

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

Інститут інформатики та радіоелектроніки
Факультет комп'ютерних наук та технологій
Кафедра комп'ютерних систем та мереж

Пояснювальна записка

до дипломного проекту (роботи)

магістра

(ступінь вищої освіти (освітній ступінь))

на тему СИСТЕМА КЕРУВАННЯ ОБСЛУГОВУВАННЯМ ПАЦІЄНТІВ
МЕДИЧНОГО ЗАКЛАДУ

Виконав: студент 2 курсу, групи КНТ-511м
спеціальності _____

123 «Комп'ютерна інженерія»

(код і назва спеціальності)

Демченко Андрій Сергійович

(прізвище та ініціали)

Керівник Ільяшенко М.Б.

(прізвище та ініціали)

Рецензент Самойлик С.С.

(прізвище та ініціали)

м. Запоріжжя
2022 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування вищого навчального закладу)

Інститут, факультет інформатики та радіоелектроніки, комп'ютерних наук і технологій
 Кафедра «Комп'ютерні системи та мережі»
 Ступінь вищої освіти (освітній ступінь) магістерський
 Спеціальність 123 Комп'ютерна інженерія
 (код і назва)
 Освітня програма (спеціалізація) Комп'ютерні системи та мережі
 (назва)

ЗАТВЕРДЖУЮ

Зав. кафедри Кудерметов Р.К.

“ ” 2022 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Демченко Андрій Сергійович

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Система керування обслуговуванням пацієнтів медичного закладу

керівник проекту (роботи) Ільяшенко Матвій Борисович, к. т. н., доцент

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

затверджені наказом вищого навчального закладу від “07” листопада 2022 року № 367

2. Строк подання студентом проекту (роботи) 01 грудня 2022 року

3. Вихідні дані до проекту (роботи) Медичний заклад, база даних mysql

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1) Аналіз моделей та алгоритмів обслуговування в медичній установі

2) Модель управління потоками пацієнтів у черзі на обслуговування;

3) Проектування модуля розкладу прийому пацієнтів.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Плакати 1 – Архітектура системи;

Плакати 2 – Діаграма класів;

Плакати 3 – Фізична модель бази даних;

Плакати 4 – Діаграма розгортання системи

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-3	Ільяшенко М.Б., к. т. н., доцент		
Нормоконтроль	Щербак Н.В., ст. викл.		

7. Дата видачі завдання 10.10.2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз технічного завдання	10.10.2022 р.	
2	Модель управління потоками пацієнтів у черзі на обслуговування.	15.10.2022 р.	
3	Розробка засобів управління розкладом прийому пацієнтів	20.10.2022 р.	
4	Проектування бази даних	01.10.2022 р.	
5	Побудова логічної моделі бази даних	25.11.2022 р.	
6	Проектування програмних засобів	01.12.2022 р.	
7	Захист роботи	16.12.2022 р.	

Студент

(підпис)

А.С. Демченко

(ініціали та прізвище)

Керівник проекту (роботи)

(підпис)

М.Б. Ільяшенко

(ініціали та прізвище)

РЕФЕРАТ

ПЗ: 68 с., 18 рис., 18 табл., 21 джерело.

МЕДИЧНА УСТНОВА, МОДЕЛЬ, МЕРЕЖІ ПЕТРІ, ПАЦІЄНТИ, СИСТЕМА

Метою магістерської роботи розробка моделі управління обслуговуванням пацієнтів у медичній установі та розробка алгоритму розподілу потоків пацієнтів, що дозволяють підвищити ефективність та якість обслуговування пацієнтів у медичних закладах.

Відповідно до поставленої мети виділено такі завдання роботи:

- – провести аналіз предметної галузі, зокрема, процесів обслуговування пацієнтів у медичній установі;
- – провести аналіз відомих моделей масового обслуговування медичного закладу для визначення методів контролю потоків пацієнтів на прийомі у лікаря;
- – провести аналіз відомих алгоритмів формування черг заявок на обслуговування;
- – розробити модель розподілу потоків пацієнтів у черзі у медичній установі;
- – розробити алгоритм формування потоків пацієнтів у черзі на прийом у спеціаліста медичного закладу;
- – розробити модуль медичної інформаційної системи щодо формування розкладу прийомів пацієнтів.

Об'єктом дослідження є процес обслуговування пацієнтів у медичній організації.

Предметом дослідження є моделі та алгоритми управління обслуговуванням пацієнтів медичного закладу.

ЗМІСТ

Вступ.....	6
1 Аналіз моделей та алгоритмів обслуговування в медичній установі.....	9
1.1 Порядок прийому пацієнта у медичній установі	9
1.2 Аналіз допоміжних засобів роботи медичного закладу	12
1.3 Огляд готових рішень.....	14
1.4 Аналіз поліклініки як системи масового обслуговування.....	16
1.5 Аналіз алгоритмів формування черг у багатоканальній системі масового обслуговування.....	20
2 Модель управління потоками пацієнтів у черзі на обслуговування....	25
2.1 Опис моделі потоків пацієнтів у поліклініці.....	25
2.2 Алгоритм формування потоків пацієнтів для прийому у спеціаліста у медичній установі.....	33
3 Проектування модуля розкладу прийому пацієнтів.....	41
3.1 Розробка засобів управління розкладом прийому пацієнтів	41
3.2 Проектування бази даних.....	46
3.3 Побудова логічної моделі бази даних.....	51
3.4 Побудова фізичної моделі бази даних	53
3.5 Розробка збережених процедур, функцій та тригерів бази даних	54
3.6 Проектування програмних засобів.....	56
3.7 Тестування програми.....	62
Висновки	65
Перелік джерел посилання.....	66
Перелік графічних матеріалів:	
Плакат 1 – Діаграма варіантів використання;	
Плакат 2 – Інтерфейс головної сторінки додатка;	
Плакат 3 – Фізична модель бази даних;	
Плакат 4 – Діаграма розгортання системи.	

ВСТУП

В даний час медицина є однією з найважливіших і найскладніших галузей людської діяльності, задоволення потреб населення в якісній медицині є найважливішим завданням в галузі охорони здоров'я. Однією з цілей охорони здоров'я є підвищення ефективності роботи державних медичних установ, не тільки за рахунок оптимізації лікувальних та діагностичних процесів, а й за рахунок удосконалення інформаційних технологій у галузі медицини та модифікації процесів взаємодії пацієнтів, співробітників та спеціалістів медичного закладу.

На даний момент найбільш важливими напрямками реорганізації в галузі охорони здоров'я є:

- підвищення ефективності роботи лікарів;
- організація управління обслуговуванням пацієнтів у лікувальному закладі;
- прискорення доступу до довідкової інформації медичного закладу;
- підвищення рівня достовірності інформації при формуванні медичної документації;
- оптимізація інформаційної взаємодії між співробітниками та фахівцями медичного закладу.

Вирішення цих проблем забезпечить зниження кількості черг на прийом до спеціаліста медичного закладу, зниження часу очікування пацієнтів у черзі, збільшить показники якості обслуговування пацієнтів, а також надасть можливість швидкої та простої організації взаємин пацієнта з медичною організацією.

Сучасна медицина вже досягла значних успіхів у діагностиці та лікуванні багатьох захворювань, за рахунок використання більш складних інструментальних засобів та методів. Водночас відбувається поступова стандартизація багатьох процесів у галузі надання медичної допомоги. Складність медицини як галузі людської діяльності відображається у функціонуванні медичних установ, а отже, показує проблематику розробки, впровадження та використання медичних інформаційних систем.

Останнім часом інформаційні технології все частіше використовуються в медицині у вигляді системного аналізу та підтримки прийняття рішень. До переваг даного підходу варто віднести можливість створення різних сценаріїв функціонування системи, а також їх дослідження з метою оптимізації процесу управління обслуговуванням пацієнтів.

Існує безліч досліджень процесів обслуговування пацієнтів у поліклініці чи стаціонарі. У всіх дослідженнях медичні установи представлені як багатоканальна система масового обслуговування (СМО) з очікуванням і необмеженою чергою, але в різних моделях виділяються різноманітні найважливіші характеристики СМО, що описують стан системи. Прикладами таких характеристик є пріоритети заявок обслуговування чи кількість пацієнтів у черзі. Однак усі проведені дослідження не враховують можливість контролю навантажень на канали обслуговування та розподілу потоків заявок у черзі, оскільки навантаження спеціаліста медичної установи так само суттєво впливає на очікування пацієнтів у черзі. Крім вузькоспрямованих розробок існують узагальнені дослідження складних моделей СМО, які є комбінованими зі змінами режимів обслуговування. Для процесу розподілу потоків заявок необхідно здійснити огляд існуючих алгоритмів розподілу заявок у чергах. Існуючі дослідження алгоритмів прийому заявок обслуговування у чергах також прив'язані до конкретної предметної області.

Метою магістерської роботи розробка моделі управління обслуговуванням пацієнтів у медичній установі та розробка алгоритму розподілу потоків пацієнтів, що дозволяють підвищити ефективність та якість обслуговування пацієнтів у медичних закладах.

Відповідно до поставленої мети виділено такі завдання роботи:

- провести аналіз предметної галузі, зокрема, процесів обслуговування пацієнтів у медичній установі;
- провести аналіз відомих моделей масового обслуговування медичного закладу для визначення методів контролю потоків пацієнтів на прийомі у лікаря;

- провести аналіз відомих алгоритмів формування черг заявок на обслуговування;
- розробити модель розподілу потоків пацієнтів у черзі у медичній установі;
- розробити алгоритм формування потоків пацієнтів у черзі на прийом у спеціаліста медичного закладу;
- розробити модуль медичної інформаційної системи щодо формування розкладу прийомів пацієнтів.

1 АНАЛІЗ МОДЕЛЕЙ ТА АЛГОРИТМІВ ОБСЛУГОВУВАННЯ В МЕДИЧНІЙ УСТАНОВІ

1.1 Порядок прийому пацієнта у медичній установі

Одним із головних завдань охорони здоров'я нині є підвищення якості обслуговування пацієнтів у державних медичних закладах, зокрема у поліклініках. Тривале очікування у черзі знижує доступність амбулаторно-поліклінічної допомоги та викликає невдоволення населення. Найбільш гостро ця проблема виражена у наданні вузькоспеціалізованої допомоги.

Поліклінікою прийнято називати багатoproфільний лікувально-профілактичний заклад, де надається медична допомога пацієнтам, закріпленим за ним за місцем проживання на догоспітальному етапі.

Кількість відділень та кабінетів лікарів визначаються потужністю поліклініки та кількістю штатних посад, ці фактори, у свою чергу, залежать від чисельності населення, закріпленого на цій території. Структура поліклініки також залежить від потреб населення в медичній допомозі та здатності самої поліклініки надати цю допомогу.

Таким чином, до основних функцій поліклініки відносять:

- надання першої медичної допомоги пацієнтам вдома;
- надання медичної допомоги людям із гострими захворюваннями, травмами, отруєннями та інших невідкладних станах незалежно від місця проживання хворого;
- госпіталізація хворого, який потребує стаціонарного лікування;
- визначення тимчасової непрацездатності, звільнення пацієнта з роботи, видача напрямів на медичні експертизи людям із ознаками втрати працездатності;
- організація набору профілактичних заходів, спрямованих на зниження захворюваності, інвалідності та смертності населення, що проживає у певному районі;

- організація медичного огляду та диспансеризації населення;
- направлення пацієнтів на санітарно-курортне лікування;
- організація та проведення заходів, спрямованих на санітарно-гігієнічне виховання населення та пропаганда здорового способу життя [1].

Робота поліклініки побудована за територіальним принципом – до неї, як правило, прикріплені лише ті пацієнти, які прописані у тому ж районі, де розташована поліклініка. Головною структурною одиницею є територіальна терапевтична ділянка. Населення дільницями розподіляється керівниками медичного закладу.

Для контролю стану здоров'я населення у поліклініках проводиться диспансеризація. Диспансеризація спрямована на збереження та швидке відновлення здоров'я населення та зниження захворюваності.

Профільні фахівці проводять роботу з виявлення захворювань на різних стадіях, здійснюють систематичне спостереження та своєчасне лікування пацієнта. Крім того, лікар повинен попередити виникнення та розповсюдження спалахів хвороб. Для кожного спеціаліста встановлюється свій перелік пацієнтів, які підлягають диспансеризації та медичному огляду.

Дільничний лікар у поліклініці (як правило, він є терапевтом) веде прийом хворих як вдома, так і в кабінеті, залежно від стану здоров'я хворого, тому має гнучкий графік.

Під час аналізу процесів обслуговування пацієнта лікарем існують різні алгоритми. Порядок прийому пацієнта у медичному закладі багато в чому залежить від його типу. У поліклініках існує первинний та повторний прийом. Крім того, порядок обстеження пацієнта залежить від спеціалізації лікаря (лікар загальної практики або профільний спеціаліст) та причин звернення.

Якщо пацієнт звертається до медичного закладу вперше, такий прийом є первинним. Пацієнт може звернутися до фахівця або лікаря загальної практики щодо гострого захворювання або загострення хронічного захворювання. При зверненні до лікаря загальної практики проводиться первинний огляд пацієнта. Лікар наголошує на озвученій проблемі, ставить попередній діагноз і складає план

лікування, призначає додаткову діагностику або дає направлення до профільного фахівця. Крім того, залежно від проблеми, лікар може встановити прогноз лікування та його обсяг, інформувати пацієнта про можливі ускладнення. Порядок та результати огляду фіксуються в електронній карті пацієнта [2].

Після того, як пацієнт отримав план лікування, пройшов це лікування, він може знову звернутися до лікаря через деякий час. У такому разі прийом класифікується як повторний. При повторному прийомі лікар може знову оглянути пацієнта, оцінити його стан, призначити новий план лікування та додаткову діагностику. Усі дані, отримані під час огляду та діагностики, потрапляють у медичну інформаційну систему (МІС) і передаються з її допомогою до різних структур поліклініки для прискорення процесу надання допомоги.

На даний момент пацієнтів у поліклініці можна класифікувати за різними критеріями. Одним із найважливіших критеріїв є причина звернення до лікаря.

Як правило, в поліклініку пацієнти звертаються з однієї з таких причин: консультація, медичний огляд, лікування або оформлення лікарняного листа. Час, який потрібен лікареві на роботу з кожним із пацієнтів, відповідно різний і залежить від причини звернення.

Крім цього, пацієнт може прийти за попереднім або без запису. Запис найчастіше відбувається через телефонний дзвінок до реєстратури. За попереднім записом зазвичай приходять пацієнти з конкретною проблемою, які зазвичай проводять у кабінеті більше часу, ніж інші. Пацієнти без запису найчастіше приходять за лікарняним листом або за консультацією, у таких випадках лікарю не потрібно багато часу на прийом [3].

На даний момент основною проблемою поліклінік є великі потоки пацієнтів. На зростання черг до фахівців впливає низка факторів. Всі ці фактори є взаємозалежними, одними з найважливіших є високе навантаження на лікарів та їх кількість. Через часткову нестачу фахівців навантаження перекладається на лікарів, які перебувають у штаті поліклініки, що ускладнює їхню роботу та впливає на зростання черг. Донедавна на час обслуговування одного пацієнта впливала також необхідність заповнення фахівцем зайвої документації, яку можна оцифрувати і

зберігати в єдиній базі. Ця проблема вже вирішена за рахунок створення медичних інформаційних систем (МІС). Дані системи справді скоротили час роботи лікаря з одним пацієнтом, проте потоки черг скоротилися ненабагато.

Іншим фактором, що впливає на тривалість прийому, є процес огляду пацієнта. Оскільки пацієнти звертаються з різними цілями (необхідність лікування, консультація, отримання довідкової інформації, отримання лікарняного листа), і обслуговування кожного пацієнта йде різний час. Крім того, часто пацієнт звертається з проблемами, які не входять до компетенції конкретного лікаря. Таким чином лікар може дати пацієнтові направлення до профільного фахівця, який і проводитиме відповідну діагностику. Це означає, що пацієнтові доведеться звернутися до призначеного лікаря, на прийом якого також з'являється черга. Тобто необхідна додаткова організація прийому у профільних фахівців [4].

1.2 Аналіз допоміжних засобів роботи медичного закладу

Збереженням та обліком усієї медичної інформації про пацієнта займається реєстратура. Донедавна вся інформація велася у паперовому вигляді, проте тепер усі дані зберігає інформаційна система. Медичні інформаційні системи виробляють облік, зберігання та оформлення не тільки карт хворого, але й інших медичних документів, регулюють навантаження на лікарів за допомогою талонної системи та самозапису пацієнтів.

Таким чином, медичні інформаційні системи (МІС) у поліклініках забезпечують:

- інформаційну підтримку надання медичної допомоги (ведення медичних карт пацієнта та контроль медико-технологічних процесів);
- інформаційну підтримку процесу управління медичної організації (управління адміністративною та господарською діяльностями формування передачі даних про технічне лікарське забезпечення та витрати на них);

- інформаційну підтримку процесів запису пацієнтів на прийом;
- інформаційної взаємодії між кількома медичними установами в рамках надання медичної допомоги (напрямок пацієнтів до інших медичних закладів для проведення докладної діагностики та отримання медичної допомоги);
- інформаційної взаємодії з іншими регіональними та федеральними установами в частині обміну інформацією, пов'язаною з лікувально-діагностичним процесом.

Медичні інформаційні системи створені та впроваджені з такими цілями:

- підвищення якості та доступності медичної допомоги
- зниження витрат на надання медичної допомоги при збереженні чи підвищенні рівня результату;
- підвищення ефективності роботи медичного закладу;
- підтримка прийняття лікарських рішень;
- створення бази даних науково-дослідної роботи

Медичні інформаційні системи зберігають великі обсяги різних груп інформації, які можна розділити на три великі класи:

- –структурована інформація прості типи (числа, рядки, дати);
- –слабко структурована інформація;
- – неструктурована інформація (в основному це графічні файли або документи з різних офісних програм, з їх вмістом МІС безпосередньо не працює).

Джерелами цієї інформації, як правило, є:

- різні медичні записи, що створюються у процесі надання допомоги;
- дані, лічені медичною технікою;
- дані, передані із зовнішніх інформаційних систем.

Медичні інформаційні системи повинні взаємодіяти з іншими зовнішніми ресурсами чи суміжними інформаційними системами. До них належать системи управління ресурсами організації, бухгалтерського обліку, фінансово-

економічного планування, управління проектами, федеральними та регіональними медичними сервісами. Іншими словами, структура будь-якої МІС є взаємопов'язаними модулями, кожен з яких відповідає за певні аспекти роботи медичної організації.

Таким чином, основною функцією медичних інформаційних систем є збирання медичної та адміністративної інформації. Ця функція спрямована на прискорення роботи медичної організації, в тому числа, скорочення потоку пацієнтів за рахунок скорочення часу наповнення медичної документації [5].

1.3 Огляд готових рішень

Для детального аналізу можливостей медичних інформаційних систем у частині процесу обслуговування пацієнтів буде розглянуто два готові рішення. Слід зазначити, що кожна з МВС створена розробниками, а отже, повинна задовольняти вимоги до розробки МІС, встановленими державними стандартами, частина яких описана в пункті 1.2. Також усі представлені системи є продуктами вітчизняних розробників.

