

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

«Система електронного голосування на основі цифрового підпису
“SecureVote”»

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

Виконали:
студенти 3-го курсу
групи УІ-191
Савіцька Ю. О.
Перевірив:
Блажко О. А.

Одеса-2021

Анотація

В курсовій роботі розглядається процес створення програмного продукту «Система електронного голосування на основі цифрового підпису “SecureVote”» на етапах визначення вимог до програмного продукту та планування процесів розробки.

Робота виконувалась в команді з одного учасника: Савіцька Юлія Олегівна.

В робота пов'язана з такими матеріальними потребами споживача як безпека. Аналіз вказаних потреб визначає інформаційну потребу - цінність, актуальність.

При визначенні ступеня готовності існуючих програмних продуктів до вирішення інформаційної потреби проаналізовано наступні програмні продукти: ГОЛОС, Віче, Mobile-RADA.

Поточну версію пояснювальної записки до результатів роботи розміщено на *GitHub*-репозиторії за адресою:

<https://github.com/JulSav/ElectronicVoiceSystem>

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

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

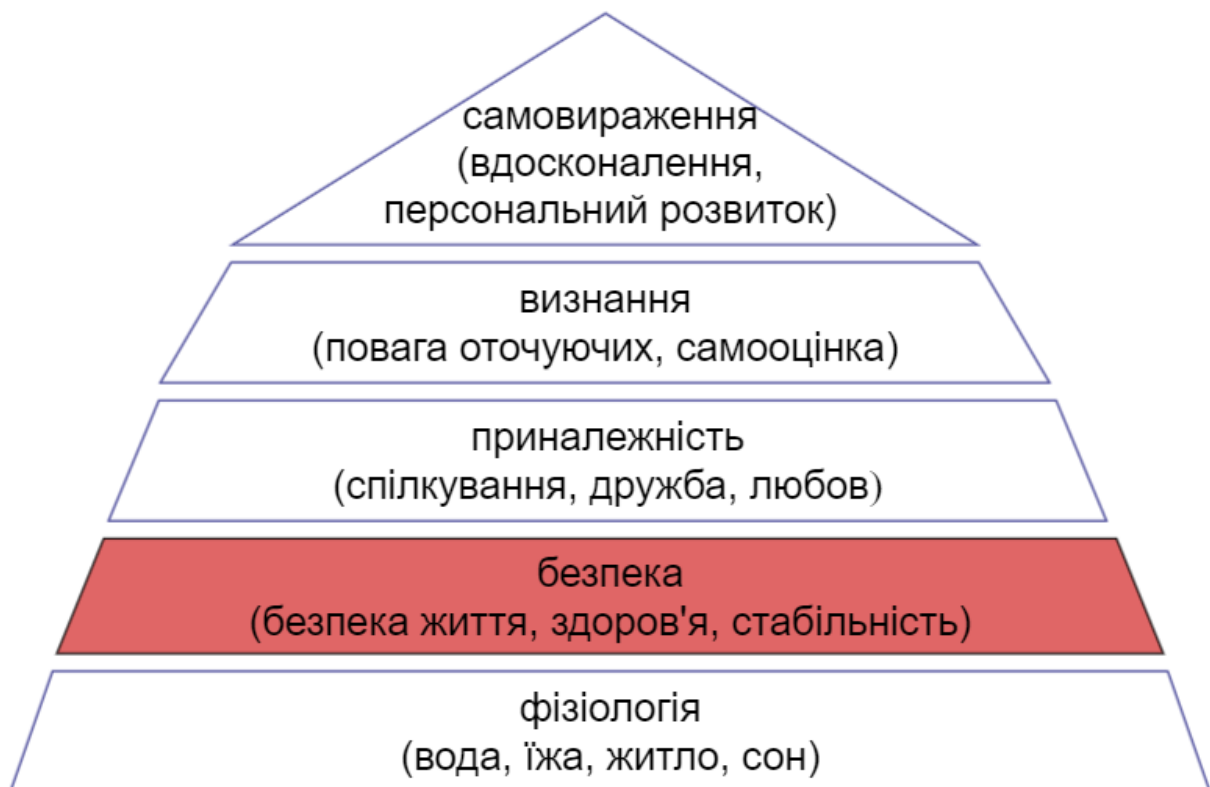


Рис. 1.1 – Приклад ієрархії потреби споживача

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

№	Загальний опис проблеми
1	Майже неможливо захиститися від фальсифікації та забезпечити анонімність
2	Багато часу уходить на підрахунок голосів
3	Під час епідемій та інших непередбачених випадках неможливо отримати доступ до голосування

1.2.1.2 Опис цільової групи споживачів

Споживач - це громадянин України від 18 років, адже за законодавством України, тільки люди, які дійшли до віку 18 років мають право голосу.

