

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки

«Затверджую»
Зав. кафедри теоретичної та
прикладної системотехніки
д.т.н., проф. С. І. Шматков
«__» _____ 2024 р.

Пояснювальна записка


до кваліфікаційної роботи
бакалавра

на тему: «Модель системи управління персоналом медичного закладу».

Захищено на засіданні
Атестаційної комісії № 44
протокол № __ від __.06.2024 р.
Оцінка ____ / ____
Голова Атестаційної комісії

(підпис) Скоб О. Ю.
(прізвище та ініціали)

Виконав:
студент 4 курсу, групи КУ– 41
Галузь знань: 15 – Автоматизація та
приладобудування
Спеціальність: 151 – «Автоматизація та
комп'ютерно-інтегровані технології»
Нестеренко Владислав Валентинович




(підпис)

Керівник: к.т.н., доц. ЗВО кафедри ТПС
Стрілець Вікторія Євгенівна



(підпис)

Рецензент: к.т.н., доц. ЗВО, в.о. зав.
кафедри теоретичної та прикладної
інформатики
Меняйлов Євген Сергійович



(підпис)

Харків – 2024

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел і чотирьох додатків. Загальний обсяг роботи складає 72 сторінок, із яких 48 сторінок основної частини з 25 рисунками, 7 таблицями, 11 найменуваннями списку використаних джерел та двома додатками.

Метою кваліфікаційної роботи є підвищення ефективності управління медичним персоналом за допомогою створення і впровадження відповідної моделі системи управління.

Об'єкт дослідження – процес управління персоналом медичного закладу.

Предмет дослідження – моделі та методи управління персоналом медичного закладу. Модель управління персоналом має спиратися на структуру персоналу медичного закладу, включаючи рівні кваліфікації, категорії працівників. Процес управління персоналом включає оптимізацію робочих процесів, розробку стандартів та процедур управління персоналом, зокрема планування графіків роботи, обліку робочого часу, ведення документації тощо.

Проблема, яка вирішується в кваліфікаційній роботі полягає в тому, щоб розробити комп'ютерну модель системи управління персоналом для медичного закладу з метою підвищити ефективність роботи медичного персоналу.

Область застосування – розробка додатку для постачання працівникам медичної корпорації, лікарні, медичного центру або системи охорони здоров'я загалом.

Ключові слова: система управління персоналом, медичний заклад, інформаційна система, Python, PostgreSQL, SQL, HR.

ABSTRACT

An explanatory note to the master's attestation work is created in the introduction, three sections, conclusions, a list of sources used and four additional substances.

The total volume of work is 72 pages, of which 48 pages of the main part with 25 figures, 7 tables, 11 names of the list of used sources and four additions. The purpose of the qualification work is to improve the efficiency of medical personnel management through the implementation and appropriate model of the management system.

The problem that is to develop a computer model of a personnel management system for medical work in order to promote the efficiency of work for medical personnel.

Scope – the development of supplements for the provision of health care services to doctors of a medical corporation, hospital, medical center or health care system.

Keywords: *personnel management system, medical institution, information system, Python, PostgreSQL, SQL, HR.*

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП	6
РОЗДІЛ 1. АНАЛІЗ СИСТЕМ УПРАВЛІННЯ ПЕРСОНАЛОМ МЕДИЧНОГО ЗАКЛАДУ	9
1.1. Аналіз організаційної структури медичного закладу	9
1.2 Проблеми управління персоналом в закладах охорони здоров'я....	13
1.3. Інформаційні системи управління персоналом	17
1.4 Аналіз існуючих систем управління медичними закладами.....	21
Висновки за розділом 1	28
РОЗДІЛ 2. ПРОЄКТУВАННЯ МОДЕЛІ УПРАВЛІННЯ ПЕРСОНАЛОМ МЕДИЧНОГО ЗАКЛАДУ.....	29
2.1 Модель управління персоналом	29
2.2 Структура моделі інформаційної системи медичного закладу	30
2.3 Проєктування бази даних інформаційної системи	33
Висновки за розділом 2	39
РОЗДІЛ 3. РОЗРОБКА ТА ТЕСТУВАННЯ МОДЕЛІ УПРАВЛІННЯ ПЕРСОНАЛОМ МЕДИЧНОГО ЗАКЛАДУ.....	40
3.1 Обґрунтування обраних технологій та засобів розробки інформаційної системи	40
3.2 Опис моделі системи управління персоналом медичного закладу .	42
Висновки за розділом 3	49
ВИСНОВКИ.....	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	52
ДОДАТКИ.....	54

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ

ВООЗ	–	Всесвітня Організація Охорони Здоров'я;
ЕСОЗ	–	електронна система охорони здоров'я;
HR	–	human resources;
МІС	–	медичні інформаційні системи;
ЦБД	–	централізована база даних;
БД	–	база даних;
SQL	–	structured query language;
РСКБД	–	реляційна система керування базами даних;
ОРСКБД	–	об'єктно-реляційна система керування базами даних.

ВСТУП

У сучасних умовах медичні заклади змінюють свої підходи до управління, використовуючи інноваційні методи в управлінні персоналом.

Головна роль в організації діяльності закладів охорони здоров'я належить медичному працівнику, а розробка ефективної системи управління персоналом є основним завданням менеджерів з управління персоналом у цій галузі. У сучасних умовах основне завдання управління персоналом закладів охорони здоров'я є комплексним процесом, при реалізації якого необхідно враховувати внутрішню специфіку лікувальної роботи закладу для вибору рішень відповідно до чинних кадрових ситуацій, забезпечити організацію роботи спеціалістів, організувати раціональне використання кадрового потенціалу організації, а також контролювати підвищення кваліфікації медичних працівників.

Перед сучасною системою охорони здоров'я України стоїть складне завдання адаптації до ринкових умов ведення бізнесу шляхом системної реструктуризації та вдосконалення механізмів її державного регулювання, а головне – впровадження комплексного розвитку та управління медичним персоналом. Аналіз досвіду європейських країн із розвиненою системою охорони здоров'я дозволяє зробити висновок про необхідність впровадження оптимальних механізмів розвитку кадрового потенціалу, спрямованих на навчання, підтримку, збереження та ефективне використання наявних ресурсів, що є одним з важливих напрямів управління персоналом установи. Крім того, стратегія реформування наголошує на необхідності міжгалузевої зв'язки (між секторами освіти, охорони здоров'я, праці та фінансів), практичне аналітичне обґрунтування та узгодження політичних рішень між секторами економіки.

Управління медичним закладом є складною сферою діяльності, що вміщує в собі такі ключові функції, як організація та координація, контроль,

фінансування, мотивація та представництво. У контексті охорони здоров'я, управління охоплює регулювання робочого часу медичних працівників, матеріальну стимуляцію, врахування соціально-психологічних мотивів, а також аналіз інформації для покращення роботи системи управління медичним закладом. Ресурсами управління є мета організації, людські, часові, матеріальні, владні, технічні та інформаційні ресурси. Однак головна складова – це керування людськими ресурсами. У цьому контексті виникає актуальна задача в розробці та впровадженні комп'ютерної системи управління персоналом для медичних закладів. Така система має не лише відтворити традиційні методи управління персоналом, але й інтегрувати сучасні інформаційні технології та аналітичні інструменти для підтримки стратегічних та операційних рішень в галузі управління персоналом.

Актуальність роботи. Сучасні медичні заклади мають впроваджувати нові рішення для автоматизації процесу управління. Тому створення моделі системи управління персоналом медичного закладу є актуальною задачею.

Система може значно полегшити процеси прийняття рішень, підвищити ефективність роботи персоналу, зменшити ризики та витрати. Використання сучасних інформаційних технологій дозволяє автоматизувати та оптимізувати багато аспектів управління підлеглими, що є перевагою для забезпечення якості медичної допомоги.

Модель управління має враховувати структуру персоналу медичного закладу, яка охоплює рівні кваліфікації, категорії працівників; оптимізація робочих процесів, розробка стандартів та процедур управління персоналом, планування графіків, облік робочого часу, ведення документації.

Метою дослідження є підвищення ефективності управління медичним персоналом за допомогою створення і впровадження відповідної моделі системи управління.

Об'єкт дослідження – процес управління персоналом медичного закладу.

Предмет дослідження – моделі та методи управління персоналом медичного закладу.

Методи дослідження: аналіз робочих процесів, аналіз нормативної бази, експертна оцінка даних, аналіз конкурентних систем. Дослідження та аналіз організаційної структури медичного закладу та медичних працівників.

Завдання дослідження:

- проаналізувати організаційну структуру медичного закладу та існуючих систем управління персоналом медичних закладів;
- створити модель управління персоналом медичного закладу;
- розробити програмну реалізацію моделі управління персоналом медичного закладу;
- провести тестування моделі системи управління персоналом медичного закладу.

РОЗДІЛ 1

АНАЛІЗ СИСТЕМ УПРАВЛІННЯ ПЕРСОНАЛОМ МЕДИЧНОГО ЗАКЛАДУ

1.1 Аналіз організаційної структури медичного закладу

У всіх медичних установах існує спеціалізована організаційна структура, яка є ключовою для надання медичного обслуговування населенню. Будь-яка структура має стандартний формат і вміщує в собі чітко визначені інструкції, функціональні обов'язки та положення для кожного підрозділу. Крім того, робота закладу охорони здоров'я регулюється статутом цього закладу, який охоплює організаційні аспекти та інші питання.

Організаційна діяльність лікарні керується наступними документами та принципами:

- Головний лікар визначає порядок внутрішньої організації та діяльності структурних підрозділів лікарні;
- Структурні підрозділи мають право оперативного використання майна лікарні у межах, визначених керівником;
- Лікарня працює згідно з чинним законодавством, приймаючи на роботу кваліфікованих працівників;
- Посадові обов'язки та інструкції для працівників лікарні затверджує головний лікар [1].

Додатково до вищезазначеного, існують специфічні особливості у закладах охорони здоров'я:

- Необхідність чіткої координації діяльності між різними працівниками (рис. 1.1) та підрозділами (рис. 1.2);
- Вимагається високий рівень спеціалізації працівників;
- Необхідний точний облік ресурсів, зокрема лікарських засобів;
- Робота вимагає терміновості та невідкладності.



Рисунок 1.1 – Координація роботи працівників



Рисунок 1.2 – Координація роботи підрозділів та служб

Різні види закладів охорони здоров'я мають свої особливості, які впливають на їх управління та організацію. Основні класифікаційні ознаки можна побачити на рис. 1.3.

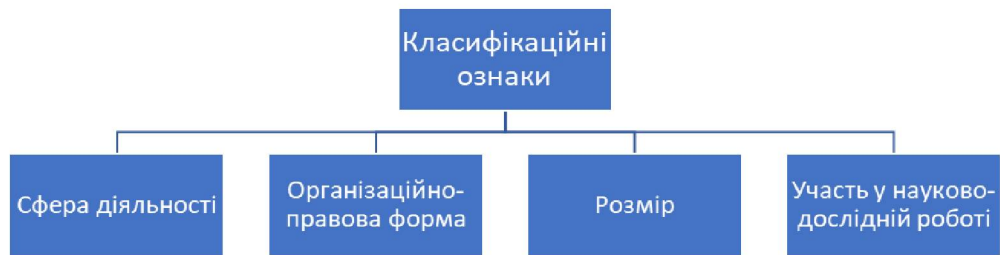


Рисунок 1.3 – Ознаки відмінності в різних видах закладів охорони здоров'я

Заклади охорони здоров'я можуть функціонувати як самостійні установи або бути частиною складу інших організацій, а також можуть бути підпорядковані іншим адміністративним одиницям. В Україні такі заклади можуть належати до державної (загальнодержавної та комунальної) або приватної (комерційної) власності [2]. Учасниками відносин у сфері господарювання виступають:

- персонал закладу охорони здоров'я, у тому числі управлінський, виробничий медичний, допоміжний персонал, бухгалтери, економісти, програмісти тощо;
- пацієнти як споживачі медичних послуг;
- пацієнти, їх законні представники, страхові компанії, фірми, які придбали медичні послуги для своїх працівників у ролі покупців;
- органи державної влади та органи місцевого самоврядування, які мають господарську компетенцію;
- громадяни, громадські та інші організації, що виступають у ролі засновників закладів охорони здоров'я або мають організаційно-господарські повноваження на основі відносин власності.

Для забезпечення медичної допомоги населенню існує різноманітна мережа лікувально-профілактичних установ, які можуть бути амбулаторно-поліклінічними або стаціонарними. Серед них є лікарняні заклади, поліклініки, диспансери, станції швидкої медичної допомоги, установи охорони материнства та дитинства, а також санаторні заклади. Лікарняні

установи, госпіталі та диспансери, забезпечують допомогу хворим у стаціонарних умовах. Для деяких пацієнтів, особливо з нескладним перебігом захворювання, медична допомога може надаватися в амбулаторній формі. У лікарнях здійснюється нагляд за хворими, виконуються процедури та забезпечується ліжковий режим. Сучасні медичні заклади мають обладнання для діагностики та лікування, а також у них створені умови для комфортного перебування пацієнтів. Клініка, що входить до складу медичного вищого навчального закладу або науково-дослідного інституту, спеціалізується на наданні спеціалізованої медичної допомоги, а також виконує навчальну та наукову діяльність.

Загальну структуру медичного закладу показано на рисунку 1.2.



Рисунок 1.4 – Типова структура медичного закладу

Хоча догляд за хворими є важливою складовою медичного обслуговування як в амбулаторних, так і в стаціонарних умовах, його значення особливо велике у стаціонарних відділеннях. У кожному відділенні встановлюються спеціальні місця для чергових медичних сестер, які наглядають за хворими та виконують призначення лікарів. Лікувально-

діагностичний підрозділ включає лікувальні та діагностичні кабінети, спеціалізовані палати, процедурні кабінети, палати для хворих, кабінети для медичного персоналу, зони відпочинку, їдальні та інші допоміжні приміщення. Ці зони є робочими місцями молодшого медичного персоналу у лікувально-профілактичних закладах.

1.2 Проблеми управління персоналом в закладах охорони здоров'я

З 6 листопада 2017 року в Україні увійшов в дію Закон України «Про внесення змін до деяких законодавчих актів України щодо удосконалення законодавства з питань діяльності закладів охорони здоров'я» від 6 квітня 2017 року № 2002, що запровадив автономізацію закладів охорони здоров'я [5].

