

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій, обліку та фінансів  
Кафедра комп'ютерних технологій  
і моделювання систем

Кваліфікаційна робота  
на правах рукопису

Сергійчук Максим Олександрович

УДК 004.6:004.75

## **КВАЛІФІКАЦІЙНА РОБОТА**

### **ІНФОРМАЦІЙНА СИСТЕМА ПРОВЕДЕННЯ ТЕХНІЧНИХ ІНТЕРВ'Ю**

126 «Інформаційні системи та технології»

Подається на здобуття освітнього ступеня бакалавр

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

---

(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи  
Молодецька Катерина Валеріївна,  
доктор технічних наук, професор

Житомир – 2024

**Висновок кафедри** \_\_\_\_\_  
за результатами попереднього захисту: \_\_\_\_\_

Протокол засідання кафедри \_\_\_\_\_  
№ \_\_\_ від «\_\_\_\_\_» \_\_\_\_\_ 20\_\_ р.

Завідувач кафедри \_\_\_\_\_  
\_\_\_\_\_  
(науковий ступінь, вчене звання) (підпис) (прізвище, ім'я, по батькові)  
«\_\_\_\_\_» \_\_\_\_\_ 20\_\_ р.

### **Результати захисту кваліфікаційної роботи**

Здобувач вищої освіти \_\_\_\_\_ захистив (ла)  
(прізвище, ім'я, по батькові)

кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою \_\_\_\_\_

за шкалою ECTS \_\_\_\_\_

за національною шкалою \_\_\_\_\_

Секретар ЕК

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

## АНОТАЦІЯ

Сергійчук М. О. Інформаційна система проведення технічних інтерв'ю – Кваліфікаційна робота на правах рукопису.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавра за спеціальністю 126 – Інформаційні системи та технології. - Поліський національний університет, Житомир, 2024.

У роботі досліджено реалізації інформаційної системи проведення технічних інтерв'ю. Для досягнення даної мети у теоретичній частині роботи було проведено дослідження особливостей реалізації інформаційної системи проведення технічних інтерв'ю, а саме: наведено існуючі підходи та інструменти та проведено аналіз вимог до інформаційної системи. У практичній частині було розроблено узагальнену структурну схему системи, також спроектовано базу даних для системи та проведено реалізацію математичної моделі для даної кваліфікаційної роботи. Було реалізовано інформаційну систему, для цього створено інтерфейс та основні функції програми. Систему було протестовано та наведено керівництво користувачу.

Ключові слова: інформаційна система, база даних, співбесіда, інтерв'юер, кандидат, платформа, технічна посада, тестування.

## SUMMARY

Serhiichuk M.O. Information system for conducting technical interviews - Qualification work on the rights of a manuscript.

Qualification work for obtaining a bachelor's degree in specialty 126 – Information systems and technologies. – Polissya National University, Zhytomyr, 2024.

The work investigates the implementation of the information system for conducting technical interviews. To achieve this goal, the theoretical part of the work conducted a study of the features of implementing an information system for conducting technical interviews, namely: existing approaches and tools were described, and an analysis of the requirements for the information system was conducted. In the practical part, a generalized structural scheme of the system was developed, a database for the system was designed, and a mathematical model for this qualification work was implemented. An information system was implemented by creating an interface and the main program functions. The system has been tested and a user manual has been provided.

Keywords: information system, database, interview, interviewer, candidate, platform, technical position, testing.

## ЗМІСТ

<b>ЗМІСТ</b> .....	5
<b>ВСТУП</b> .....	7
<b>1 АНАЛІЗ СУЧАСНИХ ІНФОРМАЦІЙНИХ СИСТЕМ ПРОВЕДЕННЯ ТЕХНІЧНИХ ІНТЕРВ'Ю</b> .....	9
<b>1.1 Огляд існуючих підходів та інструментів</b> .....	9
<b>1.2 Аналіз вимог до інформаційної системи</b> .....	10
<b>Висновки до першого розділу</b> .....	12
<b>2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПРОВЕДЕННЯ ТЕХНІЧНИХ ІНТЕРВ'Ю</b> .....	13
<b>2.1 Узагальнена структурна схема системи</b> .....	13
<b>2.2 Моделювання ІС CASE засобами</b> .....	14
<b>2.3 Проектування бази даних системи</b> .....	18
<b>2.4 Реалізація математичної моделі системи</b> .....	21
<b>Висновки до другого розділу</b> .....	23
<b>3 РОЗРОБКА І РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПРОВЕДЕННЯ ТЕХНІЧНИХ ІНТЕРВ'Ю</b> .....	24
<b>3.1 Розроблення інтерфейсу та основних функцій системи</b> .....	24
<b>3.2 Тестування системи і керівництво користувачу</b> .....	27
<b>Висновки до третього розділу</b> .....	29
<b>ВИСНОВКИ ТА ПРОПОЗИЦІЇ</b> .....	30
<b>ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ</b> .....	31
<b>ДОДАТОК 1</b> .....	33

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

IDEF3 – Integrated DEFinition for Process Description Capture Method;

HR – Human Resources;

ТЗ – Технічне Завдання;

ER – Entity Relationship Diagram;

DFD – Data Flow Diagrams;

БД – База даних;

exe – розширення виконуючого файлу;

SQL – діалогова мова програмування для здійснення запиту і внесення змін до бази даних;

СУБД – система управління базами даних;

MySQL – вільна система керування реляційними базами даних, яка була розроблена компанією «ТсХ» для підвищення швидкодії обробки великих баз даних;

UML – Unified Modeling Language.

## ВСТУП

У сучасному світі майже в кожному великому місті є ІТ-компанії. В кожній компанії постійно оновлюються робітники. Для того щоб прийняти кандидата на технічну посаду проводяться інтерв'ю. Щоб провести технічне інтерв'ю зазвичай витрачається багато часу і сил як роботодавців так і кандидатів.

Технічні інтерв'ю починаються з того що компанії потрібен новий робітник на технічну посаду, на основі чого створюється вакансія. Після відгуку на вакансію, переглядається резюме кандидата і випадку його відповідності потребам, організовується HR-інтерв'ю, або одразу технічне інтерв'ю. На співбесіді кандидата опитують, дають технічні завдання і на основі його результатів виносяться рішення про прийняття нового працівника[1].

Актуальність роботи полягає у тому що, кожна компанія постійно оновлює своїх спеціалістів за різних обставин, яких може бути безліч. Існувати без робочої сили компанії не в змозі, а для залучення нових співробітників, необхідно проводити інтерв'ю, щоб роботодавець був впевнений у кваліфікованості нових працівників.

Метою роботи є створення інформаційної системи для збільшення автоматизації та покращення процесу проведення технічного інтерв'ю, що дозволить роботодавцю економити час на підготовку питань та завдань і збільшити кількість опитаних кандидатів.

Основні завдання цієї роботи включають:

- аналіз інформаційних потреб та вимог від роботодавця під час проведення технічного інтерв'ю;
- розроблення інформаційної системи проведення технічного інтерв'ю;
- програмна реалізація моделі інформаційної системи проведення технічного інтерв'ю;

Предметом дослідження є методи та моделі розроблення інформаційної системи проведення технічного інтерв'ю.

Об'єктом є процес розроблення інформаційної системи проведення технічного інтерв'ю.

Дана робота збільшить швидкість та простоту проведення технічних інтерв'ю. Робота дасть змогу прискорити пошук необхідного спеціаліста на конкретну посаду та зменшити кількість потрачених зусиль для цього.

Під час роботи над курсовою роботою використовувались аналітичні, математичні, графічні, проектування та інші методи дослідження.

