

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики



Побудова інформаційної системи для рекрутингу та проведення співбесід

Текстова частина магістерської роботи

за спеціальністю „Інженерія програмного забезпечення” 121

Керівник магістерської роботи:

д.т.н., доц. Ковалюк Т. В.

(підпис)

“ ____ ” _____ 2023 р.

Виконав:

студент Кузьменко І. Є.

“ ____ ” _____ 2021 р.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	4
ВСТУП.....	5
РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Теоретичні відомості.....	7
1.2 Огляд існуючих систем для відбору кадрів.....	22
1.3 Постановка задачі.....	26
1.4 Математична формалізація задачі	30
Висновки розділу 1.....	31
РОЗДІЛ 2. АНАЛІЗ АРХІТЕКТУРНИХ ТА ТЕХНІЧНИХ РІШЕНЬ	33
2.1 Огляд архітектурних підходів.....	33
2.2 Аналіз вимог та визначення ключових характеристик програмної системи	36
2.3 Огляд технологій, методів та інструментів розробки	41
2.4 Розробка стратегії реалізації програмної системи.....	43
Висновки розділу 2.....	44
РОЗДІЛ 3. РЕАЛІЗАЦІЯ КРОСПЛАТФОРМЕННОГО ВЕБ-ЗАСТОСУНКУ ДЛЯ ПРОВЕДЕННЯ СПІВБЕСІД.....	45
3.1 Проектування архітектури.....	45
3.2 Розробка модуля автентифікації.....	50
3.3 Розробка модуля проведення співбесід	53
3.4 Оцінювання кандидатів та формування результатів проведення співбесіди	55
3.5 Розгортання	57
3.6 Опис програмного продукту	59
Висновки розділу 3.....	61

ВИСНОВКИ	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	65
ДОДАТКИ.....	71

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ІС – Інформаційна Система

ПЗ – Програмне Забезпечення

БД – База Даних

HR – Human resources

DDD – Domain-driven design

ATS – Applicant Tracking System

ORM – Object Relational Mapping

ВСТУП

Актуальність.

Рекрутинг – це загальний процес виявлення, пошуку, перевірки, короткого списку та співбесіди з кандидатами на роботу в організації. Підбір кадрів є однією із центральних функцій управління, тому що саме працівники забезпечують ефективне використання будь-яких ресурсів, які знаходяться у розпорядженні організацій, а саме від працівників у кінцевому результаті залежать конкурентоздатність компанії та її економічні показники.

Усі організації, підприємства, компанії постійно відчують потребу в персоналі. Залучення персоналу відбувається в декілька етапів основним з яких є співбесіда.

Завданням співбесіди з боку підприємства є виявлення з числа претендентів на певну вакансію найбільш релевантного та професійно-підготовленого кандидата, а з боку кандидата – дізнатися, чи підходить йому посада, на яку він претендує, та перевірити власні компетентності (знання, уміння). Для цього рекрутеру необхідно визначити найбільш релевантний перелік питань та способів вимірювання здібностей кандидата до виконання вимог, що потребує посада.

Мета дослідження. Оптимізувати роботу менеджерів з підбору персоналу, покращити якість відбору персоналу на етапі проведення співбесід, впровадити прозорі оцінки результатів співбесід.

Завдання дослідження. Зробити огляд та аналіз сучасних типів та методів проведення співбесід. Проаналізувати наявні методи та категорії суб'єктивного оцінювання кандидатів на посаду в ході співбесід. Ознайомитися з існуючими бібліотеками та інструментами, котрі потрібно використовувати для автоматизації процесів рекрутингу. Дослідити способи структурування програмної системи для оптимізації розробки та можливості подальшого розвитку автоматизованої системи.

Об'єкт дослідження: програмна система для автоматизації процесу рекрутингу із проведенням співбесід для відбору персоналу.

Предмет дослідження: методи, засоби та інструменти, що застосовуються під час розробки автоматизованої системи рекрутингу, архітектурні підходи для масштабування програмної системи.

Джерела дослідження: електронні ресурси (в тому числі вебсайти компаній, котрі пропонують системи автоматизації, форуми, тощо), друкована література, довідники, матеріали конференцій, вихідні коди програм та бібліотек.

Наукова новизна полягає в автоматизації процесів відбору кадрів та покращенні підходів до залучення кадрових ресурсів, завдяки підвищенню якості оцінювання знань та навичок кандидатів на посаду, оптимізувати час менеджера з відбору персоналу для проведення співбесіди.

РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

Огляд предметної області є важливим етапом для поглиблення знань та відображає конкретну галузь діяльності, проблеми чи інтереси, які стосуються певної теми. Вона включає терміни, поняття, правила, бізнес-процеси та бізнес-об'єкти, які характерні для певної галузі.

У контексті розробки програмного забезпечення предметна область стосується сфери діяльності, для якої створюється програмний продукт або програмна система. Розуміння предметної області допомагає розробникам створювати ефективні функціональні рішення, які задовольняють потреби користувачів та відповідають специфіці галузі. В процесі розробки програмного забезпечення аналіз предметної області є важливим етапом, який передуює проектуванню, розробці та тестуванню програмної системи.

Для глибокого розуміння предметної області розробники співпрацюють з експертами в галузі, вивчають документацію та аналізують існуючі рішення або системи. Це допомагає виявити та проаналізувати ключові аспекти, зокрема:

- основні поняття та терміни, що пов'язані з предметною областю;
- структура та відношення між різними об'єктами та процесами предметної області;
- бізнес-правила та обмеження, які визначають функціонування системи в рамках предметної області;
- сценарії використання системи, що відображають різні аспекти діяльності користувачів у контексті предметної області.

1.1 Теоретичні відомості

Управління персоналом – це комплексний процес, спрямований на організацію ефективної роботи персоналу організації для досягнення цілей, як організації, так і особистих. Його основною метою є забезпечення компанії кваліфікованими та мотивованими співробітниками, які відповідають потребам бізнесу та мають можливість проявляти свій потенціал.

Управління персоналом включає ряд етапів:

1. Планування кадрів включає в себе визначення потреби в робочій силі, встановлення критеріїв відбору співробітників та розробку стратегії управління персоналом.
2. Набір та відбір персоналу передбачає пошук та відбір кандидатів на відповідну посаду, враховуючи їх кваліфікацію, досвід та мотивацію.
3. Розміщення співробітників передбачає організацію їх вступу в роботу та адаптації до нового робочого місця.
4. Розвиток персоналу включає навчання, тренінги та відвідування конференцій для підвищення рівня знань та навичок співробітників.
5. Оцінка та оплата праці дозволяє визначити відповідальність та результативність співробітника, встановити систему оцінки та винагородження.
6. Утримання персоналу передбачає організацію комфортної атмосфери для співробітників, забезпечення їх соціальними пакетами та безпекою на робочому місці.

Дотримання всіх цих етапів допомагає забезпечити ефективну роботу компанії та зберегти кваліфікованих співробітників на тривалий термін.

Інформаційна система для управління персоналом дозволить автоматизувати процеси найму, оцінки та оплати праці, а також забезпечити ефективний моніторинг та аналіз продуктивності та задоволеності співробітників.

Важливою складовою управління персоналом є: визначення кандидатів, які мають потрібні знання та вміння для успішного виконання завдань на робочому місці. В свою чергу набір та відбір персоналу – це два різних етапи процесу управління персоналом, які мають різну мету та завдання.

Набір персоналу включає в себе пошук та залучення кандидатів на відповідну посаду в компанії. Головна мета набору полягає в тому, щоб знайти кандидата, який відповідає вимогам компанії та може бути успішним на певній вакансії. Набір персоналу включає в себе такі етапи, як публікація оголошень про вакансії, розгляд пропозицій кандидатів, співбесіди та інтерв'ю. Метою такого

процесу є створення необхідного резерву працівників. Визначення необхідного обсягу набору кадрів зазвичай визначається різницею між наявною робочою силою та майбутніми потребами в ній.

Натомість відбір персоналу є наступним етапом та включає в себе процес вивчення психологічних та професійних якостей кандидатів, які пройшли набір та задовольняють вимогам компанії. Головна мета відбору полягає в тому, щоб обрати найбільш відповідного до посади кандидата зі списку тих, хто пройшов набір. Відбір персоналу включає в себе такі етапи, як перевірка резюме, академічних і професійних рекомендацій, додаткові інтерв'ю, оцінка знань та навичок кандидатів.

Для ефективного набору та відбору персоналу важливо мати чіткий план дій та правила відбору, які мають бути доступні всім співробітникам, які беруть участь у цьому процесі. Також важливо мати певні критерії відбору, такі як освіта, досвід, навички та інші, які допоможуть зробити правильний вибір.

Відбір кадрів можуть здійснювати різні люди або відділи в компанії, залежно від розміру компанії та внутрішньої організації. Це може бути HR-відділ, рекрутери, менеджери, керівники підрозділів, а також власник компанії або члени його керівного органу. Кожен з цих фахівців може мати свої вимоги та методи відбору кандидатів, але загалом їх мета – знайти найкращих кандидатів, які відповідають вимогам компанії та підходять для вакансії. Крім того, вони повинні мати розуміння того, які навички та характеристики потрібні для успішної роботи в компанії та на конкретній посаді. Важливо також враховувати етичні принципи та стандарти відбору кандидатів.

Ключові ролі

У контексті відбору кадрів важливо відзначити такі ключові ролі, як Recruiter (Рекрутер), Hiring Manager (Менеджер з найму) та Human Resources (Людські ресурси) – це три різних професійні ролі, пов'язані з процесом найму та управлінням персоналом в організації. Основні відмінності між ними полягають у їхніх обов'язках та функціональних областях.

Рекрутер – це фахівець з найму, який відповідає за пошук, приваблення та відбір кандидатів на різні вакансії в організації. Рекрутери зазвичай співпрацюють з менеджерами з найму, щоб зрозуміти вимоги до вакансії та забезпечити відповідність кандидатів вимогам. Вони також можуть проводити співбесіди з кандидатами та розробляти стратегії залучення талантів.

Менеджер з найму – це менеджер, який відповідає за певну ділянку бізнесу та зазвичай визначає вимоги до вакансії, обирає кандидата та встановлює з ним контакт. Менеджери з найму також можуть проводити співбесіди з кандидатами та оцінювати їх під час процесу найму.

Людські ресурси (HR) – це департамент, який відповідає за управління персоналом в організації. HR включає в себе широкий спектр функцій, зокрема управління заробітною платою, підтримку культури підприємства, планування кар'єри та розвитку персоналу, виконання адміністративних функцій, пов'язаних з управлінням персоналом та забезпеченням відповідності правовим вимогам.

Хоча всі три ролі можуть працювати разом, їхні обов'язки та функціональні області можуть залежати від організації та відділу, в якому вони працюють. Наприклад, у невеликих компаніях може бути тільки один рекрутер, який відповідає за всі аспекти набору персоналу, тоді як у великих компаніях можуть бути окремі команди рекрутерів, менеджерів з найму та HR.

У кожній ролі є свої унікальні вимоги та навички. Рекрутери повинні мати досвід у пошуку талантів та використанні інструментів для залучення кандидатів. Менеджери з найму повинні мати глибокі знання в своїй галузі та бути здатними визначати вимоги до вакансії та кандидатів. HR повинен мати знання управління персоналом та здатність розуміти потреби бізнесу та розробляти стратегії управління персоналом, щоб підтримати розвиток організації.

У кожній ролі є свої виклики, включаючи управління бюджетом, забезпечення відповідності правовим вимогам, розв'язання конфліктів та забезпечення успішного залучення та управління персоналом. Всі три ролі є важливими для успішного управління персоналом в організації і повинні

працювати разом, щоб забезпечити ефективний процес найму та управління персоналом.

Етапи відбору кадрів

Етапи відбору кадрів можуть включати такі дії:

1. Планування вакансії.
2. Розміщення оголошення про роботу.
3. Відбір кандидатів.
4. Співбесіди
5. Прийняття рішення про найм.

Детально розглянемо кожен з цих етапів.

Першим етапом відбору кадрів є планування вакансії. На цьому етапі необхідно визначити, яким вимогам має відповідати кандидат на вакансію, які навички та досвід повинні бути в нього, та який рівень заробітної плати відповідає цій посаді. Цей етап допомагає менеджеру краще зрозуміти, якого працівника потрібно шукати.

Другий етап – це розміщення оголошення про роботу. На цьому етапі компанія повинна підготувати та опублікувати оголошення про роботу відповідно до вакансії, щоб залучити якнайбільше потенційних кандидатів. Оголошення можна розмістити на сайті компанії, на спеціалізованих сайтах пошуку роботи, в соціальних мережах, а також відправити розсилку зацікавленим кандидатам.

Третім етапом є відбір кандидатів. На цьому етапі рекрутер або менеджер з найму переглядає резюме та заявки від потенційних кандидатів та відбирає тих, які відповідають вимогам вакансії та мають потрібний рівень кваліфікації та досвіду. Під час відбору кандидатів важливо враховувати критерії відбору, які були встановлені на попередньому етапі.

Четвертим етапом є співбесіди з відібраними кандидатами. Цей етап може включати телефонні співбесіди, онлайн-інтерв'ю, тестування та особисті співбесіди. На цьому етапі кандидати повинні продемонструвати свої навички та досвід та показати, чому вони є відповідними для вакансії. Також на цьому етапі

можуть відбуватися перевірки зі сторони роботодавця, такі як перевірка посвідчень, референс-списків та досвіду роботи.

Останнім етапом є прийняття рішення про найм кандидата. Після того, як всі співбесіди проведені, рекрутер або менеджер з найму повинен прийняти рішення про те, хто з кандидатів найкраще відповідає вимогам вакансії та буде найбільш ефективним співробітником для компанії. Цей етап може включати запрошення вибраного кандидата на роботу та вирішення всіх питань, пов'язаних з укладанням договору та прийняттям на роботу.

Співбесіда

Співбесіда – це інтерв'ю, яке складається з розмови між кандидатом на посаду та представником роботодавця. Це необхідно роботодавцю для оцінки здібності кандидата виконувати певну роботу. В свою чергу мета людини, що шукає роботу, полягає в тому, щоб зрозуміти для себе, чи відповідають посада та умови компанії його очікуванням та вимогам. Співбесіди є одним з найпоширеніших методів відбору співробітників. Вони відрізняються за ступенем структурованості запитань, від неструктурованої вільної розмови до співбесіди з жорсткою структурою, в ході якої, кандидат повинен відповідати на упорядкований перелік заздалегідь визначених питань. Згідно з дослідженнями Шмідта та Хантера [1] структуровані співбесіди мають вищий показник діагностичної точності, порівняно з неструктурованими співбесідами, оскільки вони дають більш об'єктивну оцінку компетенцій кандидата та знижують ризик впливу сторонніх факторів, таких як особисті симпатії або антипатії до кандидата. Також було встановлено, що структуровані співбесіди можуть бути ефективними для передбачення різних аспектів успішності кандидата на посаді, таких як професійний розвиток, успішність в роботі, мотивація та здатність працювати в команді. Співбесіда при прийомі на роботу вважається одним із найкорисніших інструментів для оцінки потенційних працівників. Це також вимагає значних ресурсів від роботодавця, але, як відомо, є ненадійним у визначенні оптимальної людини для роботи. Співбесіда також дозволяє кандидату оцінити корпоративну культуру та вимоги роботи. Зазвичай

співбесіда являє собою завершальний етап в процесі відбору кандидата на посаду. Перед проведенням співбесіди проводиться оцінка резюме від зацікавлених кандидатів і відбір найбільш релевантних кандидатів на конкретну посаду. Також перед співбесідою деякі компанії пропонують кандидатам виконати тестове завдання. Таким чином, на етапі співбесіди відбирається невелика кількість найбільш релевантних кандидатів на конкретну посаду.

Під час відбору кандидатів можуть використовуватися кілька раундів співбесід та/або інші методи, якщо є багато кандидатів або робота є особливо складною чи обґрунтованою. Початкові раунди, які іноді називають «відбірковими співбесідами», можуть залучати менше співробітників роботодавців і зазвичай будуть набагато коротшими та менш глибокими. Все більш поширеним підходом до первинної бесіди є телефонна співбесіда або переписка. Це особливо часто трапляється, коли кандидати не живуть поруч із роботодавцем, мають перевагу утримувати низькі витрати для обох сторін та/або при відборі кандидатів на віддалену роботу. З 2003 року співбесіди проводяться за допомогою програмного забезпечення для відеоконференцій, наприклад, Teams, Zoom, Discord. Після співбесіди з усіма кандидатами роботодавець зазвичай вибирає найбільш бажаних кандидатів і розпочинає переговори щодо пропозиції роботи.