1.2.1.3 Метричний опис проблеми споживача

№	Метричні показники незадоволеності споживача
1	Низький рівень анонімності при оффлайн голосуванні. (Низький рівень безпечності)
2	Низький рівень швидкості при підрахуванні голосів. (Низький рівень швидкості)
3	Депутати не можуть бути присутні під час епідемії (Низький рівень доступності)

1.2.2 Мета створення програмного продукту

1.2.2.1 Проблемний аналіз існуючих програмних продуктів

№	Назва продукту	Вартість	Ступі нь готов ності	Примітка
1	ГОЛОС	Платно (дорого)	1	Недостатньо безпечний
2	Віче	Платно (дорого)	1	Система потребує попередньої закупки апаратної система

3	Mobile-RAD A	Платно (найдорожча серед аналогів)	1	Важкість перевірки достовірності особистості;
---	-----------------	---	---	--

1.2.2.2 Мета створення програмного продукту

Підвищення якості голосування через автоматизацію процесів, зменшення людського фактору, забезпечення анонімності та безпеки.

1.2.3 Назва програмного продукту

1.2.3.1 Гасло програмного продукту

SecureVote - безпечне голосування.

1.2.3.2 Логотип програмного продукту



1.3 Вимоги користувача до програмного продукту

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

Хочу розповісти історію звичайної людини, яка живе у місті Одеса, в Україні. Це молодий хлопець 19 років, який зміг вступити до ОНПУ у 2019

році. Його ім'я Сашко, він кілька років ходив голосувати за свою улюблену партію, він приймав участь у виборі нового президента та мера. Він відстоював свою ідею.

У 2020 році прийшов коронавірус і після цього стало майже неможливо брати участь у виборах. Він почав нехтувати своїми ідеями. Він вирішив, що здоров'я важливіше ніж вибори.

Але через деякий час йому розповіли друзі про додаток “SecureVote”. він був дуже радий, що навіть не зважаючи на епідемію, він зможе голосувати і приймати участь у виборах. Він почав користуватися додатком, він був у захваті, адже це дуже зручно, безпечно і не втрачається багато сил на підрахунок голосів.

1.3.2 Історія користувача програмного продукту

Громадянин завантажує додаток “SecureVote”

Громадянин реєструється або входить, якщо він вже зареєстрований

Громадянин голосує в онлайн форматі

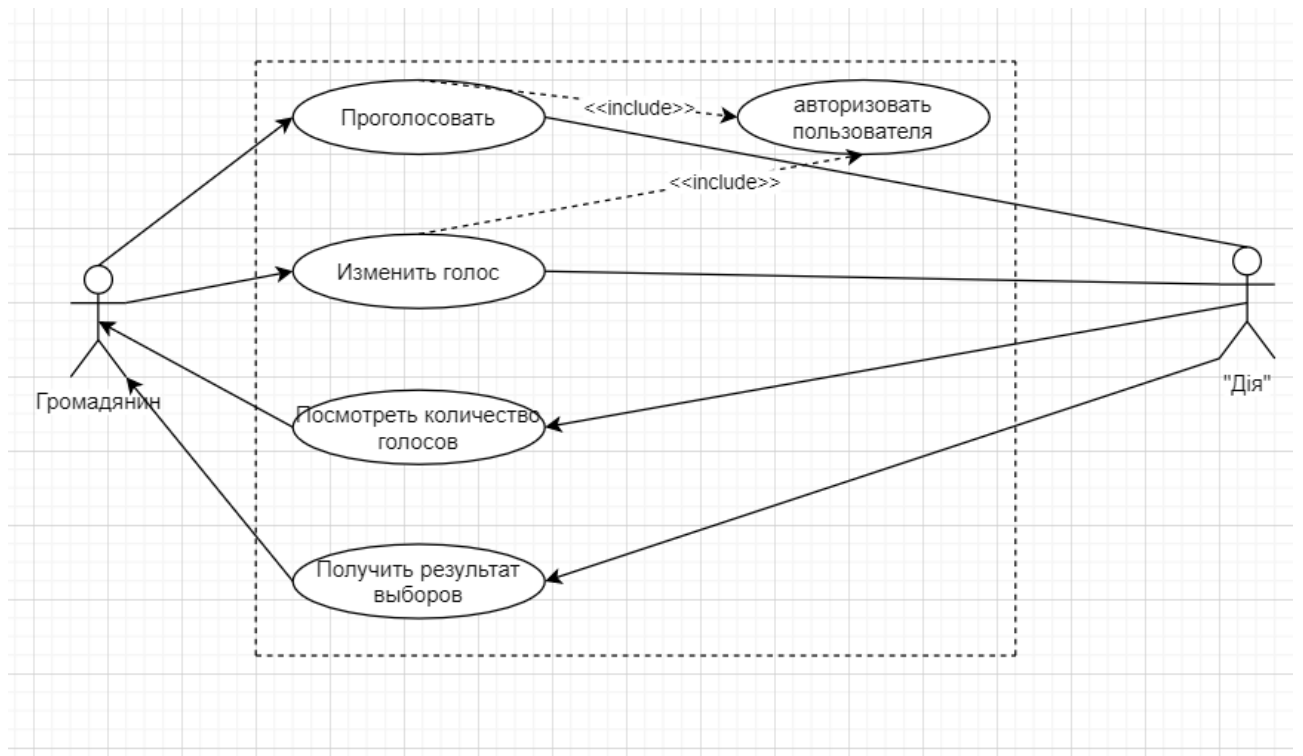
Програма рахує кількість голосів і видає відповідь користувачу

Громадянин має право змінити свій голос, поки не вийшов час

Держава підраховує голоси за допомогою програми

Громадянин отримує відповідь, у якості переможця виборів та т.д.

