

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки
Факультет комп'ютерних наук та технологій
(повне найменування інституту, факультету)

Кафедра комп'ютерних систем та мереж
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавра

(ступінь вищої освіти (освітній ступінь))

на тему РОЗРОБКА СИСТЕМИ АВТОМАТИЗОВАНОГО ОБЛІКУ В РАЙОННІЙ
АМБУЛАТОРІЇ З ВИКОРИСТАННЯМ МЕСЕНДЖЕРІВ

Виконав: студент 4 курсу, групи КНТ-517
спеціальності _____

123 «Комп'ютерна інженерія»

(код і найменування спеціальності)

Освітня програма (спеціалізація) _____

«Комп'ютерна інженерія»

Богатирьова Т.О.

(прізвище та ініціали)

Керівник Зеленьова І.Я.

(прізвище та ініціали)

Рецензент Шило Г.М.

(прізвище та ініціали)

м. Запоріжжя
2021 рік

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

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

ЗАТВЕРДЖУЮ

Завідувач кафедри Кудерметов Р.К.

“ ” 2021 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТА

Богатирьової Тетяни Олександрівни

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка системи автоматизованого обліку в районній амбулаторії з використанням месенджерів

керівник проекту (роботи) Зеленьова Ірина Яківна к.т.н, доцент

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

затверджені наказом вищого навчального закладу від “17” березня 2021 року № 81

2. Строк подання студентом проекту (роботи) 06 травня 2021 року

3. Вихідні дані до проекту (роботи) інформаційна система, яка реалізує пошук користувачів та внесення даних до бази в інтерфейсі телеграм чатбота

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

1) Аналіз предметної області, вибір наявних методів, інструментів роботи з системою

2) Проектування алгоритму роботи системи, інфологічне моделювання бази даних;

3) Розробка та випробовування системи.

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

Блок-схема алгоритму роботи системи;

Порівняльна характеристика використаних технологій;

Схема бази даних;

Демонстрація роботи інформаційної системи.

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-3	Зеленьова І.Я. к.т.н. доцент		
нормоконтроль	Щербак Н.В. ст. викл.		

7. Дата видачі завдання «01» березня 2021 року.

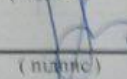
КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз методів та алгоритмів розробки, перегляд недоліків та переваг, вибір технологій	01.03.2021 р.	
2	Розробка алгоритму роботи системи та схеми БД	15.03.2021 р.	
3	Реалізація інформаційної системи	20.03.2021 р.	
4	Оформлення результатів у ПЗ	10.04.2021 р.	
5	Оформлення графічного матеріалу	10.05.2021 р.	
6	Оформлення допоміжного матеріалу	17.05.2021 р.	

Студентка

Керівник проекту (роботи)


(підпис)


(підпис)

Т.О. Богатирьова
(прізвище та ініціали)

І.Я. Зеленьова
(прізвище та ініціали)

РЕФЕРАТ

ПЗ: 61 с., 13 рис., 3 табл., 4 додатки, 35 джерел.

ІС, МЕСЕНДЖЕР, MYSQL, БАЗА ДАНИХ, ЧАТБОТ, PYTHON, API,
СКБД

Об'єктом розробки є інформаційна система автоматизованого обліку районної амбулаторії на базі чатбота месенджера.

Мета проекту – розробка інформаційної системи автоматизованого обліку зі швидким та зручним доступом для користувачів.

Проведено аналіз предметної області та можливих методів розробки, внаслідок якого було обрано чатбот, розроблений на мові Python, в месенджері Телеграм в якості інтерфейсу для користувачів та MySQL-сервер для збереження занесених даних. Побудовано інфологічну модель бази даних та блок-схему алгоритму дій на які запрограмований бот. Проведено тестування системи.

Розроблено інформаційну систему, яка забезпечує швидкий доступ до дистанційного спілкування пацієнтів та лікарів, а також реалізує зручний та зрозумілий для користувачів інтерфейс.

ЗМІСТ

Скорочення та умовні позначки	6
Вступ.....	7
1 Аналіз предметної області	10
1.1 Аналіз об'єкту розробки	10
1.2 Аналіз мови розробки	13
1.3 Аналіз середовища розробки	19
1.4 Вибір СКБД	21
1.5 Методи захисту інформації.....	26
2 Проектування інформаційної системи	28
2.1 Аналіз вимог до системи.....	30
2.2 Побудова алгоритму роботи ІС	31
2.3 Структура бази даних.....	34
3 Реалізація інформаційної системи районної амбулаторії	37
3.1 Створення чатбота у Телеграм	38
3.2 Створення проекту у IDE.....	40
3.3 Робота з СКБД	42
3.5.1 Підключення бібліотек та бази даних	45
3.5.2 Функціональна частина розробки чатбота	47
3.6 Випробування та демонстрація роботи системи	52
Висновки.....	57
Перелік джерел посилання	59

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ

ІС – інформаційна система

МІС – медична інформаційна система

БД – база даних

ПЗ – пояснювальна записка

СКБД – система керування базами даних

API – інтерфейс програмування додатків, Application Programming Interface

IDE – інтегроване середовище розробки, Integrated Development Environment

ВСТУП

Сучасний світ наповнений невраховуваним числом технологічних рішень, без яких вже не уявити повноцінне життя. Нас оточує повсюдне впровадження інформаційних технологій до усіх сфер життєдіяльності, будь це просто соціальні зв'язки чи ринкові відносили, наукова чи культурна діяльність. Наразі інформатизація вже стосується навіть таких галузей як державне і місцеве управління, матеріальне виробництво, охорона здоров'я тощо.

Інформатизація – соціально-економічне і науково-технічне явище, що характеризується процесом створення умов для формування і використання інформаційних ресурсів, реалізації інформаційних відносин у суспільній сфері.

Впровадження інформаційних технологій та обчислювальної техніки – стратегічний напрям реформування медичної галузі.

Швидкість, якість отримання та обробки інформації стали одним з найважливіших умов підвищення рівня медичної допомоги. Сучасні методи інформатики дозволяють забезпечити комплексний аналіз даних, отриманих з численних джерел, оптимізацію рішень при обстеженні, діагностиці, прогноз перебігу захворювань і виборі тактики лікування [1].

Тож актуальною проблемою сучасної охорони здоров'я, та безперечно, ключовим моментом в досягненні ефективності його інфраструктури є інформатизація. Тобто створення єдиного інформаційного простору для всіх зацікавлених сторін: медичного персоналу, пацієнтів тощо.

Застосування персональних комп'ютерів та мобільних пристроїв дозволяють значно скоротити час на витрати ручної праці співробітників, підвищує якість роботи, виключає помилки через неуважність (людський фактор), а також дозволяє значно скоротити час, витрачений на отримання необхідного результату, а так само на пошук потрібної інформації.

Тим паче пандемія, що почалася у 2019 році, значно підвищила актуальність проблеми дистанційного зв'язку з різними установами та мінімізації особистих

контактів між працівниками та клієнтами (у випадку медичного закладу: між пацієнтами та працівниками реєстратури, лікарями).

Це стосується навіть не тільки пандемії, але й звичного життя, адже хвороб, що передаються при особистому контакті не так вже й мало. До того ж, водночас це піклування й про комфорт та час пацієнтів, адже нині пошук інформації, що цікавить, має займати якомога менше часу, що може врятувати, можливо, й чиєсь життя.

Серед існуючих інформаційних ресурсів і засобів інформатизації відчутна певна несумісність, що обмежує можливість обміну інформацією між ними; недостатньо кваліфікованого персоналу, необхідного для їх експлуатації, нестачу технічних засобів для комфортної роботи працівників з інформаційними системами.

Невідповідність існуючих технічних і технологічних засобів особливо вражає на фоні експоненціального прискорення технічного прогресу в ХХІ столітті [2].

Інформатизація в сфері охорони здоров'я безупинно розвивається, проте велика кількість медичних закладів все ще мають дуже низький її рівень, що особливо стосується закладів невеликих населених пунктів, інформацію про які в інтернет-мережі знайти якщо не неможливо, то дуже важко і не мають ніякої інформації про персонал.

В Україні з 2016 року поступово впроваджується медична інформаційна система (МІС) Хелсі (сайт helsi.me), проте багато закладів з регіонів все ще не мають досить ресурсів для повноцінної роботи з цією МІС, адже для її впровадження кожен лікар повинен мати у вільному доступі персональний комп'ютер, наданий державою чи установою, бо в такому випадку він зобов'язаний вести повну звітність в електронному вигляді, тоді як до мобільних пристроїв в даному випадку система погано адаптована і дуже програє у зручності.

Інформаційна система (ІС) – являє собою сукупність бази даних, усього програмно-апаратного комплексу засобів для її зберігання, зміни і пошуку інформації та роботи з користувачем, що представлено, зокрема, СКБД [3] – системи

управління базами даних, що є спеціальним пакетом програм, який забезпечує створення, використання баз даних багатьма користувачами тощо.

Тож метою даного дипломного проекту є розробка інформаційної системи районної амбулаторії, яка буде вирішувати проблему швидкого, зручного пошуку та доступу до інформації про лікарів районної амбулаторії та їх пацієнтів, запису пацієнтів на прийом та дистанційного зв'язку за умови максимальної доступності для населення та мінімізації витрат на її розробку та обслуговування.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Попередньо до виконання проектування та реалізації система потребує чіткого аналізу до використання тих чи інших методів розробки, визначення доцільності їх використання, переваг та недоліків.

Основними напрямками аналізу є:

- визначення та аналіз об'єкту розробки, тобто з чим саме будуть мати діло користувачі;
- визначення та аналіз мови програмування, яка буде доцільнішою до використання для розробки визначеного у попередньому пункті об'єкту;
- визначення та аналіз інтегрованого середовища розробки, оптимального до використання з обраною мовою та швидкістю, зручністю роботи;
- визначення та аналіз СКБД, до якої буде занесено та у яких буде зберігатися інформація, що буде отримана під час експлуатації інформаційної системи;
- визначення можливих методів захисту бази даних.

1.1 Аналіз об'єкту розробки

Виходячи з мети проекту, потребується розробка ІС, яка буде відповідати наступним вимогам:

- доступність для населення: треба враховувати вимоги до апаратури, популярність серед користувачів, можливість використання на різних носіях та платформах;
- відповідний функціонал та швидкість виконання операцій;
- зручний, зрозумілий для користувачів інтерфейс;

– витрати на розробку та обслуговування інформаційної системи, можливість її вдосконалення.

Серед носіїв, якими користується більшість населення, можна виділити персональні комп'ютери та смартфони, тож ІС має працювати на обох видах пристроїв.

Найкращим варіантом для реалізації є створення інформаційної системи на базі існуючого додатку, вже популярного серед населення. Це значно спрощує процес розробки, та не вимагає додаткових витрат на створення окремого додатку зі сторони розробника та зберігання даних нового додатку для клієнта, що буде значно сприяти популярності проекту. Усім цим вимогам чудово відповідають месенджери.

Месенджер – це клієнтська програма, яка реалізує миттєвий обмін повідомленнями та різноманітними вкладеннями (фото, відео, файли тощо). Найчастіше месенджери підтримують голосовий і навіть відеозв'язок, не потребуючи за це окремої оплати, тобто буде можливо вийти на зв'язок з лікарем чи пацієнтом особисто, але дистанційно без будь-яких витрат.

Також значна їх перевага в тому, що сучасні популярні месенджери підтримують можливість створення чатботів.

Чатбот – програма штучного інтелекту, яка імітує інтерактивну розмову людини за допомогою ключових, заздалегідь розрахованих фраз-запитів користувача [4].

Упродовж 5-7 лютого 2021 року Київський міжнародний інститут соціології (КМІС) провів власне всеукраїнське опитування, метою якого було дізнатися – який месенджер на території України є найпопулярнішим.

Дослідження виявило, що найбільш поширеним засобом для комунікацій є Viber, яким користується 66% опитаних. Другим за популярністю є Месенджер Фейсбуку – 35%. На третьому-четвертому місцях знаходяться Телеграм (використовують 24% опитаних) та WhatsApp (21%) [5].

WhatsApp надає можливість створювати чатботи реалізовані через WhatsApp Business API, однак він платний, а сама можливість внести оплату не гарантує доступ, адже рішення про ствердження залишається за WhatsApp.

API – набір заздалегідь визначених підпрограм, функцій, бібліотек тощо.

Месенджер є відносним новачком у сфері чатботів, рішення впроваджуються повільно, тому неможливо передбачити, коли будуть введені нові функції, і чи станеться це взагалі [6].

До того ж, WhatsApp взагалі має дуже низьку функціональність у порівнянні з конкурентами, використовуючись більшістю клієнтів лише за своїм основним призначенням – обміном повідомленнями та файлами між користувачами, тож створення чатботу на його базі не найкраще рішення.

В свою чергу Viber, Телеграм та Месенджер Фейсбуку мають відкритий для всіх бажаючих API, проте серед лідерів месенджерів найбільш адаптованим до різноманітних носіїв та платформ використання є Телеграм. Порівняння наявності месенджерів на платформах наведено у таблиці 1.1

Таблиця 1.1 – Наявність лідерів-месенджерів на платформах [7]

Платформи	Viber	Facebook	Телеграм
Android	+	+	+
IOS	+	+	+
Wp	+	—	+
Windows	+	—	+
Mac	+	—	+
Linux	+	—	+
Web	—	+	+

Крім того, даний месенджер має дуже широкий функціонал не тільки для створення ботів різних типів, а й організації простору користувачів. Вже більше року тому назад було введено оновлення, в якому Телеграм отримав чергову перевагу для своїх користувачів в порівнянні з іншими месенджерами – з’явилися папки, у які можна відсортувати потрібні чати, канали, ботів тощо за власним бажанням, а отже потрібний чатбот може бути завжди у швидкому та зручному доступі для будь-якого клієнта інформаційної системи за потреби.

Тож Телеграм – один з найперспективніших месенджерів сьогодення, розвиток та популярність якого у перспективі, дуже вірогідно, пережене усі, наведені вище. За усіма критеріями вибір саме цієї платформи задля створення інформаційної системи є найбільш оптимальним.

1.2 Аналіз мови розробки

Мова програмування – це система позначень для опису алгоритмів та структур даних, певна штучна формальна система, засобами якої можна виражати алгоритм [8]. Якщо описати більш розгорнуто, мовою програмування можна назвати безліч заздалегідь визначених, одноманітних і зрозумілих виконавцю (читай: інтерпретатору / компілятору / комп’ютеру / програмісту) інструкцій, призначених для запису послідовно з метою їх розуміння та виконання якимось пристроєм, що є частиною обчислювальної машини. Також мова програмування повинна володіти рядом особливостей: інструкцій повинно бути обмежене число, і всі їх повинні знати; інструкції повинні вибудовуватися певним чином для отримання певних результатів, і всі повинні про це знати; повинні існувати правила написання інструкцій і всі повинні їх знати; кожна конструкція мови повинна однозначно співвідносити те що написано з тим, що було потрібно позначити [9].

Всі мови програмування діляться на два види – мови низького і високого рівня:

Мови низького рівня – це спосіб написання комп'ютерних інструкцій на апаратній мовою, тобто в машинних кодах (у вигляді послідовності нулів і одиниць). Мови низького рівня жорстко орієнтовані на конкретний тип обладнання (система управління процесором, кожен тип процесора має свій машинний код) [10].

Мови високого рівня – це мови програмування, які використовують абстрактні дані і контролюють структури, символічні позначення і змінні, дозволяють записувати код у зручному для логіки людини вигляді, тоді як низького рівня, навпаки адаптовані так, щоб втратити якомога менше ресурсів для сприйняття коду машиною.

Ці мови орієнтовані не на систему інструкцій того чи іншого процесора, а на систему операторів (інструкцій), характерну для написання певного класу алгоритмів до мови розуміння апаратними засобами кожна програма має пройти шляхом компіляції та інтерпретації [11].

Мови високого рівня простіше у використанні, оскільки їх завдання – обслуговувати потреби програміста, а не визначати можливості комп'ютера.

Мови низького рівня, близькі до машинної мови, дозволяють створювати програми, які працюють швидше і дозволяють більш ефективно використовувати ресурси комп'ютера.

Розробка системи потребує мову, яка відповідає критеріям:

- Мова високого рівня;
- Популярна: велика кількість кваліфікованих спеціалістів;
- Швидкодіюча;
- Нескладна: простий функціонал та підтримка.

На сьогоднішній день існує велика кількість різних мов програмування і у кожного з них своя сфера застосування, але все ж для проведення аналізу на вибір кращої мови для написання чатбота, необхідно вибрати кілька найпопулярніших мов, щоб між ними проводити аналіз, тож в цьому розділі звернемося до статистики по популярності мов [12].

Для перевірки актуальності тієї чи іншої мови програмування при створенні нової програмної системи, можна звернутися до визначення індексу ТІОВЕ.

Індекс спільноти програмістів ТІОВЕ виступає показником актуальності, найбільш використовуваних мов програмування за останні пару років. До того ж, оновлення показників робиться кожного місяця, а самі рейтинги базуються на кількості кваліфікованих розробників з усього світу, курсів і сторонніх постачальників, крім того розрахунки базують свої показники за популярністю запитів у таких пошукових системах, таких як Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube, Baidu тощо.