МІС «Цифровий пацієнт» – це комплексна МІС для поліклініки, стаціонару та мережі медичних установ, що включає різні модулі для працівників медичного закладу, модулі складання звітів з діагнозів, звіти з проведених операцій.

МІС «Цифровий пацієнт» має такі модулі:

- автоматизоване робоче місце завідувача відділення;
- автоматизоване робоче місце лікаря;
- автоматизоване місце медсестри.

Розглянута МІС дозволяє користувачам здійснювати такі набори функцій:

- ведення історії хвороби пацієнтів;
- оптимізація планування лікувальних процесів;

- управління фінансовими ресурсами організації;
- зчитування результатів діагностики з лікувальних приладів;
- планування розміщення пацієнтів у стаціонарі та санаторії;
- встановлення прав доступу користувачів до ресурсів системи.

Архітектура системи «Цифровий пацієнт» клієнт-серверна, в серверній частині встановлена база даних на основі СУБД Microsoft SQL Server, клієнтські програми працюють на базі операційних систем Windows 2000, Windows XP, Windows Vista, Windows 7 і т.д. залежно від кількості необхідних автоматизованих робочих місць (доступно понад 500 місць).

Медична інформаційна система «Medods» є системою нового покоління, застосовується і в приватних клініках, і державних.

Головною перевагою «Medods» є простота її інтерфейсу, що не вимагає тимчасових витрат за навчання користувачів. Ще одна перевага цієї системи порівняно з іншими розглянутими – низька вартість програмного забезпечення.

МІС містить досить поширений та корисний функціонал:

- запис пацієнтів на прийом онлайн;
- повідомлення пацієнтів за SMS або електронною поштою;
- ведення медичної електронної картки;
- створення звітів;
- ведення складського обліку;
- інтеграція з медичною технікою;
- інтеграція з телефонією;
- інтеграція з онлайн касами.

Проведений аналіз існуючих медичних інформаційних систем показав, що описані програми мають певні переваги та недоліки. До переваг систем варто віднести їх великий функціонал, що дозволяє мінімізувати тимчасові витрати на оформлення медичної документації і скоротити час огляду одного пацієнта через відсутність документів, які потрібно заповнювати вручну [6].

Недоліком представлених систем є відсутність функціоналу, спрямованого на ведення розкладів прийому пацієнтів для кожного з лікарів, що допомогло б контролювати потоки пацієнтів за рахунок їх розподілу за часовими інтервалами в залежності від поставлених проблем кожного з пацієнтів та навантаження лікаря.

1.4 Аналіз поліклініки як системи масового обслуговування

Поліклініка є організацією, в якій процеси обслуговування, моменти приходу нових пацієнтів, інтенсивність потоку пацієнтів та час їх обслуговування відбуваються у випадкові моменти часу, що можна описати імовірнісними функціями. Такі системи можна уявити як математичні моделі, які називаються системами масового обслуговування. Будь-яка подібна система має кількість обслуговуючих приладів, вхідний і вихідний потік заявок на обслуговування у приладу. Також існує дисципліна обслуговування – набір певних умов, які має виконати заявка, у разі виконання заявка вважається виконаною і залишає систему у вхідному потоці. Будь-яка модель масового обслуговування має такі характеристики:

- вхідний потік заявок обслуговування (пацієнти, бажаючі записатися приймання);
- у разі моделі масового обслуговування з чергою існує буфер черги, який може бути обмежений чисельністю місць, або часу очікування в черзі, або не обмежений нічим;
- обслуговуючий пристрій (лікар);
- вихідний потік обслужених заявок (пацієнти, які завершили прийом лікаря).

Вважаємо, що середній час між надходженнями заявок відомий і дорівнює T . Отже, інтенсивність потоку заявок λ обернено пропорційна T , тобто

$$\lambda = \frac{1}{T} \quad (1.1)$$

Нехай середній час обслуговування заявки дорівнює $t_{обс}$ і є випадковою величиною, а тривалість обслуговування однієї заявки μ обернено пропорційна $t_{обс}$, тобто

$$\mu = \frac{1}{t_{обс}} \quad (1.2)$$

Відношення інтенсивності потоку заявок до тривалості обслуговування заявки позначено через ρ . Таким чином,

$$\rho = \frac{\lambda}{\mu} \quad (1.3)$$

Величина ρ є навантаженням на канал обслуговування, відповідно залежить від інтенсивності потоку заявок на обслуговування в каналі і часу обслуговування заявок. Цей параметр суттєво впливає на середній час очікування заявки у черзі [7].

Існує безліч досліджень систем масового обслуговування у межах поточної предметної області.

У [8] описана робота лікувального закладу як паралельно-послідовну систему масового обслуговування, в якій існують первинний потік пацієнтів і вторинний потік хворих. Лікар-терапевт займається розподілом пацієнтів, оформляючи напрямки до профільних спеціалістів. Напрями до того чи іншого спеціаліста здійснюється з певними ймовірностями. Тобто терапевт піддає первинний потік пацієнтів операції розрядження. Ця операція полягає в наступному: кожен пацієнт з ймовірністю p_i направляється до відповідного профільного кабінету, в інший профільний кабінет з ймовірністю $1 - p_i$. Одночасно

з описаним процесом відбувається підрахунок середнього часового інтервалу між пацієнтами первинного потоку, що направляються до профільних кабінетів. Крім первинного потоку пацієнтів, рухається вторинний потік хворих, які продовжують лікування. Потік хворих вирушає обслуговування до профільним спеціалістам, де поєднується з первинним потоком. Ця модель враховує кластеризацію первинного потоку пацієнтів, а також утворення вторинного потоку хворих, характерних для процесу обслуговування пацієнтів у медичних закладах. Також модель враховує час тривалості заявки та час обробки заявки.

У [9] медичний заклад представлено як сукупність одноканальних та багатоканальних СМО з чергою. Поліклініка розглянута на прикладі системи з кількома структурами: реєстратура, кабінети хірурга, терапевта та окуліста. У системі також було описано стаціонар, зокрема, відділення хірургії, офтальмології та терапії. Модель показує, що пацієнти, які залишили реєстратуру і стали в чергу на прийом до конкретного фахівця, мають певні показники пріоритетів. Значення пріоритету залежить від складності захворювання. Після обслуговування в кабінеті лікаря пацієнт або вирушає у вихідний потік обслужених пацієнтів, або стаціонар. У стаціонарі пацієнт обслуговується з тривалістю, яка залежить від складності захворювання, а після закінчення обслуговування відправляється у вихідний потік.

У [10] проведено дослідження медичного закладу як багатоканальної системи масового обслуговування з чергою з використанням нотації Кендалла, яка враховує такі фактори: час між надходженнями заявок у чергу, розподіл часу обслуговування та кількість обслуговуючих пристроїв. Стан системи визначається кількістю пацієнтів у черзі або на прийомі і ймовірністю того, що пацієнт, що надійде, стане в чергу.

В [11] проведено дослідження моделі роботи медичного закладу, взявши за основу роботу одного лікаря поліклініки змодельювавши її як одноканальну СМО з очікуванням. У роботі вирішується завдання ідентифікації вхідного потоку пацієнтів.

У роботі [12] розглядається система масового обслуговування поліклініки з циклічною дисципліною проходження заявок, тобто заявка залишить систему лише

тоді, коли буде обслужена у всіх каналах обслуговування. Усі можливі структури графа такий СМО описані через метод імовірнісних графів. Досліджувана у роботі система є багатоканальною з необмеженою чергою.

Крім вузькоспрямованих аналізів моделей СМО медичної організації існують аналізи більш узагальнених моделей СМО, але при цьому допускають зміни режимів обслуговування.

У [13] були вивчені дві моделі, відмінні їх особливості полягають у зміні числа заявок у системі та інтенсивності обслуговування. Нижче представлена модель з інтенсивністю потоку заявок, що змінюється. У системі існує l режимів надходжень заявок. Тривалість кожного режиму розподіляється за показовим законом і є незалежною випадковою величиною. На вхід до системи при i -му режимі надходить найпростіший потік заявок і встановлено кількість задіяних каналів обслуговування. Якщо кількість задіяних каналів у момент перемикавання режимів не змінюється, всі заявки, що обслуговуються, продовжують обслуговуватися у своїх каналах і довжина черги не зменшується. Якщо кількість каналів, що працюють, збільшується, то заявки, що стоять у чергах, надходять на обслуговування в нові порожні канали. Якщо кількість каналів зменшується, то заявки,

Всі описані моделі є багатоканальним СМО з необмеженою чергою. У проведених дослідженнях враховуються час надходження заявок, кількість заявок та пріоритет робіт, пов'язаних із обслуговуванням заявок. Однак розглянуті системи не враховують навантаження на канали обслуговування, оскільки у разі перевантаження канал буде нездатний обслужити наступні заявки, що прийшли в систему. У цьому випадку заявки можуть перебувати у черзі досить тривалий час, оскільки немає обмеження на час очікування у черзі, до того ж будуть надходити нові заявки, оскільки черга не обмежена. У такому разі необхідна розробка моделі системи масового обслуговування, в якій враховується ступінь навантаження на канали обслуговування для рівномірного розподілу заявок у чергах.

1.5 Аналіз алгоритмів формування черг у багатоканальній системі масового обслуговування

Питання оптимального розподілу потоків заявок у чергах вирішує теорія розкладів.

В теорії розкладів обслуговуючий пристрій (каналобслуговування), що виконує дії над заявкою прийнято називати машиною. Заявку в інтерпретації теорії черг прийнято називати роботою.

Для розгляду алгоритмів управління потоками заявок у медичній організації представлено дві множини:

- безліч машин $M = \{M_1, M_2, \dots, M_m\}$ (канали обслуговування);
- безліч робіт $J = \{J_1, J_2, \dots, J_n\}$ (потоки заявок обслуговування).

У системі функціонують кілька машин, які працюють паралельно.

Машинам не допускається виконувати одночасно дві роботи, також не допускається переривання операцій. Кожна i -та робота має тривалість p . Повинно бути враховано, що будь-яка робота може бути виконана на будь-якій із представлених машин (у поліклініці пацієнт може обстежитись у будь-якого лікаря конкретного профілю). Необхідно сформувати порядок прийому робіт обслуговування машинами з мінімальним часом завершення всіх робіт.

У межах поточної предметної області не проводилося досліджень алгоритмів формування черг. Однак є узагальнені алгоритми формування потоків заявок на обслуговування. Всі вони класифікуються між собою за різними значеннями власних показників. При дослідженні багатоканальних СМО варто розглянути алгоритми формування черг (або розкладу) для кількості машин $M \geq 2$. Далі представлені найбільш відомі алгоритми.

Для вирішення задач оптимального формування потоків заявок необхідно знайти нижню оцінку тривалості робіт L_B

$$LB = \max \left\{ \max p_i; \frac{1}{m} \sum_{i=1}^n p_i \right\} \quad (1.3)$$

де m - число машин;

n - число робіт.

Для формування порядку обслуговування робіт необхідно взяти список робіт та завантажити машини послідовно. Перша машина візьме i робіт та виконає їх в інтервалі часу $[0, LB]$, якщо остання робота не виконана до кінця на цьому інтервалі, то її залишки переносяться на наступну машину, яка виконає її на інтервалі $[0, LB]$.

Так як $LB \geq \max(p_i), i = 1 \dots n$, то робота виконується не більше ніж на одній машині. Тому цей розподіл є допустимим.

Для докладного розгляду взято випадок, коли тривалості робіт p_i різні, оскільки такий варіант системи є найбільш підходящим у межах роботи поліклініки. У такому випадку роботи з більшими тривалостями будуть ставитися на більш швидкі машини доти, доки тривалість останніх доданих у потік робіт не скоротиться настільки, що збігається з тривалістю інших робіт, прийнятих на обслуговування.

Нехай $p_i(t)$ є частиною роботи i , яка ще не виконана на момент часу t . Тоді алгоритм розподілу завдань буде рухатися, але в певний момент часу він зупиниться, щоб перерозподілити роботи по машинах. На рисунку 1.1 зображено блок-схему алгоритму розподілу потоків заявок у разі, коли час обслуговування різний для кожної із завдань.

На рисунку показано, що з робіт з ненульової тривалістю відбувається призначення робіт на певні машини. Процес розподілу робіт відмічено на схемі блоком «Призначення робіт з машин»

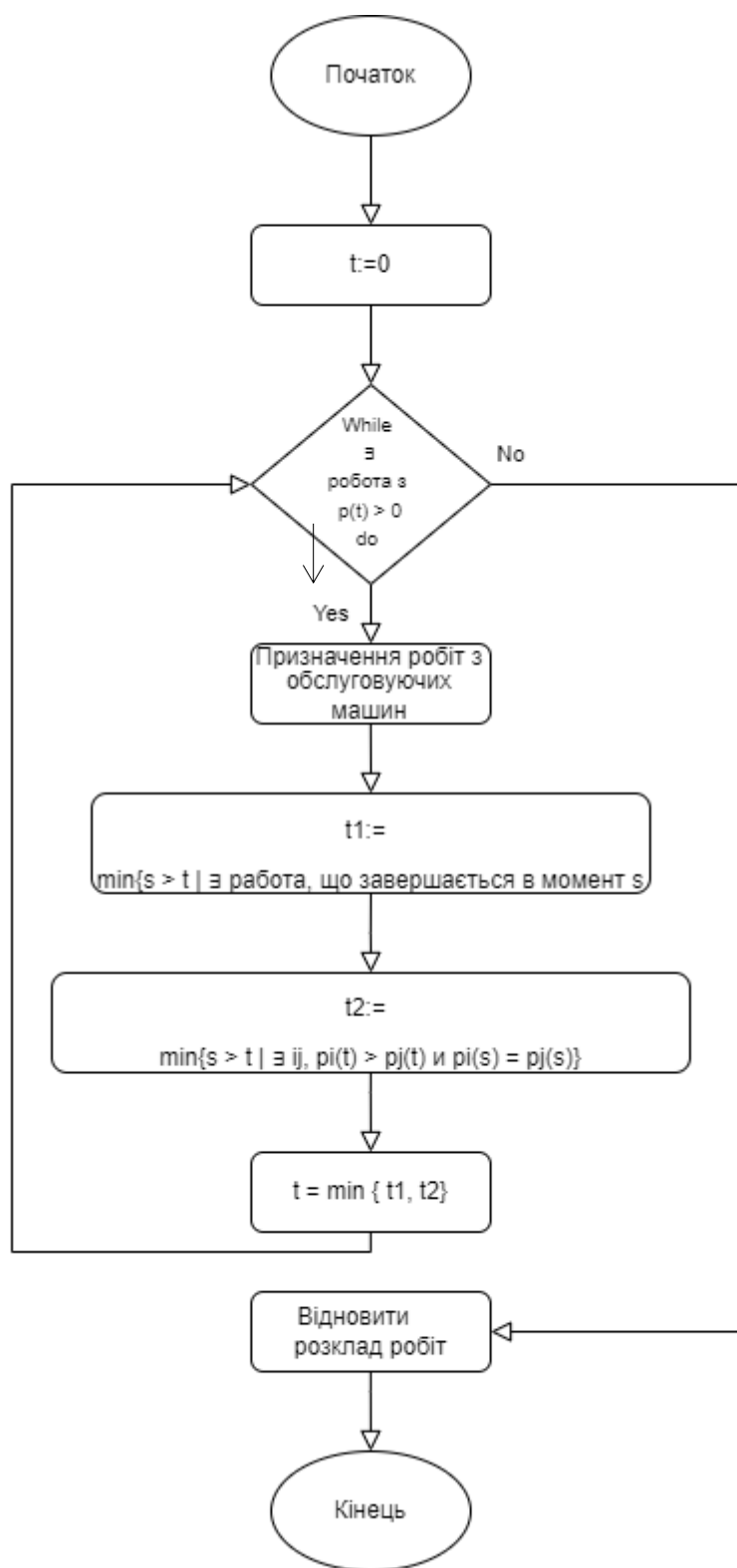


Рисунок 1.1 - Алгоритм формування потоків заявок

Алгоритм призначення робіт з машин (на рисунку 1.1 – «Виконати призначення робіт з машин») зображено на рисунку 1.2.

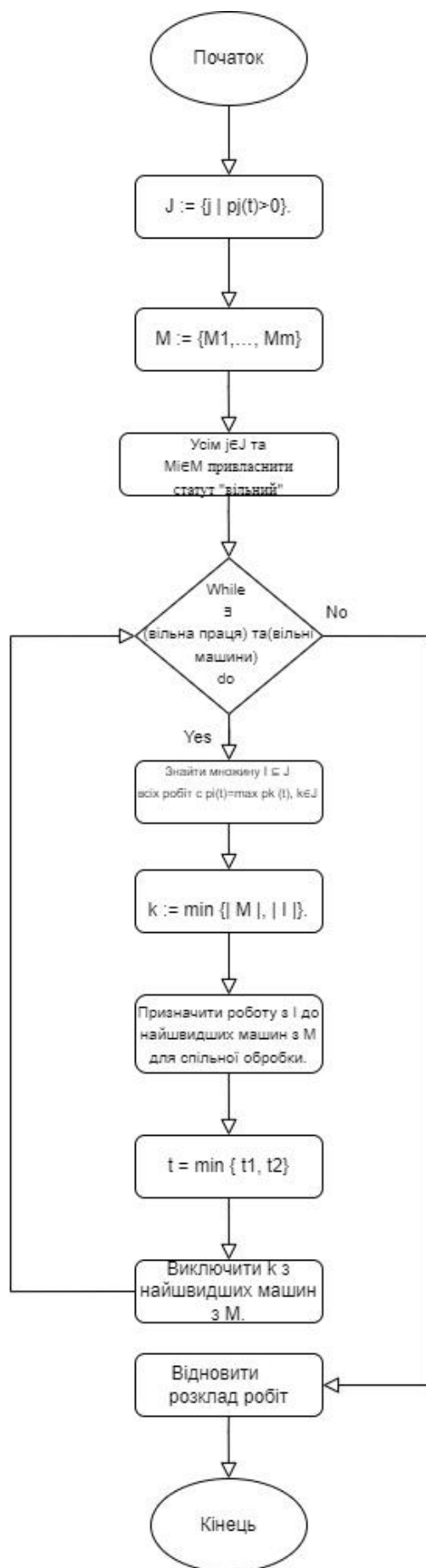


Рисунок 1.2 - Призначення робіт з машин

Недоліком розглянутих алгоритмів є необхідність розділяти одну роботу на підзавдання та перерозподілити їх для обслуговування машинами, що є неможливим до реалізації в поліклініці. Крім того, алгоритми дозволяють у конкретний момент часу перерозподілити всі роботи по машинах, що також нереалізовано, оскільки неможливо здійснити цей перерозподіл у реальному часі. Також проблемою реалізації даних алгоритмів практично в поліклініці є неможливість спільного прийому одного пацієнта двома лікарями. Тому необхідно розробити алгоритм розподілу вхідного потоку заявок на обслуговування з урахуванням виконання робіт на одній машині [14].

Аналіз предметної області та відповідних програмних засобів показав, що однією із проблем процесу обслуговування пацієнтів у медичній установі (зокрема, поліклініці) досі залишається неконтрольований потік пацієнтів, які утворюють чергу на прийом. Очікування в черзі найчастіше залежить від часу обстеження одного пацієнта та завантаженості лікаря. Ці проблеми вирішує впровадження інформаційних систем, які за рахунок невеликого зменшення часових витрат на оформлення лікарем документації. Для контролю потоків пацієнтів у поліклініці пропонується оптимізувати модель прийому нових пацієнтів.

