

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



Лабораторна робота №2

З дисципліни

“Проектування інформаційних систем”

Виконав :

Студент групи КН-314

Ляшеник Остап

Прийняв: Матьянов Д.І

Львів - 2023

MEDITRACK

REQUEST FOR PROPOSAL

1. Executive Summary

MediCorp є медичною технологічною компанією, що спеціалізується на розробці програмного забезпечення для автоматизації госпітальних операцій. Ми маємо більше 200 клієнтів у 5 країнах, наша місія — вдосконалювати медичну службу завдяки технологіям.

2. PROJECT OVERVIEW

Проект "MediTrack" передбачає створення системи для ефективного відстеження пацієнтів та медичного обладнання у госпіталі. Система забезпечить збільшення ефективності роботи персоналу та зниження витрат.

Основні причини реалізації:

- Покращення точності даних
- Ефективніше використання ресурсів
- Зменшення очікування пацієнтів

Ключові аспекти системи:

- **Monitoring:** Відстеження рухів та стану пацієнтів.
- **Tracking:** Моніторинг розташування медичного обладнання.
- **Security:** Множинні рівні доступу до інформації.
- **Integrity:** Інтеграція з існуючою системою електронної медичної картки.
- **Accessibility:** Доступ до системи з мобільних пристроїв

3. REQUIREMENT

Функціональні вимоги:

- **Моніторинг розміщення та стану пацієнтів:** Система повинна мати можливість зберігати інформацію, про те, у котрій палаті знаходиться пацієнт, зміна його стану впродовж перебування у лікарні
- **Управління Інвентарем:** Реалізувати модуль управління інвентарем для відстеження рівня запасів медичних засобів та обладнання, автоматичних замовлень та надавати сповіщення про нестачу товарів.
- **Управління персоналом:** Реалізувати можливість створення розкладу для персоналу.
- **Управління Даними Пацієнтів:** Покрім відстеження місця знаходження та стану пацієнтів, система повинна надавати можливість повноцінного управління даними пацієнтів, включаючи медичну історію, плани лікування та результати обстежень.
- **Створення початкового профілю пацієнта:** Інтеграція з існуючою системою електронної медичної картки.
- **Попередження та Сповіщення:** Система повинна бути здатною генерувати сповіщення в реальному часі для медичного персоналу щодо робочого плану та критичних випадків
- **Аналітика та Звітність:** Забезпечити інструменти аналітики та звітності для керівництва лікарні для відстеження ключових показників ефективності, використання ресурсів та результатів пацієнтів.
- **Страховання та Виставлення Рахунків:** Включити модуль страхування та виставлення рахунків для керування рахунками пацієнтів, страховими вимогами та фінансовими транзакціями, пов'язаними з медичними послугами.
- **Множинні рівні доступу до інформації:**

Є необхідність створення різних можливостей, для усіх рівнів користувачів системи, таких як:

Пацієнти : Доступ до власного профілю, де, попередньо приєднана електронна мед. Картка. Також повна інформація про історію відвідувань лікарні, де вказано лікаря, діагноз і та вартість лікування. Пацієнти також повинні мати доступ до оплати рахунку за лікування

Персонал, що займається лікуванням : Доступ до власного розкладу. Системою попереджень та сповіщень

надається основна керуюча інформація. Також необхідно надати доступ до інформації про пацієнтів, що є під наглядом.

Персонал логістики : Необхідно надати можливість керувати Системою Управління інвентарем, Управління персоналом. Створюють розклад для персоналу, що лікує та санітарного відділу.

Санітарний відділ : Надати інформацію, про власний розклад, та додаткову інформацію від відділу логістики.

Керівництво : Надати доступ до Аналітики та Звітності.

Відділ Адміністрування : Необхідно надати доступ до функцій надання доступів та Управління Даними Пацієнтів.

Нефункціональні вимоги:

- **Покращення точності даних:**

Система повинна зберігати медичні дані без втрати чи спотворення.

