 Проблемний аналіз існуючих програмних продуктів

Проблемний аналіз існуючих програмних продуктів було створено за наступними кроками:

- 1) сформовано перелік продуктів через пошук в мережі Інтернет;

2) створено таблицю-рішення проблем (рядки - назви продуктів, стовпці - проблеми, осередки - позначки про вирішення проблем продуктом).

№	Назва продукту	Ціна	Ступінь готовності	Примітка
1	Florence City Guide	Безкоштовно	1	Відсутність фокусу на історичних місцях
2	Questo	Платно	1	Є можливість зареєструватися, але історію переглянути можна лише один раз
3	Franklin Institute AR	Безкоштовно	1	Немає можливості зберігати прогрес
4	Kennedy Space Center AR	Безкоштовно	1	Немає можливості зберігати прогрес
5	National Museum of Singapore AR	Безкоштовно	1	Немає можливості зберігати прогрес

Таблиця 1.2.2.1.1 - Результати проблемного аналізу існуючих програмних продуктів

1.2.2.2 Мета створення програмного продукту

Для того, аби описати мету створення програмного продукту, її потрібно описати у метричному вигляді. Сформована у метричному вигляді мета звучить наступним чином: “Підвищити рівень інтересу студентів та викладачів до вивчення історії власного університету”. В даному випадку, метрика мети є антонімом до метрики інформаційної потреби.

1.2.3 Назва програмного продукту

У якості назви програмного продукту було обрано словосполучення “Polar”. Назва “Polar” є акронімом, що утворено від двох слів - “Polytech” та “AR”.

1.2.3.1 Гасло програмного продукту

У якості гасла програмного продукту було обрано речення "Твій університет має більше секретів, ніж тобі здається".

1.2.3.2 Логотип програмного продукту



Рисунок 1.2.3.2 - Логотип застосунку “Polar”

1.3 Вимоги користувача до програмного продукту

1.3.1 Пригодницька історія користувача програмного продукту (за бажанням членів проектної команди)

Був чудовий сонячний день. На території Одеського політеху вже можна було побачити веселе лице Олега, що був студентом третього курсу інституту

комп'ютерних систем. У нього зав'язалася розмова з друзями, в якій він дізнався про новий хіт університету - застосунок "Polar", що являє собою мобільний додаток за допомогою якого можна вивчати історію університету в ігровій формі. Олег відразу вирішив відвідати сайт щоб завантажити застосунок на свій мобільний пристрій. Він так хотів скоріше відчути ту насолоду від гарного, сучасного дизайну, та зручного користувацького інтерфейсу, що вирішив авторизуватися за допомогою свого аккаунту Google. Натискання кнопки завантаження, декілька секунд очікування і ось - Олег вже готовий вивчати історію рідного університету. Перше що побачив Олег - це екран привітання, що ознайомлює користувача з основною концепцією застосунку. Після цього перед ним відкрилася мапа університету і він чимдуж подався шукати свою першу мітку. Коли Олегу пощастило нарешті знайти довгоочікувану мітку, він скоріше прийнявся її сканувати, натиснувши найбільшу кнопку знизу екрана. Олег так захопився процесом, що не помітив як затулив своїм тілом основне джерело світла у кімнаті, через що додаток не зміг відсканувати мітку з першого разу. Збентежений, Олег повторив спробу сканування, натиснувши на відповідну кнопку. І ось, історія того місця з'явилася у його пристрої, а Олег, ознайомившись з нею, отримав підвищення свого рівня. Якщо б не заняття, Олег так і ходив би і з захопленням вивчав приховані історії звичайних для нього місць, та всьому свій час. Ввечері, прийшовши додому Олег був вже гравцем третього рівня, та він так захопився вивченням історій що зовсім забув про те що у грі є рейтинг гравців. Оскільки, він вже був за комп'ютером то вирішив подивитися його на сайті гри. Він зовсім забув що використовував вхід через Google аккаунт тому спробував авторизуватися за допомогою своєї електронної адреси, на що система вивела йому повідомлення про некоректність введених даних. Усвідомивши свою помилку, Олег нарешті авторизувався та побачив, що він знаходиться на другому місці серед рейтингу

гравців! Він відчував радість, та був змотивований до подальшого дослідження цікавих історій і підвищення свого рівня у грі.

1.3.2 Історія користувача програмного продукту

Оскільки застосунок “Polar” призначено для публічного використання, то велика увага була приділена саме роботі із аудиторією, тому що саме користувачі будуть визначати його успіх. Отже, необхідно розуміти, як користувачі будуть взаємодіяти із даним застосунком. Щоб краще проаналізувати це питання, для деталізації можливих сценаріїв взаємодії користувача були написані User Story.

1. Як користувач продукту, я хочу мати змогу вивчати історію власного університету.
2. Як користувач продукту, я хочу мати змогу зареєструватися в системі для того, щоб отримувати останні новини гри.
3. Як користувач продукту, я хочу мати змогу зареєструватись у системі для того, аби зберігати свій прогрес у грі.
4. Як користувач продукту, я хочу мати змогу у будь-який момент відредагувати свої особисті дані в акаунті, аби виправити їх у разі помилкового заповнення.
5. Як користувач продукту, я хочу мати змогу зареєструватись у системі для того, аби зберігати свій прогрес у грі.
6. Як користувач продукту, я хочу мати змогу у будь-який момент відредагувати свої особисті дані в акаунті, аби виправити їх у разі помилкового заповнення.
7. Як користувач продукту, я хочу зберігати відвідані місця для того щоб мати можливість повторно прочитати їх історію.

8. Як користувач продукту, я хочу переглядати мапу яку зручно та легко читати для того, щоб краще орієнтуватися під час гри.

9. Як користувач продукту, я хочу отримувати підказки про місцезнаходження міток на карті.

10. Як користувач продукту, я хочу мати змогу відвідати сайт гри для більш детального ознайомлення з грою.

11. Як користувач продукту, я хочу мати змогу легко отримати доступ до додатку, завантаживши його через сайт або магазин додатків, аби зекономити свій час.

12. Як користувач продукту, я хочу мати змогу відвідувати сховані місця університету, на які я б не звернув увагу без застосунку, щоб розширити свої знання про університет.