Типи співбесід

Існує багато типів співбесід, які можуть проводити організації. Однак для всіх типів інтерв'ю структура повторюється. Наскільки співбесіда буде структурована або розроблена та проведена однаково для всіх заявників, залежить від кількості певних елементів, включених до цієї співбесіди. Загалом, співбесіду можна стандартизувати як щодо змісту (тобто, які запитання ставляться), так і щодо процесу оцінювання (тобто, як оцінюються відповіді претендентів на запитання). Коли інтерв'ю є стандартизованим, підвищується ймовірність того, що оцінки співбесідника обумовлені якістю його/її відповідей, а не факторами, не пов'язаними з роботою. Вважається, що структуру інтерв'ю доцільніше мати на континуумі, починаючи від повністю неструктурованого до

повністю структурованого [2]. Проте структуру часто розглядають як таку, що має лише дві категорії (тобто структурована проти неструктурованої), що багато дослідників вважають надто простим підходом [3].

Неструктурована співбесіда

Неструктурована співбесіда або така, яка не містить значної кількості елементів стандартизації, є сьогодні найпоширенішою формою співбесіди [4]. Неструктуровані співбесіди зазвичай розглядаються як вільні; менеджер з найму може міняти місцями або змінювати запитання на свій розсуд, і різні його/її колеги можуть не однаково оцінювати відповіді потенційного працівника. Також немає вказівок щодо того, як інтерв'юер та респондент повинні взаємодіяти до, під час або після інтерв'ю. Неструктуровані інтерв'ю, по суті, дозволяють інтерв'юєру проводити співбесіду так, як він або вона вважає найкращим.

Враховуючи, що неструктуровані інтерв'ю можуть змінюватися залежно від того, хто може бути інтерв'юєром, не дивно, що рекрути зазвичай віддають перевагу неструктурованим співбесідам [5]. Інтерв'юєри, як правило, розвивають впевненість у своїй здатності точно оцінювати співбесідників [6], виявляти, чи претенденти підробляють свої відповіді [7], і довіряти їхньому судженню про те, чи є ця особа хорошим кандидатом на певну посаду [8]. Неструктуровані інтерв'ю дозволяють інтерв'юєрам робити це більш вільно. Проте дослідження показують, що неструктуровані інтерв'ю насправді є дуже ненадійними або непослідовними аніж інтерв'ю. Це означає, що два інтерв'юєри, які проводять співбесіду з тією самою людиною, можуть не погоджуватися й бачити кандидата по-різному, навіть якщо вони були на одній співбесіді з цим претендентом. Часто інтерв'юєри, які проводять неструктуровані співбесіди, не можуть визначити високоякісних кандидатів на роботу [9].

Структурована співбесіда

Структура співбесіди – це ступінь ідентичності співбесід та їх проведення між усіма заявниками [2]. Структуровані співбесіди також відомі як керовані, систематичні чи шаблонні співбесіди, що спрямовані на уніфікацію змісту і критеріїв оцінювання для кожного кандидата, з яким проходила співбесіда.

Зокрема, дослідники зазвичай звертаються до 15 елементів [10], які можуть бути використані, щоб зробити зміст інтерв'ю та процес оцінювання подібними. Ступінь структурованості інтерв'ю часто розглядається як ступінь, до якого ці елементи включені під час проведення інтерв'ю.

Структура інтерв'ю створюється відповідно до поставлених завдань, але можна виокремити основні елементи:

- запитання мають відповідати роботі, яка зазначена в пропозиції;
- усі респонденти мають отримувати однакові запитання;
- інтерв'юери мають обмежити кількість додаткових запитань та підказок;
- допоміжна інформація має бути доступна для інтерв'юерів, така як резюме;
- у претендентів має бути мінімум запитань під час співбесіди.

Стратегії інтерв'ю

Дослідники з питань рекрутингу намагалися визначити стратегії інтерв'ю або шаблони, які можуть допомогти інтерв'юерам обрати найкращого кандидата. Дослідження показують, що співбесіди охоплюють широкий спектр характеристик претендента. Конструкції можна класифікувати на три категорії: контент, що має відношення до роботи, продуктивність інтерв'юера (поведінка, не пов'язана з роботою, але яка впливає на оцінку), і упередження інтерв'юера, які не мають відношення до роботи.

Зміст інтерв'ю, пов'язаний з роботою: запитання для співбесіди зазвичай створені для визначення характеристик кандидата, які мають відношення до роботи, на яку претендує особа. Кандидат, який має відношення до роботи, пояснює, що питання, які нібито оцінюють, вважаються необхідними для успішного виконання роботи. Релевантні для роботи конструкції, які були оцінені під час співбесіди, можна класифікувати за трьома категоріями: загальні риси, фактори досвіду та основні елементи роботи. Перша категорія відноситься до відносно стабільних ознак заявника. Друга категорія відноситься до професійних знань, які претендент набув з часом. Третя категорія відноситься до знань, навичок і навичок, пов'язаних з роботою.

Загальні риси кандидатів, які оцінюються на співбесіді

На співбесіді оцінюються різноманітні риси та характеристики кандидата, які можуть включати:

- розумові здібності: здатність заявників слухати, спілкуватися, працювати в команді, бути уважним до деталей, а також вивчати й обробляти інформацію [11];
- особистість: сумлінність, приємність, емоційна стабільність, екстравертність, відкритість до нового досвіду [11];
- інтереси, цілі та цінності: мотиви, цілі та відповідність особи та організації [11].
- досвід: відповідні для роботи знання, отримані з попереднього досвіду;
- освіта: знання, пов'язані з роботою, отримані з попередньої освіти;
- навчання: відповідні для роботи знання, отримані в результаті попереднього навчання.
- декларативні знання: набуті знання кандидата [12];
- процедурні навички та вміння: здатність кандидатів виконувати завдання, необхідні для виконання роботи [13];
- мотивація: готовність претендентів докласти зусиль, необхідних для виконання роботи [14].

Оцінка інтерв'юером відповідей кандидатів також, як правило, залежить від того, як кандидат поводить себе під час співбесіди. Така поведінка може не бути безпосередньо пов'язаною з конструкціями, які були розроблені для оцінки запитань співбесіди, але може бути пов'язана з аспектами роботи, на яку вони претендують. Кандидати, не усвідомлюючи цього, можуть вчинити ряд дій, які впливатимуть на оцінку їх ефективності. Респондент міг набути такої поведінки під час навчання або під час попереднього досвіду співбесіди. Ці конструкти продуктивності кандидатів також можна класифікувати за трьома категоріями: навички соціальної ефективності, міжособистісна презентація та особисті/контекстуальні фактори.

У ході співбесіди фахівці з найму оцінюють навички соціальної ефективності, такі як керування враженнями, яке може бути використано під час співбесіди для досягнення певної мети, наприклад, отримання роботи [15, 16]. Це може включати використання тактик, таких як відображення певних рис характеру, встановлення сприятливої взаємодії зі співбесідником, демонстрацію знань та навичок, висвітлення своїх досягнень тощо.

Керування враженнями не означає підробки або нечесності. Це лише процес забезпечення позитивного враження про себе, яке може допомогти кандидату виступити більш переконливо на співбесіді та отримати більшу імовірність найняття на роботу. Однак кандидатам, важливо залишатися вірним собі та не перестаратися з керуванням враженнями, щоб не виглядати нещирим або неправдивим.

Також оцінюються соціальні навички, такі як: здатність кандидата адаптуватися відповідно до вимог ситуації, щоб позитивно вплинути на інтерв'юера [17]; самоконтроль та регулювання поведінки кандидатів для контролю образу, представленого інтерв'юеру [18]; реляційний контроль або спроба кандидата контролювати хід розмови [19]. Важливою є міжособистісна презентація яка включає в себе: словесне вираження (висота, швидкість, паузи, тон) [20, 21] та невербальну поведінку (погляд, посмішка, рух рук, орієнтація тіла) [22].

Упередження інтерв'юерів, які не стосуються роботи

Нижче наведені особисті та демографічні характеристики, які потенційно можуть вплинути на оцінку інтерв'юером відповідей опитуваних. Ці фактори, як правило, не мають відношення до того, чи може особа виконувати роботу (тобто не пов'язані з продуктивністю праці), тому їх вплив на рейтинги співбесід слід мінімізувати або виключити. Фактично, у багатьох країнах існують закони, які забороняють враховувати багато з цих захищених класів людей під час прийняття рішень щодо вибору. Використання структурованих інтерв'ю з кількома інтерв'юерами в поєднанні з навчанням може допомогти зменшити вплив наступних характеристик на рейтинги інтерв'ю [23]. Нижче наведено

список упереджень інтерв'юерів, які не стосуються роботи і не повинні впливати на оцінку кандидата:

- привабливість: фізична привабливість заявника може вплинути на оцінку інтерв'юером результатів співбесіди [22];
- раса: білі, як правило, отримують вищі бали, ніж чорношкірі та латиноамериканці [11], натомість, расова схожість між інтерв'юером і заявником, з іншого боку, не виявила впливу на рейтинги інтерв'ю [23, 24];
- стать: жінки, як правило, отримують трохи вищі бали за співбесідами, ніж чоловіки [11], гендерна схожість, згідно досліджень, не впливає на рейтинги за співбесідами [23];
- подібність у минулому та ставленнях: виявлено, що інтерв'юери сприймають міжособистісну привабливість, яка впливає на рейтинги інтерв'ю [25];
- культура: кандидати з етнічним іменем та іноземним акцентом розглядаються менш прихильно, ніж кандидати лише з етнічним іменем без акценту або заявники з традиційним іменем з акцентом чи без [26].

Оцінки результатів співбесіди значно варіюються, залежно від рівня структури інтерв'ю, типів поставлених запитань, упередженості інтерв'юера чи кандидата, професійного одягу, невербальної поведінки кандидата та безлічі інших факторів. Наприклад, деякі дослідження показують, що когнітивні здібності, освіта, навчання та досвід роботи кандидата можуть бути краще охоплені під час неструктурованих співбесід, натомість знання про роботу, організаційна відповідність, навички міжособистісного спілкування та прикладні знання можуть бути краще зафіксовані під час структурованої співбесіди [11].

Процес співбесіди

Розглядати процес співбесіди можна як три окремі, хоча й пов'язані фази:

1. Фаза перед співбесідою, яка відбувається перед зустріччю інтерв'юера та кандидата.
2. Фаза співбесіди, на якій проводиться співбесіда.
3. Фаза після співбесіди, на якій інтерв'юер оцінює кваліфікацію кандидата та приймає остаточне рішення [27].

Хоча можна ці три фази розглядати окремо, проте вони пов'язані. Враження, які інтерв'юери формують на ранній стадії, можуть вплинути на те, як вони оцінюватимуть людину на наступних етапах. Нижче розглянемо детальніше кожен етап.

1. Етап перед співбесідою передбачає збір інформації про потенційного працівника, яка доступна інтерв'юеру заздалегідь, зокрема: резюме, результати тестів, інформація з соціальних мереж. На цьому етапі інтерв'юери, ймовірно, вже мають уявлення про характеристики, які роблять людину ідеальною чи кваліфікованою для цієї посади [27]. Інтерв'юери також мають інформацію про респондента, як правило, у формі резюме, результатів тестів або попередніх контактів з ним [27]. Потім інтерв'юери об'єднують інформацію, яку вони мають про кандидата, зі своїми уявленнями про ідеального працівника, щоб сформулювати оцінку кандидата перед співбесідою. Таким чином, інтерв'юери зазвичай мають враження ще до фактичної взаємодії віч-на-віч. У наш час, завдяки сучасним технологічним досягненням, рекрутери мають ще більше інформації про деяких кандидатів. Наприклад, інтерв'юери можуть отримувати інформацію з пошукових систем, блогів і навіть соціальних мереж. Хоча частина цієї інформації може бути пов'язана з роботою, частина може бути ні. У деяких випадках перевірка соцмереж може виявити небажану поведінку, наприклад пияцтво або вживання наркотиків. Будь-яка інформація, яку інтерв'юери отримують про кандидата до співбесіди, ймовірно, вплине на їхнє враження про кандидата [27, 28]. Крім того, в ході досліджень виявлено, що думки рекрутерів про кандидата перед співбесідою (фаза перед співбесідою) пов'язані з їх оцінкою кандидата після співбесіди, незважаючи на те, як кандидат міг виступити під час співбесіди [29].

2. Етап співбесіди передбачає фактичне проведення співбесіди, взаємодію між інтерв'юером та кандидатом. Початкове враження інтерв'юера про заявника до співбесіди може вплинути на кількість часу, який інтерв'юер витрачає на проведення співбесіди, поведінку інтерв'юера та його опитування [30] та оцінки інтерв'юера після співбесіди [29].

Оскільки співбесіди зазвичай проводяться віч-на-віч, по телефону або через відеоконференцію [31], вони є соціальною взаємодією між, принаймні, двома особами. Таким чином, під час співбесіди рекрутер може несвідомо передати деяку інформацію кандидату через свою мову тіла та поведінку. Тобто, інколи, під час співбесіди можна зрозуміти, як рекрутер відноситься до кандидата, позитивно або негативно [27]. Отримання такої інформації може вплинути на поведінку кандидата, що може стати причиною самовтілюваного пророцтва, коли очікування людини про певну ситуацію стають причиною її реалізації [30, 32]. У контексті співбесіди це означає, що якщо кандидат дізнається, що рекрутер ставиться до нього з негативними очікуваннями, це може вплинути на його поведінку та відповіді під час співбесіди, що може стати причиною того, що очікування рекрутера справдяться. Тобто, самовтілюване пророцтво може мати вплив на результат співбесіди та вибір кандидата на посаду. З іншого боку, кандидати, які сприймають інтерв'юера як кваліфікованого працівника, можуть відчувати себе вільніше та комфортніше під час обговорень, отже, насправді краще справляються зі співбесідою.

3. Етап після співбесіди. Після проведення співбесіди рекрутер зазвичай складає оцінку кваліфікації кандидата на вакантну посаду. Інтерв'юер, який проводить співбесіду, ретельно досліджує всю доступну інформацію про кандидата, включаючи його резюме, рекомендації та професійний досвід. Всю цю інформацію інтерв'юер поєднує, щоб сформуванати оцінку кандидата на посаду.

На завершальному етапі співбесіди інтерв'юер використовує свою оцінку кандидата, щоб прийняти остаточне рішення щодо прийому на роботу. Для уточнення оцінки кандидата іноді використовуються додаткові інструменти

відбору, такі як зразки роботи, тести когнітивних здібностей, тести особистості. Після того, як були зібрані всі необхідні дані, інтерв'юер може прийняти рішення щодо прийому кандидата на роботу.

Співбесіда є взаємним обміном інформацією, а кандидати також мають можливість прийняти рішення, чи хочуть вони працювати в даній компанії. В цілому, процес співбесіди – це складний процес, який починається з формування вражень та збору інформації з обох сторін та закінчується остаточним рішенням інтерв'юера щодо прийому на роботу кандидата. Важливо зазначити, що співбесіда – це лише один з інструментів відбору кандидатів, інші інструменти, такі як тести та зразки роботи, також можуть бути використані для отримання додаткової інформації про кандидата.

Типи запитань на співбесіді

Незалежно від формату співбесіди, інтерв'юери задають кандидатам кілька видів запитань. Два основних види, які часто вживаються та мають велику емпіричну базу, це ситуаційні запитання [33] та поведінкові запитання (також відомі як структуровані інтерв'ю на основі опису поведінки) [34]. Найкращі методи передбачають побудову обох видів питань на основі "критичних інцидентів", які важливі для виконання роботи [34], проте вони відрізняються за своїм фокусом. Критичні інциденти – це відповідні завдання, необхідні для виконання роботи, які можна зібрати шляхом проведення інтерв'ю або опитувань з поточними співробітниками, керівниками або експертами з предметної області [35, 10].

Ситуаційні запитання зосереджуються на гіпотетичних сценаріях, пов'язаних з майбутньою роботою кандидатів. Вони спрямовані на виявлення того, як кандидат може відреагувати на певні обставини, що можуть виникнути під час виконання роботи.

Поведінкові запитання, з іншого боку, звертаються до досвіду кандидата. Інтерв'юери ставлять запитання про конкретні ситуації, в яких кандидати опинялися раніше, і про те, як вони вирішували виниклі проблеми. Цей підхід дозволяє оцінити реакції кандидата на реальні ситуації, а також допомагає

зрозуміти, як вони можуть поводитися в майбутньому, засновуючись на їх минулому досвіді.

Обидва типи запитань мають свої переваги. Ситуаційні запитання допомагають оцінити потенціал кандидата в невідомих або нових ситуаціях, тоді як поведінкові запитання дають більш точне уявлення про реакцію кандидата на конкретні ситуації, з якими вони вже стикалися. Комбінування обох типів питань під час співбесіди дозволяє отримати більш повноцінне розуміння здібностей, навичок та ставлення до роботи кандидата.

1.2 Огляд існуючих систем для відбору кадрів

Рекрутери використовують різноманітні інструменти для організації та менеджменту процесу відбору кадрів. ATS (Applicant Tracking System) – система відслідковування кандидатів, що дозволяє рекрутерам зберігати та організовувати інформацію про кандидатів, що подали заявки на вакансії. Функціонал ATS: пошук резюме, їх імпорт у власну базу, публікація вакансій, зв'язок із кандидатом, планування співбесід, повідомлення претендентам про відмову або пропозицію про співпрацю, найм працівників по всьому світу за допомогою електронного підпису. Недоліком системи ATS є ризик, що під час автоматизованої перевірки та прийняття рішення щодо відбору та найму кандидатів у компанію система може усунути хороших кандидатів [36].