Моделі, вивчені у пункті 1.4, не враховують можливості контролю за навантаженням.

Математичні алгоритми управління потоками пацієнтів, досліджені в пункті 1.5, припускають можливість розбиття роботи, перенесення частини роботи на іншу машину та множинного перерозподілу по функціонуючих машинах.

В умовах поліклініки такий підхід некоректний, оскільки пацієнт у кращому випадку має бути обслужений один раз одним лікарем, у гіршому, записаний прийом ще раз. Однак навіть за такого можливого сценарію вважається, що один прийом рівноцінний одній роботі і його не можна розбивати на підзавдання. Однак, розглянуті алгоритми при повному обслуговуванні роботи на одній машині дозволяють оптимально розподілити потік пацієнтів на прийом, що допоможе контролювати потік черг пацієнтів на прийом.

Для зручності моніторингу черг пацієнтів потрібна розробка допоміжного модуля медичної інформаційної системи, де зберігатиметься інформація про записи до того чи іншого лікаря, а також інформація про майбутні прийоми.

2 МОДЕЛЬ УПРАВЛІННЯ ПОТОКАМИ ПАЦІЄНТІВ У ЧЕРЗІ НА ОБСЛУГОВУВАННЯ

2.1 Опис моделі потоків пацієнтів у поліклініці

Канали обслуговування виконують роботу нерівномірно, тобто із різною інтенсивністю обслуговування, що є випадковою величиною. Тоді недоцільно відправляти заявки на цей канал, варто розподілити їх по більш вільних каналах. Крім того, модель СМО поліклініки має бути з обмеженням на довжину черг, але обмеження не буде фіксованим, а залежатиме від навантаження каналу обслуговування та змінюватися в момент часу t .

Зміну довжини черги у часі слід розглядати як режим обслуговування. З урахуванням високого навантаження на канал обслуговування можливу кількість місць у черзі m варто обмежити доти, доки значення навантаження на канал не впаде. Так як система масового обслуговування багатоканальна, і навантаження на канали розподілена нерівномірно, велика ймовірність того, що існує канал з нижчим навантаженням, у цьому випадку необхідно перенаправити нові заявки до порожнього каналу. У момент часу t останній стан системи фіксується і нові заявки, що прийшли в момент t , розподіляються в залежності від стану системи.

Для визначення максимального та мінімального навантаження між фахівцями введемо максимальне значення навантаження каналу $r_{\max}$ та мінімальне значення $r_{\min}$. Важливо відзначити, що навантаження, наприклад, лікаря-терапевта має бути порівнянним відповідно лише з навантаженнями інших лікарів терапевтом медичного закладу. Пошук максимального та мінімального

значення навантажень каналів обслуговування, що функціонують у системі, здійснюється шляхом сортування значень від мінімального до максимального. Отже, ступінь навантаження ρ на i -й канал перебуває в інтервалі:

$$\rho_{min} \leq \rho_i \leq \rho_{max} \quad (2.1)$$

Якщо навантаження ρ_i наближається до значення ρ_{min} , то допустима кількість місць у черзі може бути збільшена на величину l , інакше – зменшено через зменшення числа місць у черзі m . Таким чином, величина m є динамічною.

У разі запропонованої моделі масового обслуговування алгоритм руху заявки по системі виглядає так. У момент часу t на вхід надходить заявка, перед якою кілька черг та кілька каналів обслуговування. У разі високої зайнятості i -го каналу обслуговування черга обмежується, якщо місць у черзі немає, заявку переводять у чергу на інший канал обслуговування, якщо його навантаження нижче за навантаження i -го каналу. Якщо навантаження всіх каналів високі і місць в обмежених чергах немає, заявка залишає систему необслуженою до наступного моменту t_1 . Якщо навантаження каналів знижується, обмеження на чергу скасовується.

На рисунку 2.1 зображено граф моделі СМО із тимчасовим обмеженням на довжину черги медичного закладу.

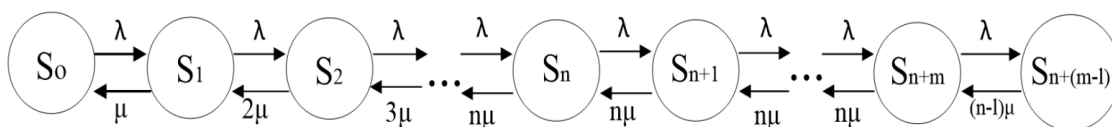


Рисунок 2.1 - СМО з тимчасовим обмеженням на довжину черги

На рисунку видно, що модель має такі стани:

- у системі 0 заявок;
- 1 заявка, черга порожня;
- 2 заявки, черга порожня;

- n заявок, черга порожня;
- n заявок на обслуговування, 1 заявка у черзі;
- n заявок на обслуговуванні, m заявок у черзі, черга обмежена та заповнена (при максимальному завантаженні каналу);
- n заявок на обслуговування, m заявок у черзі, обмеження на чергу знято (при зниженні навантаження каналу).

Таким чином, модель медичного закладу може бути комбінованою системою масового обслуговування зі зміною режиму черг.

Для подальшої перевірки оптимальності запропонованої моделі наведено детальний опис алгоритму обслуговування пацієнтів у системі масового обслуговування із зміною режимів черг. На вхід до системи надходить єдиний потік пацієнтів, які звертаються до реєстратури. Співробітник реєстратури обслуговує кожного пацієнта за допомогою опитування та направляє до того чи іншого лікаря, тим самим розподіляючи черги до фахівців. Якщо i -й лікар має високе навантаження, то черга до нього обмежена, і пацієнт прямує до j -го лікаря. Під час прийому в кабінеті лікаря пацієнт може бути перенаправлений до профільного фахівця (якщо прийом вівся у лікаря загальної практики) або на додаткову діагностику. Алгоритм у цьому випадку повторюється, пацієнт буде перенаправлений до фахівця з нижчим навантаженням. Пацієнти, які завершили обслуговування,

Одним із засобів ціни ефективності моделей обслуговування є апарат мереж Петрі. Щоб автоматизувати процес побудови мереж Петрі, використовується програмний пакет «Petri net emulator». Мережа Петрі, яка описує розглянуту модель, представлена рисунку 2.2. Мережею Петрі називається безліч вершин (P) і переходів (T), з-поміж них встановлюється зв'язок. У даному варіанті мережі Петрі підбезліч вершин мають на увазі можливі стани каналів обслуговування, переходи характеризують рух черг обслуговування. У даній мережі розглядається взаємодія трьох каналів обслуговування досліджуваної СМО. У цьому випадку канали обслуговування інтерпретуються як лікарі одного профілю поліклініки.

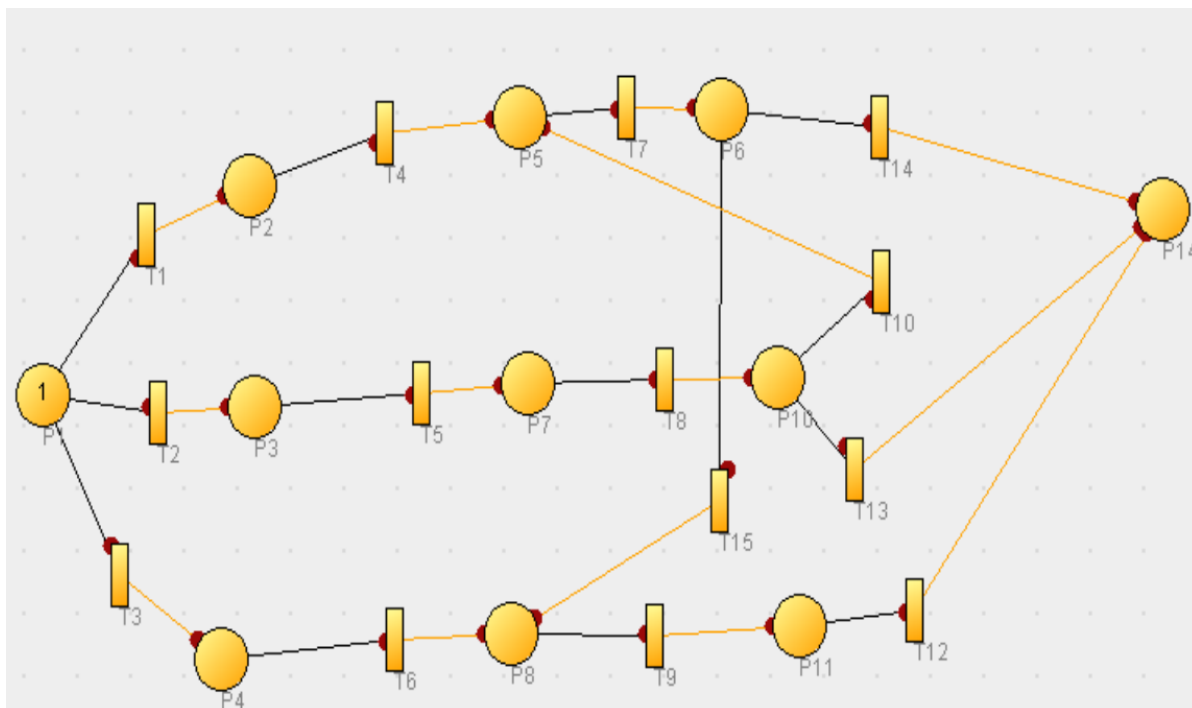


Рисунок 2.2 - Мережа Петрі, що імітує, модель масового обслуговування поліклініки.

Описи станів системи, представлені в мережі у вигляді вершин, наведені в таблиці 2.1

Таблиця 2.1 - Умови спрацювання переходів

Вершина	Умова спрацювання
P_1	Усі три канали порожні, у черзі немає заявок, початок обслуговування
P_2	У першому каналі одна заявка, черги немає
P_3	У другому каналі одна заявка, черги немає
P_4	У третьому каналі одна заявка, черги немає
P_5	У першому каналі одна заявка, у черзі m заявок, навантаження на канал низька
P_6	У першому каналі одна заявка, у черзі n заявок, навантаження на канал висока
P_7	У другому каналі одна заявка, у черзі m заявок, навантаження на канал низька
P_8	У третьому каналі одна заявка, у черзі m заявок, навантаження на канал низька
P_{10}	У другому каналі одна заявка, у черзі n заявок, навантаження на канал висока
P_{11}	У третьому каналі одна заявка, у черзі n заявок, навантаження на канал висока
P_{14}	Завершення обслуговування

Як правило, мережі Петрі описуються безліччю вершин (P) та переходів (T), а також вхідними (I) та вихідними функціями (O).

Далі наведено опис структури мережі Петрі:

$$C = (P, T, I, O),$$

$$P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_{10}, p_{11}, p_{14}\},$$

$$T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{12}, t_{13}, t_{14}\}.$$

У таблиці 2.2 описані вхідні та вихідні функції для вершин цієї мережі Петрі:

Таблиця 2.2 – Опис вхідних та вихідних функцій вершин мережі Петрі

$I: P \rightarrow T_\infty$	$O: P \rightarrow T_\infty$
$I(p_1) = \{\}$	$O(p_1) = \{t_1, t_2, t_3\}$
$I(p_2) = \{t_1\}$	$O(p_2) = \{t_4\}$
$I(p_3) = \{t_2\}$	$O(p_3) = \{t_5\}$
$I(p_4) = \{t_3\}$	$O(p_4) = \{t_6\}$
$I(p_5) = \{t_4, t_{10}\}$	$O(p_5) = \{t_7\}$
$I(p_6) = \{t_7\}$	$O(p_6) = \{t_{15}, t_{14}\}$
$I(p_7) = \{t_5\}$	$O(p_7) = \{t_8\}$
$I(p_8) = \{t_6, t_{15}\}$	$O(p_8) = \{t_9\}$
$I(p_{10}) = \{t_8\}$	$O(p_{10}) = \{t_{10}, t_{13}\}$
$I(p_{11}) = \{t_9\}$	$O(p_{11}) = \{t_{12}\}$
$I(p_{14}) = \{t_{14}, t_{13}, t_{12}\}$	$O(p_{14}) = \{\}$

У таблиці 2.3 описані вхідні та вихідні функції для вершин цієї мережі Петрі:

Таблиця 2.3 – Опис вхідних та вихідних функцій переходів мережі Петрі

$I: T \rightarrow P_{\infty}$	$O: T \rightarrow P_{\infty}$
$I(t_1) = \{p_1\}$	$O(t_1) = \{p_2\}$
$I(t_2) = \{p_1\}$	$O(t_2) = \{p_3\}$
$I(t_3) = \{p_1\}$	$O(t_3) = \{p_4\}$
$I(t_4) = \{p_2\}$	$O(t_4) = \{p_5\}$
$I(t_5) = \{p_3\}$	$O(t_5) = \{p_7\}$
$I(t_6) = \{p_4\}$	$O(t_6) = \{p_8\}$
$I(t_7) = \{p_5\}$	$O(t_7) = \{p_5\}$
$I(t_8) = \{p_7\}$	$O(t_8) = \{p_{10}\}$
$I(t_9) = \{p_8\}$	$O(t_9) = \{p_{11}\}$
$I(t_{10}) = \{p_5\}$	$O(t_{10}) = \{p_5\}$
$I(t_{12}) = \{p_{11}\}$	$O(t_{12}) = \{p_{14}\}$
$I(t_{13}) = \{p_{10}\}$	$O(t_{13}) = \{p_{14}\}$
$I(t_{14}) = \{p_6\}$	$O(t_{14}) = \{p_{14}\}$
$I(t_{15}) = \{p_6\}$	$O(t_{15}) = \{p_8\}$

Для того, щоб розглянути властивості мережі Петрі, необхідно побудувати дерево досяжності, що показує всі можливі послідовності роботи переходів. Для початку роботи потрібно поставити маркування на вершині p_1 . На рисунку 2.3 представлено дерево досяжності для побудованої мережі Петрі.

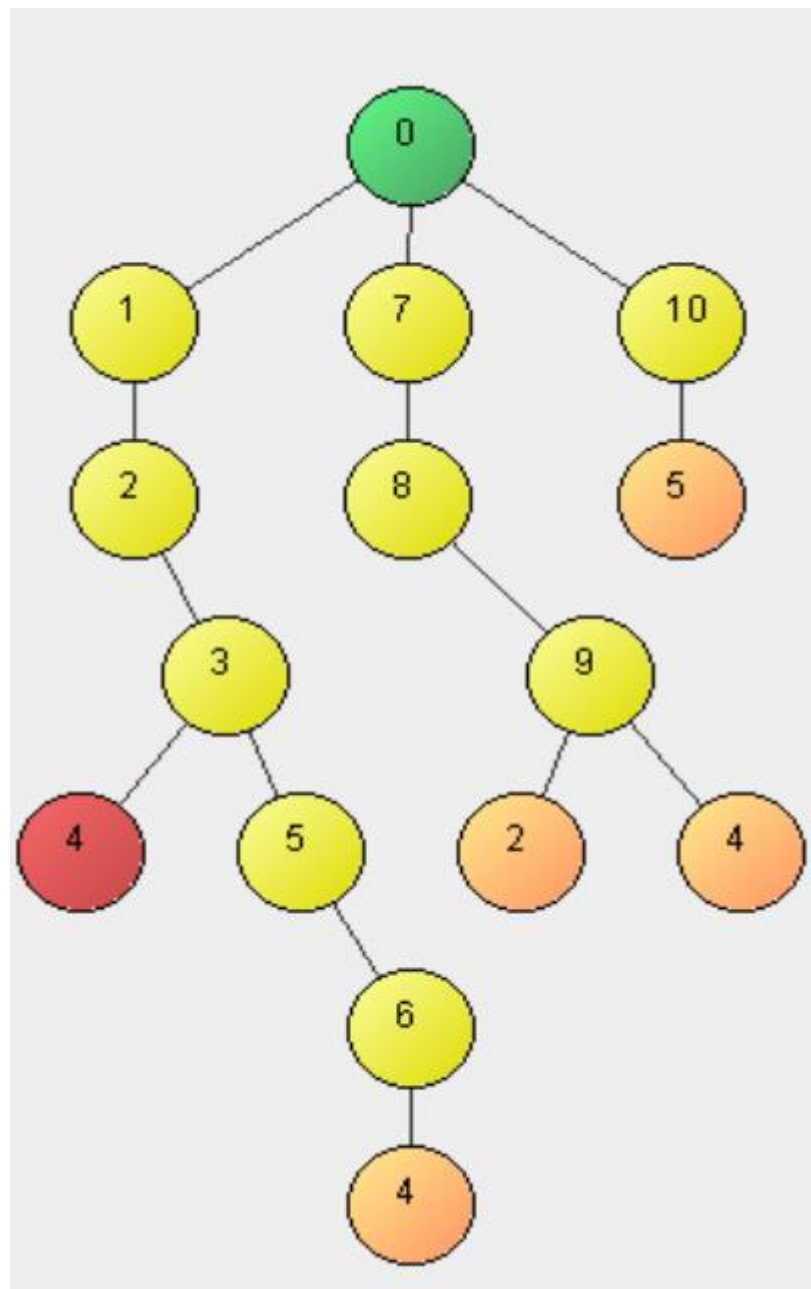


Рисунок 2.3 - Дерево досяжності мережі Петрі

На рисунку видно, що послідовність переходів має три безвихідні ситуації, які можна охарактеризувати як завершення обслуговування заявки в одному з трьох каналів.

На рисунку 2.4 представлено табличний опис дерева досяжності.

Для підтвердження коректності мережі Петрі необхідно розглянути три її властивості: безпеку, обмеженість та збереження.

#	Branch	Prev.marking	Worked Transitions	Cur. marking	MarkType	Level
1	0	M0(1,0,0,0,0,0,0,0,0,0)		M0(1,0,0,0,0,0,0,0,0,0)	ROOT	0
2	1	M0(1,0,0,0,0,0,0,0,0,0)	T1	M1(0,1,0,0,0,0,0,0,0,0)	INTERNAL	1
3	1	M1(0,1,0,0,0,0,0,0,0,0)	T4	M2(0,0,0,0,1,0,0,0,0,0)	INTERNAL	2
4	1	M2(0,0,0,0,1,0,0,0,0,0)	T7	M3(0,0,0,0,0,1,0,0,0,0)	INTERNAL	3
5	1	M3(0,0,0,0,0,1,0,0,0,0)	T13	M4(0,0,0,0,0,0,0,0,0,1)	DEADLOCK	4
6	2	M3(0,0,0,0,0,1,0,0,0,0)	T14	M5(0,0,0,0,0,0,0,1,0,0)	INTERNAL	4
7	2	M5(0,0,0,0,0,0,0,1,0,0)	T9	M6(0,0,0,0,0,0,0,0,0,1)	INTERNAL	5
8	2	M6(0,0,0,0,0,0,0,0,0,1)		M4(0,0,0,0,0,0,0,0,0,1)	REPEATED	6
9	3	M0(1,0,0,0,0,0,0,0,0,0)	T2	M7(0,0,1,0,0,0,0,0,0,0)	INTERNAL	1
10	3	M7(0,0,1,0,0,0,0,0,0,0)	T5	M8(0,0,0,0,0,0,1,0,0,0)	INTERNAL	2
11	3	M8(0,0,0,0,0,0,1,0,0,0)	T8	M9(0,0,0,0,0,0,0,0,1,0)	INTERNAL	3
12	3	M9(0,0,0,0,0,0,0,0,1,0)	T10	M2(0,0,0,0,1,0,0,0,0,0)	REPEATED	4
13	4	M9(0,0,0,0,0,0,0,0,1,0)	T12	M4(0,0,0,0,0,0,0,0,0,1)	REPEATED	4
14	5	M0(1,0,0,0,0,0,0,0,0,0)	T3	M10(0,0,0,1,0,0,0,0,0,0)	INTERNAL	1
15	5	M10(0,0,0,1,0,0,0,0,0,0)	T6	M5(0,0,0,0,0,0,0,1,0,0)	REPEATED	2

Рисунок 2.4 - Табличний опис дерева досяжності

Мережа Петрі є обмеженою, якщо число фішок у будь-якій її позиції не перевищує деякого значення K . У цьому випадку, мережа є обмеженою, оскільки за таблицею видно, що в будь-якій позиції число фішок ніколи не перевищує одиницю. Таким чином, властивість обмеженості підтверджено.

