

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ І ПРИКЛАДНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

МАЛАНЧУК ОЛЕКСАНДР ВОЛОДИМИРОВИЧ

Допускається до захисту:
в.о. завідувача кафедри
інформаційних технологій,
д-р техн. наук, професор
_____ Наталія ВЕСЕЛОВСЬКА
« ____ » _____ 2026 р.

РОЗРОБКА CRM-СИСТЕМИ ДЛЯ РЕКРУТИНГОВОЇ АГЕНЦІЇ

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (бакалаврська) робота

Керівник:
Роман БАБАКОВ, професор кафедри
інформаційних технологій,
д-р техн. наук, доцент

Оцінка: _____ / _____ / _____
(бали/за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____
(підпис)

Вінниця – 2026

АНОТАЦІЯ

Маланчук О.В. Розробка CRM-системи для рекрутингової агенції. Спеціальність 122 «Комп'ютерні науки», освітня програма «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця, 2026.

У бакалаврській роботі розглянуто процес розробки CRM-системи для рекрутингової агенції. У роботі проведено аналіз сучасних CRM-систем та особливостей автоматизації процесу рекрутингу.

Було реалізовано десктопний програмний додаток для роботи з кандидатами, вакансіями та задачами. CRM-система забезпечує збереження інформації про кандидатів, перегляд PDF-резюме, контроль етапів рекрутингу, систему задач та пошук інформації.

Для створення програмного забезпечення використовувались мова програмування Python, бібліотека PyQt6 та база даних SQLite.

У роботі також були створені UML-діаграми та ER-діаграма бази даних, проведено тестування системи та аналіз її функціональних можливостей.

Результатом роботи є створена CRM-система для автоматизації роботи рекрутингової агенції.

Ключові слова: CRM-система, рекрутинг, Python, PyQt6, SQLite, вакансії, кандидати, база даних, автоматизація.

ABSTRACT

Malanchuk O.V. Development of a CRM System for a Recruiting Agency. Specialty 122 “Computer Science”, educational program “Computer Science”. Vasyl’ Stus Donetsk National University, Vinnytsia, 2026.

The bachelor thesis considers the development of a CRM system for a recruiting agency. The paper analyzes modern CRM systems and the features of recruitment process automation.

A desktop application for working with candidates, vacancies, and tasks was developed. The CRM system provides storage of candidate information, PDF resume preview, recruitment stage control, task management, and search functionality.

Python programming language, PyQt6 library, and SQLite database were used for software development.

The thesis also includes UML diagrams, an ER database diagram, system testing, and analysis of the implemented functionality.

The result of the work is a developed CRM system for automating the work of a recruiting agency.

Keywords: CRM system, recruiting, Python, PyQt6, SQLite, vacancies, candidates, database, automation.

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА СУЧАСНИХ CRM-СИСТЕМ ДЛЯ РЕКРУТИНГУ	8
1.1 Особливості процесу рекрутингу та проблеми обліку кандидатів	8
1.2 Аналіз сучасних CRM-систем для рекрутингу	10
1.3 Постановка задачі та вимоги до системи.....	15
1.4 Аналіз засобів та технологій розробки	18
1.5 Постановка задач для реалізації CRM-системи	22
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ CRM-СИСТЕМИ	24
2.1 Проєктування архітектури CRM-системи та структури бази даних	24
2.2 UML-діаграми та моделювання системи.....	27
2.3 Алгоритм роботи CRM-системи.....	29
2.4 Опис структури бази даних.....	32
2.5 Підсумки проєктування та реалізації CRM-системи.....	34
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ CRM-СИСТЕМИ, ТЕСТУВАННЯ ТА АНАЛІЗ РОБОТИ.....	37
3.1 Опис інтерфейсу користувача.....	37
3.2 Реалізація основних функцій CRM-системи	41
3.3 Реалізація програмного коду CRM-системи	49
3.4 Тестування функціоналу CRM-системи	56
3.5 Основні переваги розробленої системи	57
3.6 Перспективи розвитку системи	59
ВИСНОВКИ.....	60
СПИСОК ВИКОРИТАНИХ ПОСИЛАНЬ.....	62
ДОДАТКИ.....	66

ВСТУП

У сучасному світі більшість компаній використовують інформаційні системи для автоматизації своєї роботи. Для рекрутингових агенцій, які щодня працюють з великою кількістю кандидатів, вакансій та замовників (роботодавців) автоматизація має суттєве значення. Використання електронних таблиць або записів вручну часто призводить до втрати інформації, помилок та незручностей у роботі. Відсутність централізованої системи обліку ускладнює контроль за етапами підбору кандидатів, планування задач та управління взаємодією з клієнтами.

Для спрощення процесу підбору персоналу все частіше використовуються CRM-системи. Такі системи дозволяють зберігати інформацію про кандидатів, вести вакансії, контролювати етапи співбесід, переглядати резюме та керувати задачами рекрутерів в одному місці.

Актуальність даної роботи полягає в тому, що багато невеликих рекрутингових агенцій не мають зручної та простої системи для роботи з кандидатами. Часто для цього використовуються Google таблиці, месенджери та інші спеціалізовані мобільні додатки та програми. Це ускладнює роботу та призводить до нераціональних витрат часу. Використання складних CRM-систем для невеликих рекрутингових агенцій, теж не є рентабельним із-за відносно високої вартості щомісячної абонентської плати за користування. Тому створення нескладної бюджетної спеціалізованої CRM-системи є корисним та актуальним завданням.

Метою бакалаврської роботи є розробка CRM-системи для невеликих рекрутингових агенцій, яка дозволить автоматизувати процес роботи з кандидатами та вакансіями. Система має бути простою у використанні, масштабованою та фінансово ефективною для невеликих рекрутингових агенцій, що не мають значних ресурсів для оплати дорогих комерційних рішень.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проаналізувати особливості роботи рекрутингових агенцій;
- дослідити існуючі CRM-системи, які використовуються у рекрутингу;
- сформулювати технічні вимоги для розробки продукту;
- обрати засоби та технології розробки;
- створити базу даних для збереження інформації;
- реалізувати інтерфейс програми;
- реалізувати функціонал роботи з кандидатами та вакансіями;
- додати можливість перегляду резюме;
- реалізувати систему задач та контролю етапів підбору персоналу;
- протестувати роботу програми.

Об'єктом дослідження є процес автоматизації роботи рекрутингової агенції.

Предметом дослідження є методи та програмні засоби створення CRM-системи для підбору персоналу.

Практичне значення роботи полягає у створенні готової CRM-системи, яку можна використовувати для роботи з кандидатами, вакансіями та задачами рекрутерів. Система має забезпечувати зручне керування інформацією та спрощувати процес рекрутингу. Система має бути фінансово ефективною для невеликих рекрутингових агенцій, тобто агенції не мають нести щомісячні фінансові витрати для роботи в системі.

Апробація результатів дослідження. Основні результати бакалаврської роботи були представлені у тезах доповіді на студентській науковій конференції кафедри інформаційних технологій Донецького національного університету імені Василя Стуса. Також розроблену CRM-систему планується передати на тестування та практичну експлуатацію в існуючу рекрутингову агенцію.

Структура кваліфікаційної (бакалаврської) роботи. Бакалаврська робота складається зі вступу, трьох розділів, висновків, списку використаних посилань та додатку. У першому розділі проведено аналіз сучасних CRM-систем та особливостей автоматизації процесу рекрутингу. У другому розділі описано процес проєктування та реалізації CRM-системи, структуру бази даних, UML-діаграми та проведено моделювання системи та розроблений алгоритм роботи CRM-системи. У третьому розділі описаний основний функціонал програми, реалізація програмного коду CRM-системи, результати проведеного тестування системи та аналіз результатів роботи CRM-системи.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА СУЧАСНИХ CRM-СИСТЕМ ДЛЯ РЕКРУТИНГУ

1.1 Особливості процесу рекрутингу та проблеми обліку кандидатів

У сучасному світі процес підбору персоналу є важливою частиною роботи будь-якої компанії. Саме від правильного підбору працівників залежить ефективність роботи підприємства, розвиток компанії та якість виконання завдань.

Процес рекрутингу включає:

- пошук кандидатів;
- отримання та аналіз резюме;
- проведення співбесід;
- оцінювання навичок;
- вибір працівників;
- найм працівників та їх адаптація.

На рис. 1.1 наведено основні етапи процесу рекрутингу.



Рисунок 1.1 – Основні етапи процесу рекрутингу

Раніше облік кандидатів часто здійснювався вручну. Інформація про кандидатів записувалась у паперових анкетах, блокнотах або звичайних таблицях на паперових носіях. Через це виникали проблеми з пошуком інформації, дублюванням даних та втратою резюме кандидатів. З розвитком інформаційних технологій компанії почали використовувати електронні таблиці та різні офісні

програми для ведення кандидатів і вакансій. Найчастіше для цього використовувались Microsoft Excel або Google таблиці. Такий підхід дозволив частково впорядкувати облік та роботу з кандидатами, проте він також має значну кількість недоліків.

Основними проблемами ручного обліку кандидатів (на паперових носіях або з використанням електронних таблиць) є:

- складність пошуку інформації;
- дублювання кандидатів;
- відсутність єдиної бази даних;
- втрата резюме;
- незручний контроль етапів співбесіди;
- велика кількість ручної роботи;
- складність роботи з великою кількістю вакансій.

Особливо гостро ці проблеми проявляються у рекрутингових агенціях, які одночасно працюють з великою кількістю кандидатів, вакансій та компаній (замовників).

Одним з найскладніших етапів рекрутингу в частині обліку кандидатів є контроль проходження кандидатів по воронці відбору на вакансіям. В рекрутингових компаніях один і той самий кандидат може розглядатися на декілька вакансій (декількох роботодавців) одночасно. Без використання спеціалізованих систем рекрутеру складно контролювати, на якому етапі відбору знаходиться кандидат, чи була проведена співбесіда та які результати були отримані.

Ще однією проблемою є збереження резюме кандидатів. Часто файли з резюме та інші документи (тестове завдання, анкетування та ін.) зберігаються в різних форматах та окремо від інформації про кандидата по відбору на вакансії, через що виникають труднощі з пошуком та переглядом даних.

У сучасних умовах використання лише таблиць або ручного обліку вже не дозволяє ефективно працювати з великим обсягом інформації. Саме тому все більшої популярності набувають CRM-системи та системи автоматизації рекрутингу.

Сучасні CRM-системи дозволяють автоматизувати процес підбору персоналу, контролювати етапи рекрутингу та централізовано зберігати інформацію про кандидатів [1].

Використання CRM-систем дозволяє:

- створити єдину базу кандидатів;
- автоматизувати процес рекрутингу;
- контролювати етапи підбору персоналу;
- зберігати резюме;
- швидко виконувати пошук інформації;
- спрощувати роботу рекрутера.

Таким чином, автоматизація процесу рекрутингу є актуальним завданням, а використання CRM-систем дозволяє значно підвищити ефективність роботи рекрутингових агенцій.

Автоматизація процесів рекрутингу дозволяє значно скоротити час роботи рекрутера та зменшити кількість помилок під час ведення кандидатів і вакансій [2].

1.2 Аналіз сучасних CRM-систем для рекрутингу

У сучасних умовах автоматизація процесу рекрутингу є важливою частиною роботи HR-відділів та рекрутингових агенцій. Для цього використовуються спеціалізовані CRM-системи, які дозволяють вести базу кандидатів, контролювати вакансії та автоматизувати процес підбору персоналу.

CRM-система (Customer Relationship Management) – це програмне забезпечення, яке використовується для збереження, обробки та управління інформацією про клієнтів або кандидатів.

CRM-системи активно використовуються у сучасних компаніях для автоматизації бізнес-процесів та централізованого збереження інформації [3, 4].

У сфері рекрутингу CRM-системи дозволяють:

- зберігати інформацію про кандидатів;
- вести вакансії;
- контролювати етапи співбесід;
- працювати з резюме;
- створювати задачі;
- автоматизувати роботу рекрутера.

1.2.1 Огляд найбільш поширених CRM-систем для рекрутингу

Сьогодні існує велика кількість CRM-систем, які використовуються для рекрутингу. Найбільш поширеними є Zoho Recruit, CleverStaff, Hurma System, BambooHR та Workable.

Zoho Recruit

Zoho Recruit є однією з найбільш відомих CRM-систем для автоматизації рекрутингу. Система дозволяє вести вакансії, працювати з кандидатами та автоматизувати процес найму.

Основними перевагами Zoho Recruit є:

- сучасний інтерфейс;
- велика кількість функцій;
- інтеграція з іншими сервісами;
- підтримка автоматизації.

Недоліками системи є:

- складність налаштування;
- велика кількість зайвих функцій;
- висока вартість платних тарифів.

CleverStaff

CleverStaff є українською CRM-системою для рекрутингу, яка використовується багатьма компаніями.

Система дозволяє:

- вести базу кандидатів;
- працювати з вакансіями;
- вести історію спілкування;
- автоматизувати процес рекрутингу.

Основним недоліком системи є залежність від інтернет-з'єднання та платна модель використання.

Hurma System

Hurma System використовується для автоматизації процесів рекрутингу. Система підтримує роботу з кандидатами, задачами та аналітикою.

Перевагами системи є:

- сучасний дизайн;
- підтримка HR-аналітики;
- зручна робота з кандидатами.

До недоліків можна віднести:

- складний інтерфейс для нових користувачів;
- платний доступ до більшості функцій.

BambooHR

BambooHR – це система, призначена для автоматизації процесів рекрутингу та управління персоналом у середніх і великих компаніях. Вона дозволяє зберігати інформацію про співробітників, спрощує рекрутинг та допомагає отримувати аналітику по процесах рекрутингу.

Переваги системи:

- сучасний та інтуїтивний інтерфейс;
- автоматизація процесів найму;
- інтеграція з іншими HR-інструментами;
- можливість формувати детальні звіти та аналітику по персоналу.

Недоліки системи:

- висока вартість платних планів;
- обмежені можливості налаштування для дуже великих організацій;
- деякі функції доступні лише у платних тарифах.

Workable

Workable – це відома CRM-система для рекрутингу, яка допомагає організовувати весь процес найму в компаніях різного масштабу. За допомогою системи можна публікувати вакансії, збирати та сортувати резюме, а також планувати співбесіди.

Переваги системи:

- зручний та зрозумілий інтерфейс;
- можливість публікації вакансій на різних платформах;
- автоматичне сортування та оцінка кандидатів;
- аналітичні інструменти для відстеження ефективності рекрутингу.

Основні недоліки:

- використання платне;

- деякі функції недоступні в базовому тарифі;
- для нових користувачів може знадобитися навчання.

Для наочності порівняння сучасних CRM-систем була створена табл. 1.1.