У зв'язку з цим законом має відбутись автоматизація медичних закладів, що передбачає такі зміни у фінансово-господарській, організаційній та управлінській діяльності:

1. Зміна форм власності.

2. Збільшення владних повноважень керівників закладів охорони здоров'я, а саме щодо автономності в розпорядженні активами, фінансовим та кадровим фондом організації, встановлення управлінської та організаційної політики.

3. Свобода дій у визначенні форм і методів оплати праці робітників закладів охорони здоров'я.

4. Запровадження сучасних інструментів фінансового менеджменту.

5. Отримання закладами медичного обслуговування, що є підприємствами, права об'єднуватися з іншими медичними закладами, які також діють як підприємства, задля перерозподілу функцій між ними та спільного оптимізованого використання наявних ресурсів.

6. Надання закладам охорони здоров'я можливості наймати лікарів – фізичних осіб, що отримали відповідну ліцензію з дозволом здійснення господарської діяльності з медичної практики [4].

Завдяки таким змінам, як набуття автономії закладами охорони здоров'я в адміністративній діяльності та розширення спектру можливих фінансових джерел, виникає необхідність максимально ефективного використання кадрових ресурсів медичних установ. Це можливо за наявності управлінців із відповідною компетентністю, здатних перетворити персонал на конкурентну перевагу, що сприятиме досягненню поставлених цілей. Головна мета ВООЗ у реалізації кадрової політики полягає у підготовці кваліфікованого персоналу та забезпеченні ним закладів охорони здоров'я для виконання поставлених завдань. Умови належного кадрового забезпечення дозволяють стратегічному менеджменту активізувати роботу медичних фахівців та максимізувати їхню ефективність.

На рисунку 1.5 зображено механізм управління персоналом медичної організації.



Рисунк 1.5 – Механізм управління персоналом медичної організації

Політика управління кадрами – це основний напрямок роботи з персоналом, який відображає систему принципів, методів, правил та норм у взаємодії з кадрами та передбачає формування основних довгострокових напрямків організації кадрової роботи установи.

Зважаючи на всі особливості управління кадрами в організаціях охорони здоров'я, слід підкреслити вплив керівників медичного персоналу та рівень якості їхньої діяльності на ефективність або неефективність стратегій управління та розвитку галузі охорони здоров'я.

Етапи циклу управління персоналом закладів охорони здоров'я наведено на рис. 1.6.

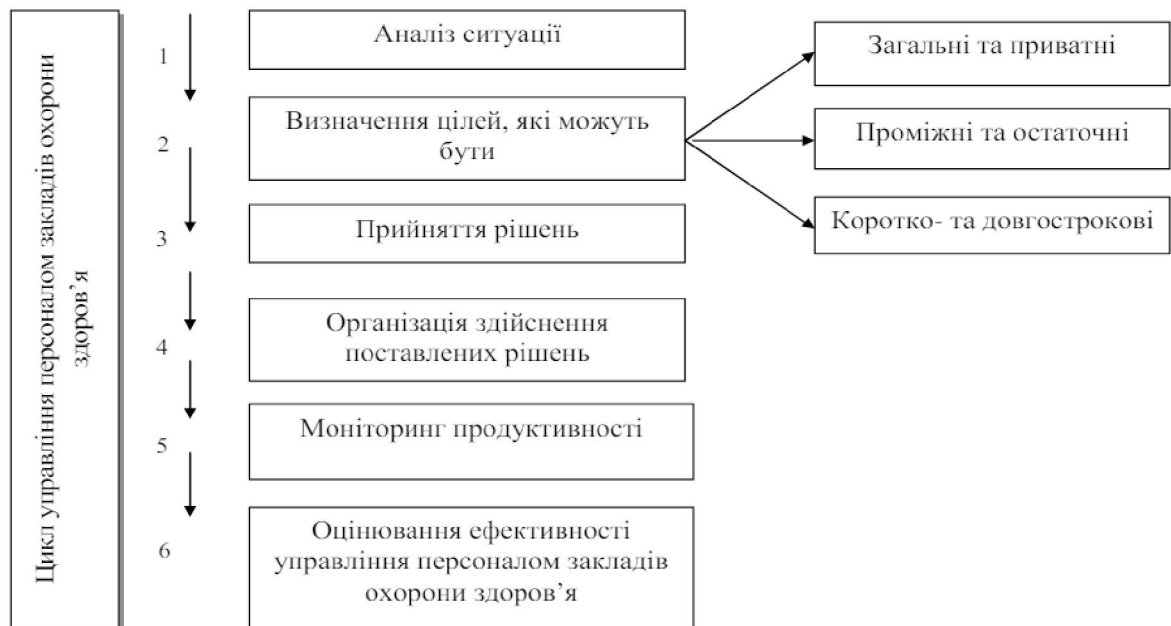


Рисунок 1.6 – Цикл управління персоналом закладів охорони здоров'я

Основна суть реформи системи охорони здоров'я полягає в зміні моделі фінансування, впровадженні ринкових механізмів через конкуренцію між медичними закладами у наданні якісніших послуг. Однією з важливих характеристик реформи є оплата за надані лікарем медичні послуги, при цьому бюджет закладу напряду залежить від кількості наданих послуг. Таким чином, вирішальну роль у фінансовому забезпеченні відіграватиме кількість кваліфікованих спеціалістів, які надають якісні послуги [3].

У процесі реформування сучасних закладів охорони здоров'я та забезпечення їх конкурентних переваг важливо не забувати про формування належного рівня організаційної культури. Це, насамперед, проявляється у

спілкуванні лікарів з колегами та пацієнтами, а також у чистоті, оформленні приміщень та кабінетів. Кожен пацієнт обиратиме той заклад, у якому крім кваліфікованої допомоги, створені гарні умови для надання послуг. Крім того, це сприятиме активізації як нематеріальних (комфортні умови праці, сприятливий мікроклімат), так і матеріальних (збільшення фінансування відповідно до більшої кількості наданих послуг) форм мотивації до праці.

Покращення управління персоналом закладів охорони здоров'я сприятиме впровадження ефективної системи планування кадрового забезпечення. Така система має відповідати потребам закладу та його фінансовим можливостям.

Зважаючи на напрями реформування системи охорони здоров'я, визначено інструменти, які можна використовувати для реалізації першочергових заходів з покращення управління персоналом медичних закладів. Серед цих заходів можна виділити збереження кадрового потенціалу, покращення забезпечення підприємств обладнанням, застосування енергоощадних технологій, ремонт приміщень для створення комфортних умов праці тощо.

В умовах зростання ролі інформаційних цифрових технологій для покращення управління персоналом та оперативного вирішення управлінських проблем сприятиме використання таких інструментів, як створення сучасної інформаційної системи та застосування проектного і процесного підходів в управлінні (рис. 1.7).

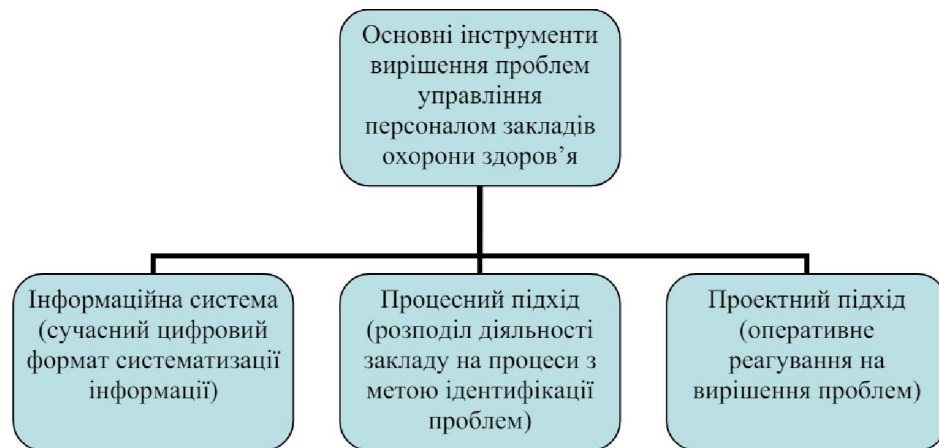


Рисунок 1.7 – Основні інструменти вирішення проблем управління персоналом закладів охорони здоров'я

Сучасне інформаційне забезпечення закладів охорони здоров'я передбачає створення цифрової інформаційної системи. Її впровадження та використання в конкретному закладі дозволить оптимізувати роботу лікарів, заощадити час на виконання професійних завдань і полегшити комунікації.

Подальша діджиталізація сприятиме створенню оперативних систем комунікацій з пацієнтами між медичними закладами, а також з органами публічного та державного управління у сфері охорони здоров'я. Використання процесного підходу передбачає поділ діяльності персоналу закладу на окремі процеси, що дозволить швидко виявляти проблеми в роботі та оперативно на них реагувати. Крім того, застосування процесного підходу в управлінні закладами охорони здоров'я допоможе уникнути дублювання деяких видів діяльності та подвійних завдань.

1.3 Інформаційні системи управління персоналом

Сучасна система управління персоналом (HR) – це комплекс технологій, які автоматизують та полегшують роботу з персоналом на всіх рівнях, від щоденного обліку даних до стратегічного прийняття рішень для розвитку компанії. У якісно побудованій системі ці технології підпорядковані загальній логіці управління персоналом відповідно до прийнятої стратегії, яка реалізується через бізнес-процеси та автоматизований електронний

документообіг. HR-система може ефективно взаємодіяти з фінансовими та виробничими системами, що використовують у своїх процесах дані про персонал [6].

Різні відділи управління персоналом використовують HR-системи для вирішення наступних завдань:

- Керівники підприємств застосовують їх як інструмент розвитку бізнесу, що підтримує розробку та реалізацію ефективної кадрової політики.
- Менеджери з персоналу та інспектори відділів кадрів використовують HR-системи для ефективного ведення кадрових процедур, документообігу, зберігання великих архівів даних та швидкого отримання будь-якої звітності по персоналу.
- Головні бухгалтери та бухгалтери з розрахунків заробітної плати користуються ними для підвищення швидкості та спрощення процесу оперативного розрахунку заробітної плати, формування бухгалтерських проводок, надання стандартної звітності та точного обліку витрат на оплату праці у собівартості продукції.
- Інспектори відділів праці та заробітної плати використовують HR-системи для виконання всіх функцій з планування та обліку праці, структури підприємства, штатного розкладу та витрат.
- Лінійні керівники застосовують їх для оцінки рівня розвитку підлеглих, планування кар'єри та заходів для розвитку співробітників, обґрунтування мотиваційних схем.
- Співробітники використовують HR-системи для аналізу відповідності займаній посаді та порівняння з вимогами до інших посад для планування свого кар'єрного зростання на підприємстві.

Системи управління персоналом виконують такі завдання:

- Управління організаційною структурою та штатним розписом на всіх рівнях, включаючи регіональні представництва та їх внутрішню структуру.
- Розрахунок заробітної плати.

- Кадровий облік, який включає ведення особових справ працівників, облік руху персоналу та підготовку необхідних паперових документів на основі даних системи.

- Облік робочого часу, включаючи ведення графіків співробітників та централізоване автоматичне формування табелів.

- Планування витрат на персонал.

- Планування кар'єри та відстеження просування співробітників по структурі.

- Робота з кадровим резервом.

- Підбір персоналу на відкриті вакансії.

- Навчання персоналу.

- Проведення атестацій.

Управління компетенціями, яке включає оцінку та підтримку рівня професійних і особистісних навичок через системи атестацій та безперервне навчання персоналу. Для реалізації цих завдань HR-системи забезпечені наступними функціональними можливостями:

- Автоматизований облік кадрового складу організації на основі структурованих особистих карток співробітників.

- Формування кадрових наказів з автоматичним внесенням інформації в особисті картки співробітників.

- Ведення архіву звільнених працівників.

- Робота зі штатним розкладом: перегляд і друк, отримання списку вакантних посад по підрозділах, отримання відомостей про заповнення штатів по кожному підрозділу і по підприємству в цілому.

- Формування звітів по співробітниках за довільний період з використанням відбору інформації та виведення у вигляді списків або статистичних таблиць.

- Підрахунок кількості лікарняних листів по підрозділах за будь-який період.

- Складання статистичної звітності з обліку кадрів: списків в електронному вигляді, отримання даних про розподіл співробітників за посадами штатного розкладу.
- Отримання зведеної таблиці про укомплектованість штатів, плинність і склад співробітників підприємства.
- Отримання тарифікаційних списків співробітників підприємства з проставленням кадрових параметрів.

Ефекти від впровадження систем управління персоналом можна класифікувати за різними аспектами (табл. 1.1).

Таблиця 1.1

Ефект від впровадження систем управління персоналом на підприємстві

Ефекти від впровадження		
організаційний	економічний	соціальний
Скорочення часу прийняття рішень на всіх рівнях управління підприємством.	Зниження витрат на управління персоналом.	Персональний облік пенсійних накопичень співробітників підприємства.
Підвищення якості кадрових рішень.	Підвищення продуктивності праці персоналу.	Ведення повної індивідуальною трудовою історії персоналу підприємства.
Оперативність підготовки звітності для органів державного управління, передбаченої законодавством	Оптимальне використання професійних якостей конкретного співробітника підприємства	Підготовка керівного резерву і просування по службі найбільш перспективних співробітників підприємства

1.4 Аналіз існуючих систем управління медичними закладами

При створенні моделі для управління персоналом медичного закладу потрібно дослідити аналогічні системи та зазначити їх переваги та недоліки.

Був проведений аналіз наступних декількох аналогів:

- eHealth;
- HELSI.ME
- Clinica WEB.

eHealth – це електронна система охорони здоров'я (ЕСОЗ), яка сприяє допомозі пацієнтам в отриманні якісних медичних послуг і лікарям – у їх наданні. Також ця система дозволяє Національному агентству забезпечення здоров'я контролювати ефективне використання державних коштів, виділених на охорону здоров'я в рамках програми медичних гарантій, та запобігати можливим зловживанням [8].

Для пацієнтів eHealth – це можливість використовувати електронні сервіси для здійснення своїх прав за програмою державних гарантій медичного обслуговування населення. Це означає, що вони можуть отримувати медичні послуги незалежно від реєстрації місця проживання, укладати декларації з лікарем, співпрацюючи через свій персональний кабінет пацієнта за допомогою електронного цифрового підпису. Пацієнти також можуть скористатися послугами телемедицини, особливо ті, що проживають у сільських районах, отримувати рецепти на ліки через електронну систему та скористатися іншими сервісами, які поступово будуть доступні.