Перелік публікацій автора за темою дослідження:

1. Сергійчук М.О. ІНФОРМАЦІЙНА СИСТЕМА ПРОВЕДЕННЯ ТЕХНІЧНИХ ІНТЕРВ'Ю: матеріали міжфакультетної студентської науково-практичної конференції, м. Житомир, 14 листопада 2023 р. Житомир: ПНУ, 2023. С. 37-39
2. Сергійчук М.О. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПРОВЕДЕННЯ ТЕХНІЧНИХ ІНТЕРВ'Ю: матеріали всеукраїнської студентської науково-практичної конференції, м. Житомир, 10 квітня 2024 р. Житомир: ПНУ, 2024. С. 23



# 1 АНАЛІЗ СУЧАСНИХ ІНФОРМАЦІЙНИХ СИСТЕМ ПРОВЕДЕННЯ ТЕХНІЧНИХ ІНТЕРВ'Ю

## 1.1 Огляд існуючих підходів та інструментів

На даний момент часу в інтернеті існує досить багато платформ пов'язаних з даною тематикою, проте основна мета більшості з них є підготовка кандидата до проходження таких інтерв'ю, а не саме проведення співбесіди. Вони націлені щоб навчати та розвивати кандидатів шляхом розв'язку різних завдань та ознайомленням з тими питаннями які їм можуть поставити інтерв'юер. Одним із аналогів є платформа «LeetCode». На ній можна підготуватися, а також провести імітаційне інтерв'ю. Також там є досить розвинута спільнота, де люди діляться досвідом і ставлять питання один одному. «Codility» ще одна платформа з даної тематики, яка на відміну від попередньої більш спрямована для реальних інтерв'ю. Платформа дозволяє роботодавцеві в реальному часі спостерігати за виконанням завдань і оцінювати їх. Порівняння даних платформ наведено в таблиці 1.1

Таблиця 1.1 – Порівняння існуючих систем проведення технічних інтерв'ю

Функція/характеристика	Codility	LeetCode	Pramp
Редактор програмного коду	+	+	+
Візуалізація результатів	+	-	-
Проведення тестового опитування	+	+	-
Керування кандидатами та співбесідами	-	-	-
Спрощений інтерфейс користувача	-	+	+
Можливість легко керувати інформацією про кандидатів	+	-	-
Автоматичне оцінювання відповідей	+	+	-
Завантаження додатку на ПК	-	-	-

В даній таблиці порівнюються три платформи, кожна з яких підтримує свій окремий функціонал. Якщо платформа може підтримувати будь-який з перелічених функціоналів, то в таблиці під нею поставлено знак «+», в іншому випадку – «-».

Основним недоліком цих платформ є відсутність автоматизованого підходу, оскільки роботодавцеві користуючись ними знадобиться стільки ж часу для

знаходження потрібного кандидату що і при звичайному інтерв'ю, але можна зазначити, що це процес відбору буде легший, а допустити помилку у виборі буде тяжче. Також потрібно додати що майже всі платформи являються веб-додатками, тому не вистачає програми яку можна було б завантажити на пристрій і користуватись без виходу в інтернет.

Створення автоматизованої системи для покращення процесу проведення технічного інтерв'ю може сприяти стандартизації оцінки кандидатів. Це означає, що всі кандидати будуть оцінюватися за однаковими критеріями, що спрощує процес вибору найкращого серед них. Крім того, така система може забезпечити більш об'єктивну оцінку, оскільки вона буде базуватися на конкретних показниках та результатах виконання завдань, а не на суб'єктивних враженнях інтерв'юера.

Враховуючи вище сказане, можна зазначити що створення автоматизованої системи для покращення процесу проведення технічного інтерв'ю дозволить роботодавцю економити час на підготовку питань та завдань і збільшити кількість опитаних кандидатів.

## **1.2 Аналіз вимог до інформаційної системи**

Визначаючи необхідність задоволення інформаційних потреб користувачів для вирішення завдань у різних сферах життя, відбувається прискорення розвитку технологій і загального прогресу. Використання інформаційної системи спростить та прискорить процес проведення технічних інтерв'ю, що може бути особливо важливим у випадку масштабних процесів. Інформаційна система може надати стандартизовані інструменти для оцінки технічних навичок кандидатів, допомагаючи відбирати кандидатів, які найкраще відповідають вимогам посади.

Для прийняття кандидата на технічну посаду проводяться інтерв'ю, на що зазвичай витрачається багато часу і сил як роботодавців так і кандидатів. Тому дана система спрямована на полегшення такого процесу. Інформаційна система проведення технічного інтерв'ю – це комплекс програмних засобів, який допомагає автоматизувати та оптимізувати процес технічного відбору кандидатів на роботу.

Вона включає в себе інструменти для створення та проведення технічних інтерв'ю, оцінки навичок кандидатів, збору та аналізу даних, а також забезпечує централізоване управління інформацією про потенційних співробітників[2]. Необхідність розробки інформаційної системи для проведення технічних інтерв'ю виникає в контексті високого попиту на ефективні та об'єктивні методи відбору та оцінки технічних навичок кандидатів під час процесу найму.

Для реалізації інформаційної системи проведення технічних інтерв'ю було визначено вимоги до функціоналу системи, які наведені в таблиці 1.2.

Таблиця 1.2 – Вимоги до інформаційної системи

Вимога	Опис	Приоритет
Збереження завдань та результатів	Всі результати та завдання повинні зберігатися в базу даних системи.	Високий
Редактор коду	Для перевірки практичних навичок та виконання ТЗ повинен бути наявний редактор коду	Високий
Автоматичне оцінювання відповідей	Система повинна автоматично оцінювати відповіді кандидатів на технічні питання та завдання.	Високий
Збір даних	Система повинна збирати дані про процес інтерв'ю та інформацію про кандидатів для подальшого аналізу та звітності.	Середній
Масштабованість	Система повинна бути здатна обслуговувати зростаючу кількість користувачів та кандидатів без втрати продуктивності.	Середній
Інтуїтивно зрозумілий інтерфейс	Інтерфейс системи повинен бути зрозумілим і простим у використанні для користувачів з різним рівнем технічних знань.	Високий
Перевірка завдань інтерв'юером	Система має надавати можливість змінити результат автоматичної перевірки для інтерв'юерів за потреби.	Низький

В першу чергу, система має включати модуль для створення та збереження шаблонів технічних завдань та питань. Також важливим є наявність інструментів для проведення інтерв'ю, що включає редагування коду, та систему запису

відповідей і результатів кандидата. Окрім того, повинен бути реалізований функціонал автоматичної оцінки відповідей, що сприяє об'єктивності в оцінюванні кандидатів і зменшенню впливу людського фактору.

Для коректної роботи системи було розроблено базу даних для взаємодії з додатком, яка зберігає всю потрібну інформацію системи та користувачів, а саме: завдання, результати, користувачі. Взаємодія бази даних та кінцевих пристроїв наведена на рисунку 1.1.

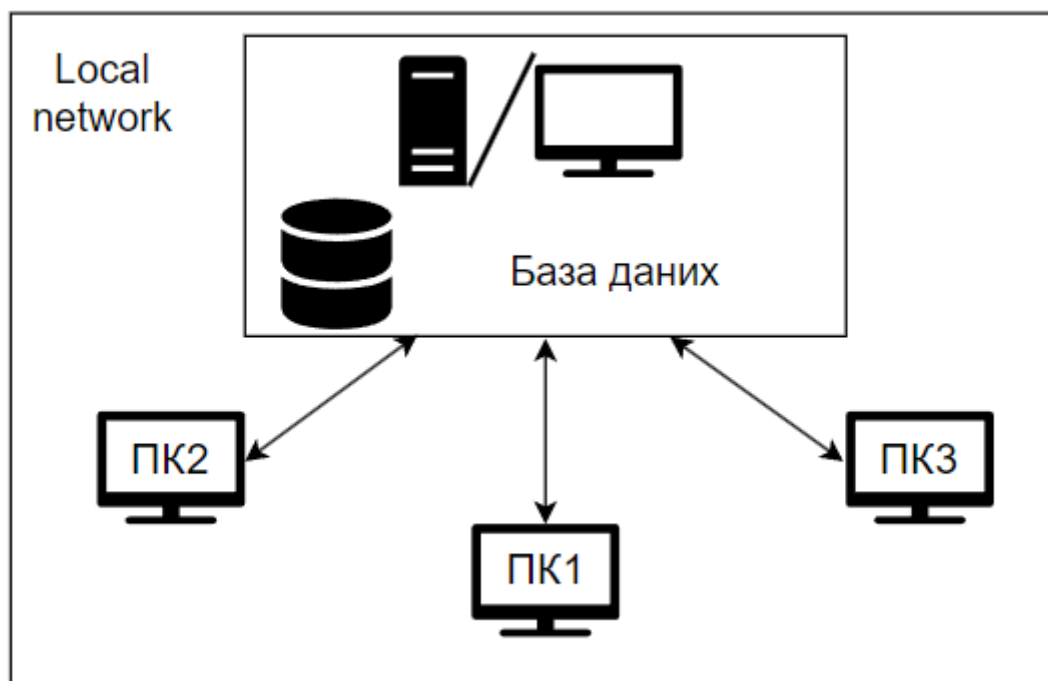


Рисунок 1.1 – Взаємодія з базою даних в локальній мережі

### Висновки до першого розділу

В даному розділі було описано аналіз вимог до інформаційної системи та визначено потреби предметної області. Визначені основні вимоги було розподілено за пріорітеми. Основною потребою є прискорення та покращення процесу відбору кандидатів на технічні посади, шляхом автоматизації. Було наведено існуючі приклади інформаційних систем для проведення технічних інтерв'ю (Codility, LeetCode, Pramp), їх переваги та недоліки.

## 2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПРОВЕДЕННЯ ТЕХНІЧНИХ ІНТЕРВ'Ю

### 2.1 Узагальнена структурна схема системи

У цьому розділі представлено узагальнену структурну схему інформаційної системи. Схема створена для того, щоб наглядно продемонструвати архітектуру системи, її ключові компоненти та взаємодію між ними. Дана схема зображена на рисунку 2.1.