Тож можна наочно переглянути весь шлях взльотів та падінь популярності кожної з мов програмування, не боячись, що інформація може застаріти чи буде зроблена на основі нерелевантної вибірки. Згідно до нього, на рисунку 1.1 наведено рейтинг найпопулярніших мов програмування на травень 2021 року в порівнянні з даними за травень 2020.

May 2021	May 2020	Change	Programming Language	Ratings	Change
1	1		C	13.38%	-3.68%
2	3	▲	Python	11.87%	+2.75%
3	2	▼	Java	11.74%	-4.54%
4	4		C++	7.81%	+1.69%
5	5		C#	4.41%	+0.12%

Рисунок 1.1 – Рейтинг ТОП-5 мов програмування за індексом ТІОВЕ [12]

Відповідно до цього можна зробити висновок, що на 2020 рік значну популярність здобула мова C і наразі є найпопулярнішою мовою програмування. У п'ятірку кращих мов увійшли також: Python, Java, C++, C#. За поданими даними можна зробити висновок, що популярність мов Java та C сильно падає, а популярність таких мов як Python та C++ продовжують зростати.

Коли справа доходить до швидкості і продуктивності, жодна інша мова не може бути настільки ж ефективною, як C++. Для розробки чатботів на її основі це стало додатковою перевагою. C++ – дуже стабільна мова, яка дозволяє швидко

створювати складні додатки за допомогою машинного навчання і розробки нейронної мережі [13].

Багато користувачів називають C ++ мовою програмування середнього рівня, адже за її допомогою виконується розробка з використанням функцій мови як високого, так і низького рівнів.

До того ж C ++ є мовою об'єктно-орієнтованого програмування, підтримує концепцію класів, перевантаження операторів, множинне спадкування, віртуальні функції, обробку виключень тощо, що є значною перевагою при розробці.

Всі алгоритми чатбота на базі штучного інтелекту можуть бути повністю написані на C++ [13], надаючи величезні можливості розробнику та крім цього володіє масштабованістю, адже на C ++ можна виконувати розробку програм для найрізноманітніших платформ і систем.

Просте серед недоліків також можна виділити підтримку модульності – підключення інтерфейсу зовнішнього модуля через препроцесору вставку заголовки (#include) серйозно уповільнює компіляцію, при підключенні великої кількості модулів [14]

Python – це високорівнева мова програмування загального призначення, орієнтований на підвищення продуктивності розробника і читання коду. Мова включає в себе:

- динамічну типізацію;
- автоматичне керування пам'яттю;
- повне самостереження;
- механізм обробки винятків
- підтримку багатопотокових обчислень
- практичні структури даних високого рівня [15].
- складається з вбудованих конструкцій, які мають функцію виконання
- зрозумілого коду в малих і великих масштабах. [16].

На даний момент серед розробників чатботів Python набув найбільшої популярності. Величезна низка вже існуючих ботів сьогодення написана саме на цій

мові і переважна кількість досі створюються. Це, в основному, стається завдяки простішому значно простішому синтаксису, що вище вже згадувався.

- Python оснащений AIML або мовою розмітки штучного інтелекту, що значно спрощує створення синтаксису для складних функцій чатбота. Ключові позитивні аспекти Python для розробки чатботів на базі штучного інтелекту;
- Python використовується в основному для створення всіх діалогових компонентів, що розміщуються на веб-сервері;
- Набір інструментів обробки природної мови (NLP), який є ключовим елементом інтелектуальних чатботів, написаний на Python;
- Python – універсальна мова з дуже простим синтаксисом;
- Python також оснащений великою бібліотекою для виконання безлічі проектів AI;
- Python дозволяє скласти стратегічну карту доріг для оцінки напрямки проекту;
- Модульний і інтерактивний код з високою сумісністю на декількох платформах [14].

Для більш чіткого уявлення у порівнянні доцільності вибору мови, можна навести приклад використання мов програмування в одній і тій самій ситуації.

Лістинг 1.1 – Фрагмент найпростішого коду на C++ [17].

```
#include <iostream>
int main()
{
    std::cout << "Hello, world!" << std::endl;
    return 0;
}
```

Лістинг 1.2 – Фрагмент найпростішого коду на Python [17].

```
print("Hello, world!")
```

Різниця між лістингами 1.1 та 1.2 наявна та цілком виправдовує попередні твердження щодо складності семантики мов.

Проте для вибору цього замало, тож можна звернутися до таблиці 1.н, в якій наведено ще декілька параметрів, відносно порівняння мов програмування C++ та Python.

Таблиця 1.2 – Порівняння C++ та Python [18]

Параметр	C++	Python
Компіляція	Компіляція	Інтерпретація
Простота у використанні	Непросто	Легко
Типізація	Статична	Динамічна
Збір сміття	Не підтримує	Підтримує
Типи	Перевірка під час компіляції	Прив'язані до значень
Швидке прототипування	Неможливе	Можливе
Функції	Обмеження по типу параметрів	Не має обмежень по типу параметрів
Ефективність	Важко підтримувати	Легко
Швидкість виконання	Швидкий	Незначно, але повільніший

Виходячи з усього вищезазначеного, недоліки Python не значно переважають його перевагу перед мовою C++, тож найоптимальнішим вибором для програмування та реалізації інформаційної системи районної амбулаторії буде використання мови програмування Python.

1.3 Аналіз середовища розробки

В узгоджені до обраної мови для розробки ІС районної амбулаторії, треба підібрати також оптимальне IDE, яке б відповідало усім потребам розробника щодо мови програмування, швидкості та зручності роботи.

IDE (integrated development environment) – інтегроване середовище розробки.

На відміну від текстових редакторів для коду, таких як Notepad++, Sublime або Atom, більш потужні інтегровані середовища розробки пропонують безліч зручних функцій [19], таких як:

- підсвічування синтаксису;
- зручний інтерфейс, що підлягає налаштуванню;
- розширені засоби навігації;
- наявність компілятора та відладника.

IDE поєднує у собі не тільки ці компоненти, але і ряд інших. Деякі з них поставляються з додатковими інструментами для автоматизації, тестування і візуалізації процесу розробки. Термін «інтегроване середовище розробки» означає, що надається все необхідне для перетворення коду в функціонуючі програми [20].

Існує безліч видів середовищ розробки, тому доцільно розглянути лише ті, які підходять під критерії для розробки даної ІС. Головне – з вільним чи дешевим доступом та сумісні з мовою Python.

Серед найпопулярніших за такими критеріями можна виділити: Visual Studio Code, Eclipse та PyCharm.

Visual Studio Code – це продукт корпорації Microsoft, розроблений для того, щоб дозволити розробникам писати код без завантаження масивної Visual Studio (3 ГБ +). Visual Studio Code – це простий редактор з відкритим вихідним кодом, який однаково добре працює в Windows, OS X і Linux. Основні функції коду Visual Studio включають підтримку більше 30 мов, автозаповнення, просту наві-

гацію і т. д. Він також включає засоби налагодження та Git для полегшення розробки [21].

Переваги:

- підтримка більш ніж 30 мов, а також основні мови Microsoft, такі як ASP.NET, C# і т.;
- невеликий розмір гарантує швидку установку і використання.

Недоліки:

- підтримка розширень потребує поліпшення;
- оновлення на ОС Linux може бути непростим завданням.

Eclipse – це середовище повністю безкоштовне, з відкритим вихідним кодом, написаним переважно на Java.

Проте, її модульна архітектура дозволяє використовувати Eclipse і з іншими мовами за допомогою величезної кількості різноманітних плагінів. Має проблеми з продуктивністю, середовище розробки далеко не нове і не так добре адаптовано до сучасної архітектури. Робота з Python якраз-таки вимагає встановлення окремого плагіна, тому це є істотним недоліком, який тільки сповільнить роботу в даній IDE.

PyCharm – це інтегроване середовище розробки спеціально для Python, так що має широкий набір можливостей, таких як автозавершення та інспекції коду, підсвічування помилок, виправлення, налагодження, система контролю версій і рефакторинг. IDE доступна на Microsoft Windows, Linux і MacOS. Є безкоштовна і платна професійна версії. Професійна IDE має кілька додаткових функцій, але безкоштовної версії досить для більшості завдань [19].

Плюси IDE:

- має важливі вбудовані функції;
- розроблено професіоналами спеціально для Python;
- швидкий у роботі з обраною мовою у порівнянні з іншими IDE.

Мінуси:

- потребує 8 ГБ оперативної пам'яті.

Таблиця 1.3 – Порівняння характеристик IDE.

	Visual Studio Code	Eclipse	PyCharm
Наявність на платформах	Windows, Linux	Windows, Linux, macOS	Windows, Linux, macOS
Можливість роботи з Python	Плагін	Плагін	Вбудована
Безплатна версія	Наявна	Наявна	Наявна
Пам'ять	Мало	Багато	Багато
Швидкість роботи	Середня	Середня	Висока

Виходячи з наведеного, оптимальний варіант для розробки – PyCharm, не дивлячись на те, що він потребує досить багато оперативної пам'яті, Eclipse та Visual Studio Code мають істотні недоліки через те, що використовують розширення та плагіни для роботи з обраною мовою Python.

1.4 Вибір СКБД

Крім клієнтської частини, що реалізується як чатбот у месенджері Телеграм, інформаційна система включає до себе базу даних, куди буде занесено і збережено дані, введені до боту.

База даних (БД) являє собою сукупність спеціальним чином організованих даних, що зберігаються в пам'яті обчислювальної системи і відображають стан об'єктів та їх взаємозв'язків в даній предметній області [22]. У порівнянні з іншими способами зберігання інформації бази даних мають низку значних переваг, а

саме виключення надмірності й можливих протиріч даних, підтримка їх коректності та цілісності, швидкий пошук необхідних даних.

Серед сучасних, популярних СКБД більшою частиною наводять Oracle, MS SQL Server, MySQL, PostgreSQL.

Критеріями для відбору стали:

- вимоги до апаратури;
- відповідність необхідних властивостей для побудови додатку інформаційної системи;
- надійність та безпека СКБД;
- ціна за її використання.

Щоб визначитися, яка ж саме СКБД підійде найкраще, слід розглянути кожну з обраних на відповідність цим критеріям.

СКБД Oracle – один з лідерів СКБД – вона є вкрай надійною, фактично це еталон надійності серед подібних систем, може використовуватись на різних операційних системах

Проте вона зроблена з урахуванням роботи з величезними базами даних і різноманітними функціями, тому вартість Oracle може виявитися непомірно високою, особливо для невеликих організацій, а крім того для нормальної роботи системи необхідні значні ресурси вже відразу ж після установки, тому можливо буде потрібно значно модернізувати обладнання для застосування СКБД Oracle [23]. Створення інформаційної системи з використанням СКБД Oracle вимагає високої кваліфікації розробників, що також призводить до збільшення вартості розробки системи.

СКБД MS SQL Server, так само як і СКБД Oracle, можна віднести до СКБД промислового рівня.

СКБД MS SQL Server має широкий спектр функціональних можливостей і за своєю продуктивністю не поступається СКБД Oracle. У ряді випадків в якості СКБД краще використовувати MS SQL Server, як продукт компанії Microsoft.

MS SQL Server – СКБД, яка використовується для роботи з базами даних розміром від персональних до великих баз даних масштабу підприємства.

SQL Server являє собою інтегроване рішення з управління та аналізу даних, яке допоможе організаціям різного масштабу:

- будувати, розгортати і керувати промисловими додатками, які є більш безпечними, масштабованими і надійними;
- збільшувати продуктивність інформаційних технологій, зменшуючи складність побудови, розгортання і управління додатками по роботі з базами даних;
- розділяти дані між платформами, додатками і пристроями для полегшення з'єднання внутрішніх і зовнішніх систем;
- контролювати вартість, не нехтуючи якістю виконання, доступністю, і безпекою.

До недоліків MS SQL Server слід віднести можливість використання її тільки у середовищі операційної системи Windows.

Крім того слід підкреслити високу вартість обох СКБД, що не дозволяє використовувати ці СКБД у інформаційній системі для бюджетних організацій та враховуючи те, що розроблювана інформаційна система не відноситься до середовищ надмірного обігу інформації, цілком виправданим рішенням буде відмовитися від перших двох варіантів, адже в даному випадку витрати на настільки потужні ресурси будуть нераціональними.

PostgreSQL – надає безліч різних можливостей, досить надійна і має хороші характеристики по продуктивності. PostgreSQL вільно поширюється і має відкритий вихідний код.

PostgreSQL вигідно відрізняється від багатьох інших СКБД. Вона володіє практично всіма можливостями, які є в інших базах даних (комерційних або Open Source), а також деякими додатковими.

Перелік функціональних можливостей PostgreSQL:

- транзакції;
- вкладені запити;
- уявлення;
- посилавна цілісність – зовнішні ключі;

- складні блокування;
- типи, визначені користувачем;
- спадковість;
- правила та перевірка сумісності версій [24].

Однак у поточний час найбільш поширеною є СКБД MySQL

MySQL є реляційною СКБД з відкритим вихідним кодом, розробкою якої займається корпорація Oracle. Вона активно підтримується своїм розробником - оновлення виходять кожні декілька місяців. MySQL вже протягом довгого часу є однією з найпопулярніших СКБД [25].

MySQL поширюється за двома ліцензіями: GNU GPL 2 та пропрієтарною. За пропрієтарною ліцензією поширюється MySQL Enterprise Edition, яка включає в себе набір просунутих утиліт для управління базами даних та технічну підтримку, що не є елементом першої необхідності, в особливості на етапі розробки. Таким чином, використання MySQL є фактично безкоштовним.

Працювати з MySQL можливо або через командну оболонку, або засобами одного з багатьох графічних інтерфейсів (MySQL Workbench, HeidiSQL, phpMyAdmin та ін.), але ця СКБД дає можливість створювати та працювати лише з елементами самої бази даних – таблицями. Клієнтський додаток, в свою чергу, потрібно розроблювати засобами однієї з мов програмування, для яких доступний API MySQL. Це ускладнює розробку інформаційної системи, оскільки в цьому випадку окрім знання SQL необхідно мати навички розробки графічного додатку на сторонній мові програмування [26].

MySQL не має визначених вимог до апаратного забезпечення, тому що у кожному окремому випадку СКБД працює у різних режимах з різними налаштуваннями. До того ж, на цьому етапі дуже складно судити про системні вимоги клієнтського додатку, який необхідно буде розробити для повного функціонування інформаційної системи.

MySQL – серверна СКБД, яка може використовуватись у середовищі багатьох сучасних операційних системах. Вона відповідає вимогам до необмеженої кількості користувачів, що одночасно працюють із БД, високої швидкості вико-

нання команд та наявність простої та ефективної системи безпеки, підтримує можливість встановлювати різний доступ для користувачів різних рівнів, досить проста та невимоглива до апаратури, до того ж відноситься до «open-source» технологій.

У порівнянні з СКБД PostgreSQL СКБД MySQL має достатньо значні переваги:

- MySQL зазвичай набагато перевершує PostgreSQL за швидкістю роботи. Крім того, в MySQL 4.0 реалізований кеш запитів. Він дозволяє у багато разів збільшити швидкість обробки;
- за кількістю користувачів MySQL також набагато перевершує PostgreSQL. Тому код тестується значно прискіпливіше і досвідченим шляхом доведено велика його надійність, ніж у PostgreSQL;
- MySQL працює в середовищі Windows краще, ніж PostgreSQL. MySQL Server запускається як сьогодення (рідне) Windows-додаток (в NT / 2000 / XP - сервіс), в той час як PostgreSQL запускається в середовищі емуляції, Cygwin;
- MySQL оснащений великою кількістю API для інших мов і підтримується великою кількістю існуючих програм, ніж PostgreSQL;
- всі типи таблиць в MySQL (крім InnoDB) реалізовані у вигляді файлів (по одній таблиці у файлі), що значно полегшує створення резервних копій, перенесення, видалення і навіть створення символьних посилань між базами даних і таблицями, навіть якщо сервер знаходиться в неробочому стані [27].

Зважаючи на усе вищенаведене, найдоцільнішим вибором для реалізації СКБД районної амбулаторії є MySQL, адже вона є некомерційною, швидкою та зручною в експлуатації, має значні переваги у порівнянні з іншими СКБД у контексті розроблюваної ІС.

1.5 Методи захисту інформації

В наш час дуже поширеною проблемою є захист інформації від несанкціонованого доступу, що у випадку роботи з медичною інформаційною системою набуває ще більшої актуальності, адже в базі мають зберігатися конфіденційні дані про пацієнтів та роботу спеціалістів

Захист інформації – діяльність щодо запобігання витоку інформації, що захищається, несанкціонованих і ненавмисних дій на захищає інформацію [28].

На поточний момент методи захисту можна розділити на кілька основних груп:

- способи зміни виду і структури передачі інформації в мережі і її зберігання, наприклад, застосування криптографічних методів захисту;
- організаційні методи захисту даних, засновані на управлінні інформаційними системами і регламентування правил безпеки;
- технічні і технологічні методи захисту інформації, які передбачають використання спеціальних програмних і апаратних засобів [29].

