НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ І
СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ

«На правах рукопису»
УДК 004.38

«До захисту допущено»
Завідувач кафедри СПСКС

_____ В.П.Тарасенко_
(підпис) (ініціали, прізвище)

"___"_____2018р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 123 Комп'ютерна інженерія
Комп'ютерні системи та компоненти

на тему ВЕБ-ОРІЄНТОВАНА СИСТЕМА ДЛЯ АВТОМАТИЗАЦІЇ РОБОТИ СТУДЕНТСЬКОЇ ПОЛІКЛІНІКИ КПІ ім. ІГОРЯ СІКОРСЬКОГО
(ПРОГРАМНА ЧАСТИНА)

Виконав: студент ІІ курсу, групи КВ-71мп
(шифр групи)

Тіменко Антон Олександрович __
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник к.т.н., доцент Клятченко Я.М. _
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент_____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018 року

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп‘ютерних систем

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність (спеціалізація) – 123 Комп’ютерна інженерія

Комп’ютерні системи та компоненти

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ В. П. Тарасенко

«___»_____2018 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Тіменку Антону Олександровичу

1. Тема дисертації «Веб-орієнтована система для автоматизації роботи студентської поліклініки КПІ ім. Ігоря Сікорського (програмна частина)», науковий керівник дисертації Клятченко Ярослав Михайлович, к.т.н., доцент, затверджені наказом по університету від «30» жовтня 2018 р. №4030-с

2. Термін подання студентом дисертації «__» грудня 2018 р.

3. Об’єкт дослідження є методи та системи автоматизації роботи медичних закладів.

4. Предмет дослідження є апаратна реалізація розробленої системи автоматизації роботи медичних закладів.

5. Перелік завдань, які потрібно розробити:

- провести аналіз існуючих систем автоматизації медичних закладів;
- дослідити програмні модулі для створення програмної частини системи;
- розробити клієнт-серверну архітектуру програмних компонентів;
- розробити програмне забезпечення для демонстрації роботи системи;
- проаналізувати результати роботи прототипу.

6. Перелік ілюстрованого матеріалу: презентація

6. Перелік публікацій:

- Тези доповіді «Прикладна математика та комп'ютинг» ПМК-2018-2
- Тези доповіді на V-й Міжнародній науково-технічній Internet-конференції НУХТ, АКС 2018

7. Дата видачі завдання «04» жовтня 2017 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Ґрунтовне ознайомлення з предметною галуззю	25.10.2017	
2.	Визначення структури магістерської дисертації; вивчення літератури, пошук додаткової літератури, патентний пошук	17.03.2018	
3.	Робота над першим розділом магістерської дисертації; проведення наукового дослідження	25.04.2018	
4.	Проведення наукового дослідження; робота над другим розділом магістерської дисертації; розроблення програмного забезпечення	24.05.2018	
5.	Проведення наукового дослідження; робота над статтею за результатами наукового дослідження	16.06.2018	
6.	Проведення наукового дослідження; робота над третім розділом магістерської дисертації	17.07.2018	
7.	Завершення роботи над основною частиною магістерської дисертації; підготовка матеріалів доповіді на конференції ПМК-2018	21.10.2018	
8.	Оформлення текстової і графічної частини магістерської дисертації	15.11.2018	

Студент А. О. Тіменко

Науковий керівник дисертації Я. М. Клятченко

РЕФЕРАТ

Актуальність теми. Візит до медичних закладів, зазвичай, пов‘язаний з певними незручностями, наприклад відсутність інформації у мережі інтернет про лікарів, які обслуговують хворих, відсутність у мережі графіку роботи лікарів, а також великі черги, у яких відвідувачам часто необхідно чекати годинами, щоб потрапити до лікаря. Крім того, графік роботи лікарів постійно змінюється, і інформація про ці зміни не публікується у мережі.

Усі ці проблеми може вирішити впровадження у медичних закладах веб-орієнтованої медичної системи, яка автоматизує певні процеси роботи лікарів, автоматизує керування чергами до кабінетів, а також надає усю актуальну інформацію про роботу медичного закладу у мережі інтернет. Система повинна мати зручний веб-інтерфейс з усією актуальною інформацією про поліклініку та програмно-апаратний комплекс, що дозволяє формалізувати та оптимізувати управління потоком відвідувачів — електронну чергу.

Об’єктом дослідження є системи автоматизації роботи медичних закладів.

Предметом дослідження є програмна реалізація системи автоматизації роботи медичних закладів.

Мета дослідження: розробка системи враховуючи досвід інших медичних установ та потреби самих відвідувачів медичних закладів.

Практична цінність розробленої системи полягає у застосуванні розробленої системи у студентській поліклініці КПІ для оптимізації її роботи. Це дозволить позбавитися від живих черг під кабінетами лікарів і дати можливість відвідувачам спланувати час відвідування установи заздалегідь.

Наукова новизна полягає в аналізі процесів у медичних установах, дослідженні способів та методів їх оптимізації та розробці програмної

складової медичної інформаційної системи на основі отриманих результатів.

Апробація роботи. Основні положення і результати роботи були представлені та обговорювались на XI науковій конференції молодих вчених «Прикладна математика та комп'ютинг» ПМК-2018-2 (Київ, 14-16 листопада 2018 р.) та на V Міжнародній науково-технічній Internet-конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами» в НУХТ на базі факультету АКС..

Структура та обсяг роботи. Магістерська дисертація складається з вступу, чотирьох розділів та висновків.

У вступі подано загальну характеристику роботи, обґрунтовано актуальність напрямку досліджень, сформульовано мету і задачі досліджень, показано наукову новизну отриманих результатів і практичну цінність роботи.

У першому розділі виконано оцінку сучасного стану проблеми, обґрунтовано актуальність напрямку досліджень, сформульовано мету дослідження, розглянуто існуючі рішення для автоматизації роботи медичних закладів.

У другому розділі описані необхідні програмні модулі для реалізації системи автоматизації роботи медичних закладів, виконана характеристика необхідних протоколів зв'язку програмної та апаратної частини.

У третьому розділі розроблено та описано алгоритми для роботи системи автоматизації роботи медичних закладів, розроблено та описано алгоритм роботи модулю керування електронною чергою.

У четвертому розділі проаналізовано розроблену систему та описані інтерфейси користувача для роботи з системою та для керування нею.

У висновках представлені результати проведеної роботи.

Магістерська дисертація виконана на __ аркушах, містить__ додатків та посилання на список використаних літературних джерел з __ найменувань. У магістерській дисертації наведено __ рисунків та __ таблиць

Ключові слова: веб-орієнтована система, електронна черга.

РЕФЕРАТ

Актуальность темы. Визит в медицинские учреждения, как правило, связаны с определенными неудобствами, например отсутствие информации в сети интернет о врачах, которые обслуживают больных, отсутствие в сети графике работы врачей, а также большие очереди в которых посетителям часто необходимо ждать часами, чтобы попасть к врачу. Кроме того, график работы врачей постоянно меняется, и информация об этих изменениях не публикуется в сети.

Все эти проблемы может решить внедрение в медицинских учреждениях веб-ориентированной медицинской системы, которая автоматизирует определенные процессы работы врачей, автоматизирует управление очередями в кабинеты, а также предоставляет всю актуальную информацию о работе медицинского учреждения в сети Интернет. Система должна иметь удобный веб-интерфейс со всей актуальной информацией о поликлинике и программно-аппаратный комплекс, позволяющий формализовать и оптимизировать управление потоком посетителей - электронную очередь.

Объектом исследования являются системы автоматизации работы медицинских учреждений.

Предметом исследования является программная реализация системы автоматизации работы медицинских учреждений.

Цель исследования: разработка системы учитывая опыт других медицинских учреждений и потребности самих посетителей медицинских учреждений.

Практическая ценность разработанной системы заключается в применении разработанной системы в студенческой поликлинике КПИ для оптимизации ее работы. Это позволит избавиться от живых очередей под

кабинетами врачей и даст возможность посетителям спланировать время посещения учреждения заранее.

Научная новизна заключается в анализе процессов в медицинских учреждениях, исследовании способов и методов их оптимизации и разработке программной составляющей медицинской информационной системы на основе полученных результатов.

Апробация работы. Основные положения и результаты работы были представлены и обсуждались на XI научной конференции молодых ученых «Прикладная математика и компьютинг» ПМК-2018-2 (Киев, 14-16 ноября 2018) и на V Международной научно-технической Internet-конференции « современные методы, информационное, программное и техническое обеспечение систем управления организационно-техническими и технологическими комплексами »в НУХТ на базе факультета АКС.

Структура и объем работы. Магистерская диссертация состоит из введения, четырех глав и выводов.

Во введении представлена общая характеристика работы, обоснована актуальность направления исследований, сформулированы цели и задачи исследований, показано научную новизну полученных результатов и практическую ценность работы.

В первом разделе выполнена оценка современного состояния проблемы, обоснована актуальность направления исследований, сформулированы цель исследования, рассмотрены существующие решения для автоматизации работы медицинских учреждений.

Во втором разделе описаны необходимые программные модули для реализации системы автоматизации работы медицинских учреждений, выполненна характеристика необходимых протоколов связи программной и аппаратной части.

В третьем разделе разработаны и описаны алгоритмы для работы системы автоматизации работы медицинских учреждений, разработаны и описаны алгоритм работы модуля управления электронной очередью.

В *четвертом разделе* проанализированы разработанную систему и описаны интерфейсы для работы с системой и для управления ею.

В выводах представлены результаты проведенной работы.

Магистерская диссертация выполнена на __ листах, содержит__ приложений и ссылки на список использованных литературных источников из __ наименований. В магистерской диссертации приведены __ рисунков и __ таблиц

Ключевые слова: веб-ориентированная система, электронная очередь.

ABSTRACT

Relevance of the topic. A visit to medical institutions is usually associated with certain inconveniences, such as the lack of information on the Internet about doctors who serve patients, the lack of doctors' schedule on the network, as well as long queues in which visitors often need to wait hours to get to the doctor. In addition, the work schedule of doctors is constantly changing, and information about these changes is not published online.

All these problems can be solved by the web-based medical system for medical institutions, which automates certain processes of doctors, automates the management of queues to offices, and also provides all relevant information about the work of the medical institution on the Internet. The system should have a convenient web interface with all relevant information about the clinic and a software and hardware system that allows to formalize and optimize the management of the flow of visitors - an electronic queue.

The object of the research is the automation systems of medical institutions.

The subject of research is the software implementation of the automation system of medical institutions.

Objective: to develop a system taking into account the experience of other medical institutions and the needs of the visitors of medical institutions themselves.

The practical value of the developed system is to apply the developed system in the KPI student polyclinic to optimize its work. This will get rid of live queues under the doctors' offices and will give visitors the opportunity to plan the time of their visit to the institution in advance.

The scientific novelty consists in the analysis of processes in medical institutions, the study of methods for their optimization and the development of

the program component of a medical information system based on the results obtained.

Approbation of work. The main provisions and results of the work were presented and discussed at the XI scientific conference of young scientists "Applied Mathematics and Computing" PMK-2018-2 (Kiev, November 14-16, 2018) as well as at the 5th International Conference at ACS NUFT (November 24-23, 2018, Kyiv, Ukraine).

Structure and scope of work. The master thesis consists of introduction, four chapters and conclusions.

The introduction presents a general description of the work, the relevance of the research direction is substantiated, the goals and objectives of the research are formulated, the scientific novelty of the results obtained and the practical value of the work are shown.

In the first section, the current state of the problem was assessed, the relevance of the research directions was substantiated, the research goal was formulated, and existing solutions for automating medical institutions were considered.

The second section describes the necessary software modules for the implementation of the automation system of medical institutions, the characteristics of the necessary software and hardware communication protocols are made.

The third section has developed and described algorithms for the operation of the automation system of medical institutions, developed and described an algorithm for the operation of the electronic queue management module.

The fourth section analyzes the developed system and describes the interfaces for working with the system and for managing it.

The findings present the results of this work.

The master's thesis is made on __ sheets, contains __ applications and links to the list of references used from __ titles. The master’s thesis contains __ figures and __ tables

Keywords: web-based system, electronic queue.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

ІС – Інформаційна система

МІС – Медична інформаційна система

СУБД – Система управління базами даних

АРМ – Автоматизоване робоче місце

MVP – Шаблон Model-View-Presenter

Python – Мова програмування

Django – Веб-фреймворк

API - Application Programming Interface (Прикладний програмний інтерфейс)

UI – User interface (Інтерфейс користувача)

TDD – Test driven development (Керована тестами розробка)

HTML - HyperText Markup Language (Мова розмітки гіпертекстових документів)

JavaScript – Мова програмування

NGINX – веб-сервер

ВСТУП

Студентська поліклініка КПІ імені Ігоря Сікорського – медична установа, яку хоча б один раз відвідував кожен студент КПІ. Візит до студентської поліклініки, зазвичай, пов‘язаний з певними незручностями, наприклад відсутність інформації у мережі інтернет про лікарів, які обслуговують певні факультети, відсутність у мережі графіку роботи лікарів, а також великі черги, у яких студенту часто необхідно чекати годинами, щоб потрапити до лікаря. Крім того, графік роботи лікарів постійно змінюється, і інформація про ці зміни не публікується у мережі.

Усі ці проблеми може вирішити впровадження у студентській поліклініці КПІ імені Ігоря Сікорського веб-орієнтованої електронної медичної системи, яка автоматизує певні процеси роботи лікарів, автоматизує керування чергами до кабінетів, а також надає всю актуальну інформацію про роботу студентської поліклініки у мережі інтернет. Система повинна мати зручний веб-інтерфейс з усією актуальною інформацією про поліклініку та програмно-апаратний комплекс, що дозволяє формалізувати та оптимізувати управління потоком відвідувачів — електронну чергу.

У цій магістерській дисертації представлено реалізацію такої системи враховуючи досвід інших медичних установ та потреби самих відвідувачів студентської поліклініки. Це дозволить позбавитися від живих черг під кабінетами лікарів і дати можливість відвідувачам спланувати час відвідування установи заздалегідь.

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА АКТУАЛЬНІСТЬ МЕДИЧНИХ АВТОМАТИЗОВАНИХ СИСТЕМ

Актуальність медичних автоматизованих систем

Актуальність створення інформаційної системи в поліклініці обумовлена сьогодні необхідністю використання великих, і постійно зростаючих, обсягів інформації при вирішенні діагностичних, терапевтичних, статистичних, управлінських та інших завдань.

Ні для кого не секрет, що велика частина прийому йде не на вирішення клінічних питань, а на супровідну і далеко не саму основну роботу - оформлення поліклінічних талонів та іншої звітної документації, записів у амбулаторній карті або історії хвороби, призначень консультацій або обстеження і т. д. Вже не викликає сумнівів, що найбільш ефективним інструментів для полегшення праці медичних працівників та підвищення його ефективності є комп'ютерні технології. Автоматизація здатна не просто полегшити роботу, вона повинна звільнити персонал від рутини і дати йому принципово новий інструмент, який прямо або побічно, але призведе до скорочення нецільових витрат інтелектуального багажу, реалізації бажання працювати і займатися саме медициною.

Перше знайомство відвідувачів з поліклінікою починається в реєстратурі. Вона є основним структурним підрозділом по реалізації прийому хворих в поліклініці. Від організації роботи реєстратури залежить в значній мірі ритмічність роботи всіх підрозділів поліклініки, забезпечення найбільш оптимального розподілу потоків відвідувачів і зменшення витрат часу хворих на відвідування поліклініки.

Метою даної роботи є розробка програмного комплексу для поліклініки, що дозволяє підвищити ефективність роботи реєстратури за рахунок скорочення тимчасових і трудових витрат, також підвищення якості роботи.

Основна діяльність реєстратури полягає у видачі талонів на прийом до лікаря і запис на різні процедури.

Основними завданнями реєстратури поліклініки є:

- організація попередньої і поточної записи хворих на прийом до лікаря;
- забезпечення регулювання інтенсивності потоку відвідувачів для рівномірного навантаження лікарів;
- своєчасний підбір і доставка медичної документації в кабінети лікарів, правильне ведення і зберігання картотеки.

Об'єднання інформації в загальне сховище даних гарантує забезпечення цілісності даних, можливість розподіленого і одночасного доступу до них. Також створення бази даних призведе до стійкої формалізації даних і зменшення паперового документообігу між відділами.

Економічна ефективність роботи полягає в наступному:

- збільшення кількості обслуговуваних пацієнтів;
- скорочення часу на оформлення медичної документації;
- зменшення числа можливих людських помилок.

1.1. Загальна характеристика медичних інформаційних систем

В Україні протягом останніх п’яти років спостерігається незначна позитивна динаміка розвитку ринку медичних інформаційних систем. Слід відзначити, що цей ринок є ризикованим, затрати на розробку сучасних систем значно переважають економічну ефективність від присутності на ньому, тому нові розробники з‘являються рідко.

Значними перешкодами на шляху до інформатизації вітчизняної системи охорони здоров’я є недостатнє фінансування медичних закладів, відсутність у керівництва розуміння можливостей, які дають високі технології для підвищення якості роботи медичних закладів, та недостатня стандартизація даних та способів її обробки.

Разом з цим система охорони здоров’я стикається з такими проблемами, як:

- підвищення вимог населення до систем охорони здоров’я;
- обмежений бюджет;
- часта зміна місця проживання пацієнтів та ін.

Застосування комплексних інформаційних систем, які дозволять організувати управління медичними закладами на сучасному рівні, суттєво підвищить не тільки якість лікування і рівень медичних послуг, але й ефективність, тобто рентабельність, використання медичних ресурсів.

Незадовільною є ситуація з інформуванням міських управлінь охорони здоров’я, санепідемстанцій та інших установ про епідеміологічну ситуацію чи поточний стан захворюваності, та наявність вільних ліжок в лікарнях тощо. Через відсутність сучасної техніки, програмного забезпечення та засобів зв’язку така інформація є неповною і запізнілою, що не дає можливості оперативно та адекватно попереджати загрози, а також реагувати на проблеми, які виникають у роботі медичних закладів. Більшість медичних інформаційних систем, які функціонують у лікувальних закладах в даний час, є морально і фізично застарілими. Переважно вони розроблені ще 10-15 років тому, і їх ніхто вже давно не підтримує і не удосконалює.

Інформаційні процеси (наприклад пошук, збирання, зберігання, передавання, опрацювання, використання, захист інформації) присутні у всіх областях медицини і галузі охорони здоров’я. Важливою складовою інформаційних процесів є інформаційні потоки. Від їх впорядкованості залежить чіткість функціонування галузі в цілому і ефективність управління нею. Потоки починаються в місцях виникнення інформації і забезпечують її доставку до місць прийняття рішень. Вони складаються з окремих повідомлень, відображених у сигналах і документах, і рухаються в просторі і часі від джерела інформації до одержувача. Для роботи з інформаційними потоками призначені інформаційні системи (ІС).

Інформаційною системою (ІС) будемо називати організаційно впорядковану сукупність документів (масивів документів) та інформаційних комп'ютеризованих технологій (з використанням засобів обчислювальної техніки і зв'язку), що реалізує інформаційні процеси. ІС розрізняють за галузями застосування.

Обробка інформації в інформаційній системі може здійснюватись ручним, механічним, автоматизованим і автоматичним способами. З появою комп'ютерів відбулась революція в процесах обробки інформації, виникли нові інформаційні технології в медицині і системі охорони здоров'я. Процес залучення нових інформаційних технологій в систему охорони здоров'я і медицину зазвичай називають інформатизацією системи охорони здоров'я.

Інформатизація – це реалізація комплексу заходів, спрямованих на забезпечення повного і своєчасного використання достовірних знань у всіх видах людської діяльності. Інформатизація системи охорони здоров'я – одна із складових цього процесу. При цьому метою інформатизації є прогрес в системі охорони здоров'я в напрямках розвитку самої служби так і контролю за станом здоров'я її пацієнтів. Технологічною і технічною основою інформатизації є досить потужна мережа інформаційних структур, орієнтованих як на медичних працівників так і на населення.

Моніторинг здоров'я – це система оперативного стеження за станом і змінами здоров'я населення, що представляє собою механізм отримання різнорівневої інформації для поглибленого оцінювання і прогнозування здоров'я населення за різні часові інтервали, тощо. Головною метою створення системи моніторингу здоров'я населення є організація на базі нових комп'ютерних технологій державної міжгалузевої системи збору, обробки, збереження і подання інформації, що забезпечує динамічну оцінку суспільного здоров'я та інформаційну підтримку прийняття рішень, направлених на їх покращення.

Інформаційні технології, що використовуються в процесах профілактики, лікування, діагностики та управління охороною здоров'я, є одним із основних об'єктів стандартизації в системі охорони здоров'я та вимагають уніфікації службових документів, основних термінів та понять; єдиного підходу до лікувального процесу.

Можна виділити такі аспекти інформатизації медичної діяльності:

- медичний;
- технічний;
- технологічний;

психолого-педагогічний;

Медичний аспект полягає у відповідній підготовці медичних, даних і знань (формалізація, єдність термінології, стандартизація), створенні інтерфейсу загальної структури інформаційної бази, побудові математичних моделей медико-біологічних процесів (фізіологічних і патологічних) і т. ін. Розробка теоретичних моделей подання даних і знань для вирішення відповідних медичних задач і конкретна програмно-апаратна реалізація інформаційної бази на основі розроблених моделей становлять технічний аспект проблеми.