Менеджери з найму майже завжди для пошуку кандидатів використовують дошки оголошень – сайти, де компанії можуть розміщувати вакансії, а кандидати можуть шукати роботу та подавати заявки на них. На сьогодні популярним та ефективним інструментом для пошуку і залучення фахівців є соціальні медіа, такі як LinkedIn [37]. Рекрутери можуть використовувати ці мережі для пошуку та зв'язку з потенційними кандидатами, а також для просування вакансії. Функціонал LinkedIn дозволяє бути представленими через наявні контакти та розширювати зв'язки; здійснювати пошук компаній, людей, груп за інтересами; публікувати професійні резюме і здійснювати пошук роботи; рекомендувати й бути рекомендованими; публікувати вакансії. Недоліки LinkedIn для управління

процесом відбору кадрів: відсутність можливості планування співбесід, найму працівників за допомогою електронного підпису, автоматизованого прийняття рішення щодо відбору та найму кандидатів у компанію, автоматичного формування повідомлень претендентам про відмову або пропозицію про співпрацю.

Системи рекрутингу Hurma [38] та Cleverstaff [39] розроблені в Україні, реалізують функції, що пов'язані із вакансіями (активні, заявки, архів, статуси), здійснюють інтеграцію з сайтами пошуку роботи та електронною поштою для взаємодії з кандидатом через пошту, мають сорсинг-плагіни для пошуку кандидатів в різних джерелах, можуть додавати події, які автоматично синхронізуються з Google-календарем, реалізують реферальний рекрутинг для роботи з відгуками на вакансії за рекомендаціями працівників, формування статистичних звітів. Недоліком систем є відсутність планування співбесід, автоматизованого прийняття рішень щодо прийняття кандидата або відмови.

Системи відеоінтерв'ювання – це програмне забезпечення, яке дозволяє проводити віддалене відеоінтерв'ю з кандидатами на вакансії (Hirevue, Spark Hire, Talview). Часто це може значно спростити процес відбору претендентів, зменшити часові витрати на організацію зустрічей.

Для тестування та оцінювання претендентів рекрутери можуть використовувати різні письмові тести, тести на комп'ютері, рольові ігри та інші методи оцінки.

Оглянемо декілька існуючих програмних інструментів, які допомагають оцінювати інтерв'ю. Вони використовуються для аналізу та оцінки інтерв'ю, що дозволяє менеджерам та спеціалістам з кадрів виявляти кращих кандидатів. Інструменти для оцінки інтерв'ю можна класифікувати за рядом критеріїв, залежно від їхніх основних функцій, типів оцінки та технологій, які вони використовують. Можна виділити три основні напрямки класифікації: за типом інтерв'ю, за типом оцінки та за використаним технологіям.

За типом інтерв'ю:

- відеоінтерв'ю (наприклад, HireVue, Spark Hire, Talview);

- текстові інтерв'ю (наприклад, Predictivehire, Xor);
- ігрові та інтерактивні тести (наприклад, Pymetrics, Harver).

За типом оцінки:

- оцінка технічних навичок (наприклад, Interview Mocha);
- оцінка soft-skills (наприклад, HireVue, Plum, Talview);
- оцінка когнітивних та емоційних характеристик (наприклад, Pymetrics, Plum);
- оцінка культурної відповідності (наприклад, HireVue, Plum, PredictiveHire).

За використанням технологій:

- інструменти з використанням штучного інтелекту (наприклад, HireVue, Pymetrics, Plum, PredictiveHire, Harver, Talview, XOR);
- інструменти без використання штучного інтелекту (наприклад, Spark Hire, Interview Mocha).

Розглянемо детальніше призначення та можливості програмних інструментів для проведення інтерв'ю:

- HireVue [40]: це відеоінтерв'ю платформа, яка використовує штучний інтелект та аналіз даних для оцінки потенційних працівників. Особливістю HireVue є можливість вимірювати візуальні та аудіо сигнали, а також аналізувати відповіді кандидатів для визначення їх кваліфікації та придатності для вакантної посади. Недоліки платформи: існує ризик, що кандидати, які не знають як поводитися на інтерв'ю отримають менші оцінки, адже алгоритми машинного навчання аналізують окрім змісту мовлення – тон, жестикуляцію та інші параметри. Також недоліком є відсутність особистого контакту між кандидатом та рекрутером, що може призвести до втрати потенційних кандидатів.
- Spark Hire [41]: платформа відеоінтерв'ю з можливістю запису інтерв'ю та їх аналізу. Spark Hire дозволяє спеціалістам з кадрів порівнювати відповіді кандидатів на однакові запитання, що сприяє об'єктивній

оцінці. Недоліки платформи: висока вартість в порівнянні з іншими системами відеоінтерв'ю; система не відображає дійсний досвід роботи, відеоінтерв'ю можуть надавати високоякісну інформацію про кандидата, але не можуть повністю відображати досвід роботи та здібності, які можуть бути важливими для конкретної вакансії.

- Pymetrics [42]: цей інструмент використовує графічні моделі для оцінки кандидатів на основі їх когнітивних та емоційних характеристик. Він допомагає виявляти кандидатів, які підходять для конкретної організації та ролі. Недоліки платформи: система Pymetrics використовує складні алгоритми машинного навчання, що може ускладнити розуміння того, як саме відбувається оцінювання кандидатів. Не є універсальним інструментом для відбору персоналу, не враховує інші аспекти, такі як досвід та навички, також важливі для прийняття рішення про найняття кандидата.
- Plum [43]: цей інструмент аналізує різні аспекти особистості кандидата, такі як мотивація, соціальні навички та креативність, для оцінки його придатності до роботи. Plum використовує алгоритми, які допомагають знайти найкращих кандидатів. Недоліки платформи: обмежений набір інструментів, що може призвести до зниження точності оцінювання. Для роботи системи Plum необхідно мати великий обсяг даних про кандидатів, що може бути витратним та вимагати багато часу та зусиль на збір та обробку цих даних.
- Sapia [44]: ця платформа використовує машинне навчання для аналізу відповідей на запитання. Недоліки платформи: Порівняно з деякими іншими системами для відбору персоналу, Sapia може мати обмежену функціональність, особливо щодо інтеграції з іншими інструментами та системами. Порівняно з деякими іншими системами для відбору персоналу, Sapia може мати обмежену функціональність, особливо щодо інтеграції з іншими інструментами та системами.

Результати порівняльного аналізу програмних інструментів для оцінки інтерв'ю подані в таблиці 1, яка демонструє основні характеристики та можливості цих програмних рішень:

Таблиця 1 – Характеристики програмних інструментів для оцінки інтерв'ю

Інструмент	Відеоінтерв'ю	Технічні навички	Софт-навички	Когнітивні та емоційні характеристики	Культурна відповідність	Штучний інтелект
HireVue	Так	Ні	Так	Так	Так	Так
Spark Hire	Так	Ні	Так	Ні	Ні	Ні
Pymetrics	Ні	Ні	Так	Так	Так	Так
Plum	Ні	Ні	Так	Так	Так	Так
PredictiveHire	Ні	Ні	Так	Ні	Так	Так
Harver	Ні	Так	Так	Так	Так	Так
Interview Mocha	Ні	Так	Ні	Ні	Ні	Ні
Talview	Так	Ні	Так	Так	Так	Так
XOR	Так	Ні	Так	Ні	Так	Так

Підсумовуючи, можна сказати, що програмні системи для оцінки інтерв'ю мають ряд переваг і недоліків. Важливо розуміти ці аспекти, щоб знати, як розробляти, покращувати та підтримувати такі інструменти.

1.3 Постановка задачі

Призначення системи: інформаційна та програмна підтримка процесу рекрутингу із проведенням співбесід для відбору персоналу та спрощення процесу пошуку кандидатів на вакансії.

Цілі створення системи: оптимізація часу та ресурсів необхідних для залучення персоналу в компанію, підвищення ефективності відбору, зменшення суб'єктивної оцінки кандидатів на вакансії через людський фактор, прогнозування подальшої успішності кандидатів.

Для досягнення поставлених цілей програмна система має реалізувати такі задачі:

1. Здійснення загального керування процесом управління персоналом в рамках стратегії розвитку компанії і підвищення ефективності бізнесу.

2. Менеджмент вакансій і кандидатів:

- a. Формування вакансій
- b. Визначення етапів відбору кандидатів
- c. Формування та ведення бази даних вакансій
- d. Розробка шаблону резюме
- e. Формування шаблону для співбесіди
- f. Планування подій для кандидатів та менеджерів з найму
- g. Формування та ведення бази даних кандидатів

3. Автоматизація процесу проведення співбесід на основі інформаційної та програмної підтримки:

- a. Формування розкладу співбесід для кандидата;
- b. Визначення результатів співбесід;
- c. Оцінювання результатів співбесід з використанням алгоритмів машинного навчання.

4. Аналіз результатів співбесід та рекрутингового процесу, формування звітності:

- a. Стратегія розвитку компанії в контексті розвитку персоналу
- b. Статистика відбору кандидатів
- c. Календарне планування проведення співбесід
- d. Результати оцінювання кандидатів в процесі проведення співбесід

5. Сервісні функції:

- a. Планування подій
- b. Обробка аудіо- та відеопотоку конференції, обробка голосу та збереження для подальшого аналізу.

с. Забезпечення комунікацій і зворотного зв'язку через електронну пошту, месенджери та інші засоби телекомунікацій

Функціональна структура системи, що розроблятиметься, подана IDEF0 діаграмами за методологією SADT (Structured Analysis and Design Technique). На рис.1. подана контекстна діаграма, рис.2 та рис.3 ілюструють декомпозицію системи з деталізацією її функціональності.

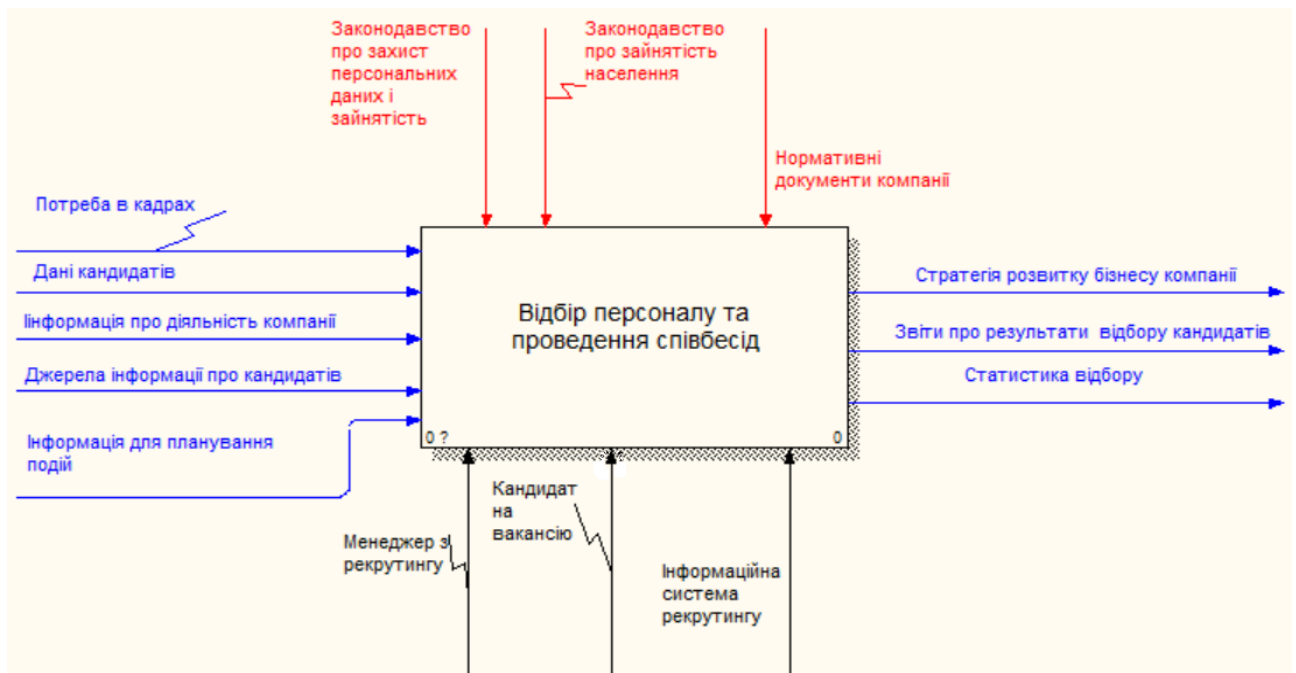


Рисунок 1 – IDEF0 контекстна діаграма інформаційної системи відбору персоналу та проведення співбесід

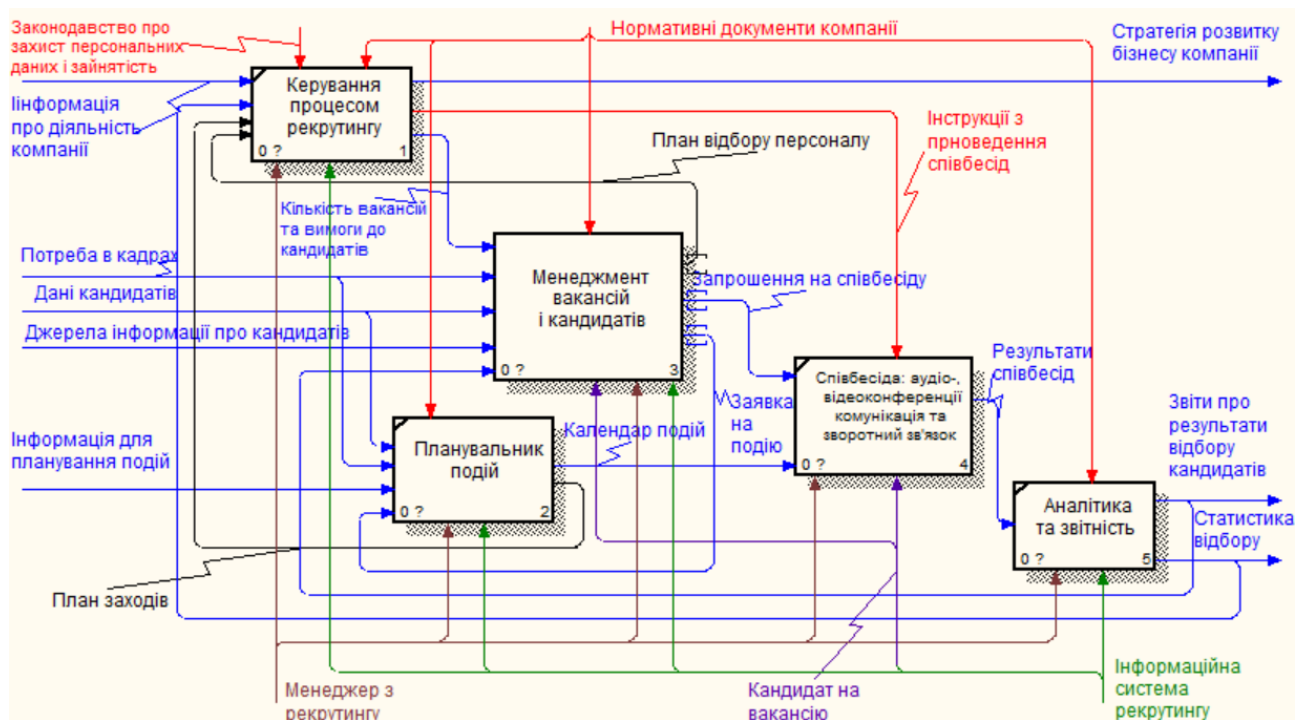


Рисунок 2 – IDEF0 діаграма бізнес-процесів інформаційної системи відбору персоналу та проведення співбесід

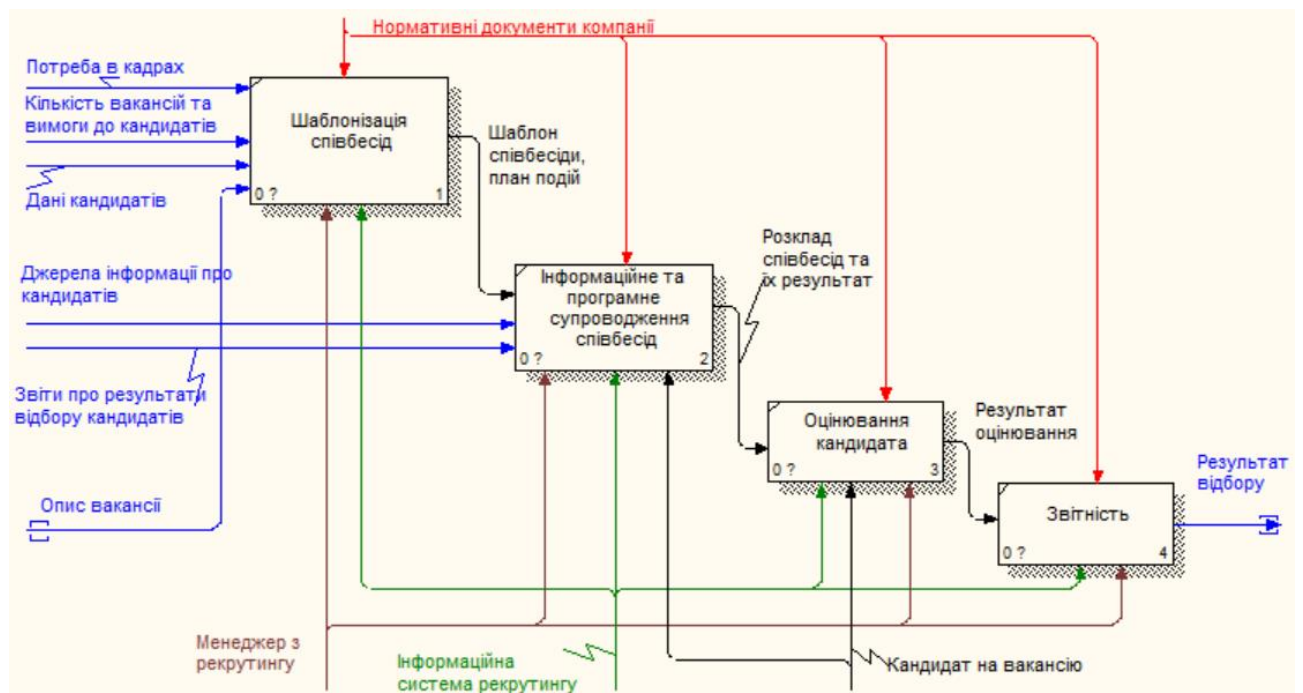


Рисунок 3 – IDEF0 діаграма бізнес-процесів на етапі менеджменту вакансій і кандидатів

1.4 Математична формалізація задачі

Задача полягає в оптимізації витрат часу та ресурсів, які компанії витрачають на підбір персоналу. Це важливий аспект для багатьох організацій, оскільки ефективний процес найму може значно зекономити час і кошти, а також забезпечити належну якість відбору кандидатів.