В інформаційній системі районної амбулаторії найбільшу увагу слід запобігання доступу до інформації про пацієнтів сторонніх осіб. Враховуючи те, що ІС має використовувати базу даних для внесення та збереження даних про користувачів, потребується працювати з забезпеченням її безпеки, адже, на жаль, бази даних досить вразливі до можливих несанкціонованих проникнень. В даному проекті в якості СКБД було обрано MySQL, тож об'єкту огляду є найпопулярніші проблеми – збереження паролів від доступу та SQL-ін'єкції.

В цілях захисту паролів зазвичай використовують такі методи як винесення даних доступу до сервера БД у окремий файл опцій, який вже потім імпортується у основний код програми за потреби зв'язку з базою даних.

SQL-ін'єкції – це атака, спрямована на сайт чи додаток тощо, в ході якої конструюється SQL-вираз з призначеного для користувача введення шляхом простої конкатенації (наприклад, \$ query = "SELECT * FROM users WHERE id =". \$

_ REQUEST ["id"]). У разі успіху атакуючий може змінити логіку виконання SQL-запиту так, як це йому потрібно [30].

Найбільш надійним способом запобігання SQL-ін'єкції є використання параметризованих SQL-параметрів. Основна ідея полягає в тому, що якщо позиція параметрів явно задана, то можна абсолютно безпечно передавати SQL-запити бази даних, виключаючи можливість для параметрів самим стати SQL-виразами (в тому числі шкідливими). Варто зауважити, що інші механізми, такі як використання примусової приведення типів (наприклад, за допомогою функції `intval ()`) в зв'язці з екрануванням рядків такими функціями, як `mysql_real_escape_string ()` або `addslashes ()`, не є абсолютно безпечними. Проблема в тому, що існують деякі варіанти для їх обходу, а отже, до їх використання необхідно підходити з максимальною увагою [31].

Наявно можна розглянути різницю між звичайним SQL-виразом та параметризованим можна розглянути на прикладі лістингів 1.3 та 1.4.

Лістинг 1.3 – Приклад звичайного SQL-виразу

```
SELECT mail, pass, login, user_name
FROM users
WHERE mail = ' x ';
```

Лістинг 1.4 – Приклад параметризованого SQL-виразу

```
SqlConnection conn = новый SqlConnection (_connectionString);
conn.Open ();
строка s = " SELECT mail, pass, login, user_name
" +
" FROM users
WHERE mail = @email";
SqlCommand cmd = новый SqlCommand (s);
cmd.Parameters.Add ("@ email", электронная почта);
Читатель SqlDataReader = cmd.ExecuteReader ();
```

Приведений у лістингу 1.4 параметризований код пропонує наступні переваги чистої продуктивності:

- менше конкатенація строк;
- не потрібно турбуватися про будь-якому екранування рядки вручну;

– у db представлена більш загальна форма запиту, тому вона, ймовірно, вже хешированого і збережена як попередньо скомпільований план виконання.

Тож, підводячи підсумки до розділу, внаслідок аналізу доступних засобів серед відповідних за вимогами, було обрано розробку інформаційної системи районної амбулаторії у вигляді чатбота у Телеграмі, задля підвищення популярності, зацікавленості користувачів та зручності й роботи. На відміну від сторонніх додатків, чатбот не займає зайвого місця на пристрої користувача та вимагає лише доступу до інтернету. До того ж Телеграм – найбільш кросплатформений додаток серед розглянутих і надає можливість підтримувати декілька сесій одного користувача з різних пристроїв одночасно.

Як мова для розробки була обрана Python, завдяки своїй популярності серед розробників в усьому світі, досить простому синтаксису, який значно спрощує розробку та при цьому не має значних недоліків в порівнянні з C++

Відповідно до вибору мови було обрано інтегровано середовище розробки PyCharm як одне з найоптимальніших у балансі за користю, зручністю та швидкістю роботи.

Система ж складається з двох частин: не тільки клієнтської, для розробки якої відібрано було все вищезазначене, але й серверної, що представлено у вигляді СКБД, в якій буде створено базу даних, підключену до ІС в цілях збереження даних користувачів, які зареєстровані у системі та забезпечення їх безпеки.

Базуючись на вибрані засоби розробки можна починати роботи над проектуванням системи.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Важливим етапом розробки є конкретний розбір функціоналу системи: що вона має виконувати та яким саме чином, забезпечення яких умов має виконуватись для її стабільної роботи.

Сутність розроблюваної системи полягає у тому, щоб забезпечити для лікарів та пацієнтів районної амбулаторії інформаційною системою, яка буде забезпечувати лікарів та пацієнтів швидким доступом до інформації, налагодженням безплатного дистанційного зв'язку в цілях мінімізації особистих контактів не тільки за умов епідемічної ситуації у країні, але й з зауваженням на те, що окрім хвороби COVID-19 існує велика низка інфекційних захворювань, що передаються при особистому контакті між людьми.

До того ж, дана система буде виступати і в якості мінімізації витраченого часу на запис до лікаря, адже у звичайному режимі, через реєстратуру, це може зайняти невизначену кількість часу: зважаючи на витрати часу на шлях як до лікарні, так і для повернення з неї, наявність та величину черги у самої реєстратури.

Аналогом до роботи може частково виступати МІС «Хелсі», або окремих сайт лікарні, які зазвичай організують запис у телефонному режимі під час розмови з оператором, проте проблема кожної з них полягає у тому, що вони мають низку недоліків для даної конкретної ситуації.

Якщо взяти до огляду МІС «Хелсі», пацієнти також обирають запис на прийом до лікаря онлайн, проте все одне змушені перед прийомом слідувати до реєстратури, аби роздрукувати талон на прийом, тому що лікар, який підпорядкований системі «Хелсі» зобов'язаний підтверджувати проведення прийому фізичною наявністю талону.

Крім того, кожний прийом має бути в повній мірі відмічений у системі безпосередньо на місці. Враховуючи те, що з лікарської сторони функціонал МІС погано адаптований до мобільного використання, кожен лікар, який є користувачем «Хелсі» повинен мати персональний комп'ютер для постійного використання. Проте державним установам, тип паче, невеликих населених пунктів, важко організувати належний рівень технічного забезпечення.

Якщо взяти до огляду сайти медичних установ – все набагато гірше, адже вони виступають лише у вигляді інформаційних сторінок для клієнтів, тобто про автоматизацію запису, чи ведення інформації про пацієнтів немає мови. Запис за допомогою телефонної розмови, безперечно, краще, аніж у реєстратурі, проте все

одно не найзручніший варіант, адже крім того, що цей процес не є авторизованим і в працівників займає це час, у наш час більшість вважатиме за краще не вступати зайвий раз до особистої розмови, адже дослідження показують, що 59,30% пацієнтів намагаються зв'язатися з установою в неробочий час, щоб призначити зустріч. В цілому, дослідження чатботів показують, що 65% споживачів вважають за краще взаємодіяти з чатботами, а не з людьми [32].

До того ж, клієнти будь-якої системи та/чи установи потребують для себе негайного відгуку та 23% потенційних клієнтів завершують зв'язок, коли їм доводиться підкорятися списку очікування. В такому випадку чатботи – ідеальне рішення, адже вони відповідають користувачу одразу і виключають можливість черги, одночасно обслуговуючи велику кількість клієнтів.

Можна згадати про додатки та сайти, на яких людям пропонують онлайн-консультації з лікарями різних профілів, проте частіше за все це або не безплатно, або може бути організоване лише через розмову повідомленнями. Та частіше за все це разові консультації, адже особисті зустрічі таких лікар не проводить. Тож аналогом таку систему навряд можна назвати, хоч і серед чатботів є подібні розробки, в тому числі такі, де лікарем виступає штучний інтелект, який ставить діагнози та дає поради на основі діагностики через опитування чи просто скарги користувачів-клієнтів.

Інших подібних аналогів до ІС районної амбулаторії при розробці знайти не вдалося, тож можна зробити висновок, що розробка є актуальною і потребує реалізації.

2.1 Аналіз вимог до системи

Основна ціль інформаційної системи районної амбулаторії – створити комфортний для користувачів інструмент для пошуку лікарів та інформації про них,

організації дистанційного зв'язку та запису на прийом в умовах мінімальних вимог до розробки та використання.

Виходячи з цього, мають виконуватися функції:

- занесення та зберігання інформації про лікарів та пацієнтів;
- пошук пацієнтів за прізвищем;
- пошук лікарів за їх спеціалізацією;
- знесення та збереження інформації про прийом кожного пацієнта;
- перегляд прийомів конкретного пацієнта як спрощений варіант амбулаторної картки;
- запис пацієнта на прийом до конкретного лікаря;
- можливість вийти на зв'язок з будь-яким користувачем.

До того ж реалізація повинна забезпечувати ще декілька вкрай важливих факторів.

Надійність:

- тобто система має бути доступною та працездатною в будь-який можливий час;

Доступність:

- система має бути доступна на більшості можливих популярних платформ та мати швидкий доступ;

Продуктивність:

- для системи не має бути проблемою підтримка одночасного звернення великої кількості користувачів.

2.2 Побудова алгоритму роботи ІС

Проектування інформаційної системи вимагає побудови чіткого алгоритму, за яким вона буде побудована за допомогою програмування.

Першою командою є «Старт» – стандартна команда, яка запускає кожного телеграм-чатбота. Вона запускає вітання та перше меню з кнопками, серед якого має зробити користувач – розподіл за роллю користувача, чи є він лікарем чи пацієнтом.

За вибором гілки пацієнта, програма має перевірити, чи є користувач вже зареєстрованим у базі даних. Якщо людина вперше користується ботом, чатбот має запросити користувача до реєстрації.

У випадку, коли користувач вже є у базі, чатбот продовжує роботу, пропускаючи гілку реєстрації та надає можливість пошуку лікарів, що працюють у районній амбулаторії.

За вибором спеціалізації пацієнт повинен отримати список лікарів, які за нею працюють та таким же чином обрати людину, яка його цікавить.

Та за введенням прізвища, чатбот має видати інформацію про лікаря, що цікавить цього пацієнта: повне ПІБ, спеціалізацію, контактний телефон та розклад роботи. Крім того, надати можливість вийти з ним на зв'язок у телеграм та записатися до нього на прийом.

З іншої сторони, за вибором гілки лікаря, користувач так само перевіряється на реєстрацію у базі, проте якщо його там немає, чатбот має видати повідомлення про помилку, адже лікарів у ІС обмежена кількість та аби виключити можливі помилкові реєстрації, їх має бути зареєстровано розробником.

При успішній авторизації користувач-лікар має отримати можливість пошуку пацієнта за ПІБ, або переглянути записи на прийом пацієнтів, які обрали його як бажаного спеціаліста.

За вибором гілки пошуку користувач-лікар отримує інформацію про пацієнта та можливість зв'язатися з ним особисто, або переглянути записи лікарів про прийоми цього пацієнта, які в спрощеній формі є аналогом звичної амбулаторної картки.

Блок - схема роботи алгоритму зображено на рисунку 2.1.

Таким чином маємо точне уявлення про те, як саме має виглядати робота програми і які будуть до неї вимоги.

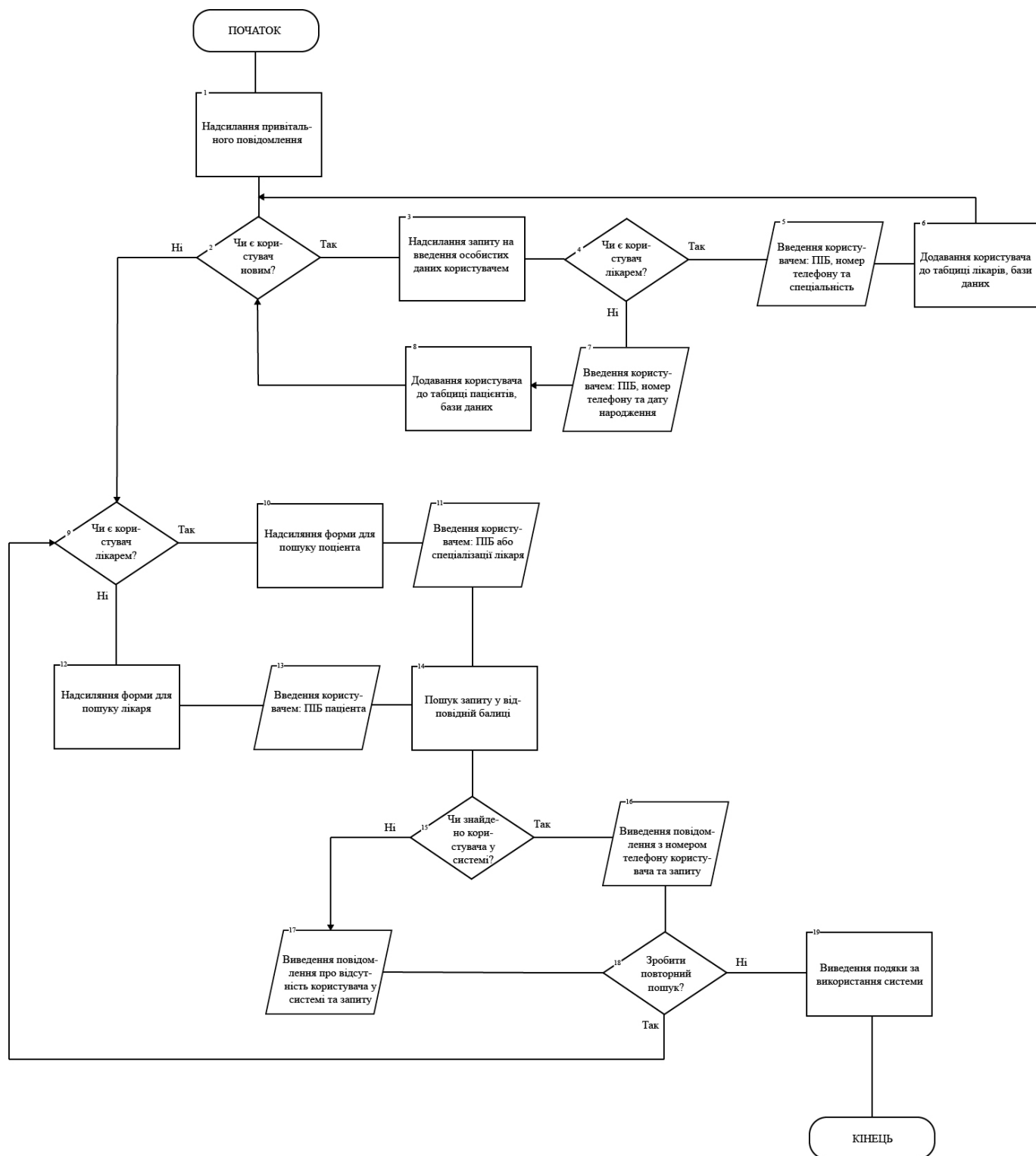


Рисунок 2.1 – Блок-схема алгоритму

2.3 Структура бази даних

Наступний крок – концептуальне (інфологічне проектування) — процес створення моделі інформації, яка не залежить від будь-яких фізичних аспектів її подання, логічних і фізичних подробиць реалізації БД.

При аналізі потреб ІС слід виділити сутності, атрибути та зв'язки.

Сутністю (entity) називається окремий тип об'єкта (людина, місце або річ, поняття або подія тощо), який потрібно представити в базі даних. Атрибутом (attribute) називається властивість, яке описує деяку характеристику розглянутого об'єкта; зв'язок (relationship) - це те, що об'єднує кілька сутностей [3].

В кінцевому підсумку концептуальне проектування передбачає отримання семантичної моделі, яка відображає інформаційний зміст конкретної предметної області, так званої концептуальної моделі даних – ER-моделі.

ER-модель – це представлення бази даних у вигляді наочних графічних діаграм «сутність»-«зв'язок», створених за деяким технічним завданням. Діаграма «сутність»-«зв'язок» представляє в графічному вигляді сутності, атрибути та зв'язки.

На підставі аналізу вимог до системи та розробки алгоритму було розроблено 5 сутностей, інформація про які зберігається в базі даних:

- сутність Doctor, яка містить інформацію про лікарів районної амбулаторії;
- сутність Таблиця Patient, яка містить інформацію про пацієнтів районної амбулаторії;
- сутність Priem, яка містить записи про прийоми пацієнтів в якості амбулаторних карток;
- сутність Zapis, яка містить дані про записи на прийоми до лікарів.

Сутність Doctor має такі поля:

- d_id – унікальний ідентифікатор користувача, первинний ключ таблиці;

- d_name – ПІБ лікаря;
- spec_id – номер спеціалізації лікаря, зовнішній ключ з таблиці

Specialty;

- d_phone – телефон лікаря;
- work_time – розклад роботи лікаря.

Сутність Specialty має такі поля:

- spec_id – унікальний номер спеціалізації, первинний ключ таблиці;
- name_spec – назва спеціалізації.