Технологічний аспект означає узгодження побудованої технічної системи з технологічною схемою лікувально-діагностичного процесу (образно кажучи, певний «рецепт вживлення» системи у лікувально-діагностичний процес). І, нарешті, психолого-педагогічний аспект передбачає відповідну підготовку медичного персоналу.

Задачі, що вирішуються за допомогою комп'ютерних технологій в медицині та охороні здоров'я, надзвичайно різноманітні. Вони розрізняються як за цілями і змістом, так і за напрямами та рівнем використання. Умовно їх можна розділити на такі:

інформаційна підтримка роботи медичного персоналу;

- інформаційне забезпечення термінової допомоги у надзвичайних ситуаціях;

- моніторинг рівня здоров’я населення;
- інформаційне забезпечення наукової роботи;

Експериментальні дослідження процесів розумової діяльності людини показали, що 85% часу іде на пошук необхідної інформації, друк, побудову графіків тощо – тобто створення передумов для розумової роботи. Це стосується й роботи медичного дослідника або практикуючого лікаря.

Для автоматизації робіт на кожному з етапів діагностично-лікувального процесу застосовують медичні інформаційні системи (МІС). Відомі, щонайменше, два підходи до визначення МІС. У широкому розумінні під медичною інформаційною системою розуміють форму організації діяльності в медицині й охороні здоров’я, що поєднує медиків, математиків, інженерів, техніків з комплексом технічних засобів і забезпечує збір, збереження, переробку і видачу медичної інформації різного профілю в процесі рішення визначених задач медицини й охорони здоров’я.

У вузькому розумінні медичною інформаційною системою називають комплекс технічних засобів і математичного забезпечення, призначений для збору, аналізу медико-біологічної інформації і видачі результатів у зручному для користувача вигляді.

Таким чином, можна дати наступне визначення: МІС – це програмнотехнічний комплекс, що готує і забезпечує процеси збирання, зберігання і обробку інформації в медицині й галузі охорони здоров’я.

Основною метою медичних ІС є інформаційна підтримка різноманітних задач надання медичної допомоги населенню, управління медичними закладами і інформаційному забезпеченні системи охорони здоров’я. Окремим завданням є інформаційна підтримка наукових досліджень, навчальної та атестаційної роботи.

Створення медичної інформаційної системи переслідує кілька цілей:

- підвищення якості діяльності медичних працівників і установ охорони здоров'я шляхом організації досконалої (відповідної рівню використовуваних технічних засобів);
- обробки медичної інформації, у тому числі шляхом удосконалювання процесів керування і планування;
- полегшення праці медичних працівників, ліквідація трудомістких малоефективних процесів ручної обробки й аналізу медичних даних;
- забезпечення ефективного обміну інформацією з іншими інформаційними системами.

За призначенням МІС класифікують на:

- системи, основною функцією яких є накопичення даних (автоматизовані системи обробки даних та (або) інформації, автоматизовані інформаційні та інформаційно-довідкові системи);
- діагностичні та консультаційні системи;
- системи, що забезпечують медичне обслуговування;

Найбільш загальні задачі МІС, що вирішуються у клінічних установах:

- об'єктивізація трактування результатів досліджень (по деяким даним, невірне тлумачення результатів рентгенологічного, електрокардіологічного і лабораторних досліджень приводить у 30% випадків до помилкового діагнозу);
- автоматизація обробки інформації на етапі попередньої роботи медичного персоналу по визначенню діагнозу і виробленню тактики лікування (лікар приймає остаточне рішення з питань діагностики і лікування хворого);
- автоматизація лабораторних досліджень: біохімічних, електрофізіологічних, рентгенорадіологічних тощо;

- створення баз (банків) даних: накопичення відомостей про кожного хворого для подальшого аналізу матеріалу, організації обробки цієї інформації відповідним математичним забезпеченням (у тому числі системами управління базами даних – СУБД);
- створення баз знань: накопичення знань експертів в області медицини й системи охорони здоров'я, необхідних для розробки експертних систем діагностики, лікування і реабілітації, профілактичних оглядів, експертизи, планування і управління;

упорядкування потоку інформації усередині медичної установи (задачі організаційного керування, задачі кадрові, матеріальнотехнічного постачання, статистичні звіти, оцінка діяльності відділень лікарень по деяких показниках тощо);

Організація МІС єдина тільки в медичному плані: стандартизація медичної інформації, форми медичної документації, прийнята система кодування, математичні моделі, тобто те, що становить медичний аспект проблеми. У технічному плані МІС дуже різнорідні, що означає програмну та апаратну несумісність існуючих МІС (використання машин різних поколінь, різні мови програмування, різні організації внутрішньо машинних баз даних і т. ін.). Це, безумовно, обмежує масове впровадження МІС у практику роботи лікувальних закладів.

Тому, на сучасному етапі відбувається перехід від окремих інформаційних систем до інформаційних середовищ. Інформаційні медичні середовища – це якісно нова форма організації обміну інформацією у медицині, яка дає можливість інтегрувати в рамках єдиного технологічного процесу МІС різних класів, що пронизані єдиним інформаційним потоком.

В ІПМС здійснюється повна інформатизація всіх етапів лікувальнодіагностичного процесу. Тобто, за єдиною технологічною схемою комплексно вирішуються такі завдання, як автоматизація процесу ведення медичної документації, машинна діагностика і прогнозування

захворювань, вироблення оптимального плану лікування, організація комп'ютерного медичного архіву.

При цьому є можливість підключати додаткові програмні засоби (утиліти) для вирішування допоміжних завдань, що характерні для відділення даного клінічного профілю (медикаментозне забезпечення, харчування, кадри тощо).

У створенні будь-якої інформаційної системи беруть участь постановник задачі, який представляє інтереси потенційного користувача, і розробник – програміст, який видає кінцеву продукцію – програмний засіб.

Процес створення інформаційної системи містить ряд послідовних етапів.

Моделювання дає уявлення про предмет, шляхи розв'язання задачі і формулювання бажаних результатів.

Наступний етап – це словесне (лінгвістичне) описання вищевикладеного з обов'язковим перерахуванням вхідних даних (вхідної інформації) і бажаних форм подання результатів розв'язання (вихідної інформації).

Далі йде формалізований (математизований) опис вищевикладеного, маючи на увазі, що чим глибший рівень формалізації, тим надійніші будуть результати роботи програміста.

Алгоритмізація рішення означає опис послідовності тих дій, які потрібно виконати над вхідною інформацією для того, щоб отримати шукані результати на виході.

Останній етап – це конкретна програмно-апаратна реалізація проекту.

Незважаючи на очевидну різницю інформаційних систем, призначених для розв'язання таких задач, сама по собі постановка кожної з них, окрім наведених вище етапів має обов'язкову внутрішню структуру, що складається з шести основних характеристик:

- рівень медичної допомоги;
- ресурси предметної області;

- засоби обчислювальної техніки;

формальні засоби чи моделі;

- алгоритмічні та програмні засоби;
- призначення;

Призначення інформаційної системи, програмного засобу, бази даних тощо містить не лише формулювання мети розроблення, а й визначення змісту та обсягу вхідної і вихідної інформації, а також способів її подальшого використання для досягнення поставленої мети.

Рівень медичної допомоги (долікарський, лікарський, догоспітальний, стаціонарний – неспеціалізований чи спеціалізований) або рівень управління (територіальний, закладу тощо), на якому буде використовуватись розробка, чітко визначає, хто, де і коли може стати користувачем задуманої системи.

Ресурси предметної області, що їх має у розпорядженні користувач, дають можливість йому отримувати всю інформацію, потрібну для введення в систему, і використовувати всю інформацію, що видається системою на виході.

Важливу роль відіграють засоби обчислювальної техніки, на котрих буде реалізована дана розробка, з урахуванням їх доступності для потенційного користувача і можливості включення в комп'ютерні мережі.

Формальні засоби чи моделі є основою побудови інформаційної системи. Для прикладу можна навести назви найчастіше використовуваних при постановці задачі видів формального моделювання: це біологічні, фізичні, кібернетичні та математичні моделі. Дуже поширеним є метод статистичного моделювання, однак використовують також логіко – імовірнісні, концептуальні, евристичні моделі. Широкі можливості для комп'ютерної реалізації дає імітаційне моделювання. Потужним інформаційним засобом є моделі, що базуються на формальному інструменті комп'ютерної графіки, на введенні даних і виведенні

результатів у вигляді зображень. До цього можна підключити також інші форми введення і виведення інформації, що пов’язані з різними видами сенсорного сприйняття (звук, тактильні відчуття, запах). Однак ці поки що «екзотичні» методи потребують не чисто програмних, а програмно-апаратних засобів, що є самостійною задачею.

Алгоритмічні і програмні засоби розв’язання задачі чи комплексу задач мають бути однією з складових частин її постановки. Такий алгоритм слід описати хоча б словами, краще – графічно, а ще краще – у вигляді загальноприйнятої стандартної схеми. В будь-якому випадку алгоритм потрібно якомога детальніше узгодити з безпосереднім розробником програми. Велике значення має також вибір програмних засобів, однак це компетенція розробника, з яким треба узгодити лише питання програмної сумісності.

Основні проблеми, що виникають при розробці МІС, можна згрупувати таким чином:

- розробка форм документів, зручних для фіксації, пошуку й обробки медичної інформації;
- вибір раціональних методів організації медичних даних, що забезпечують ефективний пошук, збереження, відновлення, вибірку інформації з пам’яті комп’ютера;

розробка комплексу програмно-технічних засобів, що забезпечують передачу даних усередині системи, обмін інформацією з зовнішніми системами, аналіз інформації;

- впровадження й експлуатація МІС. Методичні вказівки по впровадженню системи, порядок заповнення стандартизованих медичних документів, розробка інструкцій з експлуатації МІС;

Остання проблема – одна з найбільш складних і гострих, оскільки тільки на практиці можна перевірити дієвість ідей, реалізованих при розробці МІС. Тому завжди варто мати на увазі, що впровадження МІС

приводить не тільки до одержання позитивного ефекту, але і до неминучих витрат.

Ціна впровадження комп'ютерних технологій у медичну діагностику визначається наступними факторами:

- додаткові витрати на придбання, установку і обслуговування необхідного програмно-технічного забезпечення;

збільшення чисельності персоналу (за рахунок інженернотехнічних працівників);

- необхідність навчання лікарів, як мінімум, елементам комп'ютерної грамотності і, як максимум, правилам ефективної експлуатації впроваджуваної МІС;
- недосконаліть програмно-технічного забезпечення (гіпердіагностика, помилки функціонування програмного забезпечення, збої апаратури тощо);
- психологічний бар'єр з боку лікарів стосовно технічних нововведень, особливо з застосуванням комп'ютерних технологій;
- досить швидкий моральний і фізичний знос апаратно-програмних засобів і, як наслідок, необхідність витрат на їхню модернізацію;

Класифікацію МІС можна здійснювати за різними ознаками.

У залежності від ступеня автоматизації процесів збору й обробки інформації МІС поділяються на автоматизовані й автоматичні. В автоматизованих системах частина операцій по збору й обробці інформації виконується людиною. Автоматичні системи припускають повне виключення людини з процесів збору й обробки інформації.

У залежності від типу інформаційної бази МІС поділяються на системи, що оперують даними, та системи, що оперують знаннями. Системи другого типу – це експертні системи. Їхнє функціонування істотно

спирається на знання, отримані від експертів, а результати функціонування близькі результатам аналітичної діяльності експертів.

У залежності від виду розв'язуваних задач МІС можна розділити на такі групи:

- інформаційно-довідкові – системи автоматизованого пошуку, вимірювальні системи;
- інформаційно-логічні – діагностичні системи; системи прогнозу; системи моніторингу;
- керуючі або автоматизовані системи управління.

Інформаційно-довідкова система крім пошуку інформації здатна зробити визначені перетворення інформації і сформувати необхідний документ.

Інформаційно-логічна система призначена для перетворення інформації таким чином, щоб можна було одержати нову інформацію, відсутню в інформаційному масиві.

У системах управління реалізується принципово нова функція – прийняття керуючих рішень.

Найбільш широке поширення в медичних установах одержали інформаційно-пошукові системи (ІПС), які у залежності від характеру інформації поділяються на фактографічні і документальні системи.

Фактографічні ІПС містять інформаційні масиви фактичних даних. Аналогами таких систем виступають «паперові» довідники, каталоги, технічні паспорти. У комп'ютерних ІПС фактичні дані звичайно зберігаються в базах даних (БД) і являють собою таблиці, у колонках яких вказано назви різних характеристик об'єктів, а в рядках дані опису (значення характеристик) цих об'єктів.

Документальні ІПС оперують з інформацією у вигляді документів. Прикладами таких систем можуть бути бібліографічна картотека, картотека з історіями хвороб, інші картотеки. Виконуючи пошук, документальна ІПС надає або номера необхідних документів, або список заголовків, або адреси

зберігання шуканих документів. При цьому оцінку інформації, що знаходиться в знайдених документах, робить людина.

Керуючі системи реалізують збір інформації про об'єкт управління, обробку інформації, передачу даних в орган управління, формування керуючого рішення.

МІС можна класифікувати і за ієрархічним принципом, що відповідає багаторівневій структурі охорони здоров'я, як галузі. У цьому випадку їх, зазвичай, розподіляють за чотирма рівнями:

- базовий (або клінічний) рівень (лікарі різного профілю);
- рівень лікувально-профілактичного закладу (поліклініка, стаціонар, диспансер, швидка допомога тощо);
- територіальний рівень (профільні і спеціалізовані медичні служби і регіональні органи керування);
- державний рівень (державні заклади та органи управління);

У межах кожного рівня класифікація МІС здійснюється за функціональним принципом, тобто відповідно до цілей і задач, що розв'язуються системою. Розглянемо цю класифікацію більш докладно.

Медичні інформаційні системи базового рівня

Ці системи представлені системами інформаційної підтримки технологічних процесів на клінічному рівні (медико-технологічні ІС). Системи цього класу призначені для інформаційного забезпечення прийняття рішень у професійній діяльності лікарів різних спеціальностей. Основна їхня мета – комп'ютерна підтримка роботи лікаря-клініциста, гігієніста, лаборанта тощо. Ці системи дозволяють підвищити якість профілактичної і лікувально-діагностичної роботи, особливо в умовах масового обслуговування при дефіциті часу й кваліфікованих спеціалістів.

Відповідно до розв’язуваних задач медико-технологічні ІС можна розділити на наступні групи: інформаційно-довідкові системи; консультативно-діагностичні системи; приборно-комп’ютерні системи; автоматизовані робочі місця (АРМ) фахівців.

Медичні інформаційно-довідкові системи призначені для пошуку і видачі медичної інформації на запит користувача. Інформаційні масиви таких систем містять медичну довідкову інформацію різного характеру.

Медичні консультативно-діагностичні системи призначені для діагностики патологічних станів (включаючи прогноз і надання рекомендацій щодо способів лікування) при захворюваннях різного профілю та для різних категорій хворих.

Медичні приборно-комп’ютерні системи призначені для інформаційної підтримки і/або автоматизації діагностичного і лікувального процесу, здійснюваних при безпосередньому контакті з організмом хворого (наприклад, при проведенні реєстрації фізіологічних параметрів).

Системи – автоматизоване робоче місце (АРМ) лікаря призначені для автоматизації усього технологічного процесу лікаря відповідної спеціальності (лікувально-профілактичної та звітно-статистичної діяльності, ведення медичної документації, планування роботи, одержання довідкової інформації) й забезпечують інформаційну підтримку при прийнятті лікарем відповідної спеціальності діагностичних і тактичних (лікувальних, організаційних тощо) лікарських рішень.

Інформаційно довідкові системи

Необхідність накопичення великих об'ємів професійно цінної інформації і оперування ними – одна із проблем, яка виникає в професійній діяльності лікаря.

Інформаційно-довідкові системи полегшують розв'язання цієї проблеми, виступаючи як засіб надійного збереження професійних знань, забезпечує зручний і швидкий пошук необхідних відомостей.

Медичні інформаційно-довідкові системи (бази і банки даних) призначені для введення, збереження, пошуку і виведення медичної інформації відповідно до запиту користувача. Це найпростіший вид медичних інформаційних систем, що використовується на всіх рівнях системи охорони здоров'я.

Системи цього класу не виконують обробку інформації, але забезпечують швидкий доступ до потрібних даних. Інформаційні масиви таких систем містить довідкову інформацію різноманітного характеру. Це і наукова інформація з різних медичних дисциплін, і довідкова, статистична, і технологічна інформація широкого профілю.

Зазвичай, інформаційно-довідкові системи поділяють:

- за видами збереженої інформації (клінічна., наукова, нормативно-правова та ін.);
- за характером інформації (первинна, вторинна, оперативна, оглядово-аналітична, експертна, прогностична та ін.);
- за об'єктивною ознакою (матеріально-технічна база, лікарські засоби та ін.).

Крім того розрізняють документальні, документографічні, фактографічні і повнотекстові інформаційно-довідкові системи. Відповідно, види інформаційного пошуку, які можуть бути здійснені: документальний пошук, тобто пошук відомостей про той чи інший

документ, його бібліографічний опис, анотації, реферату; фактографічний пошук, тобто пошук даних та інформації, вилучених з документу.

Важливе значення має інтеграція медичних інформаційно-довідкових систем в єдину інформаційну мережу Internet, що забезпечує доступ будь-якого лікаря – користувача до інформації і обмін цією інформацією.

Консультативно-діагностичні системи

Історично консультативно-діагностичні системи (КДС) почали розвиватися одними з перших медичних діагностичних систем. В даний час консультативно-діагностичні системи представлені багато чисельними системами діагностики паталогічних станів (включаючи прогноз) при захворюваннях різноманітного профілю і для різних категорій хворих.

Вхідною інформацією для таких систем є дані про симптоми захворювань , які вводять в комп'ютер в діалоговому режимі, або в форматі спеціально розроблених інформаційних карт.

Діагностичні висновки крім власне діагнозу (або можливих діагнозів) , як правило, містить також рекомендації по вибору тактичного рішення і лікувальних заходів.

За способом розв'язання задач діагностики розрізняють імовірнісні і експертні системи. В імовірнісних системах діагностика здійснюється реалізацією одного з методів розпізнавання образів чи статистичних методів прийняття рішень. В експертних системах – реалізується логіка прийняття діагностичного рішення досвідченим лікарем.

В імовірнісних системах часто реалізується так званий байєсовський статистичний підхід, що дозволяє проводити обчислення ймовірності захворювання за його апріорною і умовною ймовірністю, які пов'язують процеси з їх характерними ознаками. Апріорна ймовірність визначається шляхом підрахунку частоти появи того чи іншого стану у вибірці.

Експертні системи належать до класу систем “штучного інтелекту», що містять базу знань з набором евристичних алгоритмів. Найбільш важливі області застосування консультативно-діагностичних систем – невідкладні та загрозливі для життя стани, що характеризуються дефіцитом часу, обмеженими можливостями обстеження і консультацій і нерідко малою клінічною симптоматикою при високому рівні загрози для життя хворого і швидких темпах розвитку процесу.

Досвід використання консультативно-діагностичних систем доводить суттєве підвищення якості діагностики, що не лише зменшує невиправдані втрати, але і дозволяє більш ефективно використовувати ресурси допомоги, регламентувати об’єм необхідних досліджень, і нарешті, підвищити професійний рівень лікарів , для яких така система слугує одночасно і навчальною

Поки консультативно-діагностичні системи не отримали широкого розповсюдження в практичній медицині і, в основному, використовуються як складова частина інших систем, наприклад, медичних приборно – комп’ютерних систем. Це пов’язано в першу чергу зі складністю задачі діагностики : в реальному житті число можливих ситуацій і, відповідно «діагностичних правил» виявилось таким великим, що система або починає вимагати велику кількість додаткової інформації про хворого, або різко знижується точність діагностики.

Медичні інформаційні системи рівня лікувально-профілактичного закладу

Вони включають в себе:

МІС консультативних центрів – призначені для забезпечення функціонування відповідних підрозділів і інформаційної підтримки лікарів при консультуванні, діагностиці та прийнятті рішень при невідкладних станах.

Банки інформації медичних установ і служб містять дані про якісний і кількісний склад працівників установи, прикріплене населення, основні статистичні відомості, характеристики районів обслуговування й інші необхідні дані.

Персоніфіковані реєстри (бази і банки даних) – це різновид інформаційно-довідкових систем, що містять інформацію про прикріплений або спостережуваний контингент населення на основі формалізованої історії хвороби або амбулаторної карти. Реєстри забезпечують дільничним, сімейним лікарям, фахівцям, ординаторам можливість швидкого одержання необхідної інформації про пацієнта, контролю за динамікою стану хворого, аналіз якості лікувально-профілактичних заходів, одержання статистичних звітних форм.

Скрінінгові системи призначені для проведення долікарського профілактичного огляду населення, а також для формування груп ризику і виявлення хворих, що потребують допомоги фахівця.