Для закладів охорони здоров'я, лікарів та медичного персоналу eHealth – це можливість автоматизації медичних процесів, ведення обліку медичних послуг і управління медичною інформацією, впровадження електронного документообігу.

На рисунку 1.8 зображено ключові ролі сервісу eHealth.

КЛЮЧОВІ РОЛІ В СИСТЕМІ ЕЛЕКТРОННОГО ЗДОРОВ'Я



Рисунок 1.8 – Ключові ролі eHealth

eHealth вміщує в собі центральний компонент - централізовану базу даних (ЦБД), що відповідає за зберігання та обробку інформації, а також медичні інформаційні системи (МІС), які можуть бути обрані та встановлені лікарнями та поліклініками на ринку (рисунок 1.9).



Рисунок 1.9 – Структура роботи системи eHealth

Держава контролює центральний компонент системи eHealth. Адмініструванням системи займається державне підприємство «Електронне здоров'я». Адміністратор встановлює вимоги щодо надійності, безпеки та конфіденційності даних, які обмінюються між медичними інформаційними системами і центральним компонентом.

HELSI.ME є електронною медичною платформою, яка розроблена для використання пацієнтами, лікарями та як державними, так і приватними медичними закладами. Вона наділена великою кількістю можливостей, наведених у таблиці 1.2.

Таблиця 1.2

Можливості системи для різних ролей HELSI.ME [9]

Пацієнти	Лікарі	Медичні заклади
<p>Можливість легко знаходити та вибрати свого лікаря;</p> <p>Швидке онлайн-записування на прийом для себе та своєї родини;</p> <p>Доступ до власної електронної медичної картки (ЕМК);</p> <p>Миттєвий доступ до результатів аналізів та діагностики прямо у профілі пацієнта;</p> <p>Можливість перегляду призначень лікаря та плану лікування.</p>	<p>Зручне ведення медичної історії та електронних медичних карток пацієнтів;</p> <p>Швидке отримання результатів діагностики та аналізів;</p> <p>Просте використання клінічних протоколів;</p> <p>Зрозумілий інтерфейс для проведення прийому пацієнтів.</p>	<p>Інтегрована автоматизація діяльності медичного закладу;</p> <p>Можливість налаштування системи Helsi з урахуванням потреб конкретного медичного закладу;</p> <p>Функціонал для участі в реформі (включаючи закріплення пацієнтів за лікарями та реєстрацію декларацій з пацієнтами);</p>

Продовження таблиці 1.2

		Моніторинг та управлінська статистика для керівництва; Формування поточних звітів та статистики.
--	--	---

Clinica WEB – це інноваційна, зручна та надійна електронна медична система, розроблена для пацієнтів, лікарів, а також державних і приватних медичних установ. Це зручний інструмент для моніторингу та підвищення ефективності роботи, який також відзначається комфортним використанням та пропонує безліч корисних функцій [7].

Можливості системи наведені в таблиці 1.3.

Таблиця 1.3

Можливості системи для різних ролей Clinica WEB

Пацієнти	Лікарі	Медичні заклади
Зручний пошук та вибір лікаря; Швидкий онлайн-запис на прийом для себе та членів родини; Доступ до своєї електронної медичної картки (ЕМК); Миттєвий доступ до результатів аналізів та	Зручне ведення історії хвороби пацієнтів та ЕМК; Оперативне отримання результатів діагностики та аналізів; Легке використання клінічних протоколів; Зрозумілий кабінет для ведення прийому пацієнтів.	Комплексна автоматизація роботи медичного закладу; Можливість налаштування Clinica Web під потреби медичного закладу; Функціонал для участі в реформі (закріплення пацієнтів за лікарями, .

Продовження таблиці 1.3

діагностики у кабінеті пацієнта; Доступ до призначень лікаря та плану лікування.		реєстрація декларацій з пацієнтами); Моніторинг та управлінська статистика для керівників; Формування поточної звітності та статистики.
---	--	---

Розглянуті аналоги мають такі переваги як система управління охорони здоров'я:

1. Надають можливість зберігати і керувати медичною інформацією пацієнтів в електронному вигляді на централізованому сервері або в хмарних обчисленнях. Це робиться з метою забезпечення зручного та безпечного доступу до медичної інформації пацієнтів для лікарів та інших медичних працівників у будь-який час із будь-якого місця. Зберігання медичних даних у електронному вигляді дозволяє швидше здійснювати діагностику, надавати адекватне лікування та координувати медичний процес. Крім того, це забезпечує безпеку та конфіденційність інформації за рахунок використання захисту даних та обмеженого доступу до них.;

2. Можливість ведення електронних медичних записів, що дозволяє лікарям ефективно документувати інформацію про пацієнтів. Це спрощує процес документування та робить його менш часо- та працезатратним. Медичні записи можуть бути захищені від несанкціонованого доступу шляхом використання сучасних методів шифрування та аутентифікації. Це допомагає зберегти конфіденційність медичної інформації та запобігти можливим порушенням приватності пацієнтів. За допомогою електронних медичних записів можна здійснювати аналіз медичних даних та формувати статистичні

звіти, що допомагає в плануванні та організації надання медичних послуг, виявленні тенденцій та удосконаленні якості медичної допомоги. Отже, можливість ведення електронних медичних записів забезпечує ефективне документування медичної інформації, що сприяє покращенню якості та доступності медичної допомоги.

3. Забезпечують ефективність керування реєстрацією та призначеннями пацієнтів на прийоми до лікарів. Ці системи дозволяють пацієнтам записуватись на прийоми до лікарів онлайн або через контактний центр, що значно економить їхній час. Пацієнти можуть обирати зручний для них час та лікаря без необхідності особисто відвідувати медичний заклад або чекати у черзі. Системи управління реєстрацією та призначеннями пацієнтів зменшують ризик помилок у записі на прийоми, оскільки процес здійснюється автоматично та систематизовано. Це допомагає уникнути подвійних записів або пропусків пацієнтів;

4. Мають вбудовані засоби для аналізу даних, що дозволяє виявляти тенденції, проводити діагностику та приймати рішення на основі даних. Системи дозволяють вести моніторинг результативності медичних процедур та лікування, що дозволяє оцінювати ефективність заходів та вчасно вносити корективи для досягнення найкращих результатів.;

5. У них розширені заходи безпеки для захисту конфіденційної медичної інформації пацієнтів. Системи управління вимагають від користувачів проходження процедури аутентифікації, яка підтверджує їхню особу перед доступом до медичної інформації. Крім того, налаштовуються рівні доступу, що забезпечує обмеження прав користувачів лише до необхідної для їхньої ролі інформації.;

6. Інтеграція з іншими системами. Це сприяє збільшенню ефективності обробки та використання медичної інформації, оскільки дані з одного джерела можуть використовуватися в іншій системі без необхідності ручного введення або переносу. Інтеграція дозволяє різним медичним закладам обмінюватися

даними про пацієнтів та координувати їхнє лікування. Наприклад, це може бути корисно для координації догляду за пацієнтами у випадку направлення на консультації до спеціалізованих лікарів або госпіталізації до інших закладів.

За кожним впровадженням, яке охоплює не маленький підрозділ, а великі регіони, області чи країни, на жаль, спостерігаємо недоліки, над усуненням яких працює велика кількість спеціалістів, яким потрібний час на вдосконалення, апробацію, ресурси, кошти, а інколи це призводить до збоїв в системі роботи електронної платформи, незадоволення пацієнтів та лікарів. Великі системи можуть зіткнутися з технічними проблемами, такими як збої в програмному забезпеченні, недоступність серверів або недостатня пропускну здатність мережі. Ці проблеми можуть призвести до тимчасової недоступності системи або сповільнення роботи, що в свою чергу може вплинути на доступність медичних послуг та задоволення пацієнтів. Великі системи можуть мати складнощі з інтеграцією з іншими системами та стандартизацією обміну даними. Несумісність між різними платформами та форматами даних може ускладнювати обмін інформацією між різними медичними закладами та спеціалістами.

Розглянуті аналоги мають такі недоліки:

- Впровадження та налаштування електронних систем є дорогим і часомістким процесом, що стає перешкодою для деяких медичних закладів;
- Складний інтерфейс користувача та недостатньо інтуїтивно зрозумілі функції, що ускладнює їхнє використання медичним персоналом та споживачами;
- Технічні проблеми або випадки непередбачуваних обставин (вимкнення світла, відсутність мережі, інтернету) можуть призвести до втрати даних, що має негативні наслідки для надання медичної допомоги та безпеки пацієнтів;

– Впровадження нових систем вимагає навчання медичного персоналу та постійної підтримки, що може бути часо- та ресурсомістким.

Висновки за розділом 1

У першому розділі було розглянуто функціональну організаційну структуру медичного закладу та структурні підрозділи. Визначено ключові підрозділи та їх функціональні обов'язки, що дозволяє зрозуміти розподіл ролей і відповідальності у закладі. Виявлено основні проблеми, що виникають у процесі управління персоналом, такі як нестача кваліфікованих кадрів та висока плинність персоналу. Підкреслено важливість автоматизації процесів управління персоналом для підвищення ефективності та зменшення адміністративного навантаження на керівників. Зазначено аналоги системи управління лікарнями, здійснено порівняльний аналіз трьох електронних платформ з надання послуг пацієнтам.

РОЗДІЛ 2

ПРОЄКТУВАННЯ МОДЕЛІ УПРАВЛІННЯ ПЕРСОНАЛОМ МЕДИЧНОГО ЗАКЛАДУ

У цьому розділі роботи розглядаються основні принципи функціонування та структура моделі управління персоналом медичного закладу та її впровадження у медичну інформаційну систему.

2.1 Модель управління персоналом

Проаналізувавши поставлену задачу було вирішено спроектувати модель з боку менеджера системи, яка дозволить використовувати основні функції для управління персоналом.

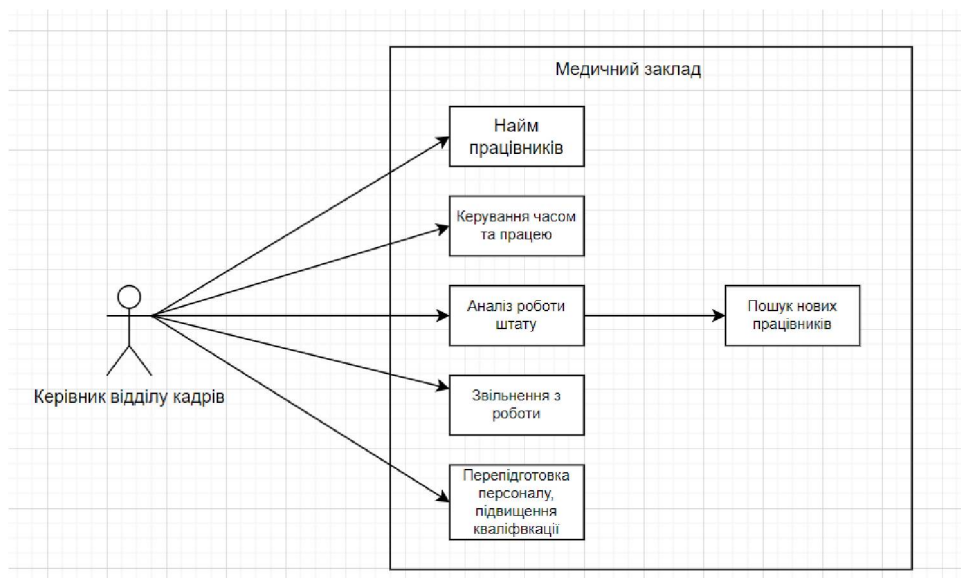


Рисунок 2.1 – Структура моделі управління персоналом

На діаграмі (рис.2.1) зображено керівника відділу кадрів, який виконує такі завдання:

- Наймання працівників: відбір та прийом на роботу нових співробітників. Створення договору та влаштування на роботу.

- Керування часом та працею: планування робочого часу, відстеження відпрацьованих годин та управління завданнями, щоб забезпечити ефективне використання ресурсів;
- Аналіз роботи штату: аналіз продуктивності та ефективності роботи персоналу, їх завантаженість, виявлення проблемних зон та розробка стратегій для покращення;
- Звільнення з роботи: збір та аналіз даних про роботу персоналу, підготовка звітів для керівництва, створення договору на звільнення;
- Перепідготовка персоналу, покращення кваліфікації: збір даних про професійний рівень та повідомлення про потребу підвищення професійного рівня співробітників та розвитку їхніх навичок, статус персоналу;
- Пошук нових працівників: пошук та залучення потенційних кандидатів на роботу в медичний заклад (при великій завантаженості).

2.2 Структура моделі інформаційної системи медичного закладу

Розглянемо вимоги до системи загалом:

1. Система має бути простою та зрозумілою у використанні, а саме має бути зручною пошук інформації, зрозумілий текст, зручний інтерфейс;
2. Програмне забезпечення має враховувати процеси управління персоналом;
3. Повинна існувати можливість швидкого переходу пунктами меню та швидкого доступу до необхідної інформації;
4. Система повинна бути стійкою до відмови і надійною.

Проаналізувавши аналогічні моделі (підрозділ 1.4) можна виділити декілька основних принципів. По-перше, ефективна інтеграція з базою даних медичного закладу, що дозволяє швидко та точно збирати та аналізувати інформацію про персонал та пацієнтів. Модель також повинна мати зручний інтерфейс, який спрощує роботу у використанні для керівників та співробітників медичного закладу. Важливою є також можливість розподілу завдань, контролю та моніторингу, що сприяє підвищенню продуктивності та

оптимізації робочих процесів. Крім того, в потрібно врахувати збереження конфіденційності медичної інформації. Також постійне оновлення та підтримка програмного забезпечення гарантує його надійність та відповідність актуальним вимогам та потребам медичного закладу.

У системі управління лікарнею планується, що кожен користувач має певні стратегії роботи. Вони мають свої обов'язки, і відповідно до обов'язків вони виконують кілька функцій. Якщо будь-який користувач не дотримується своєї стратегії роботи, вся програма може вийти з ладу. На рис. 2.2 описано структуру інформаційної системи медичного закладу.