Мережа Петрі є безпечною (частковий випадок обмеженості), якщо кількість фішок у будь-якій позиції не перевищує значення $K=1$. Як вже розглянуто раніше, дана мережа має не більше однієї фішки у кожному переході, отже, вона є безпечною.

Збереження мережі Петрі має місце, якщо загальна кількість маркерів є постійною величиною, тобто нові маркери не додаються, а існуючі не знищуються. У разі цієї мережі властивість збереження має місце.

Розглянуті характеристики цієї мережі Петрі для досліджуваної моделі обслуговування підтверджують її коректність. Використання підходу мереж Петрі дозволило розглянути всі можливі стани роботи моделі масового обслуговування медичного закладу з прикладу кількох однопрофільних фахівців.

Аналіз системи масового обслуговування було проведено з метою аналізу якості функціонування конкретної галузі та виявлення можливості оптимізації самої СМО. Якість функціонування окремих випадках має значення, що може виражатися, зокрема і кількісними показниками. У поліклініці як системі масового обслуговування особливо важливі такі критерії, як довжина черги на

обслуговування, середня тривалість обслуговування, середній час очікування заявки в черзі, час простою обслуговуючих машин [15].

2.2 Алгоритм формування потоків пацієнтів для прийому у спеціаліста у медичній установі

Нехай на момент часу t в систему надійшло безліч робіт, позначене, як $J = \{J_1, J_2, \dots, J_n\}$, кожна з яких має тривалість p_i . У системі є безліч машин $M = \{M_1, M_2, \dots, M_n\}$, що мають навантаження p_i . Під навантаженням обслуговуючої машини мається на увазі відношення середньої кількості заявок до середньої швидкості обслуговування. Необхідно скласти розклад виконання нових робіт, що надійшли до моменту часу t_1 так, щоб робота була виконана цілком на одній машині без повторного перерозподілу наступних робіт. Для виконання алгоритму необхідно спочатку визначити всі значення навантажень обслуговуючих машин, потім значення навантажень повинні бути відсортовані від мінімального до максимального значення. Для виконання набору робіт, що з'явилися на момент часу t , необхідний набір машин з мінімальним навантаженням. Вважається, що навантаження машин вже відомі на момент t . Після сортування навантажень черги заповнюються заявками, причому спочатку додаються заявки з більшою тривалістю обслуговування. Коли черга заповнена даними заявками, повторно оцінюється навантаження каналів, проводиться сортування каналів навантаження і в черзі до більш вільних каналів додаються заявки з меншою тривалістю обслуговування. Даний алгоритм враховує той факт, що заходи щодо обслуговування пацієнта не є роздільними та не можуть бути виконані двома фахівцями одночасно. Також враховується, що заявки, які вже стоять у черзі на обслуговування, не можуть бути перенесені в іншу чергу або виключені з черги.

Процес початку обслуговування пацієнтів у спеціаліста медичного закладу повинен починатися зі збору інформації про заявки, що надійшли, та оцінки

навантаження лікаря для подальшого розподілу пацієнтів у розкладі. Збір інформації про пацієнтів, що записалися на прийом, відбувається з початку прийому дзвінків від самих пацієнтів або через web-сервіс медичного закладу. Кількість записів не може бути обмежена, але оскільки вирішено, що оптимальним варіантом контролю за чергами буде прийом із запису, залежно від робочого навантаження конкретного лікаря, до нього на прийом можна записати певну кількість пацієнтів на день. Від завантаженості конкретного лікаря залежить скільки ще пацієнтів можна записати до нього на прийом.

Важливо враховувати, що бувають пацієнти, які приходять приймати без запису. Для досліджуваної моделі обслуговування доцільно формування черги лише з запису, щоб мінімізувати очікування пацієнтів у черзі. Але якщо з'явиться пацієнт без запису, необхідно записати його на прийом до лікаря за алгоритмом, що розглядається, з урахуванням встановлених навантажень на лікаря. Якщо в момент приходу пацієнта є фахівець із низьким навантаженням та низьким числом пацієнтів у черзі, то співробітник реєстратури може направити пацієнта до нього.

На рисунку 2.5 зображено блок-схему словесного алгоритму прийняття заявок на прийом до лікаря та процесу прийому пацієнтів.

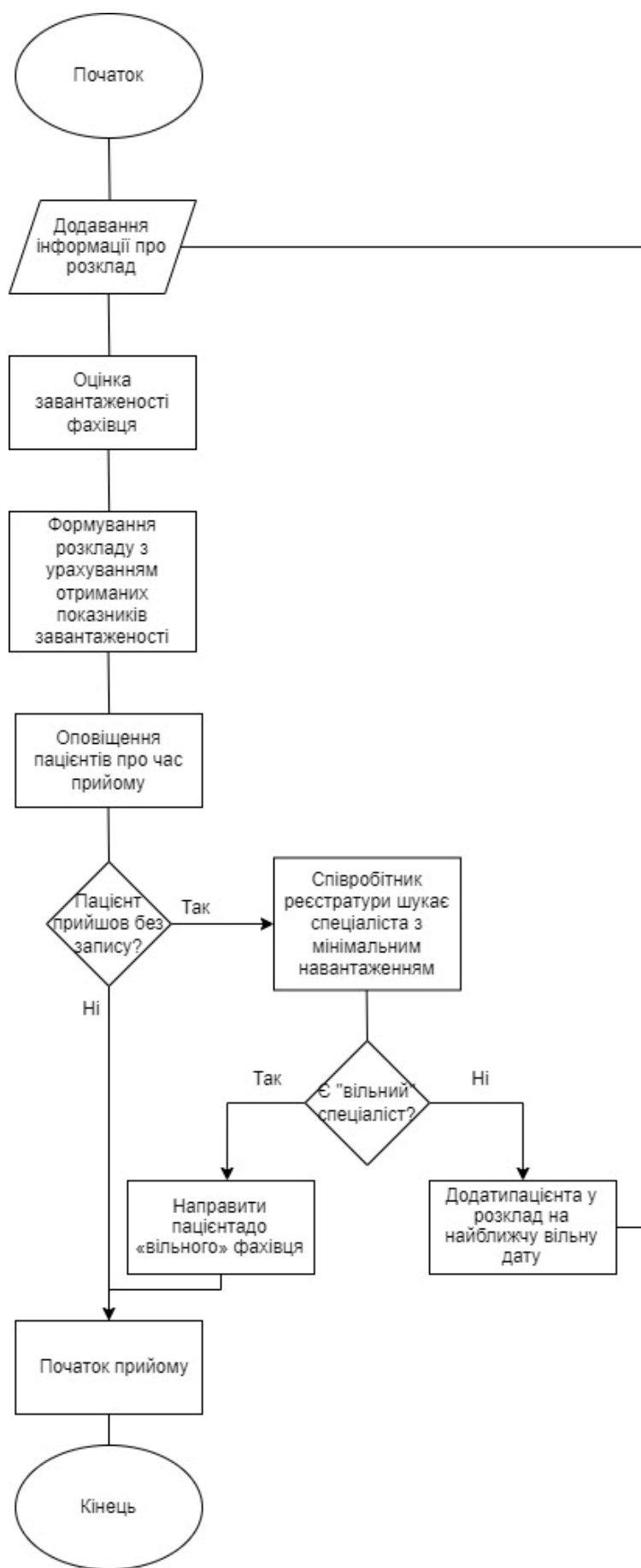


Рисунок 2.5 - Блок-схема процесу створення розкладу та прийому пацієнтів

Опис переходів мережі Петрі представлено таблиці 2.4.

Таблиця 2.4 - Опис переходів мережі Петрі

Вершина	Умова спрацьовування
P_1	початок
P_2	Додавання інформації про розклад
P_3	Оцінка завантаженості фахівців
P_4	Формування розкладу з урахуванням отриманих показників завантаженості
P_5	Оповіщення пацієнтів про час прийому
P_6	Перевірка приходу пацієнта із запису
P_7	Пацієнт прийшов без запису, співробітник шукає спеціаліста з мінімальним навантаженням
P_8	Перевірка існування фахівця з мінімальним навантаженням
P_{10}	Лікаря з мінімальним навантаженням немає, записати пацієнта на найближче дату, на крок 2
P_9	Лікар з мінімальним навантаженням є, направити пацієнта до фахівця з мінімальним навантаженням
P_{12}	Початок прийому
P_{11}	Кінець

На рисунку 2.7 зображено дерево досяжності створеної мережі Петрі.

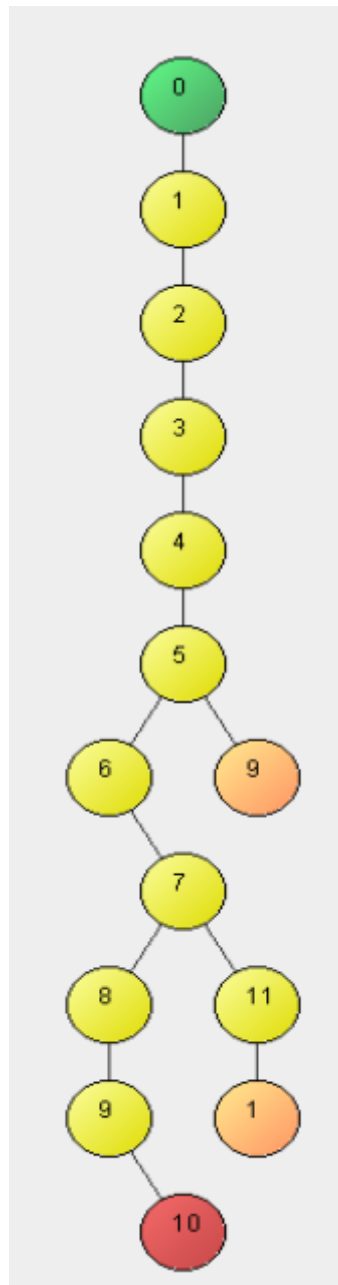


Рисунок 2.7 - Дерево досяжності мережі Петрі

Для опису властивостей мережі Петрі буде розглянуто таблицю з описом дерева досяжності, зображену рисунку 2.8.

#	Branch	Prev.marking	Worked Transitions	Cur. marking	MarkType	Level
1	0	M0(1,0,0,0,0,0,0,0,0,0,0)		M0(1,0,0,0,0,0,0,0,0,0,0)	ROOT	0
2	1	M0(1,0,0,0,0,0,0,0,0,0,0)	T1	M1(0,1,0,0,0,0,0,0,0,0,0)	INTERNAL	1
3	1	M1(0,1,0,0,0,0,0,0,0,0,0)	T2	M2(0,0,1,0,0,0,0,0,0,0,0)	INTERNAL	2
4	1	M2(0,0,1,0,0,0,0,0,0,0,0)	T3	M3(0,0,0,1,0,0,0,0,0,0,0)	INTERNAL	3
5	1	M3(0,0,0,1,0,0,0,0,0,0,0)	T4	M4(0,0,0,0,1,0,0,0,0,0,0)	INTERNAL	4
6	1	M4(0,0,0,0,1,0,0,0,0,0,0)	T5	M5(0,0,0,0,0,1,0,0,0,0,0)	INTERNAL	5
7	1	M5(0,0,0,0,0,1,0,0,0,0,0)	T6	M6(0,0,0,0,0,0,1,0,0,0,0)	INTERNAL	6
8	1	M6(0,0,0,0,0,0,1,0,0,0,0)	T7	M7(0,0,0,0,0,0,0,1,0,0,0)	INTERNAL	7
9	1	M7(0,0,0,0,0,0,0,1,0,0,0)	T9	M8(0,0,0,0,0,0,0,0,1,0,0)	INTERNAL	8
10	1	M8(0,0,0,0,0,0,0,0,1,0,0)	T12	M9(0,0,0,0,0,0,0,0,0,0,1)	INTERNAL	9
11	1	M9(0,0,0,0,0,0,0,0,0,0,1)	T13	M10(0,0,0,0,0,0,0,0,0,0,1)	DEADLOCK	10
12	2	M7(0,0,0,0,0,0,0,1,0,0,0)	T10	M11(0,0,0,0,0,0,0,0,0,1,0)	INTERNAL	8
13	2	M11(0,0,0,0,0,0,0,0,0,1,0)	T11	M1(0,1,0,0,0,0,0,0,0,0,0)	REPEATED	9
14	3	M5(0,0,0,0,0,1,0,0,0,0,0)	T8	M9(0,0,0,0,0,0,0,0,0,0,1)	REPEATED	6

Рисунок 2.8 - Табличний опис дерева досяжності

Для перевірки коректності створеної мережі Петрі необхідно повторно розглянути її властивості.

Мережа є обмеженою, оскільки кількість фішок у мережі не перевищує деякого числа K (у цій мережі кількість фішок дорівнює 1).

Мережа є безпечною, оскільки кількість фішок не перевищує числа K , що дорівнює 1.

Мережа Петрі має властивість збереження, оскільки число фішок у мережі є постійним.

Таким чином, коректність роботи цієї мережі Петрі за цим алгоритмом підтверджує оптимальність його роботи.

На рисунку 2.9 представлена інформаційна модель передачі між основними структурами поліклініки при режимі прийому по запису.

На рисунку у вигляді блоків відмічені всі можливі процеси обслуговування пацієнта в поліклініці, а стрілками позначені шляхи та способи обміну тією чи іншою інформацією (заявка чи дані про пацієнта).



Рисунок 2.9 - Модель роботи медичного закладу

Дана модель має місце, якщо процес обслуговування буде реорганізований за алгоритмом, наведеним на рисунку 2.5. У даному випадку, пацієнт, який прийшов без попереднього запису, буде в будь-якому випадку записаний на прийом в даний момент часу, якщо є вільний спеціаліст, або в інше найближче час.

Розроблена модель системи масового обслуговування враховує зміну робочого навантаження фахівця у той чи інший момент часу та, залежно від ступеня навантаження, дозволяє обмежити чергу за рахунок перерозподілу заявок на обслуговування в інших більш вільних каналах. Щоб упорядкувати процеси збору інформації про пацієнтів та навантаження фахівців, а також процеси формування потоків пацієнтів на обслуговування, сформульовано алгоритм розподілу потоків на обслуговування залежно від встановленого навантаження. За моделлю та алгоритмом створені мережі Петрі, властивості мереж підтверджують коректність їхньої роботи.

Для спрощення роботи з алгоритмом потрібна розробка допоміжного модуля медичної інформаційної системи, що дозволяє формувати розклад

прийому фахівців. Цей модуль повинен враховувати значення навантажень лікарів для запису нових пацієнтів на прийом.

3 ПРОЕКТУВАННЯ МОДУЛЯ РОЗКЛАДУ ПРИЙОМУ ПАЦІЄНТІВ

3.1 Розробка засобів управління розкладом прийому пацієнтів

Проведений раніше аналіз показав, що модуль розкладів прийому пацієнтів має виводити статистику потоку пацієнтів, що дозволяє базі складати нові розклади. Інформаційна система, що розробляється, повинна забезпечувати функціонування трьох груп користувачів – адміністратора системи, доктора та співробітника реєстратури. Короткий опис дійових осіб системи наведено в таблиці 3.1.

Таблиця 3.1 – Короткий опис дійових осіб

Назва	Опис діяльності та можливостей
Адміністратор	Керує користувачами системи (додає та видаляє нових користувачів), здійснює авторизацію в системі
Лікар	Здійснює авторизацію в системі, переглядає інформацію про розклад, статистику прийомів, пацієнтів
Співробітник реєстратури	Здійснює авторизацію в системі, складає розклади для лікарів, переглядають статистику прийомів

Таким чином, до системи мають доступ три групи користувачів. Адміністратор системи відповідає за керування користувачами. Основне завдання лікаря – перегляд медичної інформації, пов'язаної із прийомом пацієнтів, розкладами, а також перегляд статистики. Співробітник реєстратури займається

безпосередньо складанням розкладів, однак йому також може бути доступна функція перегляду статистики. До можливостей адміністратора модуля належить керування обліковими записами користувачів: додавання та видалення записів.

На рисунку 3.1 представлена діаграма варіантів використання, де зображені всі можливі варіанти дій у системі кожного з акторів.