1.3.3 Діаграма прецедентів програмного продукту



1.3.4 Сценарії використання прецедентів програмного продукту

1) Назва прецеденту: Проголосувати

Передумови: Авторизуватися

Актори: Громадянин

Зацікавлені актори: Громадянин, “Дія”

Результат: Змогли проголосувати

Приклад основного успішного сценарію прецеденту

«Проголосувати»:

1. ПП виводить користувачу можливий вибір кандидату
2. Користувач передає ПП свій вибір
3. ПП зараховує вибір користувача

Приклад альтернативного сценарію для попереднього

прикладу основного успішного сценарію прецеденту

«Авторизувати користувача»:

2.1 Користувач не проголосував, сесія закінчується, не враховуючи голос користувача

2.a.1 ПП видає повідомлення про закінчення сесії і переходить до кроці 1 основного успішного сценарію.

- 2) Назва прецеденту: Змінити голос
Передумови: Авторизуватися
Актори: Громадянин
Зацікавлені актори: Громадянин, “Дія”
Результат: Змогли змінити

Приклад основного успішного сценарію прецеденту
«Змінити голос»:

1. ПП виводить користувачу голос, який він зробив
2. Користувач передає ПП свій новий голос за іншого кандидата
3. ПП зараховує новий вибір користувача

Приклад альтернативного сценарію для попереднього
прикладу основного успішного сценарію прецеденту
«Авторизувати користувача»:

- 2.1 Користувач голосує за того ж кандидату, сесія закінчується, не враховуючи новий голос користувача
- 2.a.1 ПП видає повідомлення про закінчення сесії і переходить до кроці 1 основного успішного сценарію.

- 3) Назва прецеденту: Переглянути кількість голосів
Передумови:
Актори: Громадянин
Зацікавлені актори: Громадянин, “Дія”
Результат: Змогли переглянути голоси

Приклад основного успішного сценарію прецеденту
«Переглянути кількість голосів»:

1. Користувач запитує можливість переглянути усі голоси
2. ПП надає користувачеві кількість голосів з урахуванням його голосу.

Приклад альтернативного сценарію для попереднього
прикладу основного успішного сценарію прецеденту
«Авторизувати користувача»:

- 2.1 ПП виявляє, що користувач не проголосував
- 2.a.1 ПП видає кількість голосів, не враховуючи голос користувача, і виводить повідомлення про можливість проголосувати, щоб побачити кількість голосів з урахуванням його голосу.

- 4) Назва прецеденту: Отримати результати виборів
Предумови:
Актори: Громадянин
Зацікавлені актори: Громадянин, “Дія”
Результат: Дізналися результат виборів

Приклад основного успішного сценарію прецеденту
«Отримати результати виборів»:

1. Користувач запитує можливість отримати результат виборів
2. ПП надає користувачеві кандидата, який переміг у виборах та кількість голосів 2-го та 3-го за голосами кандидатів.

Приклад альтернативного сценарію для попереднього прикладу основного успішного сценарію прецеденту
«Авторизувати користувача»:

- 2.1 ПП виявляє, вибори ще не закінчилися
- 2.a.1 ПП виводить повідомлення, що вибори ще в процесі, тому можливості побачити результат немає.

- 5) Назва прецеденту: Авторизуватися
Предумови:
Актори: Громадянин
Зацікавлені актори: Громадянин, “Дія”
Результат

Приклад основного успішного сценарію прецеденту
«Авторизувати користувача»:

1. ПП запитує у користувача параметри авторизації (Ідентифікатори / аутентифікатор)
2. Користувач передає ПП свої ІА-параметри
3. ПП надає користувачеві доступ до інших прецедентів ПП.

Приклад альтернативного сценарію для попереднього прикладу основного успішного сценарію прецеденту
«Авторизувати користувача»:

- 3.1 ПП виявляє, що користувач передав їй неправильні значення ІА-параметрів
 - 3.a.1 ПП видає повідомлення про помилку і переходить до кроці 1 основного успішного сценарію.
- Користувач зможе голосувати та змінювати свій голос

1.4 Функціональні вимоги до програмного продукту

1.4.1. Багаторівнева класифікація функціональних вимог

Ідентифікатор функції (назва)	Назва функції
FR1(Зареєструватись)	Реєстрація користувача
FR1.1	Створення запиту у користувача на отримання його параметрів реєстрації
FR1.2	Передача від користувача його параметрів реєстрації
FR1.3	Валідація параметрів
FR1.4	Створення акаунту
FR1.5	Повернення до сторінки авторизації
FR1.4.1	Створення запиту у користувача на надання валідних даних реєстрації