Таблиця 1.1 - Порівняння сучасних CRM-систем для рекрутингу

CRM система	Робота з кандидатами	PDF	Kanban	Задачі	Безкоштовна версія
Zoho Recruit	Так	Так	Так	Так	Частково
CleverStaff	Так	Так	Так	Так	Ні
Hurma System	Так	Так	Так	Так	Ні
BambooHR	Так	Так	Так	Так	Частково
Workable	Так	Так	Ні	Так	Ні

1.2.2 Аналіз існуючих CRM-систем для рекрутингу та обґрунтування необхідності власної розробки

В результаті аналізу сучасних CRM-систем було встановлено, що більшість існуючих рішень мають широкий функціонал, проте часто є складними у використанні. Багато CRM-систем містять велику кількість функцій, які не завжди потрібні невеликим рекрутинговим агенціям. Через це інтерфейс систем може бути перевантаженим та складним для нових користувачів. Складна система налаштувань вимагає додаткового навчання персоналу.

У багатьох системах також відсутня зручна робота з PDF-резюме або обмежені можливості локального збереження даних.

Розглянуті CRM-системи працюють лише через веб-інтерфейс та потребують постійного підключення до інтернету та використання хмарних ресурсів для

збереження інформації. У разі відсутності інтернет-з'єднання робота з такими системами стає неможливою.

Ще одним недоліком є висока вартість використання. Більшість професійних CRM-систем працюють за моделлю щомісячної підписки, що може бути незручним для невеликих рекрутингових агенцій.

Невеликі рекрутингові агенції зазвичай мають обмежений обсяг оборотних коштів та нестабільний дохід. Тому для невеликих рекрутингових агенцій основним недоліком використання сучасних CRM-систем є необхідність несення витрат на платну щомісячної підписку.

Виходячи з вищенаведених недоліків сучасних CRM-систем було прийняте рішення розробити власну CRM-систему для невеликих рекрутингових агенцій.

Розробка власної CRM-системи для невеликих рекрутингових агенцій є актуальною та виправданою оскільки система:

- забезпечуватиме основний необхідний функціонал для роботи рекрутера;
- буде простою у використанні;
- не буде мати складних налаштувань;
- буде незалежною від хмарних ресурсів (працюватиме як десктопний додаток);
- не вимагатиме щомісячних грошових витрат на підписку.

Актуальність даної розробки також була підтверджена у перемовинах з реальною рекрутинговою агенцією м. Вінниці, яка в подальшому погодилася провести тестування системи та можливе подальше співробітництво по розвитку системи в процесі її експлуатації.

1.3 Визначення вимог до системи

Основною метою розробки CRM-системи є автоматизація роботи рекрутингової агенції та спрощення процесу підбору персоналу. Система повинна

забезпечувати зручну роботу з кандидатами, вакансіями, задачами та резюме в одному програмному середовищі.

Під час роботи рекрутингових агенцій виникає велика кількість інформації, яку необхідно зберігати та швидко обробляти. До такої інформації належать резюме та анкети кандидатів, контактні дані, вакансії компаній, результати співбесід та різні задачі рекрутерів. Використання звичайних таблиць або ручного ведення записів є незручним та може призводити до втрати важливих даних.

Для вирішення даної проблеми була визначена необхідність створення CRM-системи з графічним інтерфейсом, яка дозволить автоматизувати основні процеси рекрутингу.

Розроблювана система повинна забезпечувати такі можливості:

- додавання та редагування інформації про кандидатів;
- збереження контактної інформації;
- завантаження та перегляд резюме кандидатів у PDF-форматі;
- створення та редагування вакансій;
- прикріплення кандидатів до вакансій;
- контроль етапів підбору персоналу;
- створення задач та нагадувань;
- пошук та фільтрацію інформації;
- ведення бази даних кандидатів;
- перегляд статистики системи.

На рис. 1.2 показані основні функції CRM-системи для рекрутингової агенції. Система дозволяє працювати з кандидатами, вакансіями, задачами та резюме кандидатів, а також виконувати пошук і фільтрацію даних. Це значно спрощує роботу рекрутера та дозволяє швидко отримувати необхідну інформацію.

Основні можливості користувача CRM-системи



Рисунок 1.2 – Основні можливості користувача CRM-системи

Однією з важливих функцій системи є робота з резюме кандидатів. Користувач повинен мати можливість завантажувати PDF-файли з резюме та переглядати їх без відкриття сторонніх програм.

Також система повинна мати простий та зрозумілий інтерфейс користувача. Це дозволить швидко навчитися працювати з програмою навіть користувачам без технічних знань.

В процесі розробки особлива увага має приділятися зручності використання системи. Інтерфейс програми має бути створений таким чином, щоб основні функції знаходились у швидкому доступі для користувача.

В результаті було сформовано основні вимоги до CRM-системи, які необхідно реалізувати в процесі створення програмного продукту.

Для коректної роботи CRM-системи були визначені функціональні та нефункціональні вимоги.

До функціональних вимог належать:

- робота з кандидатами;
- робота з вакансіями;
- система задач;
- система пошуку;
- робота з базою даних;
- перегляд резюме;
- фільтрація інформації;
- збереження даних.

До нефункціональних вимог належать:

- стабільна робота програми;
- швидке відкриття вікон;
- простий інтерфейс;
- зручна навігація;
- безпечне збереження даних;
- можливість подальшого розширення системи.

1.4 Аналіз засобів та технологій розробки

Для створення CRM-системи для рекрутингової агенції необхідно обрати сучасні та зручні засоби розробки. При створенні програми акцент робиться на простоті використання, високій швидкості та потенціалі розширення функцій системи.

Під час вибору технологій особлива увага приділяється стабільності роботи, простоті підтримки програмного забезпечення та можливості подальшого розширення функціоналу системи [5].

Для розробки програмного забезпечення такого виду CRM-системи доцільно обрати мову програмування Python. Дана мова є однією з найпопулярніших мов програмування у світі завдяки простому синтаксису, великій кількості бібліотек та підтримці різних напрямків розробки програмного забезпечення.

Python є однією з найпоширеніших мов програмування для створення desktop-додатків, автоматизації процесів та роботи з базами даних [6, 7, 8].

Python дозволяє швидко створювати десктопні програми, працювати з базами даних, реалізовувати графічний інтерфейс та автоматизувати різні процеси. Також перевагою Python є велика кількість документації та прикладів, що значно спрощує процес розробки.

Для створення графічного інтерфейсу можливо використати бібліотеку PyQt6. Дана бібліотека дозволяє створювати сучасні віконні програми з кнопками, таблицями, формами та іншими елементами інтерфейсу.

Бібліотека PyQt6 дозволяє створювати сучасні графічні інтерфейси користувача та підтримує велику кількість компонентів для desktop-додатків [9, 10, 11].

За допомогою PyQt6 у програмі є можливість реалізувати:

- головне вікно системи;
- таблиці кандидатів та вакансій;
- форми додавання та редагування даних;
- вкладки та панелі керування;
- систему задач;
- перегляд PDF-резюме;
- фільтрацію та пошук інформації.

Однією з головних переваг PyQt6 є можливість створення сучасного інтерфейсу користувача та підтримка великої кількості елементів керування.

Для збереження інформації доцільно обрати базу даних SQLite. SQLite є легкою вбудованою базою даних, яка не потребує встановлення окремого сервера. Уся інформація зберігається у спеціальному файлі бази даних.

SQLite була обрана через:

- простоту використання;
- швидкість роботи;
- легку інтеграцію з Python;
- підтримку SQL-запитів;
- можливість роботи без підключення до інтернету.

У базі даних CRM-системи зберігаються:

- кандидати;
- вакансії;
- задачі;
- резюме;
- користувачі;
- нотатки;
- історія змін.

Для роботи з PDF-файлами резюме можливо використати додаткові можливості Python та PyQt6. Це дозволить реалізувати перегляд резюме без відкриття сторонніх програм.

Під час розробки програми також раціонально використовувати об'єктно-орієнтована модель програмування. Це дозволяє розділити програму на окремі модулі та спростити підтримку коду.

Структура програми логічно відокремити на декілька основних модулів:

- модуль роботи з кандидатами;
- модуль вакансій;
- модуль задач;

- модуль бази даних;
- модуль авторизації;
- модуль перегляду резюме.

На рис. 1.3 показана структура CRM-системи для рекрутингової агенції.

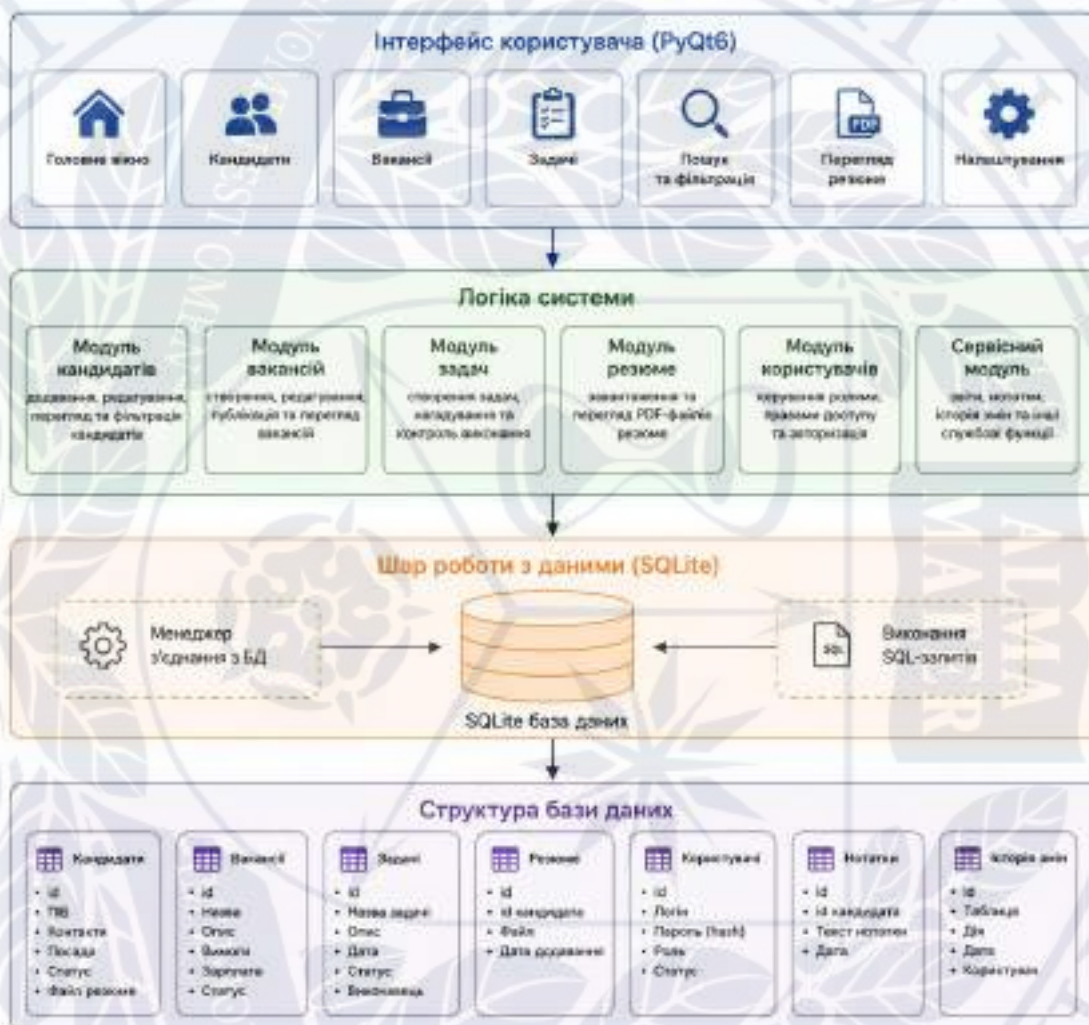


Рисунок 1.3 – Структура CRM-системи для рекрутингової агенції

Система складається з графічного інтерфейсу користувача, логіки роботи програми та бази даних SQLite. Основні модулі системи відповідають за роботу з кандидатами, вакансіями, задачами, резюме та користувачами системи. Така структура є зручною для організації роботи програми та забезпечує швидкий доступ до інформації.

Описаний підхід дозволяє легко додавати нові функції та підтримувати стабільну роботу програми.

Для написання коду доцільно використовувати середовище розробки PyCharm. Дане середовище забезпечує зручну роботу з Python-проектами, автоматичну перевірку помилок та спрощує процес налагодження програми.

В результаті аналізу засобів та технологій розробки було встановлено, що використання Python, PyQt6 та SQLite є хорошим рішенням для створення CRM-системи рекрутингової агенції, оскільки ці технології забезпечують простоту розробки, зручний інтерфейс та стабільну роботу системи.

1.5 Постановка задач для реалізації CRM-системи

Основною метою розробки системи є автоматизація процесу роботи з кандидатами, вакансіями та етапами рекрутингу.

Під час роботи рекрутингових агенцій виникає велика кількість інформації, яку необхідно швидко зберігати, обробляти та редагувати. Використання ручного обліку або звичайних таблиць створює труднощі під час роботи з великою кількістю кандидатів та вакансій.

Для вирішення даної проблеми, враховуючи сформовані вимоги до системи та обрані засоби та технології розробки, була проведена деталізація задач створення CRM-системи з графічним інтерфейсом користувача.

В процесі розробки необхідно реалізувати такі задачі:

- систему роботи з кандидатами;
- систему роботи з вакансіями;
- збереження контактної інформації;
- завантаження та перегляд PDF-резюме;
- систему задач та нагадувань;
- kanban-дошку етапів рекрутингу;

- пошук та фільтрацію інформації;
- dashboard зі статистикою;
- резервне копіювання бази даних;
- експорт інформації в таблицю Excel.

Для реалізації CRM-системи необхідно:

- спроектувати архітектуру CRM-системи та структуру бази даних;
- розробити алгоритм роботи CRM-системи;
- реалізувати графічний інтерфейс користувача;
- забезпечити взаємодію між модулями системи;
- реалізувати збереження даних;
- протестувати стабільність роботи програми.

Під час створення системи особливу увагу потрібно приділяти:

- простоті використання;
- швидкості роботи;
- зручності інтерфейсу;
- стабільності збереження даних;
- можливості подальшого розширення функціоналу.

Майбутня CRM-система повинна забезпечувати швидку та зручну роботу рекрутера з кандидатами, вакансіями та задачами в одному програмному середовищі.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ CRM-СИСТЕМИ

2.1 Проєктування архітектури CRM-системи та структури бази даних

На рис. 2.1 показана загальна архітектура CRM-системи для рекрутингової агенції. Система складається з графічного інтерфейсу, логіки роботи програми та бази даних SQLite. Основними модулями системи є модуль кандидатів, вакансій, задач, пошуку та керування користувачами. Усі дані зберігаються у базі даних, що дозволяє швидко отримувати доступ до інформації та забезпечує стабільну роботу системи.