1.3.3 Діаграма прецедентів програмного продукту

Діаграма прецедентів (Use Case UML-діаграма) включає в себе:

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

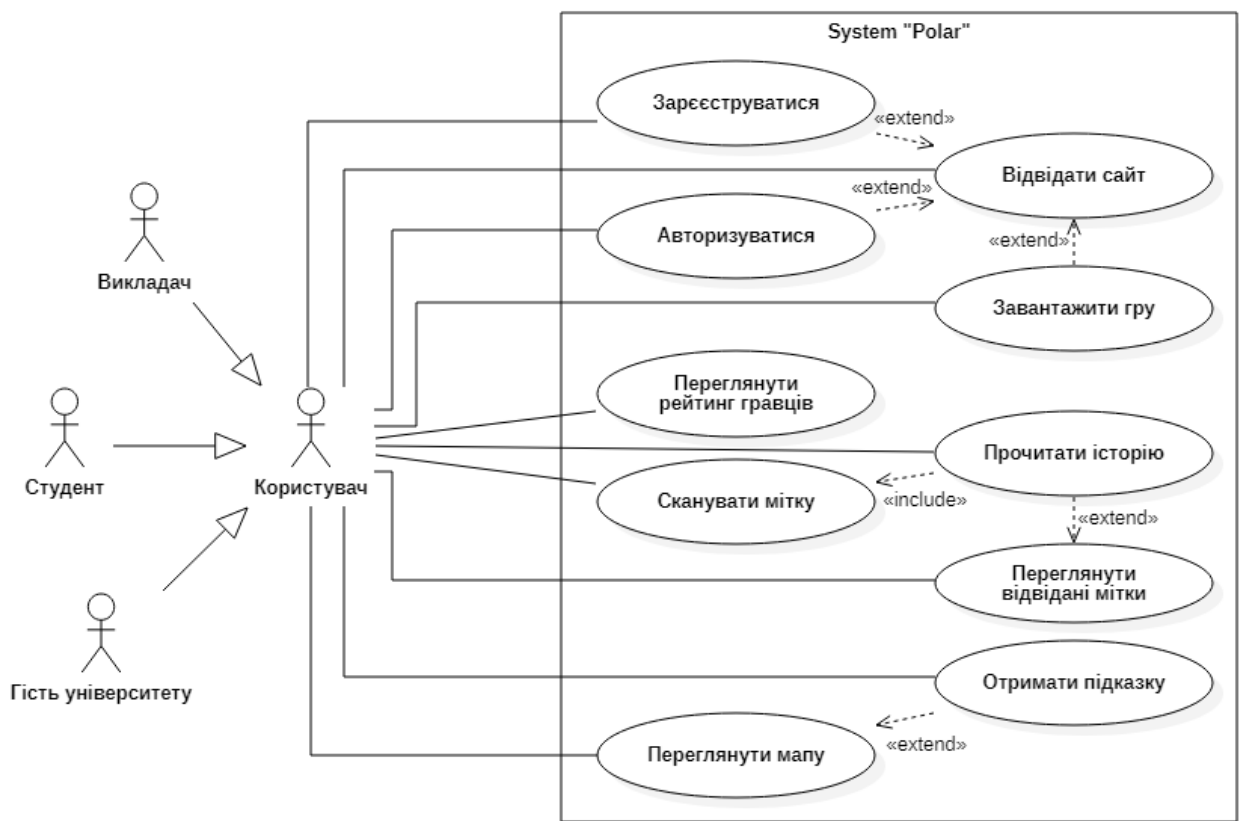


Рисунок 1.3.3.1 - Діаграма прецедентів для застосунку “Polar”

1.3.4 Сценарії використання прецедентів програмного продукту

1.3.4.1 Прецедент №1

1.3.4.1.1 Назва прецеденту: “Завантажити додаток”

1.3.4.1.2 Передумови виконання прецеденту: Зацікавлений в історії університету студент або викладач зайшов на сайт додатку.

1.3.4.1.3 Актори: Студент, викладач.

1.3.4.1.4 Гарантії успіху:

- працездатний додаток;
- можливість вивчати історію університету;
- можливість відкривати для себе сховані місця університету.
- можливість змагатись з іншими користувачами.

1.3.4.1.5 Основний успішний сценарій:

- 1) Користувач переходить на сайт додатку;
- 2) Користувач завантажує та встановлює додаток;
- 3) Користувач вперше запускає додаток

1.3.4.1.6 Альтернативні сценарії:

1.3.4.1.6.1 Користувач не захотів завантажувати додаток: використання додатку неможливе.

1.3.4.1.6.2 Операційна система користувача не підтримується додатком: використання додатку неможливе.

1.3.4.2 Прецедент №2

1.3.4.2.1 Назва прецеденту: “Зареєструватися”

1.3.4.2.2 Передумови виконання прецеденту: Усі події прецеденту “Завантажити додаток”.

1.3.4.2.3 Актори: Студент, викладач.

1.3.4.2.4 Гарантії успіху:

наявність працездатного облікового запису;

можливість зберегти свій прогрес у грі;

1.3.4.2.5 Основний успішний сценарій:

Користувач вводить свій e-mail та пароль;

Система створює новий аккаунт користувача;

Користувач входить у додаток.

1.3.4.2.6 Альтернативні сценарії:

1.3.4.2.6.1 Е-mail адреса, яку ввів користувач вже зареєстрована у системі: перехід до варіанту використання “Авторизуватися”.

1.3.4.3. Прецедент №3

1.3.4.3.1 Назва прецеденту: “Авторизуватися”

1.3.4.3.2 Передумови виконання прецеденту: Усі події прецеденту “Завантажити додаток” та “Зареєструватися”.

1.3.4.3.3 Актори: Студент, викладач.

1.3.4.3.4 Гарантії успіху:

можливість використання додатку;

можливість використання сайту.

1.3.4.3.5 Основний успішний сценарій:

Користувач вводить свій e-mail та пароль;

Система перевіряє дані користувача;

Користувач входить у додаток.

1.3.4.3.6 Альтернативні сценарії:

1.3.4.3.6.1 Користувач ввів некоректний пароль від облікового запису: перехід до варіанту використання “Авторизуватися”.

1.3.4.4 Прецедент №4

1.3.4.4.1 Назва прецеденту: “Відвідати сайт”

1.3.4.4.2 Передумови виконання прецеденту: Студент або викладач побачили/почули рекламу додатку та вирішили ознайомитися з ним.

1.3.4.4.3 Актори: Студент, викладач.

1.3.4.4.4 Гарантії успіху:

успішне завантаження сайту;

ознайомлення з інформацією про додаток;

можливість завантажити додаток через сайт;

1.3.4.4.5 Основний успішний сценарій:

Користувач бачить рекламу додатку;

Користувач переходить на сайт додатку;

Користувач ознайомлюється з інформацією про додаток;

1.3.4.4.6 Альтернативні сценарії:

1.3.4.4.6.1 Сайт додатку тимчасово недоступний: неможливість ознайомлення з додатком та його завантаження.

1.3.4.5 Прецедент №5

1.3.4.5.1 Назва прецеденту: “Переглянути мапу”

1.3.4.5.2 Передумови виконання прецеденту: Пункти прецеденту “Зареєструватися” або “Авторизуватися”

1.3.4.5.3 Актори: Студент, викладач.

1.3.4.5.4 Гарантії успіху:

успішний перегляд мапи гри;

можливість дізнатися місцезнаходження наступної мітки.

