NIRVANA

Специфікація вимог до програмного забезпечення

Версія 1.0

Готвянський Кирило

Примаченко Микола

13.05.2025

# **Історія версій**

| **Дата** | **Опис** | **Автор** | **Коментар** |
| --- | --- | --- | --- |
| 13.05.2025 | Версія 1.0 | Примаченко Микола | Створення документу |
| 15.05.2025 | Версія 1.1 | Готвянський Кирило | Додавання нових пунктів |
|  |  |  |  |
|  |  |  |  |

# **Зміст**

[**Історія версій 2**](#_ds17lzrzxt2c)

[**Зміст 3**](#_26o7x7b4w8vm)

[**1 Вступ 4**](#_nmroqhtwf51m)

[1.1 Огляд продукту 4](#_57htb13z8zpr)

[1.2 Мета 4](#_9rxzdsze0xks)

[1.3 Межі 4](#_9fcicc94t2tx)

[1.4 Посилання 5](#_jlvpdvbn2a4d)

[1.5 Означення та абревіатура 5](#_9cdr88t3yh4y)

[**2 Загальний опис 6**](#_x8ql1d9t703c)

[2.1 Перспективи продукту 6](#_uygz42f9kvot)

[2.2 Функції продукту 7](#_fht1ku8xszp)

[2.3 Характеристика користувачів 8](#_wivdm7wa7gb4)

[2.4 Загальні обмеження 8](#_8a8r19aommva)

[2.5 Припущення та залежності 9](#_ca5sdpu3j1wr)

[**3 Конкретні вимоги 10**](#_k5bzdjprlcpf)

[3.1 Вимоги до зовнішніх інтерфейсів 10](#_w5it2rzdutwo)

[3.1.1 Інтерфейс користувача 10](#_6tuuopdhzge8)

[3.1.2 Інтерфейс користувача 12](#_idbtevz3al5z)

[3.1.3 Програмний інтерфейс 12](#_fcvn4shc84je)

[3.1.4 Комунікаційний протокол 13](#_hj08wf8mc1u9)

[3.1.5 Обмеження пам’яті 13](#_31mi92vv4bo)

[3.1.6 Операції 13](#_zltya5a3m8k)

[3.1.7 Функції продукту 13](#_opys32pizqwq)

[3.2 Вимоги до зовнішніх інтерфейсів 14](#_hrrln2dgyniv)

[3.3 Атрибути програмного продукту 15](#_r8r93vxg2v4z)

[3.3.1 Надійність 15](#_81cm6wekikct)

[3.3.3 Безпека 15](#_nt7eqvgl6c7t)

[3.3.4 Супроводжуваність 16](#_q5tbzzn4jedo)

[3.3.5 Переносимість 16](#_erltk69hj2f8)

[3.3.6 Продуктивність 16](#_oqe0qa1phihi)

[3.4 Вимоги бази даних 16](#_e46omfeoa0cu)

[3.4 Інші вимоги 17](#_d8ul64bncq5i)

[**4 Додаткові матеріали 18**](#_urxmrrn02nn6)

[4.1 Схеми баз даних 18](#_3c6j4m3o6suq)

[4.2 Діаграма прецедентів 18](#_gv5jjn26dzq8)

# 

# **1 Вступ**

## **1.1 Огляд продукту**

Nirvana — це програмна система для автоматизації управління персоналом та інвентарем та завданнями у готельному бізнесі. Продукт дозволяє адміністрації готелів ефективно керувати працівниками, створювати та контролювати виконання завдань, вести облік наявного інвентарю та моніторити витрати. Nirvana надає інструменти для онлайн взаємодії між працівниками різних підрозділів, організації робочих графіків, швидкого повідомлення про завдання та події, а також ведення статистики та аналітики діяльності готелю.

## **1.2 Мета**

Мета проекту: надати можливість адміністрації готелю та менеджерам ефективно керувати персоналом і обліком інвентарю через зручний веб-застосунок. Ідея проекту: забезпечити моніторинг виконання завдань та наявності інвентарю, що дозволить оптимізувати внутрішні процеси готелю.