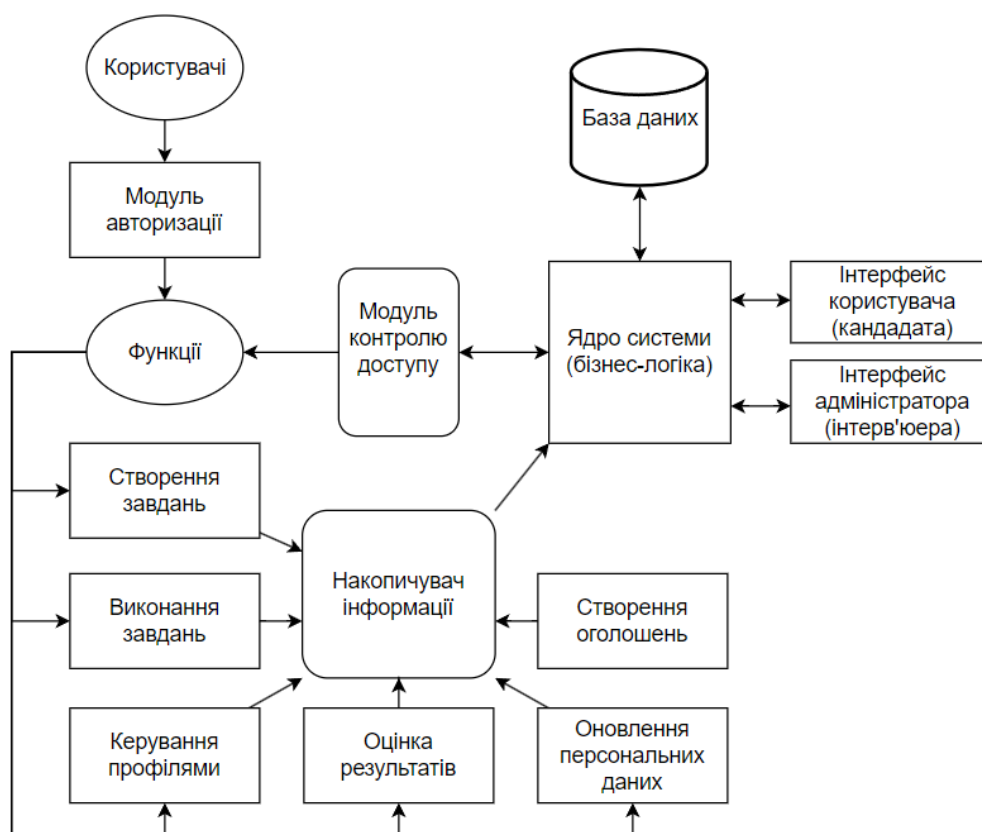


Рисунок 2.1 – Структурна схема застосунку

З малюнку видно, що користувачі системи мають змогу взаємодіяти з системою через два основних інтерфейси: інтерфейс кандидата та інтерфейс адміністратора (інтерв'юера). Ці інтерфейси забезпечують доступ до різних функціональних можливостей залежно від ролі та прав доступу користувача.

Модуль авторизації відіграє важливу роль у визначенні ідентичності користувачів та наданні їм відповідного рівня доступу до функцій системи. Цей модуль працює тісно з модулем контролю доступу, який забезпечує дотримання політик безпеки та обмежень на різні частини системи.

Функціональний блок системи: дозволяє інтерв'юерам формувати завдання для технічних інтерв'ю, надає кандидатам інтерфейс для виконання завдань під час інтерв'ю, включає управління даними профілів користувачів, автоматизує процес аналізу та оцінювання відповідей кандидатів, дозволяє оновлювати свою персональну інформацію, надає інструмент для публікації оголошень в системі.

Ядро системи (бізнес-логіка) є ключовим елементом, який координує всі процеси в системі, включаючи обробку запитів від користувачів, виконання бізнес-правил, інтеграцію з базою даних та управління даними користувачів.

База даних є фундаментальною складовою системи, де зберігаються всі дані, необхідні для роботи системи. Вона забезпечує швидкий доступ до даних та їх цілісність.

## **2.2 Моделювання ІС CASE засобами**

На рисунку 2.2 зображено IDEF3 діаграму, яка демонструє процеси при проведенні технічного інтерв'ю. Загалом вона має тринадцять блоків та п'ять розгалужень, одне з них синхронне «або», всі інші виключні «або». Схема розпочинається з першого блоку «Перевірка резюме», після чого є два варіанти, перший коли кандидат виявляється одразу не відповідає вимогам вакансії і другий, який допускає кандидата до виконання завдань. Кінцевих результатів також два – це, або кандидат успішно пройшов співбесіду, або його відхилили через не задовільний результат роботи чи резюме.

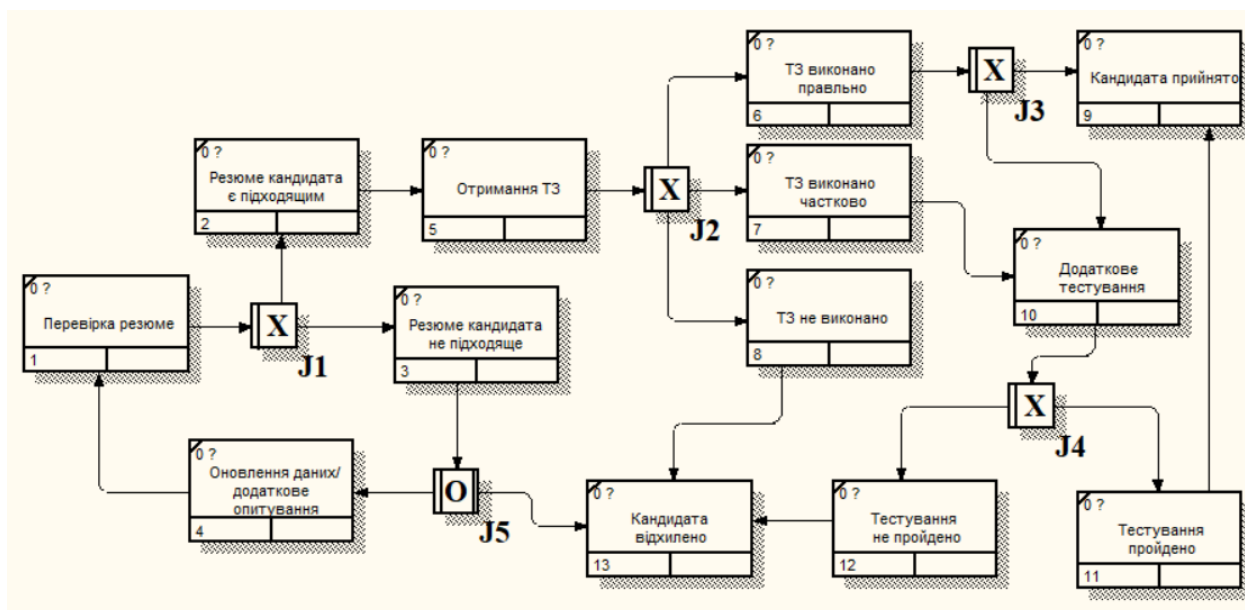


Рисунок 2.2 – Діаграма IDEF3 процесу проведення технічного інтерв'ю

Перший блок необхідний щоб визначити чи є сенс подальшої роботи з кандидатом, чи є в нього потрібні знання для конкретної вакансії, оскільки не можливо провести співбесіду з усіма бажаючими, кожен має спочатку себе зарекомендувати. З цього випливають два наступні блоки «Резюме кандидата є підходящим» і «Резюме кандидата не підходяще», на схемі це зображено розгалуженням виключним «або». Дані блоки говорять самі за себе, їх результат залежить від попереднього блоку.

У випадку неподходящого резюме також можливий варіант для оновлення інформації про себе кандидатом. Кандидат має можливість переглянути вимоги і надати більш коректні дані, або уточнити наявні, що дозволить йому почати співбесіду.

Дія в блоці п'ять «Отримання ТЗ» виконується у випадку схвалення рекрутером резюме. В цьому випадку кандидат отримує технічне завдання по тій спеціальності на яку він претендує. Технічне завдання не являється якимось питанням на яке можна відповісти так, або ні. Є декілька варіантів розвитку подій після його отримання. Якщо кандидат високо кваліфікований, з відмінними навичками і досвідом, то, як правило, він впорюється з завданням (блок 6 «ТЗ виконано правильно»). В протилежному випадку, коли кандидат не розуміється на

завданні і провалює його, вступає в дію 8 блок «ТЗ не виконано». Зазвичай задання для технічних посад не мають єдиного розв'язку, кожен може реалізувати його по своєму, і в це показує блок 7 «ТЗ виконано частково», коли інтерв'юер бачить що кандидат був близький до вирішення, або його реалізація має не належний вигляд, проте необхідний функціонал є наявним і так далі.