Один зі способів досягнення оптимального використання ресурсів і максимізації продуктивності в цій області полягає у використанні методів машинного навчання. Ці методи можуть допомогти розробити оптимальні розклади роботи менеджерів з найму та забезпечити ефективне управління процесом відбору, де потрібно ефективно розподіляти час та ресурси. Оптимізація розкладу полягає в такому розподілі різних завдань або ресурсів у часовому просторі, щоб досягти поставлених цілей, максимізувати продуктивність та мінімізувати затрати.

1.4.1 Математична модель задачі

Для формалізації задачі оптимізації витрат часу та ресурсів у процесі підбору персоналу можна використовувати математичні моделі. Задача полягає в мінімізації функції F , яка включає витрати на підбір персоналу для кожної вакансії, з дотриманням обмежень щодо часу та ресурсів. Формула для даної задачі може бути записана наступним чином:

$F = \sum(x * w)$ – оптимізаційна функція, що представляє суму витрат на підбір персоналу для всіх вакансій.

де:

x – булева змінна, яка приймає значення 1, якщо кандидат відповідає вимогам вакансії, або 0 - у протилежному випадку.

w – вага кандидата для вакансії, що відображає його рейтинг або оцінку зі співбесіди.

$\sum(x * t) \leq T_{max}$ – сумарний час, витрачений на розгляд кандидатів для всіх вакансій, не повинен перевищувати максимально допустимий час.

де:

t – час, витрачений на розгляд кандидата для вакансії.

T_{max} – максимально допустимий час, який можна витратити на відбір персоналу.

Метою оптимізації є знайти значення змінних x , які мінімізують функцію F і водночас задовольняють обмеженням щодо часу. Це дозволить ефективно розподілити ресурси та забезпечити оптимальний підбір персоналу для компанії.

1.4.2 Методи машинного навчання, що можуть бути застосовані для оцінки співбесід

Для перевірки відповідності кандидата певним критеріям, наприклад, чи він має достатньо досвіду або навичок для вакансії, може бути застосований алгоритм класифікації, а саме алгоритм дерева ухвалення рішень. Він дозволяє побудувати модель, яка дає оцінку параметрам та приймає рішення на основі набору правил, що розділяють вхідні дані на різні класи.

Також корисним буде використання алгоритмів кластеризації, які згрупувають кандидатів за спільними характеристиками такі як навички та досвід.

Висновки розділу 1

В розділі 1 проаналізовані переваги автоматизації процесу відбору персоналу та проведення співбесід; визначені функції інформаційних систем для рекрутингу персоналу та проведення співбесід, а саме: рекрутинг, відбір, керування кандидатами, аналітика; визначені загальні вимоги до інформаційної системи відбору персоналу та проведення співбесід, зокрема легкість у використанні та можливість інтеграції з іншими програмними засобами

компанії, гнучкість та адаптивність до потреб конкретної компанії та галузі, допустимий рівень безпеки та конфіденційності даних.

Для проведення ефективного та об'єктивного інтерв'ю з кандидатами рекомендується створити чіткий план співбесіди, який включає як ситуаційні, так і поведінкові запитання, які безпосередньо пов'язані з критичними інцидентами та компетентностями, необхідними для успішного виконання роботи.

РОЗДІЛ 2. АНАЛІЗ АРХІТЕКТУРНИХ ТА ТЕХНІЧНИХ РІШЕНЬ

Сучасний світ відзначається стрімким розвитком інформаційних технологій, що стають все більш важливим інструментом для організацій різного рівня. Важливим аспектом створення ефективної інформаційної системи є ретельний аналіз архітектурних та технічних рішень, які визначають структуру системи, взаємодію компонентів та відповідність вимогам замовника.

Даний розділ присвячений дослідженню та аналізу різних архітектурних та технічних рішень, які можуть бути використані для розробки інформаційної системи. Метою даного розділу є визначення найбільш оптимальних та ефективних рішень, які забезпечать надійність, масштабованість та продуктивність системи.

У подальшому, обрані архітектурні та технічні рішення будуть слугувати основою для планування та розробки інформаційної системи. Вони також стануть важливими вихідними даними для розробки проектної документації, формування команди розробників, визначення технічних ресурсів та планування часових рамок реалізації проекту.

У підсумку, аналіз архітектурних та технічних рішень є важливим етапом у процесі розробки інформаційної системи, що дозволяє вибрати оптимальні рішення для досягнення мети проекту та забезпечення високої якості кінцевого продукту.

2.1 Огляд архітектурних підходів

Архітектура програмної системи – це певна структура, що надається системі творцями, вона утворюється розподілом системи на компоненти, їх організація та визначення способів взаємодії між ними. Метою архітектури є спрощення розробки, розгортання та супроводу програмної системи в продовж її життєвого циклу. Як зазначив Роберт Мартін [45]: «Головна стратегія такого спрощення полягає в тому, щоб якомога довше мати найбільше варіантів».

Огляд архітектурних підходів дозволяє зрозуміти, які стратегії розробки та структури можуть бути використані для побудови інформаційної системи. Розглянемо детально чотири підходи: монолітна архітектура, мікросервісна архітектура, доменна архітектура та чиста архітектура.

Монолітна архітектура (Monolithic Architecture) додатку представляє собою систему, як єдиний загальний модуль, що поставляється через єдине розгортання. Основними перевагами такої архітектури додатку є: простота розробки та розгортання, легкість управління кодом та ресурсами, низькі вимоги до комунікації між компонентами, що значно впливає на вартість розробки на початкових етапах реалізації продукту. В свою чергу така архітектура має і недоліки, а саме: такі додатки важко та дорого масштабувати, повільно оновлюються, компоненти мають жорстку залежність один від одного, що ускладнює розвиток та підтримку системи. Прикладом використання таких систем є невеликі та середні веб-сайти, внутрішні корпоративні системи, прототипи.

Мікросервісна архітектура (Microservices Architecture) представляється у вигляді невеликих служб (сервісів), які мають незалежні точки розгортання. Перевагами такого підходу є: незалежність та гнучкість компонентів, легкість масштабування, можливість використання різних технологій для різних сервісів, широкі можливості паралелізації роботи та розподілення навантаження. Основними недоліками є: складність управління та моніторингу, високі вимоги до комунікації між сервісами, можливі проблеми з безпекою та стабільністю. Прикладами використання можуть бути: великі та складні веб-проекти, системи з великим обсягом користувачів або даних, системи з високими вимогами до масштабування.

Доменна архітектура (Domain-driven design, DDD) фокусується на бізнес-логіці та доменних правилах, сприяє створенню модульної структури з незалежними компонентами, підвищує зрозумілість системи та сприяє легкій співпраці між розробниками та зацікавленими сторонами. Як і всі інші має недоліки, її важко застосувати для простих (невеликих) проєктів або проєктів без

чітко сформульованих бізнес-правил, вимагає глибокого розуміння доменної області та бізнес-потреб. Прикладами використання є складні бізнес-системи, інформаційні системи з високими вимогами до моделювання доменної області та бізнес-правил.

Чиста архітектура (Clean Architecture) – це концепція проектування архітектури програмного забезпечення, запропонована Робертом Мартіном. Ця концепція базується на принципах SOLID та передбачає розділення логіки додатку на чітко визначені компоненти, що дозволяє забезпечити гнучкість та легкість у внесенні змін до системи, зберігаючи при цьому її якість та продуктивність. Являє собою компіляцію ідей щодо організації архітектури, а саме[45]:

- Гексагональна архітектура, розроблена Алістером Кобкерном та описана Стівом Фріманом і Натом Прайсом [46];
- DCI, запропонована Джеймом Коплієном і Трюгве Ресенскаугом;
- BCE, запропонована Іваром Якобсоном [47].

Така архітектура має багато переваг: незалежність від фреймворків, простота тестування, незалежність від інтерфейсу користувача, незалежність від бази даних та будь-яких інших зовнішніх агентів. Також варто відзначити: розділення відповідальності між компонентами, гнучкість та легкість зміни або заміни компонентів системи. З іншого боку, на початкових етапах розробки виникають складнощі з розробкою та налаштуванням системи, також досить складно керувати залежностями, має негативний вплив на продуктивність та швидкість розробки. Приклади використання: середні та великі проекти з високими вимогами до гнучкості, тестування та якості коду, проекти, які можуть зазнавати значних змін у вимогах чи технологіях протягом їх життєвого циклу.

Обираючи архітектурні підходи для інформаційної системи, важливо враховувати специфіку та вимоги конкретного проекту, ресурси, доступні для розробки та досвід розробників. Правильний вибір може суттєво підвищити ефективність розробки, стабільність та масштабованість інформаційної системи.

2.2 Аналіз вимог та визначення ключових характеристик програмної системи

Аналіз вимог інформаційної системи є критично важливим кроком у процесі розробки інформаційної системи, оскільки допомагає визначити ключові характеристики та функціональність, які повинні бути виконані для досягнення бізнес-цілей та врахування потреб користувачів.

Проаналізуємо основні функціональні вимоги та ключові характеристики для інформаційної системи проведення співбесід:

- a) Клієнт має змогу зареєструвати компанію в системі.
 - 1. Клієнту надається форма реєстрації компанії.
 - 2. Система надає можливість зареєструвати компанію після перевірки введених даних та приймає умови ліцензії на використання ПЗ.
- b) Клієнт має змогу авторизуватися в системі як компанія.
 - 1. Клієнту надається форма авторизації.
 - 2. Система надає клієнту можливість увійти в систему після валідації даних.
- c) Клієнт має змогу керувати оплатою та обирати план використання.
 - 1. Клієнт може обрати необхідний план використання системи з відповідними обмеженнями.
 - 2. Клієнт може керувати оплатою за необхідний план використання системи за допомогою інтегрованих платіжних сервісів.
- d) Компанія має змогу додавати інтерв'юерів.
 - 1. Система надає форму для заповнення інформації інтерв'юера.
 - 2. Компанія має змогу ввести інформацію інтерв'юера (ім'я, електронну адресу, пароль) та обрати одну з доступних ролей.
- e) Компанія може переглядати, редагувати та видаляти профілі інтерв'юерів.
 - 1. Компанії надається інтерфейс для керування профілями інтерв'юерів.

2. Компанії надається графічний інтерфейс для перегляду статусів виконання роботи інтерв'юерами.
- f) Компанія має можливість керувати вакансіями.
1. Система надає інтерв'юеру форму для заповнення інформації про вакансію.
 2. Компанія має змогу ввести інформацію кандидата (назва, опис, вимоги) та обрати одну з доступних ролей.
 3. Компанія має можливість переглядати вакансії.
 4. Компанія має можливість додавати та редагувати вимоги до вакансії.
 5. Компанія призначає інтерв'юерів для роботи з вакансією.
- g) Інтерв'юер має можливість працювати з вакансіями.
1. Інтерв'юер може переглядати призначені компанією вакансії.
 2. Інтерв'юер може переглядати вимоги до вакансії.
 3. Інтерв'юер має можливість додавати кандидатів до вакансії.
- h) Інтерв'юер додає кандидатів до вакансії.
1. Система надає інтерв'юеру форму для заповнення інформації про кандидата.
 2. Інтерв'юер має змогу ввести інформацію кандидата (ім'я, електронну адресу, CV) та обрати вакансію.
 3. Інтерв'юер має змогу планувати час співбесід з кандидатом за допомогою планувальника.
- i) Інтерв'юер має можливість керувати співбесідами.
1. Інтерв'юер може створювати, оновлювати, переглядати та видаляти шаблони співбесід за допомогою графічного інтерфейсу.
 2. Інтерв'юер має можливість призначати час для проходження співбесід кандидатами.
 3. Інтерв'юер має можливість розпочати процес проходження співбесіди з кандидатом використовуючи шаблон інтерв'ю.

4. Під час проходження співбесіди інтерв'юер отримує можливість робити необхідні для оцінювання коментарі на кожне питання.
 5. Інтерв'юер отримує можливість оцінювати кандидата.
- j) Інтерв'юер має змогу планувати події.
1. Інтерв'юеру надається інтерфейс для налаштування та планування подій.
 2. Інтерв'юер визначає назву, час та учасників події.
 3. Після створення події система надсилає сповіщення на електронну адресу учасників події.
- k) Кандидат має змогу обмежено взаємодіяти з системою.
1. Кандидат повинен дати згоду на використання ПЗ, посилення на яке він отримує на пошту після внесення його в систему інтерв'юером компанії.
 2. Кандидат отримує можливість переглядати звіти про проходження співбесід за допомогою електронної пошти.
 3. Кандидат отримує можливість переглядати звіти про проходження співбесід за допомогою графічного інтерфейсу системи.
 4. Кандидат отримує сповіщення про етапи відбору на електронну пошту.
- l) Кандидат за допомогою електронної пошти має можливість відмовитися від наступних кроків проходження інтерв'ю.