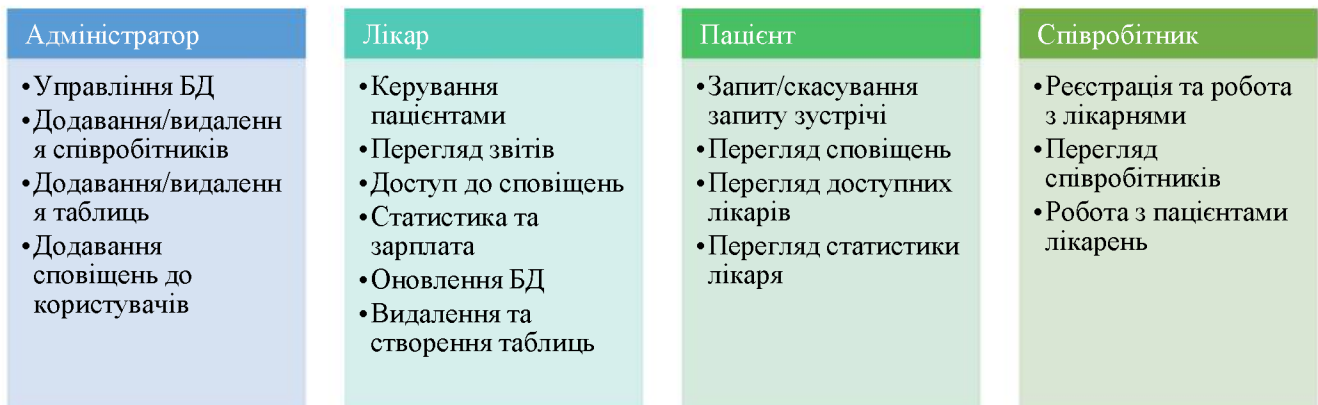


Рисунок 2.2 – Структура інформаційної системи медичного закладу

Адміністратор є головним у додатку. Весь додаток буде створено на основі стратегії його роботи. Для успішного запуску цієї програми адміністратор буде повинен спочатку увійти на сервер бази даних. Потім вказує категорію у функції входу та вводить дані авторизації. Пароль адміністратора буде зберігатися не в базі даних, а в змінній середовища. Адміністратор може зберегти свій пароль у змінній середовища та отримати пароль. Він може безпосередньо змінити змінну у відповідному модулі. Потім адмін має викликати функцію початку роботи програми, оскільки вона встановлює середовище за замовчуванням і створює сервер бази даних і таблиці. Адміністратор зможе додавати співробітників, видаляти

співробітників, бачити співробітників, які працюють додатково для лікарні, що є частиною, яка відноситься до управління персоналом. Також підрахувати загальний зарібок лікарні, створити таблицю лікарів, пацієнтів і співробітників і видалити їх. І важливою частиною є те, що він зможе додавати сповіщення до записів лікаря, співробітника або пацієнта, щоб вони залишалися в курсі своїх робочих процесів. Якщо будь-який лікар, працівник або пацієнт попросить адміністратора оновити базу даних, лише адміністратор матиме можливість оновити базу даних. Адміністратор також має право видаляти та створювати базу даних. Він також зможе остаточно видалити лікаря, співробітника або пацієнта з бази даних.

У випадку лікаря потрібно буде спочатку увійти в додаток або зареєструватися, ввівши правильну інформацію. Для розділу спеціальності лікар введе свою категорію, але зможе задати лише одну спеціальність. Після входу в програму отримає можливість додавати співробітників, видаляти співробітників і бачити своїх співробітників, які працюють під його керівництвом. Також зможе побачити своє останнє сповіщення від адміністратора, переглядати інформацію та звіти своїх пацієнтів, видаляти їх, бачити свою загальну зарплату. Він зможе аналізувати, які пацієнти належать до його спеціальності та хто звертається до нього на прийом.

Пацієнт повинен буде спочатку увійти або зареєструватися в програмі. Під час реєстрації він зможе запросити прийом лікаря (якщо він не знайомий з лікарями, то пропустить це питання). Після входу він може відвідувати всіх лікарів, які йому потрібні та запросити зустріч. Після того як його запит буде схвалено адміністратором, він зможе переглянути статистику свого лікаря. Він також матиме можливість видалити свій запит і побачити загальну вартість лікування.

Співробітник спочатку повинен увійти в програму, або він зможе зареєструватися. Щоб зареєструватися, йому потрібно вибрати свою зарплату за роботу, а також свою роботу. Робота співробітника може бути

реєстратором, помічником, прибиральницею і т.д. Як тільки він зареєструється на сервері бази даних, він отримає свого першого клієнта, а це не що інше, як лікарня. Він отримає лише 10% від своєї зарплати за роботу за працевлаштування у лікарні. Адміністратор лікарні може додавати співробітників до роботи в лікарні. Якщо адмін їх додасть, то вони отримають 100% своєї зарплати. Їх також можуть додати лікарі. Працівник може працювати в лікарні, а також у кількох лікарів.

2.3 Проектування бази даних інформаційної системи.

Побудуємо модель бази даних (рис. 2.3). БД використана в кваліфікаційній роботі складається з наступних сутностей:

- 1) «Admin» – інформація про головного користувача додатку.
- 2) «Doctor» – зберігається інформація про доктора, його записи та ціну за послуги.
- 3) «Patient» – зберігає дані про пацієнта та його записи на прийом.
- 4) «Employee» – зберігає дані про співробітника, його роботу та зарплатню.

Ці сутності складаються з множин атрибутів. Розглянемо їх детально.

Таблиця admin:

– username: унікальне ідентифікаторне ім'я користувача адміністратора.

Використовується для входу в систему та ідентифікації користувача;

– fullname: повне ім'я адміністратора. Використовується для відображення інформації про користувача у системі або для формальних сповіщень;

– email: електронна адреса адміністратора. Використовується для сповіщень, комунікації та інших аспектів взаємодії з користувачем;

– date_of_birth: дата народження адміністратора. Використовується для відображення віку користувача, персоналізації послуг або для перевірки на доступ до певних функцій в залежності від віку;

– password: зашифрований пароль адміністратора. Використовується для аутентифікації користувача під час входу в систему та забезпечує безпеку його облікового запису;

– notifications: інформація про сповіщення, що стосуються адміністратора, історія сповіщень.

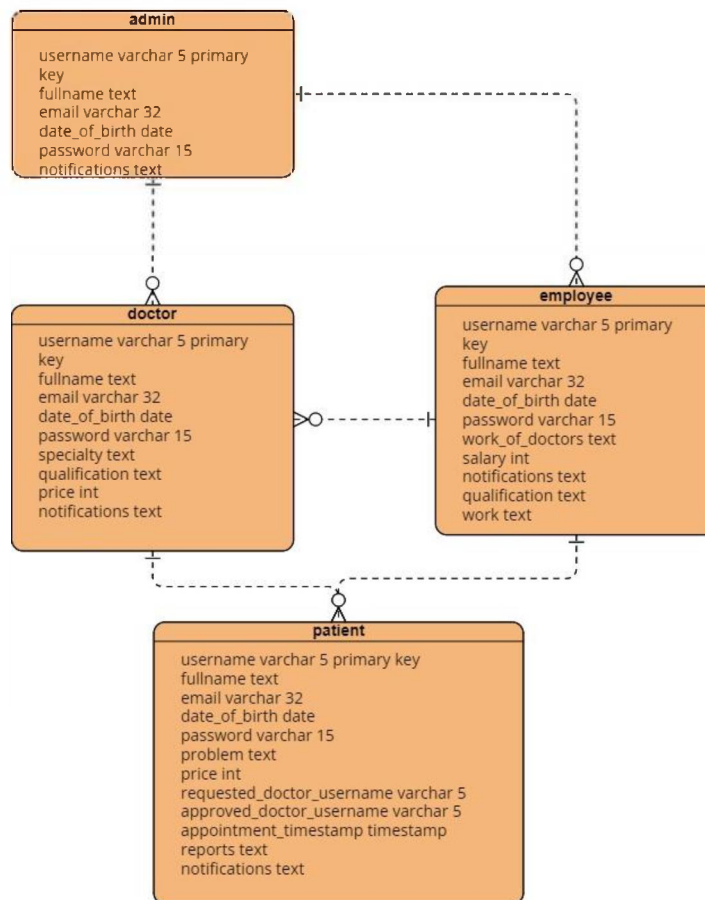


Рисунок 2.3 – Датологічна модель бази даних

Таблиця doctor:

– username: унікальне ідентифікаторне ім'я лікаря. Використовується для входу в систему та ідентифікації користувача;

– fullname: повне ім'я лікаря. Використовується для відображення інформації про лікаря у системі або для формальних сповіщень;

- email: електронна адреса лікаря. Використовується для комунікації з лікарем, надсилання сповіщень та інших аспектів взаємодії з ним;
- date_of_birth: дата народження лікаря. Використовується для визначення віку лікаря та персоналізації послуг для нього;
- password: зашифрований пароль лікаря. Використовується для аутентифікації лікаря при вході в систему та захисту його облікового запису;
- specialty: інформація про спеціалізацію лікаря. Вказує на медичну область, в якій лікар має досвід та експертизу;
- qualification: інформація про кваліфікацію лікаря. Вказує на професійні знання, навички і досвід лікаря.
- price: цінова політику лікаря. Вказує на вартість консультації або медичних послуг, які надає лікар;
- notifications: інформація про сповіщення, що стосуються лікаря, історія сповіщень.

Таблиця patient:

- username: ідентифікаторне ім'я пацієнта. Використовується для входу в систему та ідентифікації користувача;
- fullname: повне ім'я пацієнта. Використовується для відображення інформації про пацієнта у системі або для формальних сповіщень;
- email: електронна адреса пацієнта. Використовується для комунікації з пацієнтом, надсилання сповіщень та інших аспектів взаємодії з ним;
- date_of_birth: дата народження пацієнта. Використовується для визначення віку пацієнта та персоналізації послуг для нього;
- password: зашифрований пароль пацієнта. Використовується для аутентифікації пацієнта при вході в систему та захисту його облікового запису;
- problem: опис проблеми або стану здоров'я пацієнта, яка потребує медичної допомоги;
- requested_doctor_username: ім'я користувача лікаря, якого запитав пацієнт для консультації чи лікування;

- `approved_doctor_username`: ім'я користувача лікаря, який був схвалений для надання медичної допомоги пацієнту;
- `appointment_timestamp`: дата та час призначеного прийому чи консультації з лікарем;
- `reports`: медичні звіти або інша документація, пов'язана зі станом здоров'я пацієнта та його лікуванням;
- `notifications`: інформація про сповіщення, які стосуються пацієнта.

Таблиця `employee`:

- `username`: ідентифікаторне ім'я працівника. Використовується для входу в систему та ідентифікації користувача;
- `fullname`: повне ім'я працівника. Використовується для відображення інформації про працівника у системі або для формальних сповіщень;
- `email`: електронна адреса працівника. Використовується для комунікації з працівником, надсилання сповіщень та інших аспектів взаємодії з ним;
- `date_of_birth`: дата народження працівника. Використовується для визначення віку працівника та персоналізації послуг для нього;
- `password`: зашифрований пароль працівника. Використовується для аутентифікації працівника при вході в систему та захисту його облікового запису;
- `work`: інформація про робочу посаду працівника;
- `qualification`: інформація про кваліфікацію працівника. Вказує на професійні знання, навички і досвід працівника.
- `work_of_doctors`: інформація про роботу лікарів, з якими працює цей працівник (наприклад, список лікарів, з якими він пов'язаний або яким від підпорядкований);
- `salary`: інформація про заробітну плату працівника;
- `notifications`: інформація про сповіщення, які стосуються працівника.

Наступним кроком є аналіз структури таблиць бази даних. Для кожного поля в таблиці вказується його розмір (кількість символів) та тип даних. Для первинних ключів, які є унікальними ідентифікаторами записів, необхідно заборонити значення "NULL" (тобто невизначені значення). Щодо інших полів, можливість допуску невизначених значень визначається вимогами до бази даних, які визначаються у завданні.

Таблиця 2.2

Склад таблиці admin

Найменування атрибута	Тип поля	Розмір поля	Допустимість невизначених значень
username	varchar	5	Not Null
fullname	text		Not Null
email	varchar	32	Not Null
date_of_birth	date		Not Null
password	varchar	15	Not Null
notifications	text		Not Null

Таблиця 2.3

Склад таблиці doctor

Найменування атрибута	Тип поля	Розмір поля	Допустимість невизначених значень
username	varchar	5	Not Null
fullname	text		Not Null
email	varchar	32	Not Null
date_of_birth	date		Not Null
password	varchar	15	Not Null
specialty	text		Not Null
qualification	text		Not Null
price	int		Not Null
notifications	text		Not Null

Таблиця 2.4

Склад таблиці patient

Найменування атрибута	Тип поля	Розмір поля	Допустимість невизначених значень
username	varchar	5	Not Null
fullname	text		Not Null
email	varchar	32	Not Null
date_of_birth	date		Not Null
password	varchar	15	Not Null
problem	text		Not Null
price	int		Not Null
requested_doctor_username	varchar	5	Null
approved_doctor_username	varchar	5	Null
appointment_timestamp	timestamp		Null
reports	text		Null
notifications	text		Not Null

Таблиця 2.5

Склад таблиці employee

Найменування атрибута	Тип поля	Розмір поля	Допустимість невизначених значень
username	varchar	5	Not Null
fullname	text		Not Null
email	varchar	32	Not Null
date_of_birth	date		Not Null
password	varchar	15	Not Null
work_of_doctors	text		Null
salary	int		Not Null
notifications	text		Not Null
qualification	text		Not Null
work	text		Not Null

Висновки за розділом 2

У другому розділі було проведено процес проектування моделі управління персоналом медичного закладу, створено модель управління персоналом, яка враховує специфіку медичних закладів та спрямована на підвищення ефективності роботи персоналу, а також розроблено структуру бази даних. Ці кроки є ключовими для подальшої реалізації системи управління медичним закладом і забезпечення її ефективної роботи.

РОЗДІЛ 3

РОЗРОБКА ТА ТЕСТУВАННЯ МОДЕЛІ УПРАВЛІННЯ ПЕРСОНАЛОМ МЕДИЧНОГО ЗАКЛАДУ

3.1 Обґрунтування обраних технологій та засобів розробки системи

Для створення додатку було обрано наступні технології та засоби розробки систем:

- Python;
- PostgreSQL.

Розглянемо переваги обраних технологій.

Python — це інтерпретована об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою. Вона має репутацію мови, зручної для початківців, замінивши Java як найпоширенішу початкову мову, оскільки вона справляється з більшою частиною складності для користувача, дозволяючи початківцям зосередитися на повному розумінні концепцій програмування, а не на найменших деталях [10].