Дані мають оновлюватися в режимі реального часу.

Враховувати можливість автоматичної валідації та перевірки наявності помилок в медичних даних.

- **Ефективне використання ресурсів:**

Система має мінімізувати час на обробку та передачу даних.

Вимагається ефективне використання обчислювальних ресурсів, мережевих ресурсів та сховища даних.

Система повинна мати механізми автоматичного управління ресурсами для оптимізації витрат.

- **Зменшення очікування пацієнтів:**

Система повинна надавати можливість пацієнтам реєструватися на прийом та процедури заздалегідь через веб-портал або мобільний додаток.

Повинен бути механізм ефективного розподілу часових інтервалів для прийому та процедур, щоб уникнути зайвих очікувань.

- **Безпека:**

Для захисту конфіденційності медичних даних, система повинна використовувати сучасні методи шифрування при зберіганні та передачі даних.

Повинна бути реалізована автентифікація та авторизація користувачів для обмеження доступу до конфіденційних даних.

- **Інтеграція:**

Система повинна підтримувати стандарти обміну даними, що використовуються в існуючій системі електронної медичної картки.

- **Доступність:**

Система повинна бути вільно доступною з мобільних пристроїв через мобільні додатки або веб-версію.

Вимагається гарантування доступності системи в будь-який час, включаючи можливість роботи в автономному режимі при відсутності мережевого зв'язку.

4. ASR

- **Швидкодія:**

Система повинна бути здатною фіксувати рухи та стан пацієнтів у реальному часі, щоб надавати актуальну інформацію медичному персоналу.

Моніторинг розташування медичного обладнання також має бути миттєвим, щоб забезпечувати доступність обладнання у відповідний момент.

Швидкодія є критичною для попередження небезпеки та надання швидкої допомоги пацієнтам.

- **Безпека та конфіденційність:**

Система повинна використовувати сильне шифрування для зберігання та передачі медичних даних, щоб запобігти несанкціонованому доступу або перехопленню інформації.

Доступ до системи має бути строго контрольованим, з різними рівнями доступу для різних категорій користувачів (лікарі, медсестри, пацієнти тощо).

Аудиторський журнал повинен ретельно фіксувати всі дії користувачів для можливості аналізу та аудиту.

- **Гнучкість:**

Система повинна підтримувати стандарти обміну даними, такі як , для взаємодії з іншими медичними системами, лабораторіями та медичним обладнанням.

Інтеграція з існуючими системами електронної медичної картки дозволить надавати злагоджену медичну інформацію та покращити якість діагностики і лікування.

Забезпечення взаємодії з іншими системами допоможе уникнути подвійної роботи та забезпечити цілісність пацієнтських даних.

5. TECHNICAL DESIGN

Фронтенд:

- **Веб-додаток (React):** Інтерактивний додаток для користувачів, що включає модулі для моніторингу стану пацієнтів, управління ресурсами, персоналом та розкладами. Дозволяє пацієнтам переглядати інформацію про свій стан та історію лікування.
- **Мобільний додаток (React Native):** Додаток для мобільних пристроїв з аналогічним функціоналом, оптимізований для використання на смартфонах та планшетах.

Бекенд:

- **API Сервіси (Node.js + Express):** Набір RESTful API для обробки запитів від фронтенду, управління даними пацієнтів, ресурсами, персоналом, сповіщеннями та звітністю.

2. База даних:

- **SQL Database:** Централізована база даних для збереження всієї інформації, яка вимагається для функціонування системи. Включає таблиці для даних пацієнтів, ресурсів, персоналу, фінансових транзакцій тощо.

3. Зовнішні інтеграції:

- **Insurance & Billing Systems:** Системи для обробки страхових вимог та фінансових транзакцій. Інтеграція здійснюється через API для автоматичного обміну даними.
- **Electronic Medical Record System (EMR):** Система для збереження електронних медичних записів. Інтеграція дозволяє синхронізувати дані між EMR та нашою системою.