МІС лікувально-профілактичного закладу засновані на об'єднанні всіх інформаційних потоків у єдину систему й забезпечують автоматизацію різних видів діяльності ЛПУ. Відповідно до видів лікувально-профілактичної установи розрізняють такі програмні комплекси: інформаційні системи «Стаціонар», «Поліклініка» та «Швидка допомога». Вихідна інформація таких систем використовується як для вирішення задач керування відповідного лікувально-профілактичної установи , так і для вирішення задач системи охорони здоров'я вищого рівня. Розглянемо деякі з них.

Інформаційні системи консультативних центрів

Інформаційні системи консультативних центрів відносяться до медичних інформаційних систем рівня лікувально-профілактичного закладу і призначені для забезпечення функціонування відповідних

підрозділів та інформаційної підтримки лікарів при консультуванні, діагностиці і прийнятті рішень при невідкладних станах.

Інформаційні системи консультативних центрів поділяються на:

лікувальні консультативно-діагностичні системи для служб швидкої і невідкладної допомоги;

системи для дистанційного консультування і діагностики невідкладних станів в педіатрії та інших клінічних дисциплінах.

Скрінінгові системи

Як відмічалось раніше, моніторинг здоров'я населення відноситься до пріоритетних напрямків інформатизації системи охорони здоров'я.

Задачі побудови системи моніторингу здоров'я населення розв'язуються на державному, територіальному та на рівні установи. Однією із задач моніторингу є збір інформації про стан здоров'я населення. Схожі задачі розв'язуються сьогодні з допомогою комп'ютерних технологій.

Одним із типів інформаційних систем, що забезпечують розв'язання задачі збору інформації про стан здоров'я населення, є скрінінгові системи.

Скрінінгові системи представляють собою медичні інформаційні системи рівня лікувально-профілактичного закладу. Вони призначені для проведення долікарського профілактичного огляду населення, а також для формування груп ризику і виявлення хворих, які потребують допомоги спеціаліста. Скрінінг здійснюється на основі розроблених анкетних карт чи прямого діалогу пацієнта з комп'ютером.

Задачі, що розв'язується подібними інформаційними системами на рівні амбулаторного закладу, формулюються наступним чином: підвищення медичної ефективності профілактичних оглядів за всіма основними профілями патології (в 6–10 разів) і перехід від формальної звітності до реального кількісного контролю здоров'я.

Отримання спектру здоров'я не лише окремого пацієнта, а й колективів людей, і відповідно виявлення в інтегральних профілях негативних причин, безпосередньо пов'язаних з особливостями життя даного колективу; виявлення захворювання на ранніх стадіях проведення і реальна оцінка якості лікувальних і реабілітаційних заходів.

Найважливішим різновидом скрінінгових систем є автоматизовані системи профілактичних оглядів населення. Основним завданням цих систем є виявлення пацієнтів, що потребують направлення до лікарів спеціалістів

МІС характеризуються наявністю, як правило, великих обсягів данихі знань. Обробка даних і знань зводиться до трьох основних етапів. На першому етапі елементи інформації розміщуються у визначених структурах – базах даних (БД) і базах знань (БЗ). На другому етапі БД і БЗ піддаються упорядкуванню: змінюється їхня структура, порядок розміщення інформації, характер взаємозв'язків між елементами інформації. На третьому етапі здійснюють експлуатацію БД: пошук потрібної інформації, прийняття рішень, редагування баз даних і знань.

Інформаційне забезпечення МІС складають: історії хвороби, виписки з історій хвороби, епікризів, стандартизованих карт обстеження, діагностичні й інформативні оцінки показників і станів, критерії ефективності обстеження і лікування, каталог медичних понять і термінів.

У наш час закінчується період автономних медичних комп'ютерних систем, що створюються автономно окремими медичними підрозділами для вирішення своїх задач, і настає період МІС, що взаємодіють між собою. Ця взаємодія має багато аспектів:

По-перше, це використання загально прийнятих і доступних відкритих стандартів як для даних, що зберігаються й обробляються в цих системах, так і для забезпечення способів і механізмів їхньої взаємодії.

По-друге, це технічна (технологічна) стандартизація медичних комп'ютерних систем. Зрозуміло, що інструментальні засоби, що

використовуються цими системами, можуть і повинні бути різними (в залежності від певних умов їх створення та використання), але й тут необхідно передбачити максимально можливу стандартизацію (це може стосуватися стандартів до інтерфейсу, протоколів обміну даними, форматів даних, що використовуються).

Сучасні тенденції розвитку МІС свідчать про необхідність і реальну можливість такої стандартизації.

В процесі впровадження інформаційних систем особливої уваги набувають задачі надійності збереження інформації про пацієнтів, швидкого доступу до даних, можливості взаємообміну інформацією між різними лікувальними установами та проведення статистичного аналізу зведених даних.

1.2. Аналіз існуючих рішень

У процесі вивчення сучасного стану автоматизації роботи медичних установ були розглянуті наступні програмні продукти:

Продукт компанії ArchiMed, модуль «Реєстратура»

Реєстратура - модуль системи автоматизації медичних установ ArchiMed, призначений для реєстрації пацієнтів, створення і друку амбулаторних карт, складання розкладу роботи фахівців, записи пацієнтів на прийом, друку талонів електронної черги. Середня вартість продукту 220$.

Основні можливості модулю:

- побудова багаторівневих угруповань за даними;
- широкі можливості фільтрації;
- складання розкладу роботи на довільний період часу;
- швидкий пошук карти пацієнта в архіві карт;

- запис пацієнта без створення амбулаторної картки (карта створюється безпосередньо під час відвідування установи, що дозволяє зменшити кількість невикористаних електронних амбулаторних карт в архіві);
- відстеження по статусу (запис, прибув, часткова оплата, повна оплата тощо). Перемикання статусів відбувається в автоматичному режимі, що задається при налаштуванні системи;
- можливість перенесення даних про пацієнта з інших систем за допомогою спеціальних плагінів (наприклад, перенесення гостей в конфігурації «ArchiMed-санаторій» з готельної системи UCS Shelter);
- реєстрація виїздів лікарів додому. Друк карти з адресою виклику;

Medtime, автоматизоване робоче місце (АРМ) «Реєстратура»

Medtime, АРМ Реєстратура. На сьогоднішній момент ведення документально-го обліку пацієнтів займає більшу частину робочого часу лікаря, але при цьому є необхідним атрибутом будь-якого медичного закладу. Пропонована медична інформаційна система дозволяє вести повний облік наданих пацієнту медичних послуг, автоматично формує необхідну медичну документацію (первинний огляд, щоденники, протоколи додаткових досліджень, виписки, стандартні бланки для медичної установи), створює детальні звіти про роботу лікарні і персоналу за встановленими статистичними і довільним формами.

Можливості Medtime:

- можливість автоматизувати рутинні операції співробітників реєстратури поліклініки, дозволяючи їм виконувати свою роботу швидше і ефективніше;
- можливість більш ефективно і оперативно організувати процедуру прийому пацієнтів і запис їх до лікарів-спеціалістів;

- можливість оптимально складати розклад прийому лікарів, виключаючи можливість здійснення помилок;
- формувати і роздруковувати медичну документацію необхідну для роботи реєстратури.

1С-Рарус:Амбулаторія

Компанія «1С-Рарус» пропонує програмні продукти для комплексної автоматизації медичних установ. Клієнтами «1С-Рарус» в сфері охорони здоров‘я є державні та комерційні медичні організації, багатопрофільні клініки, поліклініки, диспансери, спеціалізовані медичні центри, відновно-оздоровчі та реабілітаційні центри, лікарські та масажні кабінети.

Типове рішення «1С-Рарус:Амбулаторія» призначено для комплексної автоматизації медичних закладів, які надають амбулаторну допомогу пацієнтам, які представляють собою як поодинокі, так і мережеві структури. Рішення дозволяє підвищити якість обслуговування пацієнтів, надавати медичні послуги по ОМС, ДМС і за готівковий розрахунок, вести облік медичних послуг за стандартами МОЗ, планувати роботу медичного персоналу, враховувати товарно-матеріальні цінності, здійснювати лікувальний і фінансовий контроль діяльності підприємства, проводити взаєморозрахунки з контрагентами.

2. ХАРАКТЕРИСТИКА ІНСТРУМЕНТАРІЮ ТА ВИКОРИСТАНИХ ПРОГРАМНИХ БІБЛІОТЕК

2.1. Електронна система керування чергою

Системи керування чергою допомагають уникнути скупчення людей в місцях прийому відвідувачів і організувати «цивілізований» порядок обслуговування клієнтів. В основному застосовується для розподілу, оптимізації та обліку клієнтів в черзі і виклику їх до кабінетів за допомогою звукового сигналу і візуального відображення індивідуального номера черги клієнта. Найбільш типові застосування подібних систем: каси з продажу ж / д і авіаквитків, сервіс центри з надання послуг на вокзалах, каси прийому платежів, державні установи, офіси великих фірм, банки та ін.

Конфігурація і логіка роботи подібних систем розробляється виходячи з конкретних потреб.

До складу систем можуть входити різні компоненти: принтери для друку порядкового номера клієнта, пульти управління для операторів і адміністратора, групові інформаційні табло, табло оператора, інтерфейс з комп'ютером і т.д. Додатково можливе збереження інформації про роботу системи з функціями контролю і підрахунку статистики, що в багатьох випадках дозволяє мати повну інформацію по завантаженості пункту обслуговування клієнтів, роботі окремих операторів і ін.

Над кожним кабінетом встановлено світлове табло, що відображає поточний номер черги пацієнта, що обслуговується в даний момент. Звуковий сигнал супроводжує виклик нового пацієнта.

Система, дозволяє не тільки ефективно керувати чергою, а й одночасно піклується про те, щоб кожен пацієнт отримав індивідуальне обслуговування в комфортній обстановці, коли ніхто не завадить конфіденційної роботі лікаря з пацієнтом. У будь-який момент співробітники можуть отримати практично будь-яку статистичну

інформацію про обслуговування клієнтів і роботі операціоністів. Використання системи управління чергою дозволяє прискорити час обслуговування відвідувачів більш ніж удвічі, що вигідно для організації та клієнтів.

Чекати не любить ніхто - це завжди негативно впливає на якість послуги. Проте, це явище не таке вже й рідкісне в повсякденному житті. Фактично в діяльності будь-якої організації де-небудь та виникають черги.

А адже саме емоції, які виносять клієнти з черг, - головне, що визначає їхнє ставлення до рівня отриманого сервісу в цілому і перспективи повторного обслуговування (покупки) в цьому місці.

Лімітована продуктивність завжди є слабким місцем для сфери послуг, оскільки послуги не можуть вироблятися заздалегідь і зберігатися до моменту, коли будуть затребувані. Сучасна клієнт - орієнтована організація повинна намагатися розробити стратегію, яка привносить порядок, передбачуваність і справедливість в чергу.

Поряд з вирішенням завдань управління потоками клієнтів є можливість формувати за допомогою цієї системи різні статистичні бази даних.

Застосування системи електронної черги в реєстратурі поліклініки забезпечує:

- впровадження сучасної технології обслуговування клієнтів, розподіл і оптимізацію потоків клієнтів;
- середній час очікування на кожен тип обслуговування за тривалістю робочого дня, тижня (у вигляді графіків і діаграм з різних відрізках часу);
- статистика по типам обслуговування, що дозволяє оперативно визначати завантаження оператора і розподіляти їх по типу обслуговування;

- отримання оперативної інформації в реальному часі про поточну роботу кожного лікаря, кількості працюючих кабінетів, кількості обслужених пацієнтів, кількості пацієнтів, які чекають в черзі і ін.

Існують різні модифікації системи від простої, що забезпечує тільки просування черги до системи з розширеними сервісними функціями.

Перебуваючи в «зоні очікування» пацієнти можуть підготувати необхідні документи, ознайомитися зі зразками правильного заповнення бланків і заяв або просто переглянути наявні тут довідники і почитати газети. І все це за відсутності «живої» черги з усіма її негативними моментами. Звуковий сигнал, що привертає увагу при зміні інформації на табло, не дасть можливості клієнтам пропустити свою чергу. Є навіть можливість надавати пацієнтам орієнтовний час обслуговування або середній час очікування. Завдяки цьому пацієнт може при бажанні відлучитися із залу очікування у своїх справах і повернутися до моменту обслуговування. Це створює додаткову зручність при великому потоці пацієнтів.

Ще однією перевагою системи є наявність програми статистичного обліку. Вона фіксує роботу кожного лікаря, дозволяє аналізувати зібрані дані і відповідним чином планувати і змінювати роботу поліклініки в залежності від дня тижня, сезону і потреб. Ця програма збирає дані про кількість пацієнтів, обслугованих кожним працівником і відділом у цілому в певний момент часу - годину, день, тиждень, місяць і т.д., а також час обслуговування і очікування.

У практиці розробки та реалізації систем електронного керування чергою існує великий діапазон їх функціональних і інженерних рішень.

Необхідність цивілізованого підходу до вирішення проблем, пов'язаних з обслуговуванням клієнтів, тільки підтверджує, що за такими системами майбутнє.

Електронна черга має дві головні переваги:

- Комфорт клієнта: можна отримати інформацію про потрібну послугу, отримати час виклику по електронній черзі, записатися на прийом, а також дати оцінку якості роботи співробітників та лікарів поліклініки.
- Комфорт персоналу: система управління чергою дає чітку картину розподілу навантаження на лікарів і дозволяє грамотно побудувати графік роботи без стресових навантажень.

2.2. Загальна характеристика використаних програмних бібліотек

Основною вимогою при розробці проекту є вибір інструментів з відкритим вихідним кодом. Це значно зменшить кінцеву вартість проекту, і дасть можливість поширювати систему без обмежень.

Для розробки системи використано мову програмування Python.

Python - високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності програміста і читання коду. Синтаксис Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій.

Python підтримує кілька парадигм програмування, в тому числі структурне, об'єктно-орієнтоване, функціональне, імперативне і аспектно-орієнтоване програмування. Основні архітектурні риси - динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточних обчислень і зручні високорівневі структури даних. Код в Python організовується у функції та класи, які можуть об'єднуватися в модулі (вони в свою чергу можуть бути об'єднані в пакети).

Python – мова програмування, що активно розвивається, нові версії (з додаванням / зміною властивостей) виходять приблизно раз в два з половиною роки. Внаслідок цього і деяких інших причин на Python відсутні стандарт ANSI, ISO або інші офіційні стандарти, їх роль виконує CPython.

В якості основного інструменту при створенні системи був обраний веб-фреймворк Django, написаний на мові програмування Python. Він використовується в таких сайтах, як Instagram, NASA, The Washington Post та ін., що є ознакою його надійності.

Ключові можливості Django:

- ORM, API доступу до БД з підтримкою транзакцій;

вбудований інтерфейс адміністратора, з уже наявними перекладами багатьма мовами;

- диспетчер URL на основі регулярних виразів;
- розширювана система шаблонів з тегами і наслідуванням;
- система кешування;
- інтернаціоналізація;
- «Generic views» - шаблони функцій контролерів;
- авторизація та аутентифікація, підключення зовнішніх модулів аутентифікації: LDAP, OpenID та ін.;
- система фільтрів («middleware») для побудови додаткових обробників запитів, як наприклад включені в дистрибутив фільтри для кешування, стиснення, нормалізації URL і підтримки анонімних сесій;
- бібліотека для роботи з формами (наслідування, побудова форм по існуючої моделі БД);

автоматична документація по тегам шаблонів і моделей даних, доступна через адміністративний додаток;

Django використовує шаблон проектування Model-View-Presenter (MVP).

Модель–Представлення–Пред'явник (Model-View-Presenter) — шаблон проектування, похідний від MVC, що відділяє візуальне

відображення та поведінку обробки подій у різні класи, а саме: Представлення (View) та Пред'явник (Presenter).

Передумови для застосування MVP:

- виникає необхідність відділити програмну логіку від логіки інтерфейсу користувача, щоб зробити його простішим для розуміння та підтримки;
- стоїть задача надати доступ до коду різним сторінкам, що потребують однакової поведінки;
- потрібно максимізувати обсяг коду, що підлягає автоматизованому тестуванню;

Складові частини та різновиди:

Модель. Являє собою клас для визначення даних, які будуть відображатися або над якими будуть проводитися інші дії у інтерфейсі користувача;

Представлення. Керує елементами на сторінці, та направляє події до класу Пред'явника;

- Пред'явник. Містить логіку реагування на події, оновлює Модель (бізнес-логіки і даних з програми) і, в свою чергу, маніпулює станом Представлення. Для полегшення тестування Пред'явника, він повинен мати посилання на інтерфейс Представлення замість посилання на конкретну реалізацію. Як наслідок, можливо легко замінити діюче Представлення на макет для виконання тестів;

Різновиди MVP:

Passive View

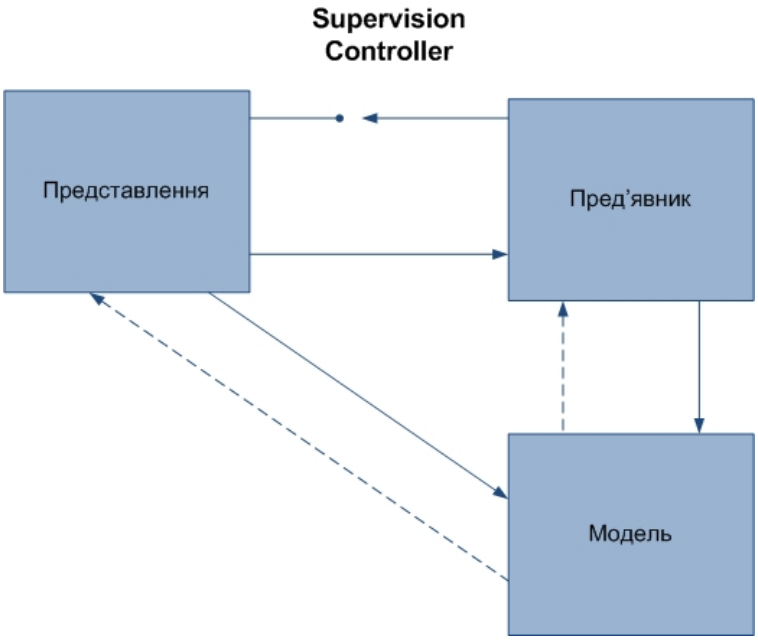

Рисунок 2.3.1 – Passive View

Представлення майже не містить логіки. Пред'явник — посередник між Представленням та Моделлю. Крім того, Представлення та Моделі повністю закритий доступ один до одного. Модель може викликати події, але Пред'явник підписується на них для оновлення Представлення.

У "Пасивному Представленні" немає прямої прив'язки даних, натомість, Представлення надає set-властивості, які Пред'явник використовує для надання значень даним. Всі стани керуються Пред'явником, а не Представленням.

Плюси: легкість тестування, прозорий поділ Представлення та Моделі.

Мінуси: витрати часу на самостійну прив'язку даних.

Supervising Controller

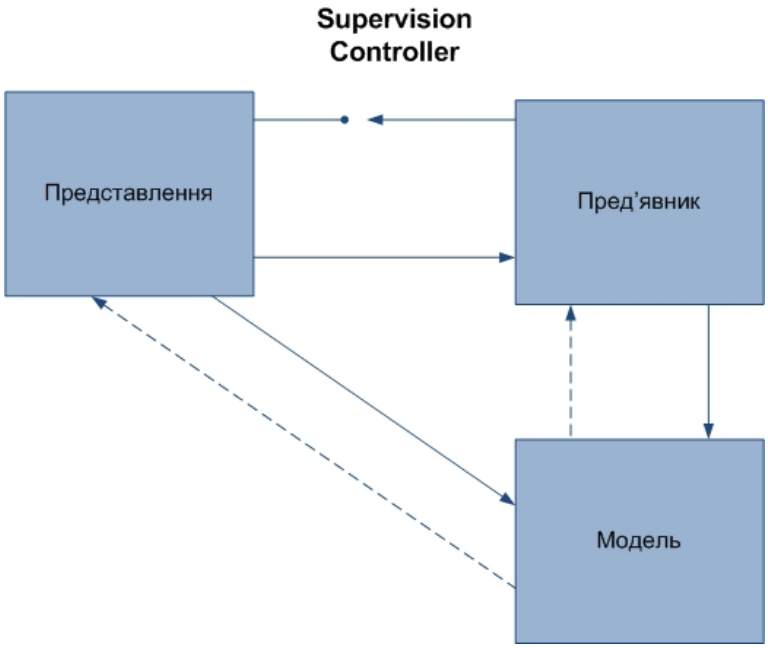

Рисунок 2.3.2 – Supervising Controller

В даному випадку, Пред'явник обробляє події користувача. Представлення зв'язується з Моделлю безпосередньо через прив'язку даних. У цьому випадку, завдання Пред'явника "пройти" від Моделі до Представлення, таким чином, щоб могла відбутися прив'язка даних. Пред'явник буде також містити логіку для таких подій як натискання клавіш, навігації і т.д.

Плюси: зменшення обсягу коду, за рахунок використання прив'язки даних.

Мінуси: більш складне тестування, зменшення інкапсуляції у Представленні через прямий зв'язок з Моделлю.

