**Software Requirements Specification for**

**<** **Програмна система для планування та моніторингу виконання особистих задач і досягнень. Серверна частина>**

**Version 1.1 approved**

**Prepared by <Надія Дашко>**

**<ПЗПІ-21-1>**

**<03.06.2025>**

**ЗМІСТ**

[Вступ 3](#_Toc200137771)

[1 Аналіз предметної галузі 4](#_Toc200137772)

[1.1 Актуальність теми 4](#_Toc200137773)

[1.2 Формування завдання з урахуванням потреб користувачів 6](#_Toc200137774)

[2 Формування вимог до програмної системи 8](#_Toc200137775)

[2.1 Постановка мети 8](#_Toc200137776)

[2.2 Загальний опис 8](#_Toc200137777)

[2.3 Загальні обмеження 11](#_Toc200137778)

[2.4 Припущення та залежності 12](#_Toc200137779)

# **ВСТУП**

У сучасному світі, що стрімко розвивається завдяки технологічному прогресу та соціальним трансформаціям, дедалі більше людей прагнуть до саморозвитку й особистісного зростання. Це дозволяє їм залишатися конкурентоспроможними, адаптуватися до викликів сьогодення та досягати гармонії між особистими цілями й темпом життя. У динамічному ритмі сучасності планування стало необхідним етапом на шляху до досягнення будь-якої мети. Водночас після її постановки вкрай важливо систематично відстежувати рівень власного прогресу, щоб підтримувати мотивацію та впевнено рухатись до результату.

Однак збереження стабільної мотивації, ефективне планування задач і контроль за власним розвитком є складним аналітичним процесом, що потребує якісних інструментів підтримки. У разі їх відсутності процес досягнення мети може бути суттєво ускладнений або навіть виявитися неможливим.

Основна мета програмної системи «Naviria» – надати користувачам інструмент для структурованого формування особистих цілей, розбиття їх на досяжні задачі, розвитку корисних звичок, відстеження досягнень і збереження мотивації у довгостроковій перспективі. Система поєднує логіку конструктора різних типів задач і підзадач, моніторинг прогресу у виконанні завдань, досягненні цілей, розвитку навичок, а також персоналізовану статистику, що дозволяє краще розуміти динаміку власного зростання.

З урахуванням важливості емоційної підтримки та залученості, «Naviria» впроваджує елементи гейміфікації, що включають систему рівнів користувачів, особисті досягнення, підтримку з боку інших учасників через соціальні механіки. Додатково вбудований персональний AI-асистент надає допомогу у формуванні задач, структуризації планів та відповідає на супутні запити користувача, що робить роботу із системою ще ефективнішою та зручнішою.

# **АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ**

* 1. Актуальність теми

Сучасний ритм життя, постійний потік інформації та необхідність виконання багатьох завдань одночасно створюють значне навантаження на користувачів, що часто призводить до неефективного використання часу та зниження продуктивності. Багато людей стикаються з проблемами, пов'язаними з організацією задач та відстеженням прогресу у їх виконанні. Зокрема, відсутність чіткої структури у розподілі завдань та недостатня видимість прогресу сприяють накопиченню завдань, що в свою чергу призводить до стресу та фрустрації. У результаті, користувачі не можуть ефективно реалізувати свої цілі, що знижує їхню мотивацію та продуктивність. Іншою проблемою є недостатня підтримка мотивації, що є особливо актуальним у разі довгострокових проєктів або складних задач. Без належного моніторингу прогресу та відсутності чітких орієнтирів для досягнення цілей, люди можуть втратити зацікавленість у процесі виконання завдань, що знижує їхню ефективність. Крім того, часто люди не мають інструментів для своєчасного виявлення та коригування помилок у процесі виконання, що призводить до затримок і зниження результативності.

Отже, основною проблемою є відсутність інтегрованих систем, які б не лише дозволяли організувати завдання, але й надавали б користувачеві інструменти для детального моніторингу прогресу, надавали персоналізовані рекомендації та активно підтримували мотивацію. Більшість існуючих рішень зосереджена лише на управлінні завданнями, не надаючи достатньої підтримки в процесі досягнення цілей або у впровадженні змін у звичку користувачів. Це і є основною проблемою, яку має вирішити розроблюваний додаток.

Для вирішення вищезгаданих проблем, розроблюваний додаток пропонує інтегровану систему управління особистими задачами, яка враховує не лише планування, а й детальний моніторинг виконання завдань, забезпечує мотиваційний супровід та дає можливість зворотного зв'язку в реальному часі. Однією з основних складових вирішення є гнучке управління задачами. Додаток дозволяє користувачам створювати, редагувати та класифікувати задачі за різними критеріями, такими як терміни, пріоритети та категорії. Це дозволяє уникнути хаосу у списках завдань та забезпечує зручну структуру для їх виконання. Користувач може чітко бачити весь обсяг роботи та розподіляти завдання по днях, що допомагає зменшити стрес і підвищити ефективність роботи.