1.3.4.5.5 Основний успішний сценарій:

Користувач знаходиться у додатку;

Користувач переглядає мапу;

1.3.4.6 Прецедент №6

1.3.4.6.1 Назва прецеденту: “Отримати підказку”

1.3.4.6.2 Передумови виконання прецеденту: Пункти прецеденту “Переглянути мапу”.

1.3.4.6.3 Актори: Студент, викладач.

1.3.4.6.4 Гарантії успіху:

отримання інформації про приблизне місцезнаходження наступної мітки;

1.3.4.6.5 Основний успішний сценарій:

Користувач продивляється мапу;

Користувач отримує підказку;

1.3.4.7 Прецедент №7

1.3.4.7.1 Назва прецеденту: “Сканувати мітку”

1.3.4.7.2 Передумови виконання прецеденту: Студент або викладач успішно знайшли сховану мітку.

1.3.4.7.3 Актори: Студент, викладач.

1.3.4.7.4 Гарантії успіху:

можливість отримання інформації про історію місця, де знаходиться мітка;

1.3.4.7.5 Основний успішний сценарій:

Користувач знайшов мітку;

Користувач сканує мітку;

1.3.4.7.6 Альтернативні сценарії:

1.3.4.7.6.1 Помилка сканування мітки: повторна спроба сканування.

1.3.4.8 Прецедент №8

1.3.4.8.1 Назва прецеденту: “Прочитати історію”

1.3.4.8.2 Передумови виконання прецеденту: Пункти прецеденту “Сканувати мітку” або “Переглянути відвідані мітки”.

1.3.4.8.3 Актори: Студент, викладач.

1.3.4.8.4 Гарантії успіху:

отримання інформації про історію місця, де знаходиться мітка;

1.3.4.8.5 Основний успішний сценарій:

Користувач отримує інформацію в інтерактивному вигляді;

Користувач ознайомлюється з інформацією;

Користувач завершує процес перегляду історії.

1.3.4.9 Прецедент №9

1.3.4.9.1 Назва прецеденту: “Переглянути відвідані мітки”

1.3.4.9.2 Передумови виконання прецеденту: Користувач захотів повторно прочитати історію яка закріплена за міткою.

1.3.4.9.3 Актори: Студент, викладач.

1.3.4.9.4 Гарантії успіху:

відображення відвіданих міток;

можливість повторного перегляду історії.

1.3.4.9.5 Основний успішний сценарій:

Користувач вибирає відповідний пункт у меню додатку;

Користувач отримує список відвіданих міток;

Користувач вибирає мітку та повторно читає історію.

1.3.4.9.6 Альтернативні сценарії:

1.3.4.9.6.1 Користувач ще не відсканував жодної мітки: відображення пустого списку.

1.3.4.10 Прецедент №10

1.3.4.10.1 Назва прецеденту: “Переглянути рейтинг гравців”

1.3.4.10.2 Передумови виконання прецеденту: Користувач захотів порівняти свою успішність з успішністю інших користувачів.

1.3.4.10.3 Актори: Студент, викладач.

1.3.4.10.4 Гарантії успіху:

відображення рейтингу гравців;

1.3.4.10.5 Основний успішний сценарій:

Користувач вибирає відповідний пункт у меню додатку або на сайті;

Користувач отримує рейтинг гравців.

1.4 Функціональні вимоги до програмного продукту

1.4.1. Багаторівнева класифікація функціональних вимог

З метою упорядкування функцій ПП було створено багаторівневу класифікацію функціональних вимог (Functional Requirements - FR), виявлених в сценаріях використання прецедентів. Було використано такий формат структур, який має багато спільного зі структурою декомпозиції робіт (Work Breakdown Structure - WBS), але без вказівки на складність і тривалість. Кожну функцію в ієрархії було унікальним ієрархічним ідентифікатором FRi, тобто, FR1, FR1.1 і т.д. При визначенні функцій функції першого рівня ієрархії було визначено на основі назв прецедентів, а функції другого рівня ієрархії було визначено на основі опису кроків основного успішного та альтернативного сценаріїв роботи прецедентів. Повні назви функцій надано у формі таблиці (таблиця 1.4.1.1).

Ідентифікатор функції (назва)	Назва функції
FR 1	Завантаження застосунку
FR 1.1	Завантаження застосунку на пристрій користувача з власного інтернет-ресурсу
FR 2	Реєстрація користувача
FR 2.1	Створення запиту у користувача на отримання його e-mail та паролю
FR 2.2	Створення нового аккаунту користувача
FR 2.3 (Alt)	Виведення повідомлення що введена e-mail адреса вже зареєстрована
FR 2.4 (Alt)	Запит у користувача паролю від вже існуючого облікового запису
FR 3	Авторизація користувача
FR 3.1	Створення запиту у користувача на отримання його e-mail та паролю
FR 3.2	Перевірка даних користувача
FR 3.3 (Alt)	Вивід повідомлення про некоректність введених даних
FR 3.4 (Alt)	Повторний запит e-mail та паролю
FR 4	Сканування мітки

FR 4.1	Захоплення мітки та її ідентифікація
FR 4.2 (Alt)	Вивід повідомлення про невдале сканування
FR 4.3 (Alt)	Спроба повторного сканування
FR 5	Показ історії
FR 5.1	Створення AR-записки на екрані користувача
FR 5.2	Розгортання AR-записки
FR 5.3	Показ історії на записці
FR 6	Перегляд відвіданих міток
FR 6.1	Одержання списку відвіданих міток
FR 6.2	Вибір відвіданої мітки та перегляд історії (без AR)
FR 7	Перегляд рейтингу гравців
FR 7.1	Відображення рейтингу на сайті
FR 7.2	Відображення рейтингу в застосунку

Таблиця 1.4.1.1 - Багаторівнева класифікація функціональних вимог

Опис ієрархічної WBS-структури багаторівневої класифікації функціональних вимог представлено на рисунку 1.4.1.1.