FR2(Авторизація)	Авторизація користувача
FR2.1	Створення запиту у користувача на отримання його параметрів ідентифікації та аутентифікації
FR2.2	Передача від користувача параметрів
FR2.3	Валідація параметрів
FR2.4	Авторизація
FR2.4.1	Створення запиту у користувача на отримання валідних параметрів ідентифікації та аутентифікації

FR3(Верифікація)	Верифікація користувача
FR3.1	Перевірка часу завершення терміну придатності токєну
FR3.2	Успішна верифікація
FR3.2.1	Оновлення токєну

FR4(Проголосувати)	Громадянин голосує
FR4.1	Показ інтерфейсу можливих кандидатів
FR4.2	Вибір кандидата користувачем
FR4.3	Додання голосу у БД

FR5(Змінити голос)	Громадянин змінює свій голос
FR5.1	Відображення користувачу інформації про його попередній голос
FR5.2	Вибір користувача нового кандидату
FR5.3	Оновлення інформації про голоси в БД в залежності від вибору

FR6(Отримати кількість голосів)	Отримання клієнтом кількості голосів за кожного кандидата
FR6.1	Запит користувача на отримання кількості голосів для кожного кандидата
FR6.2	Надсилання інформації про кожного кандидата

FR7(Отримати результат виборів)	Отримання клієнтом результату виборів
FR7.1	Запит користувача на отримання результату виборів

FR7.2	Надсилання інформації про переможця виборів
-------	---

1.4.2 Функціональний аналіз існуючих програмних продуктів

Ідентифікатор функції (назва)	ГОЛОС	Віче	Mobile-RADA
FR1.1	+	+	+
FR1.2	+	+	+
FR1.3	+	+	+
FR1.4	+	+	+
FR1.5	+	+	+
FR1.4.1	+	+	+
FR2.1	+	+	+
FR2.2	+	+	+
FR2.3	+	+	+
FR2.4	+	+	+
FR2.4.1	+	+	+

FR3.1	+-	+	+-
FR3.2	+-	+	+-
FR3.2.1	+-	+	+-
FR4.1	+	+	+

FR4.2	+	+	+
FR4.3	+	+	+
FR5.1	+	+	+
FR5.2	+	+	+
FR5.3	+	+	+

FR6.1	+	+	+
FR6.2	+	+	+
FR7.1	+	+	+
FR7.2	+	+	+

1.5 Нефункціональні вимоги до програмного продукту

1.5.1 Опис зовнішніх інтерфейсів

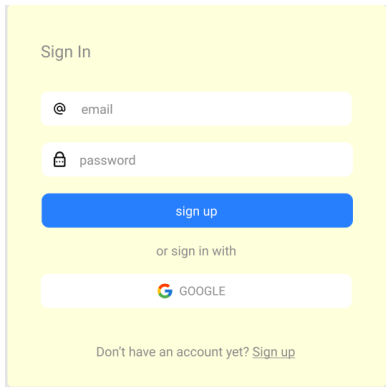
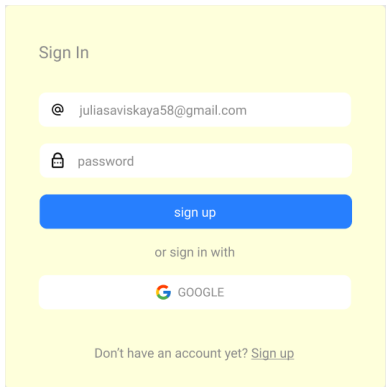
1.5.1.1 Опис інтерфейсів користувача

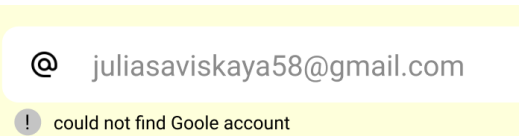
1.5.1.1.1 Опис INPUT-інтерфейсів користувача

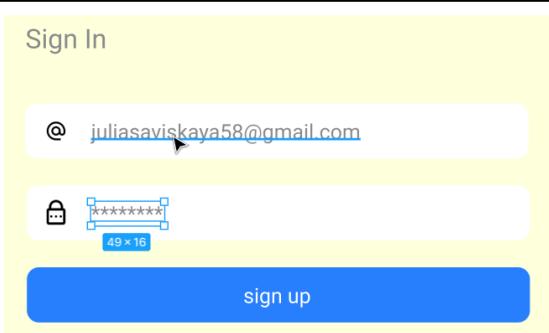
1)	сенсорний екран (Touchscreen, Touchpad, Multi-touch);
----	---

1.5.1.1.2 Опис OUTPUT-інтерфейсів користувача

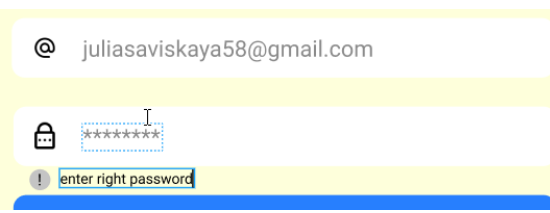
Ідентифікатор функції(назва)	Засіб OUTPUT - потоку	Особливості використання
------------------------------	-----------------------	--------------------------