4. Безпека:

- **Аутентифікація та Авторизація:** Реалізація системи ідентифікації користувачів через безпечні методи, такі як OAuth 2.0.
- **Шифрування:** Використання протоколу HTTPS для шифрування всіх запитів та відповідей між клієнтом та сервером.
- **Рівні доступу:** Конфігурація різних рівнів доступу для користувачів залежно від їх ролей у системі.

5. Розгортання та Моніторинг:

Контейнеризація: Використання Docker для контейнеризації бекенд-сервісів забезпечує ізоляцію залежностей, легке масштабування та спрощення процесу розгортання.

- **Оркестрація контейнерів:** Застосування Kubernetes для управління контейнерами, забезпечуючи автоматизоване розгортання, масштабування та управління застосунками.
- **CI/CD:** Налаштування неперервної інтеграції та розгортання за допомогою інструментів, таких як Jenkins або GitLab CI, для автоматизації тестування та деплою.
- **Моніторинг та логування:** Використання Prometheus та Grafana для моніторингу стану системи та Alertmanager для налаштування сповіщень. ELK Stack (Elasticsearch, Logstash, Kibana) для агрегації та візуалізації логів.

6. Бекап та Відновлення:

- **Стратегії резервного копіювання:** Регулярне резервне копіювання баз даних та системи на цілому за допомогою рішень, таких як AWS RDS або рішень на базі Veeam.
- **План відновлення після збою:** Розробка та тестування плану дій на випадок збоїв, щоб мінімізувати час простою та забезпечити швидке відновлення служб.

7. Сумісність та Розширення:

- **API-First Підхід:** Розробка API з урахуванням можливості майбутнього розширення та інтеграції з іншими системами та сервісами.

- **Мікросервісна Архітектура:** Використання мікросервісної архітектури для забезпечення модульності та незалежності компонентів системи, що сприяє легшому оновленню та масштабуванню.

8. Юзабіліті та Доступність:

- **Інтерфейс користувача:** Оптимізація UX/UI для забезпечення інтуїтивно зрозумілого взаємодії з системою для всіх типів користувачів.
- **Доступність:** Забезпечення високої доступності системи через використання хмарних рішень та георозподілених центрів обробки даних.

6. TECNYCAL EXPLAINING

- **Node.js + Express для бекенд-сервісів:**
 - **Node.js** є популярним вибором для створення серверного програмного забезпечення через його неблокуючу модель вводу/виводу, яка добре підходить для I/O-інтенсивних застосунків, таких як веб-сервіси.
 - **Express** є легковажним фреймворком для веб-додатків на платформі Node.js, який полегшує створення API шляхом надання набору корисних утиліт для маршрутизації, обробки запитів та відповідей.
- **React для фронтенду:**
 - **React** є бібліотекою від Facebook для побудови інтерфейсів користувача. Його вибір зумовлений ефективністю при створенні динамічних веб-сторінок, легкістю створення інтерактивних UI та компонентно-орієнтованою архітектурою.
 - **React Native** для мобільних додатків дозволяє розробникам використовувати React та JavaScript для створення нативних мобільних додатків для iOS та Android, забезпечуючи високу продуктивність та відмінний користувацький досвід.
- **SQL Database:**
 - Бази даних SQL, такі як PostgreSQL або MySQL, є надійними рішеннями для зберігання структурованих даних, які часто використовуються в медичних системах. Вони забезпечують сильні гарантії щодо цілісності даних, підтримують складні запити та транзакції, що є важливим для обробки медичної інформації.
- **Інтеграція з зовнішніми системами:**
 - Інтеграція з **системами страхування та білінгу та електронними медичними записами (EMR)** є критичною для забезпечення безперервності медичних послуг та ефективного управління фінансами. Ці системи зазвичай надають API для інтеграції, що дозволяє бекенд-сервісам обмінюватися даними в реальному часі.

7. C4 MODEL