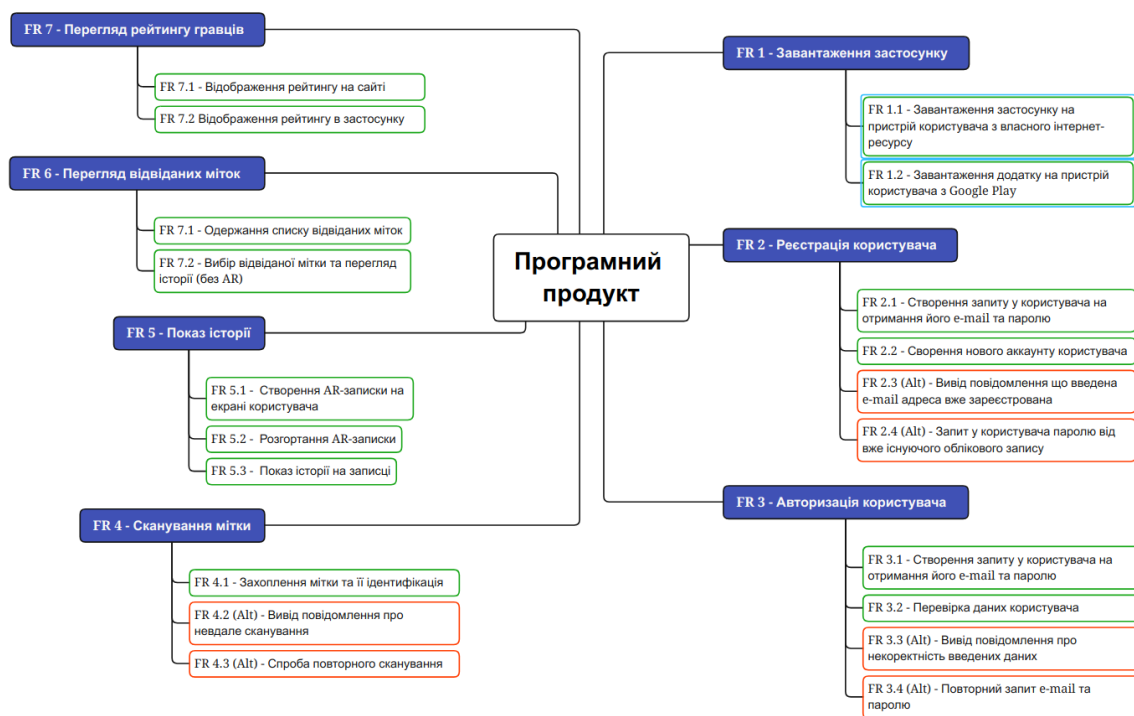


Рисунок 1.4.1.1 - Ієрархічна WBS-структура багаторівневої класифікації

1.4.2 Функціональний аналіз існуючих програмних продуктів

З метою швидкого прототипування роботи ПП проведіть функціональний аналіз існуючих ПП у вигляді таблиці функцій, де:

- рядки – ідентифікатори функцій нижнього рівня ієрархії (листя в дереві ієрархії);
- осередки - позначки про реалізацію функції в ПП, наприклад, позначка «+» – функція реалізована в ПП, позначка «-» – функція не реалізована.

Функціональний аналіз існуючих ПП представлено в таблиці 1.4.2.1.

Ідентифікатор функції	Florence City Guide	Questo	Franklin Institute AR	Kennedy Space Centre AR	National Museum of Singapore AR
FR 1.1	-	+	+	-	-

FR 2.1	+	+	-	-	-
FR 2.2	+	+	-	-	-
FR 2.3 (Alt)	+	+	-	-	-
FR 2.4 (Alt)	+	+	-	-	-
FR 3.1	+	+	-	-	-
FR 3.2	+	+	-	-	-
FR 3.3 (Alt)	+	+	-	-	-
FR 3.4 (Alt)	+	+	-	-	-
FR 4.1	+	+	+	+	+
FR 4.2 (Alt)	+	+	+	+	+
FR 4.3 (Alt)	+	+	+	+	+
FR 5.1	-	-	-	-	-
FR 5.2	-	-	-	-	-
FR 5.3	+	+	+	+	+
FR 6.1	-	-	-	-	-
FR 6.2	-	-	-	-	-
FR 7.1	-	-	-	-	-
FR 7.2	-	-	-	-	-

Таблиця 1.4.2.1 - Функціональний аналіз існуючих ПП

1.5 Нефункціональні вимоги до програмного продукту

1.5.1 Опис зовнішніх інтерфейсів

1.5.1.1 Опис інтерфейсів користувача

1.5.1.1.1 Опис INPUT-інтерфейсів користувача

У даному пункті було виконано опис інтерфейсу користувача з урахуванням INPUT-інформаційного потоку. Для опису було проаналізовано кожну функцію ПП нижнього (2-го) рівня ієрархії з метою вибору відповідного інтерфейсу.

INPUT-інтерфейси були підібрані таким чином, щоб вони змогли наблизити користувача до досягнення мети ПП, а не навпаки – ускладнити досягнення його мети при використанні ПП.

Для однієї функції можливе використання декількох засобів. В стовпчику «Особливості використання» надайте можливі деталі використання інтерфейсу, якщо вони є. Результати представлені у таблиці 1.5.1.1.1.1.

Ідентифікатор функції	Засіб INPUT-потoku	Особливості використання
FR 1.1	Стандартна комп'ютерна клавіатура; 2/3-кнопочний маніпулятор типу "миша"	Застосування клавіатури для пошуку сайту в мережі Інтернет, застосування "миші" для навігації по сайту та натискання кнопки "завантажити".
FR 2.1	Стандартна комп'ютерна клавіатура;	Застосування клавіатури для введення паролю та електронної пошти

FR 2.2	2/3-кнопочний маніпулятор типу "миша";	Застосування "миші" - для надсилення реєстраційних даних на сервер
FR 2.4 (Alt)	Стандартна комп'ютерна клавіатура;	Запит у користувача пароллю від вже існуючого облікового запису
FR 3.1	Стандартна комп'ютерна клавіатура;	Застосування клавіатури для введення пароллю та електронної пошти
FR 3.4 (Alt)	Стандартна комп'ютерна клавіатура;	Застосування клавіатури для введення пароллю та електронної пошти
FR 4.1	Камера смартфону	Застосування камери для сканування мітки
FR 4.3 (Alt)	Камера смартфону	Застосування камери для сканування мітки
FR 6.1	Тачскрін смартфону	Застосування тачскріну для вибору списку просканованих точок шляхом натискання на відповідну кнопку на екрані
FR 6.2	Тачскрін смартфону	Застосування тачскріну для вибору відвіданої мітки та перегляд історії