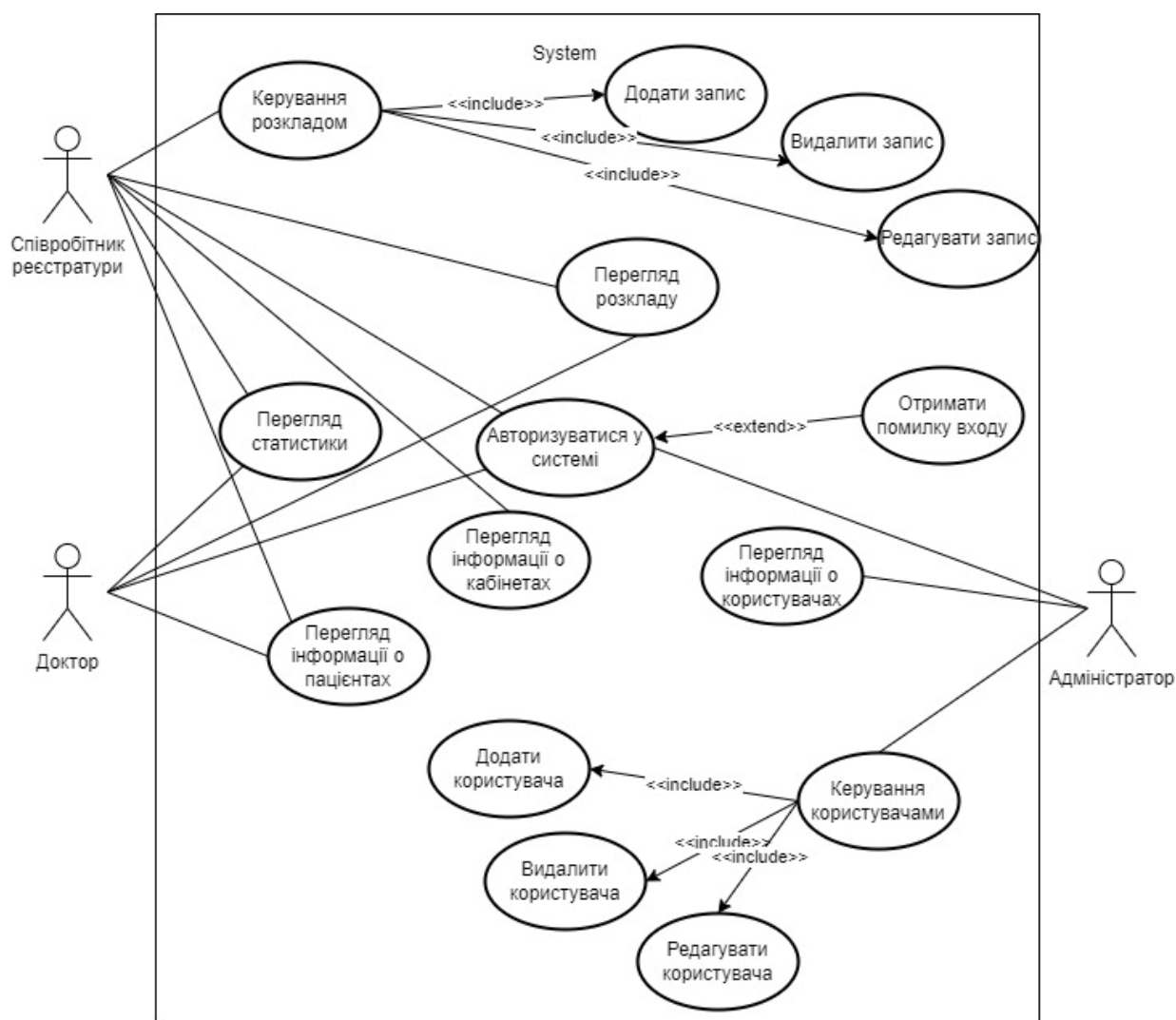


Рисунок 3.1 – Діаграма варіантів використання

Далі розглянуто один із найважливіших варіантів використання – Додавання запису. Актор, який виконує даний варіант використання, є співробітником реєстрації. Актор відкриває вікно програми, призначене для додавання нового запису на прийом, який включає дані про пацієнта, дані про лікаря та його профіль. Після заповнення даних про пацієнта актор натискає кнопку «Додати». Введена

інформація повинна мати правильні типи даних. Якщо дані введені не так, то має з'явитися відповідна помилка. Дані додаються до таблиці обліку пацієнтів та таблиці з розкладом одночасно. Якщо дані про пацієнта вже існують в базі (тобто пацієнт одного разу вже був на прийомі), дані додаються тільки в таблицю з розкладом. Якщо пацієнт звернувся вперше, то дані про нього додаються до таблиці

"Пацієнти". Час і дата прийому є величинами, що змінюються, або можуть бути порожніми, але для того, щоб скласти підсумковий розклад, їх необхідно ввести.

Іншою важливою функцією цього модуля є "Перегляд статистики". Функція полягає в тому, що співробітник медичної організації переглядає останню актуальну інформацію про прийоми конкретного лікаря. Особливо важливо у статистиці врахувати такі атрибути, як причини звернень та кількість прийомів пацієнта, а також профіль фахівця, до якого пацієнти записувалися на прийом. Ці характеристики, крім загальної кількості звернень, можна застосувати визначення робочої навантаження конкретного лікаря.

Актори, які мають доступ до цієї функції – співробітник реєстратури та лікар. Актор відкриває вікно перегляду статистики і вибирає у списку профіль лікаря, пацієнтів якого він хоче побачити. Після вибору профілю повинен з'явитися список пацієнтів, які зверталися до фахівців цього профілю на прийом останнім часом. Також актор може вибрати конкретний часовий проміжок, у який надходила певна кількість пацієнтів. Крім профілю, можливий вибір пацієнтів, які прийшли на прийом з конкретною причиною, а також виведення пацієнтів, які прийшли на первинний або повторний прийом.

Для того щоб змодельовати поведінку системи, що розробляється, потрібно не тільки представити процеси змін її станів, а й детально показати особливості алгоритмічної та логічної реалізації дій, які система виконує. Для моделювання подібних дій потрібна побудова діаграми діяльності, яка показує покрокові дії актора системи та реакцію системи на ці дії. Для кожної конкретної дії зазвичай потрібна побудова окремої діаграми. У рамках розробки даної системи

представлено дві діаграми для найважливіших варіантів використання – «Додавання запису» та «Перегляд статистики».

На рисунку 3.2 зображено діаграму діяльності для варіанта використання «Додати запис», доступна співробітнику реєстратури.

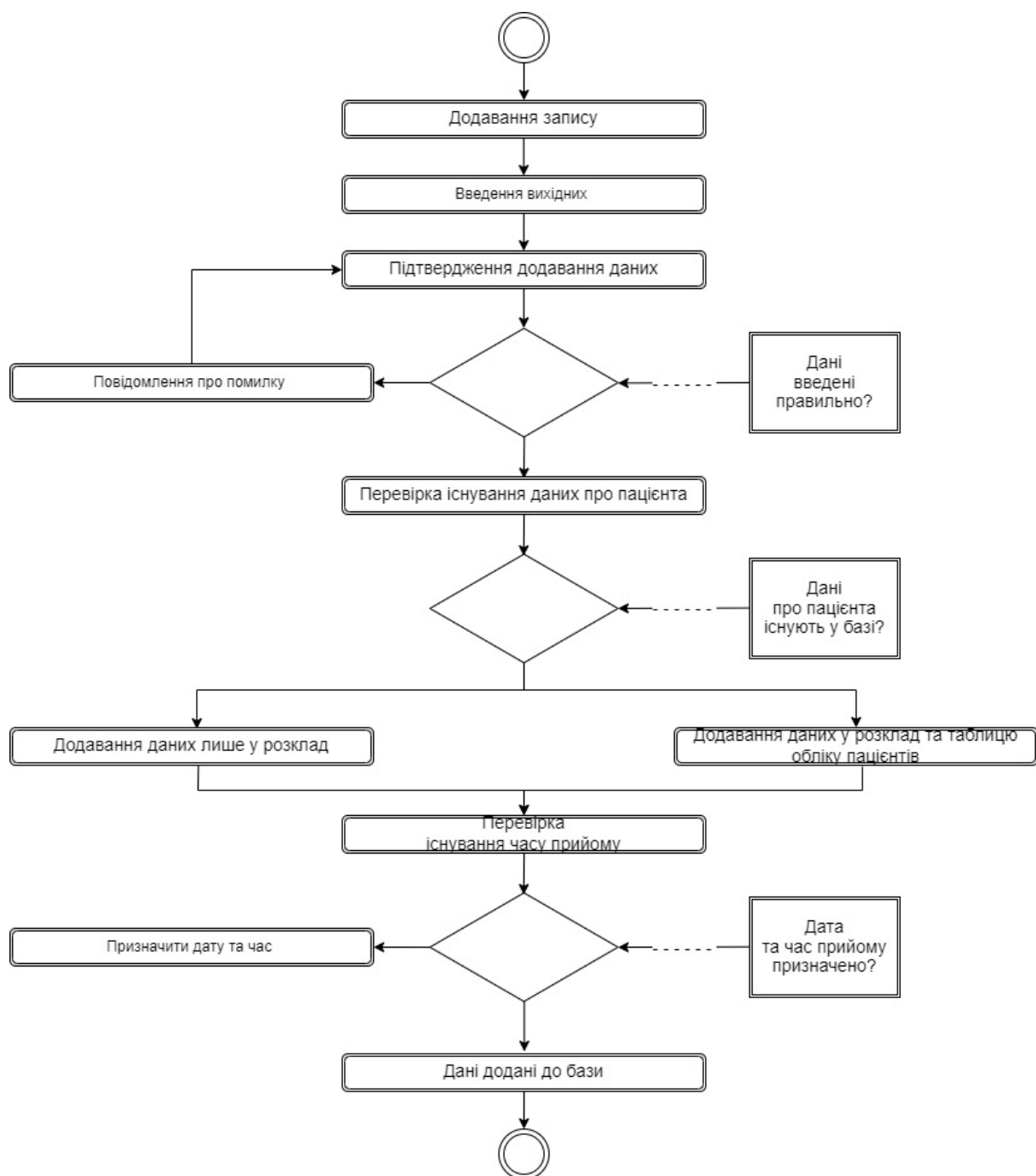


Рисунок 3.2 Діаграма діяльності для варіанта використання
"Додати запис"

На рисунку 3.3 зображено діаграму діяльності для варіанта використання «Перегляд статистики», доступна як працівнику реєстратури, і лікарю. На цьому рисунку виведення статистики з прийомів у медичному закладі здійснюється за профілем лікаря.

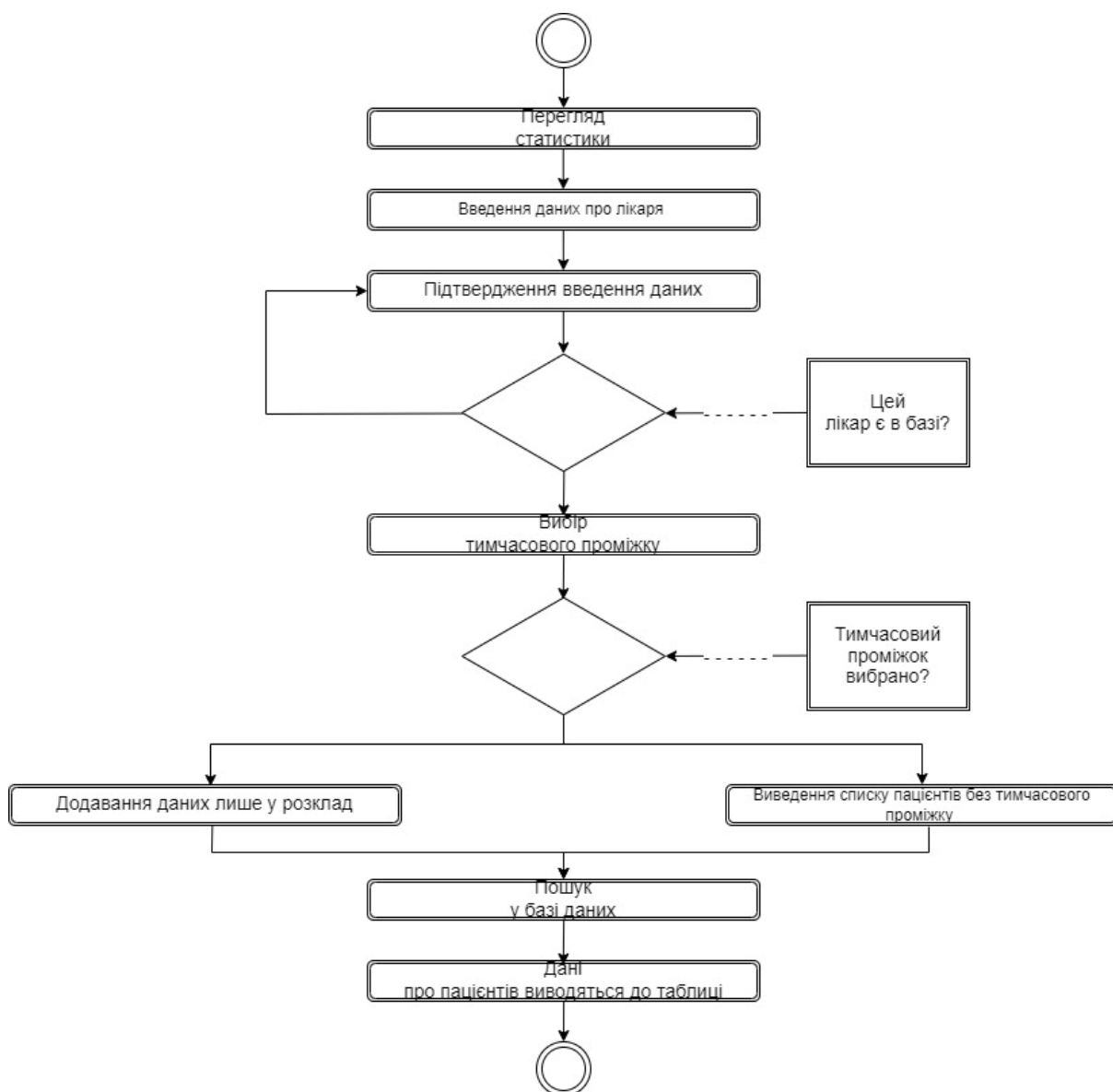


Рисунок 3.3 – Діаграма діяльності для варіанта використання "Перегляд статистики"

Після розробки вимог до проектованої системи, визначення варіантів використання та користувачів системи необхідно розробити базу даних.

3.2 Проектування бази даних

Особливістю всіх МІС, наведених у главі 1, є їх проектування з урахуванням реляційних баз даних. Як було зазначено раніше, медичні інформаційні системи здійснюють зберігання як структурованої, так і частково структурованої чи зовсім неструктурованої інформації. Реляційні бази цілком підходять для зберігання такого роду даних, проте мають недоліки. Головна складність полягає у складанні складних запитів до таблиць.

Так як типові МІС розробляються з використанням реляційних баз даних, доцільніше робити розробку додаткового модуля розкладів у реляційній СКБД. Однак не виключено варіант перенесення всієї системи на об'єктно-орієнтовану базу даних. Об'єктно-орієнтовані СКБД мають певні переваги перед реляційними. Головною їхньою перевагою є можливість зберігання даних, що не мають жорсткої структури [17].

База даних, що розробляється, будується з використанням системи управління бази даних Microsoft SQL Server. Побудова бази даних проходить у кілька етапів: створення концептуальної моделі, логічної та фізичної моделі.

Побудова концептуальної моделі полягає у визначенні об'єктів, їх характеристик та зв'язків між ними. У процесі створення концептуальної моделі формулюється набір сутностей та його характеристик. Список сутностей та його короткі описи представлені у таблиці 3.2.

Таблиця 3.2- Опис сутностей бази даних

Ім'я сутності	Опис
Лікар	Сутність містить відомості у лікарі
Пацієнт	Сутність містить первинні відомості про пацієнта
Кабінет	Сутність зберігає відомості про кабінети, в яких ведеться прийом
Проблема	Сутність відображає дані про основні проблеми, якими звертаються пацієнти
Прийом	Сутність зберігає дані про кількість прийомів (первинний або повторний)
Обліковий запис	Сутність містить логіни та паролі всіх користувачів системи
Розклад	Сутність відображає дані про розклад прийомів пацієнтів

Атрибути всіх сутностей та їх описи наведено у таблицях.

У таблиці 3.3 описані атрибути сутності "Доктор". Сутність «Доктор» містить необхідні відомості про конкретного фахівця медичного закладу: ім'я, профіль, відомості про середнє навантаження. Сутність "Доктор" прив'язана до сутностей "Акаунт" та "Кабінет" за допомогою відповідних зовнішніх ключів.

Таблиця 3.3 - Сутність "Доктор"

Ім'я стовпця	Змістовний опис	Тип даних	Розмірність	Можливість значення Null	Роль	приклад
ID_Doctor	Ідентифікатор лікаря	Ціле		Ні	РК	3
Name_Last Name_Patro	Прізвище ім'я по батькові	Текст	128	Ні		Іванов Іван Іванович
Profile	Профіль	Текст	128	Ні		Терапевт
ID_Cabinet	Номер кабінету	Ціле		Ні	FK	303
ID_Account	Ідентифікатор облікового запису	Ціле		Ні	FK	34
Workload	Ступінь навантаження лікаря (в процентах)	Дійсне		Так		24

У таблиці 3.4 наведено атрибути сутності «Пацієнт». Ця сутність містить основну інформацію про пацієнта: ім'я, прізвище, по батькові, дату народження, номер страхового поліса, номер телефону пацієнта.

Таблиця 3.4 - Сутність "Пацієнт"

Ім'я стовпця	Змістовний опис	Тип даних	Розмірність	Можливість значення Null	Роль	приклад
ID_Patient	Ідентифікатор лікаря	Ціле		Ні	РК	3
Last_name	Прізвище	Текст	128	Ні		Іванов
Name	Ім'я	Текст	128	Ні		Іван
Patronymic	По-батькові	Текст	128	Ні		Іванович
Birthday	дата народження	Дата		Ні		30.11.1996
Policy	Номер страхового поліса	Ціле		Ні		5848300086 9000181
Telephone	Номер телефону пацієнта	Ціле		Ні		8960329396 3

У таблиці 3.5 описані атрибути сутності "Акаунт". Сутність

«Акаунт» призначена для зберігання інформації про паролі та логіни співробітників медичного закладу. Також сутність містить такі атрибути, як дата ініціалізації облікового запису та тип облікового запису (лікар, співробітник реєстратури або адміністратор).

Таблиця 3.5 - Сутність "Акаунт"

Ім'я стовпця	Змістовний опис	Тип даних	Розмір - ність	Можливість значення Null	Роль	приклад
Login	Логін	Текст	10	Ні		Admin
ID_Account	Ідентифікатор акаунту	Ціле		Ні	РК	7
Password	Пароль	Текст	8	Ні		Sc215iL5
Initialization_Date	Дата ініціалізації	Дата		Ні		20.09.2020
Account_Type	Тип облікового запису	Текст		Ні		Доктор

У таблиці 3.6 наведено атрибути сутності «Прийом». Ця сутність зберігає інформацію про два типи прийому: первинний чи повторний. Сутність пов'язана з сутністю «Розклад» у вигляді відповідного первинного ключа.

Таблиця 3.6 - Сутність "Прийом"

Ім'я стовпця	Змістовний опис	Тип даних	Розмірність	Можливістьзначення Null	Роль	приклад
ID_Appointment	Ідентифікатор прийому	Ціле		Ні	РК	1
Name_Appointment	Найменування	Текст	128	Ні		Первинний

У таблиці 3.7 описані атрибути сутності "Кабінет". Сутність «Кабінет» призначений для збору інформації про функціонуючі кабінети конкретного медичного закладу та прив'язаний до запису про конкретного лікаря за допомогою відповідного первинного ключа.

Таблиця 3.7 - Сутність "Кабінет"

Ім'я стовпця	Змістовний опис	Тип даних	Розмірність	Можливістьзначення Null	Роль	приклад
ID_Cabinet	Номер кабінету	Ціле		Ні	РК	303
Floor	Поверх	Ціле		Ні		3
Phone	Номер телефону в кабінеті	Текст	4	Ні		22-33

У таблиці 3.8 описано атрибути сутності «Проблема». Сутність містить основну інформацію про проблеми, з якими пацієнт може звертатися до лікаря. При цьому інформація про проблему може бути записана як у загальному вигляді (наприклад, «Хронічне захворювання»), так і з поясненням конкретної проблеми. Ця інформація може бути корисною пізніше для збору статистики щодо захворюваності.

Таблиця 3.8 - Сутність "Проблема"

Ім'я стовпця	Змістовний опис	Тип даних	Розмірність	Можливість значення Null	Роль	приклад
ID_Problem	Номер проблеми	Ціле		Ні	PK	3
Name_Problem	Найменування проблеми	Текст		Ні		Консультація
Description	Опис проблеми	Текст	128	Ні		79
Complexity	Складність проблеми	Дійсне	10	Ні		3

У таблиці 3.9 описані атрибути сутності "Розклад". Сутність є основною основою, оскільки містить інформацію про кількість прийомів для конкретного лікаря у конкретний час. Ця сутність переважно складається із зовнішніх ключів інших сутностей бази. Також сутність «Розклад» може використовуватися у створенні процедур, що зберігаються, спрямованих на збір статистики з прийомів пацієнтів.