Python використовується для веб-розробки на стороні сервера, розробки програмного забезпечення, математики та системних сценаріїв, і популярна для швидкої розробки додатків, а також як мова сценаріїв або з'єднувальна мова для зв'язування існуючих компонентів завдяки своїм високорівневим вбудованим структурам даних, динамічному типу і динамічному зв'язуванню. Витрати на технічне обслуговування програми зменшуються завдяки Python завдяки легкому освоєнню синтаксису та акценту на зручності читання. Крім того, підтримка Python модулів і пакетів полегшує модульні програми та повторне використання коду. Python — це мова спільноти з відкритим кодом, тому багато незалежних програмістів постійно створюють бібліотеки та функціональні можливості для неї.

Особливості та переваги Python:

- сумісний із різними платформами, включаючи Windows, Mac, Linux, Raspberry Pi та інші;
- використовує простий синтаксис, який можна порівняти з англійською мовою, що дозволяє розробникам використовувати менше рядків, ніж інші мови програмування;
- працює на системі інтерпретатора, яка дозволяє негайно виконувати код, швидко відстежуючи прототипування;
- можна обробляти процедурним, об'єктно-орієнтованим або функціональним способом.

Python — мова з динамічною типізацією, є особливо гнучкою, усуває жорсткі правила створення функцій і пропонує більшу гнучкість у вирішенні проблем за допомогою різноманітних методів. Вона також дозволяє компілювати та запускати програми аж до проблемної області, оскільки використовує перевірку типу під час виконання, а не під час компіляції. Її гнучкість, оскільки вона динамічна типізована, вимагає значної кількості посилань, щоб отримати правильне визначення, що сповільнює продуктивність.

PostgreSQL — це розширена реляційна база даних корпоративного класу з відкритим кодом, яка підтримує як SQL (реляційні), так і JSON (нереляційні) запити. Це високостабільна система керування базами даних. Однією з головних причин для використання PostgreSQL є його великий список функцій. PostgreSQL має неймовірну кількість функцій, пов'язаних із продуктивністю. Один із найфундаментальніших способів відмінності PostgreSQL від більшості інших реляційних баз даних полягає в його структурі ядра. Більшість реляційних баз даних краще описати як системи керування реляційними базами даних (РСКБД). РСКБД — це програмне забезпечення, спеціально розроблене для роботи з реляційними базами даних, де дані зберігаються в табличних структурах із попередньо визначеними стовпцями та типами даних. Дані можна запитувати, змінювати та отримувати за

допомогою методів, заснованих на реляційній алгебрі, зазвичай за допомогою мови структурованих запитів (SQL). PostgreSQL технічно є системою керування об'єктно-реляційною базою даних (ОРСКБД). Це означає, що він має такі ж реляційні можливості, як і РСКБД, але додатково має деякі об'єктно-орієнтовані функції [11].

PostgreSQL дозволяє визначити :

- Власні складні типи даних;
- Функції перевантаження для роботи з різними типами даних аргументів;
- Визначити зв'язок успадкування між таблицями.

Підвищена гнучкість дає змогу моделювати різні типи та зв'язки в системі баз даних, а не ззовні у програмах. Це може допомогти підтримувати узгодженість і наблизити заплановану поведінку до фактичних даних, безпеки, програмних розширень та конфігурації.

3.2 Опис моделі системи управління персоналом медичного закладу

Системи управління персоналом медичного закладу — це інтегроване програмне забезпечення, яке керує різними напрямками робочих процесів медичного закладу. Воно забезпечує безперебійну роботу системи охорони здоров'я разом із адміністративним, медичним, юридичним і фінансовим контролем. Це запорука успішної роботи закладу охорони здоров'я. Система є серверною частиною системи управління лікарнею.

Адміністратор є головним виконавцем для усієї бази даних, він має більший набір функцій і може здійснювати їх, щоб відстежувати діяльність кожного користувача. Кожен користувач відрізняється своїм унікальним 3-значним іменем користувача. Імена користувачів є первинними ключами кожної таблиці. Усі операції виконуються за допомогою імені користувача.

Після запуску додатку з'явиться головне вікно (рис. 3.1), де є кнопка «Let's Start!» для початку роботи в системі.

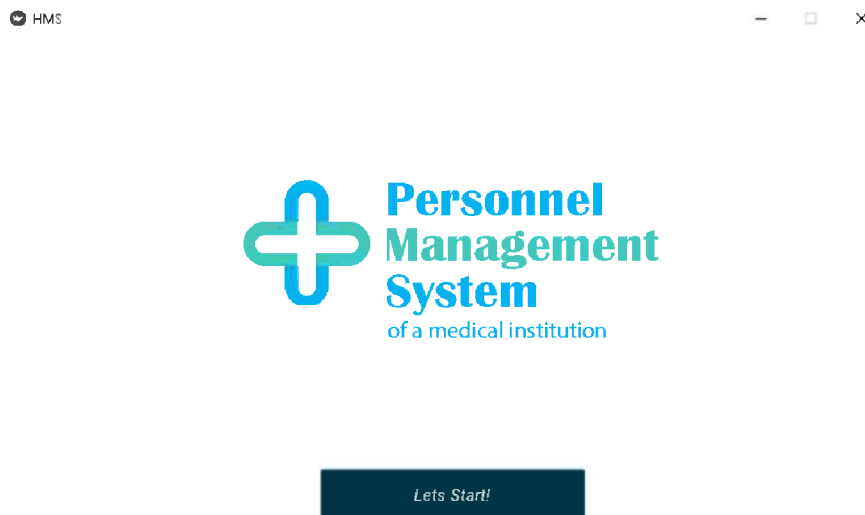


Рисунок 3.1 – Головне вікно системи управління персоналом медичного закладу

Після запуску системи з'являється вікно авторизації, у якому можна увійти (zareєструватися) як адміністратор, лікар, пацієнт або співробітник. На рис. 3.2 зображено вікно авторизації адміністратора.

The screenshot shows a window titled 'HMS' with standard window controls. The main content area is titled 'SignUp Form' in a large, dark grey font, with the subtitle 'Choose Your Category' in a smaller, blue font below it. Below the subtitle, there are four dark grey buttons with white text: 'Admin', 'Doctor', 'Patient', and 'Employee'. The 'Admin' button is highlighted with a blue border. Below these buttons, there is a registration form with the following fields: 'Username:' (text input), 'Fullname:' (text input), 'Email:' (text input), 'Date of Birth:' (text input with the placeholder 'YYYY-MM-DD'), and 'Password:' (text input). Below the form, there is a blue link that says 'Already have an account? Log in.' and a dark blue rectangular button with the white text 'Submit'.

Рисунок 3.2 – Вікно авторизації адміністратора

Коли адміністратор авторизується, то отримує можливість виконувати основні функції системи (рис 3.3). Він може керувати лікарями та співробітниками, переглядати список пацієнтів та оновлювати його.

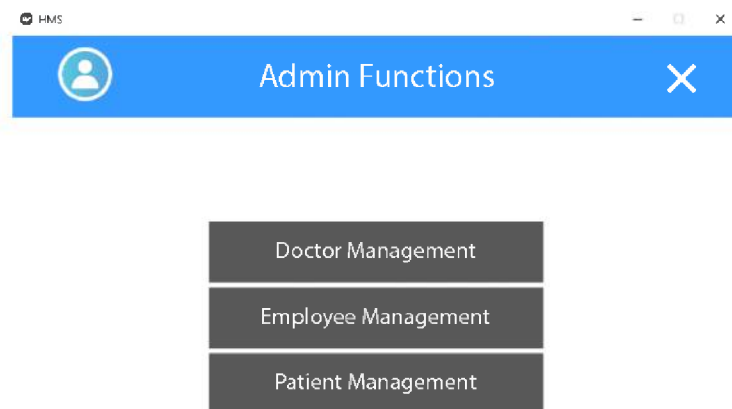


Рисунок 3.3 – Вікно функцій адміністратора

Переглянемо вікно управління лікарями на рис.3.4.

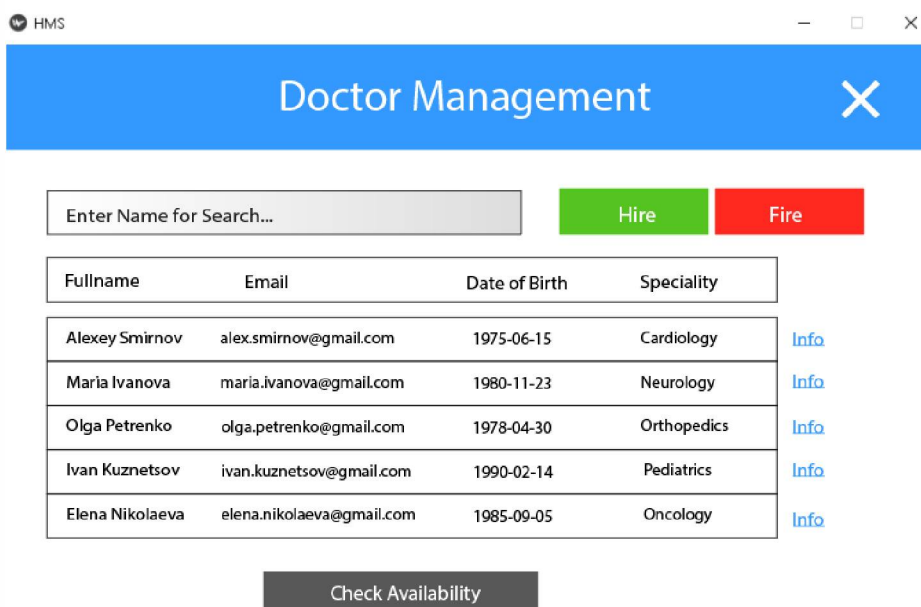


Рисунок 3.4 – Вікно управління лікарями

Адміністратор має доступ до інформації про лікарів (рис. 3.5)., може аналізувати їх зайнятість у зазначену дату (рис. 3.6).

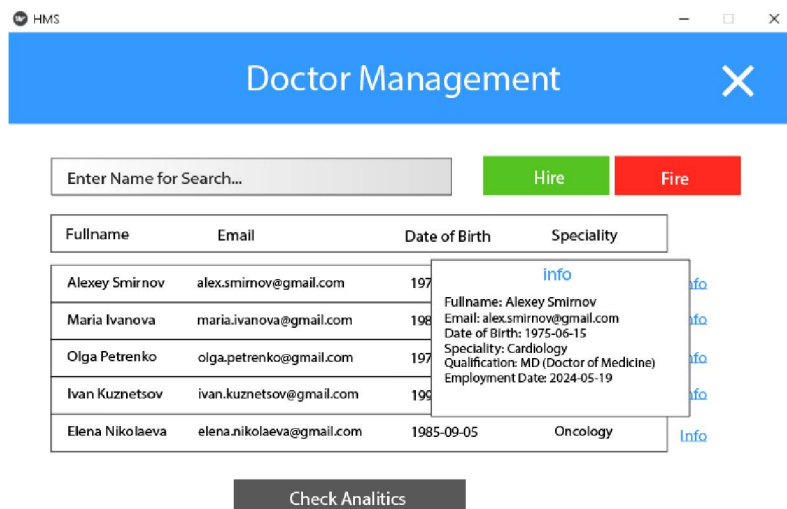


Рисунок 3.5 – Перегляд повної інформації про лікаря

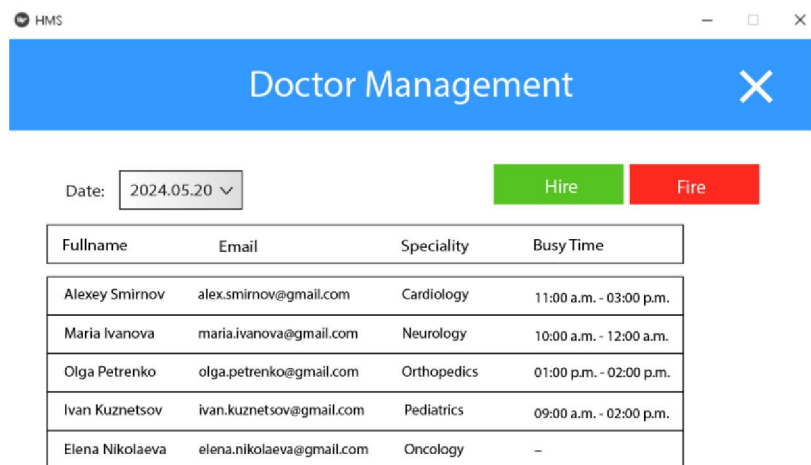


Рисунок 3.6 – Вікно зайнятості лікарів

Він може найняти (рис.3.7) чи звільнити (рис.3.8) лікаря, при цьому завантаживши відповідний договір.

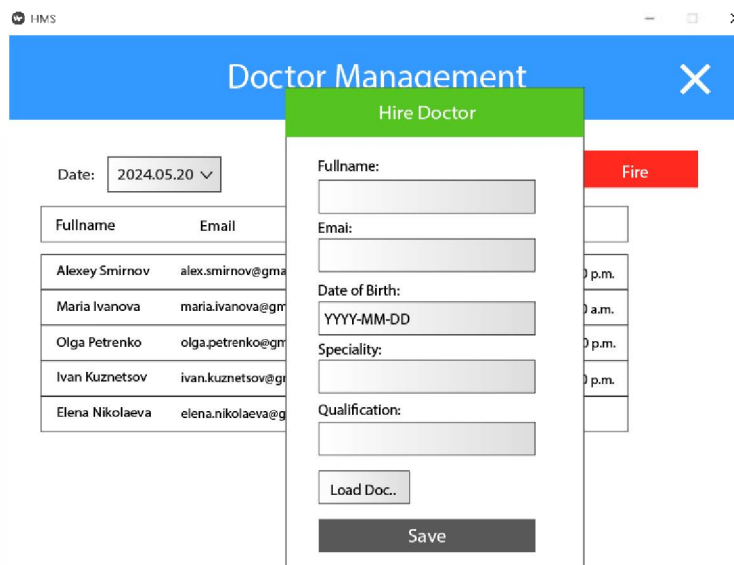


Рисунок 3.7 – Влаштування лікаря на роботу

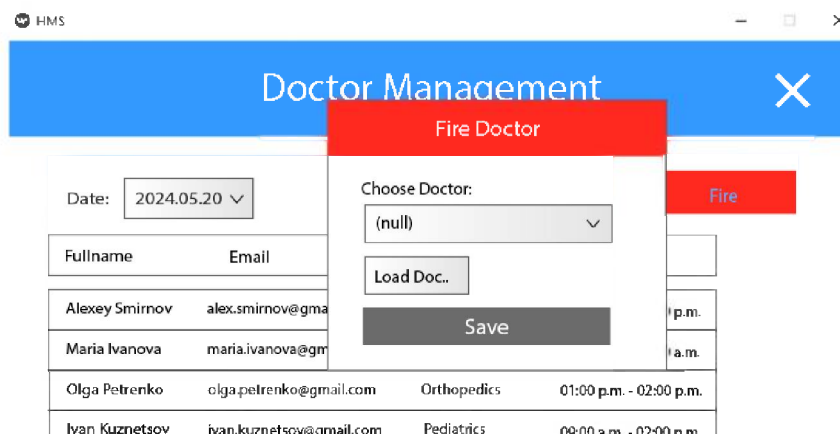


Рисунок 3.8 – Звільнення лікаря

Розглянемо управління співробітниками, воно не відрізняється від управління лікарями. Є можливість переглядати інформацію про співробітників, звільняти та приймати на роботу, аналізувати зайнятість.