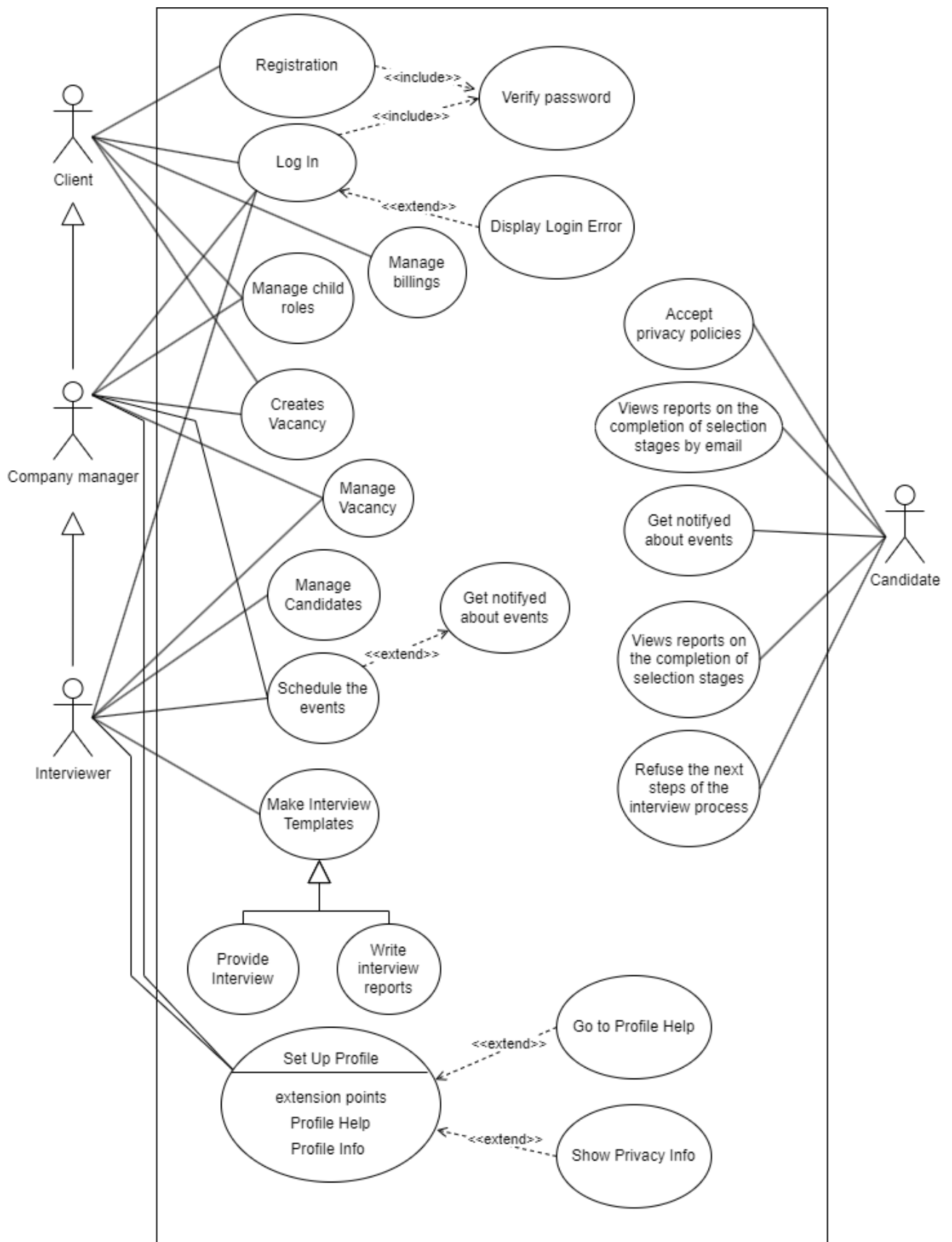


Рисунок 4 – The Use-Case View діаграма

Також варто визначити нефункціональні вимоги, які визначають якість системи та вимоги до її роботи:

- **продуктивність:** система повинна забезпечувати швидке завантаження та відгук на дії користувачів. Відповідь на запити користувачів має бути не більше 1-2 секунд;
- **масштабованість:** система повинна мати можливості до масштабування, щоб підтримувати збільшення кількості користувачів та обсягу даних без втрати продуктивності;
- **надійність:** система повинна гарантувати стабільну роботу та забезпечувати захист від відмов. Рівень доступності (uptime) має становити 99,9% або більше;
- **безпека:** інформаційна система повинна забезпечувати безпеку даних користувачів та захист від несанкціонованого доступу, втручання та атак. Система повинна відповідати стандартам безпеки та використовувати сучасні методи шифрування та аутентифікації;
- **сумісність:** система повинна бути сумісною з різними операційними системами та браузерами, а також підтримувати адаптивний дизайн для зручного використання на мобільних пристроях;
- **легкість використання:** інтерфейс системи повинен бути інтуїтивно зрозумілим та зручним для користувачів з різним досвідом та рівнем компетенції;
- **модульність:** архітектура системи має бути гнучкою та модульною, що дозволить легко розширювати та модифікувати функціональність системи в майбутньому без втрати стабільності та продуктивності. Варто розглянути можливість розширення функціоналу інформаційної системи до інструменту до багатофункціонального HR-інструменту;
- **підтримка та оновлення:** система повинна мати можливість легкого оновлення та підтримки, що забезпечить своєчасне усунення помилок та внесення поліпшень відповідно до змін у вимогах користувачів та ринку;
- **документація та навчання:** інформаційна система повинна мати належну документацію для користувачів та розробників, яка допоможе

зрозуміти принципи роботи системи та навчитися її використовувати. Крім того, має бути передбачена підтримка та навчальні матеріали для співробітників компаній, які будуть використовувати систему.

Враховуючи ці вимоги, інформаційна система буде ефективною, стабільною, безпечною та зручною для користувачів. Важливо постійно відслідковувати та оцінювати роботу системи, щоб вчасно виявляти та усувати проблеми, а також вдосконалювати її функціональність згідно зі змінами у потребах користувачів та бізнесу.

2.3 Огляд технологій, методів та інструментів розробки

Найбільш значущі фактори для вибору технологій є наступні фактори: вимоги до проєкту, тренд розвитку технологій, наявність готових до використання рішень тощо.

На основі цього для реалізації інформаційної системи обрано мову TypeScript та технології Node.js, React, Express, TypeORM. PostgreSQL в якості БД – об'єктно-реляційна база даних з відкритим вихідним кодом [48].

Основною перевагою такого набору технологій є спільна мова програмування для реалізації клієнтської та серверної частини.

TypeScript – це строго типізована надбудова над JavaScript, розроблена компанією Microsoft, яка розширює можливості JavaScript, додаючи типи та інші функції. Використання TypeScript для реалізації інформаційної системи для проведення співбесід може принести ряд переваг, які будуть обґрунтовані нижче.

- строга типізація: TypeScript використовує статичну типізацію, яка допомагає розробникам забезпечити більш безпечний та надійний код. Завдяки строгій типізації можна виявити багато потенційних проблем ще на етапі розробки та написання коду, що зменшує кількість помилок у фінальній версії продукту;
- сумісність з JavaScript: TypeScript повністю сумісний з JavaScript, що дозволяє легко інтегрувати TypeScript у існуючі JavaScript-проєкти або

використовувати весь спектр наявних JavaScript-бібліотек та фреймворків;

- підтримка об'єктно-орієнтованого програмування: TypeScript дозволяє використовувати принципи об'єктно-орієнтованого програмування (ООП), такі як класи, інтерфейси та спадкування, що сприяє створенню чистого, модульного та легко підтримуваного коду;
- розвинуті інструменти розробки: TypeScript підтримується багатьма популярними середовищами розробки (IDE), такими як Visual Studio Code, WebStorm тощо. Це забезпечує зручність роботи з кодом, автодоповнення, підказки щодо типів та виявлення помилок в реальному часі;
- широка спільнота: TypeScript має активну та широку спільноту розробників, яка постійно працює над покращенням мови та документації. Це забезпечує доступ до численних ресурсів, включаючи навчальні матеріали, приклади коду, бібліотеки та фреймворки, а також швидке вирішення проблем та підтримку від спільноти;
- висока продуктивність: TypeScript забезпечує кращу продуктивність розробки порівняно з JavaScript завдяки строгій типізації, покращеному автодоповненню, рефакторингу та іншим можливостям, які допомагають розробникам швидше писати та налагоджувати код.

В якості системи контролю версій використовується Git, що є одним з найпопулярніших та найбільш надійних інструментів для керування версіями коду в проектах розробки програмного забезпечення. Використання Git при розробці інформаційної системи має ряд переваг. Ефективне керування версіями, що дозволяє відслідковувати зміни у коді та легко перевіряти історію змін. Можливість створення гілок для окремих завдань полегшує роботу над різними функціями чи виправленнями помилок без впливу на основний код. Git використовує хеш-функцію SHA-1 для ідентифікації змін, що забезпечує надійність та цілісність даних. Крім того, розробники можуть використовувати

відмітки (теги) для позначення важливих версій, що спрощує контроль версій та розгортання.

Отже, враховуючи переваги, TypeScript може виявитися оптимальним вибором для реалізації інформаційної системи для проведення співбесід.

2.4 Розробка стратегії реалізації програмної системи

Для реалізації програмної системи необхідно визначити ключові етапи реалізації проєкту:

1. Аналіз вимог: визначення основних вимог до проєкту, його мети і обсягу. Встановлення потреб користувачів і визначення бізнес-цілей проєкту.
2. Проектування архітектури: розроблення загальної архітектури проєкту, включаючи вибір технологій, платформи, інфраструктури та структури додатку. Планування бази даних і інтеграції компонентів системи.
3. Розробка: реалізація програмного коду і функціональності проєкту згідно з вимогами та специфікаціями. Розробка модулів, компонентів і інтеграція системи.
4. Тестування: виконання тестів для перевірки функціональності, коректності і надійності системи. Виявлення і виправлення помилок, оптимізація роботи системи.
5. Розгортання: підготовка системи до впровадження, включаючи налаштування середовища, міграцію даних, установку та конфігурацію програмного забезпечення.
6. Експлуатація та підтримка: після впровадження системи забезпечення постійної підтримки, усунення помилок, вдосконалення та розширення функціональності відповідно до потреб користувачів.
7. Оцінка проєкту: аналіз результатів проєкту, оцінка відповідності вимогам і досягнення поставлених цілей. Визначення успіху проєкту та можливих напрямків подальшого розвитку.

Висновки розділу 2

В розділі 2 оглянуті та проаналізовані переваги та недоліки архітектурних підходів, що застосовуються для розробки програмних систем. Розроблено функціональні та нефункціональні вимоги до системи. Визначено основні дійові особи, які будуть використовувати систему та описано варіанти використання системи.

На основі аналізу вимог визначено необхідні технології та інструменти для реалізації інформаційної системи та описано рекомендації щодо подальших кроків у процесі розробки, включаючи планування, проектування, програмування, тестування та впровадження системи.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ КРОСПЛАТФОРМЕННОГО ВЕБ-ЗАСТОСУНКУ ДЛЯ ПРОВЕДЕННЯ СПІВБЕСІД

У сучасному світі, коли технології розвиваються надзвичайно швидко, роль кросплатформених веб-застосунків у сфері бізнесу, освіти та повсякденного життя стає все більш важливою. У контексті ринку праці, ефективне проведення співбесід є одним із ключових чинників успішного підбору кандидатів та досягнення бізнес-цілей компаній. Отже, створення кросплатформеного веб-застосунку для проведення співбесід відіграє важливу роль в оптимізації процесу найму та забезпеченні якісного відбору кадрів.

У даному розділі роботи розглянуто процес реалізації кросплатформеного веб-застосунку для проведення співбесід. Основна мета цього розділу – детально описати етапи розробки та реалізації системи, включаючи проектування архітектури, вибір технологій та інструментів, розробку компонентів та їх взаємодію, а також тестування та впровадження системи.

3.1 Проектування архітектури

Проектування архітектури є одним з найважливіших етапів розробки програмного забезпечення, оскільки воно впливає на структуру, зручність розробки та підтримки, а також масштабованість системи. Відповідна архітектура допомагає забезпечити ефективне використання ресурсів, оптимізувати роботу системи, уникнути виникнення проблем з проектуванням та реалізацією коду.

Важливість проектування архітектури полягає у тому, що воно дозволяє розробникам створювати системи, які легко адаптуються до змін та вимог ринку, мають високий рівень надійності, пропускну здатність та доступність. Завдяки грамотно спроектованій архітектурі, код стає більш зрозумілим, модульним та легко тестованим, що сприяє ефективній роботі команди розробників.

Проектування архітектури кросплатформеного веб-застосунку для проведення співбесід вимагає комплексного підходу, враховуючи різні аспекти розробки, такі як модульність, шари додатка, забезпечення продуктивності, масштабованості та безпеки. У цьому підрозділі будуть розглянуті ключові компоненти архітектури та їх взаємодія.

3.1.1 Структура проекту

Інформаційна система для підбору кадрів має бути запроєктована як комплексне програмне забезпечення, яке допомагає підприємствам проводити ефективні співбесіди з кандидатами на вакансії. Основна мета інформаційної системи проведення співбесід полягає у спрощенні процесу відбору кандидатів, зниженні часових та фінансових витрат і забезпеченні високої якості кандидатів, які проходять співбесіду.

Загальна концепція інформаційної системи проведення співбесід полягає у створенні електронного сервісу для проведення співбесід між кандидатами та рекрутерами, який буде простим та зрозумілим для використання. Система повинна дозволяти рекрутерам вести звіти та вести статистику про виконану роботу. Кандидати повинні мати можливість переглянути відео своїх співбесід та зберігати їх для майбутнього використання. Основним функціоналом інформаційної системи проведення співбесід має бути підтримка відеоконференції, яка дозволить рекрутерам та кандидатам зв'язуватися один з одним з будь-якого місця з використанням інтернету.

Основні модулі інформаційної системи проведення співбесід повинні включати модуль передвідбору, який дозволить рекрутерам зібрати інформацію про кандидатів та оцінити їхні навички та досвід роботи.

Виходячи з вимог до програмного продукту, що була розглянута у другому розділі, проект можна розділити на такі основні модулі та компоненти:

- `.env` та `config.ts`: Файли конфігурації, що містять налаштування середовища та конфігурацію додатка.

- **src:** Головний каталог з вихідним кодом проекту, який містить підкаталоги для різних аспектів додатка.
- **db:** Каталог, що містить код, пов'язаний з роботою з базою даних PostgreSQL.
- **infrastructure:** Каталог, що містить важливі утиліти та міжпрограмне забезпечення для роботи з додатком, такі як обробка помилок та аутентифікація.
- **lib:** Каталог, що містить загальні моделі, валідатори та утиліти, які використовуються в усьому проекті.
- **module:** Каталог, що містить модулі різних функціональних аспектів додатка, таких як авторизація, шаблони співбесід, категорії питань, кроки, категорії кроків та ін. Кожен модуль містить підкаталоги для моделей, портів, використання випадків, валідаторів та репозиторіїв.

3.1.2 Модульність та розділення відповідальності

Щоб забезпечити гнучкість та стабільність системи, проект розроблено з використанням модульного підходу, який дозволяє розділяти функціональність на окремі частини. Це сприяє кращому розумінню коду, полегшує його тестування та підтримку. Розділення відповідальності між компонентами забезпечує незалежність різних частин системи та дозволяє легко вносити зміни або розширювати функціональність.

Архітектурні шари відіграють важливу роль у структуруванні коду та розділенні відповідальності між різними компонентами системи. У даному проекті використовується чотири основних шари:

1. Шар даних: Цей шар забезпечує взаємодію з базою даних PostgreSQL та іншими джерелами даних. Він містить наступні компоненти:
 - а. Моделі сутностей, які представляють структуру таблиць бази даних та відображають їх на класи у коді.

- b. Репозиторії, які відповідають за керування даними. Вони забезпечують методи для створення, оновлення, видалення та отримання даних з бази даних.
 - c. Джерело даних, яке реалізує взаємодію з базою даних та може бути замінено для роботи з іншими СУБД.
2. Шар доменної логіки: Цей шар містить бізнес-логіку додатка та реалізацію основних випадків використання. Його компоненти включають:
- a. Моделі домену, які представляють основні концепції системи та містять бізнес-правила.
 - b. Порти, які визначають інтерфейси для взаємодії між різними компонентами системи, такими як репозиторії, випадки використання та валідатори.
 - c. Випадки використання, які реалізують основні сценарії роботи системи, забезпечуючи координацію дій між різними компонентами.
 - d. Валідатори, які забезпечують перевірку даних на відповідність вимогам та бізнес-правилам.
3. Шар інфраструктури: Цей шар надає спільні утиліти та міжпрограмне забезпечення, необхідне для роботи системи. Інфраструктурний шар включає такі компоненти:
- a. Міжпрограмне забезпечення (middleware), яке забезпечує авторизацію та аутентифікацію користувачів, перевіряючи токени доступу та ролі користувачів перед обробкою запитів.
 - b. Утиліти, які допомагають в реалізації специфічної логіки або спрощують використання певних функцій, таких як створення та перевірка паролів, генерація та верифікація токенів, обробка помилок тощо.
4. Шар API: Цей шар відповідає за взаємодію з користувачами та сторонніми системами через веб-інтерфейс. Він містить наступні компоненти:
- a. Маршрутизатори, які визначають доступні ендпоінти для обробки HTTP-запитів та спрямовують їх до відповідних контролерів.

- б. Контролери, які викликають відповідні випадки використання та перетворюють результати у вигляді JSON-відповідей, які будуть відправлені користувачам або стороннім системам.
- с. Валідатори вхідних даних, які перевіряють коректність даних, надісланих користувачами або сторонніми системами, перед передачею їх до випадків використання.

Розділення коду на архітектурні шари сприяє зручності розробки, підтримки та масштабування системи. Кожен шар має відповідальність за окрему частину функціональності та відокремлений від інших шарів, що забезпечує низьку залежність між компонентами та полегшує рефакторинг або заміну окремих частин системи.

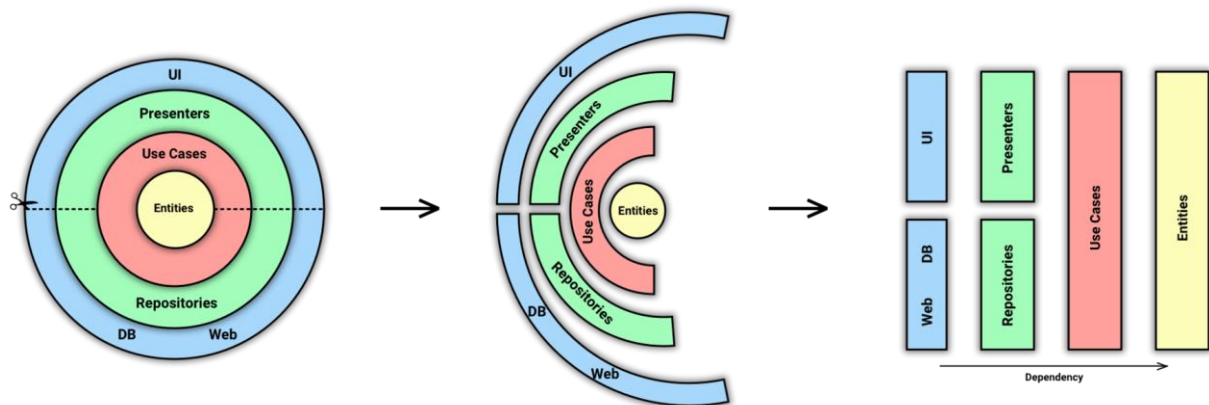


Рисунок 5 – Схема архітектури ПЗ

Також, важливо додати опис логічної структури БД, що має ключове значення в проектуванні, розробці та управлінні даними. Логічна структура БД визначає спосіб організації та взаємозв'язку даних в межах системи управління базами даних (СУБД). Вона відображає логічну модель даних, що використовується для представлення реального світу у вигляді таблиць, полів і зв'язків між ними.