Після впровадження даного продукту адміністрація зможе оперативно призначати завдання, контролювати виконання, вести облік витрат інвентарю та комунікувати з персоналом, а працівники — отримувати актуальні завдання та повідомлення у реальному часі. Це дозволить покращити організацію роботи, зменшити витрати часу на управлінські процеси та підвищити якість сервісу для гостей.

## **1.3 Межі**

Програмна система Nirvana має надавати можливості для управління персоналом та інвентарем у межах одного готелю. Продукт складається з веб-застосунку, через який здійснюється вся взаємодія користувачів — як адміністрації, так і персоналу.

Можливість створення облікового запису для працівника надається лише адміністратору. Керування обліковими записами співробітників у межах готелю здійснюється самостійно працівниками цього готелю.

Серверна частина має бути реалізована з використанням Clean Architecture та реалізовувати RESTful API. При розробці використовувався фреймворк ASP.NET Web API та мова програмування C# відповідно.

Клієнтська частина розробляється за допомогою Next.js у поєднанні з React. Контроль над процесом розробки, планування та відстеження виконання завдань здійснюється за веб-застосунком Linear.

## **1.4 Посилання**

1. Посилання на бекенд репозиторій -  
   <https://github.com/Gatvik/diploma-backend>.
2. Посилання на фронтенд репозиторій -  
   <https://github.com/kolyasikS/nirvana>.

## **1.5 Означення та абревіатура**

В документі використовуються наступні означення та абревіатури:

HTTP - HyperText Transfer Protocol

REST - REpresentational State Transfer

# **2 Загальний опис**

## **2.1 Перспективи продукту**

Програмна система Nirvana для управління персоналом та інвентарем у готельному бізнесі має значний потенціал у підвищенні ефективності внутрішніх бізнес-процесів та покращенні якості обслуговування клієнтів. Ось деякі ключові перспективи продукту:

* централізоване зберігання та управління даними: всі дані про працівників, завдання, повідомлення та інформацію про інвентар будуть зберігатися в єдиній системі. Це забезпечить зручний та безпечний доступ до актуальної інформації, а також полегшить пошук, перегляд та обробку даних;
* аналітика та звітність: система дозволятиме генерувати графіки та аналітичні звіти про робоче навантаження персоналу, виконання завдань, розподіл інвентарю та інші важливі показники. Це дасть можливість керівництву готелю оперативно контролювати роботу підрозділів, виявляти слабкі місця та приймати обґрунтовані управлінські рішення;
* контроль відповідності політикам та стандартам: система може бути налаштована відповідно до внутрішніх регламентів і стандартів готелю щодо роботи персоналу, розподілу змін, використання інвентарю та виконання завдань. Це дозволить уникнути порушень внутрішніх правил та забезпечує якість обслуговування на стабільно високому рівні;

Програмна система Nirvana буде актуальною для готельних комплексів, які прагнуть покращити організацію роботи персоналу, оптимізувати внутрішні процеси та забезпечити високий рівень сервісу для своїх клієнтів. Очікується, що система стане надійним і затребуваним інструментом для автоматизації кадрового та інвентарного управління.

## 

## **2.2 Функції продукту**

Програмна система Nirvana для управління персоналом та інвентарем у готельному бізнесі складатиметься з веб-додатку. Функціональні можливості буде поділено відповідно до ролей користувачів системи:

Адміністратор (веб-версія):

* створювати нові акаунти користувачів у системі;
* переглядати список усіх користувачів сервісу;
* редагувати профілі користувачів та змінювати їхні дані;
* змінювати ролі користувачів та їх доступи.

Менеджер (веб-версія):

* переглядати список усіх технічних працівників та покоївок;
* призначати їм завдання на певну дату;
* підтверджувати виконання завдань або відхиляти їх;
* переглядати статистику виконаних завдань;
* переглядати призначені завдання для конкретного працівника;
* створювати нових типів завдань;
* приймати або відхиляти виконання завдання.