Таблиця 1.5.1.1.1.1 - Результати аналізу INPUT-потоків

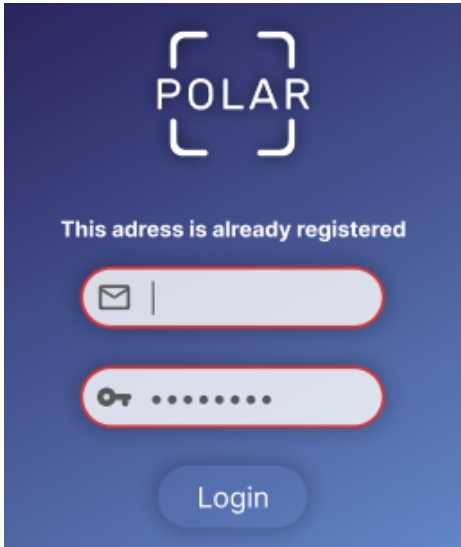
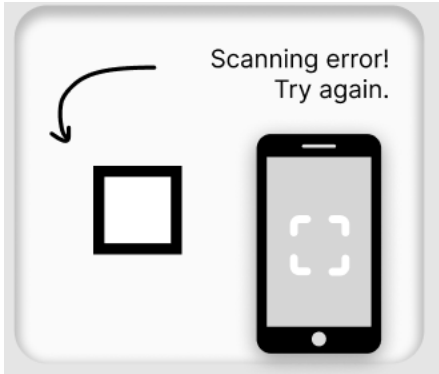
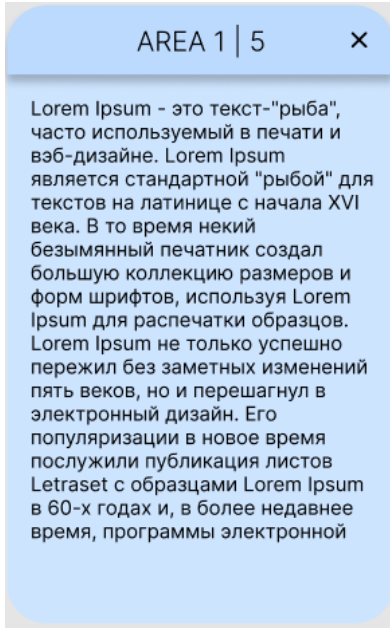
1.5.1.1.2 Опис OUTPUT-інтерфейсів користувача

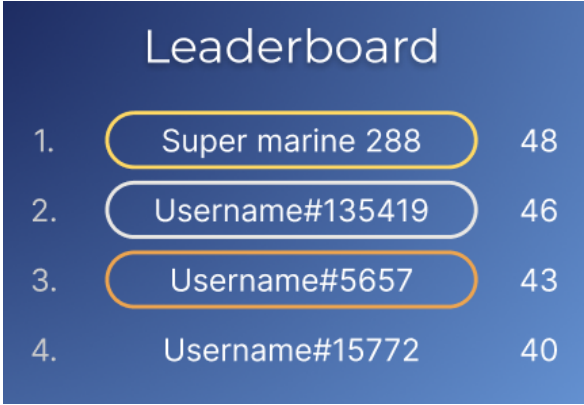
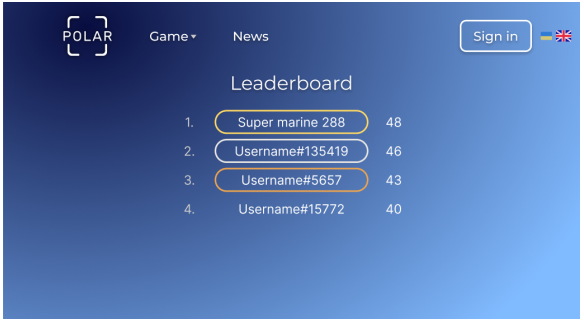
У даному пункті було виконано опис інтерфейсу користувача з урахуванням OUTPUT-інформаційного потоку. Для опису було проаналізовано кожну функцію ПП нижнього (2-го) рівня ієрархії з метою вибору відповідного інтерфейсу.

OUTPUT-інтерфейси були підібрані таким чином, щоб вони змогли наблизити користувача до досягнення мети ПП, а не навпаки – ускладнити досягнення його мети при використанні ПП. Для кожного пункту були представлені Москир-прототипи.

Для однієї функції можливе використання декількох засобів. В стовпчику «Особливості використання» надайте можливі деталі використання інтерфейсу, якщо вони є. Результати представлені у таблиці 1.5.1.1.2.1.

Ідентифікатор функції	Засіб INPUT-потоків	Особливості використання
FR 2.3 (Alt)	Графічний інтерфейс	

FR 3.3 (Alt)	Графічний інтерфейс	
FR 4.2 (Alt)	Графічний інтерфейс	
FR 5.1	Графічний інтерфейс	

FR 7.1	Графічний інтерфейс	
FR 7.2	Графічний інтерфейс	

Таблиця 1.5.1.1.2.1 - Результати аналізу OUTPUT-потоків

1.5.1.2 Опис інтерфейсу із зовнішніми пристроями

Для опису цього пункту було проаналізовано кожну функцію ПП нижнього (2-го) рівня ієрархії з метою вибору зовнішніх пристроїв (обладнання). Прилади були підібрані таким чином, щоб вони змогли наблизити користувача до досягнення мети ПП, а не навпаки – ускладнити досягнення його мети при використанні ПП.

Результат аналізу представлено у таблиці 1.5.1.2.1.

Ідентифікатор функції	Зовнішній пристрій
FR 1.1	Смартфон, планшет
FR 2.1	Смартфон, планшет, ПК, Ноутбук
FR 2.2	Смартфон, планшет, ПК, Ноутбук
FR 2.3 (Alt)	Смартфон, планшет, ПК, Ноутбук
FR 2.4 (Alt)	Смартфон, планшет, ПК, Ноутбук
FR 3.1	Смартфон, планшет, ПК, Ноутбук
FR 3.2	Смартфон, планшет, ПК, Ноутбук
FR 3.3	Смартфон, планшет, ПК, Ноутбук
FR 3.4 (Alt)	Смартфон, планшет, ПК, Ноутбук
FR 3.5 (Alt)	Смартфон, планшет, ПК, Ноутбук
FR 4.1	Смартфон, планшет
FR 4.2 (Alt)	Смартфон, планшет
FR 4.3 (Alt)	Смартфон, планшет
FR 5.1	Смартфон, планшет
FR 5.2	Смартфон, планшет
FR 5.3	Смартфон, планшет
FR 6.1	Смартфон, планшет

FR 6.2	Смартфон, планшет
FR 7.1	Смартфон, планшет, ПК, ноутбук
FR 7.2	Смартфон, планшет

Таблиця 1.5.1.2.1 - Результати аналізу зовнішніх пристроїв

1.5.1.3 Опис програмних інтерфейсів

У даному пункті було виконано опис версій операційних систем та програмних бібліотек, які знадобляться при реалізації більшості функцій ПП.

Результати представлені у таблиці 1.5.1.3.1

Програмний інтерфейс	Версія
OS Windows	7 SP1 ESU +
OS Ubuntu Linux	16.04(LTS) / 18.04(LTS) / 19.10 +
Nginx	1.21.3
MySQL	8.0.26
.NET SDK	5.0 +
Entity Framework Core	6.0
Angular CLI	12 +
Unity	2021.x +
AR Foundation	4.2 +

Таблиця 1.5.1.3.1 - Опис програмних інтерфейсів

1.5.1.4 Опис інтерфейсів передачі інформації

У даному пункті було виконано опис інтерфейсів передачі інформації, які знадобляться при реалізації більшості функцій ПП.

Результати представлені у таблиці 1.5.1.4.1.