Іншою важливою складовою є візуалізація прогресу. Додаток використовує графіки та діаграми для відображення прогресу виконання завдань, що дає користувачам можливість оцінити, скільки завдань виконано, а скільки ще залишилось. Це дозволяє користувачам зрозуміти, на якому етапі вони знаходяться, і створює відчуття контролю та завершеності. Візуалізація також допомагає знизити рівень невизначеності, який часто є однією з причин відкладання важливих справ.

Ще однією важливою складовою є мотивація користувачів через гейміфікацію. Додаток інтегрує елементи гейміфікації, такі як бали, рівні, досягнення та таблиці лідерів. Це стимулює користувачів до регулярного виконання завдань і дає їм відчуття прогресу навіть при виконанні дрібних задач. Користувачі отримують віртуальні нагороди та рівні, що додає елемент змагання і знижує ризик втоми від повторення однотипних завдань. Відчуття досягнення та успіху важливе для збереження мотивації, і ці елементи забезпечують додаткову підтримку в процесі виконання завдань.

Для користувачів, які шукають індивідуальний підхід до виконання задач, додаток пропонує AI-асистента, що надає персоналізовані рекомендації для організації робочого процесу. AI-асистент допомагає користувачам налаштувати пріоритети завдань, оптимізувати їхнє виконання та коригувати стратегії досягнення цілей. Він також підтримує користувачів у моменти, коли потрібна додаткова мотивація або поради щодо ефективного виконання задач.

Також важливою складовою є профілактика забування та пропуску термінів. Додаток автоматично генерує нагадування та сповіщення про наближення дедлайнів, що дає змогу користувачам своєчасно реагувати на будь-які зміни у планах. Це зменшує ймовірність виникнення затримок і сприяє своєчасному виконанню завдань. Завдяки цьому користувачі мають змогу підтримувати високий рівень організованості та відповідальності.

Розроблюваний додаток також включає можливість для соціальної взаємодії між користувачами. Це дає змогу додавати друзів, переглядати їхні досягнення та взаємодіяти у межах системи. Користувачі можуть брати участь у спільних викликах, обмінюватися досвідом і підтримувати один одного на шляху до досягнення цілей. Така соціальна складова додає елемент спільного розвитку і підтримки, що є додатковим стимулом до виконання завдань.

Загалом, розроблюваний додаток допомагає вирішити основні проблеми, пов'язані з неорганізованістю, відсутністю мотивації та недостатнім моніторингом прогресу, надаючи користувачам інструмент для ефективного управління завданнями, досягненнями та розвитку особистих навичок.

* 1. Формування завдання з урахуванням потреб користувачів

Потреби клієнтів на ринку управління задачами та особистою продуктивністю зростають, і вони мають високі вимоги до функціональності таких систем. Клієнти прагнуть знайти інструменти, які дозволяють ефективно організовувати та виконувати задачі, мінімізуючи час на їх планування і відстеження. Вони шукають рішення, які зможуть допомогти виконанню рутинних операцій, даючи змогу зосередитися на важливіших для себе моментах.

Однією з основних потреб є можливість чітко відслідковувати виконання задач, своєчасно реагувати на їх невиконання або прострочення, а також автоматичне нагадування про важливі етапи або дедлайни. Клієнти шукають системи, які дозволяють не тільки ставити задачі, але й бачити прогрес у їх виконанні через зручні візуалізації – графіки, діаграми, інтерактивні інтерфейси. Це дозволяє користувачам ефективно планувати свій час і бачити, які задачі вимагають найбільше уваги.

Крім того, важливою вимогою є простота у використанні та адаптивність системи. Клієнти хочуть мати інструмент, який не потребує значних витрат часу на освоєння і дозволяє зручно налаштувати процеси під свої індивідуальні потреби. Вони шукають інтерфейси, які є інтуїтивно зрозумілими, що дозволяє швидко отримувати доступ до всіх функцій без необхідності навчання.

Мотиваційна складова також стає важливою частиною ринку. Клієнти хочуть, щоб системи не лише допомагали у плануванні, але й активно підтримували мотивацію до виконання завдань. Інструменти гейміфікації, такі як бали, рівні, досягнення, є важливою складовою для багатьох користувачів, які шукають додаткові стимули для постійного досягнення своїх цілей.