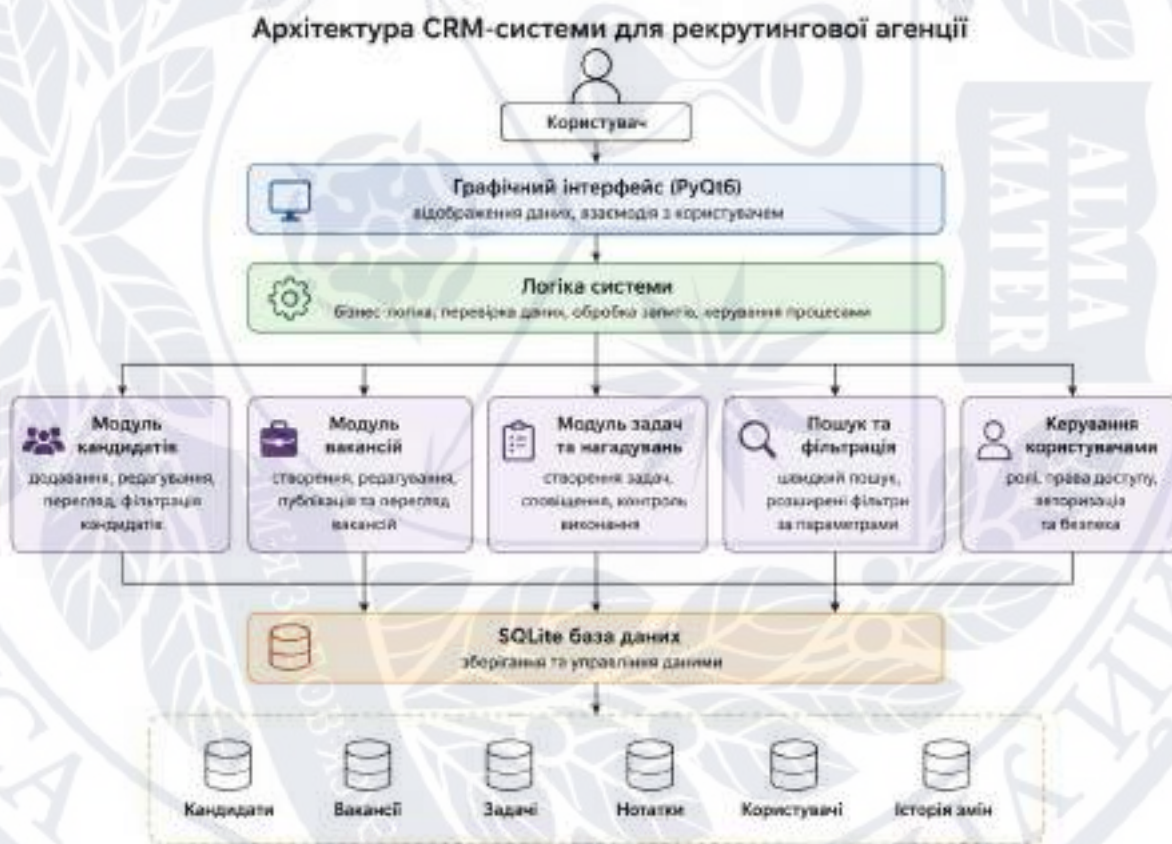


Рисунок 2.1 – Архітектура CRM-системи для рекрутингової агенції

Архітектурно CRM-система складається з декількох основних таблиць, між якими реалізовано зв'язки для обміну інформацією.

Основними таблицями системи є:

- таблиця кандидатів;
- таблиця вакансій;
- таблиця задач;
- таблиця користувачів;
- таблиця навичок;
- таблиця зв'язків кандидатів та вакансій.

Таблиці в CRM-системі створені для структурованого збереження та обробки даних, необхідних для ефективного управління рекрутинговими процесами.

- Таблиця кандидатів використовується для зберігання інформації про кандидатів: ПІБ, номер телефону, електронну пошту, навички, джерело пошуку та шлях до PDF-резюме.
- Таблиця вакансій містить інформацію про відкриті вакансії: назву вакансії, компанію, клієнта, заробітну плату, тип роботи, локацію та вимоги до кандидатів.
- Таблиця задач створюється для планування та контролю роботи рекрутерів. Вона дозволяє відстежувати етапи підбору персоналу та контроль виконання задач.
- Таблиця користувачів забезпечує авторизацію в системі та містить логіни, паролі та ролі користувачів.
- Таблиця навичок використовується для збереження професійних навичок кандидатів та дозволяє виконувати пошук і фільтрацію кандидатів за необхідними навичками (наприклад, Python, SQL, Excel або іншими технологіями).
- Таблиця зв'язків кандидатів і вакансій дозволяє прикріплювати кандидатів до певних вакансій та зберігати інформацію про етапи проходження кандидата (новий, на розгляді, співбесіда, прийнятий або відхилений).

Між таблицями реалізовано зв'язки, які дозволяють прикріплювати кандидатів до вакансій та зберігати інформацію про їхній статус в процесі рекрутингу.

В процесі розробки CRM-системи важливу роль відіграє база даних, оскільки саме вона забезпечує збереження всієї інформації про кандидатів, вакансії, задачі та користувачів системи.

Для даного програмного продукту була використана база даних SQLite. Вибір SQLite обумовлений простотою використання, високою швидкістю роботи та відсутністю необхідності встановлення окремого серверу баз даних. Це дозволяє запускати систему навіть на звичайному персональному комп'ютері без додаткового налаштування.

На рис. 2.2 показана структура бази даних CRM-системи.

Основними таблицями системи є таблиці кандидатів, вакансій, задач, користувачів та навичок. Для зв'язку кандидатів і вакансій використовується окрема таблиця `candidate_vacancies`, яка дозволяє зберігати інформацію про етап проходження кандидата. За допомогою даної таблиці система підтримує роботу kanban-дошки та дозволяє відстежувати статус кандидата у процесі рекрутингу.

Використання зв'язків між таблицями дозволяє забезпечити цілісність даних та зручну роботу з інформацією у системі. База даних SQLite забезпечує швидке збереження та отримання даних без необхідності використання окремого серверу.

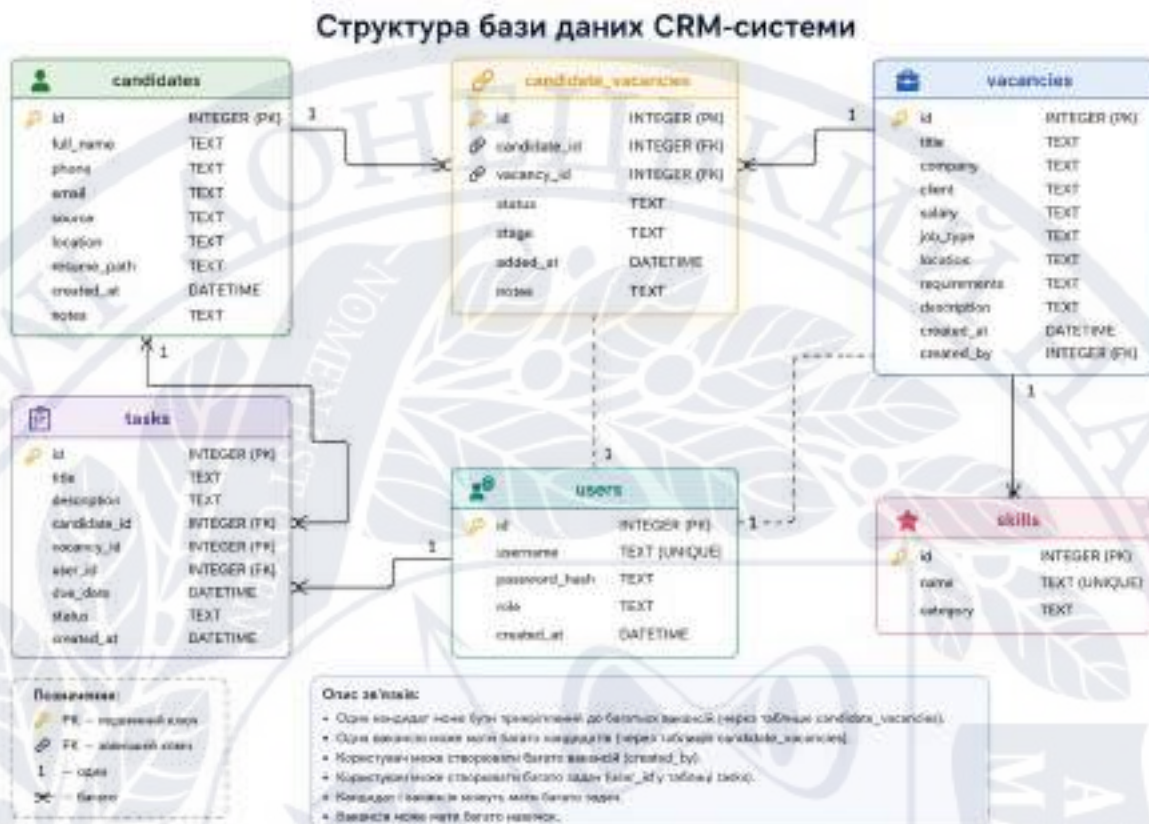


Рисунок 2.2 – Структура бази даних CRM-системи

2.2 UML-діаграми та моделювання системи

Під час проєктування CRM-системи було виконано моделювання основних процесів та структури програми. Для цього використовувались UML-діаграми, які дозволяють наочно показати взаємодію користувача із системою та структуру програмного забезпечення.

Використання UML-діаграм дозволяє спростити процес проєктування системи, краще організувати структуру програми та виявити основні зв'язки між компонентами системи.

В процесі проєктування CRM-системи були створені такі діаграми:

- діаграма варіантів використання (Use Case Diagram);
- діаграма класів.

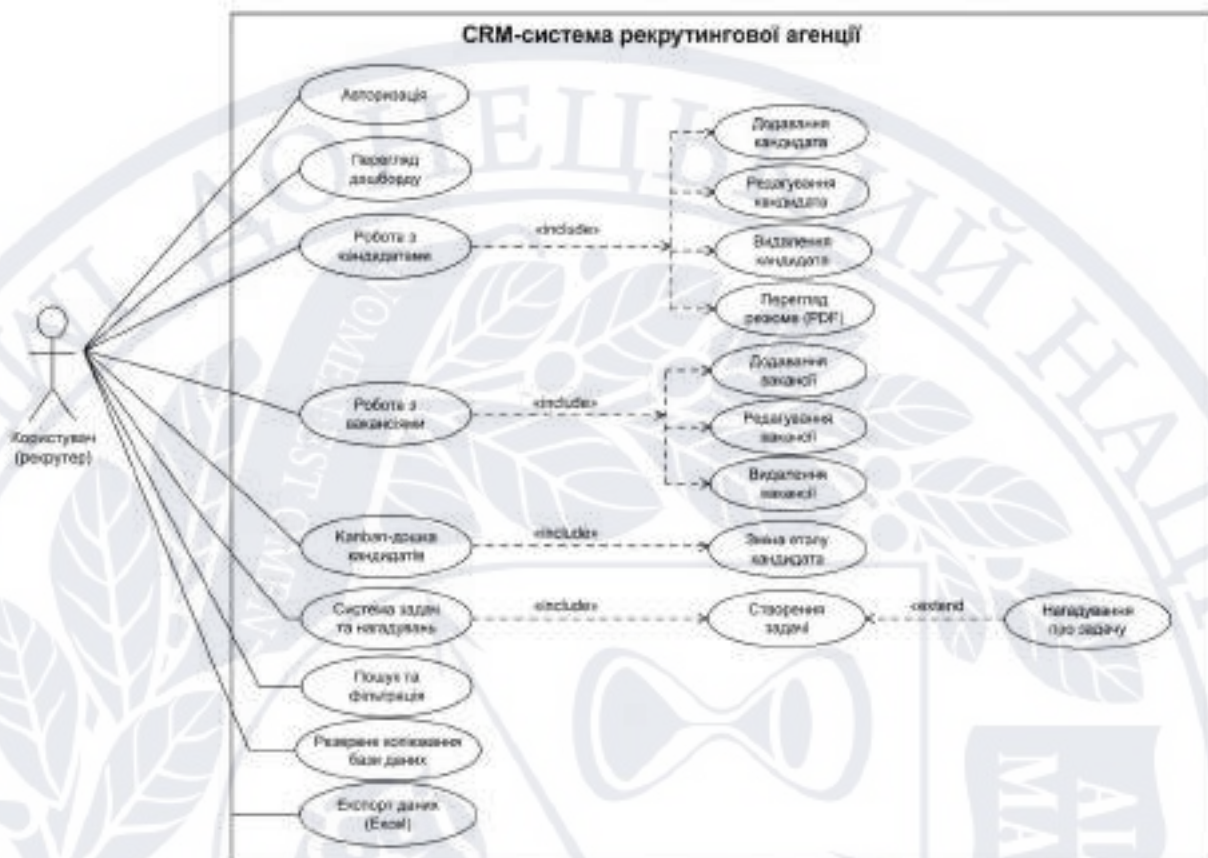


Рисунок 2.3 – UML-діаграма варіантів використання CRM-системи

На рис. 2.3 UML-діаграма варіантів використання показує, які дії користувач може виконувати у CRM-системі. Основним користувачем є рекрутер. Він може авторизуватись, працювати з кандидатами та вакансіями, переглядати резюме, користуватись канбан-дошкою, створювати задачі, виконувати пошук, робити резервне копіювання бази даних та експортувати інформацію в Excel.