Model-View-Presenter патерн складається з трьох компонентів. У цьому патерні відображенням немає потреби підписуватися на зміни моделі, тепер контролер, перейменований в Presenter дає знати відображенням про зміни. Даний підхід дозволяє створювати абстракцію відображення. Реалізувати даний патерн можна за допомогою винесення

інтерфейсів відображення. У кожного уявлення будуть інтерфейси з певними наборами методів і властивостей, необхідних презентеру, презентер в свою чергу ініціалізується з даними інтерфейсом, підписується на події представлення і по необхідності передає дані. Даний підхід дозволяє розробляти додатки з використанням методології TDD (Test-driven development).

Шаблон MVP дозволяє виділити рівень представлення від рівня логіки, для того що б поведінка програми не залежало від його конкретного зовнішнього вигляду. В ідеалі, використовуючи MVP ми доб'ємося того, щоб одна і та ж логіка могла мати зовсім різні, а головне взаємозамінні UI представлення.

MVP - не архітектурний шаблон, тобто він відповідальний тільки за рівень представлення. Проте, його використання покращує архітектуру системи.

Розділити інтерфейс і логіку в мобільних додатках не просто, але з шаблоном Model-View-Presenter це стає трохи простіше запобігаючи перетворення нашого коду в нагромадження класів з сотнями і навіть тисячами рядків коду. У великих проектах особливо важливо мати добре структурований код, так як інакше ми зіткнемося неможливістю розширення і підтримки.

У якості бази даних для системи було обрано реляційну систему управління базами даних PostgreSQL.

PostgreSQL - це об'єктно-реляційна система управління базами даних (ОРСУБД, ORDBMS), заснована на POSTGRES, Version 4.2 - програмою, розробленою на факультеті комп'ютерних наук Каліфорнійського університету в Берклі. У POSTGRES з'явилося безліч нововведень, які були реалізовані в деяких комерційних СУБД набагато пізніше.

PostgreSQL - СУБД з відкритим вихідним кодом, основою якого був код, написаний в Берклі. Вона підтримує більшу частину стандарту SQL і пропонує безліч сучасних функцій:

- складні запити;
- зовнішні ключі;
- тригери;
- змінювані уявлення;
- транзакційна цілісність;
- багатоверсійність;

Крім того, користувачі можуть всіляко розширювати можливості PostgreSQL, наприклад створюючи свої:

- типи даних;
- функції;
- оператори;
- агрегатні функції;
- методи індексування;
- процедурні мови;

Завдяки вільній ліцензії, PostgreSQL дозволяється безкоштовно використовувати, змінювати і поширювати всім і для будь-яких цілей - особистих, комерційних чи навчальних.

Також система буде включати API для подальшої розробки, наприклад, мобільних додатків Android та iOS, або Telegram боту. Це дозволить отримати велику аудиторію, так як наявність мобільних додатків викличе більший інтерес серед студентської аудиторії, ніж при використанні одного лише сайту.

Мобільні додатки будуть пов'язані з серверною частиною через REST API за допомогою POST і GET запитів. Текстовим форматом обміну даними було обрано JSON. Він хороший тим, що передані дані мають менший

розмір, ніж при використанні XML. Це дозволить прискорити обмін даними між серверами і мобільним додатком через стільниковий зв'язок.

Веб-сервером для системи було обрано вільний веб-сервер nginx.

В nginx робочі процеси обслуговують одночасно безліч з'єднань, мультиплексуючи їх викликами операційної системи select, epoll (Linux) і kqueue (FreeBSD). Робочі процеси виконують цикл обробки подій від дескрипторів (див. подієво-орієнтоване програмування). Отримані від клієнта дані розбираються з допомогою кінцевого автомата. Розібраний запит послідовно обробляється ланцюжком модулів, що задається конфігурацією. Відповідь клієнту формується в буферах, які зберігають дані або в пам'яті, або вказують на відрізок файлу. Буфери об'єднуються в ланцюжки, що визначають послідовність, в якій дані будуть передані клієнтові. Якщо операційна система підтримує ефективні операції вводу-виводу, такі як writev і sendfile, то nginx застосовує їх при нагоді.

Конфігурація HTTP-сервера nginx дозволяє визначати віртуальні веб-сервери (директива server), які фізично знаходяться й обслуговуються одним сервером. Віртуальні сервери поділяються на локації (location). Для віртуального сервера можливо задати адреси і порти, на яких будуть прийматися з'єднання, а також імена, які можуть включати * для позначення довільній послідовності в перший і останній частині, або задаватися регулярним виразом.

Локації можуть задаватися точним URI, частиною URI, або регулярним виразом. location'и можуть бути налаштовані для обслуговування запитів зі статичного файлу, проксування на http, fastcgi чи memcached сервер.

Для ефективного керування пам'яттю nginx використовує пули — це послідовність попередньо виділених блоків динамічної пам'яті. Довжина блоку змінюється в межах від 1 до 16 кілобайт. Спочатку пулу надається лише один блок. Блок розподіляється на зайняту область й незайняту.

Виділення дрібних об'єктів виконується шляхом просування покажчика на незайняту область з урахуванням вирівнювання. Якщо незайнятої області в усіх блоках бракує для виділення нового об'єкта, то виділяється новий блок. Якщо розмір виділеного об'єкта перевищує значення константи NGX_MAX_ALLOC_FROM_POOL або довжину блоку, то він повністю виділяється з купи.

Таким чином, дрібні об'єкти виділяються дуже швидко та мають накладні витрати тільки на вирівнювання.

nginx містить модуль географічної класифікації клієнтів за IP-адресою. У його основу входить база даних відповідності IP-адрес географічному регіону, представлена у вигляді Radix tree в оперативній пам'яті. nginx попередньо розподіляє перші кілька рівнів дерева, таким чином, щоб вони займали рівно 1 сторінку пам'яті. Це гарантує, що при пошуку IP-адреси для перших декількох вузлів при трансляції адреси завжди знайдеться запис у буфері асоціативної трансляції (TLB).

У якості платформу для хостингу системи було обрано Amazon Web Services.

Amazon Web Services (AWS) є дочірньою компанією Amazon.com, що надає платформу хмарних обчислень в оренду приватним особам, компаніям та урядам на основі платної підписки. Існує і безкоштовна підписка, яка доступна протягом перших 12 місяців. Технологія дозволяє абонентам мати у своєму розпорядженні повноцінний віртуальний кластер комп'ютерів, який завжди доступний через Інтернет. Віртуальні комп'ютери AWS мають більшість атрибутів реального комп'ютера, включаючи апаратні пристрої (процесор, відеокарту, локальну та оперативну пам'ять, жорсткий диск або SSD-накопичувач); операційну систему на вибір; мережу; і попередньо встановлені прикладні програми, такі як веб-сервер, база даних, CRM і т. д. Кожна система AWS також віртуалізує консольний ввід/вивід (клавіатура, дисплей і миша), що дозволяє користувачам AWS

підключитися до своєї системи AWS за допомогою браузера. Браузер виступає як вікно у віртуальний комп'ютер, дозволяючи користувачу входити в систему, налаштовувати та використовувати свої віртуальні системи так само, як справжній, фізичний комп'ютер. Це дозволяє їм налаштувати систему так, щоб надавати інтернет-орієнтовані сервіси та послуги своїм клієнтам.

Технологія AWS базується на серверних кластерах (фермах), розташованих по всьому світі. Плата за користування базується на комбінації використання апаратних засобів/ОС/програмного забезпечення/мережевих функцій, вибраних користувачем, а також вимог до доступності, надлишковості (redundancy), безпеки та додаткових параметрів. Виходячи з того, що користувач потребує і оплачує, він може зарезервувати один віртуальний комп'ютер (VM) , кластер віртуальних комп'ютерів (VM Cluster), фізичний (реальний) комп'ютер (Server), призначений для його виняткового використання, або навіть кластер фізичних комп'ютерів (Server Cluster). Компанія Amazon зобов'язується керувати та оновлювати програмне та апаратне забезпечення для дотримання необхідних стандартів безпеки. AWS працює в багатьох географічних регіонах, у тому числі в Канаді, Німеччині, Ірландії, Сінгапурі, Токіо, Сіднеї, Пекіні, Лондоні і т. д.

Amazon рекламує AWS як спосіб отримання обчислювальної потужності що масштабується швидше та дешевше, ніж побудова власного фізичного серверного кластеру. Усі послуги оплачуються залежно від використання, однак кожна служба вимірює використання своїм методом.

3. ОПИС РОЗРОБЛЕНИХ ПРОГРАМНИХ РІШЕНЬ ТА АЛГОРИТМІВ

3.1. Структура бази даних

В процесі розробки були створені таблиці, що містять необхідні поля, для реалізації наступних функцій:

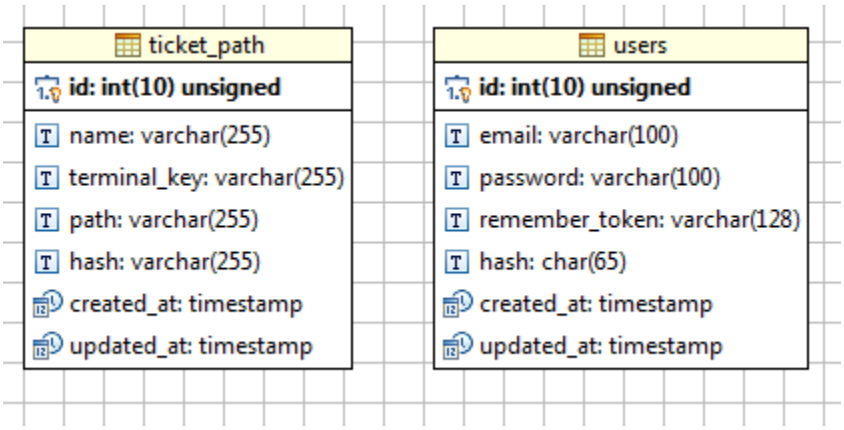

Рисунок 3.1.1 – Таблиці ticket_path та users

Таблиця ticket_path слугує для маршрутизації талонів в системі, містить поля name для імені талона, terminal_key для зберігання індивідуального ключа терміналу, що запросив талон, поле Path для зберігання ідентифікатора черги, в яку направлений талон, поле hash для зберігання індивідуальної контрольної суми талона, поле created_at для зберігання дати створення талона і поле updated_at для зберігання часу останньої зміни талона.

Таблиця users слугує для зберігання даних про користувачів системи, таких як адреса електронної пошти в полі email, особистого пароля користувача в полі password, поле remember_token слугує для зберігання даних про улюблений варіант відображення інтерфейсу, поле hash для зберігання індивідуальної контрольної суми користувача, поле created_at

для зберігання дати створення користувача і поле updated_at для зберігання часу останньої зміни користувача.

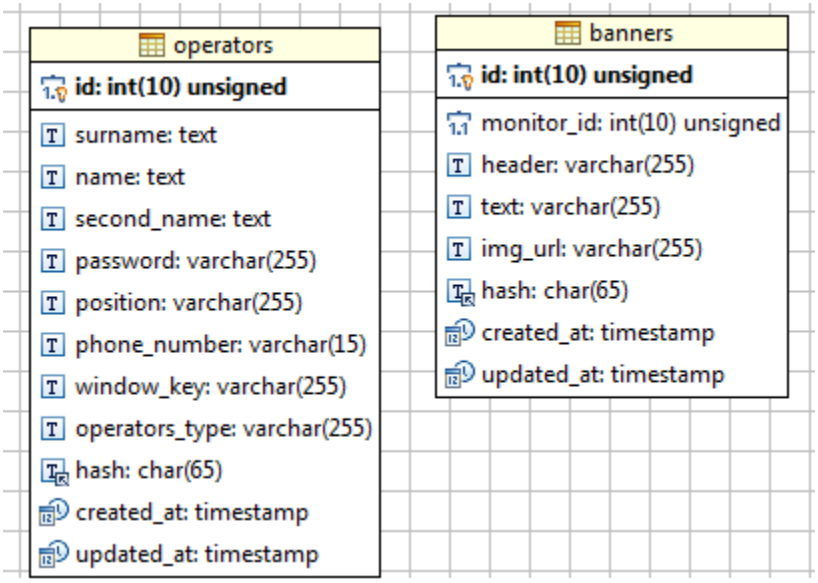

Рисунок 3.1.2 – Таблиці operators та banners

Таблиця operators слугує для зберігання даних операторів системи управління потокаміпаціентов. В поле surname зберігається прізвище, в поле name - ім'я оператора, в поле second_name – по батькові оператора. Поле password служить для зберігання особистого пароля оператора, для авторізациі в системі, поле position містить дані про посади оператора, поле phone_number для зберігання номера телефону оператора, поле window_key містить дані по кабінетах / вікнах реєстратури, в яких оператор може почати роботу, поле operators_type містить дані про тип оператора

Таблиця banners містить дані про відображені на моніторах системи банерах, в поле monitor_id зберігається індивідуальний номер набору даних на моніторі, в поле header міститься рядок, що відображається у верхній частині банера, в поле text зберігається текст виведений на банері, в полі img_url міститься шлях до файлу із зображенням, що відноситься до цього банеру, поле hash служить для зберігання індивідуальної контрольної суми

банера, поле created_at для зберігання дати створення банера і поле updated_at для зберігання часу останньої зміни банера.

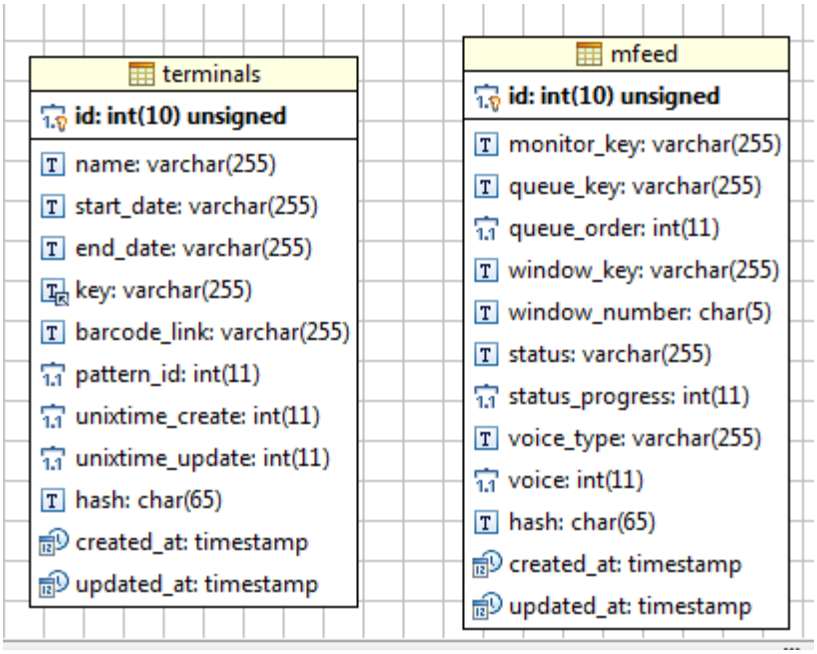

Рисунок 3.1.3 – Таблиці terminals та mfeed

Таблиця terminals слугує для зберігання даних про підключені до системи управління потоками пацієнтів термінали для видачі талонів, поле name містить рядок з ім'ям терміналу, поле start_date містить час початку обслуговування установи, поле end_date містить час закінчення обслуговування установи, в поле key міститься індивідуальний ключ терміналу, в поле barcode_link містяться дані про режим роботи сканера штрих-кодів, в поле pattern_id міститься ідентифікатор обраного шаблон талона для терміналу, в поле unixtime_create міститься час створення терміналу в форматі Unix time, в поле unixtime_update міститься час останнього вимірювання даних терміналу в форматі Unix time. У полях created_at і updated_at зберігаються ті ж дані, що і в unixtime_create і unixtime_update, але в стандартному варіанті представлення часу.

Таблиця mfeed слугує для зберігання даних про вміст головного табло системи управління потоками пацієнтів. Поле monitor_key містить

індивідуальний ключ монітора, поле queue_key містить індивідуальний ключ черги, поле queue_order містить пріоритет черги, поле window_key містить індивідуальний ключ вікна / кабінету для якого був виданий талон, поле status містить статус талона - в очікуванні, викликаний , в роботі або завершений. В поле status_progress зберігається числове значення, відповідне варіанту відображення талона на головному табло, поля voice_type і voice_int містять вимовний роботом при виклику талона текст і номер. Фраза генерується з заздалегідь записаних звукових доріжок.

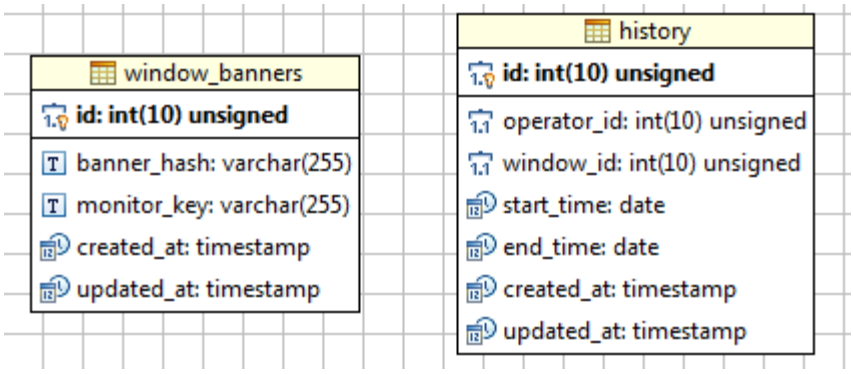

Рисунок 3.1.4 – Таблиці window_banners та history

У таблиці window_banners містяться дані для зв'язку банерів з моніторами на яких вони відображаються, в поле banner_hash зберігається индивідуальна hash-сума банера, в поле monitor_key зберігається ключ монітора на якому буде відображений банер, поле created_at слугує для зберігання дати створення банера і поле updated_at для зберігання часу останньої зміни банера.

У таблиці history міститься історія роботи операторів, в полі operator_id міститься індивідуальний номер оператора черги, в полі window_id міститься індивідуальний номер кабінету або вікна в реєстратурі черги, в полі start_time міститься час початку роботи оператора, в полі end_time міститься час завершення роботи оператора, поле created_at слугує для зберігання дати створення і поле updated_at для зберігання часу останнього вимірювання історії.

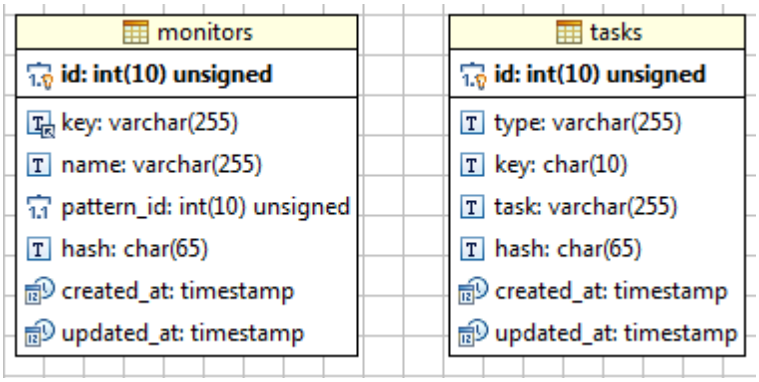

Рисунок 3.1.5 – Таблиці monitors і tasks

У таблиці monitors містяться дані про підключених до системи управління потоками пацієнтів центральних табло. В полі key зберігається індивідуальний ключ монітора, в полі name зберігається назва табло, в полі pattern_id зберігається ідентифікатор обраного шаблону талона, в полі hash зберігається індивідуальна hash-сума для табло, поле created_at слугує для зберігання дати створення і поле updated_at для зберігання часу останньої зміни в таблиці.

У таблиці tasks містяться дані про заплановані завдання по обслуговуванню системи, формування звітів та інших, які виконуються за запитом користувача або автоматично. В полі type міститься опис завдання, в полі key міститься позначка про успішне виконання або номер помилки, в полі task міститься посилання на скрипт виконуючий завдання, в полі hash міститься індивідуальна hash-сума для завдання, поле created_at слугує для зберігання дати створення завдання і поле updated_at для зберігання часу останньої зміни в таблиці.

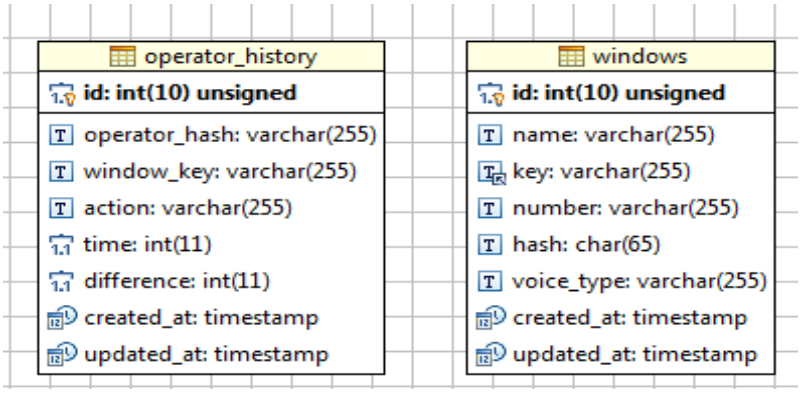

Рисунок 3.1.6 – Таблиці operator_history та windows

Таблиця operator_history слугує для реєстрації дій оператора, таких як початок і завершення роботи або установка паузи. Поле operator_hash містить унікальну hash-суму для кожного оператора, поле window_key містить індивідуальний ключ вікна / кабінету, поле action містить обрану оператором дію, поле time містить обраний оператором часовий проміжок, поле difference містить код причини. Поле created_at слугує для зберігання дати створення і поле updated_at для зберігання часу останньої зміни в таблиці.