Інвентарний менеджер (веб-версія):

* переглядати повний список інвентарю на складі;
* отримувати інформацію про нестачу ресурсів;
* поповнювати запаси інвентарю;
* переглядати історію змін кількості інвентарю.

Технічний працівник та покоївка (веб-версія):

* переглядати календар із завданнями;
* обирати дату та переглядати список своїх завдань на цей день;
* змінювати статус виконання завдань;
* при завершенні роботи оновлювати кількість використаного інвентарю на складі.

Усі ролі:

* змінювати та відновлювати пароль у разі необхідності.

## **2.3 Характеристика користувачів**

Ґрунтуючись на функціональних можливостях системи, виділено основні ролі користувачів, які взаємодіятимуть із системою Nirvana:

* адміністратор — користувач, який має можливість створювати нові акаунти користувачів у системі, переглядати список усіх користувачів сервісу, редагувати профілі користувачів, змінювати їхні дані та призначати ролі;
* менеджер — користувач, який має можливість переглядати список усіх технічних працівників та покоївок, призначати їм завдання, підтверджувати виконання завдань або відхиляти їх, створювати нови типи завдань, приймати та відхиляти готовність завдання, переглядати статистику виконаних завдань, а також переглядати завдання, призначені конкретним працівникам;
* інвентарний менеджер — користувач, який має можливість переглядати повний список інвентарю на складі, отримувати інформацію про нестачу ресурсів, поповнювати запаси та переглядати історію змін кількості інвентарю;
* технічний працівник та покоївка — користувачі, які виконують призначені їм завдання у готелі. Вони мають можливість переглядати календар із завданнями, обирати дату, переглядати список своїх завдань на цей день, змінювати статус виконання завдань і оновлювати кількість використаного інвентарю.

Усі ролі — користувачі системи мають можливість змінювати та відновлювати пароль у разі необхідності.

## **2.4 Загальні обмеження**

Нижче вказані всі обмеження, які будуть накладатися на технології, що будуть використовуватися при розробці продукту. Програмна система для автоматизації управління персоналом та інвентарем у готельному бізнесі буде складатися з двох частин: серверної та клієнтської. Вибраний підхід, заснований на чистій архітектурі (Clean Architecture), сприятиме швидшій розробці продукту, забезпечуючи гнучкість і зручність у керуванні. Для розгортання застосовується Docker із деплоєм на платформу Render. Взаємодія всередині системи реалізована через патерни Mediator і CQRS, що забезпечує чітке розділення команд і запитів. Як база даних використовується реляційна СУБД PostgreSQL. Для управління структурою даних застосовується Entity Framework, а міграції виконуються за допомогою окремого сервісу. Усі запити до бази даних обробляються через Entity Framework. Система оперує єдиною базою даних, яка містить усі необхідні дані. Сайт буде розроблятися за допомогою Next.js у поєднанні з React. Для генерації стилів буде використано технологію Tailwind CSS, для відображення статистики у вигляді діаграм - [React Charts](https://docs.google.com/document/d/1lhh-Rx-3Q7L3eoAFwKUuX2qKS9RI8CKE/edit#heading=h.jv06whr0fdsl).

## **2.5 Припущення та залежності**

П-1: платформою будуть користуватися готельна компанія, для спрощення процесу управління персоналом та інвентарем;

П-2: платформою будуть користуватися покоївки, менеджери та інші співробітники компаній;

П-3: працівники будуть вказувати витрати інвентарю після виконання завдання для подальшого поповнення інвентарним менеджером.

З-1: працездатність додатка буде залежати від завантаженості сайту та від швидкості роботи хостингу, на якому буде зберігається сайт;

З-2: залежність коректного відображення сайту від версії браузера та пристрою, на якому відкривається сайт;

З-3: залежність коректності введення інвентарю від введеної співробітником інформації.