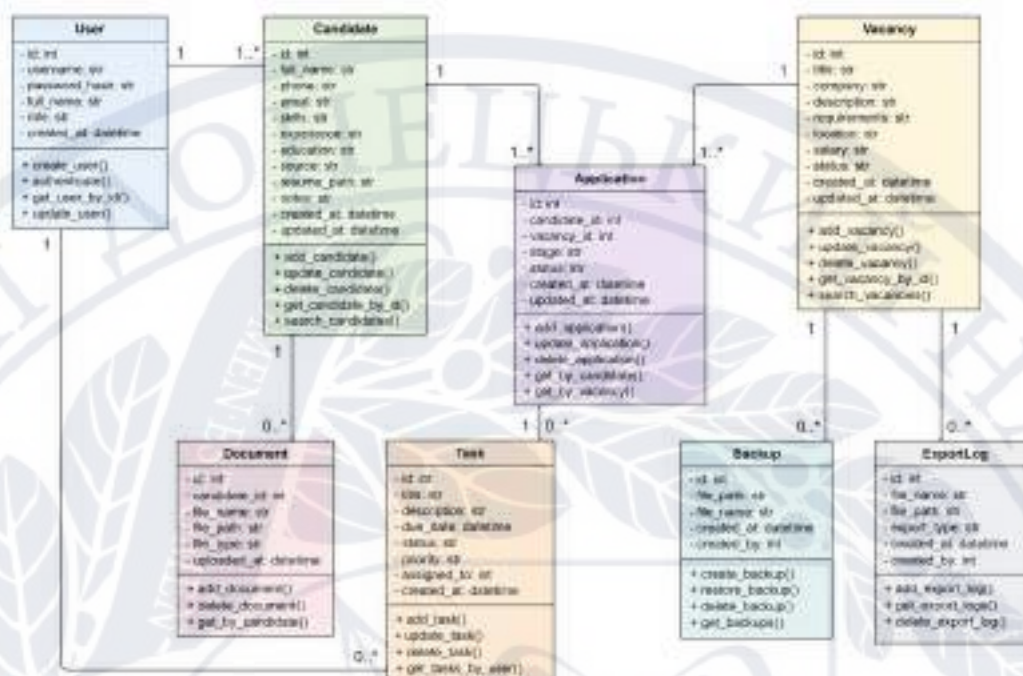


Рисунок 2.4 – UML-діаграма класів CRM-системи

UML-діаграма класів (рис. 2.4) показує основні класи CRM-системи, їх атрибути, методи та зв'язки між ними. Основними класами системи є Candidate, Vacancy, Application, User та Task. Діаграма демонструє структуру програмного забезпечення та взаємодію між основними компонентами системи.

2.3 Алгоритм роботи CRM-системи

Для забезпечення правильної роботи CRM-системи було спроектовано загальний алгоритм взаємодії користувача з програмою (алгоритм роботи CRM-системи).

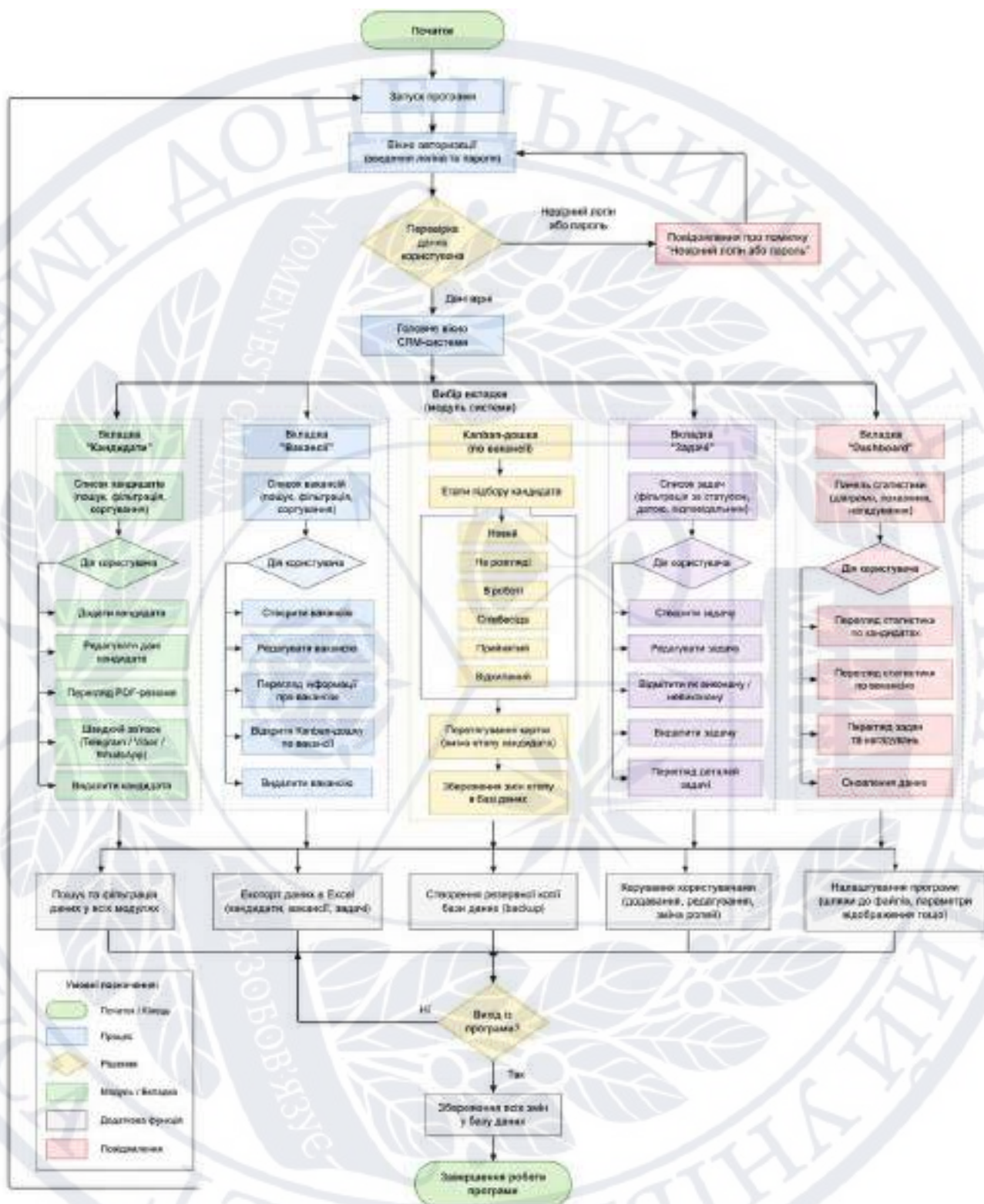


Рисунок 2.5 – Блок-схема алгоритму роботи CRM-системи

На рис. 2.5 представлено загальний алгоритм роботи CRM-системи для рекрутингової агенції. Система забезпечує авторизацію користувачів, роботу з кандидатами, вакансіями та задачами, підтримує kanban-дошку, перегляд PDF-резюме, пошук і фільтрацію даних, резервне копіювання бази даних та експорт інформації у формат Excel.

Після запуску програми користувач проходить авторизацію шляхом введення логіна та пароля. У разі успішної перевірки відкривається головне вікно CRM-системи.

Головне вікно містить основні вкладки:

- кандидати;
- вакансії;
- задачі;
- dashboard.

У вкладці кандидатів користувач може:

- додавати нових кандидатів;
- редагувати інформацію;
- переглядати PDF-резюме;
- використовувати швидкий перехід до Telegram, Viber та WhatsApp.

У вкладці вакансій користувач може:

- створювати вакансії;
- редагувати записи;
- прикріплювати кандидатів;
- працювати з kanban-дошкою.

Kanban-дошка дозволяє змінювати статус кандидатів та контролювати етапи проходження вакансії.

У вкладці задач користувач може створювати нагадування та контролювати виконання задач.

Також система підтримує:

- пошук та фільтрацію;
- резервне копіювання бази даних;
- експорт інформації у Excel.

Після завершення роботи користувач може закрити програму, при цьому всі дані автоматично зберігаються у базі SQLite.

2.4 Опис структури бази даних

Для збереження інформації у CRM-системі використовується база даних SQLite. Даний тип бази даних був обраний через простоту використання, швидку роботу та відсутність необхідності встановлення окремого серверу.

База даних CRM-системи забезпечує збереження:

- інформації про кандидатів;
- вакансій;
- задач;
- користувачів;
- резюме;
- історії змін.

Структура бази даних була розроблена таким чином, щоб забезпечити швидкий доступ до інформації та зручну взаємодію між таблицями.

Основними таблицями системи є:

- candidates;
- vacancies;
- applications;
- tasks;
- users.

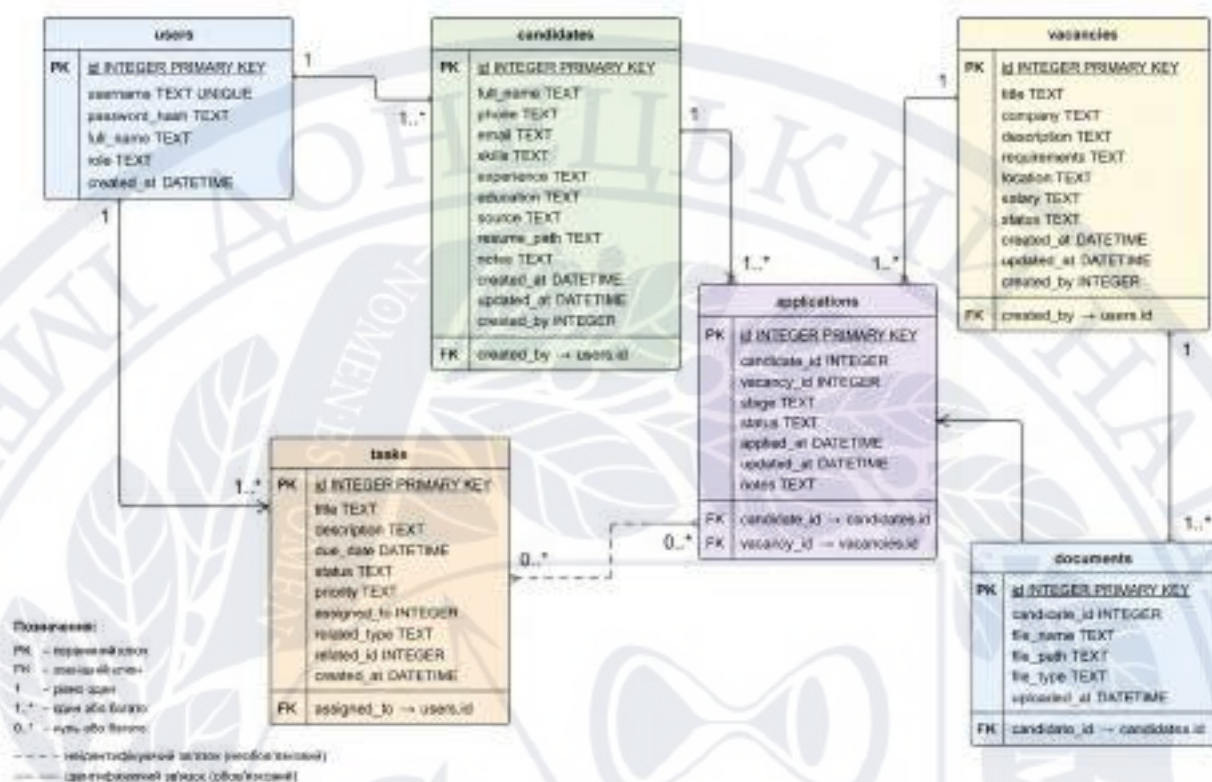


Рисунок 2.6 – ER-діаграма бази даних CRM-системи

На рис. 2.6 показано структуру бази даних CRM-системи та зв'язки між основними таблицями. Центральною таблицею системи є applications, яка забезпечує зв'язок між кандидатами та вакансіями. Таблиця candidates зберігає основну інформацію про кандидатів, включаючи ПІБ, контакти, навички, джерела надходження та шлях до PDF-резюме. Таблиця vacancies містить дані про відкриті вакансії та їхні характеристики, що дозволяє прикріплювати кандидатів і контролювати етапи їх проходження. Таблиця tasks забезпечує планування та контроль виконання завдань рекрутерами, тоді як таблиця users регулює авторизацію та права доступу в системі. Таблиця skills використовується для фільтрації кандидатів за ключовими навичками, а таблиця documents дозволяє зберігати додаткові файли кандидатів, прикріплені до відповідних записів у базі даних.

Завдяки такій структурі забезпечується прозорий потік даних між усіма модулями CRM, що дозволяє ефективно відстежувати рекрутингові процеси та підтримувати масштабованість системи для подальшого розвитку.

2.5 Підсумки проектування та реалізації CRM-системи

У цьому розділі кваліфікаційної (бакалаврської) роботи було проведено проектування реалізації CRM-системи для рекрутингової агенції.

В процесі проектування було створено архітектуру CRM-системи та структуру бази даних. Виконано моделювання основних процесів та структури програми з використанням UML-діаграм. Також був розроблений алгоритм роботи CRM-системи.

У CRM-системі реалізовано:

- систему роботи з кандидатами;
- систему вакансій;
- kanban-дошку етапів рекрутингу;
- систему задач та нагадувань;
- перегляд PDF-резюме;
- систему авторизації користувачів;
- пошук та фільтрацію інформації;
- резервне копіювання бази даних;
- експорт даних у формат Excel.

Основними модулями в структурі CRM-системи є:

- Модуль «Кандидати»:
 - зберігає всю базову інформацію про кандидатів: ПІБ, контакти, навички, джерела надходження та шлях до PDF-резюме;
 - є центральним елементом системи, оскільки більшість процесів рекрутингу обертаються навколо кандидатів;

- забезпечує швидкий доступ до інформації та фільтрацію за ключовими параметрами.
- Модуль «Вакансії»:
 - зберігає інформацію про відкриті позиції: назву вакансії, компанію, клієнта, зарплатний діапазон, тип роботи, локацію та вимоги до кандидатів;
 - дозволяє прикріплювати кандидатів до вакансій та відстежувати етапи їхнього проходження;
 - створює логічний зв'язок між кандидатами та вакансіями, що полегшує контроль рекрутингових процесів.
- Модуль «Задачі»:
 - призначений для планування та контролю роботи рекрутерів: створення завдань, призначення відповідальних, відстеження дедлайнів і статусів виконання;
 - забезпечує структуровану роботу команди та контроль за виконанням ключових етапів підбору персоналу.
- Модуль «Користувачі»:
 - відповідає за авторизацію, управління ролями та правами доступу користувачів;
 - дозволяє розмежувати права рекрутера і адміністратора, що забезпечує безпеку даних та контроль дій у системі.

Структура CRM-системи є оптимальною так як в ній реалізовано:

- логічна сегментація – кожен модуль відповідає за окремий аспект рекрутингу, що робить систему зрозумілою і легкою для підтримки;
- взаємозв'язок модулів – кандидати прикріплюються до вакансій, задачі прив'язані до кандидатів та користувачів, що забезпечує прозорість процесів;

- масштабованість – при необхідності легко додати нові модулі або функції (аналітика, інтеграції з LinkedIn, штучний інтелект);
- безпека – модуль користувачів дозволяє контролювати права доступу та захищати конфіденційну інформацію.

В процесі розробки передбачено реалізацію додаткових функцій, які покращують зручність використання системи, зокрема така вкрай необхідна для рекрутерів функція, як швидкий перехід до Telegram, Viber та WhatsApp для зв'язку з кандидатами.