FR1.1 Створення запиту у користувача на отримання його параметрів реєстрації)	Графічний інтерфейс	
FR1.2 Передача від користувача його параметрів реєстрації		
FR1.3 Валідація параметрів	-	-
FR1.4 Створення акаунту	-	-

FR1.5 Повернення до сторінки авторизації	Графічний інтерфейс	
--	---------------------	--

FR2.1 Створення запиту у користувача на отримання його параметрів ідентифікації та аутентифікації	Графічний інтерфейс	
---	---------------------	--

FR2.2 Передача від користувача параметрів		
FR2.3 Валідація параметрів	-	-
FR2.4 Авторизація	-	-

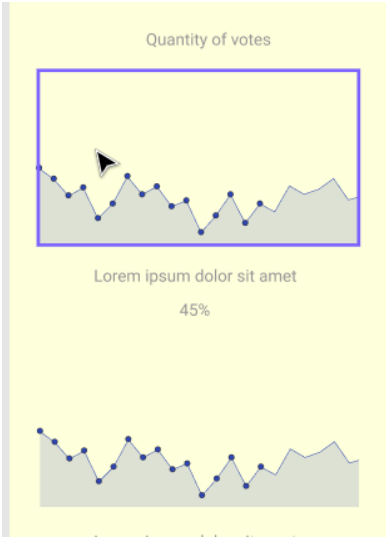
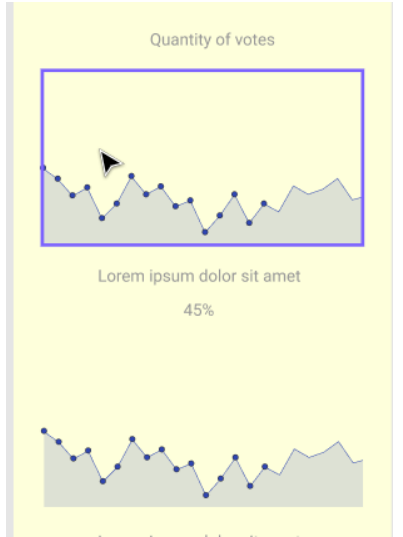
Ідентифікатор функції (назва)	Засіб OUTPUT - потоку	Особливості використання
FR2.4.1 Створення запиту у користувача на отримання валідних параметрів ідентифікації та аутентифікації	Графічний інтерфейс	

FR3.1	-	-
FR3.2	-	-
FR3.2.1	-	-

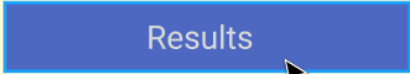

Засіб OUTPUT - потоку	Графічний інтерфейс		
Ідентифікатор функції (назва)	FR4.1 Показ інтерфейсу можливих кандидатів	FR4.2 Вибір кандидата користувачем	FR4.3 Додавання голосу у БД

Особливості використання			-
--------------------------	---	--	---

Засіб OUTPUT - потоку	Графічний інтерфейс		
Ідентифікатор функції (назва)	FR5.1 Відображення користувачу інформації про його попередній голос	FR5.2 Вибір користувача нового кандидату	FR5.3 Оновлення інформації про голоси в БД в залежності від вибору
Особливості використання			-

Засіб OUTPUT - поток	Графічний інтерфейс	
Ідентифікатор функції (назва)	FR6.1 Запит користувача на отримання кількості голосів для кожного кандидата	FR6.2 Надсилання інформації про кожного кандидата
Особливості використання		

Засіб OUTPUT - поток	Графічний інтерфейс	
Ідентифікатор функції (назва)	FR7.1 Запит користувача на отримання результату виборів	FR7.2 Надсилання інформації про переможця виборів

Особливості використання		<p>Results</p>  <p>Lorem ipsum dolor sit amet</p>
--------------------------	---	--

1.5.1.2 Опис інтерфейсу із зовнішніми пристроями

Бездротовий інтерфейс (Wi-Fi)

1.5.1.3 Опис програмних інтерфейсів

Операційні системи: Android(5.0.0+), IOS(7.1.2+).

Зовнішні ПП: IIS, AWS.