Сутність Patient має такі поля:

- p_id – унікальний ідентифікатор користувача, первинний ключ таблиці;

- p_name – ПІБ пацієнта;
- p_address – адреса пацієнта;
- p_phone – телефон пацієнта;
- birth_data – дата народження пацієнта.

Сутність Priem має такі поля:

- priem_id – унікальний номер запису про прийом, первинний ключ таблиці;

- zametki – коментар лікаря щодо прийому;
- priem_date – дата проведення прийому;
- p_id – ідентифікатор пацієнта, що прийшов на прийом, зовнішній ключ з таблиці Patient;

- d_id – ідентифікатор лікаря, що провів прийом, зовнішній ключ з таблиці Doctor.

Сутність Zapis має такі поля:

- zap_id – унікальний номер запису до лікаря, первинний ключ таблиці;
- d_id – ідентифікатор лікаря, до якого записався пацієнт, зовнішній ключ з таблиці Doctor;

- p_id – ідентифікатор пацієнта, що записується на прийом, зовнішній ключ з таблиці Patient;

- date_zap – обрана дата запису;
- time_zap – обраний час запису.

Виходячи з вищеописаного можна визначити зв'язки між сутностями, завдяки реалізації зовнішніх ключів – ключів, що в інших сутностях є первинними.

Сутності «Specialty» та «Doctor» знаходяться у зв'язку типу 1:Б (один-до-багатьох), адже одному лікарю може належати лише одна спеціалізація, але за однією спеціалізацією може працювати безліч лікарів.

Тоді як між сутностями «Doctor» та «Patient» складається зв'язок Б:Б – один лікар може приймати у себе безліч пацієнтів, а один пацієнт потребує прийому у великої кількості різних спеціалістів.

Проте за допомогою введення асоціативних відношень «Zapis» та «Priem» воно розкладається на 1:Б та Б:1, що, на відміну від Б:Б, може бути реалізовано у середовищі реляційних БД, відповідно до обраної СКБД MySQL.

Таким чином отримується зв'язок: «Doctor» та «Zapis», «Patient» та «Zapis», «Doctor» та «Priem», «Patient» та «Priem» як 1:Б.

На підставі створених сутностей побудована ER-модель, що представлена на рисунку 2.2

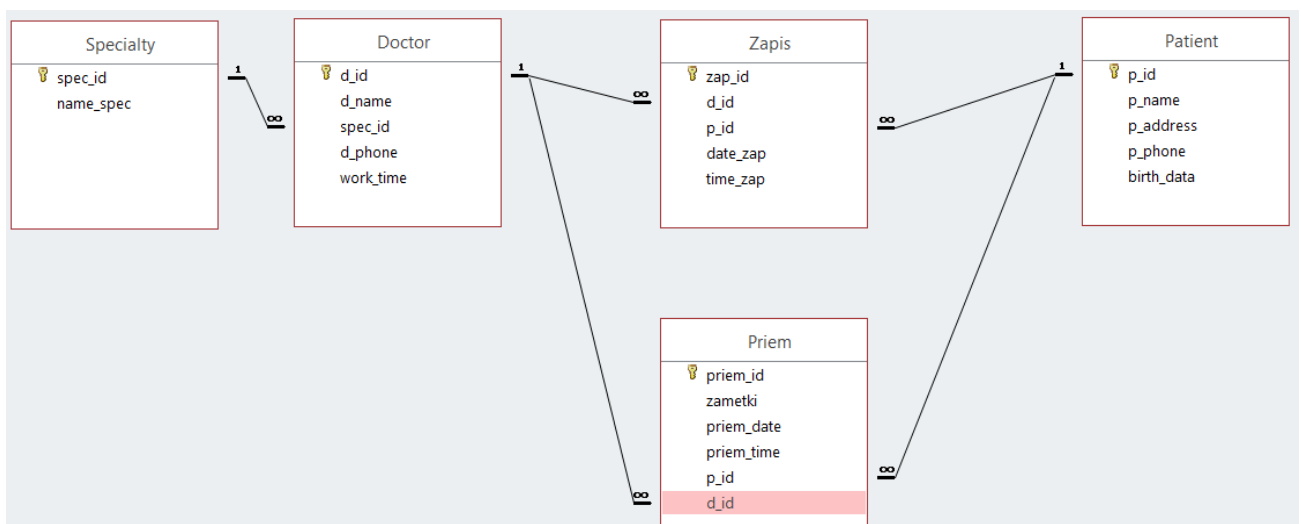


Рисунок 2.2 – ER-модель таблиць та зв'язків між ними ІС районної амбулаторії

Тож в процесі проектування було докладно проаналізовано функціонал інформаційної системи, її призначення та актуальність на сьогоднішній день. Визначено та

спроектовано згідно вищезазначеного алгоритм роботи системи, розроблено ER-модель таблиць та зв'язків між ними з чітким визначенням сутностей.

3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ РАЙОННОЇ АМБУЛАТОРІЇ

В результаті проведеного аналізу вимог та обраних методів розробки, можна перейти до фактичного створення інформаційної системи.

Реалізація потребує:

Встановлення на комп'ютер мови Python та IDE PyCharm. Наявності месенджера Телеграм.

Телеграм можна використовувати як за допомогою окремого додатку, так і відкривши його веб-версію у будь-якому браузері – працювати в будь-якій версії він буде однаково.

Також для зручності та спрощення розробки телеграм-чатботу необхідно встановити відповідну бібліотеку. Адже на сьогодні конкурентоспроможний софт по визначенню занадто складний і потужний, щоб програмісти писали їх повністю самостійно – це нерентабельно, не реалістично та займає занадто багато часу, якщо говорити про темпи розробки. Багато алгоритмів, класів і функцій, які повинні бути в коді, вже кимось реалізовані і оприлюднені в якості бібліотек. Вони не тільки полегшують розробку і прискорюють процес, але і знижують кількість помилок в коді [33], що робить їх незамінними в роботі програміста.

В даному випадку система потребує встановлення бібліотеки `pyTelegramBotAPI`, яка бере на себе всі нюанси відправки та отримання запитів, дозволяючи сконцентруватися безпосередньо на логіці [34].

Встановити бібліотеку можна через термінал за допомогою команди `«pip install pytelegrambotapi»`. Дочекатися повідомлення про успішне встановлення `«Successfully installed pip-8.1.1»` та за потреби, оновити, якщо з'явилося попередження з командою-пропозицією для цього.

Крім того, треба також встановити бібліотеку для взаємодії з базою даних MySQL: «mysql-connector-python-master».

Маючи повний необхідний програмний комплекс, можна почати безпосередню розробку системи.

3.1 Створення чатбота у Телеграм

Перш за все, для початку розробки інформаційної системи потрібно створити чатбота у Телеграмі, аби потім вже наповнювати його функціоналом за допомогою розробки у IDE.

Для даної цілі у месенджері Телеграм є власний зручний бот – @BotFather.

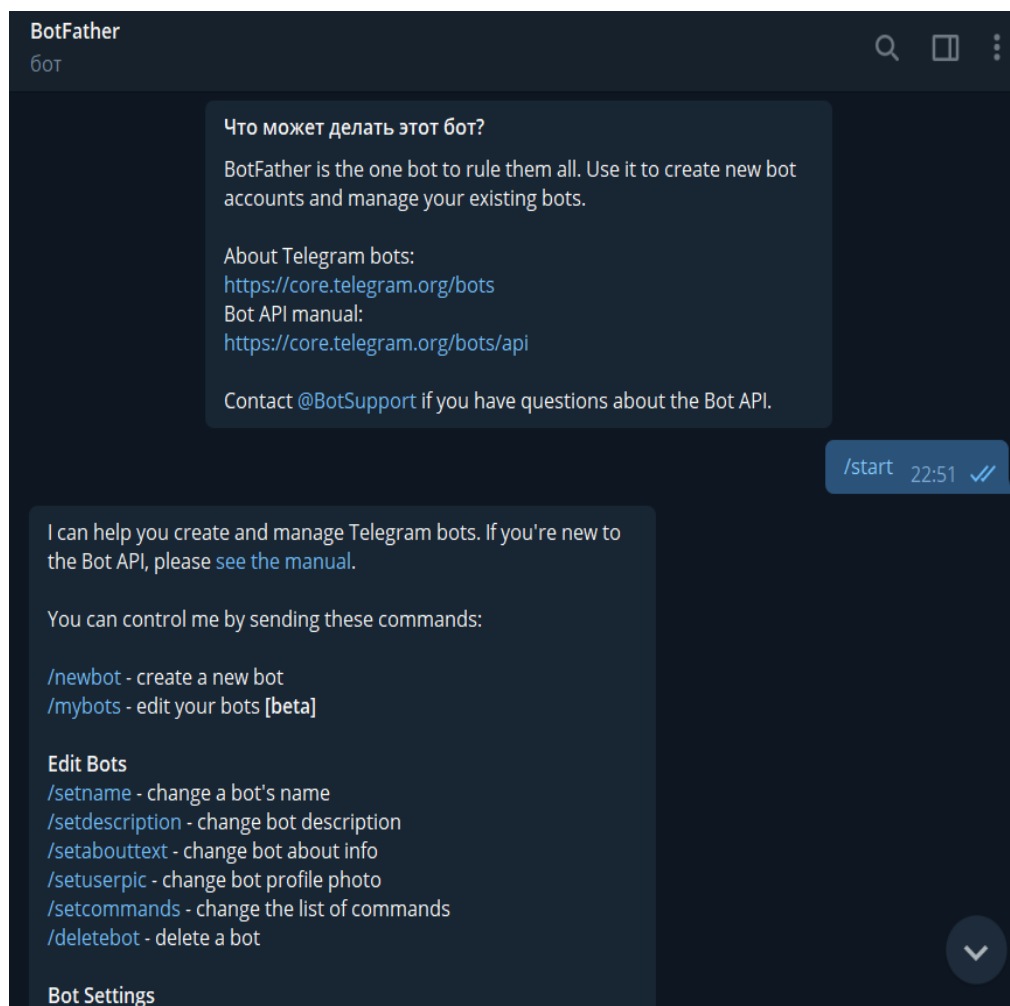


Рисунок 3.1 – Діалогове вікно бота BotFather

Як видно на рисунку 3.1, для створення нового бота потрібно лише запустити бот компдою старт /start та ввести до BotFather команду /newbot. Дати йому певну унікальну назву, за якою користувачі будуть до нього звертатися.

У відповідь BotFather передає посилання на новий бот та унікальний токен API, за допомогою якого буде здійснюватися доступ до бота для його розробки, як можна побачити на рисунку 3.2.

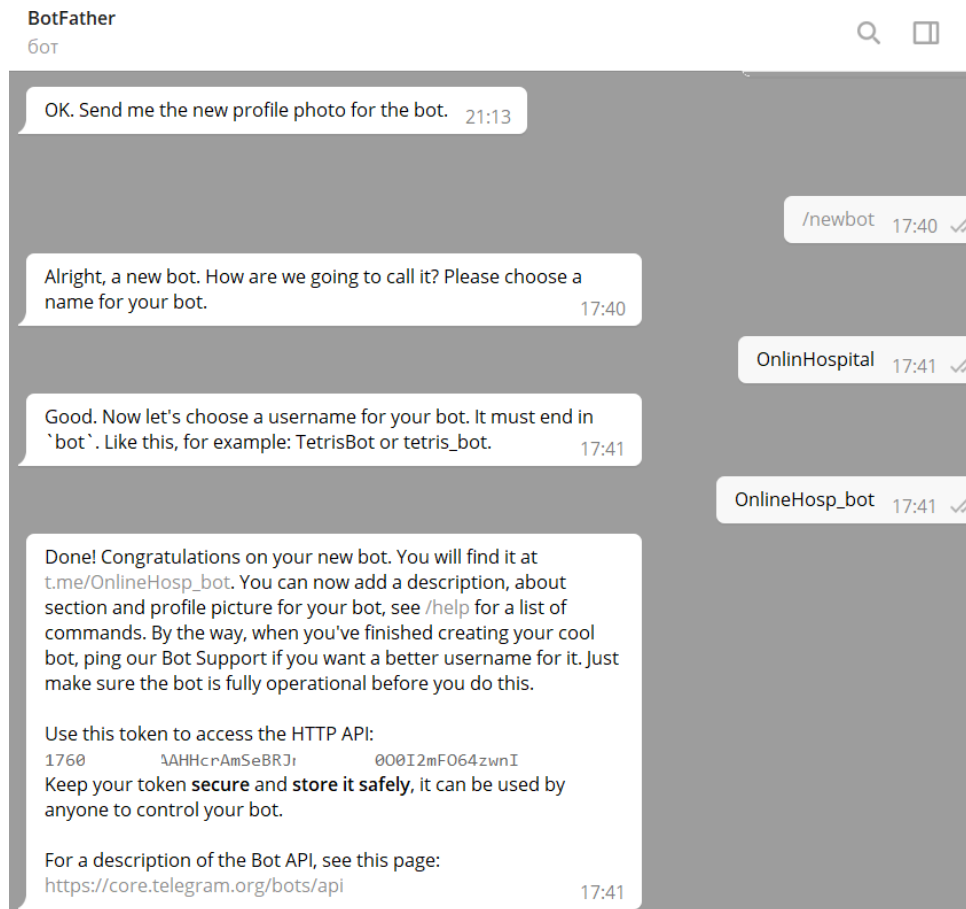


Рисунок 3.2 – Демонстрація створення нового чатбота

Крім того, більшість початкових налаштувань також здійснюються саме через цього бота, наприклад, встановлення аватару – зображення, яке може характеризувати користувача/бота або змінити його опис, змінити назву на нову тощо, як видно зі списку можливих команд на рисунку 3.3.

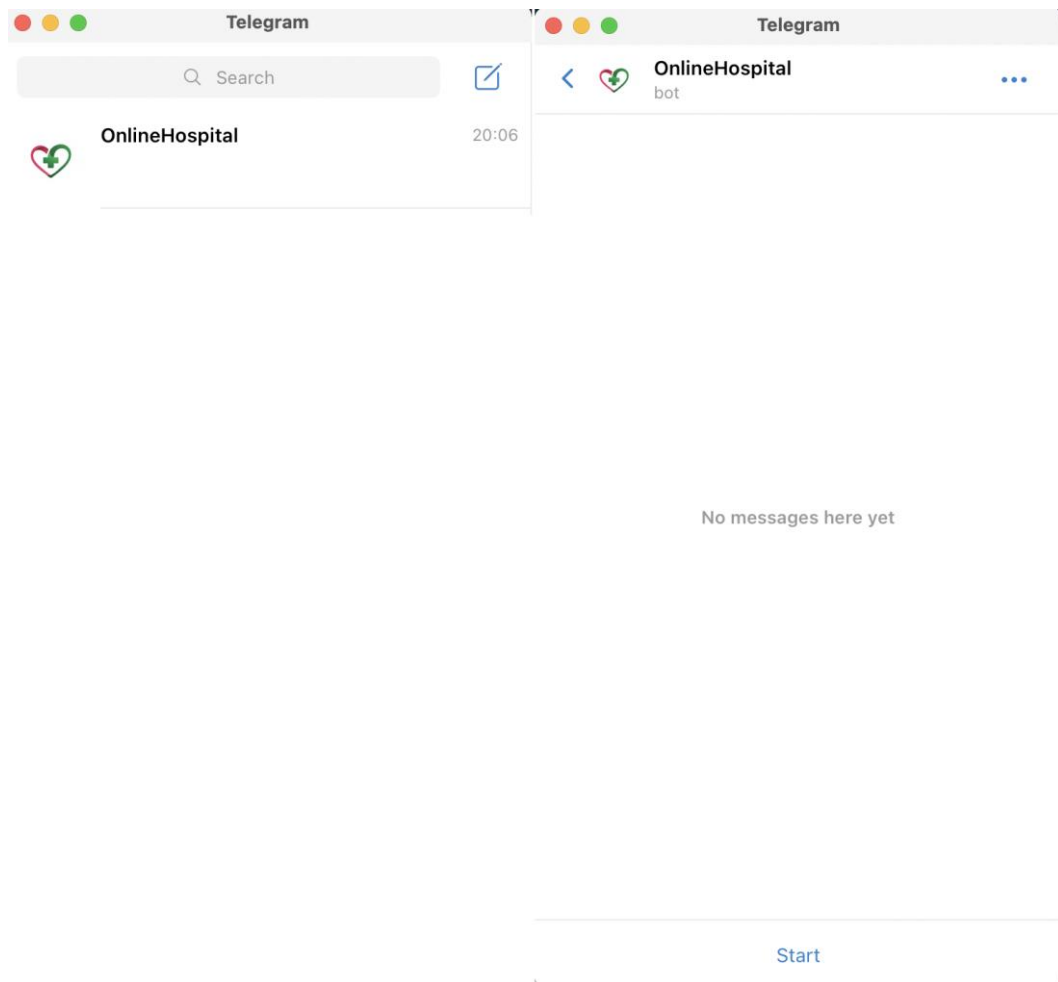


Рисунок 3.2 – Демонстрація створеного чатбота та його стартової сторінки

Маючи діючий токен API, можна почати розробку у IDE.

3.2 Створення проекту у IDE

Після проведеного аналізу та вибору IDE у вигляді PyCharm починається розробка.