Враховуючи ці потреби, розроблюваний додаток має забезпечити інтеграцію функцій для ефективного управління задачами, надання аналітики виконання, а також включати мотиваційні елементи для підтримки користувачів у процесі досягнення їхніх цілей.

# **ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ**

* 1. Постановка мети

Метою даної кваліфікаційної роботи є розробка клієнтської частини програмної системи для планування та моніторингу виконання особистих задач і досягнень.

Застосунок має забезпечити користувачів ефективними інструментами для організації їхніх завдань, відстеження прогресу і досягнення поставлених цілей. Система повинна бути адаптована до індивідуальних потреб користувачів, що мають різні рівні досвіду в управлінні часом та завданнями. Основна мета полягає в створенні інтуїтивно зрозумілого та зручного інтерфейсу, який дозволяє користувачам швидко та ефективно взаємодіяти з додатком. Функціональність системи повинна включати можливість створення, редагування та категоризації завдань, візуалізацію прогресу, систему нагадувань і сповіщень, а також мотиваційні елементи, такі як: бали, рівні та досягнення. Клієнтська частина повинна бути оптимізована для роботи на різних пристроях та платформах, забезпечуючи швидку реакцію і адаптацію до потреб користувачів [5].

2.2 Загальний опис

У межах даної кваліфікаційної роботи реалізується клієнтська частина веборієнтованої програмної системи, призначеної для організації, планування та контролю виконання особистих задач і досягнень користувача. Клієнтська частина взаємодіятиме з сервером, отримуючи необхідні дані та передаючи їх для відображення в інтерфейсі користувача.

Мають бути реалізовані наступні функції:

1. Реєстрація та авторизація користувачів:
2. Користувачі матимуть можливість зареєструватися та авторизуватися у системі. Всі дані користувачів, такі як ім’я, електронна пошта та паролі, будуть зберігатися на сервері.
3. Клієнтська частина взаємодіятиме з сервером для перевірки облікових даних користувачів, що дозволить здійснити авторизацію та доступ до персонального контенту.
4. Профіль користувача:
5. Клієнтська частина надасть користувачам можливість редагувати свої персональні дані, такі як ім’я, фото профілю тощо. Всі зміни зберігатимуться на сервері і будуть доступні для користувача після авторизації.
6. Користувачі також зможуть переглядати свої досягнення та статистику виконаних завдань, які будуть отримуватися з серверу в реальному часі для відображення на інтерфейсі.
7. Управління задачами:
8. Клієнтська частина дозволить користувачам створювати, редагувати та категоризувати завдання. Усі зміни будуть зберігатися на сервері.
9. Дані про завдання, їх категорії, терміни виконання будуть передаватися на сервер для збереження та подальшого відображення у клієнтському інтерфейсі.
10. Користувачі зможуть працювати з завданнями на клієнтській стороні, а дані синхронізуватимуться з сервером для збереження і подальшої обробки.
11. Аналітика та відстеження прогресу:
12. Прогрес користувачів щодо виконання завдань буде візуалізуватися через графіки та діаграми.
13. Клієнтська частина буде запитувати дані про виконані завдання з сервера та на основі цих даних генерувати відповідні візуалізації. Це дозволить користувачам відстежувати свої досягнення в реальному часі.
14. Гейміфікація:
15. Всі елементи гейміфікації, такі як бали, рівні, досягнення та таблиці лідерів, будуть зберігатися на сервері. Клієнтська частина буде взаємодіяти з сервером для отримання актуальних даних про досягнення користувачів та їх позицію в таблиці лідерів, а також для оновлення рівнів та нагород.
16. Push-сповіщення:
17. Сповіщення, що надходять користувачам про дедлайни, мотиваційні цитати та виклики від друзів, будуть генеруватися на сервері та доставлятися на клієнтську частину для відображення. Ці сповіщення будуть синхронізовані з даними на сервері, щоб забезпечити своєчасність і точність повідомлень.
18. Соціальна взаємодія:
19. Клієнтська частина дозволить користувачам переглядати акаунти інших, додавати та видаляти друзів. Всі ці дії будуть зберігатися на сервері, що дозволить синхронізувати соціальну активність користувачів між різними пристроями.

Усі зазначені функціональні можливості мають бути реалізовані у межах клієнтської частини програмного додатка, яка відіграє ключову роль у забезпеченні взаємодії між користувачем та системою. Саме клієнтська частина відповідає за побудову, відображення та оновлення інтерфейсу, через який здійснюється доступ до основних можливостей: перегляду профілю, керування задачами, відстеження прогресу, взаємодії з досягненнями та обробки повідомлень. Вона функціонуватиме як динамічний фронтенд, що отримує необхідні дані від серверної частини шляхом надсилання HTTP-запитів до API. Такий підхід дозволить забезпечити актуальність і цілісність даних, а також ефективну синхронізацію інформації між пристроями.