# **3 Конкретні вимоги**

## **3.1 Вимоги до зовнішніх інтерфейсів**

### **3.1.1 Інтерфейс користувача**

При першому заході на сайт користувач має потрапляти на сторінку логіну, де потрібно ввести емейл та пароль для входу (див. рис. 3.1).

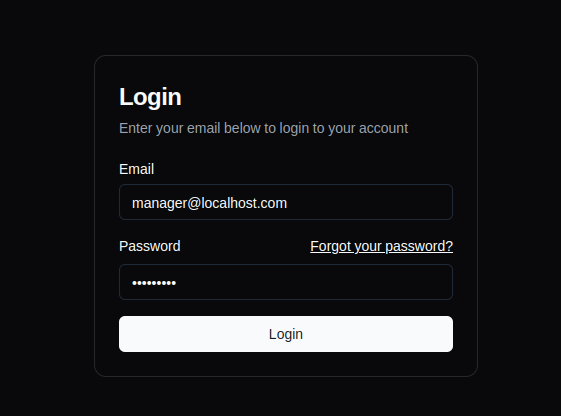


Рисунок 3.1 – Сторінка входу до системи

Після успішного входу, користувач попадає до своєго дашборду, де він має відповідний функціонал для своєї посади.

Для реєстрації користувача адміністрітор має заповнити форму створення аккаунта (див. рис. 3.2). Форма має містити наступні поля: ім’я, прізвище, адрес електронної пошти, посада, гендер.

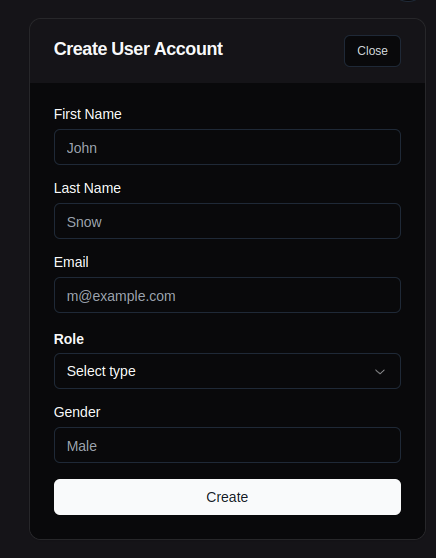


Рисунок 3.2 – Форма для створення нового аккаунта для користувача

Кожен користувач має окремий функціонал, який він може знайти на своєму дашборді. Наприклад дашборд менеджера (див. рис. 3.3) має інформацію про всіх працівників готелю, функціонал створення, тасок, перегляд графіку призначених завдань і т.д.