№	Назва
1	Ethernet
2	Wi-Fi
3	3G / 4G LTE / 5G

Таблиця 1.5.1.4 - Опис інтерфейсів передачі інформації

1.5.1.5 Опис атрибутів продуктивності

Із множини атрибутів якості в даній курсовій роботі було розглянуто лише продуктивність.

У цьому пункті вказані такі характеристики продуктивності:

- максимальний час реакції ПП на дії користувачів;
- максимальна кількість одночасно обслуговуваних користувачів.

Для опису було проаналізовано кожну функцію ПП нижнього (2-го) рівня ієрархії. Результати занесено у таблицю 1.5.1.5.1.

Ідентифікатор функції	Максимальний час реакції ПП на дії користувачів (секунди)	Максимальна кількість одночасно обслуговуваних користувачів
FR 1.1	4	15
FR 2.1	2	5

FR 2.2	2	5
FR 2.3 (Alt)	2	5
FR 2.4 (Alt)	2	5
FR 3.1	2	5
FR 3.2	2	5
FR 3.3	2	5
FR 3.4 (Alt)	2	5
FR 3.5 (Alt)	2	5
FR 4.1	5	10
FR 4.2 (Alt)	1	10
FR 4.3 (Alt)	1	10
FR 5.1	3	20
FR 5.2	4	20
FR 5.3	1	20
FR 6.1	4	20
FR 6.2	4	20
FR 7.1	4	15
FR 7.2	4	15

Таблиця 1.5.1.5.1 - Опис атрибутів продуктивності

2. ПЛАНУВАННЯ ПРОЦЕСУ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ

2.1 Планування ітерацій розробки програмного продукту

З метою забезпечення вимог таких рекомендацій IEEE-стандарту, як необхідність, корисність при експлуатації, здійсненність функціональних вимог до ПП, визначено функціональні пріоритети, які будуть використані при плануванні ітерацій розробки ПП. Результати представлено в таблиці 2.1

У таблиці 2.1.1 було виконано опис функціональних пріоритетів для процесу розробки застосунку.

Ідентифікатор функції	Функціональні залежності	Вплив на досягнення мети, %	Пріоритет функції
F 1.1	-	4.29%	M
F 2.1	-	4.29%	M
F 2.2	-	4.29%	M
F 2.3 (Alt)	-	4.29%	M
F 2.4 (Alt)	-	4.29%	M
F 3.1	F 2.2	4.29%	M
F 3.2	F 3.1	4.29%	M
F 3.3 (Alt)	F 3.2	4.29%	M
F 3.4 (Alt)	F 3.2	4.29%	M
F 4.1	-	4.29%	M

F 4.2 (Alt)	F 4.1	4.29%	M
F 4.3 (Alt)	F 4.1	4.29%	M
F 5.1	F 4.1	2%	S
F 5.2	F 4.1	2%	S
F 5.3	F 4.1	2%	S
F 6.1	FR 3.2	4.29%	M
F 6.2	FR 3.2	4.29%	M
F 7.1	-	2%	S
F 7.2	FR 3.2	2%	S

Таблиця 2.1.1 - Опис функціональних пріоритетів

2.2 Концептуальний опис архітектури програмного продукту

“Polar” можна віднести до Mobile Application (MA) через те, що він:

- працює на портативних пристроях;
- підтримує простий UI, що підходить для використання на невеликому екрані;
- не залежить від пристрою (єдиний фактор - наявність камери);
- залежить від можливості мережевого підключення.

У цьому пункті також було описано, відповідальні за три рівня роботи, які називають:

- рівень представлення – Presentation Level (PL), який містить компоненти зовнішнього інтерфейсу (програмний інтерфейс користувача), наприклад, Client Operation System, Input, Output,
- рівень бізнес-логіки – Business Level (BL), який містить апаратно-програмні компоненти обробки даних, наприклад, Server Operation System, Application Server;
- рівень зберігання даних – Access Level (AL), який містить апаратно-програмні компоненти керування даними, наприклад, Server Operation System, DataBase Server.

Для опису Presentation, Business та Access рівнів було створено UML-діаграму розгортання (рисунок 2.2.1).

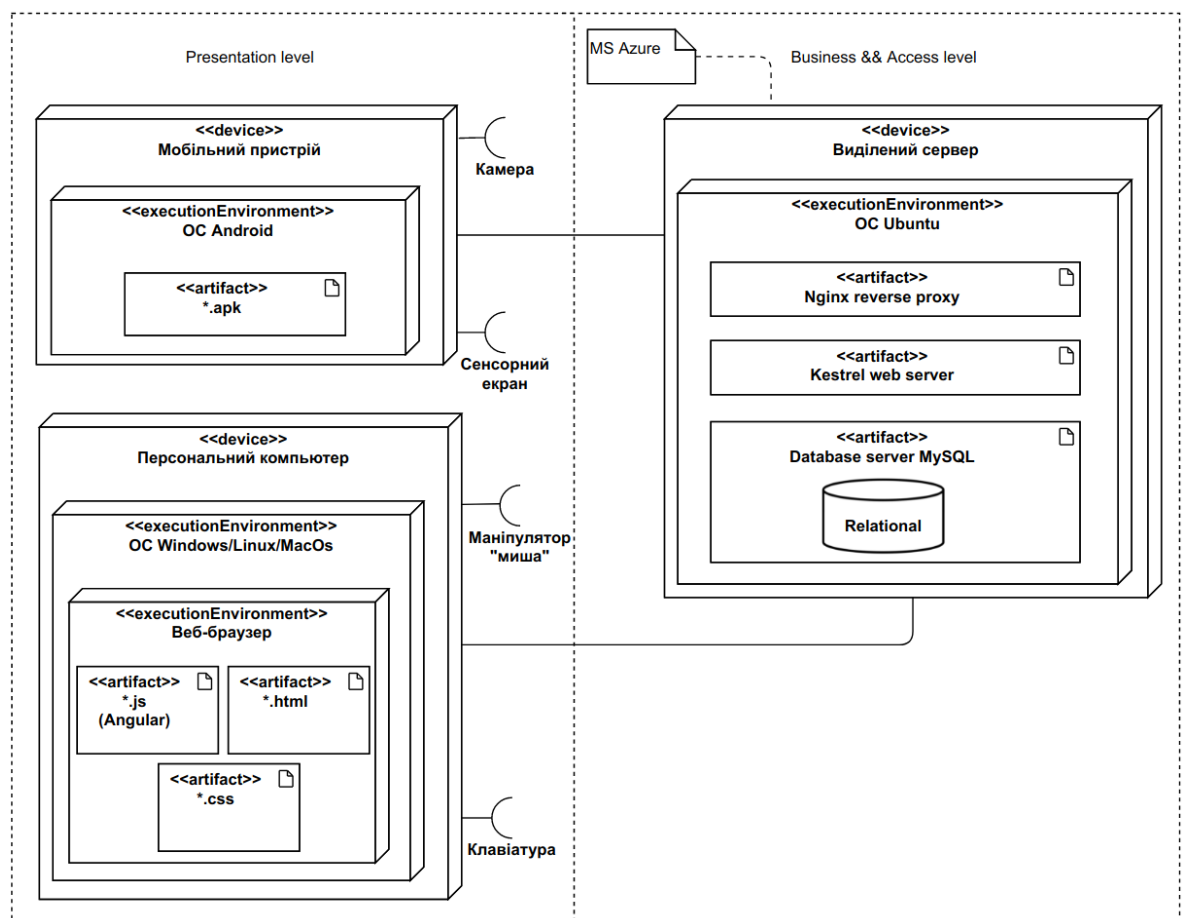


Рисунок 2.2.1 - UML-діаграма розгортання

2.3 План розробки програмного продукту

2.3.1 Оцінка трудомісткості розробки програмного продукту

2.3.1.1 Визначення невиправленого показника UUCP (Unadjusted Use Case Points)

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 мережових 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 (таблиця 2.3.1.1).