На рисунку 3.9 показане головне вікно управління співробітниками.

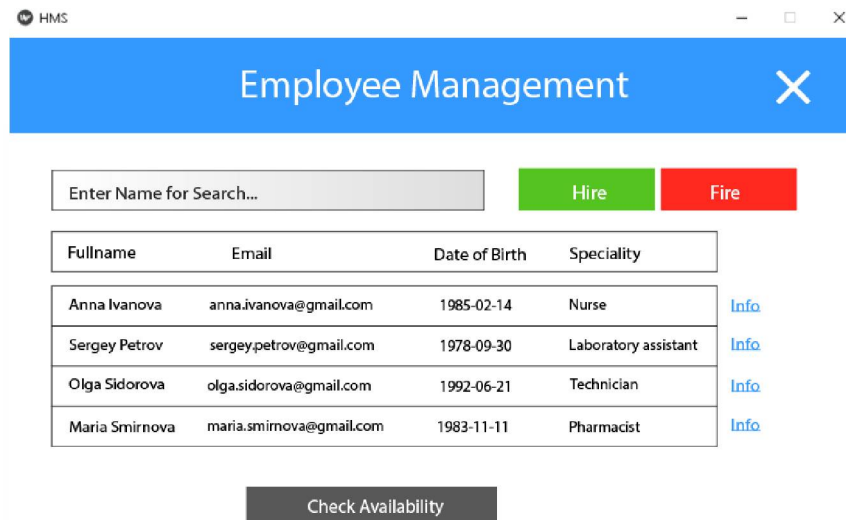


Рисунок 3.9 – Вікно управління співробітниками

Вікно управління пацієнтами (рис 3.10) має такі функції:

- Перегляд записів;
- Додавання або видалення записів;
- Перегляд інформації про пацієнта.

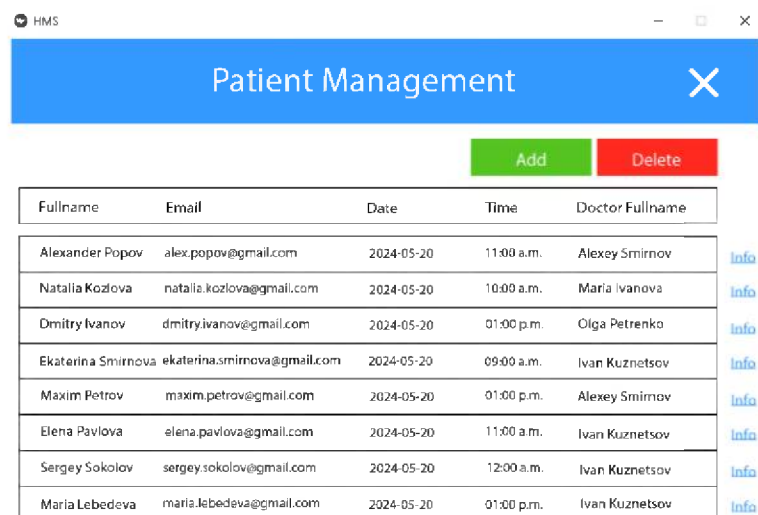


Рисунок 3.10 – Вікно управління запитами пацієнтів

Розглянемо функціонал системи зі сторони лікаря (рис.3.11). Він має такі функції:

- перегляд інформації про запис пацієнтів та назначених співробітників;
- додавання або видалення записів;
- відправлення звітів;
- призначити співробітника (наприклад медсестру) як помічника.

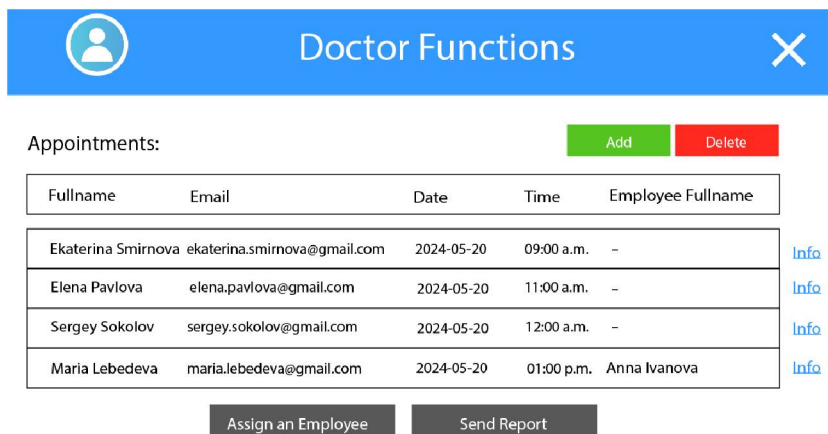


Рисунок 3.11 – Вікно функцій лікаря

Якщо входить у систему у ролі співробітника (рис 3.12), то є можливість переглядати свої призначення та інформацію про них.

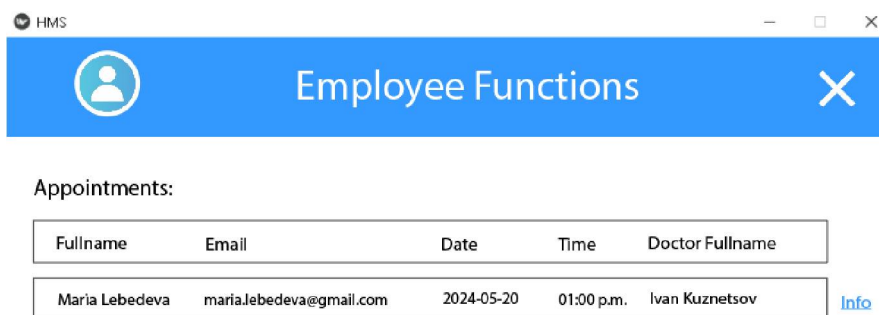


Рисунок 3.12 – Вікно функцій співробітника

Розглянемо функціонал системи зі сторони пацієнта (рис. 3.13). Пацієнт може переглядати свої записи на прийом, відмінити та створити запис, переглядати список лікарів у медичному закладі.

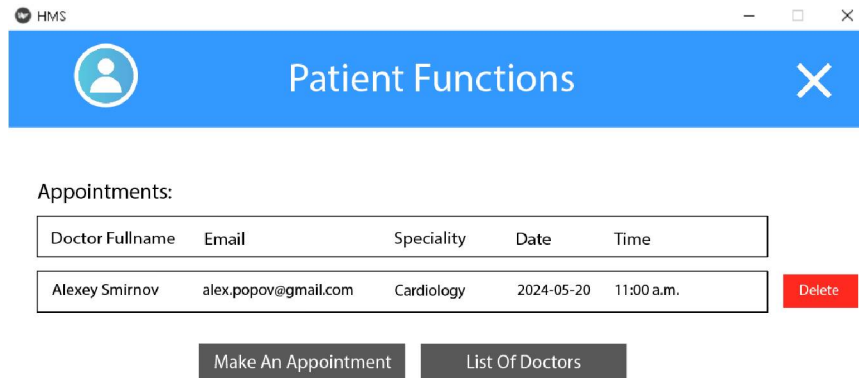


Рисунок 3.13 – Вікно функцій пацієнта

Висновки за розділом 3

У третьому розділі було розглянуто процес розробки та тестування моделі управління персоналом медичного закладу. Проведено вибір технологій для розробки інформаційної системи, що базується на критеріях надійності, масштабованості, безпеки та зручності використання. Результати показали готовність системи до використання з можливими рекомендаціями щодо подальшого вдосконалення.

ВИСНОВКИ

У кваліфікаційній роботі, присвяченій розробці моделі системи управління персоналом медичного закладу, було проведено комплексний аналіз структури медичного закладу. У ході дослідження було визначено ключові компоненти ефективної моделі управління персоналом, такі як підбір, адаптація, оцінка ефективності, управління кар'єрою та підтримка працівників. Кожен з цих компонентів відіграє важливу роль у забезпеченні високої якості медичних послуг, а їх взаємодія створює комплексну систему управління, спрямовану на підвищення

Важливим аспектом роботи стало використання сучасних інформаційних технологій для автоматизації процесів управління персоналом. Розробка та впровадження інформаційної системи дозволило значно зменшити адміністративне навантаження на керівників, підвищити точність даних про персонал, а також забезпечити швидкий доступ до необхідної інформації. Цей технологічний підхід сприяв підвищенню ефективності управлінських рішень і оперативності їх виконання.

Проведене дослідження також підкреслило важливість безперервного вдосконалення процесів управління персоналом. З огляду на швидкий розвиток медичних технологій та зміну умов праці, необхідно постійно переглядати та оновлювати методи і підходи до управління людськими ресурсами. Це включає впровадження нових технологій, удосконалення навчальних програм та адаптацію систем до нових вимог і очікувань працівників.

Особлива увага була приділена проектуванню структури моделі управління персоналом. Це сприяло визначенню сильних та слабких сторін в управлінні персоналом і дозволило вносити необхідні корективи для покращення моделі.

Модель забезпечує системний підхід до управління людськими ресурсами, інтегруючи сучасні технології. Це дозволяє не тільки підвищити якість обслуговування, але й створити сприятливі умови для роботи, що є ключовим фактором у підвищенні ефективності роботи персоналу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Васюк Н. О. Керівні кадри в системі охорони здоров'я: сутність понять. Завдання держави у забезпеченні ефективної кадрової політики галузі охорони здоров'я / Н. О. Васюк // Матеріали науково-практичної конференції. – Київ. – 2015. – С. 18-23.
2. Кошелупов І. Ф. Соціально-економічна модель механізму мотивації праці управлінців підприємств / І. Ф. Кошелупов, В. О. Кравченко // Вісник соціально-економічних досліджень. – Одеса : ОДЕУ, 2008. – С. 124-131.
3. Transforming our world: the 2030 Agenda for Sustainable Development. Resolution adopted by the General Assembly on 25 September 2015. URL: <https://documentsddsny.un.org/doc/UNDOC/GEN/N15/291/89/PDF/N1529189.pdf?OpenElement>(accessed 12 April 2020).
4. Vvedeno v diyu zakon shchodo avtonomizatsiyi zakladiv okhorony zdorov'ya. Apteka.online.ua. [The law on the autonomy of health care institutions has been put in place. Pharmacy.online.ua.]. URL: <https://www.apteka.ua/article/433045> (accessed 15 April 2020).
5. Пономаренко В.С. Інформаційні системи в управлінні персоналом : навч. посібн. / В.С. Пономаренко, І.В. Журавльова, І.Л. Латишева. - Харків : Вид-во ХНЕУ, 2008. - 336 с.
6. Ринейська Л.С. Менеджмент персоналу як фактор підвищення ефективності міжнародного бізнесу/ Л.С. Ринейська// Економіка і регіон. – 2009. - №2. – с. 209-212.
7. Медична інформаційна система Clinica Web. [Електронний ресурс]. – режим доступу: URL: <https://www.clinica-web.ua/about/> (дата звернення - 19.05.2024).
8. Електронна медична система HELSI.ME. [Електронний ресурс]. – режим доступу: URL: <https://helsi.me/about> (дата звернення - 15.04.2024).

9. Електронна система охорони здоров'я eHealth. [Електронний ресурс]. – режим доступу: URL: <https://ehealth.gov.ua/>(дата звернення - 15.04.2024).

10. PostgreSQL: The World's Most Advanced Open Source Relational Database. [Електронний ресурс]. – режим доступу: URL: <https://www.postgresql.org/about/> (дата звернення - 15.04.2024).

11. Python would rather not use anything else. [Електронний ресурс]. – режим доступу: URL: <https://www.python.org/about/> (дата звернення - 15.04.2024).

ДОДАТКИ


Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) бакалавр
галузь знань: 15 – Автоматизація та приладобудування
спеціальність: 151 – Автоматизація та комп'ютерно-інтегровані технології

ЗАТВЕРДЖУЮ

Завідувач кафедри теоретичної
та прикладної системотехніки

 д.т.н., проф. Шматков С. І.
«21» грудня 2024 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Нестеренка Владислава Валентинівна

(прізвище, ім'я, по батькові студента)

1. Тема роботи «Модель системи управління персоналом медичного закладу»

керівник роботи Стрілець Вікторія Євгенівна, канд. техн. наук, доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від від «03» травня 2024 року № 4101-5/909

2. Строк подання студентом роботи 31 травня 2024 р.

3. Перелік питань, які потрібно розробити

- 1) Аналіз організаційної структури медичного закладу та існуючих систем управління персоналом медичного закладу.
- 2) Створення моделі управління персоналом медичного закладу.
- 3) Розробка програмної реалізації моделі управління персоналом медичного закладу.
- 4) Тестування моделі системи управління персоналом медичного закладу.

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Постановка задачі розробки моделі системи управління персоналом медичного закладу.	21.12.2023 - 10.01.2024
2	Аналіз організаційної структури медичного закладу.	19.12.2023 - 25.12.2023
3	Порівняльний аналіз існуючих систем управління персоналом медичного закладу.	25.12.2023 - 2.01.2024
4	Проектування бази даних для моделі управління.	2.01.2024 - 2.02.2024
5	Створення дизайну інтерфейсу та реалізація моделі управління персоналом.	2.01.2024 - 2.02.2024
6	Тестування моделі системи управління персоналом медичного закладу.	3.02.2024 - 30.03.2024
7	Оформлення пояснювальної записки до кваліфікаційної роботи	3.03.2024 - 30.04.2024
8	Представлення кваліфікаційної роботу керівнику та рецензенту	31.03.2024 - 27.05.2024

5. Дата видачі завдання 21.12.2023 р.