В результаті виконання другого розділу кваліфікаційної роботи було створено повноцінну архітектуру CRM-систему для автоматизації роботи рекрутингової агенції, яка забезпечує зручну роботу з кандидатами, вакансіями та задачами в одному програмному середовищі.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ CRM-СИСТЕМИ, ТЕСТУВАННЯ ТА АНАЛІЗ РОБОТИ

3.1 Опис інтерфейсу користувача

Для зручної роботи рекрутерів у CRM-системі було створено графічний інтерфейс користувача за допомогою бібліотеки PyQt6. Інтерфейс програми дозволяє швидко працювати з кандидатами, вакансіями, задачами та резюме.

Після запуску програми користувач бачить головне вікно CRM-системи, яке містить основні вкладки та елементи керування.

Основний інтерфейс системи складається з:

- панелі керування;
- вкладки кандидатів;
- вкладки вакансій;
- системи задач;
- панелі статистики;
- системи пошуку та фільтрації.

На головному екрані відображається коротка статистика системи: кількість кандидатів, вакансій, активних задач та прийнятих кандидатів.

На рис. 3.1 показано головне вікно CRM-системи. Користувач може швидко перейти до вкладок кандидатів, вакансій та задач, а також переглянути основну статистику роботи системи.



Рисунок 3.1 – Головне вікно CRM-системи

Однією з основних частин CRM-системи є вкладка роботи з кандидатами. Саме у цьому розділі користувач може переглядати інформацію про кандидатів, редагувати дані, додавати резюме та виконувати пошук.

В таблиці кандидатів відображаються:

- ім'я кандидата;
- номер телефону;
- електронна пошта;
- навички;
- джерело кандидата;
- прикріплені вакансії;
- статус кандидата.

Для зручності користувача у системі реалізовано пошук та фільтрацію кандидатів за різними параметрами. Це дозволяє швидко знаходити необхідну інформацію навіть при великій кількості записів.

Також у системі реалізовано можливість перегляду PDF-резюме без використання сторонніх програм. Резюме відкривається безпосередньо у вікні CRM-системи.

ID	Ім'я кандидата	Телефон	Email	Прізвище	Діагностика	Вимоги	Статус	Рівень	Функція
18	Степанов Максим	+380999999999	maxim@gmail.com	Інженер / IT-спеціаліст	Діагностика	1	IT - Manager - IT-спеціаліст	---	Адміністратор системи
17	Олександр Коваленко	+380999999999	oleksandr@gmail.com	Інженер + Інженер + Інженер	Діагностика	1	IT - Manager - IT-спеціаліст	---	Адміністратор системи
28	Людмила Гордієнко	+380999999999	ludmila@gmail.com	IT-спеціаліст	Діагностика	1	IT - Manager - IT-спеціаліст	---	Адміністратор системи
15	Петренко Дмитро	+380999999999	dmitro@gmail.com	IT-спеціаліст + IT-спеціаліст + Інженер	Діагностика	1	Project Manager - IT-спеціаліст	---	Адміністратор системи
14	Коваленко Дмитро	+380999999999	dmitro@gmail.com	Інженер + Інженер	Діагностика	1	Project Manager - IT-спеціаліст	---	Адміністратор системи
13	Савченко Максим	+380999999999	maxim@gmail.com	IT-спеціаліст + Інженер	Діагностика	1	Project Manager - IT-спеціаліст	---	Адміністратор системи
12	Романенко Дмитро	+380999999999	dmitro@gmail.com	Інженер + Інженер + IT-спеціаліст	Діагностика	1	Project Manager - IT-спеціаліст	---	Адміністратор системи
27	Павленко Дмитро	+380999999999	dmitro@gmail.com	IT-спеціаліст	Діагностика	1	IT - Manager - IT-спеціаліст	---	Адміністратор системи
25	Коваленко Дмитро	+380999999999	dmitro@gmail.com	Інженер + Інженер	Діагностика	1	Project Manager - IT-спеціаліст	---	Адміністратор системи
16	Коваленко Дмитро	+380999999999	dmitro@gmail.com	IT-спеціаліст + IT-спеціаліст	Діагностика	1	Project Manager - IT-спеціаліст	---	Адміністратор системи
18	Коваленко Дмитро	+380999999999	dmitro@gmail.com	Інженер	Діагностика	1	Project Manager - IT-спеціаліст	---	Адміністратор системи
17	Коваленко Дмитро	+380999999999	dmitro@gmail.com	Інженер + Інженер + Інженер	Діагностика	1	Project Manager - IT-спеціаліст	---	Адміністратор системи
16	Коваленко Дмитро	+380999999999	dmitro@gmail.com	Інженер + Інженер	Діагностика	2	IT - Manager - IT-спеціаліст	---	Адміністратор системи
13	Тетяна	+380999999999	tetiana@gmail.com	Інженер	Реконструкція	1	Project Manager - IT-спеціаліст	Високий	Адміністратор системи
14	Людмила Гордієнко	+380999999999	ludmila@gmail.com	Інженер / IT-спеціаліст	Діагностика	1	Project Manager - IT-спеціаліст	Високий	Адміністратор системи
15	Коваленко Дмитро	+380999999999	dmitro@gmail.com	Інженер + Інженер	Реконструкція	2	Project Manager - IT-спеціаліст	Високий	Адміністратор системи

Рисунок 3.2 – Вкладка роботи з кандидатами

На рис. 3.2 показана вкладка роботи з кандидатами. В таблиці відображається основна інформація про кандидатів, а також доступні функції додавання, редагування, видалення та пошуку даних.

Наступною важливою частиною системи є модуль роботи з вакансіями. В даному модулі користувач може створювати нові вакансії, вказувати вимоги до кандидатів, зарплату, тип роботи та іншу інформацію.

Для кожної вакансії у системі можна:

- переглядати детальну інформацію;
- прикріплювати кандидатів;
- змінювати статус;
- переглядати етапи проходження кандидатів.

ID	Назва вакансії	Компанія	Клієнт	Зарплата	Локація	Тип роботи	Статус	Відповідальний	Кількість кандидатів
1	IT-Менеджер	ITC	Спеціалізована	2000\$ - 2500\$	Київ	Фул-тайм	Відкрито	Олександр Мамонтов	7
2	Конструктор	Інженер	Сервіс	1000\$ - 1500\$	Вінниця	Північ	Відкрито	Андрій Петрович	3
3	Продюсер	ITC	Спеціалізована	2000\$ - 2500\$	Київ	Фул-тайм	Відкрито	Олександр Мамонтов	11

Рисунок 3.3 – Вкладка роботи з вакансіями

На рис. 3.3 представлена вкладка роботи з вакансіями. В таблиці відображається основна інформація про вакансії: назва, компанія, клієнт, зарплата, локація та кількість кандидатів.

Для роботи з вакансіями у CRM-системі реалізовано набір функцій керування записами.

Користувач може:

- додавати нові вакансії;
- редагувати існуючі записи;
- видаляти вакансії;
- переглядати детальну інформацію;
- прикріплювати кандидатів до вакансії.

Функція редагування дозволяє змінювати інформацію про вакансію без необхідності створення нового запису.

Функція видалення використовується для очищення бази даних від неактуальних вакансій.

Кнопка перегляду деталей дозволяє відкривати додаткову інформацію про вакансію та переглядати кандидатів, прикріплених до неї.

На рис. 3.3 також зображено основні елементи керування вакансіями: кнопки додавання, редагування, видалення та перегляду деталей вакансії.

3.2 Реалізація основних функцій CRM-системи

В процесі розробки CRM-системи було реалізовано набір функцій, необхідних для автоматизації роботи рекрутингової агенції. Основною метою реалізації функціоналу було спрощення процесу роботи з кандидатами та вакансіями.

Однією з головних функцій системи є відображення кандидатів за етапами проходження вакансії. Для цього в програмі реалізовано спеціальну kanban-дошку.

Kanban-дошка дозволяє розподіляти кандидатів за такими статусами:

- новий;
- на розгляді;
- в роботі;
- співбесіда;
- прийнятий;
- відхилений.

Такий підхід дозволяє швидко контролювати процес підбору персоналу та бачити поточний стан кожного кандидата.



Рисунок 3.4 – Kanban-дошка кандидатів у CRM-системі

На рис. 3.4 представлена система відображення кандидатів за етапами проходження вакансії. Користувач може швидко переглядати поточний статус кандидатів та контролювати процес рекрутингу.

Ще однією важливою функцією CRM-системи є можливість перегляду PDF-резюме без використання сторонніх програм. Це дозволяє рекрутеру швидко переглядати резюме кандидатів безпосередньо у програмі.

Для реалізації даної функції було використано вбудований механізм відображення PDF-файлів у графічному інтерфейсі PyQt6.

Користувач може:

- завантажувати PDF-резюме;
- переглядати сторінки резюме;
- перемикались між сторінками;
- відкривати резюме без сторонніх програм.

На рис. 3.5 показано вбудований перегляд PDF-резюме у CRM-системі. Завдяки цьому рекрутер може переглядати резюме кандидатів без використання сторонніх програм.

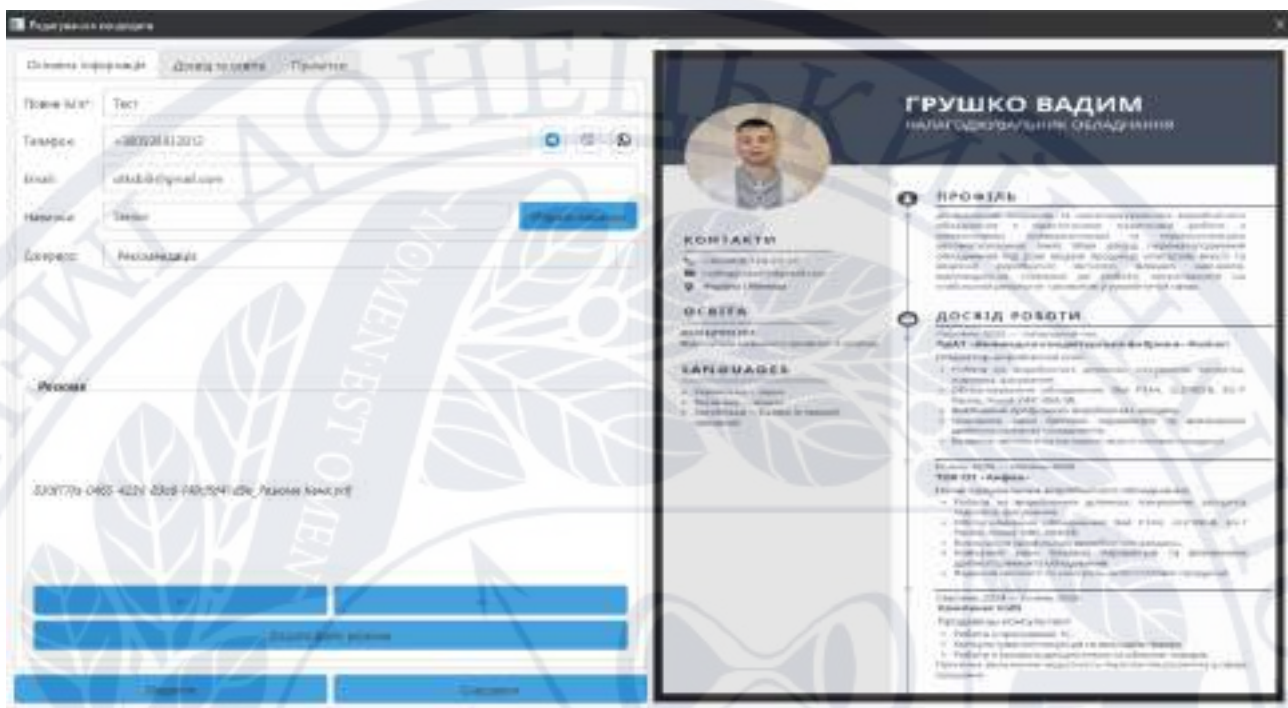


Рисунок 3.5 – Перегляд PDF-резюме кандидата

У CRM-системі також реалізовано систему задач та нагадувань. Даний модуль дозволяє рекрутеру створювати задачі, контролювати їх виконання та не втрачати важливу інформацію під час роботи з кандидатами.

Користувач може створювати задачі для:

- проведення співбесіди;
- повторного контакту з кандидатом;
- перевірки резюме;
- зв'язку з компанією;
- контролю етапів підбору персоналу.

Кожна задача містить:

- назву;
- опис;
- дату;

- статус виконання.

Реалізація системи задач дозволяє організувати роботу рекрутера та спрощує контроль процесу рекрутингу.

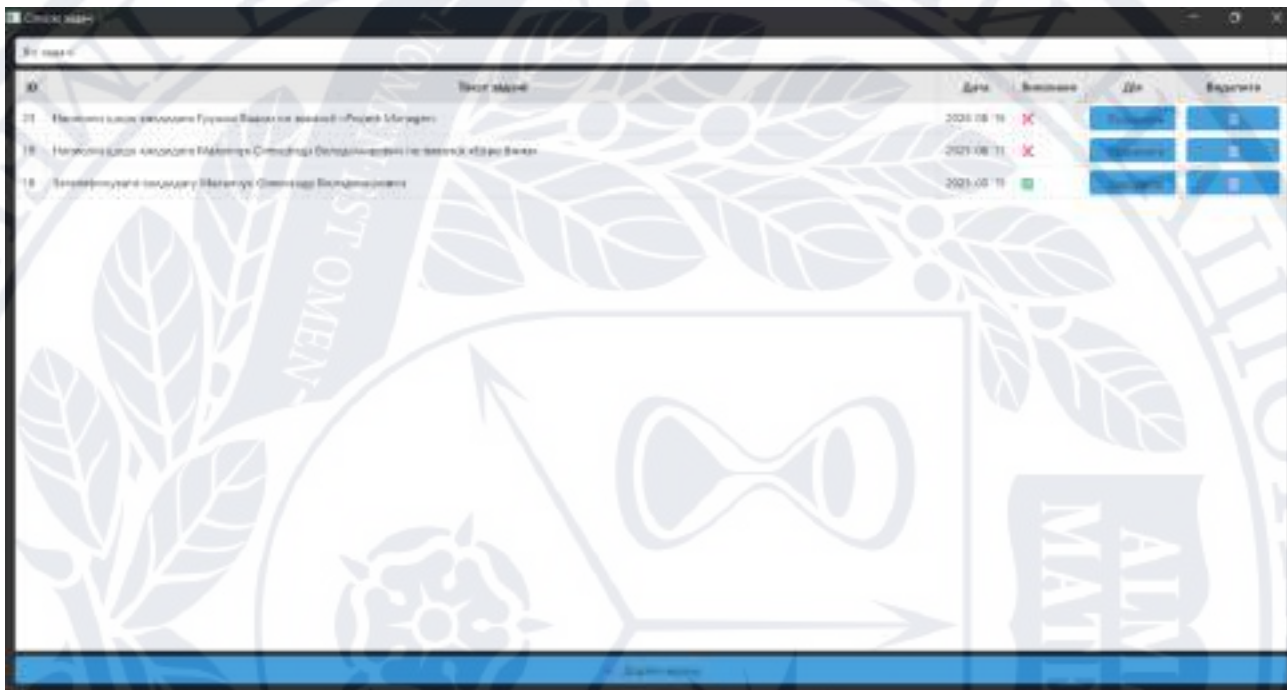


Рисунок 3.6 – Система задач та нагадувань

На рис. 3.6 показана система задач CRM-системи. Користувач може створювати задачі, контролювати їх виконання та відстежувати процес роботи з кандидатами.