Таблиця 3.9 - Сутність "Розклад"

Ім'я стовпця	Змістовний опис	Тип даних	Розмірність	Можливість значення Null	Роль	приклад
ID_Doctor	Ідентифікатор лікаря	Ціле		Ні	FK	3
ID_Patient	Ідентифікатор пацієнта	Ціле		Ні	FK	7
Time	Час прийому	Час		Ні		12.50
Date	дата прийому	Дата		Ні		20.09.2020
ID_Cabinet	Ідентифікатор кабінету	Ціле		Ні	FK	303
ID_Reception	Ідентифікатор прийому	Ціле		Ні	FK	2
ID_Problem	Ідентифікатор проблеми	Ціле		Ні	FK	203

У таблиці 3.10 наведено атрибути сутності «Співробітник реєстратури». Сутність містить інформацію про співробітників реєстратури та, як і сутність «Доктор», прив'язана до сутності «Акаунт». Сутність містить найважливіші для роботи модуля атрибути: унікальний номер співробітника, ідентифікатор облікового запису, прив'язаний до таблиці «Акаунт» та ініціали співробітника.

Таблиця 3.10 - Сутність "Співробітник реєстратури"

Ім'я стовпця	Змістовний опис	Тип даних	Розмірність	Можливість значення Null	Роль	приклад
ID_Employee	Ідентифікатор лікаря	Ціле		Ні	PK	4
ID_Account	Ідентифікатор облікового запису	Ціле		Ні	FK	10
Initials	Прізвище ім'я по батькові	Текст	128	Ні		Маркіна Олена Петрівна

З наявних даних можна побудувати логічну і фізичну модель бази даних, які представлені далі як наочних схем [18].

3.3 Побудова логічної моделі бази даних

Логічною моделлю слід називати модель бази даних, що розробляється, яка не прив'язана до конкретної СКБД. При побудові логічної моделі виділяються конкретні сутності, атрибути та зв'язки між ними. На цьому етапі розробки бази даних можна призначити типи даних атрибутів. Також на етапі розробки логічної моделі можна виділити первинні та зовнішні ключі, які зв'язуватимуться з іншими сутностями.

Крім позначення сутностей та атрибутів необхідно позначити зв'язки між сутностями. Основні типи зв'язків у даній моделі - "один до багатьох" і "один до одного". Наприклад, зв'язок «один до одного» присутній у сутності «Доктор» та «Акаунт», або «Співробітник реєстратури» та

"Акаунт", оскільки один співробітник медичного закладу має лише один аккаунт.

Логічна модель бази даних модуля складання та перегляду розкладів прийому лікарів, що є частиною медичної інформаційної системи, спроектована серед Erwin Data Modeler [19].

На рисунку 3.4 зображена схема логічної моделі бази даних, що розробляється.

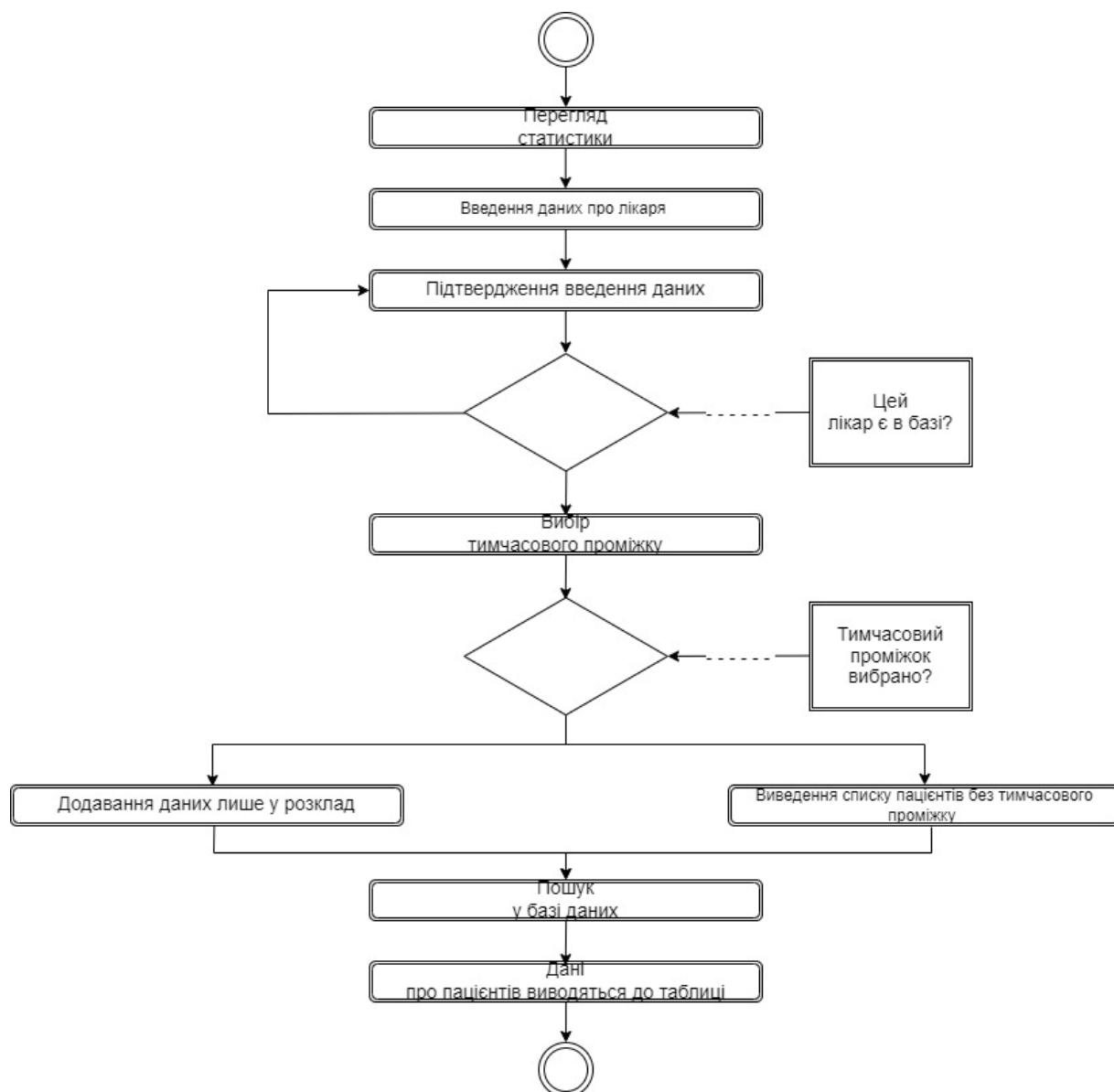


Рисунок 3.4 – Логічна модель бази даних

Наступним етапом проектування є побудова фізичної моделі бази даних.

3.4 Побудова фізичної моделі бази даних

Фізичною моделлю бази даних прийнято називати створену логічну модель, але інтерпретовану у межах конкретної системи керування базами даних (СКБД).

На рисунку 3.5 зображено фізичну модель бази даних проєктованої системи.

Наступним кроком після створення моделі бази даних і безпосередньо самої бази в обраній СУБД необхідно створити всі потрібні процедури, що зберігаються, тригери і уявлення, які будуть працювати в даній базі [20].

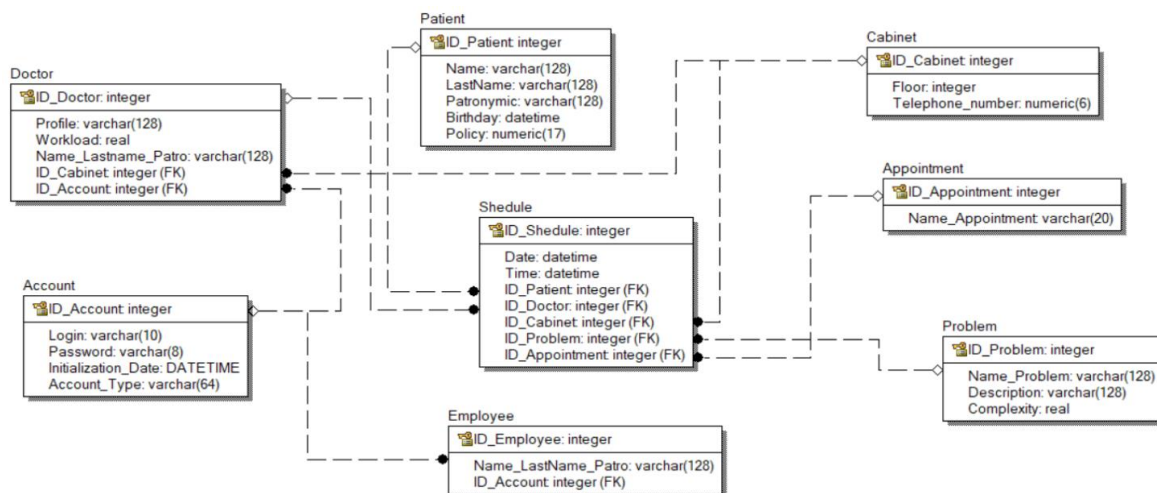


Рисунок 3.5 - Фізична модель бази даних

Основні процедури, що зберігаються, спрямовані на роботу з додаванням, видаленням і зміною даних у створених таблицях, однак також необхідно створити збережену процедуру для збору статистики за пацієнтами, що надійшли за останній час, і середньому навантаженню лікарів.

3.5 Розробка збережених процедур, функцій та тригерів бази даних

Крім стандартних функцій додавання, зміни та видалення інформації, необхідна розробка спеціалізованих функцій. У рамках розробки прототипу модуля розкладів виділено три необхідні функції.

Зберігається процедура «Виведення статистики» здійснює вибірку з таблиць «Пацієнти», «Доктор» та «Розклад» усіх пацієнтів, які прийшли на прийом до лікаря конкретного профілю у встановлені проміжки часу. При створенні процедури особливо важливо врахувати такі атрибути пацієнтів, як число прийомів

і причина звернення, оскільки вибірку можна здійснювати за цими атрибутами. Процедура створюється з метою отримання статистичних даних про захворюваність та оцінку завантаженості фахівців конкретного профілю.

Збережена процедура «Додавання запису» необхідна для додавання до бази даних про пацієнта та запису пацієнта на прийом. Якщо пацієнт вже одного разу відвідував лікаря у цій установі, то додаються лише дані про запис на новий прийом. Перевірку існування пацієнта можна перевіряти щодо існування даних про пацієнта в базі, а також значення поля «Прийом» у таблиці «Розклад». Якщо прийом повторний, то передбачається, що вже є у базі. Час і дата прийому можуть ставитися відразу при записі, і пізніше.

Процедура «Перегляд розкладу», що зберігається, полягає у вибірці розкладу для конкретного лікаря в конкретну дату. Атрибути, за якими здійснюється вибірка – ідентифікатор лікаря та дата.

Крім описаних процедур у базі повинні бути збережені процедури, що зберігаються для адміністратора бази, наприклад, процедура додавання нового користувача. Адміністратор додає всі дані про співробітника чи лікаря, призначає їм логін та пароль. Якщо адміністратор додає до бази лікаря, слід врахувати атрибут «Ступінь навантаження». За замовчуванням адміністратор надає цьому атрибуту значення 0. Також адміністратор може відповідно змінювати та видаляти дані про співробітників медичної установи, які мають до цього модуля МІС доступ. Крім того, адміністратор може додавати дані про кабінети.

Для зміни завантаженості лікаря необхідний тригер зміни значення атрибуту «Ступінь завантаження», крім того, можливе створення процедури, що зберігається, обчислення ступеня робочого навантаження лікаря в залежності від дня тижня [21].

3.6 Проектування програмних засобів

Після етапу проектування бази даних слідує етап розробки програмного забезпечення. Для розробки програмних засобів модуля медичної інформаційної системи використовуються методики об'єктно-орієнтованого програмування. Цей підхід до створення програм передбачає уявлення програми як набору взаємозалежних класів і методів. У цьому випадку класи містять інформацію про частини інтерфейсу програми (кнопки, таблиці, вікна для введення інформації). Також класи містять методи підключення програми до серверної частини програми – бази даних, звідки у вікна програми виводиться інформація. Кнопки прив'язані до методів обробки отриманої інформації та її оновлення у базі даних. Через війну розробки має з'явитися діаграма класів створюваного докладання.

Діаграма класів – це структурна діаграма, що створюється в рамках мови UML, яка наочно показує ієрархію класів у системі, що розробляється, а також відображає їх атрибути, методи, інтерфейси та взаємозв'язки між ними.

На рисунку 3.6 зображена діаграма класів для модуля, що розробляється.

Оскільки архітектура програми клієнт-серверна, класи в клієнтському додатку призначені саме для відображення інформації, що зчитується з бази даних, включаючи процедури додавання, зміни та видалення інформації, а також вибірки певної інформації [22].

На діаграмі видно, що точкою входу до програми є клас Program. Цей клас відкриває вікно авторизації, що забезпечує користувачеві доступом до функціоналу залежно від цього, яким конкретно актором системи є користувач. Тобто, якщо користувач увійде до системи, використовуючи логін та пароль співробітника реєстратури, то йому будуть доступні функції додавання та зміни записів у розкладі. Проте існує функціонал, доступний користувачам кількох типів. Так, вікна перегляду статистики можуть бути доступні одночасно і співробітнику реєстратури, і лікарю.

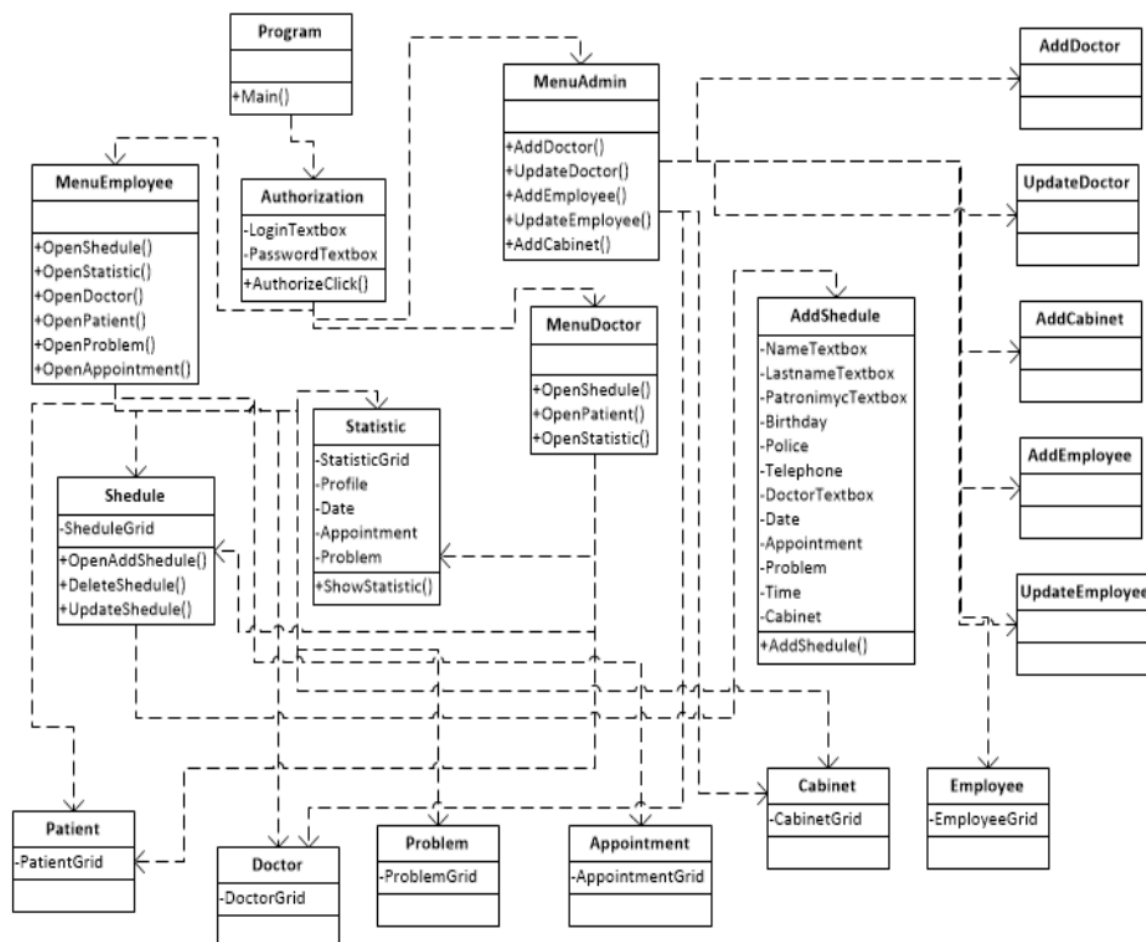


Рисунок 3.6 - Інтерфейс головної сторінки додатка

Далі розглянуті класи, виділені на діаграмі докладніше. У таблиці 3.10 наведено всі створені класи програми та його функціональне призначення.

Ця діаграма відображає функціонал співробітника реєстратури та лікаря медичного закладу.

Проектування класів здійснено з використанням середовища Microsoft Visual Studio мовою C#, що дозволяє реалізувати розробку програми з використанням об'єктно-орієнтованого підходу [23].

Розробка інтерфейсу модуля медичної інформаційної системи зроблено також серед Microsoft Visual Studio з використанням інтерфейсу Windows.Forms.

Таблиця 3.10 – Опис класів програми

Ім'я класу	Функціональне призначення
Authorization	Реалізація вікна авторизації користувачів системи.
MenuDoctor	Реалізація головного вікна лікаря.
MenuEmployee	Реалізація головного вікна працівника реєстратури.
MenuAdmin	Реалізація головного вікна адміністратора.
Patient	Реалізація вікна з таблицею, що відображає список пацієнтів.
Доктор	Реалізація вікна з таблицею, що відображає список лікарів.
Shedule	Клас реалізує вікно відображення розкладу.
AddShedule	Клас реалізує вікно та функцію додавання нової записи та пацієнта.
Statistic	Клас реалізує збережену процедуру бази даних "Перегляд статистики".
AddDoctor	Клас реалізує функцію додавання лікаря.
AddEmployee	Клас реалізує вікно додавання працівника.
AddCabinet	Клас реалізує вікно додавання кабінету.
UpdateEmployee	Клас реалізує вікно зміни даних про співробітника.
UpdateDoctor	Клас реалізує вікно зміни даних про лікаря.
Appointment	Клас реалізує вікно відображення списку прийомів.
Problem	Клас реалізує вікно відображення списку скарг.
Cabinet	Клас реалізує вікно відображення списку кабінетів.
Employee	Клас реалізує вікно відображення списку співробітників.

Щоб зрозуміти взаємозв'язок представлених класів, необхідно побудувати діаграму компонентів. Як фізичні компоненти в даному випадку представлені класи програми. Діаграма компонентів проектного модуля, представленого мовою моделювання UML представлено рисунку 3.7.