Тип актора	Ваговий коефіцієнт	Кількість акторів	Результат
Простий	1	1	1
Середній	2	0	0
Складний	3	3	9

Таблиця 2.3.1.1 - Вагові коефіцієнти акторів

The Unadjusted Actor Weight (UAW) = $1 + 0 + 9 = 10$

2.3.1.2 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 (таблиця 2.3.1.2).

Тип прецедента	Кількість кроків сценарію	Ваговий коефіцієнт	Кількість прецедентів	Результат
Простий	≤ 3	5	5	25
Середній	4-7	10	2	20
Складний	> 7	15	0	0

Таблиця 2.3.1.2 - Вагові коефіцієнти прецедентів

The Unadjusted Use Case Weight (UUCW) = $25 + 20 + 0 = 45$

UUCP = $10 + 45 = 55$

2.3.1.3 Визначення технічної складності проекту

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

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

0 означає відсутність значимості показника для даного проекту,

5 - високу значимість).

Значення TCF обчислюється за формулою:

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

Показники технічної складності проекту TCF представлені у таблиці 2.3.1.3.

Показник	Опис показника	Вага	Значимість	Результат
T1	Розподілена система	2	3	6
T2	Висока продуктивність (пропускна здатність)	1	3	3
T3	Робота кінцевих користувачів в режимі онлайн	1	3	3
T4	Складна обробка даних	-1	3	-3
T5	Повторне використання коду	1	1	1
T6	Простота установки	0,5	4	2
T7	Простота використання	0,5	4	2
T8	Переносимість	2	1	2
T9	Простота внесення змін	1	2	2
T10	Паралелізм	1	2	2
T11	Спеціальні вимоги до безпеки	1	3	3
T12	Безпосередній доступ до системи з боку зовнішніх користувачів	1	0	0
T13	Спеціальні вимоги до навчання користувачів	1	1	0

Таблиця 2.3.1.3.1 - Показники технічної складності проекту TCF

$$TCF = 0,6 + (0,01 * 23) = 0,83$$

2.3.1.4 Визначення рівня кваліфікації розробників

Рівень кваліфікації розробників (EF - Environmental Factor) було обчислено з урахуванням наступних показників (таблиця 2.3.1.4.1).

Кожному показнику присвоюється значення в діапазоні від 0 до 5.

Для показників F1 - F4: 0 означає відсутність, 3 - середній рівень, 5 - високий рівень.

Для показника F5: 0 означає відсутність мотивації, 3 - середній рівень, 5 - високий рівень мотивації.

Для F6: 0 означає високу нестабільність вимог, 3 - середню, 5 - стабільні вимоги.

Для F7: 0 означає відсутність фахівців з частковою зайнятістю, 3 - середній рівень, 5 - всі фахівці з частковою зайнятістю.

Для показника F8: 0 означає просту мову програмування, 3 - середню складність, 5 - високу, складність.

Показник	Опис показника	Вага	Рівень	Результат
F1	Знайомство з технологією	1,5	2	3
F2	Досвід розробки додатків	0,5	2	1
F3	Досвід використання об'єктно-орієнтованого підходу	1	3	3
F4	Наявність провідного	0,5	0	0

	аналітика			
F5	Мотивація	1	5	5
F6	Стабільність вимог	2	4	8
F7	Часткова зайнятість	-1	3	-3
F8	Складні мови програмування	-1	0	0

Таблиця 2.3.1.4.1 - Показники рівня кваліфікації розробників

$$EF = 1,4 + (-0,03 * 17) = 0,89$$

2.3.1.5 Остаточне значення UCP (Use Case Points)

Остаточне значення UCP (Use Case Points) було розраховано за наступною формулою:

$$UCP = UUCP * TCF * EF.$$

$$UCP = 55 * 0,83 * 0,89 = 40,63.$$

2.3.1.6 Оцінка трудомісткості проекту

Під час виконання роботи було обрано значення в 20 люд.-год на одну UCP.

В такому випадку трудомісткість проекту складає:

$$39,16 * 20 = 783,2 \text{ (люд.-год)}$$

2.3.2 Визначення дерева робіт з розробки програмного продукту

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

Кожна функція 1-го рівня ієрархії перетворюється в Work Package (WP)

Кожна функція 2-го рівня ієрархії перетворюється в Work Task (WT).

Для кожної задачі визначаються підзадачі - Work SubTask (WST) з урахуванням базових процесів розробки програмних модулів: проектування, конструювання, модульне тестування, збірка та системне тестування. Отримати доступ до дерева робіт можна за посиланням, представленим нижче.

<https://drive.google.com/file/d/1kTkEYa0AjJlBeILpmtvsUqddKfuG9jhQ/view?usp=sharing>

2.3.3 Графік робіт з розробки програмного продукту

2.3.3.1 Таблиця з графіком робіт

Для кожної підзадачі було визначено виконавця, що було зафіксовано у вигляді таблиці. У таблиці наявні такі колонки: WST-ідентифікатор, дата початку, дні, дата завершення, виконавець. Датою завершення роботи було прийнято 9 грудня 2021 року, як останній день навчання осіннього семестру.

Графік робіт представлено у таблиці 2.3.3.1.1.

WST	Дата початку	Дні	Дата завершення	Виконавець
WST 1.1.1	15.10.2021	2	16.10.2021	Кисельов Д.С.
WST 1.1.2	17.10.2021	2	18.10.2021	Кисельов Д.С.
WST 1.1.3	19.10.2021	1	19.10.2021	Кисельов Д.С.
WST 1.2.1	20.10.2021	2	21.10.2021	Кисельов Д.С.
WST 1.2.2	22.10.2021	2	23.10.2021	Кисельов Д.С.
WST 1.2.3	24.10.2021	1	24.10.2021	Кисельов Д.С.
WST 1.3.1	25.10.2021	2	26.10.2021	Кисельов Д.С.
WST 1.3.2	27.10.2021	1	27.10.2021	Кисельов Д.С.
WST 3.1.1	28.10.2021	3	30.10.2021	Кисельов Д.С.
WST 3.1.2	31.10.2021	3	2.11.2021	Кисельов Д.С.