Для створення нового проекту потрібно натиснути File – New Project..., у відповідь на це програма видасть вікно, наведене на рисунку 3.4.

В ньому головними функціями є: локація проекту та його назва, в нашому випадку проект було названо в узгодженні до назви чатбота в Телеграм та інтерп-

ретатор, який буде підходити краще до виконання програми. У випадку розробки даної ІС було підключено стандартну на цей час версію мови Python 3.

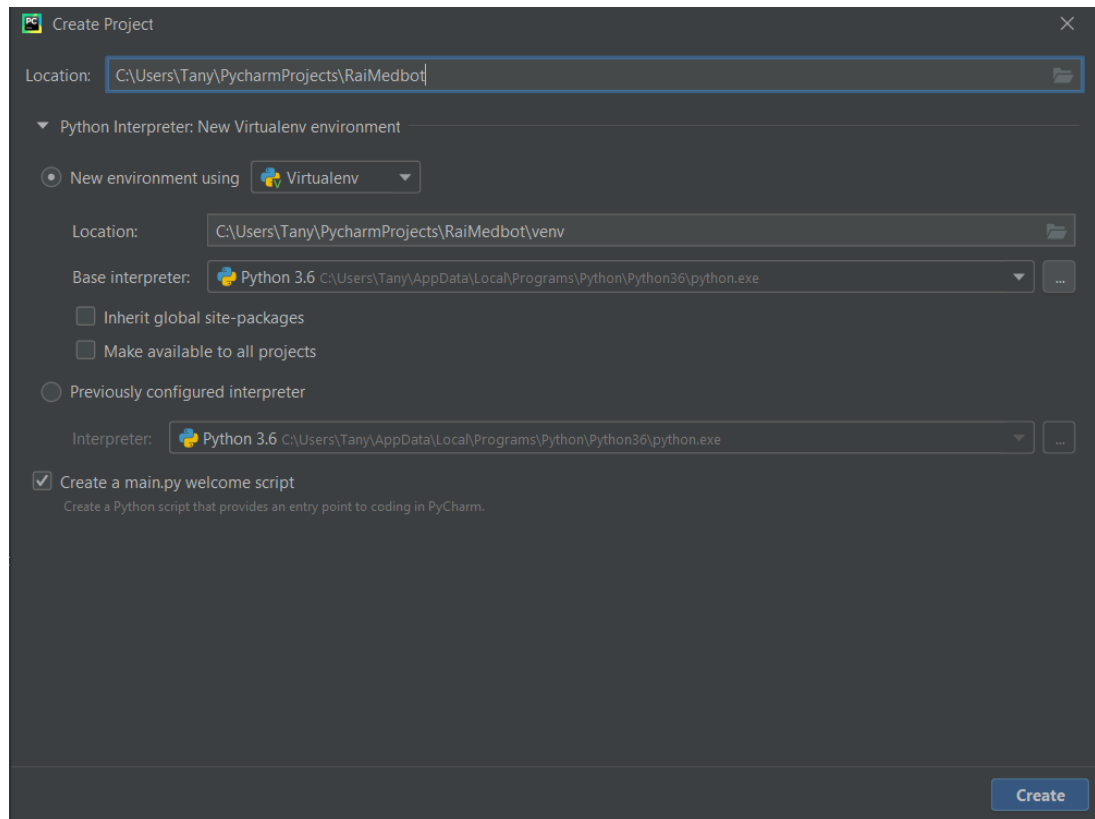


Рисунок 3.4 – Вікно створення нового проекту у PyCharm

Після завершення конфігурації нового проекту IDE створює віртуальне середовище, в якому воно реалізується. У правому вікні можна побачити структуру створеного проекту та стартову сторінку з базовою програмою, яка при виконанні видасть у консоль текст «Hi, PyCharm» для наведення прикладу роботи з кодом, як можна побачити на рисунку 3.5.

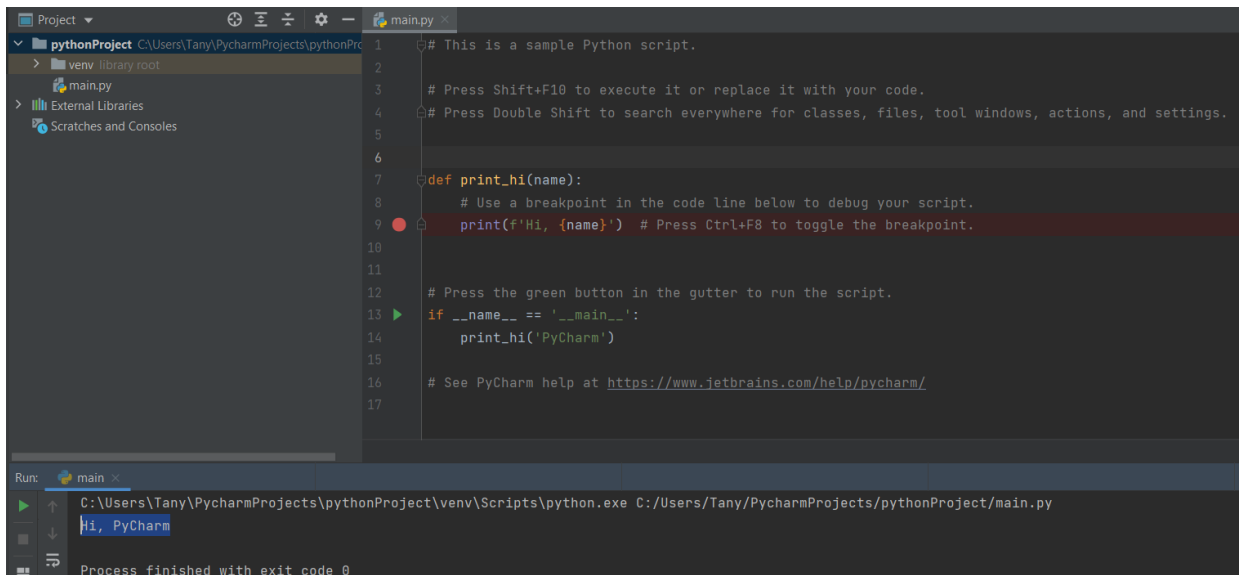


Рисунок 3.5– Стартове вікно нового проекту PyCharm

Переходимо до зміни проекту відносно описаних вище потреб.

3.3 Робота з СКБД

Робота з базою даних може бути виконана як за допомогою коду в програмі основного проекту чи командної строки MySQL-ю, так і за допомогою сторонніх, спеціально призначених для цього додатків, які спрощують методи розробки.

Одним з таких додатків є HeidiSQL, яка надає можливість створювати та редагувати бази даних як за допомогою графічного інтерфейсу, так і за SQL-запитами.

HeidiSQL – це безкоштовний та відкритий клієнт, так званий «фронтенд», для управління базами даних, який підтримує з'єднання та роботу з MySQL.

Щоб управляти базою даних з HeidiSQL, користувач має увійти на локальний або віддалений сервер MySQL з прийнятним паролем, створивши сесію. В рамках цієї сесії користувач може управляти базами даних MySQL на сервері MySQL, і від'єднатися після закінчення роботи. Можливості програми цілком достатні для більшості операцій із загальними та просунутими базами даних, табли-

цями та записами, але розробка залишається у активному стані, щоб забезпечити повну функціональність, котра очікується від фронтенду MySQL. [35].

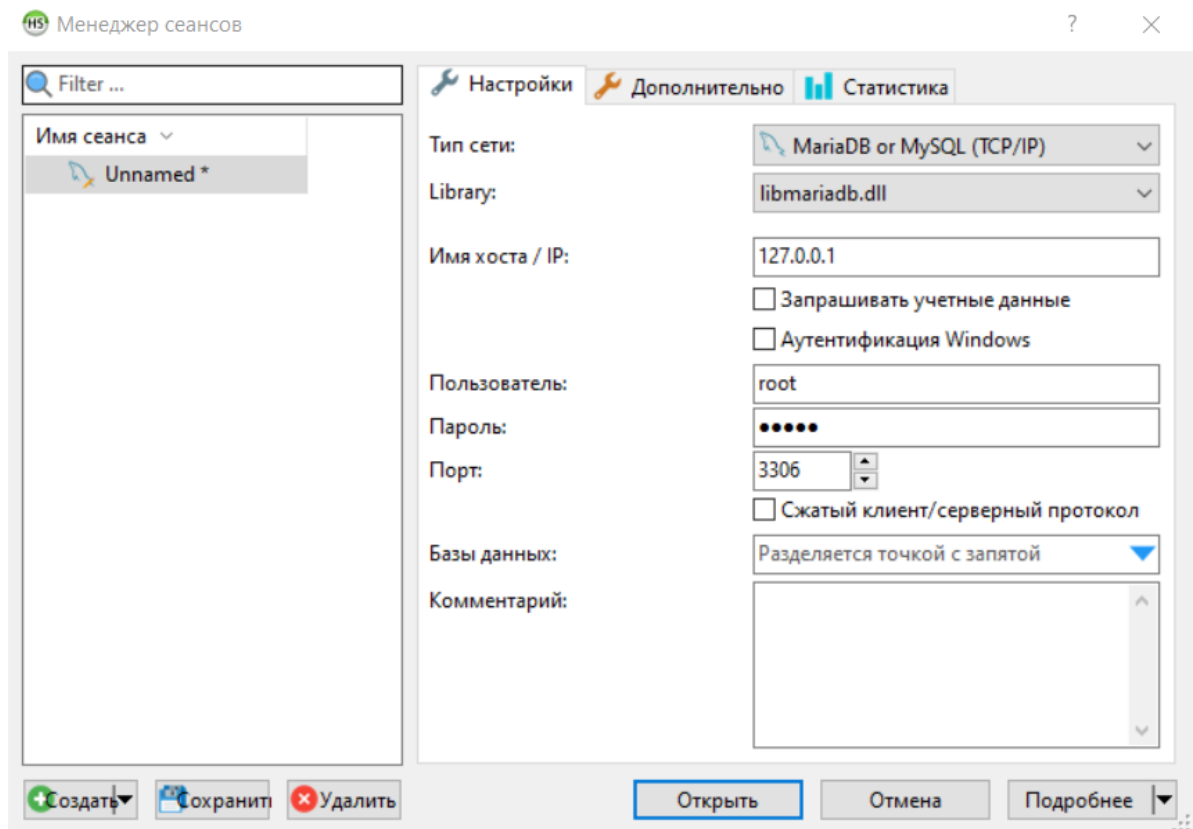


Рисунок 3.6 – Стартова сторінка для доступу до HeidiSQL

Після запуску додатку користувача зустрічає стартова сторінка, зображена на рисунку 3.5, за допомогою якого можна отримати доступ до сервера тільки за умови наявності даних про хост, порт, користувача та пароль, які були визначенні при встановленні MySQL. За допомогою тих самих даних буде здійснений доступ і в програмній частині.

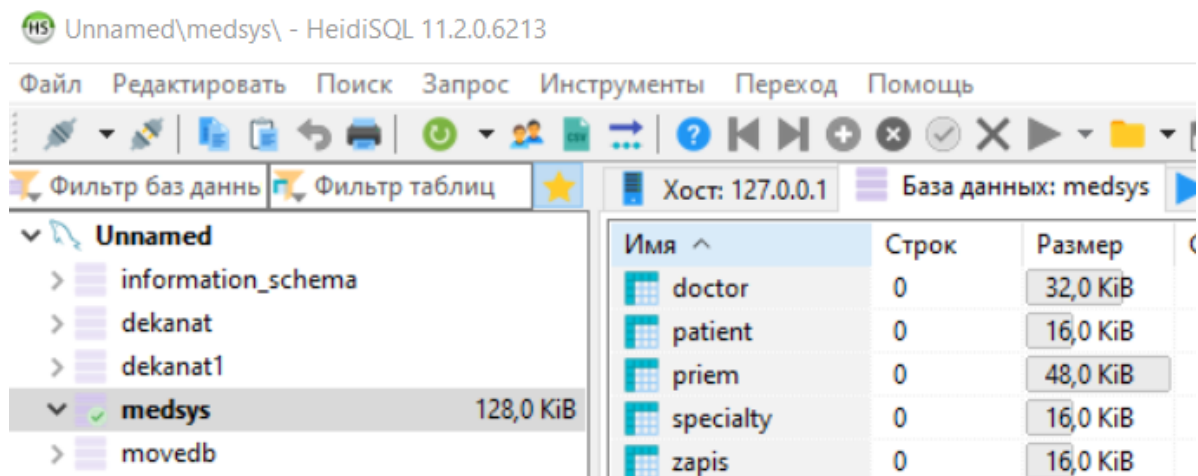


Рисунок 3.7 – База даних medsys

Головна сторінка HeidiSQL містить дані по всім базам даних на відповідному сервері. Для даного проекту створюємо нову базу даних – «medsys», до якої додаємо таблиці та їх атрибути у відповідності до пункту з розробки структури бази даних. Готову, та ще незаповнену, базу даних можна побачити на рисунку 3.6.

При створенні таблиць важливо враховувати те, що перш за все створюються базові таблиці – ті, які мають лише первинний ключ, а вже після таблиці, які їх використовують як зовнішні. Тож в випадку даної системи порядок створення:

- таблиця Specialty – базова таблиця з первинним ключем;
- таблиця Doctor – асоціативна таблиця, використовує зовнішній ключ з таблиці Specialty;
- таблиця Patient; – базова таблиця з первинним ключем;
- таблиця Priem – асоціативна таблиця, використовує зовнішніми ключами з таблиць Doctor та Patient;
- таблиця Zapis– асоціативна таблиця, використовує зовнішніми ключами з таблиць Doctor та Patient.

За вимоги можна створювати таблиці лише за допомогою коду, тоді ці дії буде виконано відповідно до лістингу 3.1.

Лістинг 3.1 – Приклад створення бази та таблиць

```
create_database = "CREATE DATABASE medsys_db"
```

```

CREATE TABLE `Specialty` (
  `spec_id` int(10) NOT NULL AUTO_INCREMENT,
  `name_spec` longtext DEFAULT NULL,
  PRIMARY KEY (`spec_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `Doctor` (
  `d_id` int(10) NOT NULL,
  `d_name` longtext DEFAULT NULL,
  `spec_id` int(10) DEFAULT NULL,
  `d_phone` varchar(250) DEFAULT NULL,
  `work_time` longtext DEFAULT NULL,
  PRIMARY KEY (`d_id`)
  FOREIGN KEY (`spec_id`) REFERENCES Specialty (spec_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

3.5.1 Підключення бібліотек та бази даних

Перше, що треба зробити у програмі: підключити вже встановлену бібліотеку `pyTelegramBotAPI` за допомогою імпортування за ім'ям `telebot` та дати доступ до токена API чатбота (в даному випадку токен було винесено в окремий файл `constants`), аби програма мала доступ до нього та можливість редагувати його функціонал.

Як наведено в лістингу 3.2, звертання до боту реалізоване через змінну `bot`, що містить звертання до бібліотеки телеграма та токена самого чатбота, який було скопійовано з повідомлення головного чатбота – `BotFather` – при створенні.

Лістинг 3.2 – Підключення бібліотек та токена

```

import telebot
from telebot import TeleBot
import constants
import mysql.connector

bot: TeleBot = telebot.TeleBot(constants.token)

```

Крім того, дуже важливо одразу вписати до кінця коду функцію `bot.polling`, адже програма має виконуватись постійно, перевіряючи, чи не надходять до чат-

бота нові запити, а без цієї функції алгоритм буде виконано один раз та одразу ж завершено. Функцію наведено у лістингу 3.3

Лістинг 3.3 – Забезпечення безперервної роботи чатбота

```
if name == "main":
    bot.polling(none_stop=True, interval=0)
```

Слід зауважити, що усі звернення до бази даних робляться всередині функції обробки виключень try: except з ціллю запобігання аварійного завершення програми у випадках виникнення можливих помилок та збоїв. В лістингу 3.4 можна побачити як реалізується ця функція. Крім того, треба звернути увагу на те, що в даному випадку у ній відбувається звернення до бази даних, з якою має працювати система – «database="medsys_db"». Як і було вказано у пункті 3.3 зі створенням бази за допомогою додатку HeidiSQL, програма також потребує підтвердження звернення до бази через введення порту, паролю тощо, в цілях підтримки безпеки доступу до даних у випадку. Слід зауважити, що згідно пункту про безпеку даний фрагмент має бути в окремому файлі, як і токен розроблюваного чатбота.

Лістинг 3.4 – Підключення бази даних

```
try:
    db = mysql.connector.connect(
        host="127.0.0.1",
        user="root ",
        passwd="teygu",
        port="3306",
        database="medsys"
    )
    print(db)
    cursor = db.cursor()
except Exception as e:
    print(e)
    print("Помилка підключення")
```

3.5.2 Функціональна частина розробки чатбота

Усі телеграм-чатботи мають одну загальну команду, яка використовується для їх запуску, тому після визначення бібліотек та роботи з базою даних потрібно прописати строку з лістингу 3.5.

Лістинг 3.5 – Обробка команди «Старт»