У разі часткового виконання ТЗ кандидату пропонують пройти додаткове тестування, щоб пересвідчитись у його навичках. Воно може проходити як і в усній так і в письмовій формі. Також у випадку, коли в інтерв'юера з'являються сумніви щодо кандидата, котрий успішно виконав завдання, його також можуть додатково опитати. Цей процес описує десятий блок «Додаткове тестування». Так само як і з ТЗ, вдале проходження тесту означає отримання посади «Кандидата прийнято» (блок 9), в і іншому випадку, коли тестування не пройдено (блок 12), кандидата відхиляють «Кандидата відхилено» (блок 13), це також є одним із кінцевих результатів даної діаграми IDF3 разом з блоком 9.

Ще одним способом продемонструвати структуру системи є DFD діаграми. Дана діаграма дозволить графічно продемонструвати потоки даних і інформаційній системі. Зазвичай розробники розпочинають роботу над проектом, створюючи діаграму потоків даних на рівні контексту. Ця діаграма вказує на взаємодію між системою та зовнішніми модулями. Після цього вона уточнюється за допомогою деталізації процесів та потоків даних, щоб надати більш докладний огляд розроблюваної системи[3]. Рисунок 2.3 є прикладом DFD створеної за темою роботи.



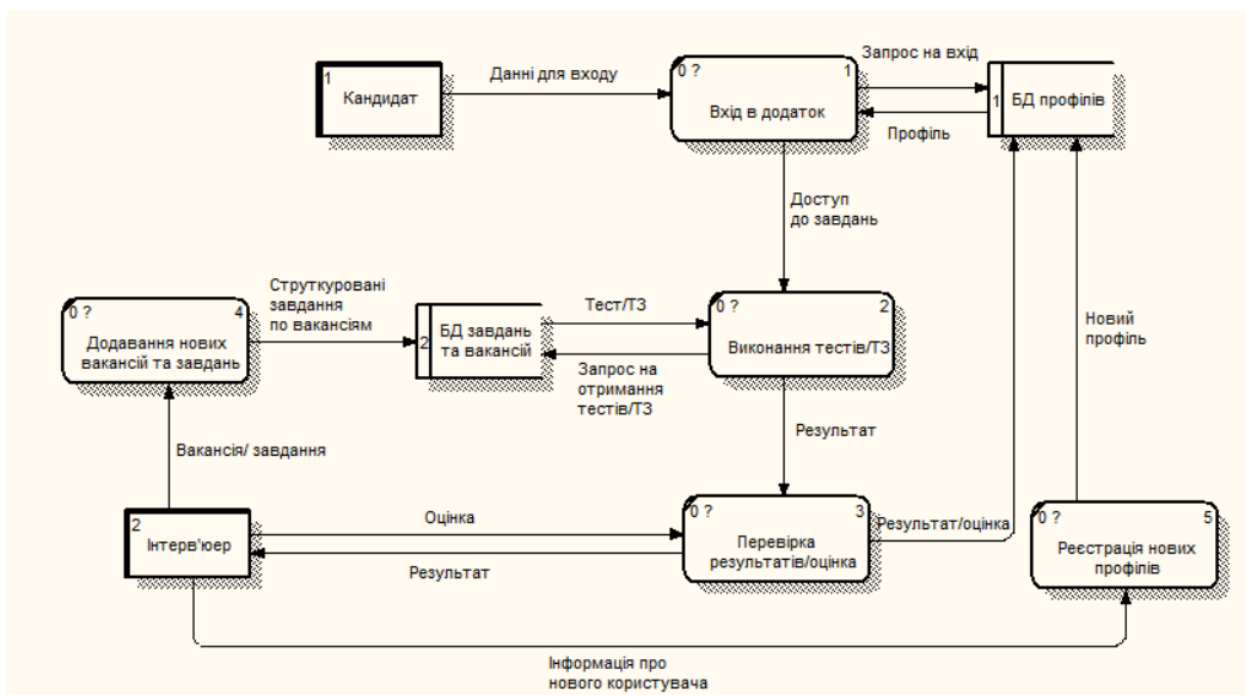


Рисунок 2.3 – DFD діаграма системи

Дана діаграма складається з двох сутностей, п'яти функцій, двох БД та потоків даних між ними. Першою функцією є вхід в додаток, який виконує кандидат для доступу до самої програми. Функція виконується шляхом перевірки введених користувачем даних на вхід з наявними у базі. При виникненні збігу кандидат може почати використовувати додаток, в іншому випадку йому доведеться ввести дані ще раз.

Ще однією сутністю є інтерв'юер, саме він вносить більшість інформації до БД, за винятком, коли система автоматично додає результати кандидатів після виконання завдань. Також однією з його можливостей є створення нових профілів для майбутнього їх використання кандидатами. Інтерв'юер може виправляти і переглядати всі дані пов'язані з кандидатами, а також окремо оцінювати виконані завдання.

Для кращого розуміння роботи застосунку, рисунку 2.4 наведена UML діаграма станів, яка визначає зміну станів об'єкту у часі.

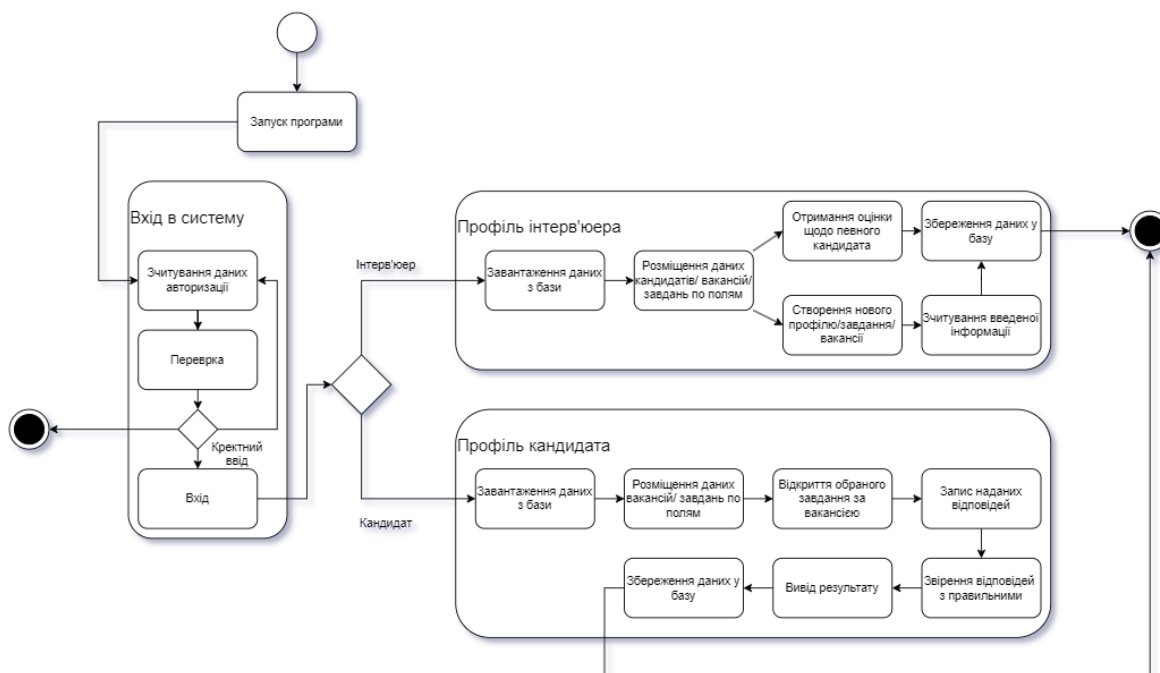


Рисунок 2.4 – UML станів

На даному малюку видно, що точкою входу є запуск програми. Наступним кроком є авторизація з трьома варіантами розвитку подій. Після входу та отриманню ролі, користувачу стає доступний функціонал, що детальніше розписано в розділі 3.1.

### 2.3 Проектування бази даних системи

Одним із важливих аспектів розробки інформаційної системи для проведення технічних інтерв'ю є створення ефективної та надійної бази даних, здатної обробляти, зберігати та ефективно поширювати інформацію. Проектування такої бази даних передбачає розробку схеми, яка відповідає всім технічним вимогам системи та забезпечує високий рівень безпеки та інтеграції з іншими модулями системи. База даних в основному складається з таблиць для зберігання даних про кандидатів, включаючи персональні дані, резюме, результати технічних тестів та співбесід. Також таблиці необхідні для зберігання інформації про вакансії, вимоги до роботи, інтерв'юерів, а також зразків завдань і тестових питань.