Таблиця windows слугує для зберігання даних про кабінети і вікна реєстратури установи. Поле name містить назву вікна або кабінету, поле key містить унікальний ключ кабінету або вікна, поле number містить номер вікна або кабінету, поле hash містить унікальну hash-суму для кожного вікна або кабінету, поле voice_type слугує для зберігання вимовної фрази. Поле created_at слугує для зберігання дати створення і поле updated_at для зберігання часу останньої зміни в таблиці.

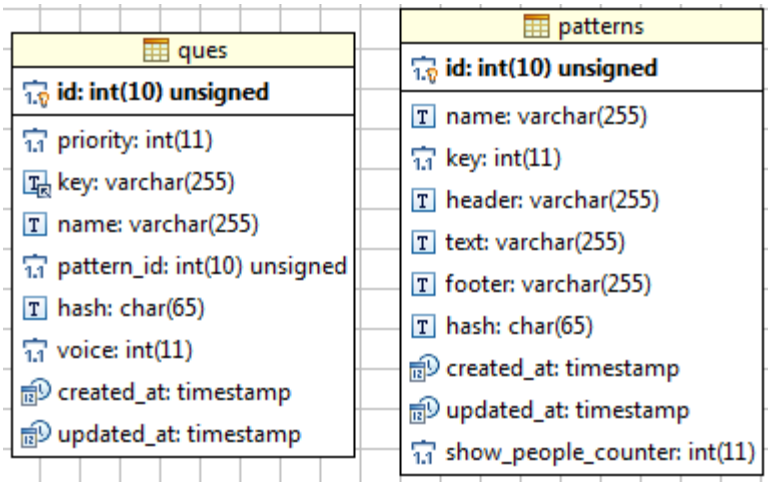

Рисунок 3.1.7 – Таблиці ques та patterns

У таблиці ques містяться дані про створені в системі черги. В полі priority міститься позначка про пріоритет черги, в полі key міститься унікальний ключ черзі, в полі name зберігається назва черги, в полі pattern_id міститься ідентифікатор обраного шаблону талона для термінала, в полі hash міститься унікальна hash-сума для кожної черги, в полі voice міститься позначка про варіант звукового супроводу для черги. Поле created_at слугує для зберігання дати створення і поле updated_at для зберігання часу останньої зміни в таблиці.

Таблиця patterns містить дані про шаблони талонів, в полі name зберігається назва шаблону, в полі key міститься унікальний ключ шаблону, в полі header міститься рядок-заголовок талона, в полі text містяться текстові дані, що виводяться на талоні, в полі footer містяться текстові дані, що виводяться під номером у черзі на талоні, в полі hash міститься унікальна hash-сума для кожного шаблону талона. Поле created_at слугує для зберігання дати створення і поле updated_at для зберігання часу останньої зміни в таблиці. В полі show_people_counter міститься позначка про те чи потрібно виводити на талоні кількість осіб в черзі перед клієнтом.

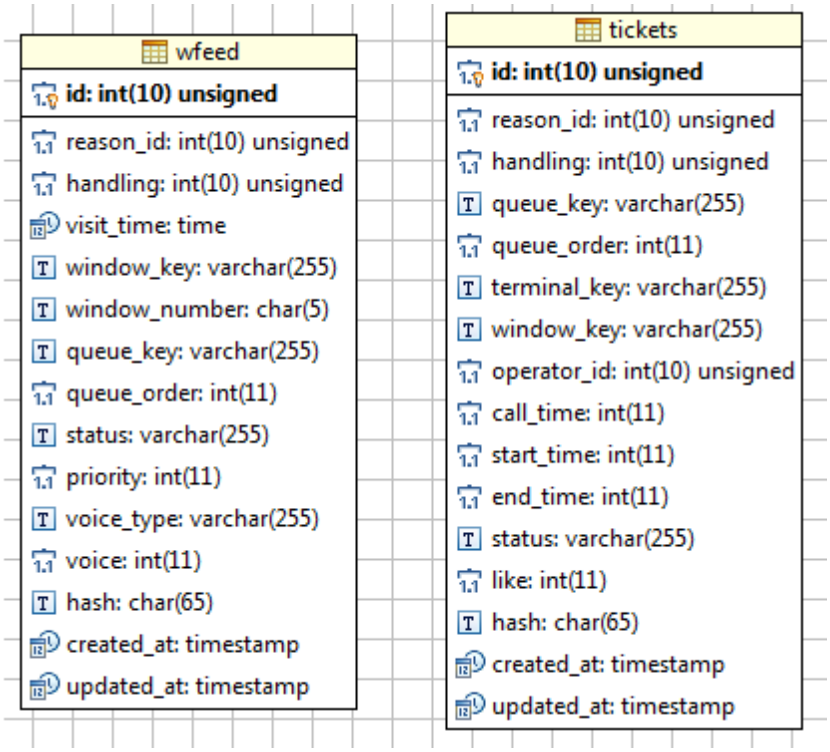

Рисунок 3.1.8 – Таблиці wfeed та tickets

У таблиці wfeed містяться дані про талони, які надійшли в вікно або кабінет і діях з ними, в полі reason_id міститься код причини дії, в полі handling міститься позначка про встановлення паузи, в полі visit_time міститься час прийому зіставлений до талону з розкладу, в полі window_key зберігається унікальний ключ вікна або кабінету, в полі window_number міститься номер кабінету або вікна, в полі queue_key міститься унікальний ключ черги, в полі status міститься статус талона в черзі, в полі priority міститься ключ обраного для талона пріоритету, поле voice_type слугує для зберігання вимовної фрази, в полі voice міститься позначка про вибраний варіант озвучення талона, в полі hash міститься унікальна hash-сума для талона, поле created_at слугує для зберігання дати створення і поле updated_at для зберігання часу останньої зміни в таблиці.

У таблиці tickets містяться дані про зареєстровані у системі талони, в полі reason_id міститься код причини дії, в полі handling міститься позначка про встановлення паузи, в полі queue_key міститься унікальний ключ черги,

в полі queue_order міститься позначка про пріоритет черги, в полі terminal_key міститься унікальний ключ терміналу, який видав талон. В полі window_key міститься назва вікна або кабінету, в полі operator_id міститься унікальний ключ оператора, в полі call_time міститься час виклику талона оператором, в полі start_time міститься час початку обслуговування, в полі end_time міститься час завершення обслуговування, в полі status міститься відмітка про статус талона, в полі like міститься позначка про обрану пацієнтом оцінку обслуговування, в полі hash міститься унікальна hash-сума для талона, поле created_at слугує для зберігання дати створення і поле updated_at для зберігання часу останньої зміни в таблиці.

3.2. Опис необхідного функціоналу

Загальносистемний необхідний функціонал:

- авторизація основних типів користувачів: пацієнт, лікар, адміністратор;
- зміна паролю в особистому кабінеті користувачів;
- перегляд інформації про клініку
- перегляд інформації про лікарів;
- запит на реєстрацію;
- перегляд відкритого пакету документів клініки;

Система буде складатися з трьох частин. Для зручності користування вона буде визначати, ким є користувач і змінювати свій інтерфейс під потрібний функціонал.

Першою частиною буде інтерфейс пацієнта. Він повинен дозволяти користувачеві записуватися до лікаря в слушний йому час, переглядати список своїх минулих і поточних записів, скасовувати запис, надавати інформацію про поліклініку, список лікарів які працюють в його поліклініці і дозволяти дивитися персональну інформацію.

Необхідний функціонал:

- перегляд особистої сторінки;
- запис на прийом до лікаря, здачу аналізів;
- перегляд інформації про клініку;

перегляд інформації про лікарів;

- надання дозволу на доступ до медичної картки;

Другою частиною буде інтерфейс лікаря. Він повинен дозволяти лікарям записувати до себе пацієнтів, переглядати список записаних до нього пацієнтів, додавати дні прийомів і видаляти години прийому.

Необхідний функціонал:

- перегляд своєї особистої сторінки;
- перегляд своєї черги пацієнтів;

направлення пацієнтів на аналізи;

- зміна свого розкладу;
- введення даних до медичних карток пацієнтів;

Третьою частиною буде інтерфейс адміністратора. Він служить для того, щоб поліклініка могла керувати інформацією про своїх лікарів, кабінетів і спеціальностей, які будуть в базі поліклініки. Також через нього можна буде додавати або переводити в інші поліклініки пацієнтів, а також записувати їх до лікарів.

Необхідний функціонал:

- контроль електронної черги;
- контроль розкладу роботи лікарів;
- створення облікових записів для користувачів;
- деактивація облікових записів користувачів;
- редагування облікових записів пацієнтів;
- зміна значень системних параметрів (тексти стандартних повідомлень, керування розкладом і т.п.);

Система може використовуватись у будь-якій клініці. Система надає доступ до інформації кожному своєму користувачу лише після його авторизації та ідентифікації.

Система призначена для постійної, щоденної роботи. Користувачі працюють в діалоговому режимі в реальному масштабі часу (on-line) з базою даних системи, функціонуючої на сервері бази даних.

Окремі частини системи працюють в сеансах, кількість і тривалість яких, визначається потребами конкретних користувачів. Сервер бази даних системи повинен працювати в безперервному цілодобовому режимі, крім періодів проведення регламентних робіт з копіювання даних системи, проведення регламентних ремонтних або відновлювальних робіт.

Система повинна безвідмовно функціонувати, не зважаючи на наявність ймовірних помилок (дефектів), які можуть проявлятися під час експлуатації. Виправлення виявлених помилок (дефектів) повинно здійснюватись на етапах попередніх випробувань, дослідної експлуатації системи та в процесі введення в дію і промислової експлуатації.

Система повинна коректно взаємодіяти з іншими програмно-технічними комплексами, програмами та програмними засобами, не порушувати їх функціонування.

Відмова програмного забезпечення системи не повинна викликати руйнувань даних в інформаційних сховищах.

Протягом усього терміну зберігання для облікових записів повинна бути забезпечена їх цілісність і незмінність. Для цього використовуватиметься метод резервного копіювання. Система повинна бути добре захищена від різного роду зловмисних атак із метою заволодіння інформації. Також Система повинна витримувати великі навантаження, обслуговуючи значну кількість користувачів. Крім цього повинна забезпечуватись конфіденційність персональної інформації. Вимоги, що

пред'являються до програмного продукту: програма повинна виконувати свої функції, бути зрозумілою для користувача та мати зручний інтерфейсу.

Представлення даних у базі знань повинно володіти основними властивостями інформації: повнота, несуперечливість, достовірність, адекватність, захищеність, ергономічність і т.д.

Програма повинна виконувати всі базові функції по роботі з інформацією: пошук інформації, передача даних, зберігання даних, видалення даних та їх перегляд.

При реалізації графічного інтерфейсу користувача системи повинні бути враховані вимоги та рекомендації, щодо розміщення вікон, підказок, призначення керівних та функціональних клавіш та т. ін.

Інтерфейс системи повинен бути зрозумілим і зручним не повинен бути переобтяжений графічними елементами і повинен забезпечувати швидке відображення екранних форм. Навігаційні елементи повинні бути виконані в зручній для користувача формі.

Інтерфейс користувача системи повинен дозволяти оперувати професійно-орієнтованими поняттями предметної області діловодства державною мовою.

Інтерфейс користувача системи повинен забезпечувати можливість зворотності дій користувача та необхідність підтвердження потенційно руйнівних дій користувача з модифікації та відновлення даних.

В інтерфейсі користувача повинні бути передбачені засоби одержання довідкових даних про можливості системи, тобто забезпечена можливість отримання на екрані підказок щодо виконання операцій, функцій та інше.

Введення-виведення даних, прийом управляючих команд і відображення результатів їх виконання повинні виконуватися в інтерактивному/діалоговому режимі.

Екранні форми повинні проектуватися з урахуванням вимог уніфікації: всі екранні форми інтерфейсу користувача повинні бути виконані в єдиному графічному дизайні, з однаковим розташуванням

основних елементів управління і навігації. Для позначення схожих операцій повинні використовуватися схожі графічні значки, кнопки і інші елементи. Терміни, що використовуються для позначення типових операцій, а також послідовності дій користувача при їх виконанні повинні бути уніфіковані. Зовнішня поведінка схожих елементів інтерфейсу (реакція на наведення покажчика, перемикання фокусу, натискання кнопки) повинні реалізовуватися однаково для однотипних елементів.

Інтерфейс повинен відповідати сучасним ергономічним вимогам і забезпечувати зручний доступ до основних функцій і операцій системи.

З метою запобігання невірному введенню даних користувачем має бути реалізовано контроль та перевірка вхідних даних на етапі їх внесення. У разі спроби внесення невірних даних користувачем, видається повідомлення про помилку.

В системі забезпечено доступ до даних шляхом авторизації користувачів шляхом введення логіну та паролю. Далі відбувається перевірка даних авторизації, і при їх успішній перевірці користувач отримує доступ до системи. Розмежування прав на роботу з системою налаштовуються адміністратором та відбувається шляхом надання прав користувачу або шляхом внесення користувача до існуючої групи користувачів, яка має відповідні права доступу до документів.

3.3. Проектування системи

Розглянемо діаграму прецедентів для системи. На рисунку 3.3.1 зображено основних акторів системи. Всі користувачі системи поділяються на дві основні групи: зареєстровані користувачі (особи, зареєстровані у системі, що звертаються до неї за одержанням потрібної їм інформації, щоб користуватися нею) та незареєстровані користувачі (особи, що звертаються до інформаційної системи, але не зареєстровані в ній).

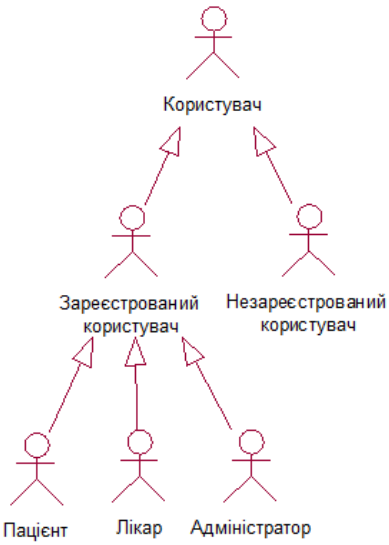

Рисунок 3.3.1 - Основні актори Системи

На рисунку 3.3.2 зображені юзкейси для загального користувача. Його можливостями можуть користуватися і зареєстровані, і незареєстровані користувачі.

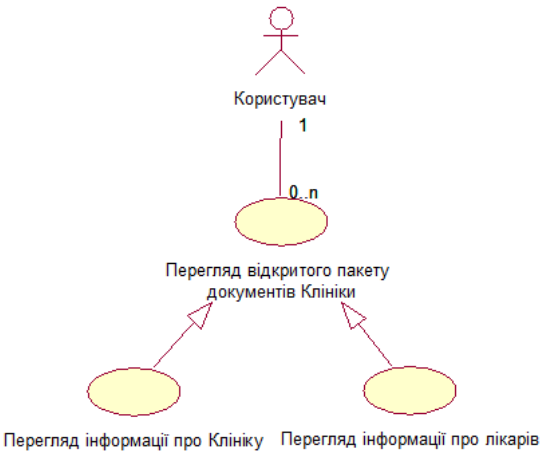

Рисунок 3.3.2 - Можливості користувача

На рисунку 3.3.3 зображені юзкейси для зареєстрованих користувачів: пацієнтів (особи, які отримують медичну допомогу, лікування чи медичне спостереження в клініці), лікарів (фахівці із повною вищою медичною освітою, які в установленому законом порядку займається підтримкою або відновленням людського здоров'я, через запобігання, розпізнавання та лікування захворювань і травм) та адміністраторів (особи, які контролюють

роботу лікарів та підтверджують реєстрацію нових користувачів, які будуть користуватись послугами, які надає клініка).

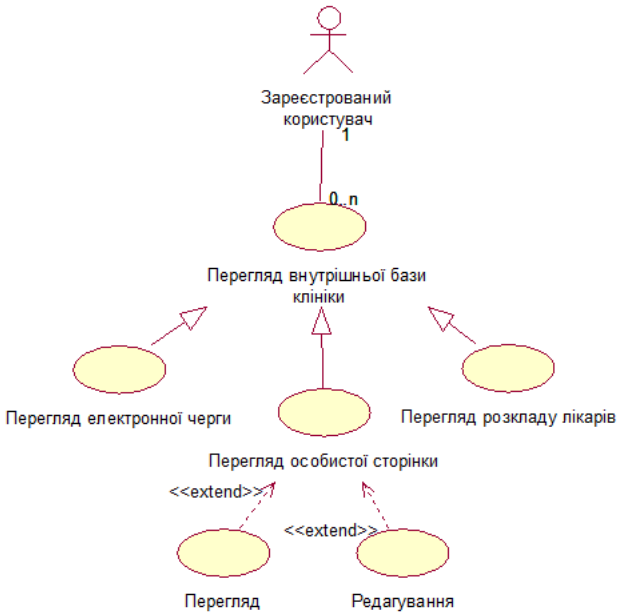

Рисунок 3.3.3 - Можливості зареєстрованих користувачів

На рисунку 3.3.4 зображені юзкейси для незареєстрованих користувачів.

Рисунок 3.3.4 - Можливості незареєстрованих користувачів

На рисунках 3.3.5-3.3.7 зображені юзкейси для пацієнта, лікаря та адміністратора.

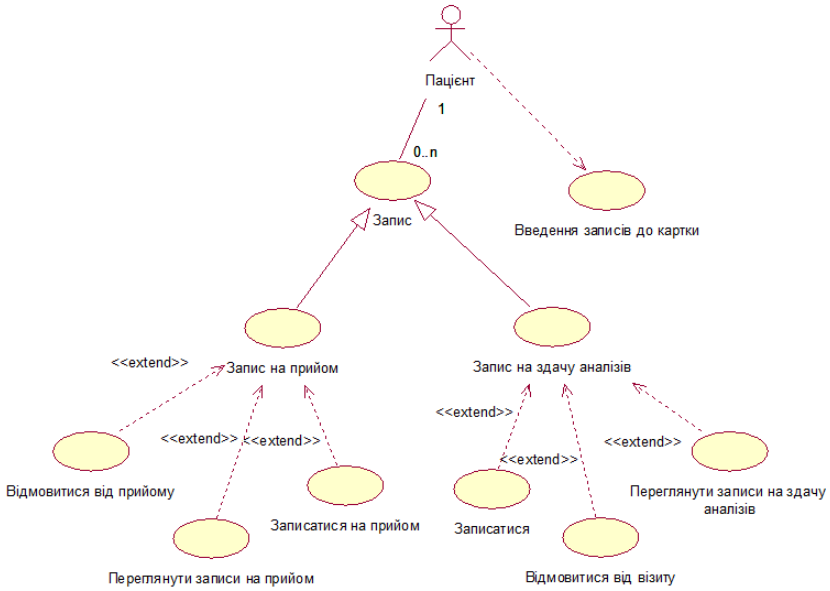

Рисунок 3.3.5 - Можливості пацієнта

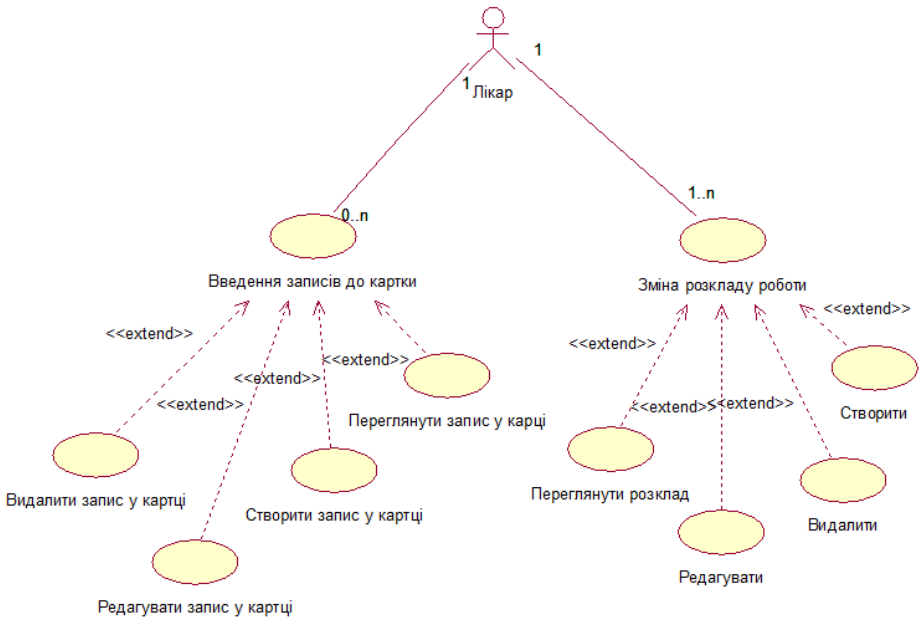

Рисунок 3.3.6 - Можливості лікаря

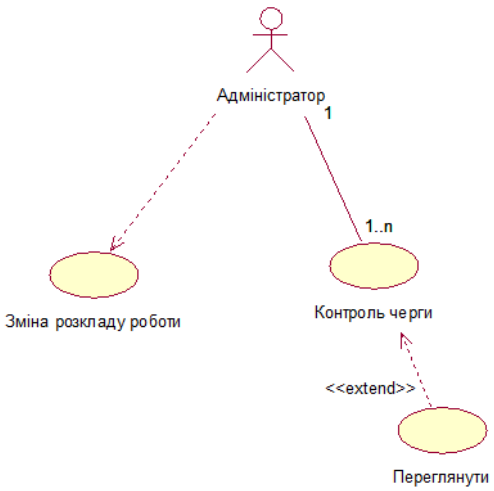

Рисунок 3.3.7 - Можливості адміністратора

3.4. Основні сценарії виконання

Перегляд особистої сторінки:

1. Зареєстрований користувач входить в систему;
2. Зареєстрований користувач вводить логін та пароль;
3. Система перевіряє дані на коректність та створює сеанс роботи;
4. Зареєстрований користувач отримує доступ до особистої сторінки;
5. Зареєстрований користувач переглядає інформацію на особистій сторінці;

Виключні ситуації:

1. Зареєстрований користувач невірно ввів логін та пароль або залишив поля порожніми;
2. Порушено зв’язок між сервером, базою даних та сайтом;

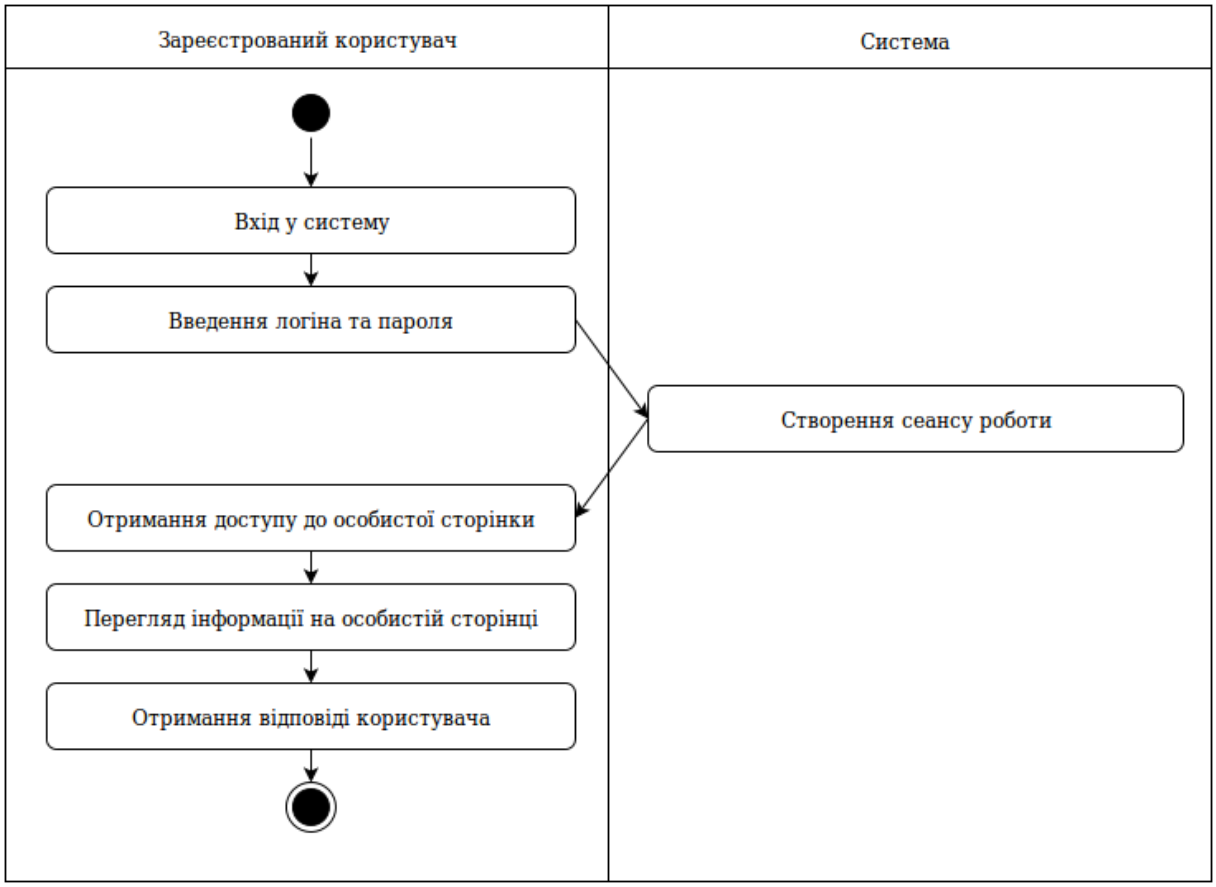

Рисунок 3.4.1 – Перегляд особистої сторінки

Зміна даних в медкартці пацієнта лікарями:

1. Лікар входить в систему;
2. Лікар вводить логін та пароль;
3. Пацієнт вводить логін та пароль;
4. Система перевіряє дані на коректність та створює сеанс роботи.
5. Лікар працює з медичною карткою пацієнта;
6. Система надсилає пацієнту повідомлення на електронну пошту з інформацією про діагноз, необхідні ліки та направлення на аналізи;

Виключні ситуації:

1. Лікар або пацієнт невірно ввів логін та пароль або залишив поля порожніми;
2. Порушено зв’язок між сервером, базою даних та сайтом;

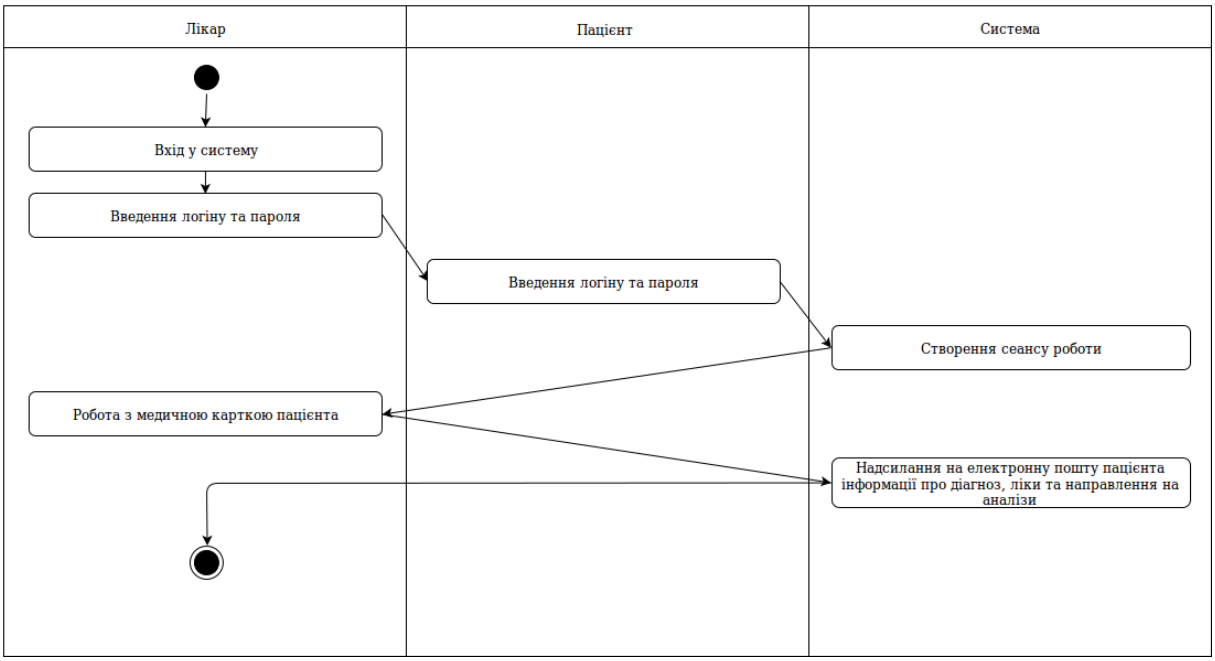

Рисунок 3.4.2 – Зміна даних в медкартці пацієнта лікарями

Перегляд інформації:

1. Користувач входить в систему;
2. Користувач натискає кнопку «Переглянути інформацію» на сайті;
3. Система повертає інформацію, яку обрав користувач;
4. Користувач переглядає інформацію;

Виключні ситуації:

1. Порушено зв’язок між сервером, базою даних та сайтом;

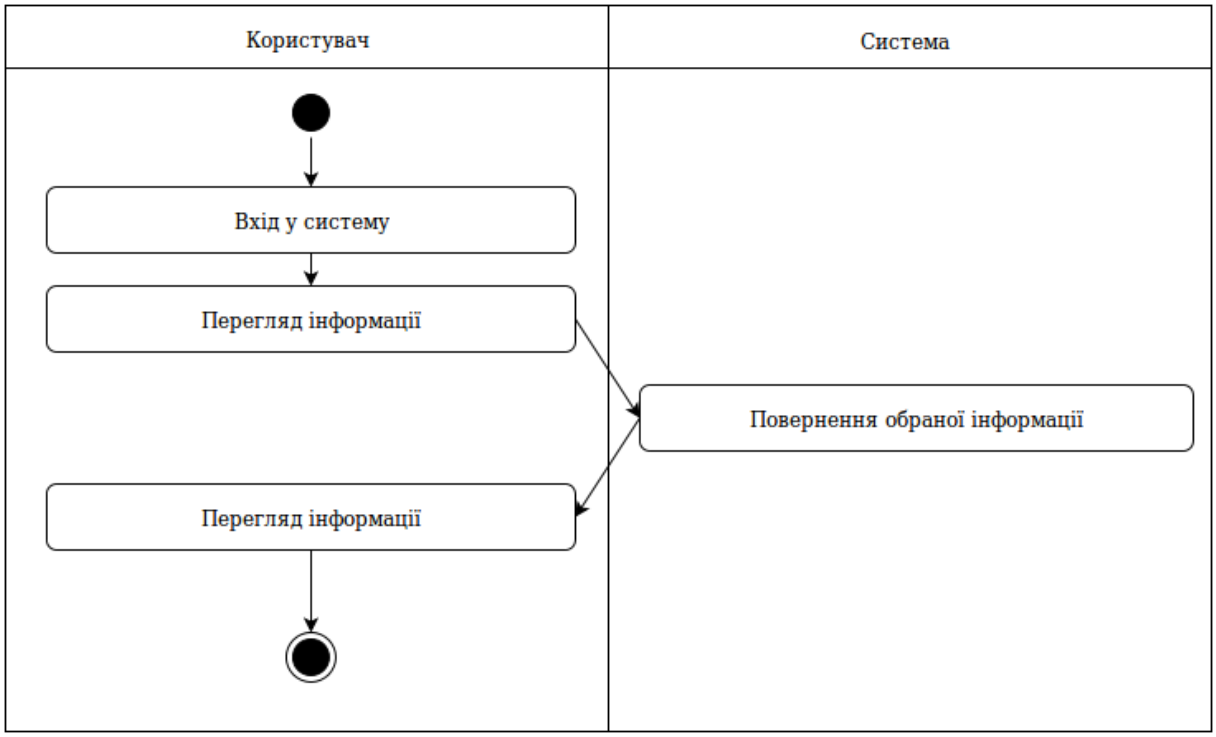

Рисунок 3.4.3 – Перегляд інформації

Перегляд електронної черги:

1. Зареєстрований користувач входить в систему;
2. Зареєстрований користувач вводить логін та пароль;
3. Система перевіряє дані на коректність та створює сеанс роботи;
4. Зареєстрований користувач натискає кнопку «Електронна черга»;
5. Зареєстрований користувач обирає лікаря і дату, чергу якого він хоче переглянути;

Виключні ситуації:

1. Зареєстрований користувач невірно ввів логін та пароль або залишив поля порожніми;
2. Порушено зв’язок між сервером, базою даних та сайтом;

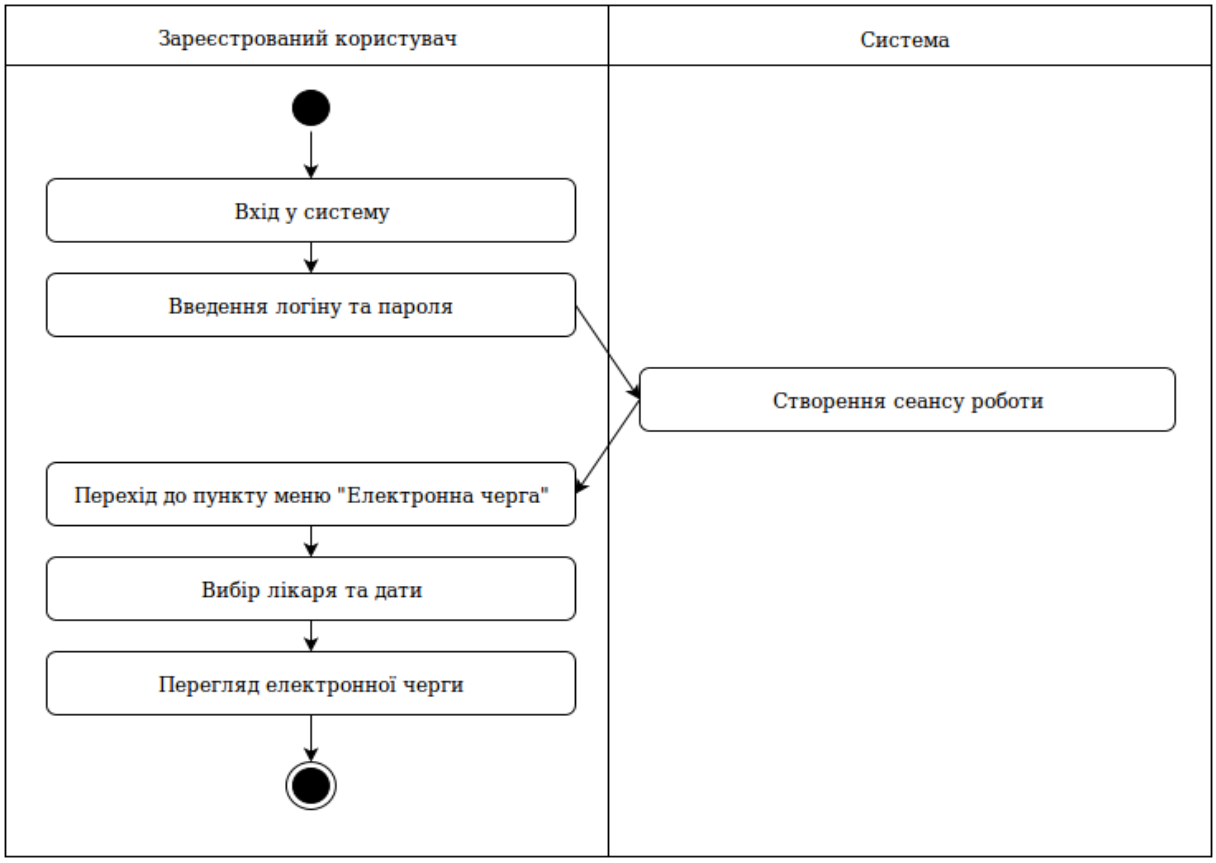

Рисунок 3.4.4 – Перегляд електронної черги

Зміна розкладу:

1. Лікар входить в систему;
2. Лікар вводить логін та пароль;
3. Система перевіряє дані на коректність та створює сеанс роботи;
4. Лікар натискає кнопку «Мій розклад»;
5. Лікар вносить зміни (час, дата, кабінет) до свого розкладу;

Лікар натискає кнопку «Підтвердити зміни»;

6. Система створює розклад лікаря у стані «не активний»;

Виключні ситуації:

1. Лікар невірно ввів логін та пароль або залишив поля порожніми;
2. Лікар не підтвердив факт зміни розкладу;
3. Порушено зв’язок між сервером із базою даних та сайтом;

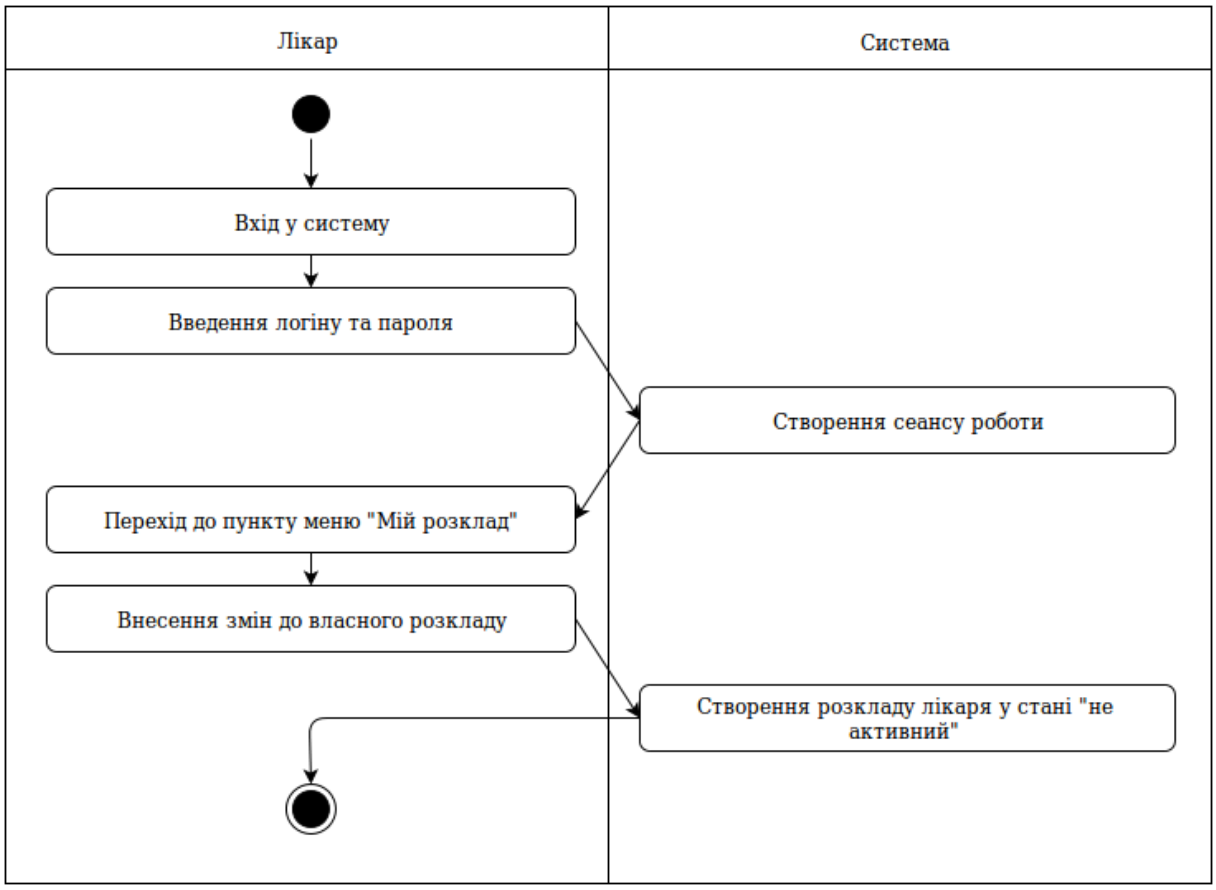

Рисунок 3.4.5 – Зміна розкладу

3.5. Реалізація серверної частини на базі Django

Спочатку створюється модель класів у файлі models.py. Далі створюються контролери сторінок в файлі views.py, через які клієнти матимуть можливість вносити зміни в базі даних. Щоб їх викликати, потрібно в файлі urls.py додати шаблон HTTP-запиту і вказати назву функції необхідного контролеру.

```
class Polyclinic(models.Model):
    name            = models.CharField(max_length = 100)
    site            = models.CharField(max_length = 100, blank = True)
    address_street  = models.CharField(max_length = 100)
    address_number  = models.CharField(max_length = 100)
    registry_phone  = models.CharField(max_length = 100, blank = True)
    reception_phone = models.CharField(max_length = 100, blank = True)
    email           = models.CharField(max_length = 100, blank = True)

    def __unicode__(self):
        return self.name
```

Рисунок 3.5.1 - Приклад моделі класу

```
urlpatterns = [
    url(r'^$', views.index_page, name = 'index_page'),

    # login & logout API
    url(r'^login/$', views.login_page, name = 'login_page'),
    url(r'^login_user/$', views.login_user),
    url(r'^logout_user/$', views.logout_user),

    # mobile client API
    url(r'^api/mobile/login_user/$', views.login_mobile_user),
    url(r'^api/mobile/record_patient/get_specialities/$', views.get_specialities_mobile),
    url(r'^api/mobile/record_patient/get_doctors/$', views.get_doctors_mobile),
    url(r'^api/mobile/record_patient/get_times/$', views.get_times_mobile),
    url(r'^api/mobile/record_patient/record/$', views.record_mobile_patient),
    url(r'^api/mobile/get_records/$', views.get_records_mobile),
    url(r'^api/mobile/cancel_record/$', views.cancel_record_mobile),
    url(r'^api/mobile/get_polyclinic_info/', views.get_polyclinic_info_mobile),
    url(r'^api/mobile/get_doctors_info/', views.get_doctors_info_mobile),
    url(r'^api/mobile/get_patient_info/', views.get_patient_info_mobile),
```