Бібліотеки: Bootstrap 5

1.5.1.4 Опис інтерфейсів передачі інформації

1.5.1.5 Опис атрибутів продуктивності

Ідентифікатор функції (назва)	Максимальний час реакції ПП на дії користувачів(с.)
FR1.1 Створення запиту у Користувача на отримання його параметрів реєстрації	1

FR1.2 Передача від користувача його параметрів реєстрації	1
FR1.3Валідація параметрів	2
FR1.4Створення акаунту	4
FR1.5Повернення до сторінки авторизації	1
FR1.4.1Створення запиту у користувача на надання валідних даних	1

FR2.1 Створення запиту у користувача на отримання його параметрів ідентифікації та аутентифікації	1
FR2.2 Передача від користувача його параметрів	2
FR2.3 Валідація параметрів	2
FR2.4 Авторизація	3
FR2.4.1Створення запиту у користувача на отримання валідних параметрів ідентифікації та аутентифікації	1

FR3.1 Перевірка часу завершення терміну придатності токена	1
FR3.2 Успішна верифікація	1

FR3.2.1 Продовження терміну придатності токenu (оновлення)	1
--	---

Ідентифікатор функції (назва)	Максимальний час реакції ПП на дії користувачів(с.)
FR4.1 Показ інтерфейсу можливих кандидатів	1
FR4.2 Вибір кандидата користувачем	1
FR4.3 Додання голосу у БД	3

FR5.1 Відображення користувачу інформації про його попередній голос	1
FR5.2 Вибір користувача нового кандидату	1
FR5.3 Оновлення інформації про голоси в БД в залежності від вибору	2

FR6.1 Запит користувача на отримання кількості голосів для кожного кандидата	1
FR6.2 Надсилання інформації про кожного кандидата	2

FR7.1 Запит користувача на отримання результату виборів	2
FR7.2 Надсилання інформації про переможця виборів	2

2 Планування процесу розробки програмного продукту

2.1 Планування ітерацій розробки програмного продукту

Ідентифікатор функції (назва)	Функціональні залежності	Вплив на досягнення мети, %	Пріоритет функції
FR1(Зареєструватись)		20%	M
FR1.1		3%	M
FR1.2	1.1	5%	M
FR1.3	1.2	3%	M
FR1.4	1.3	5%	M
FR1.5	1.4	2%	M
FR1.4.1	1.3	2%	M

FR2(Авторизація)		15%	M
FR2.1	1	5%	M
FR2.2	2.1	5%	M
FR2.3	2.2	5%	M
FR2.4	2.3	3%	M
FR2.4.1	2.3	2%	M

Ідентифікатор функції (назва)	Функціональні залежності	Вплив на досягнення мети, %	Пріоритет функції
FR3(Верифікація)		10%	M
FR3.1	2	5%	M
FR3.2	3.1	3%	M
FR3.2.1	3.1	2%	M

FR4(Проголосувати)		20%	M
FR4.1	2	5%	M
FR4.2	4.1	10%	M
FR4.3	4.2	5%	M

FR5(Змінити голос)		10%	C
FR5.1	4	3.5%	C
FR5.2	5	3.5%	C
FR5.3	5.3	3%	C

Ідентифікатор функції (назва)	Функціональні залежності	Вплив на досягнення мети, %	Пріоритет функції
FR6 (Отримати кількість голосів)		15%	C
FR6.1		7.5%	C
FR6.2	6.1	7.5%	C

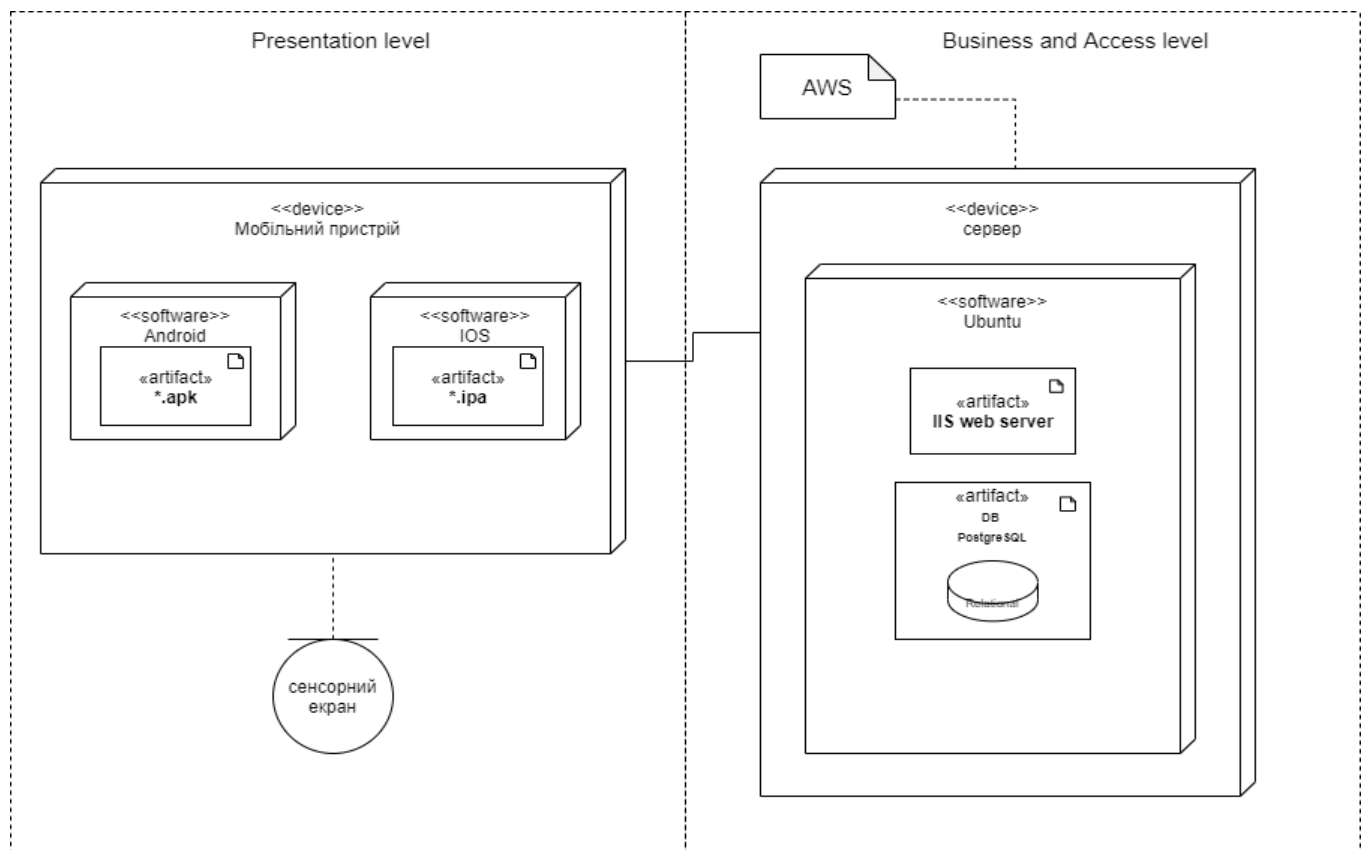
FR7(Отримати результат виборів)		10%	S
FR7.1		5%	S
FR7.2	7.1	5%	S