```
@bot.message_handler(commands=["start"])
```

За допомогою `message_handler` (оброблювача повідомлень) чатбот реагує на команду «start».

Лістинг 3.6 – простий приклад створення привітального повідомлення. Функція `def handle_command(message):` відповідає вітанням користувача після його запуску вищенаведеною командою «старт».

За допомогою `bot.send_message` здійснюється вивід тексту, який має відправити чатбот та, щоб бот завжди знав кому конкретно він має вислати нове повідомлення, функція `message` містить в собі параметр `message.chat.id`, який передає унікальний ідентифікатор користувача в телеграмі.

Лістинг 3.6 – Текстова відповідь бота на команду «старт»

```
def handle_command(message):
    bot.send_message(message.chat.id, 'Добро пожаловать! \n'
                                     'Сейчас я расскажу как
нам начать работу. ')
```

Одразу після вітання чатбот має перевірити реєстрацію у системі за допомогою пошуку у базі за унікальним ідентифікатором користувача у телеграм. У таблицях з даними про лікарів та пацієнтів вони виступають в якості первинних ключів в полях `d_id` та `p_id` відповідно.

Через те, що прямого зв'язку між таблицями лікарів та пацієнтів немає, об'єднати та одночасно проводити перевірку в обох таблицях неможливо, тому

треба організувати пошук послідовно. Спочатку в таблиці лікарів, що містить значно менше даних, слідом, якщо збігів не буде знайдено, пошук переходить до таблиці пацієнтів, відповідно. Як виглядає пошуковий запит на мові SQL можна побачити в лістингу 3.7 на прикладі пошуку лікарів, пошук в інших таблицях виконується аналогічно.

Лістинг 3.7 – Пошук ідентифікатора в таблиці Doctor

```
cursor.execute(
    "SELECT *
    FROM Doctor
    WHERE d_id f'{message.from_user.id}'")
doctor = cursor.fetchall()
```

Якщо користувача не було знайдено ні в одній з таблиць бази, чатбот надсилає користувачу повідомлення з запитом до реєстрації та водночас з цим меню, у якому він має обрати свою роль: лікар чи пацієнт, за допомогою якого буде визначено алгоритм роботи для наступний дій.

Дане меню було реалізовано за допомогою кнопочового меню під полем введення даних користувача – функція `user_markup`. Приклад реалізації наведено у лістингу 3.8. Для зручності роботи користувача одночасно з появою кнопок було додано текст з поясненням до дій.

Лістинг 3.8 – Реалізація функціоналу кнопок у чатботі під час реєстрації

```
user_markup = telebot.types.ReplyKeyboardMarkup(True, False)
user_markup.row("ПАЦІЄНТ", "ВРАЧ")
bot.send_message(message.chat.id, "Я еще не вижу вас в моей
базе \n
Пожалуйста, пройдите простую регистрацию. \n
Нажми на панель ниже и выбери нужный пункт. ",
reply_markup=user_markup)
```

У відповідь користувачу надсилається функція введення ПІБ, яка одразу ж заносить дані користувача в таблицю. Послідовно те ж саме відносно адреси проживання та дати народження користувача. Приклад реалізації даного запиту наведено у лістингу 3.9. Можна також звернути увагу, що в цьому лістингу можна ро-

зглянути застосування параметризації, що біло наведено яка метод захисту від SQL-ін'єкцій.

Лістинг 3.10 – Оновлення даних таблиці з занесенням ПІБ

```
try:
    sql = UPDATE TABLE Pacient
    SET p_name='%s'
    WHERE p_id= f'{message.from_user.id}'
    cursor.execute(sql)
    db.commit()
    print(cursor.rowcount, "Запись добавена")
except Exception as e:
    print(e)
    print("Пользователь уже существует")
```

Слід також зазначити, що номер телефону до телеграм дуже зручно відправляти через функцію доступу до контактів, яка мінімізує потреби до зусиль користувача. Приклад наведено у лістингу 3.11.

Лістинг 3.11 – Автоматизований вибір номеру-контакту користувача

```
@bot.message_handler(commands=["getphone"])
def getphone(message):
    keyboard = types.ReplyKeyboardMarkup(row_width=1,
    resize_keyboard=True)
    button_phone = types.KeyboardButton(text="Отправить номер
    телефона", request_contact=True)
    keyboard.add(button_phone)
    bot.send_message(message.chat.id, "Отправь мне свой номер
    телефона!", reply_markup=keyboard)
```

В разі успішної реєстрації, користувачу надається можливість знайти лікаря за його прізвищем, або спеціалізацією. Так як в даному випадку таблиці з інформацією про лікарів та переліком спеціалізацій пов'язані зовнішнім ключем `спес_id`, пошук, на відміну від випадку з пошуком лікарів та пацієнтів, можна виконати одним запитом за допомогою використання оператора «INNER JOIN» у сумісності з «SELECT», що поєднує між собою таблиці під час пошуку.

Крім того, в даному випадку використовується також оператор «LIKE» у поєднанні з «WHERE», адже в звичайному випадку «WHERE» повертає значення, що цілком співпадають зі словами з пошукового запиту, тоді як оператор «LIKE»

надає можливість відбирати усі записи, у яких міститься одне з шуканих слів. Тобто, наприклад, аби знайти лікаря, так як в розробленій базі немає розділення стовпців ПІБ на окремі, клієнту в випадку використання лише «WHERE» довелося б вводити повні ПІБ, тоді як з «LIKE» можна шукати за будь-якою частиною повного імені.

Лістинг 3.12 – Пошук лікаря за прізвищем або спеціалізацією

```
@bot.message_handler(content_types=["text"])
def handle_command(message):
    if message.text == "НАЙТИ ВРАЧА":
def check_user (message):
cursor.execute(
    "SELECT d_name, name_spec
    FROM Doctor INNER JOIN Specialty
    ON Doctor.spec_id = Specialty.spec_id
    WHERE name_spec LIKE '%s' OR p_name '%s'"
)
doctor = cursor.fetchall()
if doctor:
    for doctor in result:
        print(doctor)
else:
    doctor = 0
    bot.register_next_step_handler(sent, 'Что-то введено не
так! Проверьте, все ли написано верно.', check_user)
```

В лістингу 3.12 важливим фрагментом є останній рядок, в якому використовується функція `bot.register_next_step_handler(sent, check_user)`, яка змушує бота не покидати користувача після виконання однієї команди, а залишитися в режимі очікування на відповідь.

В даному випадку відповідь користувача – повні ПІБ бажаного лікаря, завдяки чому йому надається уся інформація про нього: чатбот має вивести з бази у повідомлення ПІБ лікаря, його спеціалізацію, контактний телефон та розклад роботи. Реалізацію цієї частини програми можна побачити в лістингу 3.13.

Лістинг 3.13

```
cursor.execute(
    "SELECT
    d_name, name_spec, d_phone, work_time
    FROM Doctor INNER JOIN Specialty
    ON Doctor.spec_id = Specialty.spec_id
```

```

WHERE d_name LIKE '%s')
val = (message.chat.id, d_name, name_spec, d_phone,
work_time)
        cursor.execute(sql, val)
        db.commit()
        doctor = cursor.fetchall()

```

В результаті здійснення пошуку лікаря можна передати отримані, відповідні до результатів, ПІБ до меню під клавіатурою, аби клієнти мав змогу не вводи їх вручну, з ціллю підвищення рівня автоматизації та прискорення роботи з телеграм-чатобом, як наведено в лістингу 3.14.

Лістинг 3.14 – Кнопкове меню з кнопками по вибірці з БД.

```

doctor = cursor.fetchall()
if doctor == True
    for x in result
        @bot.message_handler(content_types=['text'])
        bot.edit_message_text(message.chat.id, text=x, "%s"
        reply_markup=makeKeyboard)
    else doctor = 0
    bot.register_next_step_handler(sent, 'Что-то введено не
так! Проверьте, все ли написано верно.', check_user)

```

Крім того, потрібно реалізувати кнопки за допомогою яких можна повернутися назад у по ходу алгоритму виконання програми, наприклад, після запису до лікаря перейти до пошуку іншого тощо. Реалізацію наведено у лістингу 3.15.

Лістинг 3.15 – Кнопкове меню з поверненням назад.

```

user_markup.row("НАЗАД")
@bot.message_handler(commands=[НАЗАД])
def handle_start(message):
    if menu_plumbing:
        menu_plumbing = False

        user_markup =
        telebot.types.ReplyKeyboardMarkup(
        resize_keyboard=True)
        user_markup.row('Найти врача')