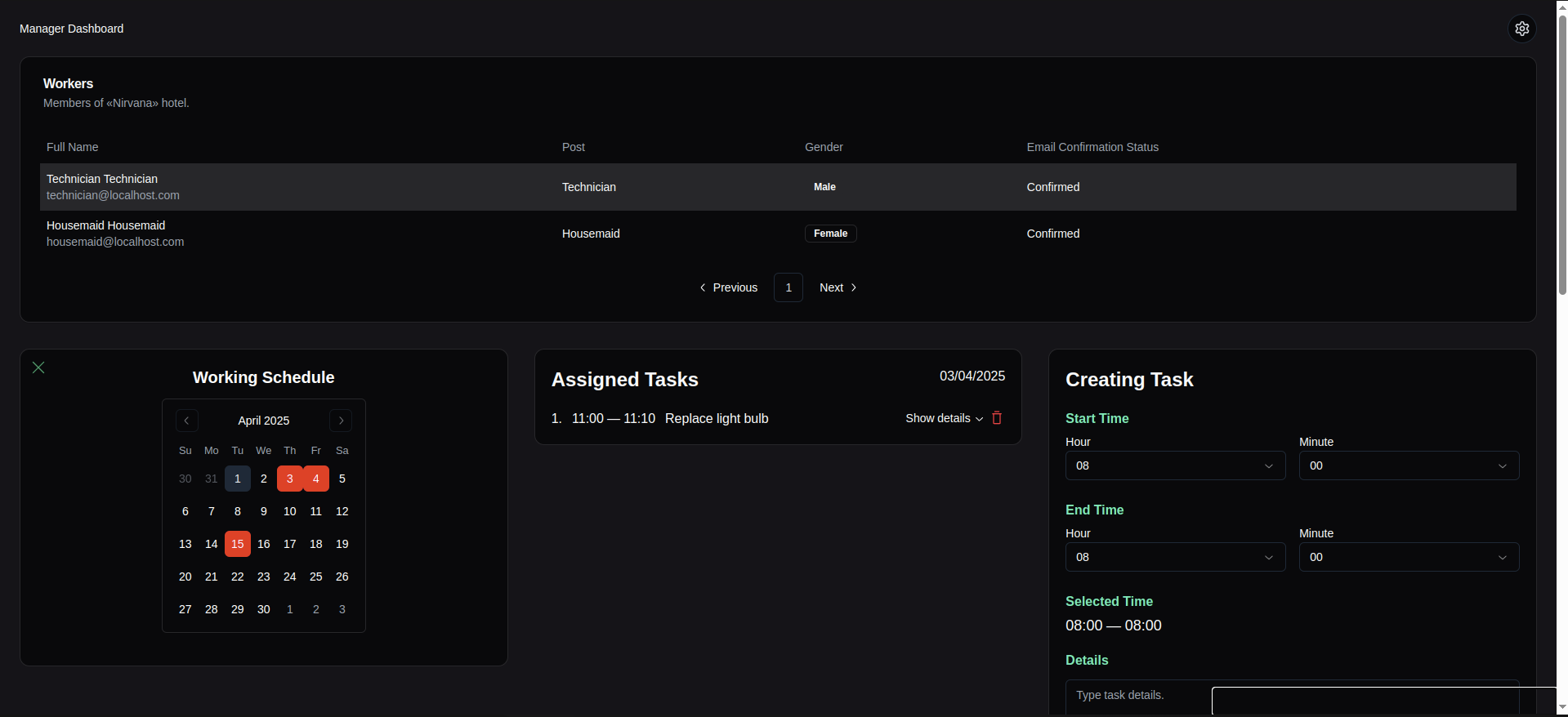


Рисунок 3.3 – Дашборд менеджера

Для кращої взаємодії додатка з користувачами, веб-додаток містить повідомлення (див. рис. 3.4), та модальні вікна (див. рис. 3.5) для підтвердження якоїсь критичної дії або заповнення окремих форм.

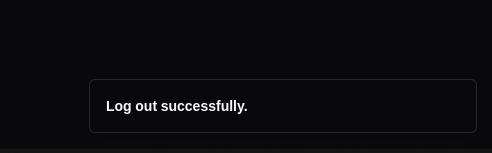


Рисунок 3.4 – Повідомлення для користувача

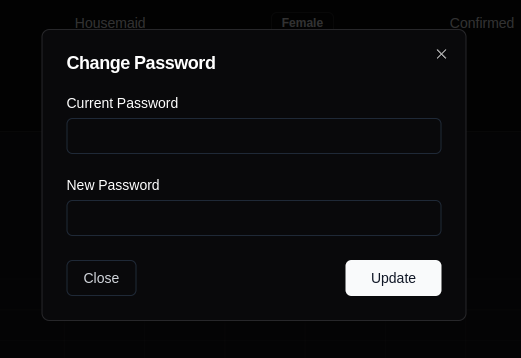


Рисунок 3.5 – Модальне вікно для змінення паролю

### **3.1.2 Інтерфейс користувача**

Оскільки телеграм бот і веб-сайт не мають призначеного обладнання, прямих апаратних інтерфейсів немає.

### **3.1.3 Програмний інтерфейс**

Програмне забезпечення для перегляду веб-сторінок має використовуватись користувачем для доступу до веб-системи. Він повинен підтримувати Javascript, HTML5, CSS3.