У результаті, незалежно від того, на якому пристрої здійснюється вхід – персональному комп’ютері, планшеті чи мобільному телефоні – користувач матиме постійний доступ до персоналізованого контенту, включаючи задачі, статистику, досягнення, повідомлення та інші інтерактивні елементи. Забезпечення стабільної та узгодженої роботи клієнтської частини гарантує, що зміни, внесені користувачем, будуть коректно збережені, а інтерфейс завжди відображатиме актуальний стан даних, що є критично важливим для підтримання довіри до системи та її ефективного використання.

2.3 Загальні обмеження

Клієнтська частина програмної системи реалізована на мові програмування JavaScript [6]. Використовується бібліотека React (v19) для створення компонентного інтерфейсу та Vite як інструменту для збірки й розробки [7]. Архітектура застосунку орієнтована на одно-сторінкову структуру (SPA), із використанням React Router для навігації між маршрутами. Для інтеграції з обліковими записами Google застосовується бібліотека @react-oauth/google, яка забезпечує базовий механізм автентифікації.

Застосунок розгортається у середовищі браузера і потребує стабільного інтернет-з'єднання для отримання та надсилання даних до серверної частини через HTTP API-запити. Уся бізнес-логіка, пов’язана зі збереженням, обробкою й аналітикою даних, виконується на сервері, тоді як клієнтська частина відповідає за візуалізацію, взаємодію з користувачем та ініціацію запитів до API.

Система підтримується на сучасних браузерах, зокрема:

* Google Chrome (версії не старші за дві актуальні);
* Mozilla Firefox;
* Microsoft Edge;
* Safari.

Мінімальні вимоги до клієнтського середовища:

* Операційна система: Windows 10/11, macOS 11+, Linux (із підтримкою сучасних браузерів);
* Процесор: 2-ядерний, не нижче 2.0 GHz;
* Оперативна пам’ять: від 4 GB;
* Екран: мінімальна роздільна здатність 1024×768;
* Інтернет-з’єднання: не менше 5 Мбіт/с.

Поза онлайном застосунок працювати не може, оскільки всі динамічні дані завантажуються з серверної частини, а локальне кешування реалізоване лише для технічних цілей. Відсутність повноцінної офлайн-підтримки зумовлює залежність функціоналу від постійного доступу до мережі. У випадку втрати з'єднання користувач позбавлений можливості взаємодіяти з основними розділами інтерфейсу, що вимагає передбачення відповідних повідомлень про помилки та механізмів повторної синхронізації після відновлення доступу до сервера.

2.4 Припущення та залежності

Під час розробки клієнтської частини програмної системи зроблено низку припущень, які впливають на логіку функціонування додатка та його взаємодію з серверною частиною.

Передбачається, що серверна частина уже реалізована або буде реалізована у відповідності до REST-архітектури з доступом до основних ресурсів (користувачі, завдання, досягнення, сповіщення тощо) через стандартизовані HTTP-запити. Клієнтська частина функціонує як інтерфейс для обробки та відображення цих даних, і без доступного API її повноцінна робота є неможливою.

Очікується, що сервер надає:

* стабільний та захищений API для аутентифікації користувача (зокрема підтримку OAuth 2.0 для Google-авторизації);
* доступ до бази даних із задачами, категоріями, статистикою, досягненнями;
* механізм обробки push-сповіщень і взаємодії між користувачами (друзі, виклики, таблиця лідерів).

Передбачається, що всі відповіді серверу надходять у форматі JSON, а авторизація користувачів реалізована через токенну модель (наприклад, з використанням JWT), яку клієнтська частина зберігає в локальному сховищі або оперативній пам’яті.

Крім того, існує залежність від зовнішніх бібліотек, зокрема:

* React – для побудови інтерфейсу користувача;
* React Router DOM – для маршрутизації;
* @react-oauth/google – для авторизації через Google;
* Axios або Fetch API – для взаємодії з сервером;
* Recharts або аналог – для візуалізації статистики;
* CSS або UI-бібліотеки – для реалізації адаптивного і зручного дизайну.

Також припускається, що користувач має сучасний браузер, який підтримує ES6+ та актуальні веб-API, включно з fetch, localStorage.

Таким чином, успішна робота клієнтської частини залежить від наявності стабільного інтернет-з'єднання, коректної роботи серверної частини, актуальності зовнішніх бібліотек та стандартів веббраузерів.