Важливим аспектом є розробка системи прав і ролей користувачів для забезпечення різних рівнів доступу до даних. Оптимізація процесу пошуку, сортування та аналізу даних вимагає ретельного планування структури індексування та оптимізації запитів. Це дозволить уникнути затримок у системі та забезпечити швидкий доступ до необхідної інформації.

Підчас проектування баз даних, створені схеми за допомогою ER-моделі, використовують для реалізації схем БД на їх основі. З допомогою даного типу моделей можна виділити ключові сутності та продемонструвати зв'язки, які можуть створюватися між цими сутностями[4].

Схема створеної бази даних наведена на рисунку 2.5 у вигляді ER-моделі. База містить 12 таблиць, а саме: дані для входу, інформація про користувача, файли, загальна інформація, результати тестів, кодових та логічних завдань, вакансії, 3 види завдань.

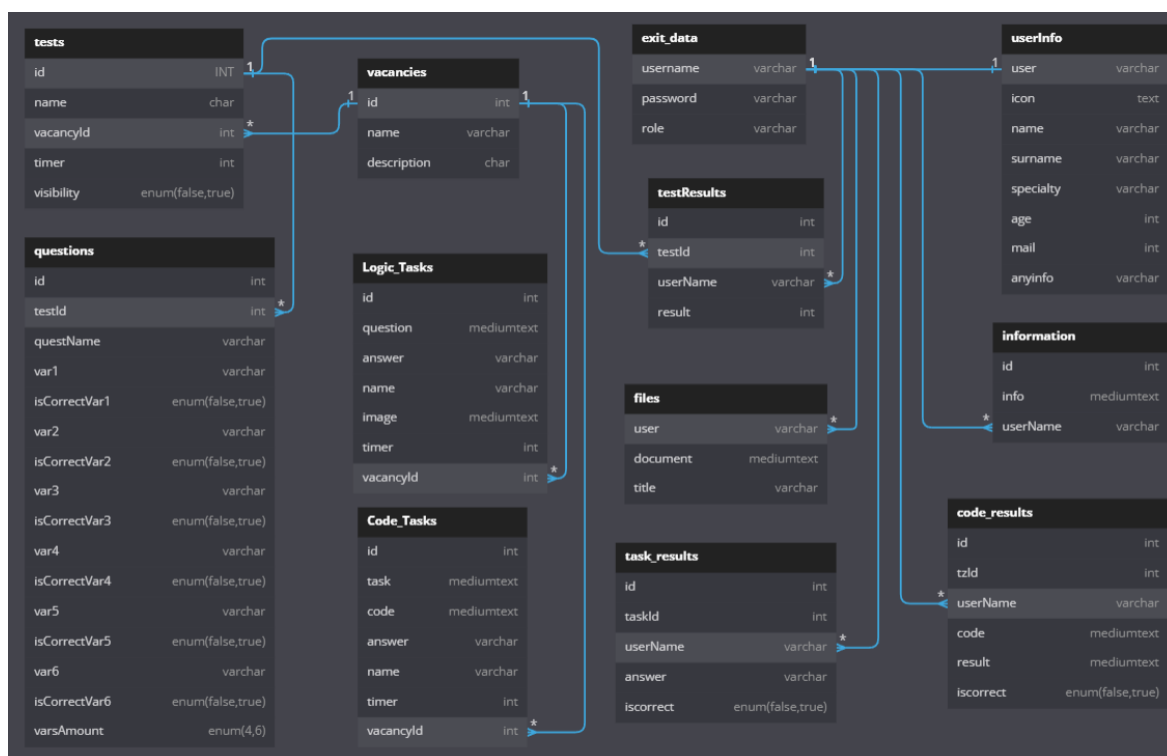


Рисунок 2.5 – ER-модель бази даних

Дана база створена запитами мовою SQL, в СУБД MySQL при використанні консольного застосунку з версією серверу 5.5. Лістинг створення кожної таблиці наведено в додатку 1.

Як видно з малюнку, кількість необхідних даних для збереження є достатньо великою. Структура є складною та заплутаною, тому для легкого користування було створено окремий клас в програмному коді, який автоматично відправляє всі запити до бази. Він дає змогу спростити код та зменшити кількість запитів, що прискорює роботу програми. Лістинг даного класу наведено на рисунку 2.6, на ньому видно всі методи, які він використовує. На частіше використовується метод «RequestWithoutAnswer» - 55 разів. Він слугує для відправлення запиту до бази не зберігаючи відповіді від самої бази даних, що зменшує час обробки.

```
internal class DataConnection
{
    private string _server;
    private string _username;
    private string _password;
    private string _database;

    6 references
    private MySqlConnection GetConnection()...

    9 references
    public DataConnection()...

    4 references
    private string GetValueFromLine(string line)...

    25 references
    public bool IsExist(string query)...

    55 references
    public void RequestWithoutAnswer(string query)...

    41 references
    public string GetValue(string query)...

    10 references
    public List<string> GetList(string query)...

    4 references
    public int GetCount(string query)...

    25 references
    public List<List<string>> GetListOfTable(string query)...
}
```

Рисунок 2.6 – Метод для взаємодії з БД

Загалом клас має вісім методів, конструктор та чотири поля, які необхідні для під'єднання до БД. Конструктор створює з'єднання з базою використовуючи метод

«GetConnection» та поля в якості вхідних даних. Всі інші методи використовуються для взаємодії з даними у базі.

#### 2.4 Реалізація математичної моделі системи

Кожна інформаційна система повинна працювати самостійно з мінімальними втручанням зі сторони людини, тобто бути максимально автоматизованою. Одною з проблем співбесід є велике витрачання часу на перевірку результатів виконання всіх завдань і тестів від кандидатів. Для того щоб пришвидшити тестування кандидатів потрібно також автоматизувати процес їх оцінювання і для нього можна використати формули наведені нижче.

Формула 2.1 надає оцінку кандидату за виконаними тестовими завданнями щодо кожної вакансії. Якщо з тестом на сто питань стає ясно, що кожна правильна відповідь рівноцінна одному балу, то зі збільшенням кількості тестів і кількості питань у них, стає важко з вирішенням остаточної оцінки.

(2.1)

$$X = \frac{(b_1 + b_2) \cdot 100}{n},$$

де  $X$  – кінцева оцінка;  $n$  – кількість запитань;  $b_1$  – кількість правильних відповідей;  $b_2$  – кількість правильних відповідей з декількома варіантами відповідей.

Результат даної формули вираховується як сума кількості правильних відповідей помножених на 100 і поділених на загальну кількість питань. Дана формула всі результати до стобальної системи і дає змогу правильно порівнювати оцінки кандидатів між собою. Структурна схема даної формули зображена на нижче рисунку 2.7.

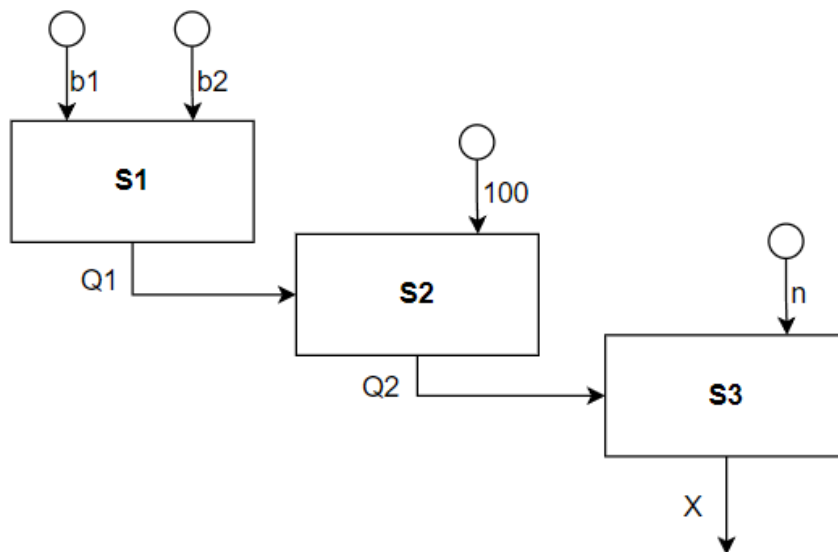


Рисунок 2.7 – Схема формули оцінки тесту

Результатом формули має стати число від 0 до 100, не залежно від кількості питань, його можна вважати відсотком правильних відповідей. В будь якому разі компанія сама має ставити свій поріг балів, на який вона очікує щодо кожної вакансії.

Проте в програмі можливо створювати запитання в яких більше 1 правильної відповіді ( $b_2$ ). Щоб зарахувати відповідь на дане питання як правильне, кандидат має виділити всі необхідні варіанти. Формула 2.2 демонструє знаходження  $b_2$ . Вданому випадку, якщо всі обрані варіанти збігають з правильними, питання буде зараховане як правильне. Навіть при правильно обраних 3 варіантів з 4, відповідь не буде зараховано.