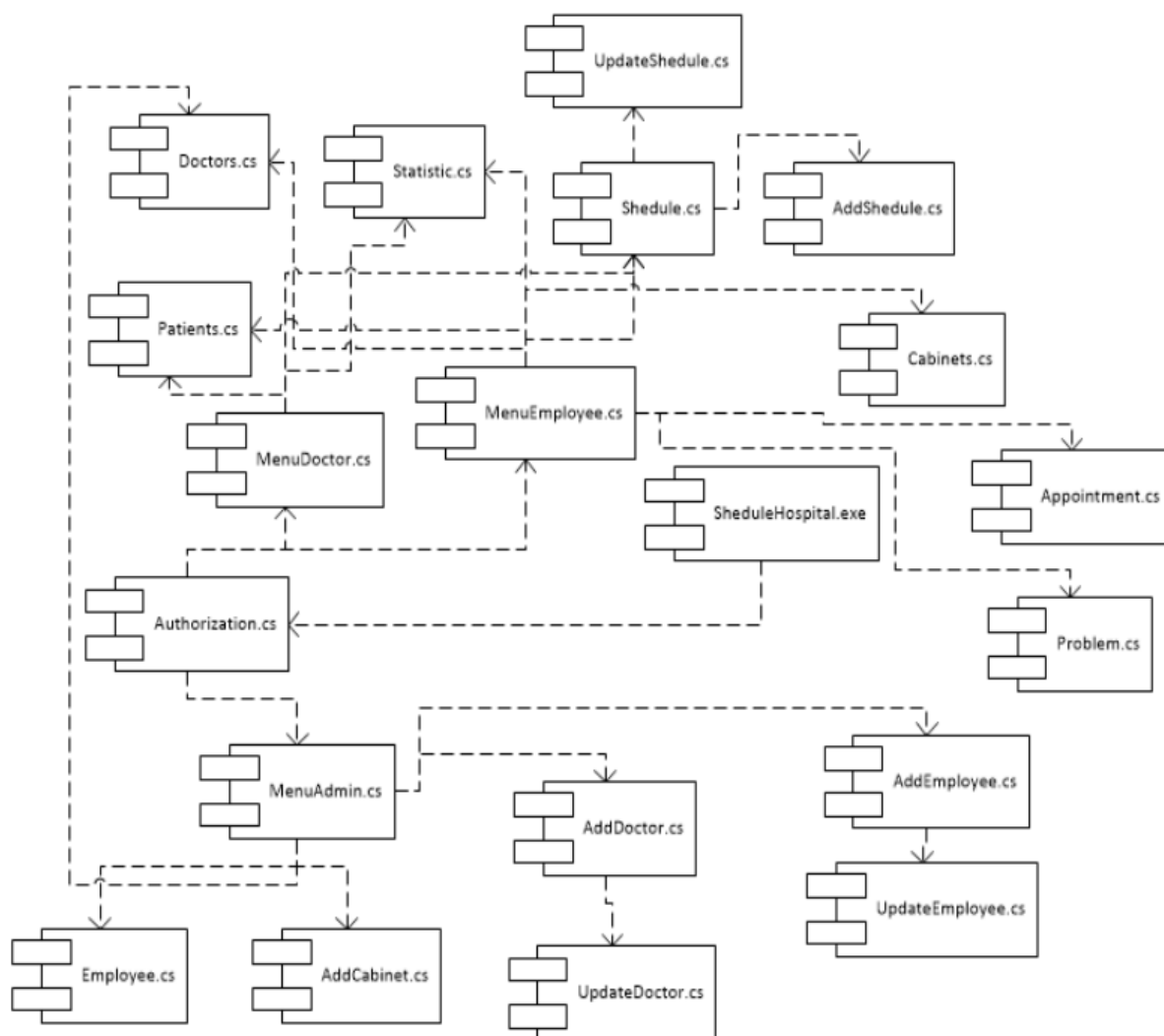


Рисунок 3.7 - Діаграма компонентів

Компонент `SheduleHospital.cs`, що зображений на діаграмі компонентів, є точкою входу в програму. Усі компоненти діаграми є класи програми, що реалізують інтерфейс користувача. Далі наведено короткий опис компонентів:

- `Authorization.cs` – реалізація вікна авторизації користувача у системі;
- `MenuDoctor.cs` – реалізація головного вікна програми, де відображається меню (для лікаря);
- `MenuEmployee.cs` – реалізація головного вікна програми, де відображається меню (для співробітника);
- `MenuAdmin.cs` – реалізація вікна головного вікна програми, де відображається меню (для адміністратора);

- Shedule.cs – реалізація вікна перегляду розкладу, вікно також містить кнопки відкриття меню додавання нового запису, вікно доступне для співробітника реєстратури та лікаря;
- Doctors.cs – реалізація вікна, що відображає дані про спеціалістів медичного закладу, вікно доступне для співробітника реєстратури;
- Patients.cs – реалізація вікна, яке відображає таблицю з інформацією про пацієнтів поліклініки, вікно доступне для співробітника реєстратури та лікаря;
- Cabinets.cs – реалізація вікна, що відображає дані про кількість кабінетів медичного закладу, вікно доступне для працівника реєстратури;
- Statistic.cs – реалізація вікна, що відображає статистичні дані про прийоми, вікно доступне для співробітника реєстратури та лікаря;
- AddShedule.cs – реалізація вікна додавання даних про нового пацієнта та запису на прийом, вікно доступне для співробітника реєстратури;
- UpdateShedule.cs – реалізація вікна зміни даних про розклад прийому пацієнтів, доступ до вікна має співробітник реєстратури;
- Appointment.cs – реалізація вікна, де відображається список прийомів (первинний чи повторний);
- Problem.cs – реалізація вікна, де відображається список поширених причин звернення до лікаря;
- Employee.cs – реалізація вікна, що відображає таблицю з інформацією про співробітників поліклініки;
- AddDoctor.cs – реалізація вікна, в якому здійснюється додавання даних про лікаря до бази, вікно доступне для адміністратора бази;
- AddEmployee.cs – реалізація вікна, у якому здійснюється додавання даних про співробітника реєстратури до бази, вікно доступне лише адміністратора;
- AddCabinet.cs – реалізація вікна, в якому здійснюється додавання даних про кабінети, вікно доступне лише адміністратора;

- UpdateDoctor.cs – реалізація вікна, в якому здійснюється зміна даних про лікаря, доступ до вікна має лише адміністратор;
- UpdateEmployee.cs – реалізація вікна, в якому здійснюється зміна даних про співробітника, доступ до вікна є лише у адміністратора. Клієнтська програма спроектована мовою програмування C# з використанням платформи .NET Framework серед програмування Microsoft Visual Studio. Серверна частина створена за допомогою СУБД Microsoft SQL Server [24]. Діаграма розгортання системи наведено рисунку 3.8.

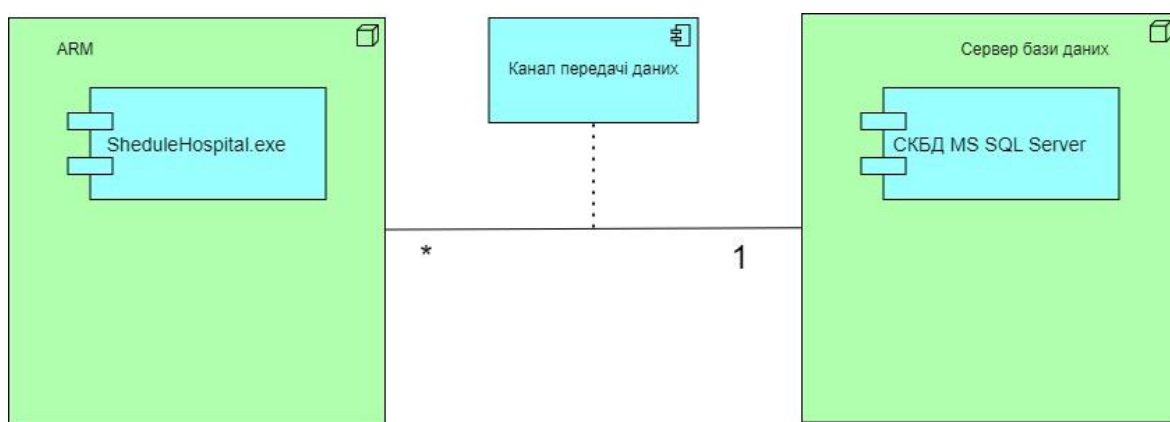


Рисунок 3.8 - Діаграма розгортання системи

На діаграмі показано, що до автоматизованого робочого місця (АРМ) прив'язаний клієнтський додаток SheduleHospital.exe. Кожне робоче місце з'єднане із серверною частиною, де знаходиться база даних, що називається Shedule_Hospital. Автоматизовані робочі місця пов'язані з базою даних зв'язком «один до багатьох», так як група комп'ютерів програмою підключено до однієї єдиної бази. Після етапу розробки необхідно провести тестування функціоналу програми.

3.7 Тестування програми

Для перевірки можливостей функціоналу та його точності та правильності роботи проводять функціональне тестування програми.

Функціональні можливості програми виділено із варіантів використання.

Для даного модуля інформаційної системи проведено один із двох відомих типів функціонального тестування – тестування «чорної скриньки». Даний тип тестування призначений для перевірки коректності роботи програмного забезпечення без доступу до коду програми.

Під час розробки модуля формування розкладів необхідно провести перевірку наступних функцій:

- введення даних про пацієнта та запис(необхідно врахувати можливість первинного та повторного запису на прийом);
- перегляд статистики щодо прийомів пацієнтів на основі даних про первинний або повторний прийом (здійснення вибірки з бази даних записів за значенням атрибуту «Найменування прийому» у таблиці «Прийом»;
- перегляд статистики щодо прийомів пацієнтів на основі даних про профіль лікаря (здійснення вибірки з бази даних за значенням атрибуту) "Профіль", розташованому в таблиці "Доктор";
- перегляд статистики щодо прийомів пацієнтів на основі даних про причину звернення (здійснення вибірки з бази даних за значенням атрибуту «Найменування проблеми» у таблиці «Проблема»;
- перегляд розкладу на поточний день (здійснення вибірки з бази даних за ініціалами лікаря та конкретною датою прийому) [25].

При тестуванні також важливо враховувати можливість захисту від неправильного введення даних, наприклад, дати. Цей факт слід врахувати на етапі розробки інтерфейсу (створення спеціальних вікон для введення дат).

У таблиці 3.11 наведено результати проведених тестів.

Таблиця 3.11 - Тестування функціоналу

№ п/п	Призначенн я тіста	Опис тесту	Результат
1	Додавання нового запису про повторний прийом	Додати дані про запис та дані про пацієнта, який записується на повторний прийом. Очікується додавання повторного запису про прийом до таблиці "Розклад".	Тест виконано успішно. Результат на рисунках Г.1, Г.2.
2	Додати дані про первинний прийом	Додати дані про запис пацієнта на первинний прийом. Очікується додавання нових записів у таблиці «Пацієнт» та "Розклад".	Тест виконано успішно. Результат рисунку Г.3, Г.4.
3	Перегляд статистики прийому	Для введених даних про тип прийому та встановлений часовий проміжок здійснюється вибірка з таблиці "Розклад"	Тест виконано успішно. Результат на Рисунку Г.5.
4	Перегляд статистики з прийому у лікаря	Для введених даних про лікаря конкретного профілю та встановлений часовий проміжок здійснюється вибірка з таблиці "Розклад"	Тест виконано успішно. Результат на риснку Г.6.
5	Перегляд статистики через звернення	Для введених даних про встановлений часовий проміжок та причини звернення здійснюється вибірка з таблиці «Розклад»	Тест виконано успішно. Результат на рисунку Г.7.

Дані тести спрямовані на перевірку роботи процедур бази даних модуля складання розкладів прийому пацієнтів у медичному закладі. Проведені тести показали точність запропонованого функціоналу та його простоту для користувачів системи.

У цьому розділі описані основні етапи розробки модуля медичної інформаційної системи, від вимог користувачів, до проектування діаграм класів та

компонентів програми. Архітектура цього програмного засобу побудована з урахуванням особливостей дослідженої предметної галузі. Цей модуль є допоміжним засобом у процесі управління обслуговуванням пацієнтів медичної організації.

За підсумками розробки проведено тестування основного функціоналу програми, внаслідок чого можна дійти невтішного висновку про ефективність розробленого кошти у процесі формування розкладів прийомів.

ВИСНОВКИ

У роботі проведено аналіз процесів обслуговування пацієнтів у медичній установі. У ході аналізу встановлено, що проблемою поточних прийомів є неконтрольовані потоки черг до лікаря. Як допоміжний засіб управління процесами обслуговування пацієнтів у поліклініці виступають медичні інформаційні системи. Проведено огляд існуючих систем, виділено їх переваги та недоліки. Встановлено, що для вирішення проблеми необхідно реорганізувати процес розподілу пацієнтів у чергах.

Для вирішення проблеми проведено аналіз відомих моделей управління обслуговуванням пацієнтів та алгоритмів управління потоками пацієнтів на обслуговування у медичній установі. У ході роботи були розроблені такі моделі та алгоритми:

- модель розподілу пацієнтів у черзі на прийом у спеціаліста медичної організації, яка, на відміну від відомих, враховує зміну робочого навантаження на спеціалістів та контролює довжину черги на обслуговування залежно від навантаження;
- алгоритм формування потоків пацієнтів на прийомі у спеціаліста в медичній організації, який, на відміну від відомих, дозволяє врахувати можливість контролю навантаження на спеціалістів та рівномірно розподілити кількість пацієнтів у чергах;
- модуль медичної інформаційної системи, що реалізує формування розкладів прийому пацієнтів.

Спроектований модуль реалізовано з використанням середовища розробки Microsoft Visual Studio, база даних створена в системі управління базами даних Microsoft SQL Server. Проведене тестування показало правильність роботи функціоналу програми та її повноту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Гулієва, І. Ф. Питання ефективності інформаційних технологій у медицині / І. Ф. Гулієва, Є. В. Рюміна, Я. І. Гулієв // Лікар та інформаційні технології. - 2011. - № 5. - С. 6 - 18.
2. Азанов, В. Г. Структурно-функціональна модель управління потоками пацієнтів регіональної консультативної поліклініки / В. Г. Азанов // Системи та засоби інформації. - 2016. - №1. - С. 13 - 29.
3. Абухай, Т. М. Моделювання, аналіз та прогнозування процесів надання кардіологічної допомоги у стаціонарі / Т. М. Абухай, С. В. Ковальчук, М. А. Балахонцева, А. В. Бухановський // Вісті вищих навчальних закладів. Приладобудування. - №8. - 2018. - С. 730 - 733.
4. Гулієва, І. Ф. Медичні інформаційні системи: витрати та вигоди / І. Ф. Гулієва, Є. В. Рюміна, Я. І. Гулієв // Лікар та інформаційні технології. - 2009. - № 3. - С. 4 - 18.
5. Гусєв, А. В. Вирішення проблеми черг пацієнтів у Карельській медичній інформаційній системі / А. В. Гусєв, Ф. А. Романов // Лікар та інформаційні технології. - 2008. - №4. - С. 46 - 53.
6. Гусєв, А. В. Огляд ринку комплексних медичних інформаційних систем / А. В. Гусєв // Лікар та інформаційні технології. - 2009. - № 6. - С. 4 - 17.
7. Таташев, А. Г. Система масового обслуговування зі змінною інтенсивністю вхідного потоку / А. Г. Таташев // Автоматика та телемеханіка. - 1995. - №12. - С. 78 - 84.
8. Хачумов, В. М. Моделювання роботи лікувального закладу як системи масового обслуговування / В. М. Хачумов, С. В. Погодін // Штучний інтелект та прийняття рішень. - 2010. - №1. - С. 49 - 56.
9. Прокоф'єва, Є.С. Удосконалення процесу обслуговування у медичних установах з урахуванням математичних моделей потоків пацієнтів //

Вища школа економіки [Електронний ресурс]. URL: <https://aspirantura.hse.com/data/2017/11/08/1158354102/2017-10-27-prokofyeva.pdf> (дата звернення: 1.09.2022).

10. Ковалевський, С. С. Моделі СМО у системах медичного обслуговування / С. С. Ковалевський, В. Б. Шувалов // Точна наука. – 2017. – №10. - С. 49 - 51.

11. Жерновий, Ю. В. Моделі системи М/М/п/г з перемиканням режимів обслуговування на момент зміни числа заявок / Ю. В. Жерновий // Інформаційні процеси. - 2010. - № 1. - С. 68 - 77.

12. Коновалов, А. А. Метод оптимізації потоків пацієнтів при організації заходів щодо надання санаторно-курортної медичної допомоги / А. А. Коновалов // Організація охорони здоров'я та громадське здоров'я. - 2011. - №5. - С. 27 - 37.

13. Чернявський, А. Л. Алгоритм для вирішення комбінаторних завдань, заснований на методі неявного перебору / А. Л. Чернявський // Автоматика та телемеханіка. - 1972. - №2. - С. 98 - 108.

14. Кормен, Т. Алгоритми: побудова та аналіз / Т. Кормен, Ч. Лейзерсон, Р. Рівест. - М: МЦНМО. 2000. - 960 с.

15. Малих, В. Л. Об'єктно-реляційний підхід до побудови сховища даних / В. Л. Малих, А. Н. Калінін, Т. Ш. Юсуфов// Програмні системи: Теоретичні основи та додатки. – 2017. – №3. - С. 169 - 187.

16. Гладков, Л. А. Генетичні алгоритми: навчальний посібник / Л. А. Гладков, В. В. Курейчик, В. М. Курейчик. - М.: Фізматліт, 2004. - 407 с.

17. Ємелін, І. В. Про стандартизацію структури електронних медичних даних / І. В. Ємелін, Б. В. Зінгерман, Г. С. Лебедєв // Інформаційно-вимірювальні та керуючі системи. - 2010. - №12. - С. 8 - 24.

18. Хомоненко, О. Д. Бази даних. Підручник для вузів/О. Д. Хомоненко, В. М. Циганков, М. Г. Мальцев. - ДМК. : Прес, 2013. - 232 с.

19. Система управління базами даних SQL Server // Програмування

мовою C#, платформа. NET Framework + [Електронний ресурс]. URL: https://professorweb.com/my/sqlserver/2012/level1/1_1.php/ (дата звернення 25.09.2022).

20. Іванова, Г. С. Об'єктно-орієнтоване програмування. / Г. С. Іванова, Є. К. Пугачов. - М.: Видавництво МДТУ імені Н.Е. Баумана, 2001. - 319 с.

21. Бахтізін, В. В. Технологія розробки програмного забезпечення / В. В. Бахтізін, Л. А. Глухова, - Мінськ БДУІР, 2010. - 408с.