Для створення безпеки даних, у CRM-системі реалізовано систему авторизації користувачів. Перед початком роботи користувачу потрібно ввести логін та пароль.

В системі створені такі ролі користувачів:

- адміністратор;
- рекрутер.

Адміністратор має доступ до всіх модулів системи та може керувати користувачами. Рекрутер працює з кандидатами, вакансіями та задачами.

Система авторизації дозволяє обмежити доступ до інформації та забезпечує безпечну роботу з даними.

Рисунок 3.7 – Вікно авторизації користувача

На рис. 3.7 представлено вікно авторизації CRM-системи. Для входу користувач повинен ввести логін та пароль. Після успішної авторизації відкривається головне вікно системи.

На рис. 3.8 представлений модуль керування користувачами. Адміністратор може створювати нових користувачів, змінювати ролі, скидати паролі та видаляти облікові записи.

Для швидкої роботи з великою кількістю інформації у CRM-системі реалізовано систему пошуку та фільтрації даних.

Управління користувачами

ID	Логін	Повне ім'я	Роль	Дата створення	Останній вхід
1	admin	Адміністратор системи	Адміністратор	2025-05-08 12:50:29	2026-05-19 17:41:42
2	Alexander	Олександр Маланчук	Адміністратор	2025-05-08 12:58:41	2025-05-14 12:51:33
5	Angelina	Ангеліна Форкош	Рекрутер	2026-05-17 13:55:29	

Додати нового користувача

Логін:

Повне ім'я:

Пароль:

Підтвердження пароля:

Роль:

Рисунок 3.8 – Модуль керування користувачами CRM-системи

Користувач може виконувати пошук кандидатів та вакансій за:

- ім'ям;
- компанією;
- статусом;
- навичками;
- локацією;
- відповідальним рекрутером.

Система фільтрації дозволяє швидко знаходити необхідну інформацію та спрощує роботу з великою кількістю записів у базі даних.

Пошук виконується у реальному часі, що забезпечує швидке оновлення результатів без перезавантаження програми.



Ім'я кандидата	Номер телефону	Електронна пошта	Посада	Департамент	Статус	Дії
Тарасенко	+380990000000	taras@gmail.com	12345 - 43210 - 12345	Marketing	1	Project Manager - Бюджетування
Іванко	+380990000000	ivan@gmail.com	123 - 456 - 789	Marketing	1	Account Manager - Продажі
Петренко	+380990000000	petr@gmail.com	987654321 - 123456789	Marketing	1	HR Manager - Рекрутинг
Сидоренко	+380990000000	sidor@gmail.com	12345 - 67890 - 12345	Marketing	1	HR Manager - Рекрутинг
Попов	+380990000000	popov@gmail.com	12345	Marketing	1	HR Manager - Рекрутинг

Рисунок 3.9 – Система пошуку та фільтрації даних

На рис. 3.9 представлена система пошуку та фільтрації CRM-системи.

Для надійного збереження даних у CRM-системі реалізовано функцію створення резервних копій бази даних. Це дозволяє зберігати копії інформації та відновлювати систему у випадку втрати даних.

Функція резервного копіювання створює окремий файл бази даних, який містить усю інформацію системи:

- кандидатів;
- вакансії;
- задачі;
- користувачів;
- резюме.

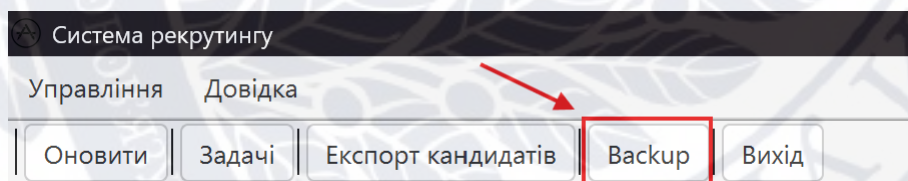


Рисунок 3.10 – Функція резервного копіювання бази даних

На рис. 3.10 показана функція створення резервної копії бази даних CRM-системи.

Також у CRM-системі реалізовано функцію експорту кандидатів у форматі Excel. Дана функція дозволяє зберігати інформацію у вигляді таблиці для подальшого використання або створення звітів.

Експорт дозволяє:

- зберігати список кандидатів;
- створювати звіти;
- переносити інформацію до інших систем;
- виконувати додатковий аналіз даних.

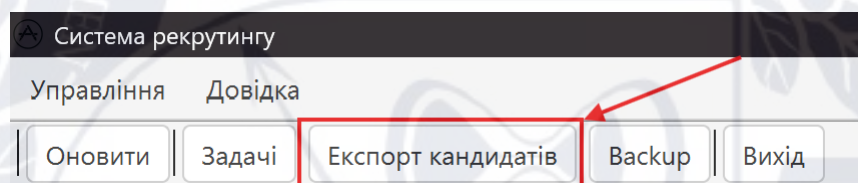


Рисунок 3.11 – Експорт кандидатів у формат Excel

На рис. 3.11 представлена функція експорту кандидатів у формат Excel.

Для покращення комунікації з кандидатами у CRM-системі реалізовано функцію швидкого переходу до месенджерів.

У формі кандидата біля поля номера телефону розміщені кнопки:

- Telegram;
- Viber;
- WhatsApp.

Після натискання на відповідну кнопку система автоматично відкриває необхідний месенджер для зв'язку з кандидатом.

Дана функція дозволяє:

- швидко переходити до спілкування;
- не копіювати номер телефону вручну.

Це забезпечує:

- економію часу рекрутера;
- спрощення процесу комунікацій.

Телефон:

+380999999999

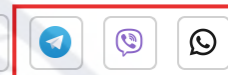


Рисунок 3.12 – Форма кандидата з кнопками швидкого зв'язку

На рис. 3.12 представлена форма кандидата з реалізованими кнопками швидкого переходу до Telegram, Viber та WhatsApp.

3.3 Реалізація програмного коду CRM-системи

Під час розробки CRM-системи програмний код був поділений на окремі модулі. Такий підхід дозволив зробити структуру проєкту більш зрозумілою та спростив подальшу підтримку програми.

Основна логіка системи реалізована за допомогою об'єктно-орієнтованого програмування. Для кожного основного модуля було створено окремі класи та файли.

Структура проєкту складається з таких основних директорій:

- ui - графічний інтерфейс користувача;
- core - логіка роботи системи;
- database - робота з базою даних;
- models - моделі даних;
- utils - допоміжні функції.

Головним елементом програми є головне вікно системи, яке відповідає за відкриття вкладок, навігацію та взаємодію між модулями.

Для реалізації графічного інтерфейсу використовувались можливості бібліотеки PyQt6. Вона дозволила створити сучасний інтерфейс із таблицями, кнопками, вкладками та формами введення даних.

Для роботи з базою даних використовувались SQL-запити та ORM-модель взаємодії з SQLite. Це дозволило зручно працювати зі збереженням та отриманням інформації.

Нижче наведено фрагмент програмного коду створення головного вікна CRM-системи.

```
class MainWindow(QMainWindow):
    def __init__(self, db_manager, auth_manager):
        super().__init__()

        self.db_manager = db_manager
        self.auth_manager = auth_manager

        self.candidate_manager = CandidateManager(db_manager,
auth_manager)
        self.vacancy_manager = VacancyManager(db_manager,
auth_manager)
        self.application_manager = ApplicationManager(db_manager,
auth_manager)

        self.init_ui()
        self.update_status_bar()

    def init_ui(self):
        self.setWindowTitle("Система рекрутингу")
        self.setMinimumSize(1000, 700)

        self.create_menu()
        self.create_toolbar()
        self.create_tabs()

        self.statusBar = QStatusBar()
        self.setStatusBar(self.statusBar)
```

Рисунок 3.13 – Фрагмент коду головного вікна CRM-системи

У цьому класі підключаються основні менеджери для роботи з кандидатами, вакансіями та прив'язками кандидатів до вакансій. Також створюється головний інтерфейс програми, меню, панель інструментів та вкладки системи.

```

def save_candidate(self):
    full_name = self.name_edit.text().strip()

    if not full_name:
        QMessageBox.warning(
            self,
            "Помилка збереження",
            "Поле 'Повне ім'я' є обов'язковим."
        )
        self.name_edit.setFocus()
        return

    phone = self.phone_edit.text().strip()
    email = self.email_edit.text().strip()
    skills = self.skills_input.text().strip()
    source = self.source_combo.currentText().strip()
    experience = self.experience_edit.toPlainText().strip()
    education = self.education_edit.toPlainText().strip()
    notes = self.notes_edit.toPlainText().strip()

    candidate_id = self.candidate_manager.add_candidate(
        full_name=full_name,
        phone=phone,
        email=email,
        skills=skills,
        experience=experience,
        education=education,
        source=source,
        resume_path=self.resume_paths[self.current_resume_index] if
self.resume_paths else None,
        notes=notes
    )

```

Рисунок 3.14 – Фрагмент коду збереження кандидата

На рис. 3.14 наведено фрагмент коду збереження кандидата. Спочатку система перевіряє, чи заповнене обов'язкове поле з іменем кандидата. Після цього з форми зчитуються контактні дані, навички, джерело, досвід, освіта, нотатки та шлях до резюме. Отримані дані передаються до менеджера кандидатів для збереження у базі даних.

Для роботи з вакансіями у CRM-системі реалізовано окремі методи створення, редагування та видалення вакансій.

В процесі збереження вакансії система перевіряє правильність введених даних та передає їх до менеджера вакансій для запису у базу даних.

```
def save_vacancy(self):  
    title = self.title_edit.text().strip()  
  
    if not title:  
        QMessageBox.warning(  
            self,  
            "Помилка збереження",  
            "Поле 'Назва вакансії' є обов'язковим."  
        )  
        self.title_edit.setFocus()  
        return  
  
    status = self.status_combo.currentData()  
    assigned_recruiter = self.recruiter_combo.currentData()  
  
    company_name = self.company_edit.text().strip()  
    client_name = self.client_edit.text().strip()  
    salary_from = self.salary_from_edit.text().strip()  
    salary_to = self.salary_to_edit.text().strip()  
    location = self.location_edit.text().strip()  
    work_type = self.work_type_combo.currentText()  
  
    description = self.description_edit.toPlainText().strip()  
    requirements = self.requirements_edit.toPlainText().strip()  
  
    vacancy_id = self.vacancy_manager.add_vacancy(  
        title=title,  
        description=description,  
        requirements=requirements,  
        status=status,  
        assigned_recruiter=assigned_recruiter,  
        company_name=company_name,  
        client_name=client_name,  
        salary_from=salary_from,  
        salary_to=salary_to,  
        location=location,  
        work_type=work_type  
    )
```

Рисунок 3.15 – Фрагмент коду створення вакансії

На рис. 3.15 наведено фрагмент коду створення та збереження вакансії. Спочатку система перевіряє, чи заповнено обов'язкове поле з назвою вакансії. Після цього з форми зчитуються основні дані: статус, відповідальний рекрутер, компанія, клієнт, зарплата, локація, тип роботи, опис та вимоги. Отримані дані передаються до менеджера вакансій для збереження у базі даних.

Для контролю етапів проходження кандидатів у CRM-системі реалізовано kanban-дошку. Система автоматично розподіляє кандидатів за статусами та дозволяє змінювати етап проходження вакансії.

Для реалізації даного функціоналу було створено окремі методи оновлення статусу кандидата та синхронізації інформації з базою даних.

```
def update_candidate_stage(self, candidate_id, vacancy_id,
new_stage):

    query = """
    UPDATE candidate_vacancies
    SET stage = ?
    WHERE candidate_id = ? AND vacancy_id = ?
    """

    self.db.execute(
        query,
        (new_stage, candidate_id, vacancy_id)
    )

    self.load_candidates()
```

Рисунок 3.16 – Фрагмент коду оновлення статусу кандидата у kanban-дошці

На рис. 3.16 наведено фрагмент коду оновлення статусу кандидата у kanban-дошці. Метод отримує ідентифікатор кандидата, вакансії та новий етап проходження. Після цього система виконує SQL-запит для оновлення статусу у таблиці candidate_vacancies та автоматично оновлює інтерфейс програми.

Для перегляду резюме у PDF-форматі у CRM-системі реалізовано вбудований механізм відкриття PDF-файлів. Це дозволяє рекрутеру переглядати резюме кандидатів без використання сторонніх програм.

Система підтримує завантаження PDF-файлів та відображення сторінок у графічному інтерфейсі PyQt6.

```
def load_pdf_preview(self, pdf_path):  
  
    self.pdf_document = fitz.open(pdf_path)  
  
    page = self.pdf_document.load_page(0)  
    pix = page.get_pixmap()  
  
    image = QImage(  
        pix.samples,  
        pix.width,  
        pix.height,  
        pix.stride,  
        QImage.Format.Format_RGB888  
    )  
  
    self.pdf_label.setPixmap(QPixmap.fromImage(image))
```

Рисунок 3.17 – Фрагмент коду відкриття PDF-резюме

На рис. 3.17 наведено фрагмент коду відкриття PDF-резюме. Метод завантажує PDF-файл, відкриває першу сторінку документа та відображає її у графічному інтерфейсі CRM-системи.

Для організації роботи рекрутерів у CRM-системі реалізовано модуль задач та нагадувань. Система дозволяє створювати задачі, змінювати їх статус та контролювати виконання.

Кожна задача може містити:

- назву;
- опис;
- дату виконання;
- статус;
- відповідального користувача.

На рис. 3.18 наведено фрагмент коду створення задачі. Метод отримує назву задачі, опис та дату виконання, після чого передає інформацію до менеджера задач для збереження у базі даних SQLite.

```
def create_task(self):  
    title = self.title_edit.text().strip()  
    description = self.description_edit.toPlainText().strip()  
    due_date = self.date_edit.date().toString("yyyy-MM-dd")  
  
    if not title:  
        QMessageBox.warning(  
            self,  
            "Помилка",  
            "Введіть назву задачі."  
        )  
        return  
  
    self.task_manager.add_task(  
        title=title,  
        description=description,  
        due_date=due_date  
    )  
    self.accept()
```

Рисунок 3.18 – Фрагмент коду створення задачі

Для створення безпеки даних у CRM-системі реалізовано модуль авторизації користувачів.