bot.send_message(message.from_user.id, 'Вы вернулись назад!
Выберите интересующий пункт меню', reply_markup=user_markup,
check_user)

```

Також в системі мають бути реалізовані:

Меню «Мої записи», в якому пацієнту буде надано список його записів на прийом до лікарів. Створення такого запиту проходить у аналогії до лістингу 3.н, пошуку лікарів за спеціалізацією та ім'ям, адже в даному випадку також потребується об'єднати таблиці по зовнішнім ключам.

Гілка алгоритму роботи зі сторони лікаря, реалізація якої виконується аналогічно до пацієнта, адже в сутності своїй потребує тих самих функцій, але в трохи іншому порядку та з іншими аргументами. Наприклад, пошук пацієнтів, що виконується запитом по унікальному ідентифікатору користувача в одній таблиці подібно лістингу 3.7, реєстрація та введення даних про прийом згідно лістингам 3.10 та 3.11, сортування записів про прийоми та на прийоми згідно лістингу 3.12.

Відповідно до усього вищезначеного, було цілком розроблено інформаційну систему автоматизованого обліку районної амбулаторії з використанням месенджера телеграм.

3.6 Випробування та демонстрація роботи системи

Завершивши усі етапи проектування та розробки, система потребує випробування та тестування перед відкриттям до використання з ціллю перевірки працездатності.

Емпіричний метод оцінки параметрів системи та її складових частин є невід'ємною частиною процесу розробки, тож головною метою даного етапу є випробування різних шляхів алгоритму праці чатбота, аби переконатися у тому, що все працює згідно до вимог, що були поставлені до інформаційної системи районної амбулаторії.

Перш за все, проведено запуск бота, який, як можна побачити на рисунку 3.8, виправно надає до відповіді вітання та одразу перевіряє, чи є користувач вже зареєстрованим у базі.

Відповідно до кнопкового меню користувач вибирає свою роль у системі та успішно реєструється.

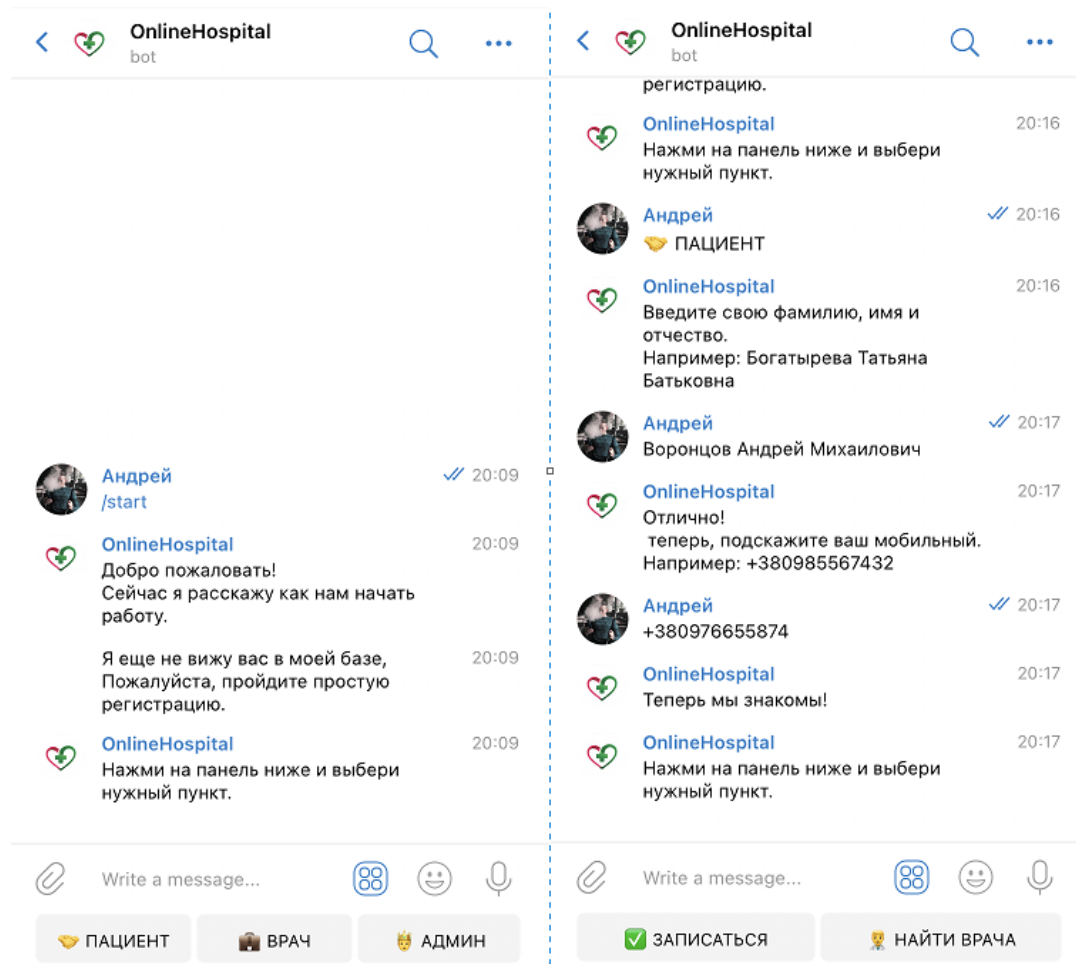


Рисунок 3.8 – Запуск чатбота та реєстрація у системі

Усі повідомлення чатбот справно надсилає у потрібний момент та ніде не зупиняється, виконуючи всі дії згідно прописаних команд та дані успішно були записані до бази даних.

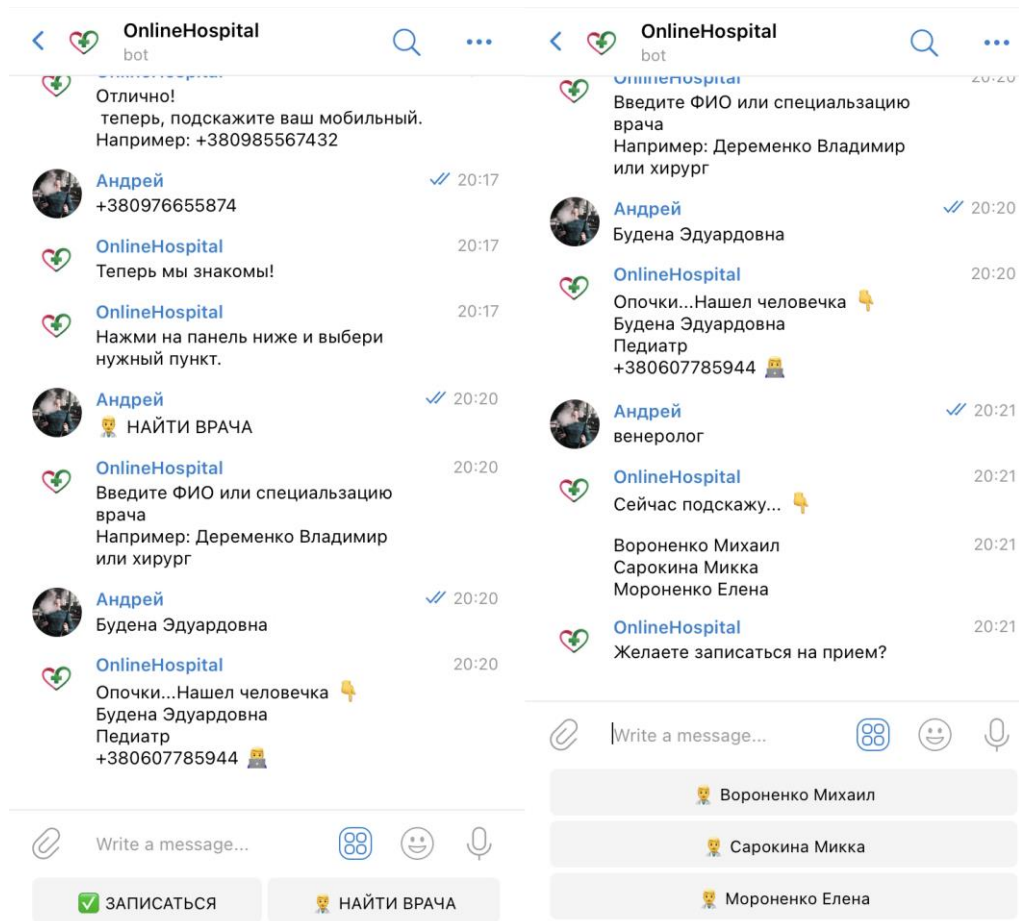


Рисунок 3.9 – Реалізація пошуку лікаря у системі

Демонстрацію цієї частини випробування надано на рисунку 3.10.

Якщо натиснути на кнопку «Записатися», чатбот передає користувачеві зручне меню вибору дня тижня, який треба обрати згідно до розкладу лікаря та введення часу потім. Чатбот успішно відправив дані до бази, адже він одразу має відправити у відповідь ті самі дані, що ввів до запису користувач та зроблено перевірку: пацієнт тисне на кнопку «Мои записи» та отримує вірну відповідь з відсортованими з бази даних значеннями, які були введені користувачем.

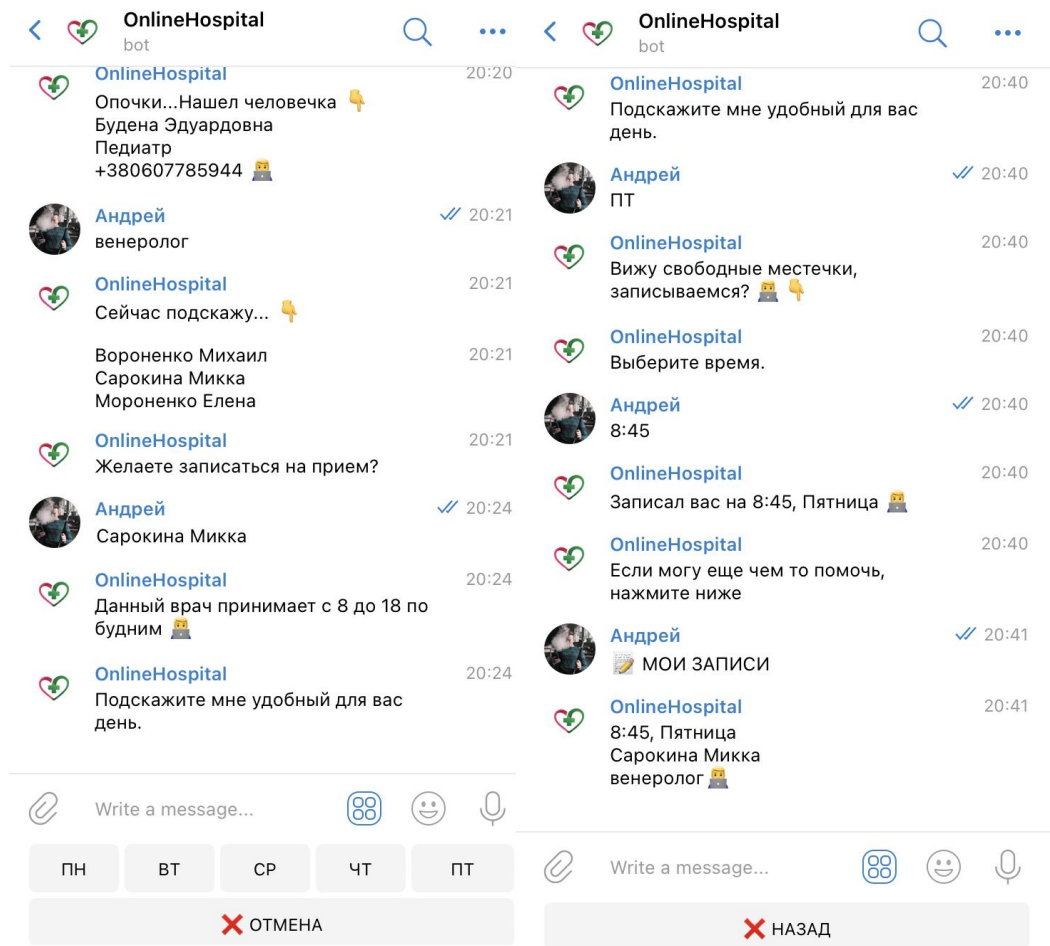


Рисунок 3.10 – Реалізація записів до лікаря

Такі ж самі дії задля перевірки працездатності системи були проведені й для гілки з роботою лікарів, пройдені були усі ситуації як з успішними наслідками, так і з виникненням помилкових дій зі сторони користувача.

Тож можна дозволити зробити висновок, що система працює справно, за дотриманням усіх вищезазначених до неї вимог, за виключенням деяких незначних помилок у текстових відповідях чатабота.

Підводячи підсумки до реалізації інформаційної системи районної амбулаторії, було створено самого чатбота у Телеграм в якості пустого шаблону, який наповнюється функціоналом за допомогою його програмування з доступом через відповідний API-токен. Використано візуальний інтерфейс для роботи з базою даних MySQL – HeidiSQL, де створено відповідні таблиці, що було визначено на етапі проектування системи.

У інтегрованому середовищі розробки PyCharm створено проект, підключено усі відповідні бібліотеки для роботи з телеграм-ботом та СКБД MySQL, виконане безпосереднє програмування телеграм-чатбота та виконано перевірку його працездатності, яку він успішно пройшов з незначними недоліками.

Тож по завершенню розробки було отримано повністю та справно функціонуючий чатбот, готовий до роботи.

ВИСНОВКИ

У роботі докладно проаналізовано об'єкт розробки для чіткого визначення вимог до функціоналу системи, актуальності такої системи, визначено проблематику, що призвела до ідеї розробки даного проекту, виконано огляд вже існуючих рішень.

Описані конкретні вимоги щодо реалізації інформаційної системи обліку районної амбулаторії зі сторони користувачів та розробників.

Задля забезпечення максимальної кросплатформенності, з огляду на потреби користувачів щодо не використання зайвої пам'яті на їх пристроях, поєднання функціоналу та простоти, мінімізації витрат на розробку та постійний максимальний доступ до системи, обрано реалізацію у вигляді чатбота у месенджері Телеграм.

На підставі визначеного об'єкту розробки у вигляді телеграм-чатботу проведено аналіз, згідно якого порівняно популярність, функціонал та простоту використання кожної. Згідно до цього обрано мову програмування – Python.

Після порівняння особливостей трьох з популярних IDE, що можуть працювати з мовою Python, обрано в результаті PyCharm, перевагою якого стала краща швидкість роботи з вибраною мовою та налагодженість роботи з Python, адже середовище розробки спеціально було створено для цієї мови програмування

Проаналізовані рішення щодо створення серверної частини, адже інформація, що буде поступати до чатбота потребує місця для збереження. З оглядом на усі потреби та вимоги, серед найпопулярніших СКБД обрано MySQL.

На основі всього вищезазначеного побудовано ER-модель бази даних, що включає в себе всі таблиці, перелік полів та даних в них, які буде містити в собі БД та зв'язки між ними, що завершує етап проектування системи.

Переходячи до розробки, в першу чергу за допомогою спеціального чатбота @BotFather було створено чатбот інформаційної системи районної амбулаторії, в

якості шаблону користувацького інтерфейсу, який буде заповнюватись функціоналом за ходом розробки.

Для зберігання та керування даними було створено базу даних MySQL. Для зручності у розробці було використано «фронтенд» додаток, який реалізує СКБД у графічному інтерфейсі.

На базі описаних вище параметрів було створено п'ять таблиць, з визначенням їх призначення, полів первинних та зовнішніх ключів, що поєднують таблиці між собою.

Створено проект у середовищі інтегрованої розробки для початку роботи. Вибрано та підключено необхідні бібліотеки для роботи з визначеними функціями допоміжними у розробці чатботів телеграм та обраною СКБД – MySQL.

Розроблено функціонал системи згідно до алгоритму, реалізованого під час проектування.

Останнім етапом стало емпіричне випробування системи, у ході якого виявлені та виправлені незначні недоліки реалізованого чатбота. За проходженням тестування, інформаційна система визнана працездатною.

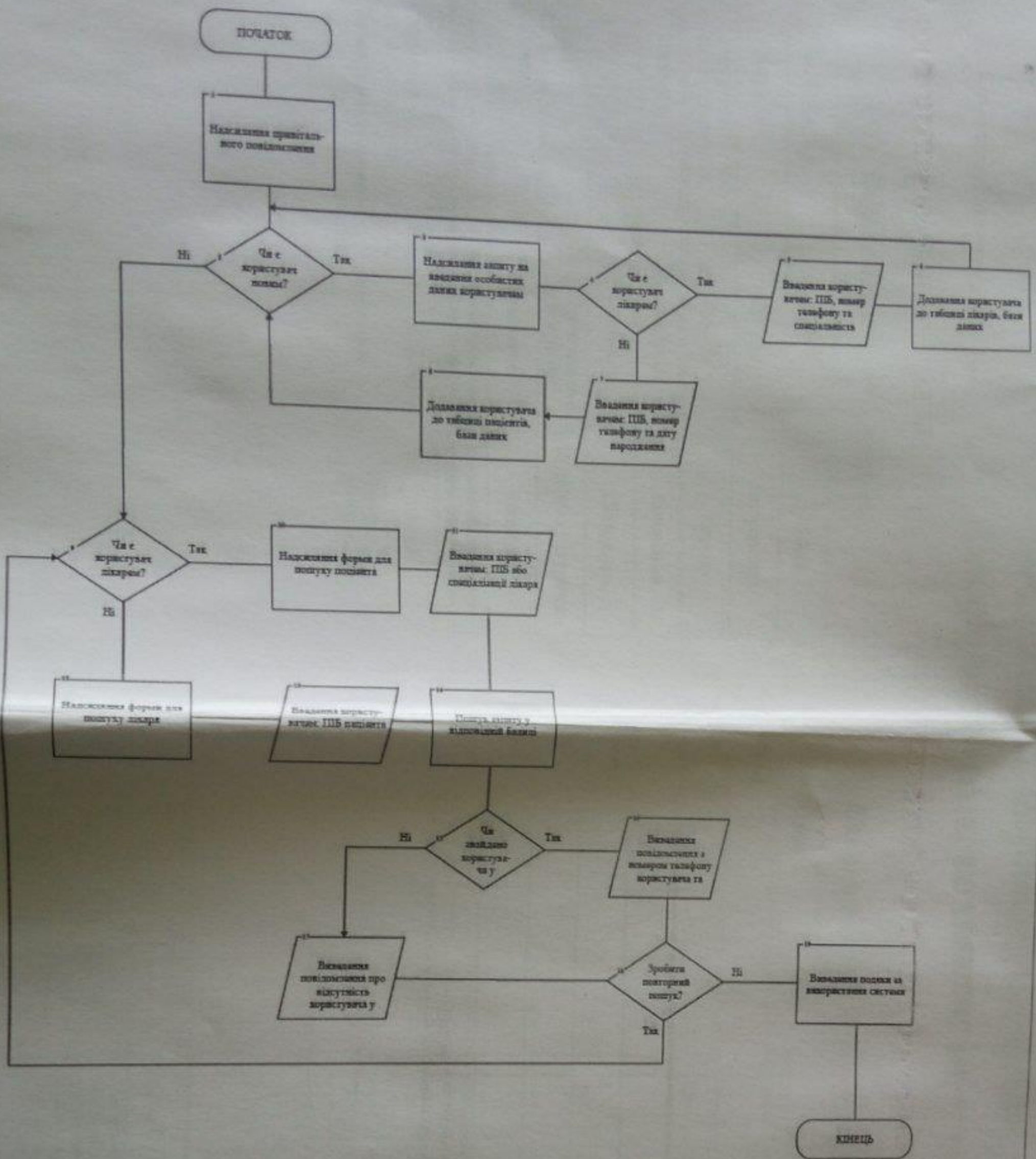