### **3.1.4 Комунікаційний протокол**

Взаємодія між компонентами системи реалізована через патерни Mediator і CQRS, що забезпечує ефективну комунікацію/ Серверна частина взаємодіє з вебклієнтом за допомогою протоколів HTTPS. Для зв’язку з базою даних PostgreSQL, розміщеною на платформі Neon, використовується стандартний протокол підключення, адаптований до хмарного середовища Neon з застосуванням TCP/IP.

### **3.1.5 Обмеження пам’яті**

Веб-сайт не може безпосередньо контролювати кількість доступної пам'яті на комп'ютері користувача. Прямого обмеження пам'яті комп'ютера для користування веб-сайтом немає.

### **3.1.6 Операції**

Для розробки веб-додатку використовується архітектурний стиль REST, що базується на принципах HTTP-протоколу. Основні операції, які можна виконувати з використанням REST, включають наступні:

* GET - для отримання ресурсу або його представлення;
* POST - для створення нового ресурсу;
* PUT - для оновлення існуючого ресурсу або створення нового, якщо він не існує;
* DELETE - для видалення ресурсу.

REST також використовує URL-шаблони для ідентифікації ресурсів та параметризації запитів.

### **3.1.7 Функції продукту**

ГФ-1: реєстрація – можливість адміністратора зареєстровувати нових користувачів;

ГФ-2: авторизація – можливість користувача зайти у існуючий обліковий запис;

ГФ-3: адміністрування – можливість адміністратора керувати обликовими записами користувачами;

ГФ-4: призначення завдань – можливість менеджера призначати завдання працівникам;

ГФ-5: управління інвентарем – можливість менеджера інвентарю поповнювати відсутні предмети;

ГФ-6: експорт даних – можливість менеджерам експортувати дані з доступних їм таблиць;

ГФ-7: керування завдань – можливість працівникам відмічати завдання виконаним з вказанням витрачених на цю таску предметів інвентарю;

ГФ-8: перевірка виконання завдання – можливість менеджеру приймати або відхиляти виконання завдання працівником;

ГФ-9: статистика – можливість переглядати статистику призначених завдань та змін інвентарю.

## **3.2 Вимоги до зовнішніх інтерфейсів**

Для доступу до системи необхідне стабільне інтернет-з’єднання. Оскільки дані будуть зберігатися віддалено на сервері, максимальний час відгуку встановлено до хвилини. Доступ до різноманітних функцій буде залежати від типу користувача. Цілісність та безпека зберігання даних буде забезпечуватися базою даних, безпека оплати забезпечується іншими відповідними сервісами.

Програмна система Nirvana розроблена з допомогою сучасних технологій таким чином, що додавання функцій у майбутніх версіях не стане проблемою. У наступних релізах планується додати мобільний додаток для співробітників з наступними функціями:

- нагадування: можливість створювати нагадування про наближення до завершення часу виконання завдання;

- прикріпити медіа файли до виконаного завдання для кращої оцінки менеджером.

Також планується створення телеграм бота для швидкого заказа нових предметів до інвентарю.

## **3.3 Атрибути програмного продукту**

### **3.3.1 Надійність**

* система повинна мати високий рівень надійності, користувач повинен отримувати тільки перевірені дані;
* в разі виникнення помилки, система не повинна припиняти свою роботу, користувач повинен побачити зрозуміле йому повідомлення з описом проблеми, що виникла (без коду помилки і інформації, яку зрозуміти зможе тільки розробник).

**3.3.2 Доступність**

* користуватися веб-додатком зможуть всі люди, що володіють англійською мовою;
* тільки авторизовані користувачі можуть отримати доступ до функціоналу додатку;
* тільки адміністратору від компанії доступні функції створення та редагування акаунтів працівників;
* тільки менеджер та інвентарний менеджер мають функції керування внутрішніми процесами.