Перед початком роботи користувач повинен ввести логін та пароль. Після успішної перевірки даних система відкриває головне вікно програми.

У CRM-системі підтримуються ролі:

- адміністратор;
- рекрутер.

Адміністратор має доступ до керування користувачами та системними функціями, а рекрутер працює з кандидатами, вакансіями та задачами.

```
def login(self):  
  
    username = self.username_edit.text().strip()  
    password = self.password_edit.text().strip()  
  
    user = self.auth_manager.authenticate(  
        username,  
        password  
    )  
  
    if user:  
        self.accept()  
    else:  
        QMessageBox.warning(  
            self,  
            "Помилка",  
            "Невірний логін або пароль."  
        )
```

Рисунок 3.19 – Фрагмент коду авторизації користувача

На рис. 3.19 наведено фрагмент коду авторизації користувача. Система отримує логін та пароль, після чого виконує перевірку через модуль авторизації. У разі успішної перевірки відкривається головне вікно CRM-системи.

3.4 Тестування функціоналу CRM-системи

Після завершення розробки CRM-системи було проведено тестування основних функцій програми. Основною метою функціонального тестування була перевірка правильності роботи системи, стабільності збереження даних та коректної взаємодії між модулями.

Тестування проводилось на операційній системі Windows 11 із використанням локальної бази даних SQLite.

Під час тестування перевірялись такі функції:

- авторизація користувача;
- додавання кандидатів;
- редагування кандидатів;
- створення вакансій;

- прикріплення кандидатів до вакансій;
- kanban-дошка;
- перегляд PDF-резюме;
- система задач;
- пошук та фільтрація;
- резервне копіювання бази даних (Backup);
- експорт кандидатів в таблицю Excel.

Для тестування використовувались тестові кандидати, вакансії та PDF-резюме.

За результатами тестування усі режими програми працюють коректно. Під час роботи програма забезпечує швидке збереження інформації та правильну взаємодію з базою даних SQLite.

Також було перевірено роботу системи при великій кількості кандидатів та вакансій. В процесі тестування критичних помилок виявлено не було.

3.5 Основні переваги розробленої системи

В результаті аналізу розробки та тестування CRM-системи було визначено основні переваги створеного програмного продукту.

Однією з головних переваг системи є простий та зрозумілий інтерфейс користувача. Усі основні функції розміщені у зручних вкладках та доступні без складних налаштувань.

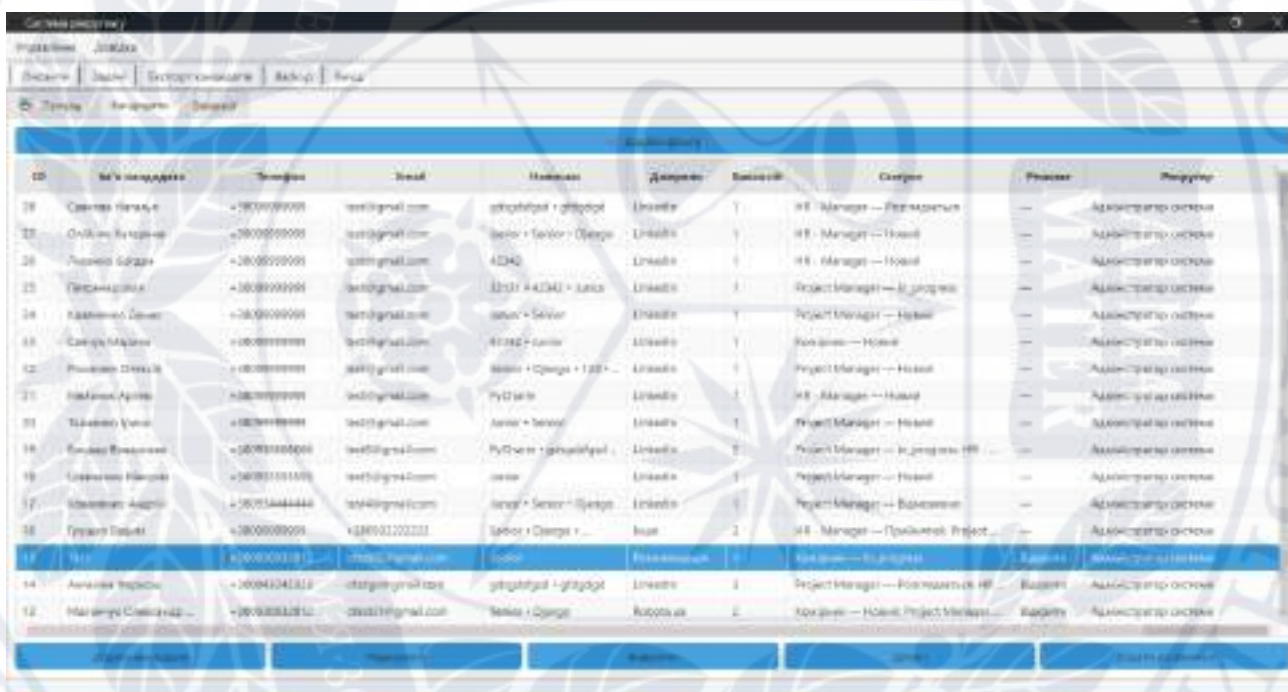
CRM-система дозволяє:

- зберігати інформацію про кандидатів;
- працювати з вакансіями;
- контролювати етапи рекрутингу;
- переглядати PDF-резюме;
- створювати задачі;

- виконувати резервне копіювання бази даних;
- експортувати інформацію в таблицю Excel.

Ще однією важливою перевагою системи є використання kanban-дошки кандидатів. Вона дозволяє рекрутеру швидко бачити поточний стан підбору персоналу та контролювати процес роботи з кандидатами.

Також перевагою системи є використання локальної бази даних SQLite. Це дозволяє використовувати програму без окремого серверу та забезпечує простоту встановлення.



ID	Ім'я кандидата	Телефон	Email	Позиція	Департамент	Статус	Роль
18	Савченко Наталія	+38099999999	nat@gmail.com	Software Engineer	IT	Available	IT Manager - Recruitment
22	Олійник Катерина	+38099999999	kat@gmail.com	Senior Software Engineer	IT	Available	IT Manager - IT
26	Лисенко Євген	+38099999999	evgen@gmail.com	QA	IT	Available	IT Manager - IT
25	Петренко Дмитро	+38099999999	dmit@gmail.com	Junior Software Engineer	IT	Available	Project Manager - IT
24	Коваленко Дмитро	+38099999999	dmit@gmail.com	Senior Software Engineer	IT	Available	Project Manager - IT
44	Савчук Максим	+38099999999	max@gmail.com	Software Engineer	IT	Available	Project Manager - IT
42	Романенко Дмитро	+38099999999	dmit@gmail.com	Senior Software Engineer	IT	Available	Project Manager - IT
21	Іванченко Андрій	+38099999999	andrii@gmail.com	Software Engineer	IT	Available	IT Manager - IT
33	Ткаченко Юлія	+38099999999	yu@gmail.com	Senior Software Engineer	IT	Available	Project Manager - IT
16	Васильченко Максим	+38099999999	max@gmail.com	Software Engineer	IT	Available	Project Manager - IT
19	Соловйов Максим	+38099999999	max@gmail.com	Senior Software Engineer	IT	Available	Project Manager - IT
17	Іванченко Андрій	+38099999999	andrii@gmail.com	Software Engineer	IT	Available	Project Manager - IT
18	Григорук Дмитро	+38099999999	dmit@gmail.com	Senior Software Engineer	IT	Available	IT Manager - IT
19	Ткаченко Юлія	+38099999999	yu@gmail.com	Senior Software Engineer	IT	Available	Project Manager - IT
14	Андрійченко Максим	+38099999999	max@gmail.com	Software Engineer	IT	Available	Project Manager - IT
12	Михайленко Дмитро	+38099999999	dmit@gmail.com	Senior Software Engineer	IT	Available	Project Manager - IT

Рисунок 3.20 - Загальний вигляд CRM-системи після завершення розробки

На рис. 3.20 представлено загальний вигляд CRM-системи після завершення розробки. Система містить основні модулі для роботи з кандидатами, вакансіями, задачами та процесом рекрутингу.

3.6 Перспективи розвитку системи

Незважаючи на реалізацію основного функціоналу CRM-системи, програмний продукт може бути додатково вдосконалений у майбутньому.

На даний момент система працює як локальний десктопний додаток та використовує базу даних SQLite. Це є зручним рішенням для невеликих рекрутингових агенцій, проте для великих компаній може виникнути потреба у використанні серверної бази даних та багатокористувацького режиму роботи.

Також у майбутньому систему можна розширити такими можливостями:

- інтеграція з соціальною мережею LinkedIn;
- автоматичний імпорт резюме;
- підбір кандидатів з використанням штучного інтелекта;
- система повідомлень;
- хмарне збереження даних;
- веб-версія CRM-системи.

Окрему увагу можна приділити покращенню системи аналітики та статистики. Це дозволить формувати детальні звіти про процес рекрутингу.

Незважаючи на можливість подальшого розвитку, поточна версія CRM-системи вже забезпечує автоматизацію основних процесів рекрутингової агенції.

Розроблену CRM-систему планується передати на тестування та практичну експлуатацію в існуючу рекрутингову агенцію м. Вінниці.

ВИСНОВКИ

В результаті виконання бакалаврської роботи було розроблено CRM-систему для рекрутингової агенції, яка дозволяє автоматизувати основні процеси роботи з кандидатами, вакансіями та задачами.

Під час виконання роботи було проаналізовано особливості роботи рекрутингових агенцій та визначено основні проблеми ручного обліку кандидатів і вакансій. Було встановлено, що використання звичайних таблиць та ручного ведення інформації ускладнює процес рекрутингу та може призводити до втрати даних.

Також було проведено аналіз сучасних CRM-систем для рекрутингу, зокрема Zoho Recruit, CleverStaff, Hurma System та інших рішень. В результаті аналізу було визначено їх основні переваги та недоліки, а також сформовано вимоги до власної CRM-системи.

В процесі роботи було обрано основні засоби та технології розробки. Для створення програмного продукту використано мову програмування Python, бібліотеку PyQt6 для реалізації графічного інтерфейсу та базу даних SQLite для збереження інформації.

В роботі було створено структуру бази даних CRM-системи та реалізований функціонал для роботи з кандидатами, вакансіями, задачами, користувачами та резюме.

Також було реалізовано графічний інтерфейс програми та основний функціонал CRM-системи:

- роботу з кандидатами;
- роботу з вакансіями;
- kanban-дошку;
- систему задач;

- пошук та фільтрацію;
- резервне копіювання бази даних;
- експорт інформації;
- систему авторизації користувачів.

Окрему увагу було приділено реалізації перегляду PDF-резюме та системи швидкого зв'язку з кандидатами через Telegram, Viber та WhatsApp.

В процесі виконання роботи було проведено тестування CRM-системи. Результати тестування показали стабільну роботу програмного забезпечення та коректну взаємодію між модулями системи.

Практичне значення роботи полягає у створенні готової CRM-системи, яка може використовуватись для автоматизації роботи невеликих рекрутингових агенцій. Основна користь створеної системи полягає у тому, що невеликим рекрутинговим агенціям не потрібно нести щомісячні витрати на абонентську плату за користування CRM-системою. Цим вирішена основна проблема невеликих рекрутингових агенцій, які не мають значних операційних фінансових ресурсів та стабільних доходів для щомісячної абонентської плати за користування CRM-системами.

Розроблену CRM-систему планується передати на тестування та практичну експлуатацію в існуючу рекрутингову агенцію м. Вінниці.

У майбутньому CRM-система може бути розширена додатковими можливостями, зокрема такими популярними функціями як інтеграція з мережею LinkedIn, підбором кандидатів за допомогою штучного інтелекту та хмарним збереженням даних.

СПИСОК ВИКОРИТАНИХ ПОСИЛАНЬ

1. HURMA HR. 7 найкращих ATS для рекрутингу на ринку України [Електронний ресурс]. – 2022. – <https://hurma.work/blog/najkrashhi-ats-dlya-rekrutingu-v-ukrayini>
2. PersiaHR. Автоматизація рекрутингу [Електронний ресурс]. – 2021. – <https://persiahr.com/uk>
3. IBM. What is CRM? [Електронний ресурс]. – 2024. – <https://www.ibm.com/topics/crm>
4. HubSpot. What is CRM Software? [Електронний ресурс]. – 2024. – <https://www.hubspot.com/products/crm>
5. Oracle. Database Design Basics [Електронний ресурс]. – 2022. – https://docs.oracle.com/cd/B28359_01/server.111/b28310/tdddg_dbdesign.htm
6. Васильєв О. М. Програмування в Python: теорія і практика: навч. посіб. – Київ: Ліра-К, 2023. – 462 с.
7. Python Software Foundation. Підручник з Python [Електронний ресурс]. – Режим доступу: <https://docs.python.org/uk/3.13/tutorial/index.html>
8. W3Schools UA. Python: підручник онлайн [Електронний ресурс]. – 2020. – <https://w3schoolsua.github.io/python/index.html>
9. Python Software Foundation. Qt for Python (PySide6) Documentation [Електронний ресурс]. – 2024. – <https://doc.qt.io/qtforpython-6/>
10. Riverbank Computing. PyQt6 Reference Guide [Електронний ресурс]. – 2023. <https://www.riverbankcomputing.com/static/Docs/PyQt6/>
11. Qt Documentation. Qt Widgets Module [Електронний ресурс]. – 2024. – <https://doc.qt.io/qt-6/qtwidgets-index.html>
12. Авраменко А. С., Авраменко В. С., Косенюк Г. В. Тестування програмного забезпечення: навч. посібник. – Черкаси: ЧНУ ім. Б. Хмельницького, 2017. – 284 с.