Рисунок 3.5.2 - Шаблони HTTP-запитів

```
@csrf_exempt
def cancel_record_mobile(request):
    error = {"message": "request method should be POST"}

    if request.method == "POST":
        username    = request.POST.get('login', '')
        password    = request.POST.get('password', '')
        record_id   = request.POST.get('record', '')

        result = {}

        user = authenticate(username = username, password = password)
        if user is not None:
            record  = Record.objects.get(id = int(record_id))
            time    = record.record_time

            record.delete()

            time.available = True
            time.save()

            result["success"] = True
        else:
            return HttpResponseForbidden()

        return HttpResponse(json.dumps(result))

    return HttpResponse(json.dumps(error))
```

Рисунок 3.3.3 - Код контролеру відміни запису для API

Щоб зробити клієнтську частину у вигляді веб-сайту була використана мова шаблонів Django. Вона дозволяє генерувати HTML-сторінки шляхом вставки значень змінних в код сторінки.

```
def record_patient_page(request, speciality_id, doctor_id):
    if is_admin(request):
        admin              = Administrators.objects.get(administrator_user = request.user)
        selected_speciality = Speciality.objects.get(id = int(speciality_id))
        selected_doctor     = Doctor.objects.get(id = int(doctor_id))

        patients        = Patient.objects.filter(polyclinic = admin.polyclinic)
        specialities    = Speciality.objects.filter(speciality_polyclinic = admin.polyclinic)
        doctors         = Doctor.objects.filter(polyclinic = admin.polyclinic, speciality = selected_speciality)
        all_times       = Time.objects.filter(doctor = selected_doctor, available = True)
        times           = []

        for time in all_times:
            if time.date > timezone.now():
                times = times + [time]

        if selected_speciality not in specialities:
            speciality_id = specialities[0].id
            return redirect(record_patient_page, speciality_id, doctor_id)

        if selected_doctor not in doctors:
            doctor_id = doctors[0].id
            return redirect(record_patient_page, speciality_id, doctor_id)

        return render(request, 'main/admin/record_patient.html', {'selected_speciality': selected_speciality, 'selected_doctor':
selected_doctor, 'patients': patients, 'specialities': specialities, 'doctors': doctors, 'times': times})
    else:
        return redirect('login_page')
```

Рисунок 3.3.4 - Генерація сторінки запису пацієнта до лікаря

```
    <h1>Записать пациента на прием</h1>
</div>
<table width='100%' border='0px'>
    <tr>
        <td class='table_td' >
            <a>Пациент</a>
        </td>
        <td class='table_td' >
            <select name='patient'>
                {% for patient in patients %}
                    <option value='{{ patient.id }}'>{{ patient.personal_data.surname }} {{ patient.personal_data.name }} {{ patient.personal_data.patron
                {% endfor %}
            </select>
        </td>
    </tr>
    <tr>
        <td class='table_td' >
            <a>Время приема</a>
        </td>
        <td class='table_td' >
            <select name='time'>
                {% for time in times %}
                    <option value='{{ time.id }}'>{{ time }}</option>
                {% endfor %}
            </select>
        </td>
    </tr>
    <tr bgcolor='white'><td colspan='2'></td></tr>
    <tr>
        <td class='submit_td' colspan='2'>
            <input type='submit' value='Добавить'>
        </td>
```

Рисунок 3.3.5 - HTML-код сторінки запису пацієнта до лікаря

4. ІНТЕРФЕЙСИ ТА АНАЛІЗ СИСТЕМИ

4.1. Інтерфейс пацієнта

При вході в систему всі користувачі потрапляють на сторінку авторизації, де вони повинні ввести свій логін і пароль, щоб отримати доступ до їх панелі управління. Користувач отримує її у вигляді HTML-сторінки.

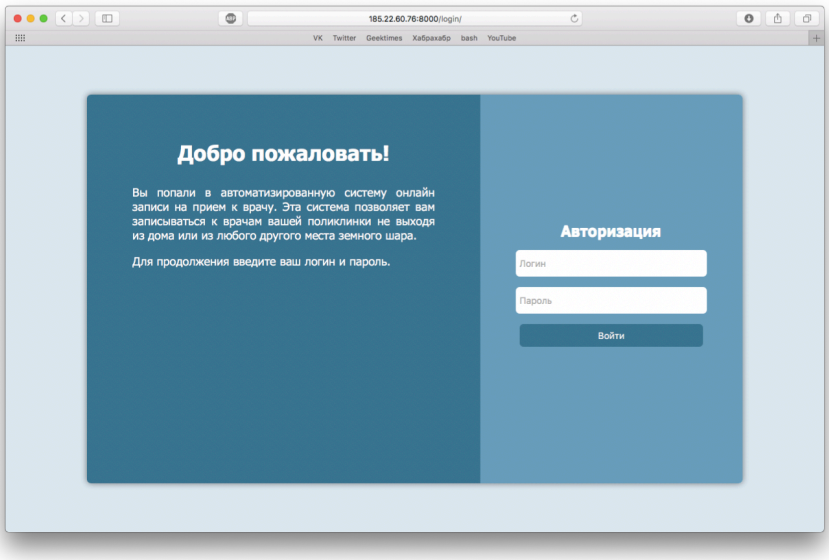

Рисунок 4.1.1 - Сторінка авторизації

Після проходження авторизації пацієнт потрапляє на головну сторінку
своєї панелі керування.

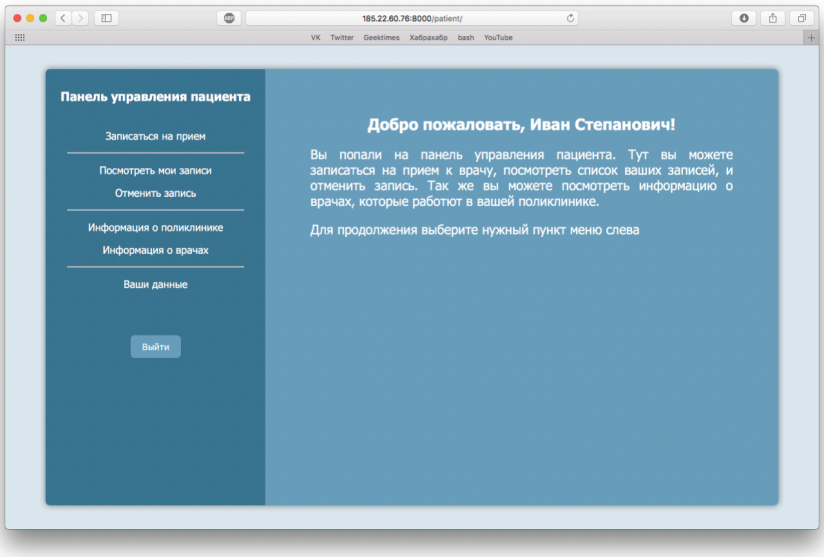

Рисунок 4.1.2 - Початкова сторінка пацієнта

Щоб записатися до лікаря, пацієнту необхідно вибрати зі списку спеціальність лікаря, потім вибрати лікаря і час, до якого він хоче записатися.

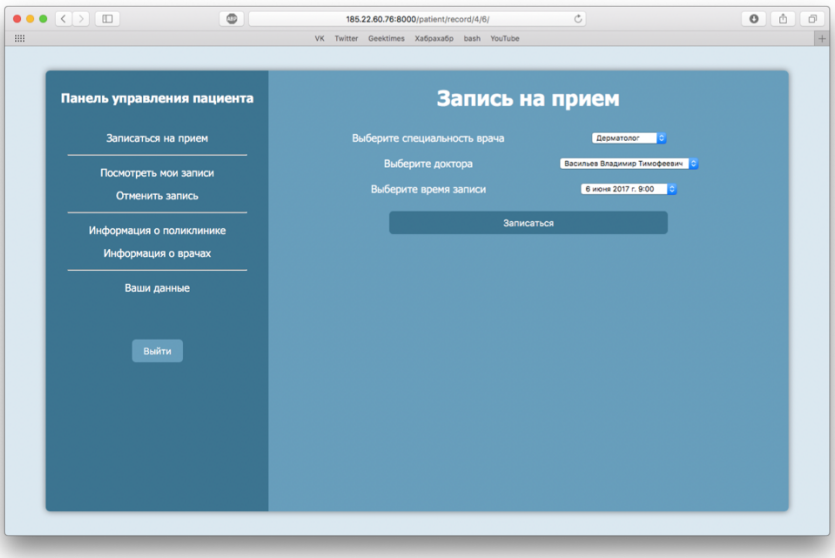

Рисунок 4.1.3 - Запис пацієнта до лікаря

Пацієнт також може подивитися список своїх минулих і поточних записів.

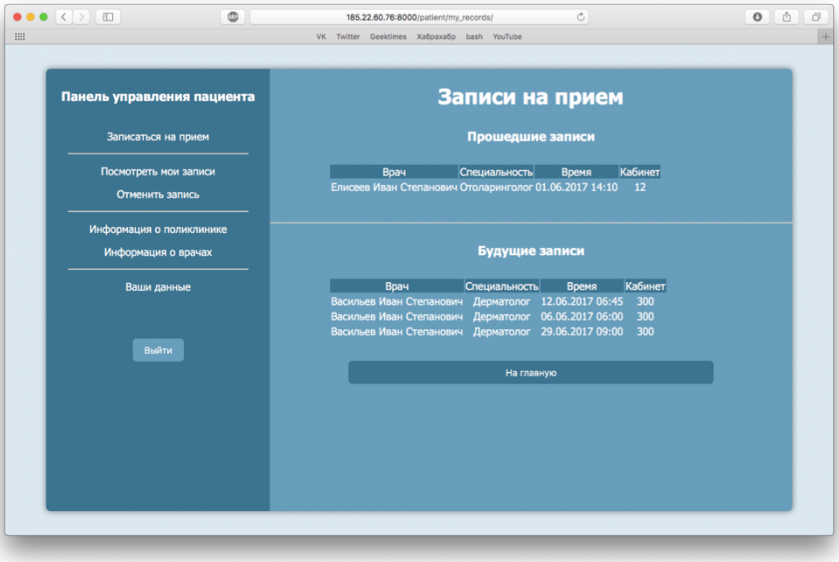

Рисунок 4.1.4 - Список записів пацієнта

Він також може скасувати запис або кілька записів шляхом їх виділення.

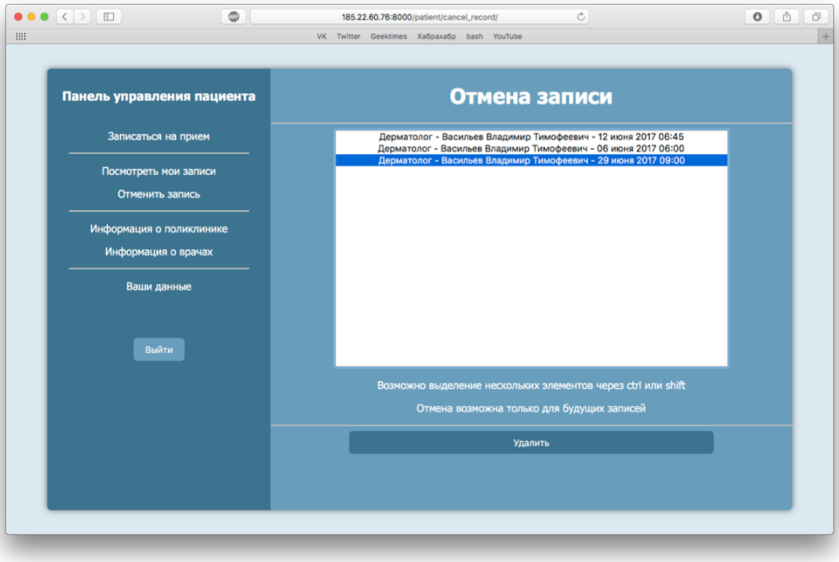

Рисунок 4.1.5 - Скасування запису пацієнта

Пацієнт також може подивитися основну інформацію по поліклініці, таку як назва, сайту, адреса, телефонні номери та електронну пошту.

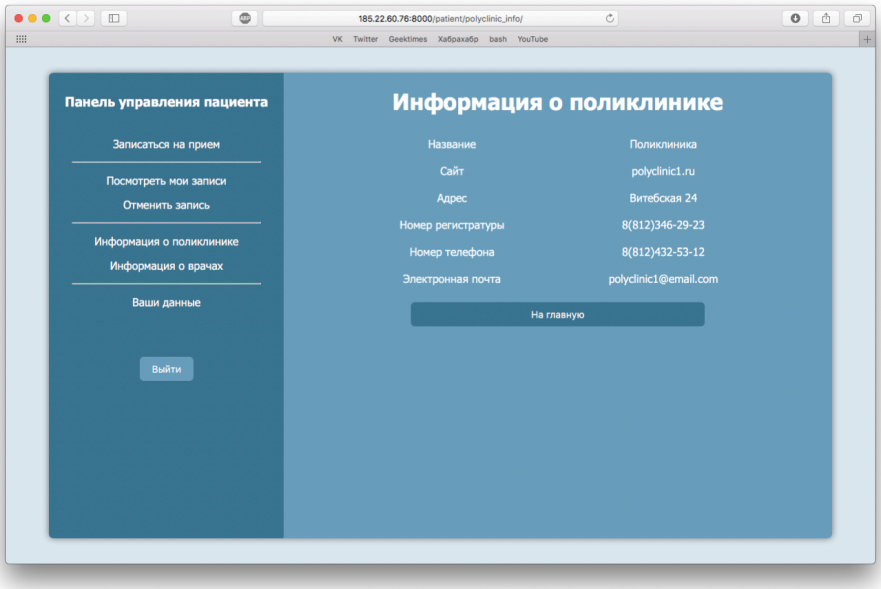

Рисунок 4.1.6 - Інформація про поліклініку

У пункті «Інформація про лікарів» пацієнт може знайти основну інформацію про лікарів, таку як їх ПІБ, спеціальність, номер кабінету і номер телефону кабінету.

4.2. Інтерфейс лікаря

Лікар потрапляє на свою панель керування шляхом авторизації через свій логін і пароль. Інтерфейс складається з чотирьох пунктів.

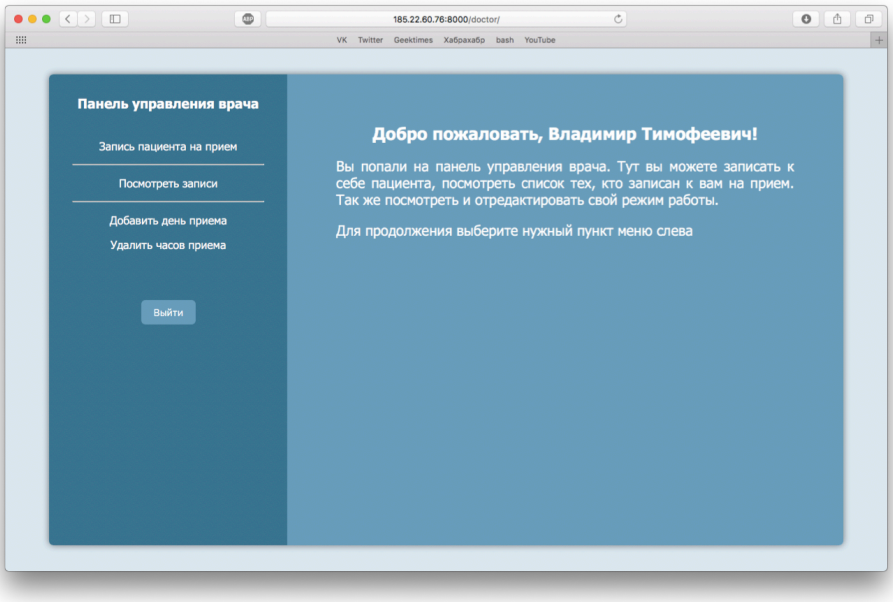

Рисунок 4.2.1 - Початкова сторінка лікаря

У пункті «Запис прийому на прийом» лікар може записувати до себе пацієнтів шляхом вибору пацієнта і часу прийому в випадаючому списку.

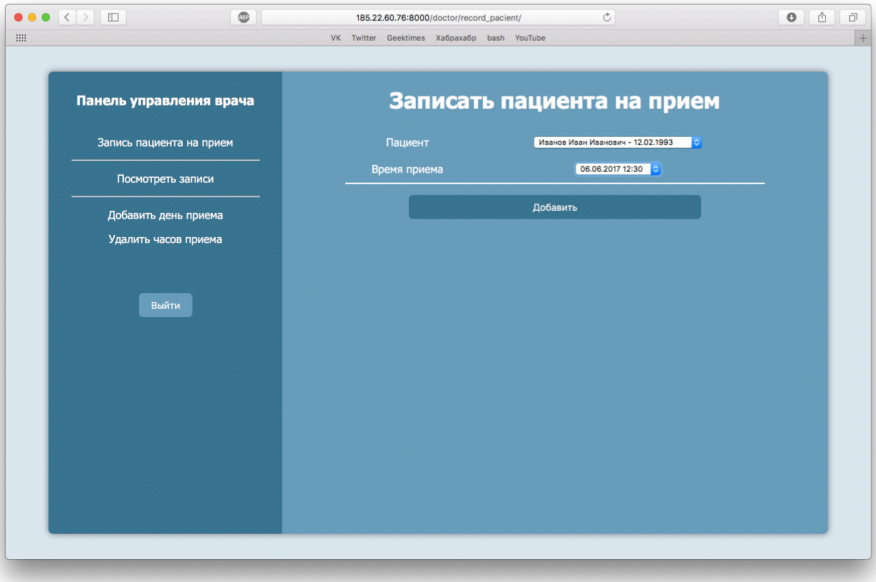

Рисунок 4.2.2 - Запис пацієнта на прийом

Лікар також може подивитися список пацієнтів, які записалися до нього на прийом.

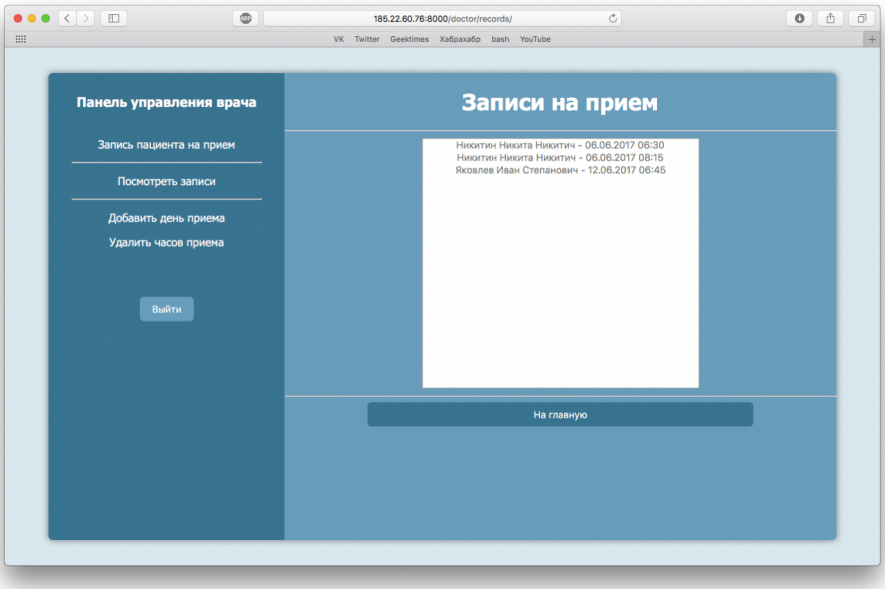

Рисунок 4.2.3 - Записи на прийом

Складання розкладу роботи відбувається на сторінці «Додавання дня прийому». Тут лікар повинен ввести день прийому, час початку і закінчення прийому, при необхідності ввести час початку і закінчення перерви, і ввести інтервал часу, з яким створювати час прийому в базі.