WST 3.1.3	3.11.2021	3	5.11.2021	Кисельов Д.С.
WST 3.2.1	6.11.2021	3	8.11.2021	Кисельов Д.С.
WST 3.2.2	9.11.2021	3	11.11.2021	Кисельов Д.С.
WST 3.2.3	12.11.2021	3	14.11.2021	Кисельов Д.С.
WST 3.3.1	15.11.2021	2	16.11.2021	Кисельов Д.С.
WST 3.3.2	17.11.2021	2	18.11.2021	Кисельов Д.С.
WST 3.3.3	19.11.2021	2	20.11.2021	Кисельов Д.С.
WST 3.4.1	21.11.2021	2	22.11.2021	Кисельов Д.С.
WST 3.4.2	23.11.2021	1	23.11.2021	Кисельов Д.С.
WST 3.4.3	24.11.2021	1	24.11.2021	Кисельов Д.С.
WST 3.5.1	25.11.2021	2	26.11.2021	Кисельов Д.С.
WST 3.5.2	27.11.2021	1	27.11.2021	Кисельов Д.С.
WST 4.1.1	15.10.2021	2	16.10.2021	Фаюк К.С
WST 4.1.2	17.10.2021	2	18.10.2021	Фаюк К.С
WST 4.1.3	19.10.2021	1	19.10.2021	Фаюк К.С
WST 4.2.1	20.10.2021	2	21.10.2021	Фаюк К.С
WST 4.2.2	22.10.2021	2	23.10.2021	Фаюк К.С
WST 4.2.3	24.10.2021	2	25.10.2021	Фаюк К.С
WST 4.3.1	26.10.2021	2	27.10.2021	Фаюк К.С
WST 4.3.2	28.10.2021	1	28.10.2021	Фаюк К.С
WST 4.3.3	29.10.2021	1	29.10.2021	Фаюк К.С
WST 4.4.1	30.10.2021	1	30.10.2021	Фаюк К.С
WST 4.4.2	31.10.2021	1	31.10.2021	Фаюк К.С
WST 5.1.1	1.11.2021	2	2.11.2021	Фаюк К.С
WST 5.1.2	3.11.2021	1	3.11.2021	Фаюк К.С
WST 5.1.3	4.11.2021	1	4.11.2021	Фаюк К.С
WST 5.2.1	5.11.2021	3	7.11.2021	Фаюк К.С
WST 5.2.2	8.11.2021	2	9.11.2021	Фаюк К.С
WST 5.2.3	10.11.2021	1	10.11.2021	Фаюк К.С

WST 5.3.1	11.11.2021	2	12.11.2021	Фаюк К.С
WST 5.3.2	13.11.2021	1	13.11.2021	Фаюк К.С
WST 5.3.3	14.11.2021	1	14.11.2021	Фаюк К.С
WST 5.4.1	15.11.2021	2	16.11.2021	Фаюк К.С
WST 5.4.2	17.11.2021	1	17.11.2021	Фаюк К.С
WST 6.1.1	18.11.2021	3	20.11.2021	Фаюк К.С
WST 6.1.2	21.11.2021	1	21.11.2021	Фаюк К.С
WST 6.1.3	22.11.2021	2	23.11.2021	Фаюк К.С
WST 6.2.1	24.11.2021	1	24.11.2021	Фаюк К.С
WST 6.2.2	25.11.2021	1	25.11.2021	Фаюк К.С
WST 6.2.3	26.11.2021	1	26.11.2021	Фаюк К.С
WST 6.3.1	27.11.2021	1	27.11.2021	Фаюк К.С
WST 6.3.2	28.11.2021	1	28.11.2021	Фаюк К.С
WST 2.1.1	15.10.2021	2	16.10.2021	Ігнатюк А.Г.
WST 2.1.2	17.10.2021	2	18.10.2021	Ігнатюк А.Г.
WST 2.1.3	19.10.2021	2	20.10.2021	Ігнатюк А.Г.
WST 2.2.1	21.10.2021	2	22.10.2021	Ігнатюк А.Г.
WST 2.2.2	23.10.2021	2	24.10.2021	Ігнатюк А.Г.
WST 2.2.3	25.10.2021	2	26.10.2021	Ігнатюк А.Г.
WST 2.3.1	27.10.2021	1	27.10.2021	Ігнатюк А.Г.
WST 2.3.2	28.10.2021	1	28.10.2021	Ігнатюк А.Г.
WST 2.3.3	29.10.2021	2	30.10.2021	Ігнатюк А.Г.
WST 2.4.1	31.10.2021	1	31.10.2021	Ігнатюк А.Г.
WST 2.4.2	1.11.2021	1	1.11.2021	Ігнатюк А.Г.
WST 2.4.3	2.11.2021	3	4.11.2021	Ігнатюк А.Г.
WST 2.5.1	5.11.2021	1	5.11.2021	Ігнатюк А.Г.
WST 2.5.2	6.11.2021	1	6.11.2021	Ігнатюк А.Г.
WST 7.1.1	7.11.2021	3	9.11.2021	Ігнатюк А.Г.
WST 7.1.2	10.11.2021	3	12.11.2021	Ігнатюк А.Г.

WST 7.1.3	13.11.2021	2	14.11.2021	Ігнатюк А.Г.
WST 7.2.1	15.11.2021	3	17.11.2021	Ігнатюк А.Г.
WST 7.2.2	18.11.2021	3	20.11.2021	Ігнатюк А.Г.
WST 7.2.3	21.11.2021	2	22.11.2021	Ігнатюк А.Г.
WST 7.3.1	23.11.2021	3	25.11.2021	Ігнатюк А.Г.
WST 7.3.2	26.11.2021	3	28.11.2021	Ігнатюк А.Г.

Таблиця 2.3.3.1.1 - Таблиця графіку робіт

2.3.3.2 Діаграма Ганта

У даному пункті курсової роботи було створено діаграма Ганта. Дану діаграму було створено на основі таблицю графіку робіт. Дати початку з колонки таблиці графіку робіт було перетворено у колонки діаграми Ганта.

Дана діаграма (складається із смуг (вісь Y), орієнтованих уздовж осі часу (вісь X). Кожна смуга – окрема підзадача в проекті, її кінці - моменти початку і завершення роботи, її протяжність - тривалість роботи. Мета створення діаграми Ганта - візуально показати послідовність процесів та можливість паралельного виконання робіт.

Отримати доступ до діаграми Ганта можна за наступним посиланням:

<https://docs.google.com/spreadsheets/d/12QiXc2NqPtqyiAPJ1mafznA1WX4uswiT85YQmt15QNk/edit?usp=sharing>