2.2 Концептуальний опис архітектури програмного продукту

“SecureVote” можна віднести до MobileApplication через те, що він

- працює на портативних пристроях
- не залежить від самого пристрою
- підтримує простий і легкий інтерфейс, що є досить пасує для використання на невеликому екрані
- програма залежить від можливості мережевого підключення

UML-діаграма розгортання ПП



2.3 План розробки програмного продукту

2.3.1 Оцінка трудомісткості розробки програмного продукту

Визначення не скоригованого показника UUCP

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

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

Визначення вагових показників прецедентів UC

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

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

UUCP = $45 + 10 = 55$

Визначення технічної складності проекту

Показник	Опис показника	Вага	ST
T1	Распределенная система	2	1
T2	Высокая производительность	1	1
T3	Работа конечных пользователей в режиме он-лайн	1	2
T4	Сложная обработка данных	-1	2
T5	Повторное использование кода	1	3
T6	Простота установки	0,5	2
T7	Простота использования	0,5	1

Показник	Опис показника	Вага	ST
T8	Переносимость	1	1
T9	Простота внесения изменений	1	2
T10	Параллелизм	1	1
T11	Специальные требования к безопасности	1	5
T12	Непосредственный доступ к системе со стороны внешних пользователей	1	1
T13	Специальные требования к обучению пользователей	1	1

$$TCF = 0,6 + (0,01 * (ST_i * Вага_i)) = 0,6 + (0,01 * 19,5) = 0,795$$

Визначення рівня кваліфікації розробників

Показник	Опис показника	Вага	ST
F1	Знакомство с технологией	1,5	3
F2	Опыт разработки приложений	0,5	5
F3	Опыт использования объектно-ориентированного подхода	1	3
F4	Наличие ведущего аналитика	0,5	3
F5	Мотивация	1	5
F6	Стабильность требований	2	3
F7	Частичная занятость	-1	1

F8	Сложные языки программирования	-1	1
----	--------------------------------	----	---