Таким чином, в результаті виконання роботи отримано працездатний, згідно до визначених умов, чатбот, який являє собою інтерфейс інформаційної системи для користувачів, що полегшує способи зв'язку та пошуку лікарів та пацієнтів, надає можливість дистанційного з'єднання, значно скорочує витрати часу на запис на прийом як зі сторони пацієнтів, так і працівників реєстратури. Надає можливість лікарям оперативно передивлятись записи на прийом, не шукати звичайні амбулаторні картки заради якоїсь простої та невідкладної інформації про пацієнта, коли у базі даних амбулаторії зберігається уся контактна інформація та короткі записи про прийоми в лікарів. Запропоновано засоби захисту інформації в системі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

- 1 Використання відеохостингу youtube для підтримки навчання медичних працівників URL: https://informatika.udpu.edu.ua/?page_id=3499
- 2 О. П. Мінцер., Інформатизація охорони здоров'я: проблеми, розв'язані та нерозв'язані. Питання впорядкованості та сингулярності. // 2013 – 11 с.
- 3 Карпова І.П., Базис даних. Навчальний посібник. // М., 2009 – 131 с
- 4 Chatbot URL: <https://www.techopedia.com/definition/16366/chatbot>
- 5 Які мобільні додатки є найбільш популярними? URL: http://kiis.com.ua/?lang=ukr&cat=reports&id=1027&page=1&fbclid=IwAR2mGKEh29wX6YTKrz2lqOQe-1jDXQ-oB9ehFdn4VUmFQjCTr_qDTSV9QwQ
- 6 Чем отличаются чат-боты в Telegram, WhatsApp, Viber, Facebook, VK URL: <https://www.rpv-bot.ru/chem-otlichaetsya-chat-bot-v-telegram-whatsapp-vk-viber-facebook>
- 7 Эталон мессенджера URL: <https://keddr.com/2018/03/etalon-messendzhera/>
- 8 Мова програмування URL: http://wikifizmat.udpu.edu.ua/index.php?title=Мова_програмування
- 9 Язык программирования URL: <https://habr.com/ru/post/231897/>
- 10 Виды языков программирования URL: <http://csaa.ru/vidy-jazykov-programmirovaniya/>
- 11 Языки программирования. Компиляторы и интерпретаторы URL: <https://tux.org.ua/yazy-ki-programmirovaniya-kompilyatory-i-interpretatory/>
- 12 TIOBE Index for May 2021 URL: <https://www.tiobe.com/tiobe-index/>
- 13 Top 7 Programming Languages to Develop an AI-Powered Chatbot URL: <https://chatbotslife.com/top-7-programming-languages-to-develop-an-ai-powered-chatbot-f253e728845c#:~:text=effect%2Dbased%20looping,-,Python,of%20Chatbots%20as%20of%20now.>

- 14 Создание программного обеспечения электронного учебника URL:
<https://smekni.com/a/114766-5/sozdanie-programmnogo-obespecheniya-elektronnogo-uchebnika-5/>
- 15 Язык программирования Python URL: <https://web-creator.ru/technologies/webdev/python>
- 16 Python против C ++ URL: <https://ru.photo-555.com/9254523-python-vs-c>
- 17 Python URL: <https://ru.wikipedia.org/wiki/Python>
- 18 Python или C ++ URL: <https://pythonru.com/baza-znaniy/python-ili-c>
- 19 10 лучших IDE и редакторов кода для веб-разработчиков URL:
<https://www.reg.ru/blog/10-luchshih-ide-i-redaktorov-koda-dlya-veb-razrabotchikov/>
- 20 Среда разработки URL: <https://sites.google.com/site/erina-program/theory/01-programming-languages/1-2-development-environment>
- 21 Код Visual Studio URL: <https://azure.microsoft.com/ru-ru/products/visual-studio-code/>
- 22 Коннолли Т., Бегг К. Базы данных: проектирование, реализация, сопровождение. Теория и практика, 3-е изд. : Пер. с англ. : Уч. пос. // М.: Изд. дом "Вильямс", 2003. – 1440 с.
- 23 Сравнение современных СКБД URL: <http://drach.pro/blog/hi-tech/item/145-db-comparison>
- 24 Что такое PostgreSQL? URL: <https://oracle-patches.com/common/3214-%D1%87%D1%82%D0%BE-%D1%82%D0%B0%D0%BA%D0%BE%D0%B5-postgresql>
- 25 MySQL – СКБД URL: <https://web-creator.ru/articles/mysql>
- 26 Обзор инструментов для работы с MySQL URL:
webmasters.by/articles/review-po/96-mysql-development-tools.html
- 27 Сравнение MySQL с другими СКБД URL:
<https://program.rin.ru/razdel/html/490-2.html>
- 28 Защита информации URL: <https://cutt.ly/4nkbEO0>

- 29 Методы и способы защиты информации URL: <https://www.anti-malware.ru/practice/methods>
- 30 Sqlamp: SQL-инъекции – это просто URL: <https://xakep.ru/2011/12/06/57950/#toc01>
- 31 Give me parameterized SQL, or give me death URL: <https://xakep.ru/2011/12/06/57950/#toc01>.
- 32 Чатботы для клиник и санаториев URL: <https://www.chatcompose.com/ru/clinic.html>
- 33 Что такое библиотеки в программировании? URL: <https://almamat.com/ru/blog/posts/35-chto-takoe-biblioteki-v-programmirovanii>
- 34 Пишем ботов для Telegram на языке Python URL: https://mastergroosha.github.io/telegram-tutorial/docs/lesson_01/
- 35 HeidiSQL URL: <https://uk.wikipedia.org/wiki/HeidiSQL>



					13.02070849.00003 A1		
Зам.	Лист	№ Акту	Підпис	Дата	Розробка системи автоматизованого обліку в районній амбулаторії з використанням месенджерів		
Розроб.	Богатирьова Т.О.				Блок-схема алгоритму роботи системи		
Перевір.	Зеленцова І.Я.				Лист 1	Листів 1	
Листов.					НУ "Засторожко пацієнтська" КНТ-317		
Н.м.пер.	Шарбак Н.В.						
Затвердж.							

Порівняння C++ та Python

Параметр	C++	Python
Компіляція	Компіляція	Інтерпретація
Простота у використанні	Непросто	Легко
Типізація	Статична	Динамічна
Збір сміття	Не підтримує	Підтримує
Типи	Перевірка під час компіляції	Прив'язані до значень
Швидке прототипування	Неможливе	Можливе
Функції	Обмеження по типу параметрів	Не має обмежень по типу параметрів
Ефективність	Важко підтримувати	Легко
Швидкість виконання	Швидкий	Незначно, але повільніший

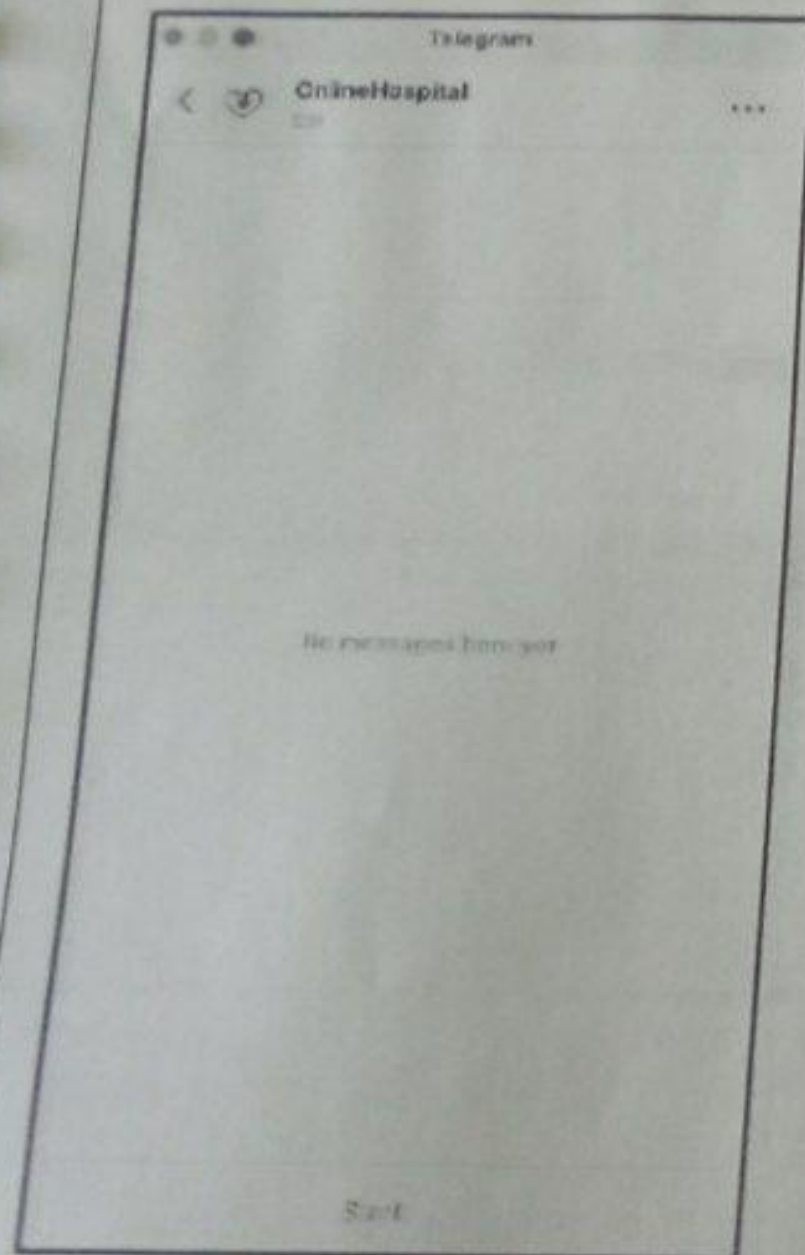
Наявність лідерів-месенджерів на платформах

Платформи	Viber	Facebook	Telegram
Android	+	+	+
IOS	+	+	+
Wp	+	-	+
Windows	+	-	+
Mac	+	-	+
Linux	+	-	+
Web	-	+	+

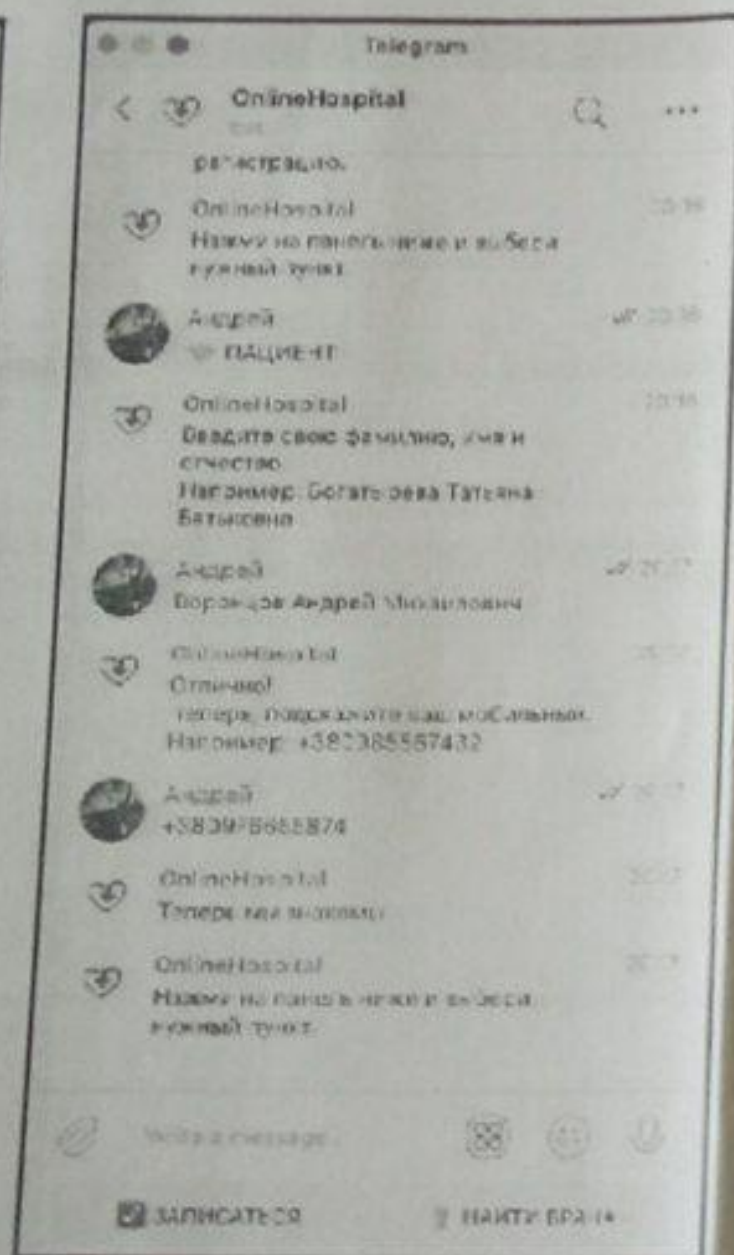
					13.02070849.000003 ПЛІ		
					Розробка системи автоматизованого обліку в районній амбулаторії з використанням месенджерів		
					Порівняльна характеристика використаних технологій		
Зам.	Лист	№ докум.	Підпис	Дата	Лист	Маса	Мощност.
Розроб.		Богданчук Т.О.					
Перевір.		Земельна Л.В.					
Технік							
Начальн.		Щербак Н.В.					
Затверд.							
					Лист 1		
					Листів 1		
					НУ "Запорізька політехніка"		
					КНТ-517		

Демонстрація роботи системи

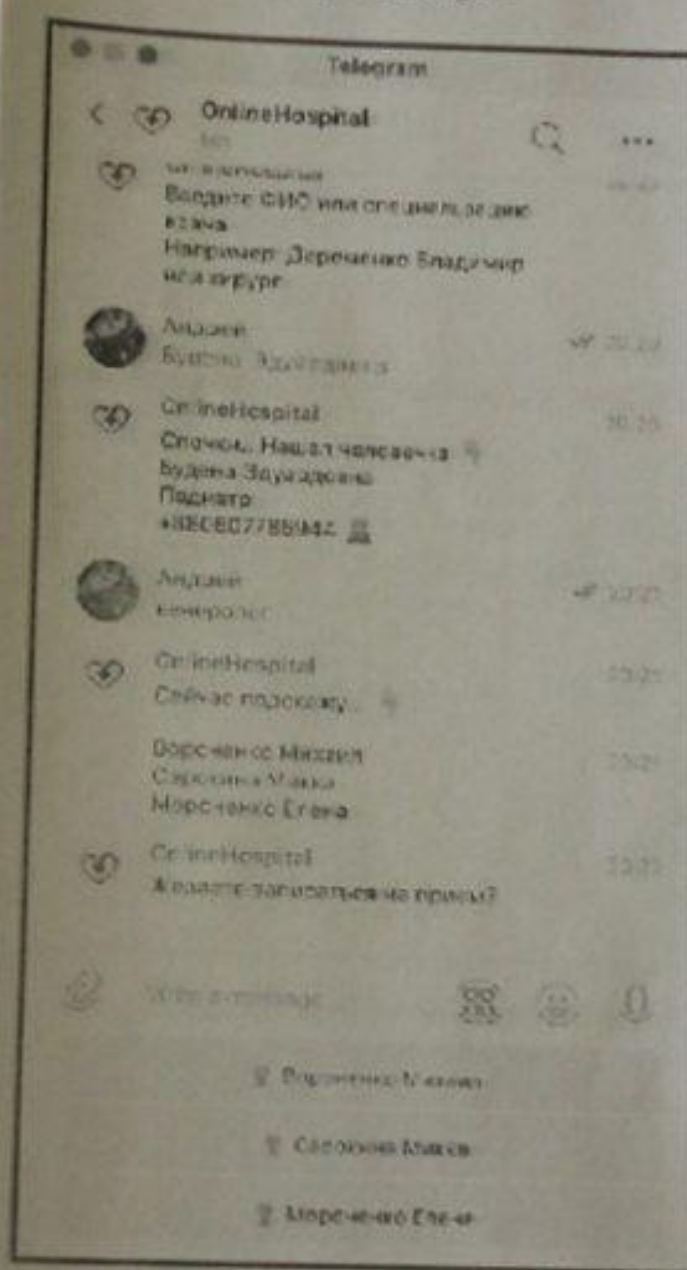
Стартова сторінка



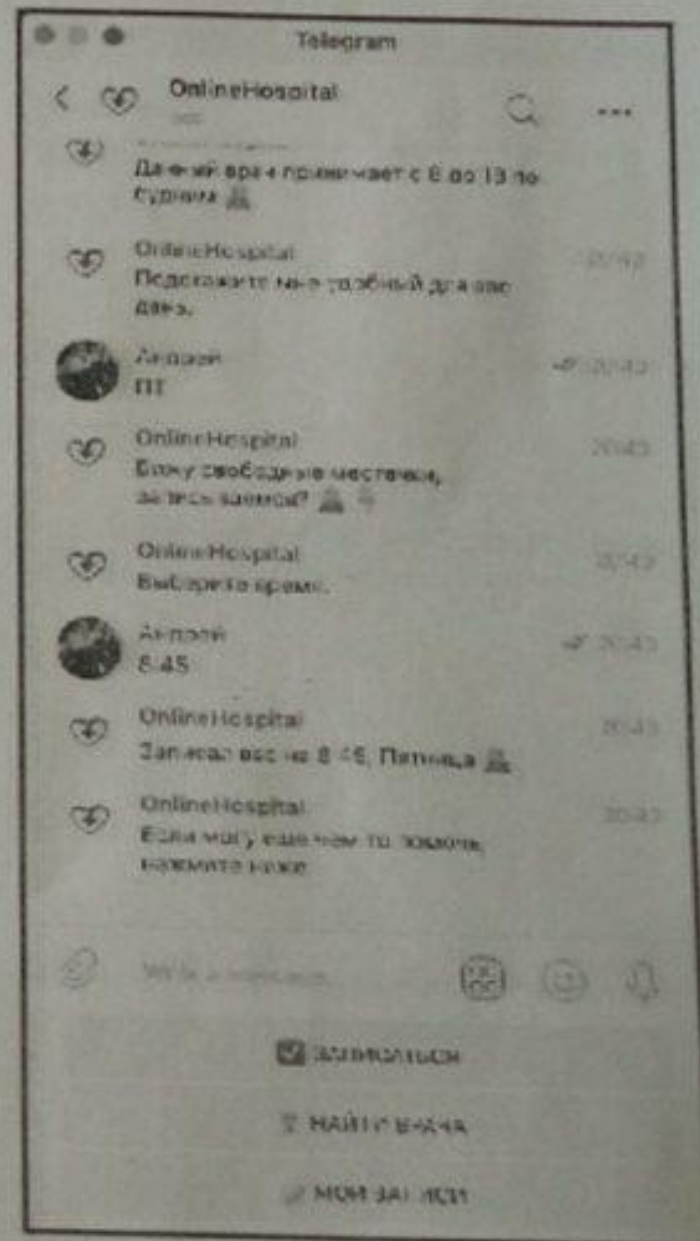
Вдала авторизація користувача



Пошук лікаря

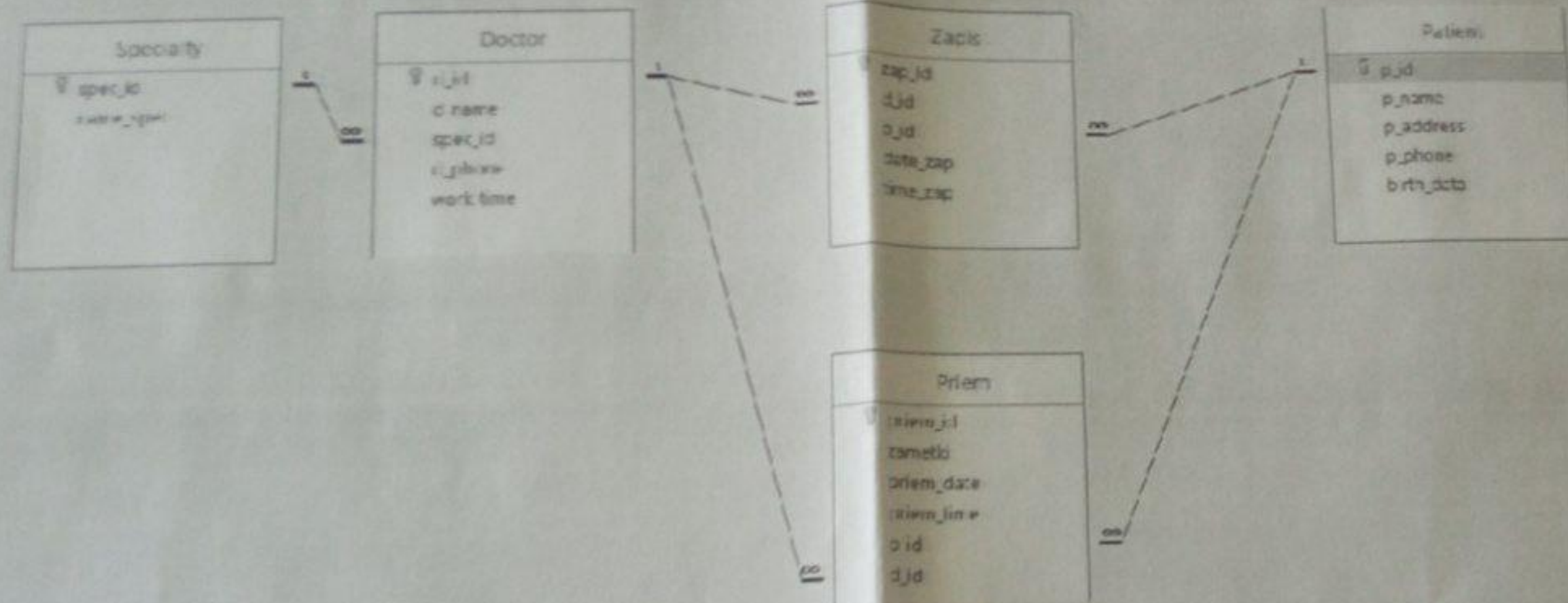


Запис до лікаря



						13.02070849.000003 ПЛ12						
Док.	Лист	№ докум.	Підпис	Дата	Розробка системи автоматизованого обліку в районній амбулаторії з використанням месенджерів Демонстрація роботи системи	Лист			Місяц		Місяць	
Розроб.		Богатирьова Т.О.										
Провер.		Земцова Л.А.										
Голов.												
						Лист 1			Листа 1			
Начальн.		Щербак Н.В.				НУ "Запорізька навігація" КНТ-317						
Заступник												

Схема Базы Данных



					13.02070849.000003 ПЛЗ		
Лек.	Леч.	Л. врач.	Л. врач.	Л. врач.	Разработка системы автоматизированного обслуживания в районной амбулатории с использованием мессенджера Схема базы данных		
Регистр.	Бухгалтерия Т.О.						
Перепр.	Земельная Л.В.						
Т. врач.							
И. врач.	Шерба Н.В.				ИУ "Защита информации" КИТ-517		
Защита							