13. Безмалко О. В. та ін. Менеджмент персоналу: навч. посібник. – Київ: МПП «ЛИНО», 2022. – 612 с.
14. Буньке О. С. Автоматизація бізнес-процесів: навч. посіб. до практич. занять. – Київ: КПІ ім. Ігоря Сікорського, 2021. – 39 с.
15. Криворучко С. В. Рекрутинг персоналу: від азів до професійності. – Київ: Гнозис, 2023. – 692 с.
16. Маттес Е. Пришвидшений курс Python: практичний, проєктно-орієнтований вступ до програмування / пер. з англ. – Київ: Старий Лев, 2021. – 800 с.
17. Мащенко О. Ю. Автоматизована система онлайн-моніторингу біржових цін криптовалют на основі мови програмування Python: кваліфікаційна робота. – Київ: ДУІКТ, 2024. – 103 с.
18. Мізюк О. Путівник мовою програмування Python [Електронний ресурс]. – <https://pythonguide.rozh2sch.org.ua>
19. Руденко Д. Чому автоматизація – це не завжди про ефективність бізнесу? – SPEKA.media, 11.12.2023. <https://speka.media/comu-avtomatizaciya-ce-ne-zavzdi-pro-efektivnist-biznesu-i-shho-z-cim-robiti-pk88dp>
20. Школа бізнесу «Нова пошта». Рекрутинг: особливості та етапи підбору персоналу [Електронний ресурс]. – 26.12.2024. – <https://online.novaposhta.education/blog/rekruting-osoblivosti-ta-etapi-pidboru-personalu>
21. Brander Studio. MVC: шаблон проектування архітектури додатку [Електронний ресурс]. – 2023. – <https://brander.ua/technologies/mvc>
22. ITVDN. SQLAlchemy ORM: зручна робота з базами даних на Python – 2022. <https://itvdn.com/ua/channel/video/sqlalchemy>

23. People First Club. Як створити ефективну рекрутинг-стратегію? 10 кроків до найму мрії [Електронний ресурс]. – 25.04.2024. <https://peoplefirst.club/media/yak-stvoryty-efektyvnu-rekrutynh-stratehiu>
24. Python Software Foundation. Модуль sqlite3 (документація Python) 2024. <https://docs.python.org/uk/3.13/library/sqlite3.html>
25. SQLAlchemy. Інструментарій SQL та об'єктно-реляційне відображення для Python [Електронний ресурс]. – Wikipedia (укр.), 2024. – Режим доступу: <https://uk.wikipedia.org/wiki/SQLAlchemy>
26. SQLite. Полегшена реляційна система керування базами даних [Електронний ресурс]. – Wikipedia (укр.), 2024. – Режим доступу: <https://uk.wikipedia.org/wiki/SQLite>
27. W3Schools UA. SQL: підручник онлайн [Електронний ресурс]. – 2020. – <https://w3schoolsua.github.io/sql/index.html>
28. Real Python. Python GUI Programming With PyQt [Електронний ресурс]. – 2024. – <https://realpython.com/python-pyqt-gui-calculator/>
29. GeeksforGeeks. Introduction to PyQt5 [Електронний ресурс]. – 2023. – <https://www.geeksforgeeks.org/python-introduction-to-pyqt5/>
30. Atlassian. Kanban methodology overview [Електронний ресурс]. – 2023. – <https://www.atlassian.com/agile/kanban>
31. Microsoft. SQL documentation [Електронний ресурс]. – 2024. – <https://learn.microsoft.com/en-us/sql/>
32. PostgreSQL Global Development Group. Database design tutorial [Електронний ресурс]. – 2023. – <https://www.postgresql.org/docs/current/tutorial.html>
33. JetBrains. PyCharm documentation [Електронний ресурс]. – 2024. – <https://www.jetbrains.com/help/pycharm/>
34. Stack Overflow. Python and PyQt questions [Електронний ресурс]. – 2024. – <https://stackoverflow.com/questions/tagged/pyqt>

35. GitHub. Python projects examples [Электронный ресурс]. – 2024. – <https://github.com/topics/python-project>
36. Django Software Foundation. Django documentation [Электронный ресурс]/ – 2024. – <https://docs.djangoproject.com/en/stable/>
37. OpenPyXL Documentation. Working with Excel files in Python [Электронный ресурс]. – 2024. – <https://openpyxl.readthedocs.io/>
38. PyInstaller Documentation [Электронный ресурс]. – 2024. – <https://pyinstaller.org/en/stable/>
39. Mozilla Developer Network. MVC architecture pattern [Электронный ресурс]. – 2023. – <https://developer.mozilla.org/en-US/docs/Glossary/MVC>
40. SQLite Documentation [Электронный ресурс]. – 2024. – <https://www.sqlite.org/docs.html>
41. Python Packaging User Guide [Электронный ресурс]. – 2024. – <https://packaging.python.org/>
42. DigitalOcean. How To Use SQLite with Python 3 [Электронный ресурс]. – 2023. – <https://www.digitalocean.com/community/tutorials/how-to-use-the-sqlite3-module-in-python-3>
43. Simplilearn. What is Recruitment Management System? [Электронный ресурс]. – 2023. – <https://www.simplilearn.com/recruitment-management-system-article>
44. Indeed for Employers. Recruitment process overview [Электронный ресурс]. – 2024. – <https://www.indeed.com/hire/c/info/recruitment-process>
45. Lucidchart. UML Diagram Tutorial [Электронный ресурс]. – 2024. – <https://www.lucidchart.com/pages/uml-diagram>

The background of the page features a large, faint, circular seal of Donets National University. The seal's outer ring contains the text "ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛА СТУСА" in Ukrainian and "DONETSK NATIONAL UNIVERSITY" in English. The central shield of the seal is divided into four quadrants: the top-left contains a stylized rose, the top-right contains an hourglass, the bottom-left contains a compass rose, and the bottom-right contains a star. A banner across the top of the shield reads "NOMEN EST OMEN" and a banner across the bottom reads "ІМ'Я ЗОБОВ'ЯЗУЄ".

ДОДАТКИ

ДОДАТОК А

Фрагмент коду головного вікна CRM-системи

```

class MainWindow(QMainWindow):
    def __init__(self, db_manager, auth_manager):
        super().__init__()

        self.db_manager = db_manager
        self.auth_manager = auth_manager

        self.candidate_manager = CandidateManager(db_manager,
auth_manager)
        self.vacancy_manager = VacancyManager(db_manager,
auth_manager)
        self.application_manager = ApplicationManager(db_manager,
auth_manager)
        self.task_manager = TaskManager(db_manager, auth_manager)

        self.setWindowTitle("CRM-система рекрутингової агенції")
        self.setMinimumSize(1200, 800)

        self.init_ui()
        self.create_menu()
        self.create_toolbar()
        self.load_dashboard_data()

    def init_ui(self):
        self.central_widget = QWidget()
        self.main_layout = QVBoxLayout()

        self.tabs = QTabWidget()

        self.dashboard_tab = DashboardWidget(
            self.candidate_manager,
            self.vacancy_manager,
            self.task_manager
        )

        self.candidates_tab = CandidateList(
            self.candidate_manager,
            self.application_manager
        )

        self.vacancies_tab = VacancyList(
            self.vacancy_manager,
            self.application_manager,
            self.candidate_manager
        )

```

```

self.tasks_tab = TaskList(
    self.task_manager,
    self.candidate_manager,
    self.vacancy_manager
)

self.tabs.addTab(self.dashboard_tab, "Головна")
self.tabs.addTab(self.candidates_tab, "Кандидати")
self.tabs.addTab(self.vacancies_tab, "Вакансії")
self.tabs.addTab(self.tasks_tab, "Задачі")

self.main_layout.addWidget(self.tabs)
self.central_widget.setLayout(self.main_layout)

self.setCentralWidget(self.central_widget)

def create_menu(self):
    menu_bar = self.menuBar()

    file_menu = menu_bar.addMenu("Файл")
    users_menu = menu_bar.addMenu("Користувачі")
    help_menu = menu_bar.addMenu("Довідка")

    export_action = QAction("Експорт в Excel", self)
    export_action.triggered.connect(self.export_to_excel)

    backup_action = QAction("Створити backup", self)
    backup_action.triggered.connect(self.create_backup)

    exit_action = QAction("Вихід", self)
    exit_action.triggered.connect(self.close)

    file_menu.addAction(export_action)
    file_menu.addAction(backup_action)
    file_menu.addSeparator()
    file_menu.addAction(exit_action)

    users_action = QAction("Керування користувачами", self)
    users_action.triggered.connect(self.open_users_dialog)
    users_menu.addAction(users_action)

    about_action = QAction("Про програму", self)
    about_action.triggered.connect(self.show_about)
    help_menu.addAction(about_action)

def create_toolbar(self):
    toolbar = QToolBar("Головна панель")
    self.addToolBar(toolbar)

```

```
add_candidate_action = QAction("Додати кандидата", self)
add_candidate_action.triggered.connect(self.add_candidate)

add_vacancy_action = QAction("Додати вакансію", self)
add_vacancy_action.triggered.connect(self.add_vacancy)

refresh_action = QAction("Оновити", self)
refresh_action.triggered.connect(self.refresh_all)

toolbar.addAction(add_candidate_action)
toolbar.addAction(add_vacancy_action)
toolbar.addAction(refresh_action)

def refresh_all(self):
    self.dashboard_tab.refresh()
    self.candidates_tab.refresh()
    self.vacancies_tab.refresh()
    self.tasks_tab.refresh()
```

ДОДАТОК Б

Фрагмент коду форми кандидата

```
class CandidateForm(QDialog):
    def __init__(self, candidate_manager, candidate_id=None,
parent=None):
        super().__init__(parent)

        self.candidate_manager = candidate_manager
        self.candidate_id = candidate_id
        self.resume_paths = []
        self.current_resume_index = 0

        self.candidate_data = None

        if candidate_id:
            self.candidate_data =
self.candidate_manager.get_candidate_by_id(candidate_id)

        self.setWindowTitle("Форма кандидата")
        self.setMinimumSize(1000, 700)

        self.init_ui()
        self.fill_form_if_editing()

    def init_ui(self):
        main_layout = QHBoxLayout()

        left_layout = QVBoxLayout()
        form_layout = QFormLayout()

        self.name_edit = QLineEdit()
        self.phone_edit = QLineEdit()
        self.email_edit = QLineEdit()
        self.skills_edit = QLineEdit()
        self.source_edit = QLineEdit()

        self.experience_edit = QTextEdit()
        self.education_edit = QTextEdit()
        self.notes_edit = QTextEdit()

        form_layout.addRow("ПІБ:", self.name_edit)
        form_layout.addRow("Телефон:", self.phone_edit)

        messenger_layout = QHBoxLayout()

        self.telegram_button = QPushButton("Telegram")
        self.viber_button = QPushButton("Viber")
```

```

self.whatsapp_button = QPushButton("WhatsApp")

self.telegram_button.clicked.connect(self.open_telegram)
self.viber_button.clicked.connect(self.open_viber)
self.whatsapp_button.clicked.connect(self.open_whatsapp)

messenger_layout.addWidget(self.telegram_button)
messenger_layout.addWidget(self.viber_button)
messenger_layout.addWidget(self.whatsapp_button)

form_layout.addRow("Швидкий зв'язок:", messenger_layout)

form_layout.addRow("Email:", self.email_edit)
form_layout.addRow("Навички:", self.skills_edit)
form_layout.addRow("Джерело:", self.source_edit)
form_layout.addRow("Досвід:", self.experience_edit)
form_layout.addRow("Освіта:", self.education_edit)
form_layout.addRow("Нотатки:", self.notes_edit)

left_layout.addLayout(form_layout)

buttons_layout = QHBoxLayout()

self.add_resume_button = QPushButton("Додати резюме")
self.save_button = QPushButton("Зберегти")
self.cancel_button = QPushButton("Скасувати")

self.add_resume_button.clicked.connect(self.add_resume)
self.save_button.clicked.connect(self.save_candidate)
self.cancel_button.clicked.connect(self.reject)

buttons_layout.addWidget(self.add_resume_button)
buttons_layout.addWidget(self.save_button)
buttons_layout.addWidget(self.cancel_button)

left_layout.addLayout(buttons_layout)

self.resume_preview = PdfPreviewWidget()

main_layout.addLayout(left_layout, 1)
main_layout.addWidget(self.resume_preview, 1)

self.setLayout(main_layout)

def fill_form_if_editing(self):
    if not self.candidate_data:
        return

```

```

        self.name_edit.setText(self.candidate_data.get("full_name")
or "")
        self.phone_edit.setText(self.candidate_data.get("phone") or
        "")
        self.email_edit.setText(self.candidate_data.get("email") or
        "")
        self.skills_edit.setText(self.candidate_data.get("skills") or
        "")
        self.source_edit.setText(self.candidate_data.get("source") or
        "")
        self.experience_edit.setText(self.candidate_data.get("experience") or
        "")
        self.education_edit.setText(self.candidate_data.get("education") or
        "")
        self.notes_edit.setText(self.candidate_data.get("notes") or
        "")

        resume_path = self.candidate_data.get("resume_path")

        if resume_path and os.path.exists(resume_path):
            self.resume_paths = [resume_path]
            self.current_resume_index = 0
            self.resume_preview.load_pdf(resume_path)

    def add_resume(self):
        file_path, _ = QFileDialog.getOpenFileName(
            self,
            "Виберіть PDF-резюме",
            "",
            "PDF Files (*.pdf)"
        )

        if file_path:
            self.resume_paths.append(file_path)
            self.current_resume_index = len(self.resume_paths) - 1
            self.resume_preview.load_pdf(file_path)

    def save_candidate(self):
        full_name = self.name_edit.text().strip()

        if not full_name:
            QMessageBox.warning(
                self,
                "Помилка",
                "Поле ПІВ є обов'язковим."
            )
        return

```

```

phone = self.phone_edit.text().strip()
email = self.email_edit.text().strip()
skills = self.skills_edit.text().strip()
source = self.source_edit.text().strip()
experience = self.experience_edit.toPlainText().strip()
education = self.education_edit.toPlainText().strip()
notes = self.notes_edit.toPlainText().strip()

resume_path = None

if self.resume_paths:
    resume_path =
self.resume_paths[self.current_resume_index]

if self.candidate_id:
    self.candidate_manager.update_candidate(
        self.candidate_id,
        full_name=full_name,
        phone=phone,
        email=email,
        skills=skills,
        experience=experience,
        education=education,
        source=source,
        resume_path=resume_path,
        notes=notes
    )
else:
    self.candidate_id = self.candidate_manager.add_candidate(
        full_name=full_name,
        phone=phone,
        email=email,
        skills=skills,
        experience=experience,
        education=education,
        source=source,
        resume_path=resume_path,
        notes=notes
    )

self.accept()

def open_telegram(self):
    phone = self.phone_edit.text().strip()

    if phone:
        url = f"https://t.me/{phone}"
        QDesktopServices.openUrl(QUrl(url))

```

```
def open_viber(self):  
    phone = self.phone_edit.text().strip()  
  
    if phone:  
        url = f"viber://chat?number={phone}"  
        QDesktopServices.openUrl(QUrl(url))  
  
def open_whatsapp(self):  
    phone = self.phone_edit.text().strip()  
  
    if phone:  
        url = f"https://wa.me/{phone}"  
        QDesktopServices.openUrl(QUrl(url))
```