Опис логічної структури БД включає в себе визначення сутностей (таблиць), атрибутів (стовпців), зв'язків і обмежень. Кожна сутність відповідає конкретному типу даних, який включає набір атрибутів для представлення властивостей цієї сутності. Атрибути визначають характеристики даних, такі як ім'я, тип, обмеження і т.д. Зв'язки встановлюють зв'язки між сутностями,

вказуючи на відношення між ними, наприклад, один до одного, один до багатьох або багато до багатьох.

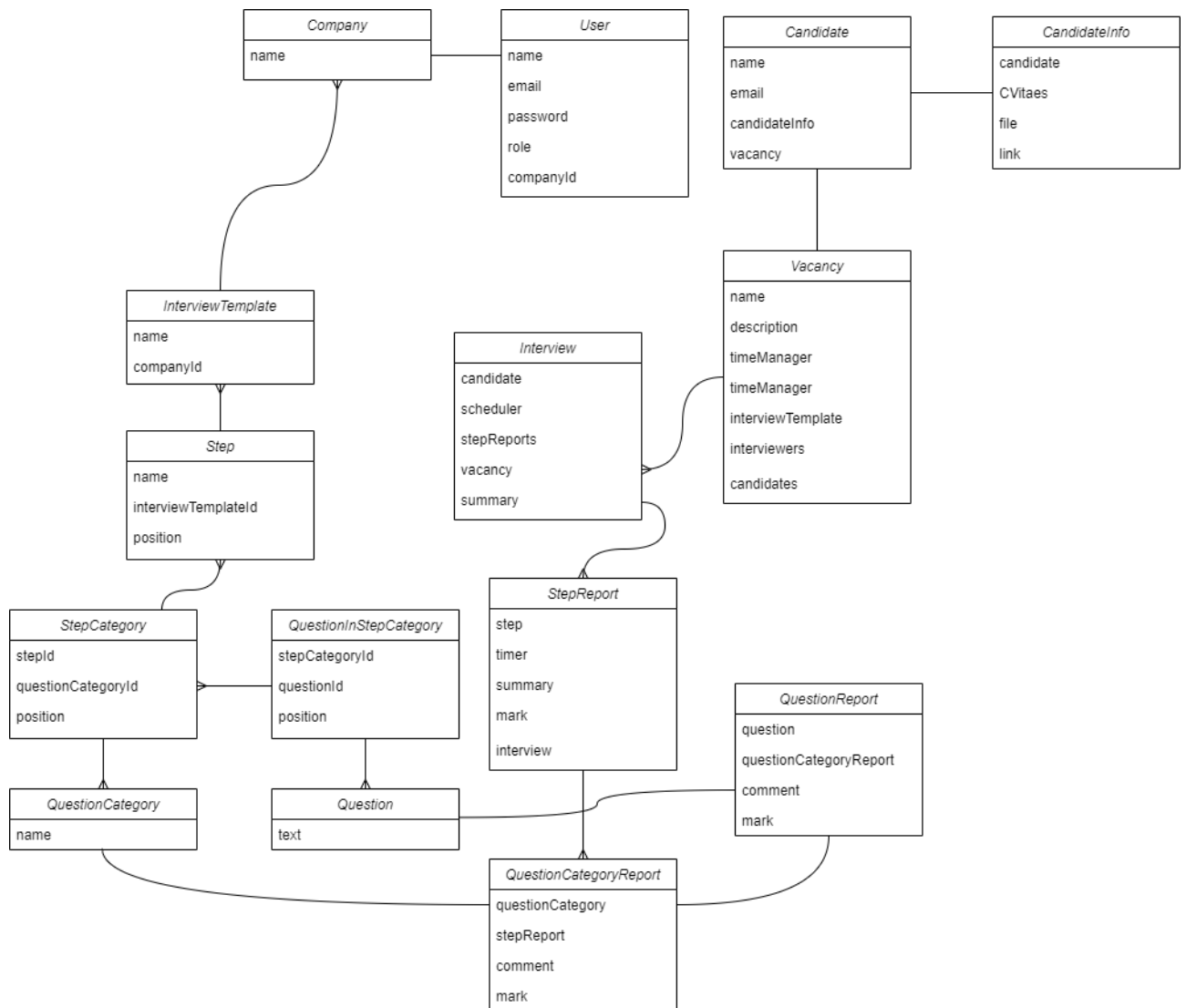


Рисунок 6 – Діаграма структури даних

3.2 Розробка модуля автентифікації

3.2.1 Аналіз вимог до модуля автентифікації

На початковому етапі розробки модуля автентифікації було визначено основні вимоги до його функціональності. Враховуючи особливості роботи веб-застосунку, модуль автентифікації мав забезпечувати наступні можливості:

1. Реєстрація нових користувачів, з можливістю вибору ролі (рекрутер або кандидат).
2. Аутентифікація користувачів за допомогою електронної пошти та пароля.
3. Відновлення забутого пароля користувача.
4. Зміна пароля користувачем.
5. Обробка та зберігання авторизаційних даних користувачів.
6. Забезпечення безпеки авторизаційних даних та процесу аутентифікації.

3.2.2 Проектування модуля автентифікації

З урахуванням вимог, було розроблено архітектуру модуля автентифікації. Модуль містить компоненти для реєстрації користувачів, аутентифікації, відновлення та зміни пароля. Для безпечного зберігання паролів та перевірки їх відповідності було використано сучасні алгоритми хешування та солі. Також було розроблено механізм генерації та перевірки JSON Web Token (JWT), який використовується для забезпечення доступу до захищених ресурсів веб-застосунку.

3.2.3 Реалізація модуля автентифікації

На етапі реалізації модуля автентифікації було створено компоненти для реєстрації та аутентифікації користувачів, а також компоненти для відновлення та зміни пароля. Всі компоненти були реалізовані з використанням технологій Node.js та TypeScript, що забезпечує типізацію даних та зручність розробки.

Для реєстрації нових користувачів було створено компонент "CreateUser", який відповідає за перевірку введених даних та додавання нового користувача до бази даних. Після реєстрації користувача, пароль зберігається у захищеному вигляді, використовуючи алгоритми хешування та солі.

Аутентифікація користувачів відбувається за допомогою компонента "AuthUseCase". Цей компонент перевіряє наявність користувача в базі даних, валідацію введених даних та відповідність пароля. У разі успішної

аутентифікації, користувачу видається JSON Web Token, який дозволяє йому отримувати доступ до захищених ресурсів веб-застосунку.

Відновлення та зміна пароля користувача реалізовані через компоненти "ResetPasswordUseCase" та "ChangePasswordUseCase". При відновленні пароля користувачу надсилається лист на електронну пошту з посиланням для встановлення нового пароля. Компонент "ChangePasswordUseCase" дозволяє змінити пароль, перевіряючи старий пароль та забезпечуючи збереження нового пароля у захищеному вигляді.

Завершуючи розробку модуля автентифікації, було проведено тестування його функціональності. Тестування підтвердило правильність реалізації компонентів та їхню відповідність вимогам. Таким чином, модуль автентифікації успішно інтегровано у веб-застосунок для проведення співбесід.

Файли, які відповідають за аутентифікацію в цьому дереві файлів, знаходяться у наступних директоріях та файлах:

- **src/infrastructure/middleware/auth.ts:** Файл містить міжпрограмне забезпечення для аутентифікації користувачів.
- **src/module/auth/user:** Директорія містить компоненти, які стосуються аутентифікації користувачів, включаючи:
 - **config/factory.ts:** Фабрика для створення сервісів аутентифікації.
 - **core/model:** Моделі даних, що стосуються аутентифікації користувачів.
 - **core/port:** Інтерфейси для реалізації репозиторіїв та сервісів аутентифікації.
 - **core/useCase:** Відповідальний за реалізацію використання випадків аутентифікації, створення користувачів та перевірку токенів.
 - **core/validator:** Валідатори для перевірки даних, пов'язаних з аутентифікацією користувачів.
 - **repository:** Реалізація репозиторіїв для роботи з користувачами та аутентифікацією.

- **user.router.ts:** Маршрутизатор для обробки HTTP-запитів, пов'язаних з аутентифікацією користувачів.
- **utils:** Утиліти для роботи з хешуванням паролів, генерації та верифікації JWT токенів.
- **src/lib/utils/ValidatorUtils.ts:** Утиліти для валідації даних, які можуть бути використані при аутентифікації користувачів.

3.3 Розробка модуля проведення співбесід

Модуль проведення співбесід є ключовим компонентом веб-застосунку і має на меті полегшити процес планування, створення, проведення та аналізу результатів співбесід для кандидатів на різні вакансії. Цей модуль повинен забезпечити зручний та ефективний інструмент для роботи як з боку рекрутерів, так і з боку кандидатів.

3.3.1 Функціональність модуля проведення співбесід

Основні функції модуля проведення співбесід включають створення інтерв'ю-шаблонів з прив'язкою до певної компанії, додавання, редагування та видалення питань та категорій питань, а також додавання, редагування та видалення кроків та категорій кроків у шаблоні інтерв'ю. Крім того, модуль дозволяє запланувати співбесіди з кандидатами, вибираючи дату, час, тривалість та інші параметри, записати результати співбесіди в ході проведення онлайн-співбесіди з кандидатами з використанням відео- та аудіозв'язку, включаючи оцінки та коментарі рекрутера, та аналізувати результати співбесід та вибирати кандидатів для подальшого відбору або пропозиції роботи.

3.3.2 Реалізація компонентів модуля проведення співбесід

Для реалізації модуля проведення співбесід було розроблено ряд компонентів, які відповідають за створення та редагування інтерв'ю-шаблонів та їх елементів, роботу з питаннями та категоріями питань, роботу з кроками та категоріями кроків у шаблоні інтерв'ю. Також були розроблені компоненти для

планування співбесід з кандидатами, включаючи календар та форму введення деталей співбесіди, а також компоненти для проведення онлайн-співбесід, які забезпечують відео- та аудіозв'язок, інструменти для нотаток та оцінок. Крім того, були розроблені компоненти для збереження результатів співбесід та їх подальшого аналізу.

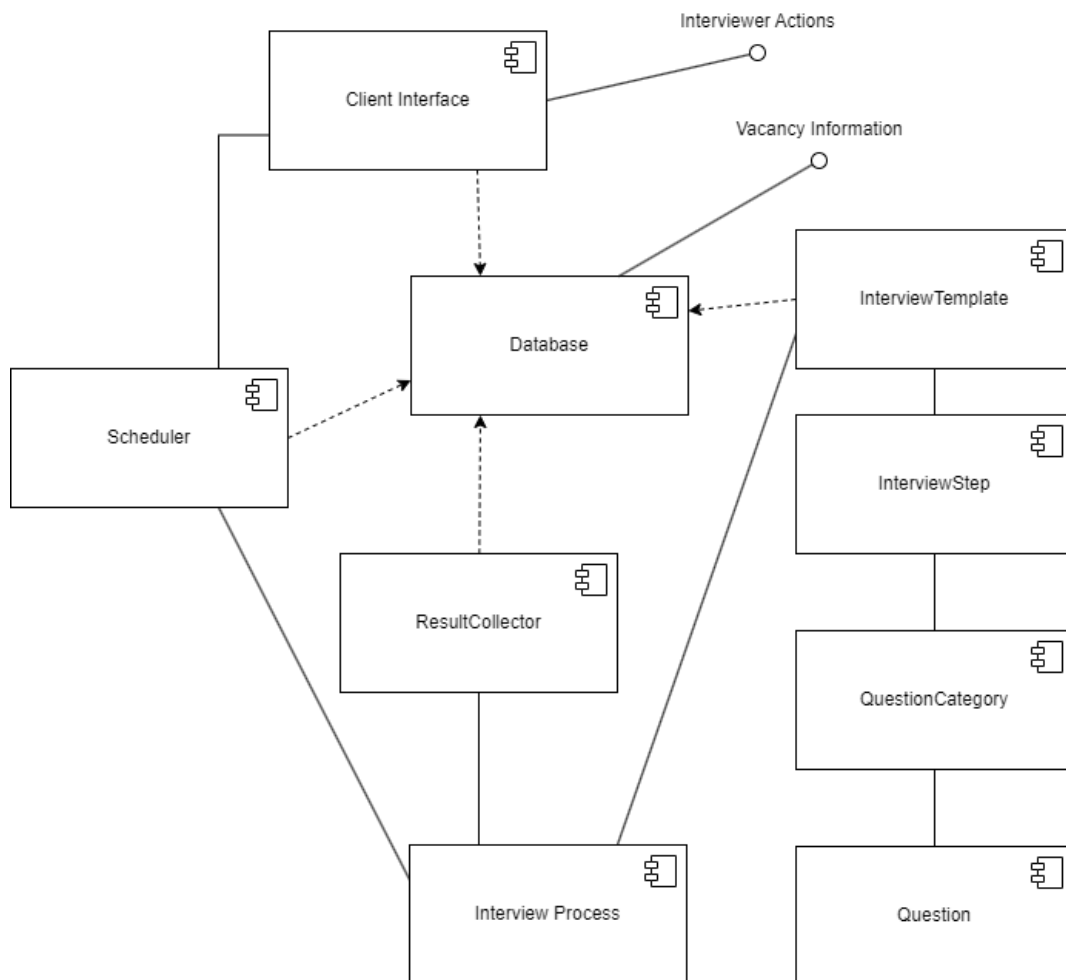


Рисунок 7 – Діаграма компонентів модуля проведення співбесід

3.3.3 Тестування модуля проведення співбесід

Після реалізації компонентів модуля проведення співбесід було проведено комплексне тестування з метою перевірки правильності роботи модуля та виявлення та виправлення можливих помилок та недоліків. Тестування включає перевірку функціональності, зокрема роботи з інтерв'ю-шаблонами, питаннями

та категоріями питань, кроками та категоріями кроків, планування та проведення співбесід, запису результатів співбесід та аналізу їх результатів.

Крім того, було проведено тестування продуктивності, стабільності та масштабованості модуля, що допомогло виявити та виправити слабкі місця та оптимізувати роботу системи під час виконання різних завдань та з великою кількістю користувачів.

В результаті тестування модуль проведення співбесід було визнано працездатним, а всі виявлені проблеми були успішно вирішені. Завдяки успішному тестуванню та відпрацюванню всіх аспектів роботи модуля, веб-застосунок готовий до впровадження на ринку та використання рекрутерами та кандидатами для проведення співбесід.

3.4 Оцінювання кандидатів та формування результатів проведення співбесіди

У системі рекрутингу та проведення співбесід, яка розробляється в рамках даної наукової роботи, передбачається наявність спеціальної форми для збереження коментарів та суб'єктивної оцінки відповідей кандидатів на конкретні питання. Це дозволить рекрутерам систематизувати та зберегти свої враження та оцінки під час співбесіди, а також забезпечить однорідність та прозорість оцінювання кандидатів.

Для проведення співбесіди рекрутер використовує спеціальний модуль, який доступний в інформаційній системі. Починаючи процес співбесіди, рекрутер має можливість обрати кандидата, з яким він бажає провести співбесіду. Система містить в собі інформацію про всіх кандидатів, що дозволяє рекрутеру знайти потрібного кандидата і перейти до наступного етапу.

Для зручності комунікації з кандидатом, система підтримує інтеграцію зі сторонніми ресурсами, що надають можливість встановити аудіо-відеозв'язок. Це дозволяє рекрутеру спілкуватися з кандидатом в режимі реального часу, незалежно від їх фізичного місцезнаходження. Крім того, інформаційна система

може також забезпечувати відправку запрошень на співбесіду через електронну пошту або повідомлення внутрішньої системи комунікації.

Під час співбесіди, рекрутер задає кандидату питання та ретельно аналізує його відповіді. В системі передбачено наявність спеціальної форми для кожного питання, де рекрутер може записати свої коментарі та суб'єктивну оцінку щодо відповіді кандидата. Крім того, система надає можливість налаштування полів оцінювання відповідей, таких як числова шкала або використання інших метрик (Numeric Rating Scale, Visual Analog Scale, Verbal Rating Scale, Faces Scale), які відповідають потребам рекрутера.