### **3.3.3 Безпека**

* кожен обліковий запис в системі буде захищено паролем, який встановлює адміністратор, та який користувач має можливість змінити. Надійний пароль має складатися з більш ніж з 6 символів і обов'язково містити в собі принаймні одну цифру;
* увійти в обліковий запис можна тільки після введення правильних пароля та логіна;
* усі дані компанії мають зберігатися в окремій захищеній базі;
* сторінки мають бути доступні в залежності від дозволів ролі співробітника.

### **3.3.4 Супроводжуваність**

* система повинна бути розроблена таким чином, щоб була можливість додавання нових функцій в майбутньому;
* система повинна бути розділена на частини (серверна, клієнтська), щоб виправлення помилок займало якомога менше часу.

### **3.3.5 Переносимість**

* веб-додаток повинен запускатися в усіх сучасних браузерах незалежно від пристрою, який використовує користувач;
* веб-додаток повинен надавати доступ користувачам з будь якого регіону та у будь який час.

### **3.3.6 Продуктивність**

* час відгуку на перехід між вкладками веб-додатку повинен бути менше 2 секунд;
* після натискання кнопки "Create" співробітник має бути одразу доданий у список користувачів додатку, йому має прийти лист з паролем на пошту протягом 5 хвилин;
* після додавання нового завдання користувачу, йому повинен в ріал-таймі прийти повідомлення в додаток про нове завдання;
* працівник не має можливості переробити завдання після відхилення його менеджером.

## **3.4 Вимоги бази даних**

Для роботи з базою даних використовується система управління базами даних PostgreSQL, розміщена на платформі Neon. Система оперує єдиною базою даних, яка містить усі дані сервісу. Для забезпечення актуальності структури бази даних застосовується механізм міграцій, реалізований через Entity Framework Core. Логіка міграцій інкапсульована в окремій бібліотеці, яка використовується різними компонентами системи:

* WebAPI для виклику міграцій через Swagger UI;
* Консольний додаток для потреб локальної розробки.

Управління структурою бази даних здійснюється через Entity Framework, а всі запити до бази виконуються виключно за допомогою цього інструменту, без прямого мапінгу SQL-запитів на класи C#. Схеми бази даних наведено в додаткових матеріалах.

## **3.4 Інші вимоги**

Система не повинна дозволяти незареєстрованим користувачам отримувати доступ до функцій сервісу. Логін сторінка має бути окремою частиною, яка буде загальнодоступною, а веб-сайт з функціональністю доступний лише після успішного логіну в додаток. Інтерфейс повинен бути інтуїтивно зрозумілий і приємний у використанні для людей будь-яких вікових категорій.