(2.2)

$$b_2 = 1 \text{ якщо } \frac{k}{c} = 1; 0 \text{ якщо } \frac{k}{c} < 1$$

де  $b_2$  – кількість правильних відповідей з декількома варіантами відповідей;  
 $c$  – кількість правильних варіантів;  $k$  – кількість правильно обраних варіантів.

З роками кількість вакансій буде рости, а кожна з них містить певну кількість завдань, які займають місце в базі. Тому для розуміння кількості необхідної пам'яті,

яку потрібно виділити в БД, наведена формула 2.3. Вона створена для прогнозування зростання кількості завдань та вакансій з часом.

(2.3)

$$N(t) = N_0 \cdot a^t,$$

де  $N(t)$  – кількість завдань або вакансій в момент часу  $t$ ;  $N_0$  – початкова кількість завдань або вакансій;  $a$  – коефіцієнт росту.

Ця формула передбачає, що кількість завдань або вакансій зростає за експоненціальним законом. Коефіцієнт росту  $a$  можна визначити за допомогою наступної формули 2.4.

(2.4)

$$a = \frac{N(t)}{N_0},$$

де  $N(t)$  – кількість завдань або вакансій в момент часу  $t$ ;  $N_0$  – початкова кількість завдань або вакансій;  $a$  – коефіцієнт росту.

Наприклад, якщо початкова кількість завдань становить 100, а через рік їхня кількість збільшилась до 200, то коефіцієнт росту дорівнює 2. За таких умов формула прогнозуватиме, що кількість завдань через два роки становитиме 400, а через три роки - 800.

Однак, слід зазначити, що ця формула є досить спрощеною. Насправді, зростання кількості завдань та вакансій може бути більш складним і залежати від багатьох факторів, таких як: попит на спеціалістів, розвиток платформи, активність компанії на ринку праці та інші.

### Висновки до другого розділу

В другому розділі було наведено структуру системи у вигляді DFD та IDF3 діаграм. Для демонстрації створеної бази даних, була створена ER-модель, в якій наведено дванадцять таблиць та створено спеціальний клас для взаємодії з базою. Також була наведена формалізація системи, яка описує всі формули які необхідно для роботи системи, а саме: формула для знаходження кінцевої оцінки і формула для прогнозування росту кількості завдань.

## 3 РОЗРОБКА І РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПРОВЕДЕННЯ ТЕХНІЧНИХ ІНТЕРВ'Ю

### 3.1 Розроблення інтерфейсу та основних функцій системи

Одним з ключових елементів інформаційної системи є інтерфейс. Проектування інтерфейсу для цієї системи вимагає врахування різних аспектів взаємодії між користувачем та системою. Правильно спроектований інтерфейс грає важливу роль у створенні ефективної та інформаційної системи орієнтованої на різних користувачів.

Проектування інтерфейсу інформаційної системи проведення технічних інтерв'ю ґрунтується на врахуванні специфіки цільової аудиторії, особливостей завдань та вимог до взаємодії користувача з системою. Далі буде розглянуто зовнішній вигляд найважливіших форм(вікон) програми та їх опис і загальний вигляд інтерфейсу на рисунку 3.1.

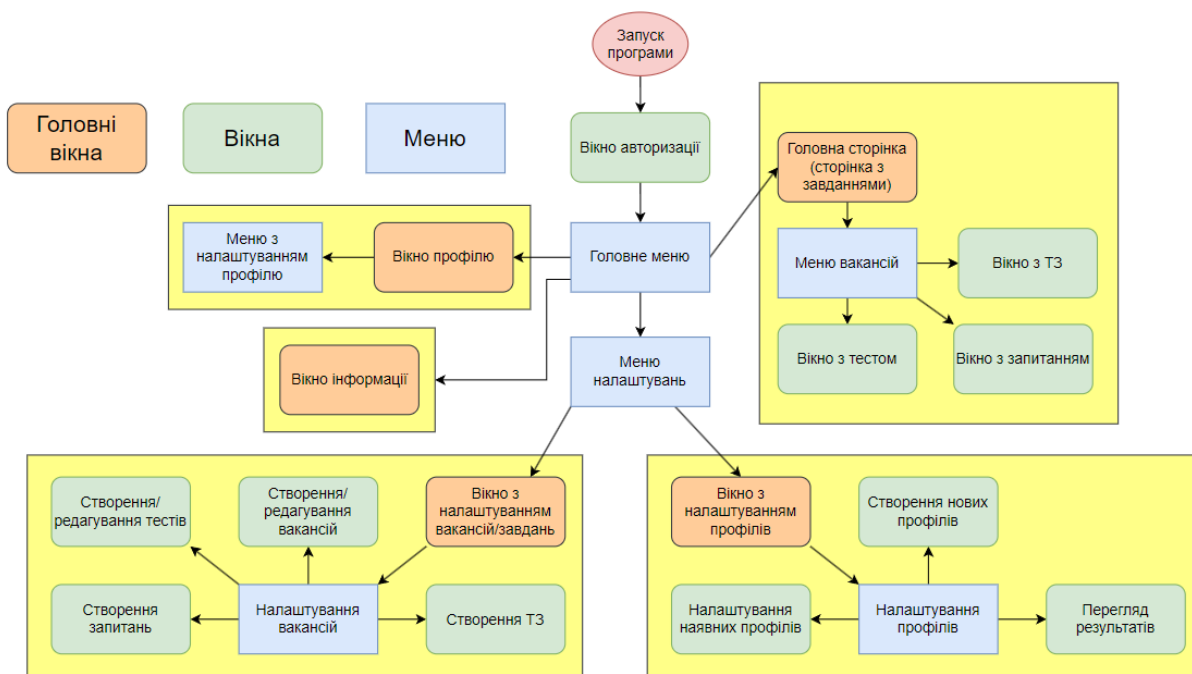


Рисунок 3.1 – Структурна схема інтерфейсу

Згідно з рисунку вище, першою формою при запуску програми є вікно авторизації, яке зображено на рисунку 3.2. Користувачу пропонується ввести логін



і пароль для подальшого користування додатком. Також наявна кнопка для вибору серверу з базою даних.

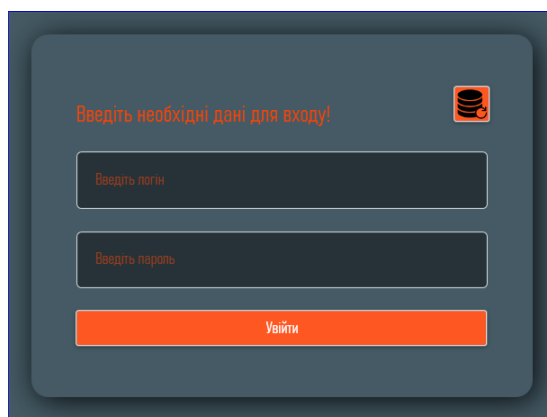
The image shows a login form with a dark blue background. At the top, there is a red heading "Введіть необхідні дані для входу!" and a small database icon. Below the heading are two input fields: "Введіть логін" and "Введіть пароль". At the bottom of the form is a red button labeled "Увійти".

Рисунок 3.2 – Форма авторизації

Користувач може за наявними даними увійти, або скасувати операцію. Реєстрація нового профілю доступно лише інтерв'юеру.

Після успішного входу, відкривається головне вікно програми – рисунок 3.3. Саме в ньому є доступ до вакансій з завданнями та до всього іншого функціоналу програми.

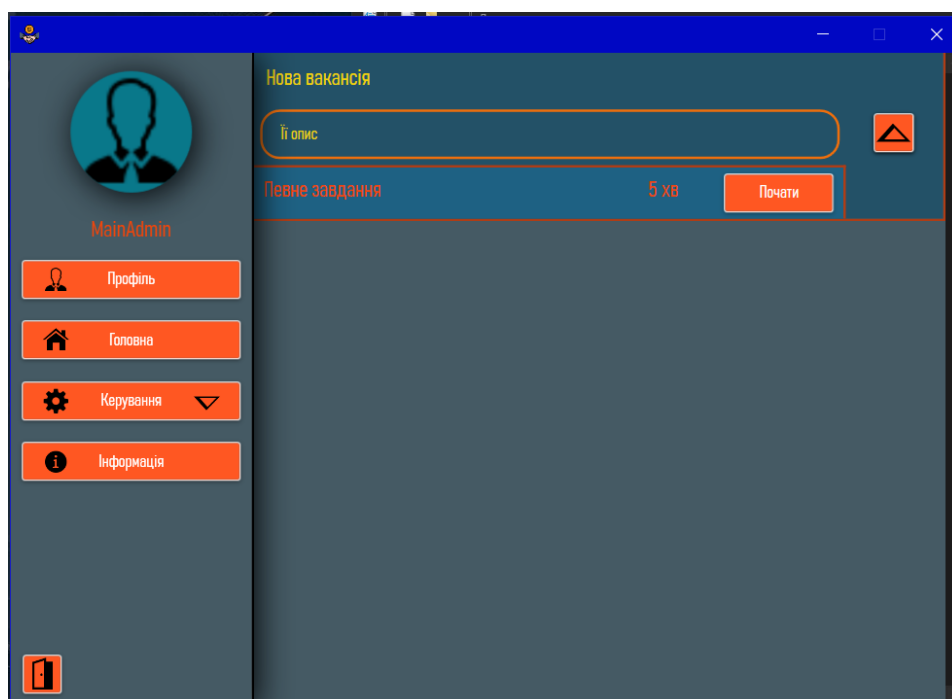


Рисунок 3.3 – Головне вікно

Кожна вакансія містить в собі по декілька завдань, які до цього інтерв'юер має занести у базу. Всі завдання структуровані за вакансіями, що зображено на рисунку 3.3. В даному випадку видно лише одне завдання з часом виконання на 5 хвилин.