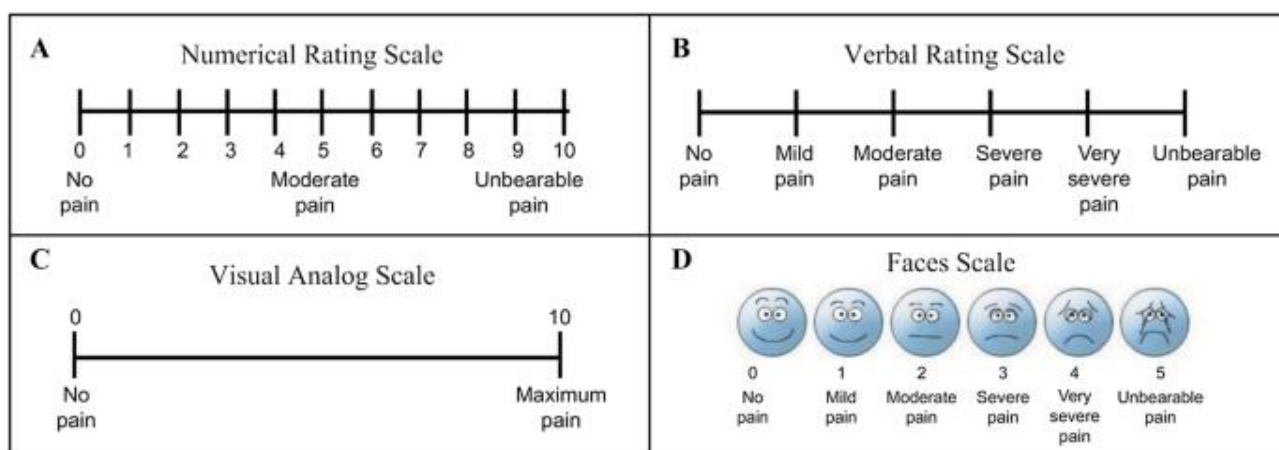


Рис. 8 – Приклад шкал оцінювання

Після завершення співбесіди, всі записи, що містять інформацію про учасників співбесіди, категорії питань, самі питання з коментарями та оцінками рекрутера, зберігаються в реляційній базі даних. Це дозволяє зберігати всю історію співбесід та здійснювати подальший аналіз даних.

Після оцінювання кожної категорії питань та загального враження від співбесіди, рекрутер формує загальний висновок щодо кандидата. У системі передбачено можливість формування PDF-документу, який містить оцінки, коментарі та загальний висновок рекрутера. Цей документ надсилається кандидату, щоб він мав можливість ознайомитися зі своїм результатом і, в разі неуспішного проходження співбесіди, використати його для аналізу та покращення своїх навичок.

3.5 Розгортання

Основною метою розгортання є запуск додатку і його підготовка до роботи у продакшн середовищі. Цей процес включає в себе кілька кроків, щоб забезпечити безперебійну роботу додатку та його доступність для користувачів.

Першим кроком розгортання є підготовка серверного середовища. Для цього можна використовувати хмарні сервіси, такі як Amazon Web Services (AWS) або Microsoft Azure, або власні фізичні сервери. Середовище повинно мати встановлені необхідні залежності, такі як Node.js та PostgreSQL, а також налаштований веб-сервер, який буде обробляти запити від клієнтів.

Другим кроком є підготовка бази даних. Додаток використовує PostgreSQL, тому необхідно створити базу даних і налаштувати права доступу до неї. Для цього можна використовувати інтерфейс командної строки або графічний інструмент, такий як pgAdmin.

Після підготовки середовища і бази даних можна переходити до розгортання серверної частини додатку. Код сервера, написаний з використанням Express, повинен бути завантажений на сервер та налаштований. Це включає встановлення необхідних залежностей, налаштування конфігураційних параметрів, таких як порт, на якому буде працювати сервер, а також налаштування маршрутів та логіки обробки запитів.

Наступним кроком є розгортання клієнтської частини додатку, реалізованої з використанням React.js. Код клієнта потрібно збілдити (build) для отримання оптимізованих і статичних файлів, які можна використовувати на сервері. Ці файли можна розмістити на веб-сервері або конфігурувати середовище, таке як Nginx або Apache, для їх обслуговування.

Останнім кроком є запуск додатку. Після успішного розгортання серверної та клієнтської частин додатку і налаштування веб-сервера, можна запустити сервер та забезпечити доступ до додатку через веб-браузер. Після запуску додаток повинен бути готовий до використання користувачами.

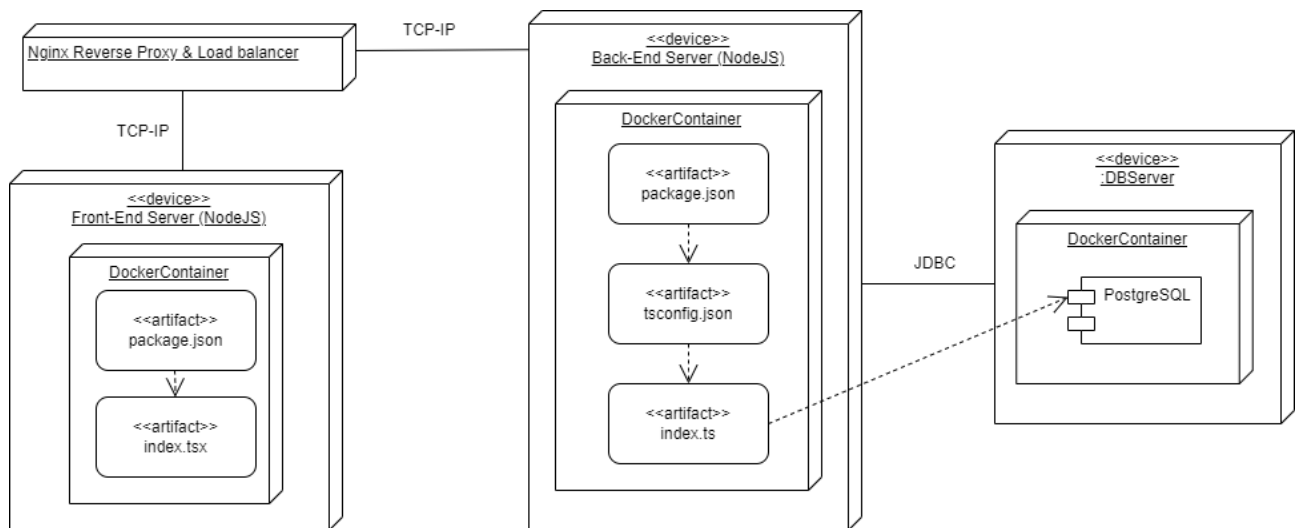


Рисунок 9 – Діаграма розгортання (deployment diagram)

Зручним та ефективним рішенням для розгортання інформаційної системи є використання Docker, що забезпечує зручність у керуванні та переносимість середовища додатку.

Для розгортання створено окремий Docker образ для кожного компонента системи, а саме: для бази даних PostgreSQL; для серверної частини, де встановлено міжплатформне середовище виконання Node.js та веб-фреймворк Express; для клієнтської частини де встановлено залежності для бібліотеки React.js.

Після створення Docker образів їх було акомпановано у docker-compose файлі, де налаштовуються конфігурації для кожного контейнера, зокрема мережі та залежності. Запускаємо docker-compose, щоб одночасно запустити всі компоненти додатку.

Контейнеризація спрощує розгортання та забезпечує переносимість додатку. Контейнери забезпечують ізольоване середовище для кожного компонента, що дозволяє уникати конфліктів залежностей. Docker також забезпечує портативність, оскільки контейнери можна легко передавати між середовищами.

3.6 Опис програмного продукту

Інформаційна система для рекрутингу та проведення співбесід є комплексним рішенням, розробленим для оптимізації процесів відбору персоналу та покращення якості співбесід. Цей програмний продукт надає зручний та ефективний інструментарій для рекрутерів та HR-фахівців, щоб допомогти їм у проведенні співбесід та оцінці кандидатів.

Основні особливості та функціонал продукту:

1. Керування кандидатами: система забезпечує централізоване керування базою даних кандидатів, де можна зберігати інформацію про їх резюме, контактні дані, відгуки та інші важливі деталі. Кандидати можуть бути організовані в категорії та підкатегорії для зручного пошуку та фільтрації.
2. Планування та організація співбесід: рекрутери можуть створювати розклади співбесід, назначати дати та час, запрошувати кандидатів та надавати їм доступ до віртуального середовища для проведення співбесіди. Система надає зручний інтерфейс для планування та управління всіма аспектами процесу співбесіди.
3. Віртуальні співбесіди: завдяки інтеграції з аудіо-відео засобами зв'язку, система дозволяє проводити віртуальні співбесіди в режимі реального часу. Рекрутери можуть задавати питання кандидатам, спостерігати за їх відповідями та записувати коментарі та оцінки.
4. Збереження коментарів та оцінок: система надає спеціальну форму для рекрутерів, де вони можуть записувати свої коментарі та суб'єктивну оцінку на відповіді кандидатів на кожне питання. Це дозволяє систематизувати та зберегти враження рекрутерів під час співбесіди.
5. Аналіз та порівняння кандидатів: зібрані дані про кандидатів, коментарі та оцінки зберігаються в реляційній базі даних. Це дозволяє рекрутерам аналізувати результати співбесід, порівнювати кандидатів та приймати обґрунтовані рішення щодо подальшого відбору.
6. Генерація звітів: після завершення співбесіди, система надає можливість інтерв'юерам оцінити кожну категорію питань та сформувати загальний

висновок щодо кандидата. За допомогою функції генерації звітів, система формує PDF-документ, який може бути надісланий кандидату. Цей звіт містить оцінки, коментарі та рекомендації, що допоможуть кандидату покращити свої навички в разі неуспішного проходження співбесіди.

7. Захист даних: Система забезпечує високий рівень безпеки та захисту конфіденційної інформації. Доступ до даних обмежений за допомогою різних рівнів автентифікації та прав доступу.

Загальні переваги програмного продукту:

- скорочення часу та зусиль, необхідних для проведення співбесід та оцінки кандидатів;
- покращення точності та об'єктивності оцінювання кандидатів;
- збереження та організація інформації про кандидатів для подальшого аналізу та порівняння;
- забезпечення ефективного спілкування з кандидатами незалежно від їх місцезнаходження;
- підвищення ефективності та професіоналізму рекрутерів та HR-фахівців.

Також потрібно врахувати недоліки для об'єктивності, які можуть виникнути в ході подальшої розробки та використання системи:

- складність впровадження: встановлення та налаштування програмного продукту може бути складним процесом, особливо для компаній з обмеженими технічними ресурсами. Вимагається фахівець з встановлення та підтримки системи;
- навчання користувачів: користувачам, зокрема рекрутерам і HR-фахівцям, може знадобитися час та навчання для повного розуміння та використання всіх функцій та можливостей програмного продукту;
- потенційні помилки в оцінюванні: оцінки та коментарі, які рекрутери вводять в систему, є суб'єктивними і можуть містити помилки або перекозчення. Це може вплинути на об'єктивність процесу відбору та оцінки кандидатів;

- вимоги до обладнання: для використання програмного продукту можуть вимагатися специфічні обладнання, такі як веб-камера, мікрофон або інші пристрої, для забезпечення аудіо- та відеозв'язку під час співбесіди. Це може становити проблему для деяких користувачів.

Загалом, інформаційна система для рекрутингу та проведення співбесід надає комплексне рішення для управління процесом підбору персоналу, що сприяє покращенню ефективності та якості відбору кандидатів.

Як результат роботи інформаційної системи для рекрутингу та проведення співбесід надаються знімки екранів та частина файлового дерева – нижче у розділі додатків.

Висновки розділу 3

У розділі 3 було описано реалізацію кросплатформенного веб-застосунку для проведення співбесід. Було розглянуто ключові етапи розробки та інтеграції системи, а також важливі функціональні можливості, які надаються користувачам.

Застосунок розроблений з урахуванням потреб рекрутерів та HR-фахівців у зручному та ефективному інструменті для проведення співбесід. Він забезпечує можливість встановлення аудіо- відеозв'язку з кандидатом, а також зберігання коментарів та суб'єктивних оцінок рекрутера на кожну відповідь.

Одним із основних переваг є його кросплатформенність, що дозволяє користувачам отримати доступ до системи з будь-якого пристрою, що має підключення до Інтернету. Це спрощує процес співбесіди та забезпечує більшу гнучкість та зручність для всіх учасників.

Програмний продукт також має свої недоліки, які зокрема включають складність впровадження та навчання користувачів. Однак, ці недоліки можуть бути враховані та опрацьовані за допомогою відповідних стратегій та підтримки з боку розробників.

ВИСНОВКИ

В рамках даної наукової роботи було поставлено мету оптимізувати роботу менеджерів з підбору персоналу, покращити якість відбору персоналу на етапі проведення співбесід та впровадити прозорі оцінки результатів співбесід. Для досягнення цих цілей, було сформульовано дослідницькі завдання, які включали огляд та аналіз сучасних типів та методів проведення співбесід, аналіз наявних методів та категорій суб'єктивного оцінювання кандидатів, ознайомлення з існуючими бібліотеками та інструментами для автоматизації рекрутингу та дослідження способів структурування програмної системи.

Дослідження було здійснене за допомогою різноманітних джерел, таких як електронні ресурси, друкована література, матеріали конференцій та вихідні коди програм та бібліотек. Цей підхід дозволив зібрати відповідну та актуальну інформацію для виконання детального аналізу та досягнення поставлених цілей дослідження.

У результаті проведених досліджень була розроблена програмна система для автоматизації процесу рекрутингу з проведенням співбесід, яка надає зручний інструмент для менеджерів з підбору персоналу. Ця система дозволяє проводити співбесіди з використанням аудіо- та відеозв'язку, зберігати коментарі та суб'єктивні оцінки рекрутерів, а також забезпечує прозорість оцінювання кандидатів. Крім того, система зберігає записи співбесід в реляційній базі даних та надає звіти, які допомагають в прийнятті рішень щодо відбору персоналу.

Розроблена програмна система забезпечує автоматизацію процесу відбору персоналу та покращення підходів до оцінювання кандидатів. Вона сприяє підвищенню якості оцінювання знань та навичок кандидатів, а також оптимізує час менеджерів з підбору персоналу, що веде до ефективнішого та економічно вигідного процесу рекрутингу.

Розроблені підходи та методи можуть бути успішно використані в реальних умовах роботи організацій, що сприятиме їхньому розвитку та підвищенню продуктивності. Ця наукова робота вносить значний внесок у

поліпшення процесів рекрутингу та сприяє вибору найкращих кандидатів для вакансій. Розроблена програмна система є потужним інструментом для підтримки рекрутерів та покращення якості відбору персоналу.

Отже, це дослідження є важливим кроком у розвитку рекрутингу та сприяє покращенню процесів відбору персоналу. Впровадження автоматизованої системи рекрутингу з проведенням співбесід покращить ефективність та точність відбору кандидатів, забезпечить зменшення суб'єктивності в оцінюванні та сприятиме швидшому прийняттю рішень. Це значно зекономить час та ресурси організації, а також підвищить задоволеність як менеджерів з відбору персоналу, так і кандидатів на посади.

З урахуванням всіх отриманих результатів, можна стверджувати, що дослідження має великий потенціал для подальшого розвитку та вдосконалення. Запровадження запропонованих рішень та рекомендацій допоможе підвищити ефективність рекрутингу та забезпечить організації вибір найкращих кандидатів для досягнення її стратегічних цілей.

Крім того, важливим аспектом розробки програмної системи для автоматизації рекрутингу є використання чистої архітектури. Чиста архітектура є підходом, який дозволяє створювати гнучкі, розширювані та легко тестувані системи. Вона розбиває програмний продукт на незалежні компоненти, що дозволяє змінювати або модифікувати одну частину системи без впливу на інші.

Використання чистої архітектури при розробці системи рекрутингу дозволяє забезпечити легкість супроводу, масштабованість та стабільність програмного продукту. Чиста архітектура дозволяє використовувати принципи SOLID (Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation, Dependency Inversion), що сприяють розкладенню системи на компоненти з чіткою відповідальністю та зменшують залежності між ними. Це робить систему більш гнучкою та легко змінюваною.