Студент

Нестеренко В. В.

ініціали, прізвище



підпис

Керівник роботи

Стрілець В. Є.

ініціали, прізвище



підпис

**Технічне завдання
на розробку програмного виробу**

«Модель системи управління персоналом медичного закладу».

1.	Введення	<p>1.1. Назва: Модель системи управління персоналом медичного закладу</p> <p>1.2. Галузь застосування: Охорона здоров'я та медичних послуг</p>
2.	Підстава для розробки	<p>2.1. Навчальний план за спеціальністю 151 – Автоматизація та комп'ютерно-інтегровані технології</p> <p>2.2. Завдання на кваліфікаційну роботу бакалавра №4101-5/895 від «23» травня 2024 (представити як Додаток А до пояснювальної записки до кваліфікаційної роботи).</p>
3.	Призначення розробки	<p>3.1. Мета розробки: впровадження інноваційного підходу до управління людськими ресурсами у медичній сфері.</p> <p>3.2. Призначення розробки створити комп'ютерну модель системи управління персоналом для медичного закладу з метою оптимізації робочих процесів, підвищення ефективності використання ресурсів та покращення якості медичного обслуговування</p> <p>3.3. Вихідні дані розробки: модель системи управління персоналом медичного закладу.</p> <p>Вхідні дані: процес управління персоналом медичного закладу.</p>
4.	Технічні вимоги до програмного виробу	<p>4.1. Вимоги до функціональних характеристик: адекватність, простота, об'єктивність, чутливість, стійкість.</p> <p>4.2. Вимоги до надійності: модель повинна мати механізми збереження даних, щоб уникнути втрати важливої інформації.</p> <p>4.3. Вимоги до умов експлуатації: немає</p> <p>4.4. Вимоги до складу і параметрів технічних засобів: для виконання програми повинен підходити ПК із будь-якою операційною системою сімейства Windows, Linux/Unix, Mac OS X, OS/2, Amiga.</p>

		<p>4.5. Вимоги до інформаційної та програмної сумісності: підтримка ОС Linux або Windows 10, підтримка різних платформ.</p> <p>4.6. Вимоги до маркування та упаковки: вимоги до маркування та упакування не представляються.</p> <p>4.7. Вимоги до транспортування і зберігання: вимоги до транспортування та зберігання не представляються.</p> <p>4.8. Спеціальні вимоги: спеціальні вимоги до програмного виробу не пред'являються.</p>	
5.	Вимоги до програмної документації	<p>Програмною документацією до виробу «Модель системи управління персоналом медичного закладу» вважати:</p> <p>1) Справжнє Технічне завдання на розробку виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).</p> <p>2) Методику розрахунку інформативності змінних стану (у вигляді глав 3.2 та 3.3 пояснювальної записки до кваліфікаційної роботи).</p> <p>3) Опис виробу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи)</p>	
6.	Вимоги до техніко-економічних показників	<p>Програмною документацією до виробу «Модель системи управління персоналом медичного закладу» вважати:</p> <p>1) Справжнє Технічне завдання на розробку виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).</p> <p>2) Опис програмного виробу (представити в Розділі 3 пояснювальної записки до кваліфікаційної роботи).</p> <p>3) Джерела базової інформації.</p>	
7.	Стадії і етапи розробки	Дата	Назва етапу
		від 21 грудня 2023 до 10 січня 2024	Постановка задачі розробки моделі системи управління персоналом медичного закладу.
		від 19 грудня 2023 до 25 грудня 2023	Аналіз організаційної структури медичного закладу.
		від 25 грудня 2023 до 2 січня 2024	Порівняльний аналіз існуючих систем управління

		від 2 січня 2024 до 2 лютого 2024	персоналом медичного закладу. Проектування бази даних для моделі управління.
		від 2 січня 2024 до 2 лютого 2024	Створення дизайну інтерфейсу та реалізація моделі управління персоналом.
		від 3 лютого 2024 до 30 березня 2024	Тестування моделі системи управління персоналом медичного закладу.
		від 3 березня 2024 до 30 березня 2024	Оформлення пояснювальної записки до кваліфікаційної роботи
		від 31 березня 2023 до 27 травня 2023	Представлення кваліфікаційного проєкту керівнику кваліфікаційної роботи та рецензенту.
8.	Порядок контролю і приймання програмного продукту (моделі)	<ol style="list-style-type: none"> 1. Перевірку ходу розробки програми виконувати раз в 3 тижні. 2. Захист розробленої моделі провести на засіданні Атестаційної комісії. 3. Пояснювальну записку подати на паперових носіях в 1 примірнику і в електронному вигляді в 1 примірнику на CD-R компакт-диску. 	

Виконавець
студентка групи КУ- 41
Нестеренко В. В.



Замовник
к.т.н.
Стрілець В. Є.



Програма і методика випробувань програмного виробу

«Модель системи управління персоналом медичного закладу»

1. Об'єкт випробувань

Назва програмного виробу : «Модель системи управління персоналом медичного закладу»

1. Галузь застосування : Охорона здоров'я та медичних послуг
2. Перераховані відомості запозичуються з відповідних розділів Технічного завдання.

2. Мета випробувань

Перевірка відповідності функціональності програмної реалізації системи заявленим функціональним можливостям в технічному завданні (Додаток Б до пояснювальної записки до кваліфікаційної роботи).

3. Загальні положення

1. Підстави для проведення випробувань

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

2. Місце і тривалість випробувань

Приймальні (приймально-здавальні) випробування проводяться на базі комп'ютерного класу кафедри в період роботи атестаційної комісії.

3. Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї програми і методики випробувань.

4. Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавця та інших осіб, присутніх на засіданні.

4. Вимоги до програми або програмного виробу

Модель повинна задовольняти наступним вимогам:

- 4.1. Вимоги до функціональних характеристик: адекватність, простота, об'єктивність, чутливість, стійкість.
- 4.2. Вимоги до надійності: модель повинна мати механізми збереження даних, щоб уникнути втрати важливої інформації.
- 4.3. Вимоги до умов експлуатації: немає.

4.4. Вимоги до складу і параметрів технічних засобів: для виконання програми повинен підходити ПК із будь-якою операційною системою сімейства Windows, Linux/Unix, Mac OS X, OS/2, Amiga.

4.5. Вимоги до інформаційної та програмної сумісності: підтримка ОС Linux або Windows 10, підтримка різних платформ.

4.6. Вимоги до маркування та упаковки: вимоги до маркування та упакування не представляються.

4.7. Вимоги до транспортування і зберігання: вимоги до транспортування та зберігання не представляються.

4.8. Спеціальні вимоги: спеціальні вимоги до програмного виробу не пред'являються.

5. Вимоги до програмної документації

Документацією до виробу «Модель системи управління персоналом медичного закладу» вважати:

- 1) Документація по мові програмування та додаткові мануали.
- 2) Програму і методику випробувань розробленої програми (представити як Додаток В до пояснювальної записки до кваліфікаційної роботи).
- 3) Опис програмного виробу (представити в Розділі 3 пояснювальної записки до кваліфікаційної роботи).
- 4) Джерела базової інформації.

6. Засоби і порядок випробувань

6.1 Засоби випробувань

Випробування проводяться на технічних засобах, яких персональний комп'ютер, ноутбук.

Випробування проводяться з використанням програмних засобів, яких Pucharm, PostgreSQL.

6.2 Порядок проведення випробувань

Як правило, випробування проводяться в два етапи:

- ознайомчий (1-й етап);
- власне випробування програмного виробу (2-й етап).

Перелік перевірок, що проводяться на 1 етапі випробувань, включає в себе:

- 1) перевірку комплектності складу програмної документації здійснюється за критерієм наявності зазначеної в ТЗ документації;
- 2) перевірку якості програмної документації. Перевірку здійснювати за критерієм відповідності вимогам ГОСТ 19.301-79 ЕСПД. «Програма і методика випробувань».

Перелік перевірок, що проводяться на 2 етапі випробувань, включає в себе:

- 1) Перевірку відповідності технічних характеристик програми вимогам технічного завдання.
- 2) Перевірку ступеня виконання функціональних вимог до програми.
- 3) Методику проведення перевірок:
 - a) запуснути програму.
 - b) порядок проведення випробувань:
 - зробити налаштування;
 - перевірити чи працює програма;
 - перевірити чи формується звіт.
- 4) Якщо перевірки на першому та другому етапах виконано успішно, то виріб вважається таким, що пройшов випробування.

Для проведення випробувань пропонується тест 1, тест 2 та тест 3.

Тест 1

1. Перевірка виконання програми;
2. Запуск головної сторінки додатку.

HMS

- □ ×



Lets Start!

Рис. В.1 Тест 1

Тест 2

1. Перевірка виконання програми;
2. Звільнення лікаря.

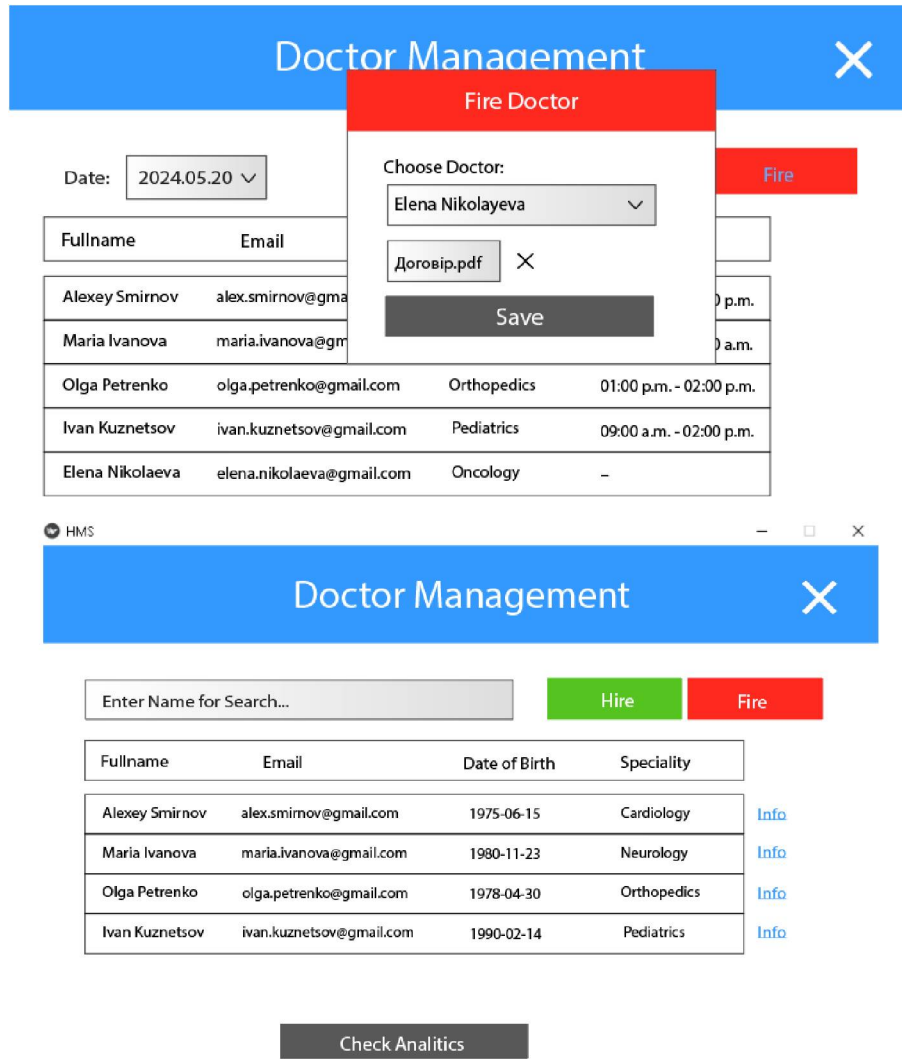


Рис. В.2 Тест 2

Тест 3

1. Перевірка виконання програми;
2. Прийом на роботу співробітника.

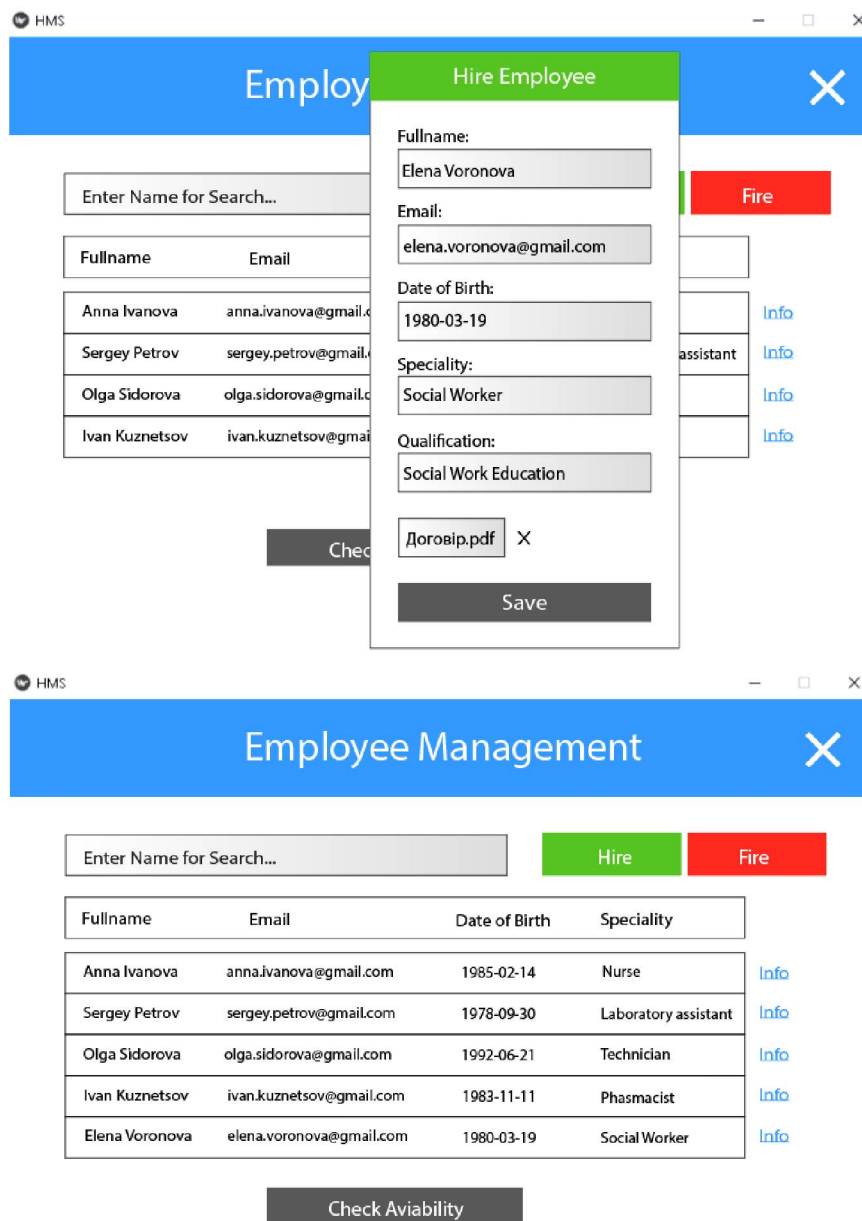


Рис. В.3 Тест 3

Тест вважається пройденим, якщо відбуваються вказані операції і їх відображення у прикладній програмі.