$$EF = 1,4 + (-0,03 * (SF_i * \text{Вар}_i)) = 1,4 + (-0,03 * 20,5) = 0,785$$

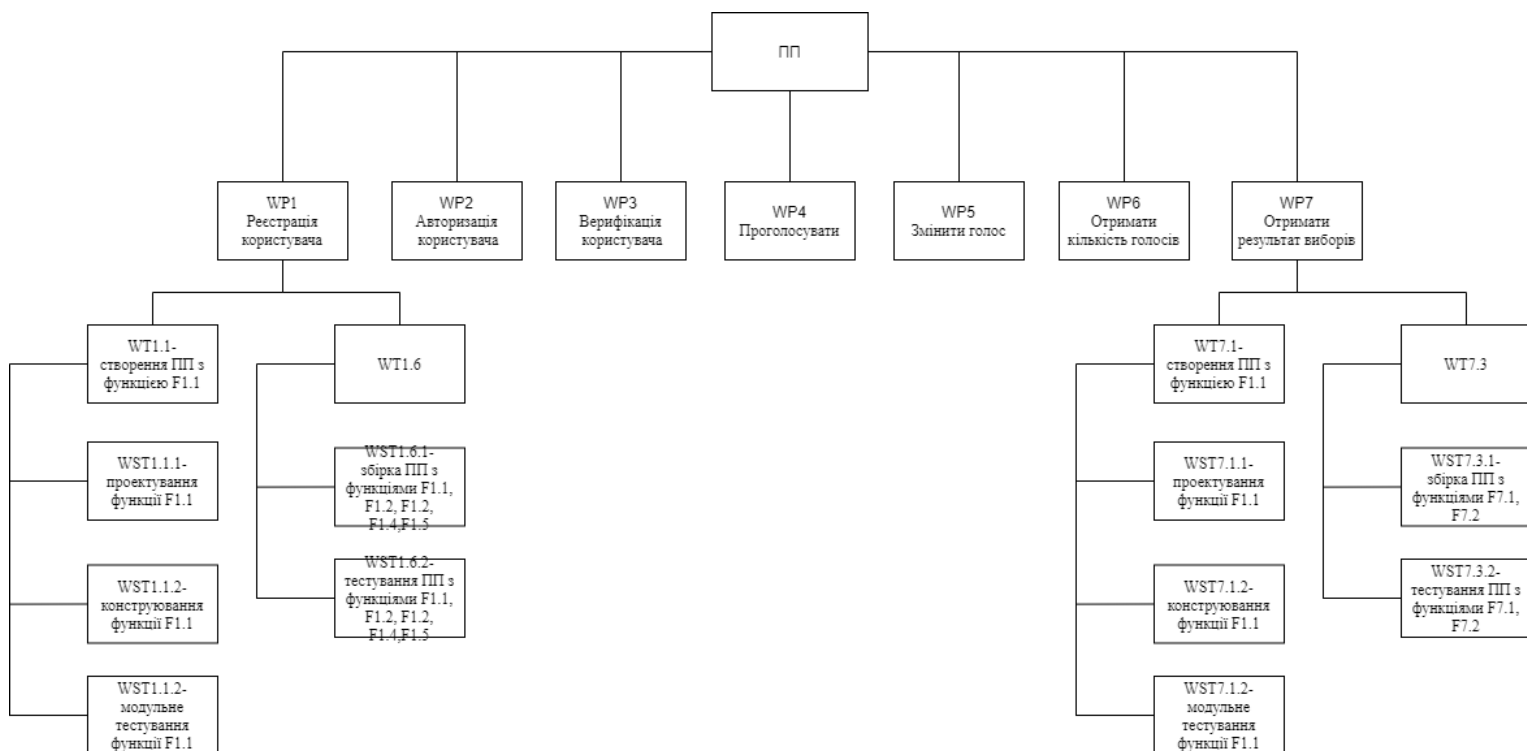
Остаточные значения UCP (Use Case Points)

$$UCP = UUCP * TCF * EF$$

$$UCP = 55 * 0,795 * 0,785 = 34,3$$

Розглянемо показники F1 - F8 і визначимо, скільки показників F1 - F6 мають значення менше 3 (0) і скільки показників F7 - F8 мають значення більше 3(0). Загальна кількість дорівнює 0, слід використовувати 20 люд.-год на одну UCP.

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

WST	Дата початку	Дні	Дата завершення	Виконавець
1.1	01.11.2021	1	02.11.2021	Савіцька Ю. О.
1.2	02.11.2021	1	03.11.2021	Савіцька Ю. О.
1.3	03.11.2021	2	05.11.2021	Савіцька Ю. О.
1.4	04.11.2021	1	05.11.2021	Савіцька Ю. О.
1.5	06.11.2021	1	07.11.2021	Савіцька Ю. О.
1.4.1	06.11.2021	1	07.11.2021	Савіцька Ю. О.
2.1	07.11.2021	1	08.11.2021	Савіцька Ю. О.
2.2	08.11.2021	1	09.11.2021	Савіцька Ю. О.
2.3	09.11.2021	1	10.11.2021	Савіцька Ю. О.
2.4	09.11.2021	2	11.11.2021	Савіцька Ю. О.
2.4.1	11.11.2021	1	12.11.2021	Савіцька Ю. О.
3.1	11.11.2021	2	13.11.2021	Савіцька Ю. О.
3.2	13.11.2021	1	14.11.2021	Савіцька Ю. О.
3.2.1	14.11.2021	1	15.11.2021	Савіцька Ю. О.
4.1	15.11.2021	1	16.11.2021	Савіцька Ю. О.
4.2	16.11.2021	2	18.11.2021	Савіцька Ю. О.
4.3	17.11.2021	1	18.11.2021	Савіцька Ю. О.
5.1	18.11.2021	3	21.11.2021	Савіцька Ю. О.
5.2	19.11.2021	1	20.11.2021	Савіцька Ю. О.
5.3	20.11.2021	1	21.11.2021	Савіцька Ю. О.
6.1	21.11.2021	1	22.11.2021	Савіцька Ю. О.
6.2	22.11.2021	1	23.11.2021	Савіцька Ю. О.
7.1	24.11.2021	4	28.11.2021	Савіцька Ю. О.
7.2	28.11.2021	3	31.11.2021	Савіцька Ю. О.

2.3.3.2 Діаграма Ганта