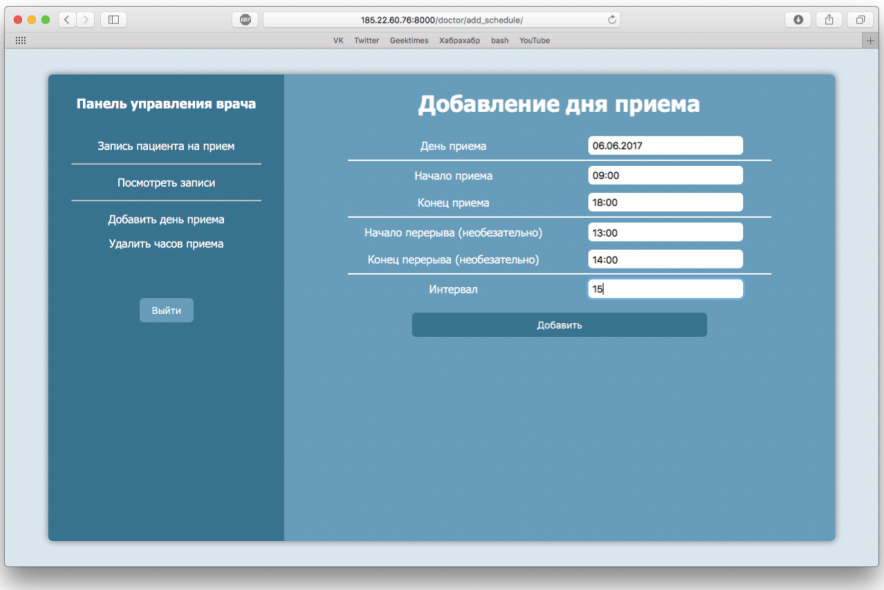

Рисунок 4.2.4 - Додавання дня прийому

Також лікар може видалити години прийому. Якщо потрібно видалити кілька годин, він може їх виділити за допомогою клавіш «Ctrl» або «Shift».

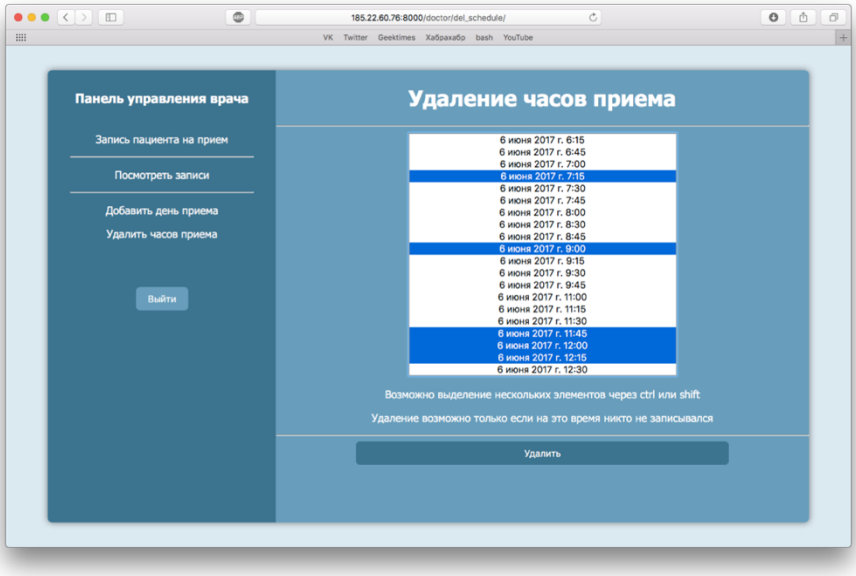

Рисунок 4.2.5 - Видалення годин прийому

4.3. Інтерфейс адміністратора

Адміністратором є людина, яка може керувати всіма сутностями, які відносяться до поліклініки, в якій він працює. Він також включає в себе функціонал працівників реєстратури.

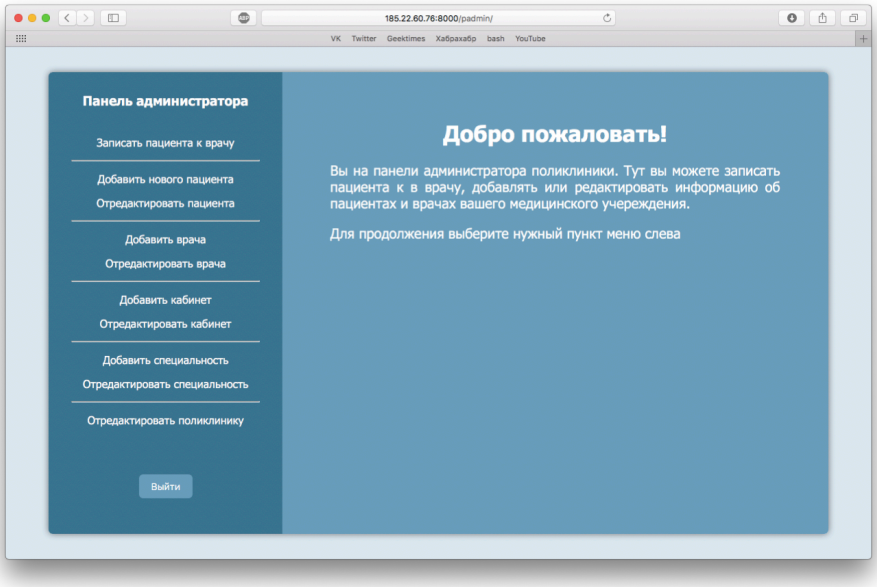

Рисунок 4.3.1 - Початкова сторінка адміністратора

Адміністратор може записувати пацієнтів до лікарів. Вибравши спеціальність лікаря, він отримає список лікарів, які працюють за цією спеціальністю, потім вибирає час і пацієнта, якого потрібно записати на прийом.

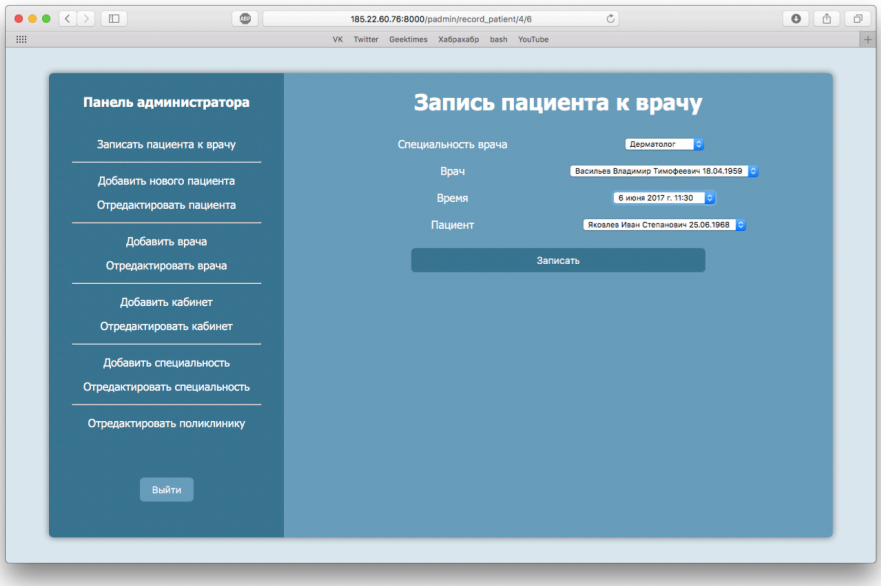

Рисунок 4.3.2 - Запис пацієнта до лікаря

Для додавання кабінету потрібно ввести його номер або назва, так само до нього можна записати його номер телефону.

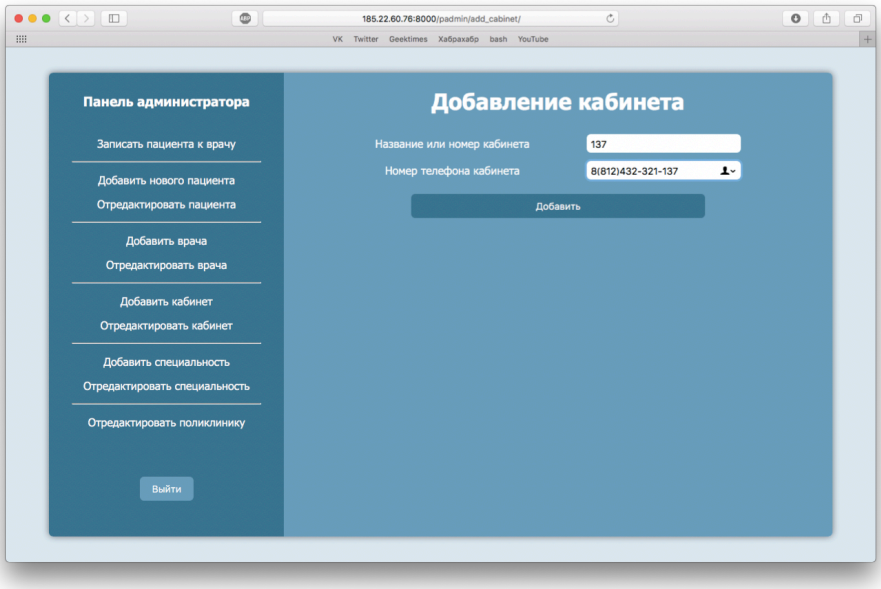

Рисунок 4.3.3 - Додавання кабінету

У разі необхідності, адміністратор може відредагувати існуючі кабінети.

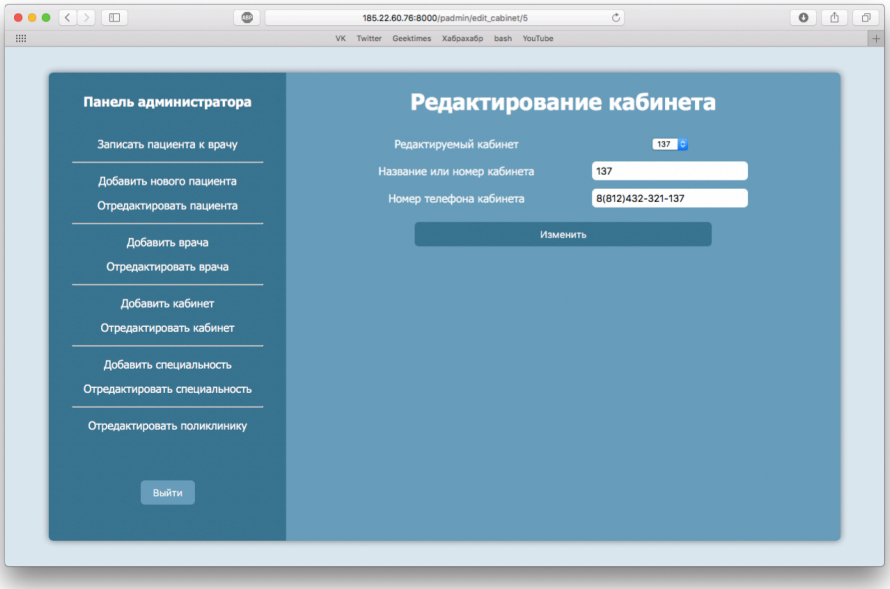

Рисунок 4.3.4 - Редагування кабінету

Для того щоб додати спеціальності лікарів, які працюють в поліклініці, потрібно ввести тільки їх назву.

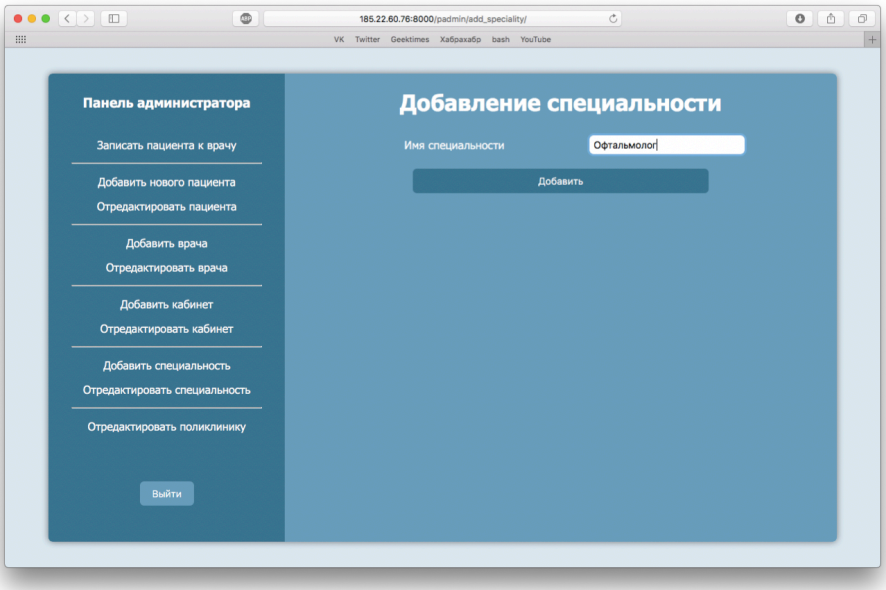

Рисунок 4.3.5 - Додавання спеціальності

Для редагування спеціальності, потрібно вибрати її зі списку і змінити назву.

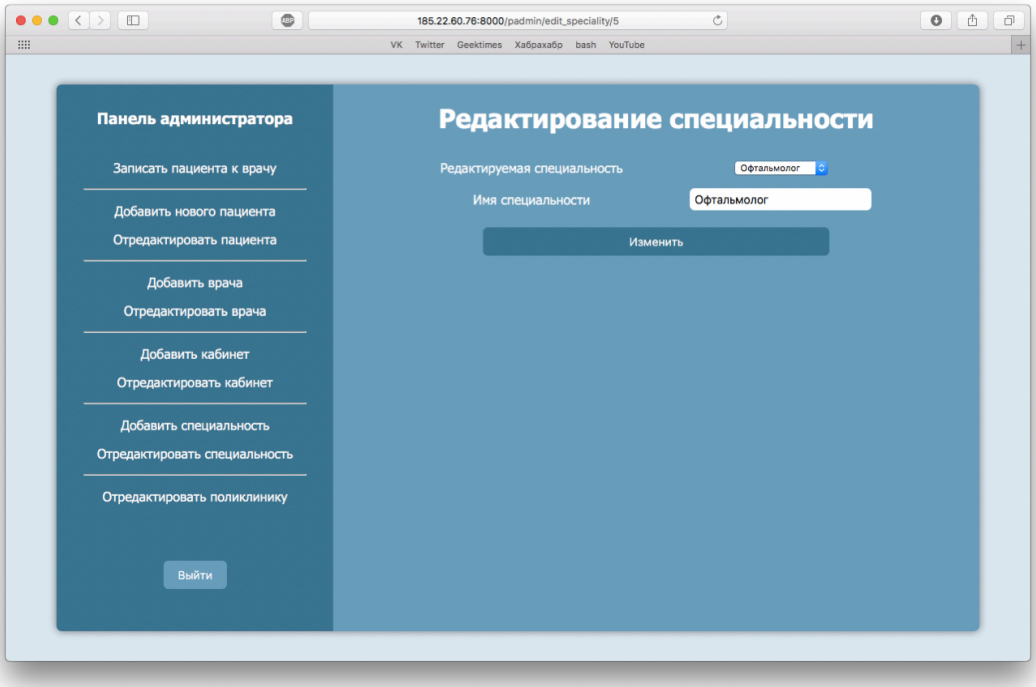

Рисунок 4.3.5 - Редагування спеціальності

Адміністратор також може змінити інформацію по поліклініці, відображається пацієнтам.

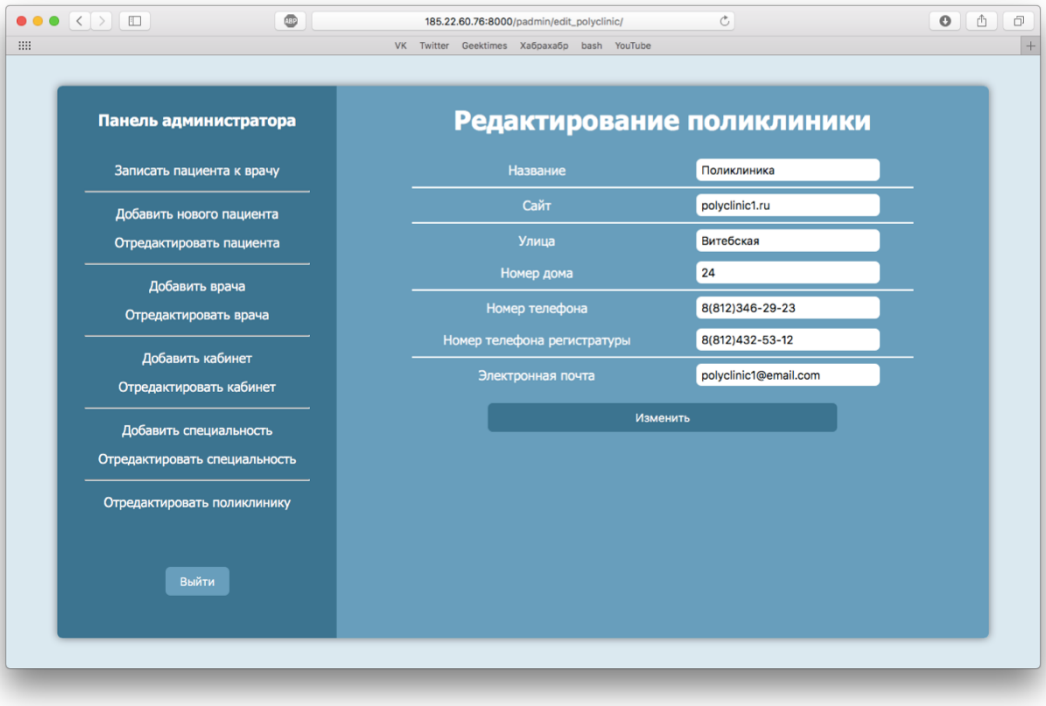

Рисунок 4.3.6 - Редагування поліклініки

ВИСНОВКИ

У магістерській дисертації було проведено аналіз роботи реєстратури поліклініки, як предметної області для дипломної роботи. Здійснено аналіз існуючих програмних продуктів для автоматизації процесів у поліклініці.

Інформатизація поліклініки звільняє працівників установ охорони здоров‘я від рутинних і паперових операцій і дозволяє їм більше часу приділяти пацієнтам. Це призводить до скорочення нецільової витрати інтелектуального багажу, дозволяє працювати і займатися саме медициною.

Реформа медичного обслуговування породжує потужні економічні стимули для впровадження інформаційних систем в медичні установи, однак головна їх перевага полягає в підвищенні якості лікування пацієнтів.

Відділ по введенню і обробці даних, а зокрема реєстратура поліклініки - центральне місце медичного закладу, де концентрується значна частина інформації про пацієнтів і лікувальний процес.

В ході виконання магістерської дисертації була розроблена система, що дозволяє організувати роботу реєстратури та впровадити електронну чергу в медичних установах, зокрема у студентській поліклініці НТУУ КПІ ім. Ігоря Сікорського. Для її реалізації були вивчені існуючі аналогічні рішення, а також оцінені переваги і недоліки різних способів організації електронної черги та автоматизації роботи реєстратури.

Основними цілями розробки було створення простої і зручної системи, як для медичних працівників, так і для пацієнтів. Система не вимагає значних грошових вкладень при установці і експлуатації в медичних установах.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Качмар В. О. Медичні інформаційні системи – стан розвитку в Україні / В. О. Качмар // Український журнал телемедицини та медичної телематики. – 2010. – Т. 8., №1.- С.12-17.
2. Gurumurthi, S. & Benjaafar, S. (2004), Modelling and Analysis of Flexibile Queuing Systems, Wiley Periodicals, Inc. DOI 10.1002/nav.20020.
3. IOM. To Err is Human: Building a Safer Health System: Institute of Medicine (IOM); 1999. 14. IOM. Key Capabilities of an Electronic Health Record System: Letter Report: Institute of Medicine;2003.
4. Підаєв А.В., Пономаренко В.М., Вороненко Ю.В. Інформаційні технології в системі охорони здоров'я. – К.: Здоров’я, 2003 –335 с.
5. Willig, a. (1999), A Short Introduction to Queuing Theory, Technical University Berlin, Tele- communication Networks Group Sekr. FT 5-2, Berlin.
6. Penttinen A.,QUEUING SYSTEMS - INTRODUCTION TO TELETRAFFIC THEORY.
7. Салихова Э.Ш. Информатизация здравоохранения регионального уровня на основе типовых медицинских информационных систем // Врач и информационные технологии. — 2009. — No1. – С. 36-41
8. Zhang, No & Tay (2000). Discrete-event simulation of queuing systems, published in the Proceedings of the sixth youth conference, Ministry of Education, Singapore.
9. Maister, David H. (2005), the psychology of Waiting Lines.
10. Чурпій І.К. Сучасний стан інформатизації в медицині / І.К. Чурпій, Н.В. Чурпій, В.Д. Скрипко // Буковинський медичний вісник. - 2011. - Т. 15, N 1. - С. 171-173.
11. Tijms, H.C; ALGORITHMIC ANALYSIS OF QUEUE - A FIRST COURSE IN STOCHASTIC MODELS, Wiley, Chichester, 2003.

12. Комплексное решение «МКТ-Современная регистратура». URL: http://medcomtech.ru/Products/SovReg/sovreg.html
13. Bose S.J., AN INTRODUCTION TO QUEUING SYSTEMS, Kluwer/Plenum Publishers, 2002
14. F. P. Kelly; NETWORKS OF QUEUE WITH CUSTOMERS OF DIFFERENT TYPES, Journal of Applied Probability, vol. 12, No. 3 (Sept. 1975).
15. Kalra D., Beale T., Heard S., Lloyd D. An HER architecture forArchetyped Health Information Systems.-http://www.openehr.org, 2003.
16. Хвищун А. І., Качмар В.О., Бунь Р.А. Принципи формування єдиної медичної інформаційної системи великого міста //Луганський інформаційний вісник.- 2008.- №1.-