ДОДАТОК А

```
CREATE TABLE Account
(
  ID_Account integer NOT NULL ,
  Login varchar(10) NOT NULL ,
  Password varchar(8) NOT NULL ,
  Initialization_Date DATETIME NOT NULL ,
  Account_Type varchar(64) NOT NULL
)
go
ALTER TABLE Account
ADD CONSTRAINT XPK Account PRIMARY KEY NONCLUSTERED (ID_Account ASC)
go
CREATE TABLE Appointment
(
  ID_Appointment integer NOT NULL ,
  Name_Appointment varchar(20) NOT NULL
)
go
ALTER TABLE Appointment
ADD CONSTRAINT XPK Appointment PRIMARY KEY NONCLUSTERED (ID_Appointment
ASC)
go
CREATE TABLE Cabinet
(
  ID_Cabinet integer NOT NULL ,
  Floor integer NOT NULL
)
go
ALTER TABLE Cabinet
ADD CONSTRAINT XPK Cabinet PRIMARY KEY NONCLUSTERED (ID_Cabinet ASC)
go
CREATE TABLE Doctor
(
  ID_Doctor integer NOT NULL ,
  Profile varchar(128) NOT NULL ,
  Workload real NOT NULL ,
  Name_Lastname_Patro varchar(128) NOT NULL ,
  ID_Cabinet integer NULL ,
```

```

ID_Account integer NULL
)
go
ALTER TABLE Doctor
ADD CONSTRAINT XPKDoctor PRIMARY KEY NONCLUSTERED (ID_Doctor ASC)
go
CREATE TABLE Employee
(
ID_Employee integer NOT NULL ,
Name_LastName_Patro varchar(128) NOT NULL ,
ID_Account integer NULL
)
go
ALTER TABLE Employee
ADD CONSTRAINT XPKEmployee PRIMARY KEY NONCLUSTERED (ID_Employee ASC)
go
CREATE TABLE Patient
(
ID_Patient integer NOT NULL ,
Name varchar(128) NOT NULL ,
LastName varchar(128) NOT NULL ,
Patronymic varchar(128) NOT NULL ,
Birthday datetime NOT NULL ,
Policy numeric(17) NOT NULL
)
go
ALTER TABLE Patient
ADD CONSTRAINT XPKPatient PRIMARY KEY NONCLUSTERED (ID_Patient ASC)
go
CREATE TABLE Problem
(
ID_Problem integer NOT NULL ,
Name_Problem varchar(128) NOT NULL ,
Description varchar(128) NOT NULL ,
Complexity real NOT NULL
)
go
ALTER TABLE Problem
ADD CONSTRAINT XPKProblem PRIMARY KEY NONCLUSTERED (ID_Problem ASC)
go
CREATE TABLE Shedule
(
ID_Schedule integer NOT NULL ,

```

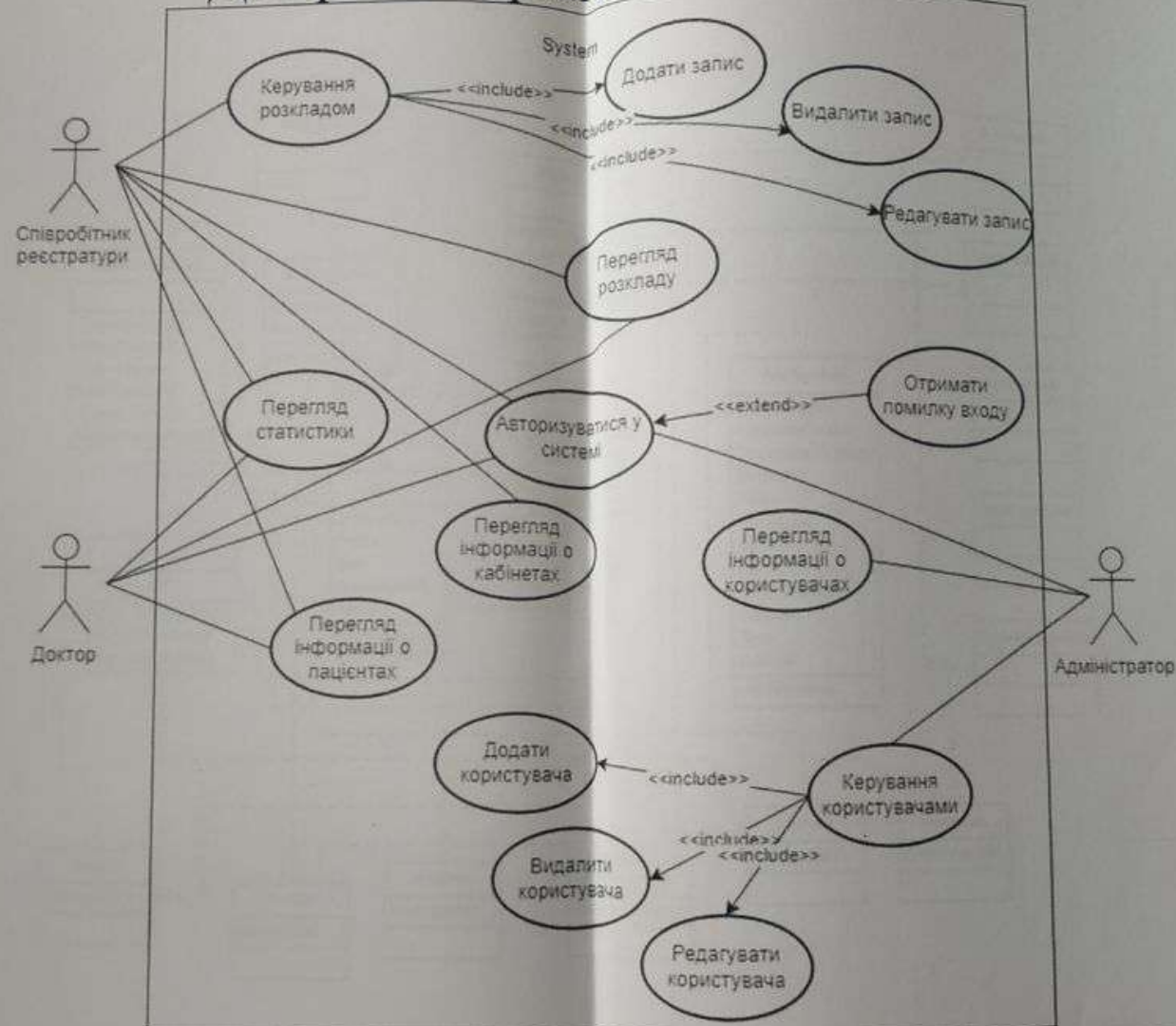
```

Date datetime NOT NULL ,
Time datetime NOT NULL ,
ID_Patient integer NULL ,
ID_Doctor integer NULL ,
ID_Cabinet integer NULL ,
ID_Problem integer NULL ,
ID_Appointment integer NULL
)
go
ALTER TABLE Shedule
ADD CONSTRAINT XPKShedule PRIMARY KEY NONCLUSTERED (ID_Schedule ASC)
go
ALTER TABLE Doctor
ADD CONSTRAINT R_1 FOREIGN KEY (ID_Cabinet) REFERENCES
Cabinet(ID_Cabinet)
ON DELETE NO ACTION
ON UPDATE NO ACTION
go
ALTER TABLE Doctor
ADD CONSTRAINT R_4 FOREIGN KEY (ID_Account) REFERENCES
Account(ID_Account)
ON DELETE NO ACTION
ON UPDATE NO ACTION
go
ALTER TABLE Employee
ADD CONSTRAINT R_2 FOREIGN KEY (ID_Account) REFERENCES
Account(ID_Account)
ON DELETE NO ACTION
ON UPDATE NO ACTION
go
ALTER TABLE Shedule
ADD CONSTRAINT R_5 FOREIGN KEY (ID_Patient) REFERENCES
Patient(ID_Patient)
ON DELETE NO ACTION
ON UPDATE NO ACTION
go
ALTER TABLE Shedule
ADD CONSTRAINT R_6 FOREIGN KEY (ID_Doctor) REFERENCES Doctor(ID_Doctor)
ON DELETE NO ACTION
ON UPDATE NO ACTION
go
ALTER TABLE Shedule
ADD CONSTRAINT R_7 FOREIGN KEY (ID_Cabinet) REFERENCES

```

```
Cabinet(ID_Cabinet)
ON DELETE NO ACTION
ON UPDATE NO ACTION
go
ALTER TABLE Shedule
ADD CONSTRAINT R_8 FOREIGN KEY (ID_Problem) REFERENCES
Problem(ID_Problem)
ON DELETE NO ACTION
ON UPDATE NO ACTION
go
ALTER TABLE Shedule
ADD CONSTRAINT R_9 FOREIGN KEY (ID_Appointment) REFERENCES
Appointment(ID_Appointment)
ON DELETE NO ACTION
ON UPDATE NO ACTION
go
```

Діаграма варіантів використання



Зам.	Лист.	№ докум.	Підп.	Дата	
Розроб.		Демченко А.І.			
Перев.		Ільїнченко М.Б.			
Т.контр.					
Н.контр.		Щербак Н.В.			
Затв.		Кузьмичов Р.К.			

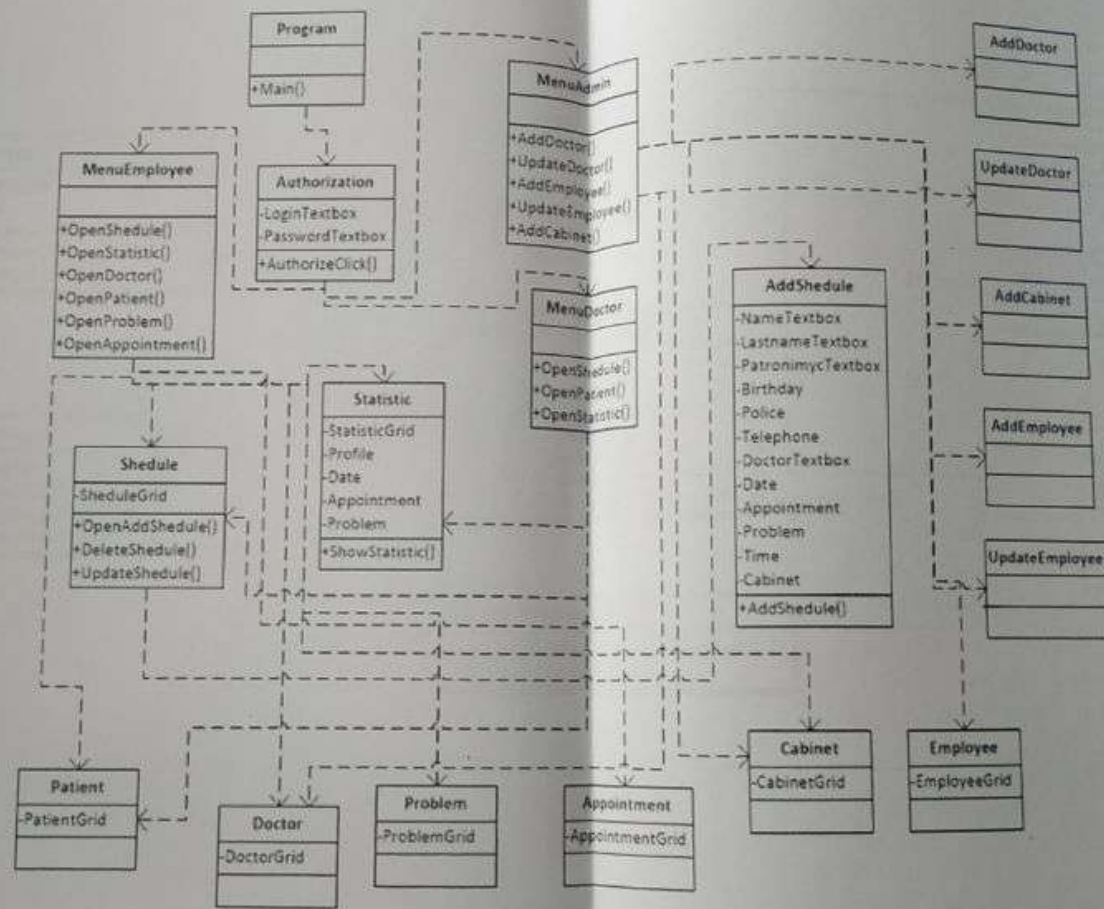
13.02070849.00003 ПЛ1

Діаграма варіантів використання
Система керування обслуговуванням
пацієнтів медичного закладу

Лист	Маса	Масштаб
Лист 1		Листів 1

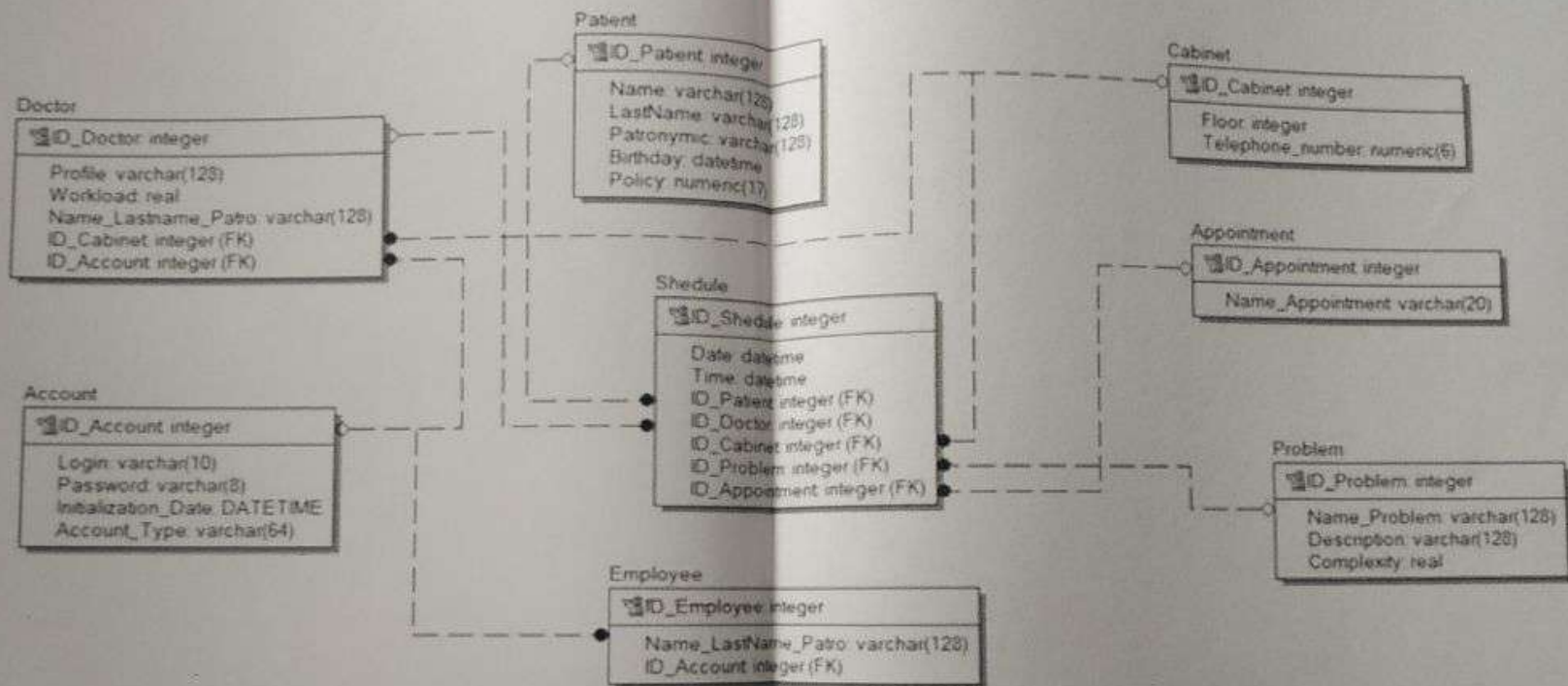
НУ «Запорізька політехніка»
КНТ-511М

Діаграма класів



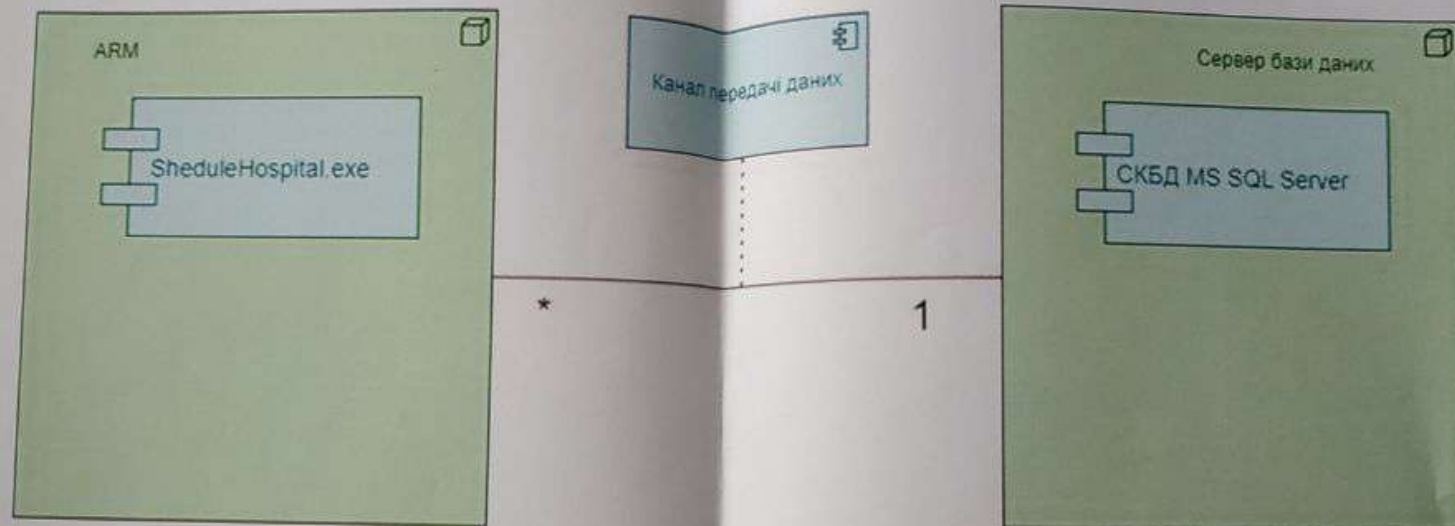
					13.02070849.00003 ПЛ2		
Зам.	Лист.	№ докум.	Підп.	Дата	Діаграма класів Система керування обслуговуванням пацієнтів медичного закладу		
Розроб.	Демченко А.І.						
Перев.	Ільченко М.Б.						
Т.контр.							
Н.контр.	Шерба Н.В.						
Затв.	Кузьменко Р.К.				Лист 1 Листів 1 НУ «Запорізька політехніка» КНТ-511М		

Фізична модель бази даних



13.02070849.00003 ПЛЗ										
Фізична модель бази даних Система керування обслуговуванням пацієнтів медичного закладу										
Зам.	Лист.	М. докум.	Підп.	Датум						
Розроб.	Демченко А.І.									
Перев.	Пилипенко М.Б.									
Т.контр.										
Н.контр.	Щербак Н.В.									
Зам.	Хуарестов Р.К.									
<table border="1"> <tr> <td>Лист</td> <td>Маск</td> <td>Маск</td> </tr> <tr> <td>Лист 1</td> <td>Лист 1</td> <td></td> </tr> </table>					Лист	Маск	Маск	Лист 1	Лист 1	
Лист	Маск	Маск								
Лист 1	Лист 1									
НУ «Запорізька політехніка» КНТ-511м										

Діаграма розгортання системи



					13.02070849.00003 ПЛ4			
					Діаграма розгортання системи Система керування обслуговуванням пацієнтів мелічного закладу			
Зам.	Лист.	№ докум.	Підп.	Дата				
Розроб.		Демченко А.І.						
Перев.		Ільченко М.Б.						
Т.контр.								
Н.контр.		Шербак Н.В.						
Затв.		Кузнецов Р.К.						
					Лист 1	Маса	Магн.т.	
					Листів 1			
					НУ «Запорізька політехніка» КНТ-511м			