Висновки: тест 1 успішно пройшов випробування, тест 2 успішно пройшов випробування і тест 3 успішно пройшов випробування. Випробування пройшло успішно.

Виконавець:

студент групи КУ-41

Нестеренко В. В.

Лістинг коду

```

import psycopg2
from doctor import salary as doctor_salary
from patient import cost as patient_cost
from employee import salary as employee_salary
from DB_config import config
import constant

def see_info(logged_in, category, username):
    if logged_in:
        username = username.strip().lower()
        with psycopg2.connect(**config()) as info_check:
            curl = info_check.cursor()
            curl.execute(f"""
                SELECT * FROM {category} WHERE username =
'{username}';
            """)
            info = curl.fetchall()
            curl.close()
            info_check.close()
            return info

def update_db(category, username, fieldname, updated_value,
logged_in):
    if logged_in:
        date = "DATE" if fieldname == "date_of_birth" else ""
        with psycopg2.connect(**config()) as updating:
            curl = updating.cursor()
            curl.execute(f"""
                UPDATE {category} SET {fieldname} =
'{updated_value}' WHERE username = {date}'{username}';
            """)
            print(curl.statusmessage)
            curl.close()
            updating.commit()
            updating.close()
            return "Database Updated"

def recent_notifications(username, logged_in, limit = 1):
    if logged_in:
        with psycopg2.connect(**config()) as latest:
            curl = latest.cursor()
            curl.execute("""
                SELECT notifications FROM admin WHERE username =
%s;
            """, [username])

```

```

        notifications_string = curl.fetchall()[0][0]
        list_of_notifications =
notifications_string.split(", ")
        curl.close()
        latest.close()
        return list_of_notifications[-limit:] if limit <
len(list_of_notifications) else list_of_notifications[-
len(list_of_notifications):]

def add_notification(notification, category, receiver_username,
logged_in):
    if logged_in:
        with psycopg2.connect(**config()) as add_notifications:
            curl = add_notifications.cursor()
            curl.execute(f"""
                SELECT notifications FROM {category} WHERE
username = '{receiver_username}';
            """)
            notifications_string = curl.fetchall()[0][0]
            list_of_notifications =
notifications_string.split(", ")
            list_of_notifications.append(notification)
            new_notifications_string = ",
".join(list_of_notifications)
            curl.close()
            cur2 = add_notifications.cursor()
            cur2.execute(f"""
                UPDATE {category} SET notifications =
'{new_notifications_string}' WHERE username =
'{receiver_username}';
            """)
            print(cur2.statusmessage)
            cur2.close()
            add_notifications.commit()
        add_notifications.close()
        return "Notification Added"

def total_earning(logged_in):
    if logged_in:
        initial_earning = 0
        with psycopg2.connect(**config()) as earning:
            curl = earning.cursor()
            curl.execute("""
                SELECT username FROM employee;
            """)
            list_of_employee = curl.fetchall()
            curl.close()
            earned_from_employee = 0
            for row in list_of_employee:
                earned_from_employee +=

```

```

int(employee_salary(row[0],True)[1])
    initial_earning += earned_from_employee
    cur2 = earning.cursor()
    cur2.execute("""
        SELECT username FROM patient;
    """)
    list_of_patient = cur2.fetchall()
    cur2.close()
    earned_from_patient = 0
    for row in list_of_patient:
        earned_from_patient +=
int(patient_cost(row[0],True)[1])
    initial_earning += earned_from_patient
    initial_earning -=
int(constant.grab_constant(True,"FIXED_COST_OF_HOSPITAL"))

    given_to_employees = 0
    cur3 = earning.cursor()
    cur3.execute("""
        SELECT salary FROM employee WHERE
work_of_doctors LIKE '%admin%';
    """)
    salary_list = cur3.fetchall()
    for salary in salary_list:
        given_to_employees += int(salary[0])
    cur3.close()
    initial_earning -= given_to_employees

    employee_pre_salary = int(doctor_salary("hpt",
True)[1])
    initial_earning -= employee_pre_salary

    earning.close()
    return (earned_from_employee, earned_from_patient,
int(constant.grab_constant(True,"FIXED_COST_OF_HOSPITAL")),
employee_pre_salary, given_to_employees, initial_earning)

def add_employee(employee_username, logged_in):
    if logged_in:
        with psycopg2.connect(**config()) as adding_employee:
            cur0 = adding_employee.cursor()
            cur0.execute("""
                SELECT work_of_doctors FROM employee WHERE
username = %s;
            """, [employee_username])

            work_of_doctors = cur0.fetchall()[0][0]
            work_of_doctors += ", " + "admin"
            cur0.close()

            cur1 = adding_employee.cursor()

```

```

        curl.execute("""
            UPDATE employee SET work_of_doctors = %s WHERE
username = %s;
        """, [work_of_doctors, employee_username])
        print(curl.statusmessage)
        curl.close()
        adding_employee.commit()
    adding_employee.close()
    return "Employee Added"

def see_my_employee(logged_in):
    if logged_in:
        with psycopg2.connect(**config()) as my_employee:
            curl = my_employee.cursor()
            curl.execute("""
                SELECT username, fullname, email, date_of_birth,
work, salary FROM employee WHERE work_of_doctors LIKE '%admin%'
            """)
            list_of_employees = curl.fetchall()
            # print part
            # for row in list_of_employees:
            #     print("Username:", row[0], "\tFullname:",
row[1], "\tEmail:", row[2], "\tWork:", row[3], "\tSalary:",
row[4])

            curl.close()
            my_employee.close()
            return list_of_employees

def remove_employee(employee_username, logged_in):
    if logged_in:
        with psycopg2.connect(**config()) as removing_employee:
            cur0 = removing_employee.cursor()
            cur0.execute("""
                SELECT work_of_doctors FROM employee WHERE
username = %s;
            """, [employee_username])

            work_of_doctors_string = cur0.fetchall()[0][0]
            work_of_doctors_list =
work_of_doctors_string.split(", ")

            work_of_doctors_list.remove("admin")
            new_work_of_doctors_string = ",
".join(work_of_doctors_list)
            cur0.close()

            curl = removing_employee.cursor()
            curl.execute("""
                UPDATE employee SET work_of_doctors = %s WHERE
username = %s;
            """, [new_work_of_doctors_string, employee_username])
            removing_employee.commit()
            curl.close()
            removing_employee.close()
            return "Employee Removed"

```

```

        """', [new_work_of_doctors_string,
employee_username])
        print(curl.statusmessage)
        curl.close()
        removing_employee.commit()
removing_employee.close()
return "Employee Removed"

def show_all_doctor(logged_in):
    if logged_in:
        with psycopg2.connect(**config()) as doctor:
            curl = doctor.cursor()
            curl.execute("""
                SELECT username, fullname, email, date_of_birth,
specialty, price FROM doctor;
            """)
            rows = curl.fetchall()
            new_rows = []
            for row in rows:
                row = list(row)
                row.append(doctor_salary(row[0], True) [2])
                new_rows.append(row)
            # print(part)
            # for row in rows:
            #     print("Username:", row[0], "\tFullname:",
row[1], "\tEmail:", row[2], "\tspecialty:", row[3], "\tPrice:",
row[4], "\tTotal Earned:", row[5])
            curl.close()
            doctor.close()
            return new_rows

def show_all_patient(logged_in):
    if logged_in:
        with psycopg2.connect(**config()) as patient:
            curl = patient.cursor()
            curl.execute("""
                SELECT username, fullname, email, date_of_birth,
problem, requested_doctor_username, approved_doctor_username
FROM patient;
            """)
            rows = curl.fetchall()
            new_rows = []
            for row in rows:
                row = list(row)
                row.append(patient_cost(row[0], True) [2])
                new_rows.append(row)
            # print part
            # for row in rows:
            #     print("Username:", row[0], "\tFullname:",
row[1], "\tEmail:", row[2], "\tProblem:", row[3],

```

```

"\tRequested:", row[4], "\tApproved:", row[5], "\tTotal Cost:",
row[6])
        curl.close()
        patient.close()
        return new_rows

def show_all_employee(logged_in):
    if logged_in:
        with psycopg2.connect(**config()) as employee:
            curl = employee.cursor()
            curl.execute("""
                SELECT username, fullname, email, date_of_birth,
work, work_of_doctors, salary FROM employee;
            """)
            rows = curl.fetchall()
            new_rows = []
            for row in rows:
                row = list(row)
                row.append(employee_salary(row[0], True) [2])
                new_rows.append(row)
            # print part
            # for row in rows:
            #     print("Username:", row[0], "\tFullname:",
row[1], "\tEmail:", row[2], "\tWork:", row[3], "\tDoctors:",
row[4], "\tSalary:", row[5], "\tTotal Earned:", row[6])
            curl.close()
            employee.close()
            return new_rows

def all_doctor_username(logged_in):
    if logged_in:
        with psycopg2.connect(**config()) as doctor:
            curl = doctor.cursor()
            curl.execute("""
                SELECT username FROM doctor;
            """)
            rows = curl.fetchall()
            new_rows = []
            for row in rows:
                new_rows.append(row[0])
            curl.close()
            doctor.close()
            return new_rows

def all_patient_username(logged_in):
    if logged_in:
        with psycopg2.connect(**config()) as patient:
            curl = patient.cursor()
            curl.execute("""

```

```

        SELECT username FROM patient;
        """
    rows = curl.fetchall()
    new_rows = []
    for row in rows:
        new_rows.append(row[0])
    curl.close()
    patient.close()
    return new_rows

def all_employee_username(logged_in):
    if logged_in:
        with psycopg2.connect(**config()) as employee:
            curl = employee.cursor()
            curl.execute("""
                SELECT username FROM employee;
                """)
            rows = curl.fetchall()
            new_rows = []
            for row in rows:
                new_rows.append(row[0])
            curl.close()
        employee.close()
        return new_rows

def remove_doctor_permanently(doctor_username, logged_in,
    final_decision = False):
    if final_decision and logged_in:
        with psycopg2.connect(**config()) as remove_doctor:
            curl = remove_doctor.cursor()
            curl.execute("""
                DELETE FROM doctor WHERE username = %s;
                """, [doctor_username])
            print(curl.statusmessage)
            remove_doctor.commit()
            curl.close()
        remove_doctor.close()
        return "Doctor Permanently Removed"

def remove_patient_permanently(patient_username, logged_in,
    final_decision = False):
    if final_decision and logged_in:
        with psycopg2.connect(**config()) as remove_patient:
            curl = remove_patient.cursor()
            curl.execute("""
                DELETE FROM patient WHERE username = %s;
                """, [patient_username])
            print(curl.statusmessage)
            remove_patient.commit()

```

```

        curl.close()
        remove_patient.close()
        return "Patient Parmanently Removed"

def remove_employee_permanently(employee_username, logged_in,
final_decision = False):
    if final_decision and logged_in:
        with psycopg2.connect(**config()) as remove_employee:
            curl = remove_employee.cursor()
            curl.execute("""
                DELETE FROM employee WHERE username = %s;
            """, [employee_username])
            print(curl.statusmessage)
            remove_employee.commit()
            curl.close()
        remove_employee.close()
        return "Employee Parmanently Removed"

def patient_joins_doctor(logged_in, left = False):
    join_type = "LEFT JOIN" if left else "JOIN"
    if logged_in:
        with psycopg2.connect(**config()) as viewer:
            cur_first = viewer.cursor()
            cur_first.execute("""
                DROP VIEW IF EXISTS final_view;
            """)
            cur_first.close()
            viewer.commit()

            cur0 = viewer.cursor()
            cur0.execute("""
                DROP VIEW IF EXISTS patient_view;
            """)
            cur0.close()
            viewer.commit()

            curl = viewer.cursor()
            curl.execute("""
                CREATE VIEW patient_view AS SELECT username,
                fullname, email, date_of_birth, problem,
                approved_doctor_username, appointment_timestamp FROM patient
                WHERE approved_doctor_username != 'hpt';
            """)
            curl.close()

            cur2 = viewer.cursor()
            cur2.execute("""
                DROP VIEW IF EXISTS doctor_view;
            """)
            cur2.close()

```

```

viewer.commit()

cur3 = viewer.cursor()
cur3.execute("""
    CREATE VIEW doctor_view AS SELECT username AS
doctor_username, fullname AS doctor_fullname, email AS
doctor_email, specialty, price FROM doctor;
    """)
cur3.close()

viewer.commit()
viewer.close()

with psycopg2.connect(**config()) as joiner:

    cur1 = joiner.cursor()
    cur1.execute(f"""
        CREATE VIEW final_view AS (SELECT * FROM
patient_view {join_type} doctor_view ON
patient_view.approved_doctor_username =
doctor_view.doctor_username);
        """)
    cur1.close()
    joiner.commit()
    joiner.close()

with psycopg2.connect(**config()) as final:
    cur1 = final.cursor()
    cur1.execute("""
        SELECT username, fullname, email, problem,
appointment_timestamp, doctor_username, doctor_fullname,
doctor_email, specialty, price FROM final_view;
        """)
    rows = cur1.fetchall()
    cur1.close()

    # print part
    # for row in rows:
    #     print(row)

joiner.close()
return rows

```