# **4 Додаткові матеріали**

## **4.1 Схеми баз даних**

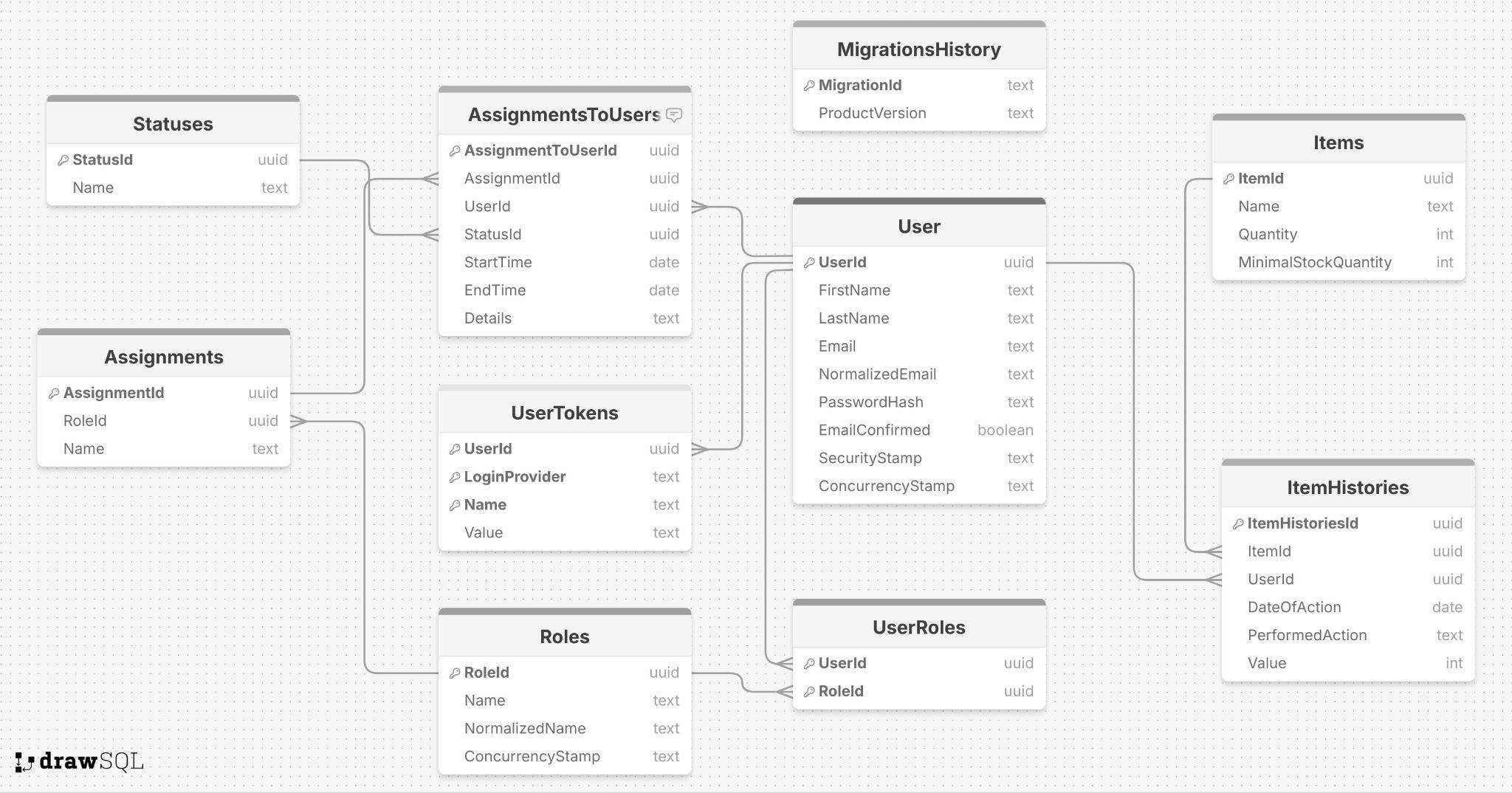


Рисунок 4.1 – Схема бази даних

## **4.2 Діаграма прецедентів**

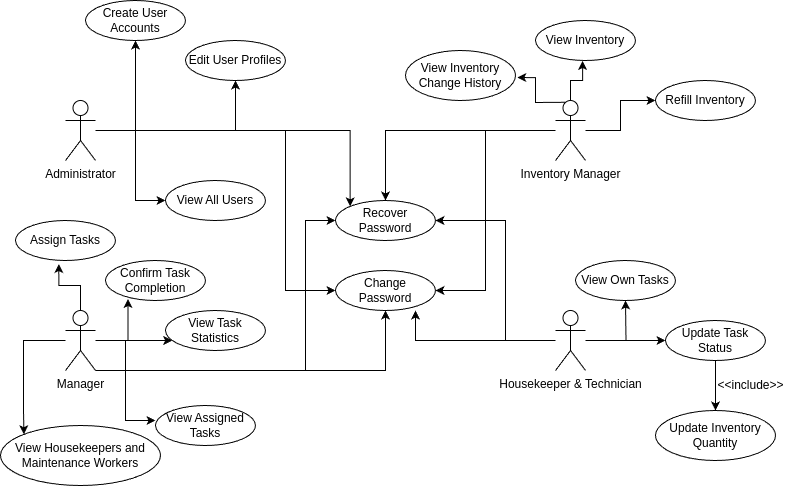


Рисунок 4.2 – Діаграма прецедентів розроблюваної системи