Не менш важливим вікном є «Керування». Це основна форма яка дозволяє керувати та змінювати данні. При відкритті вкладки стають доступні дві кнопки: «Профілі», «Вакансії». Даний функціонал доступний лише інтерв'юерам. Кожна кнопка відкриває окрему форму згідно з їх завданням. Приклади цих вікон зображено на рисунку 3.4.

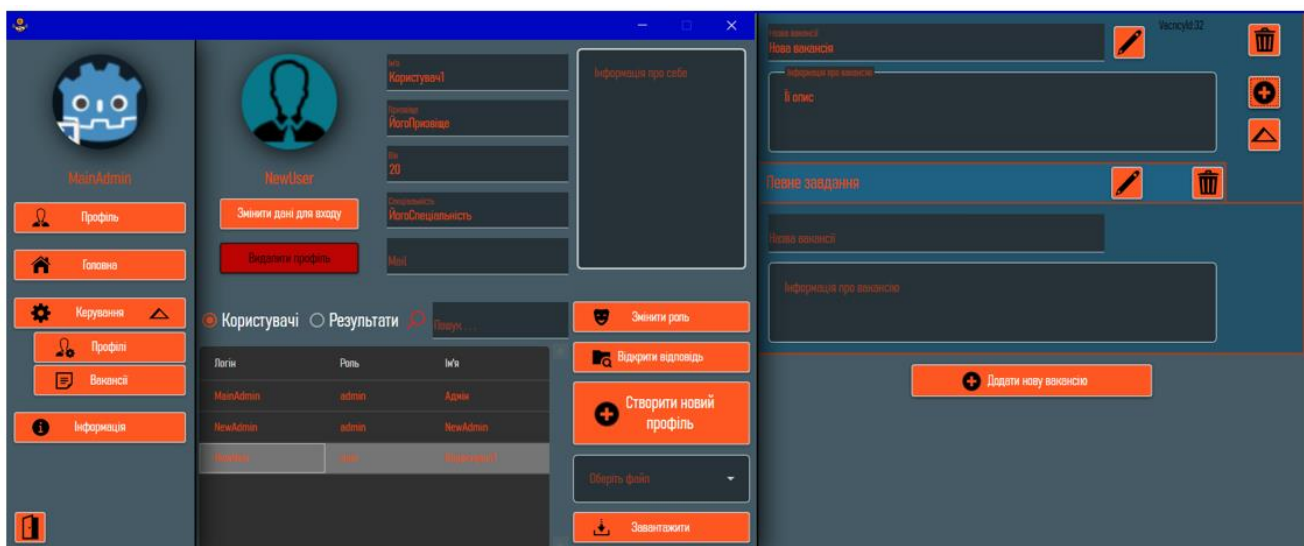


Рисунок 3.4 – Демонстрація функціоналу для керування вакансіями та профілями

Завдяки цим формам інтерв'юеру стає доступний весь необхідний функціонал для проведення інтерв'ю та пошуку необхідного кандидата. У першому вікні доступна вся інформація про користувачів та їх результати стосовно завдань. Друге вікно надає можливість для створення тестів, завдань та вакансій і їх редагування при необхідності.

Крім наведених вище вікон є ще також вікно з редагуванням власного профілю, яке доступне для всіх користувачів та схоже за функціоналом і зовнішнім

виглядом на вікно «Профілі». А також форма з інформацією, де адміністратори можуть створити певне оголошення, яке буде видиме для всіх користувачів.

### **3.2 Тестування системи і керівництво користувачу**

Для початку роботи з програмою, спершу необхідно запустити exe файл даного застосунку, потрібний файл попередньо потрібно завантажити. Його місце розташування залежить від місця завантаження.

Перед відкриттям головного вікна, спершу користувач має авторизуватися, тобто ввести логін і пароль. Ці дані можна отримати лише від інтерв'юера, який попередньо має створити новий профіль у своєму кабінеті. Також при першому запуску застосунку необхідно під'єднатися до бази даних за допомогою кнопки на формі для входу та ввести дані які потребує вікно.

Після звірення даних для входу з даними у базі, при успішному вході, користувач отримує доступ до програми, в іншому випадку йому буде запропоновано ввести їх знову. На даному етапі можливості програми змінюються в залежності з роллю користувача (admin, user).

В головному вікні користувач може почати виконувати завдання розміщені по вакансіям. Для цього потрібно натиснути на поле з вакансією, або зображення і у відкритому меню вибрати потрібний тест чи ТЗ. Після чого відкриється форма з тестом, де потрібно ставити позначку у правильний варіант відповіді, або з ТЗ де потрібно реалізувати його. Рисунок 3.5 демонструє виконання тесту користувачем.

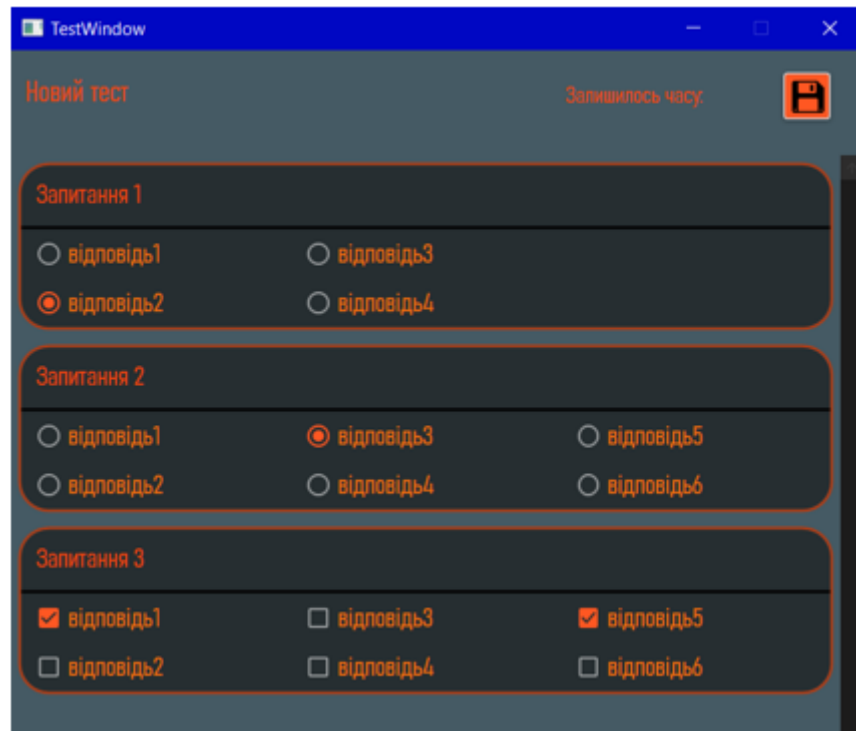


Рисунок 3.5 – Проходження тесту

Після того як пройдено всі запитання та проставлено правильні на думку кандидата відповіді, потрібно натиснути на кнопку зберегти у верхньому правому куті вікна, або ж тест самостійно завершиться при закінченні часу. Все написане вище також стосується і ТЗ.

Результат буде наведено одразу після закінчення виконання завдання. Проте при необхідності його можна ще раз переглянути у вікні «Профіль», де буде наведено всі виконані завдання користувачем. Також в даному вікні наявний функціонал для зміни інформації про даного користувача.

«Налаштування» є формою з найбільшим функціоналом, у якій можна налаштувати профіль, курувати завданнями і іншими профілями. Останні дві можливості доступні лише інтерв'юеру. У вікні редагування профілю можна змінити дані щодо кандидата, або додати нові. Також інтерв'юер може відкрити форму редагування профілів чи завдань і занести нові дані в БД, або змінити їх.

Весь функціонал застосунку працює коректно. Тестування проводилося багато разів з постійним виправленням помилок, які виникали під час роботи. Було перевірено роботу на різних пристроях з підключенням до однієї БД.

### **Висновки до третього розділу**

В останньому розділі було створено графічний інтерфейс застосунку, наведено схему загального вигляду інтерфейсу для кращого розуміння його структури. Було продемонстровано вигляд основних форм додатку та описання. А також була наведена інструкція користувачу для використання додатку.

## ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Під час роботи було описано аналіз інформаційних потреб та визначено потреби предметної області. Основною потребою є прискорення та покращення процесу відбору кандидатів на технічні посади, шляхом автоматизації. Було наведено існуючі аналоги даної інформаційної системи. Наведена вимоги до інформаційної системи та їх пріоритети.

Наведено структурну схему застосунку та описано усі її елементи та блоки. Крім того, для демонстрації структури системи також було створено DFD та IDEF3 моделі. ER-модель допоможе показати структуру бази даних необхідної для системи, в якій наведено дванадцять таблиць, та створено клас для легкої взаємодії з ними. В додатку 1 наведено лістинг для створення всіх наведених таблиць. Формалізація системи описує всі формули які необхідно для роботи системи. Згадані формули вираховують оцінки за завдання та ймовірний обсяг БД.

Кінцевий вигляд застосунку зображено по рисунках 3 розділу, де його видно графічний інтерфейс. Наведено всі вікна та вкладки необхідні для роботи, та розподілено функціонал згідно ролей. Також там описано інструкція для користування, що буде полегшувати впровадження системи.

Практичне значення створеної інформаційної системи полягає в значній економії часу роботодавців, покращенні якості відбору кандидатів на технічні посади та збільшенні обсягу опитувань. Це сприятиме підвищенню ефективності процесу найму та покращенню якості персоналу компанії.

## ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Як зробити технічну співбесіду цікавою і корисною, незалежно від результату: веб-сайт. URL: <https://dou.ua/forums/topic/40642/> (дата звернення: 13.11.2024).
2. Грицунов О. В. Інформаційні системи та технології. Навчальний посібник: 2010. 222 с.
3. John A. Introduction to Systems Engineering Practices: 2000. 52с.
4. Peter P. C. The Entity-Relationship Model — Toward a Unified View of Data: 1997. 35с.
5. Freeman A., Sussman S. Pro WPF in C# 2010: Windows Presentation Foundation in .NET 4. Apress 2010. 1216 с.
6. MacDonald M. Pro C# 2010 and the .NET 4 Platform. Apress, 2010. 1752 с.
7. Технічна співбесіда: як готуватись?: веб-сайт. URL: <https://wezom.academy/ua/tehnicheskoe-sobesedovanie-kak-gotovitsja/> (дата звернення: 03.05.2024).
8. Petzold C. Programming Windows: Writing Windows 8 Apps With C# and XAML. Microsoft Press: 2013. 1136 с.
9. Ritchie S., Venkataraman R., Stewart W. MySQL Database Usage & Administration. Sams Publishing: 2010. 792 с.
10. Ambler S. W., Sadalage P. J. Refactoring Databases: Evolutionary Database Design. Addison-Wesley Professional, 2006. 384 с.
11. Albahari J., Albahari B., Albahari O. C# 5.0 in a Nutshell: The Definitive Reference. O'Reilly Media, 2012. 1064 с.
12. Freeman A., Morrisette P. Pro ASP.NET 4 in C# 2010. Apress, 2010. 1664 с.
13. Stagner J., Delano K., Freeman A. WPF 4 Unleashed. Sams Publishing, 2010. 864 с.
14. Sharp J. Microsoft SQL Server 2008 Administration for Oracle DBAs. McGraw-Hill Education, 2009. 416 с.
15. Jeffrey R. CLR via C# (Developer Reference) 4th Edition, 2012. 986 с.

16. Watson J. Beginning Visual C# 2010. Wrox, 2010. 1080 с.
17. Lerman J. Programming Entity Framework: Code First. O'Reilly Media, 2011. 162 с.
18. Lippert E., Shilling D. Windows Presentation Foundation 4.5 Cookbook. Packt Publishing, 2012. 380 с.
19. De Smet I. C# 4.0 Unleashed. Sams Publishing, 2011. 1680 с.
20. Fitzgerald M., Kostis C. SQL in a Nutshell: A Desktop Quick Reference. O'Reilly Media, 2008. 594 с.
21. Maheshwari P., Ruhela D. A Beginner's Guide to C# Programming. Kogan Page Publishers, 2013. 336 с.
22. Albahari J., Albahari B. C# 6.0 in a Nutshell: The Definitive Reference. O'Reilly Media, 2015. 1136 с.
23. Nathan A. Windows Presentation Foundation Unleashed. Sams Publishing, 2007. 624 с.
24. Сергійчук М.О. ІНФОРМАЦІЙНА СИСТЕМА ПРОВЕДЕННЯ ТЕХНІЧНИХ ІНТЕРВ'Ю: матеріали міжфакультетної студентської науково-практичної конференції, м. Житомир, 14 листопада 2023 р. Житомир: ПНУ, 2023. С. 37-39
25. Сергійчук М.О. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПРОВЕДЕННЯ ТЕХНІЧНИХ ІНТЕРВ'Ю: матеріали всеукраїнської студентської науково-практичної конференції, м. Житомир, 10 квітня 2024 р. Житомир: ПНУ, 2024. С. 23



## ДОДАТОК 1

### Лістинг створення таблиць

```
CREATE TABLE exit_data (  
    login VARCHAR(190) NOT NULL PRIMARY KEY,  
    password VARCHAR(255),  
    role ENUM('admin', 'user')  
);  
  
CREATE TABLE user_info (  
    user VARCHAR(190) NOT NULL UNIQUE,  
    icon MEDIUMTEXT,  
    name VARCHAR(255),  
    surname VARCHAR(255),  
    age VARCHAR(255),  
    speciality VARCHAR(255),  
    mail VARCHAR(255),  
    anyInfo VARCHAR(255),  
    FOREIGN KEY (user) REFERENCES exit_data(login)  
);  
  
CREATE TABLE files (  
    user VARCHAR(190) NOT NULL,  
    document MEDIUMTEXT,  
    title VARCHAR(255) NOT NULL,  
    FOREIGN KEY (user) REFERENCES exit_data(login)  
);  
  
CREATE TABLE vacancies (  
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    name CHAR(255),  
    description CHAR(255)  
);  
  
CREATE TABLE tests (  
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    name CHAR(255),  
    vacancyId INT NOT NULL,  
    timer INT,  
    visibility ENUM('false', 'true'),  
    FOREIGN KEY (vacancyId) REFERENCES vacancies(id)  
);  
  
CREATE TABLE questions (  
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    testId INT NOT NULL,  
    questName CHAR(255),  
  
    var1 CHAR(255),  
    isCorrectVar1 ENUM('false', 'true'),  
  
    var2 CHAR(255),  
    isCorrectVar2 ENUM('false', 'true'),  
  
    var3 CHAR(255),  
    isCorrectVar3 ENUM('false', 'true'),  
  
    var4 CHAR(255),  
    isCorrectVar4 ENUM('false', 'true'),  
  
    var5 CHAR(255),  
    isCorrectVar5 ENUM('false', 'true'),
```

```

    var6 CHAR(255),
    isCorrectVar6 ENUM('false', 'true'),

    varsAmount ENUM('4', '6'),
    FOREIGN KEY (testId) REFERENCES tests(id)
);

CREATE TABLE testResults (
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    testId INT NOT NULL,
    userName VARCHAR(190),
    result INT,
    FOREIGN KEY (userName) REFERENCES exit_data(login)
);

CREATE TABLE information(
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    info MEDIUMTEXT,
    userName VARCHAR(190),
    FOREIGN KEY (userName) REFERENCES exit_data(login)
);

CREATE TABLE Logic_Tasks(
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    question MEDIUMTEXT,
    answer VARCHAR(190),
    name VARCHAR(190),
    image MEDIUMTEXT,
    timer INT,
    vacancyId INT NOT NULL,
    FOREIGN KEY (vacancyId) REFERENCES vacancies(id)
);

CREATE TABLE Code_Tasks(
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    task MEDIUMTEXT,
    code MEDIUMTEXT,
    answer VARCHAR(190),
    name VARCHAR(190),
    timer INT,
    vacancyId INT NOT NULL,
    FOREIGN KEY (vacancyId) REFERENCES vacancies(id)
);

CREATE TABLE task_results (
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    taskId INT NOT NULL,
    userName VARCHAR(190),
    answer VARCHAR(190),
    iscorrect ENUM('false', 'true'),
    FOREIGN KEY (userName) REFERENCES exit_data(login)
);

CREATE TABLE code_results (
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    tzId INT NOT NULL,
    userName VARCHAR(190),
    code MEDIUMTEXT,
    result MEDIUMTEXT,
    iscorrect ENUM('false', 'true'),
    FOREIGN KEY (userName) REFERENCES exit_data(login)
);

```