Таким чином, використання чистої архітектури при розробці програмної системи для автоматизації рекрутингу є важливим кроком у забезпеченні якості, розширюваності та ефективності системи. Цей підхід дозволяє створити

систему, яка легко супроводжується та розвивається, а також забезпечує гнучкість та швидкість внесення змін у вимоги до системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. "The Validity and Utility of Selection Methods in Personnel Psychology: Practical and Theoretical Implications of 85 Years of Research Findings" (2004)
2. Macan, Therese (September 2009). "The employment interview: A review of current studies and directions for future research". *Human Resource Management Review*. 19 (3): 203–218.
3. Chapman, Derek Scott (2000). Modeling job applicant decision processes, integrating applicant reactions to selection procedures into the critical contact framework of recruiting (Thesis).
4. Tsai, Wei-Chi; HsinHung Chen, Forrence; Chen, Hao-Yi; Tseng, Ko-Yao (March 2016). "When Will Interviewers Be Willing to Use High-structured Job Interviews? The role of personality: High-structured Interviews". *International Journal of Selection and Assessment*. 24 (1): 92–105.
5. Dipboye, R. (1997). "Structured selection interviews: Why do they work? Why are they underutilized?". In Anderson, Neil; Herriot, Peter (eds.). *Assessment and Selection in Organizations, International Handbook of Selection and Assessment*.
6. Kausel, Edgar E.; Culbertson, Satoris S.; Madrid, Hector P. (November 2016). "Overconfidence in personnel selection: When and why unstructured interview information can hurt hiring decisions". *Organizational Behavior and Human Decision Processes*. 137: 27–44.
7. Robie, Chet; Tuzinski, Kathleen A.; Bly, Paul R. (October 2006). "A survey of assessor beliefs and practices related to faking". *Journal of Managerial Psychology*. 21 (7): 669–681.
8. Highhouse, Scott (September 2008). "Stubborn Reliance on Intuition and Subjectivity in Employee Selection". *Industrial and Organizational Psychology*. 1 (3): 333–342.
9. Levashina, Julia; Hartwell, Christopher J.; Morgeson, Frederick P.; Campion, Michael A. (March 2014). "The Structured Employment Interview: Narrative and

- Quantitative Review of the Research Literature". *Personnel Psychology*. 67 (1): 241–293.
10. Campion, Michael A.; Palmer, David K.; Campion, James E. (September 1997). "A review of structure in the selection interview". *Personnel Psychology*. 50 (3): 655–702.
 11. Huffcutt, Allen I.; Conway, James M.; Roth, Philip L.; Stone, Nancy J. (2001). "Identification and meta-analytic assessment of psychological constructs measured in employment interviews". *Journal of Applied Psychology*. 86 (5): 897–913.
 12. Salgado, Jesus F.; Moscoso, Silvia (September 2002). "Comprehensive meta-analysis of the construct validity of the employment interview". *European Journal of Work and Organizational Psychology*. 11 (3): 299–324.
 13. Morgeson, Frederick P.; Reider, Matthew H.; Campion, Michael A. (September 2005). "Selecting Individuals in Team Settings: The Importance of Social Skills, Personality Characteristics, and Teamwork Knowledge". *Personnel Psychology*. 58 (3): 583–611.
 14. Campbell, J. P., McCloy, R. A., Oppler, S. H., & Sager, C. E. (1993). "A theory of performance." In N. Schmitt & W. C. Borman (Eds.), *Personnel selection in organizations* (pp. 35–70). San Francisco: Jossey-Bass.
 15. Schlenker, Barry R. (1980). *Impression Management: The Self-concept, Social Identity, and Interpersonal Relations*. Brooks/Cole Publishing Company.
 16. Kacmar, K. Michele; Delery, John E.; Ferris, Gerald R. (August 1992). "Differential Effectiveness of Applicant Impression Management Tactics on Employment Interview Decisions". *Journal of Applied Social Psychology*. 22 (16): 1250–1272.
 17. Ferris, Gerald R.; Witt, L. A.; Hochwarter, Wayne A. (2001). "Interaction of social skill and general mental ability on job performance and salary". *Journal of Applied Psychology*. 86 (6): 1075–1082.
 18. Snyder, Mark (October 1974). "Self-monitoring of expressive behavior". *Journal of Personality and Social Psychology*. 30 (4): 526–537.

19. Tullar, William L. (1989). "Relational control in the employment interview". *Journal of Applied Psychology*. 74 (6): 971–977.
20. "The Sound of Employability: Interviewers Judge Your Voice". Association For Psychological Science. Retrieved 17 March 2021.
21. DeGroot, Timothy; Motowidlo, Stephan J. (1999). "Why visual and vocal interview cues can affect interviewers' judgments and predict job performance". *Journal of Applied Psychology*. 84 (6): 986–993.
22. Burnett, Jennifer R.; Motowidlo, Stephan J. (December 1998). "Relations Between Different Sources of Information in the Structured Selection Interview". *Personnel Psychology*. 51 (4): 963–983.
23. McCarthy, Julie M.; Van Iddekinge, Chad H.; Campion, Michael A. (June 2010). "Are Highly Structured Job Interviews Resistant to Demographic Similarity Effects?". *Personnel Psychology*. 63 (2): 325–359.
24. McFarland, Lynn A.; Ryan, Ann Marie; Sacco, Joshua M.; Kriska, S. David (August 2004). "Examination of Structured Interview Ratings Across Time: The Effects of Applicant Race, Rater Race, and Panel Composition". *Journal of Management*. 30 (4): 435–452.
25. Wade, Kim J.; Kinicki, Angelo J. (February 1997). "Subjective Applicant Qualifications and Interpersonal Attraction as Mediators within a Process Model of Interview Selection Decisions". *Journal of Vocational Behavior*. 50 (1): 23–40.
26. Segrest Purkiss, Sharon L.; Perrewé, Pamela L.; Gillespie, Treena L.; Mayes, Bronston T.; Ferris, Gerald R. (November 2006). "Implicit sources of bias in employment interview judgments and decisions". *Organizational Behavior and Human Decision Processes*. 101 (2): 152–167.
27. Dipboye, R. L., & Macan, T. (1988). A process view of the selection-recruitment interview. In R. Schuler, V. Huber, & S. Youngblood (Eds.), *Readings in personnel and human resource management* (pp. 217–232). New York: West.
28. Macan, Therese H; Dipboye, Robert L (December 1988). "The effects of interviewers' initial impressions on information gathering". *Organizational Behavior and Human Decision Processes*. 42 (3): 364–387.

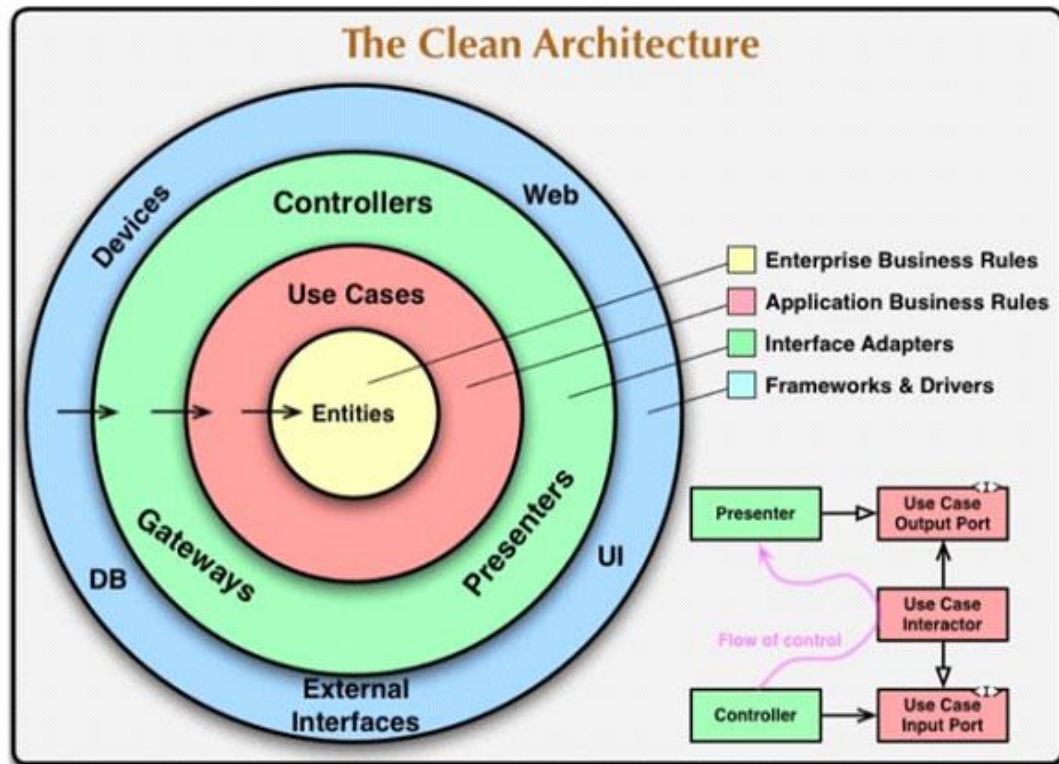
29. Macan, Therese Hoff; Dipboye, Robert L. (December 1990). "The relationship of prior experience interviewing to interview outcomes". *Journal of Applied Psychology*. 75 (5): 514–519.
30. Dipboye, R. L., & Macan, T. (1994). The effects of interview structure on recall and judgment processes. *Organizational Behavior and Human Decision Processes*, 58(2), 253–285.
31. Straus, Susan G.; Miles, Jeffrey A.; Levesque, Laurie L. (червень 2001). "The effects of videoconference, telephone, and face-to-face media on interviewer and applicant judgments in employment interviews". *Journal of Management*. 27 (3): 363–381.
32. Word, Carl O; Zanna, Mark P; Cooper, Joel (березень 1974). "The nonverbal mediation of self-fulfilling prophecies in interracial interaction". *Journal of Experimental Social Psychology*. 10 (2): 109–120.
33. Latham, Gary P.; Saari, Lise M.; Pursell, Elliott D.; Campion, Michael A. (1980). "The situational interview". *Journal of Applied Psychology*. 65 (4): 422–427.
34. Janz, Tom (1982). "Initial comparisons of patterned behavior description interviews versus unstructured interviews". *Journal of Applied Psychology*. 67 (5): 577–580.
35. Weekley, Jeff A.; Gier, Joseph A. (1987). "Reliability and validity of the situational interview for a sales position". *Journal of Applied Psychology*. 72 (3): 484–487.
36. "The Pros and Cons of Applicant Tracking Systems." Retrieved from: <https://www.4cornerresources.com/blog/the-pros-and-cons-of-applicant-tracking-systems/>
37. "Як ефективно використовувати LinkedIn." Retrieved from: <https://happymonday.ua/kak-jeffektivno-ispolzovat-linkedin>
38. Hurma. Retrieved from: <https://hurma.work/>
39. CleverStaff. Retrieved from: <https://cleverstaff.net/ua>
40. HireVue. Retrieved from: <https://www.hirevue.com/>
41. Spark Hire. Retrieved from: <https://www.sparkhire.com/>

42. Pymetrics. Retrieved from: <https://www.pymetrics.ai/>
43. Plum. Retrieved from: <https://www.plum.io/>
44. Sapia. Retrieved from: <https://sapia.ai/>
45. Мартін, Роберт. "Чиста архітектура: посібник ремісника зі структури та проектування програмного забезпечення." С. 156. Prentice Hall, 2017. (Robert C. Martin. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall, 2017.)
46. Freeman, Steve, and Nat Pryce. "Growing Object-Oriented Software, Guided by Tests."
47. Jacobson, Ivar, et al. "Object-Oriented Software Engineering: A Use Case Driven Approach."
48. PostgreSQL. Retrieved from: <https://www.postgresql.org/>
49. Мартін, Роберт С. "Чиста архітектура: посібник ремісника зі структури та проектування програмного забезпечення." Prentice Hall, 2017. (Robert C. Martin. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall, 2017.)
50. Martin, Uncle Bob. "The Clean Architecture." Retrieved from: <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>
51. Bogard, Jimmy. "Clean Architecture with ASP.NET Core 3.0." Retrieved from: <https://www.youtube.com/watch?v=dK4Yb6-LxAk>
52. Naveen Kumar Singh. Understanding Clean Architecture. <https://medium.com/@naveenksingh/understanding-clean-architecture-5ac64c0c6bea>
53. Singh, Naveen Kumar. "Understanding Clean Architecture." Retrieved from: <https://medium.com/@naveenksingh/understanding-clean-architecture-5ac64c0c6bea>
54. Khorikov, Volodymyr. "Applying the Clean Architecture to Go applications." Retrieved from: <https://www.youtube.com/watch?v=6OSePZGmqmA>
55. Seemann, Mark. "Architecture Without an End State." Retrieved from: <https://blog.ploeh.dk/2013/04/15/architecture-without-an-end-state/>
56. Newman, Sam. "Building Microservices: Designing Fine-Grained Systems." O'Reilly Media, 2015.

57. Martin Fowler. "Microservices." Retrieved from: <https://martinfowler.com/articles/microservices.html>
58. Eric Evans. "Domain-Driven Design: Tackling Complexity in the Heart of Software." Addison-Wesley Professional, 2003.
59. Eric Evans. "Domain-Driven Design: Tackling Complexity in the Heart of Software." Addison-Wesley Professional, 2003.
60. Vaughn Vernon. "Implementing Domain-Driven Design." Addison-Wesley Professional, 2013.
61. Jimmy Bogard. "Anatomy of a Highly-Effective Domain Model." Retrieved from: <https://jimmybogard.com/anatomy-of-a-highly-effective-domain-model/>
62. Scott Wlaschin. "Domain-Driven Design in F#." Retrieved from: <https://fsharpforfunandprofit.com/posts/ddd/>
63. Martin Fowler. "Anemic Domain Model." Retrieved from: <https://martinfowler.com/bliki/AnemicDomainModel.html>
64. Greg Young. "The Art of Creating Domain-Specific Languages." Retrieved from: <https://www.infoq.com/presentations/DSL-Art/>
65. Julie Lerman. "How to Apply Domain-Driven Design Patterns in EF Core 3.0." Retrieved from: <https://www.youtube.com/watch?v=BvfhB5KzU6E>
66. Pat Helland. "Data on the Outside versus Data on the Inside." Retrieved from: <https://queue.acm.org/detail.cfm?id=1056959>
67. Martin Fowler. "Domain-Specific Languages." Retrieved from: <https://martinfowler.com/tags/dsl.html>
68. Udi Dahan. "Domain Events - Salvation." Retrieved from: <https://udidahan.com/2008/02/04/domain-events-salvation/>

ДОДАТКИ

Додаток А. Діаграма Чиста Архітектура



Додаток Б. Структура файлів проєкту

```
- alladin-be-node-main/  
  - .env  
  - .eslinttrc.js  
  - .github/  
    - workflows/  
      - main.yml  
  - .gitignore  
  - config.ts  
  - docker-compose-postgre.yaml  
  - Dockerfile  
  - package.json  
  - README.md  
  - secret.ts  
  - src/  
    - db/  
      - postgre/
```

- data-source.ts
- entities/
 - BaseEntity.ts
 - CompanyEntity.ts
 - InterviewTemplateEntity.ts
 - QuestionCategoryEntity.ts
 - QuestionEntity.ts
 - StepCategory.ts
 - TemplateStepEntity.ts
 - UserEntity.ts
- index.ts
- repositories/
 - companyRepository.ts
 - interviewTemplateRepository.ts
 - questionCategoryRepository.ts
 - questionRepository.ts
 - stepCategoryRepository.ts
 - templateStepRepository.ts
 - userRepository.ts
- index.ts
- infrastructure/
 - middleware/
 - auth.ts
 - errorHandler.ts
 - utils/
 - tryExecute.ts
- lib/
 - model/
 - app-exception/
 - BaseException.ts
 - NotFoundException.ts
 - UnauthorizedException.ts
 - UniqueException.ts
 - ValidationException.ts
 - CommandUseCase.ts
 - ResponseError.ts
 - UserRole.ts
 - ValidateCommandUseCase.ts
 - Validator.ts

```
- utils/  
  - ValidatorUtils.ts  
- module/  
  - auth/  
    - company/  
      - company.router.ts  
      - config/  
        - factory.ts  
      - core/  
        - model/  
          - Company.ts  
          - CreateCompany.ts  
        - port/  
          - CompanyRepository.ts  
        - useCase/  
          - CreateCompanyUseCase.ts  
        - validator/  
          - CreateCompanyValidator.ts  
      - repository/  
        - PostgreCreateCompanyRepository.ts  
  - user/  
    - config/  
      - factory.ts  
    - core/  
      - model/  
        - AuthUser.ts  
        - CreateUser.ts  
        - ReturnUser.ts  
        - Token.ts  
        - TokenUser.ts  
        - User.ts  
      - port/  
        - CreateUserRepository.ts  
        - PasswordHashing.ts  
        - PasswordValidator.ts  
        - TokenGenerator.ts  
        - TokenVerifier.ts  
        - UserRepository.ts  
      - useCase/
```

- AuthUseCase.ts
- CreateUserUseCase.ts
- GetUserUseCase.ts
- VerifyTokenUseCase.ts
- validator/
 - AuthUserValidator.ts
 - CreateUserValidator.ts
- repository/
 - PostgreCreateUserRepository.ts
 - PostgreUserRepository.ts
- user.router.ts
- utils/
 - BcryptPasswordUtils.ts
 - JWTGenerator.ts
 - TokenVerifier.ts
- interview-template/
 - question/
 - config/
 - factory.ts
 - core/
 - model/
 - CreateQuestion.ts
 - Question.ts
 - UpdateQuestion.ts
 - port/
 - QuestionRepository.ts
 - useCase/
 - CreateQuestionUseCase.ts
 - DeleteQuestionUseCase.ts
 - PatchPositionQuestionUseCase.ts
 - UpdateQuestionUseCase.ts
 - validator/
 - CreateQuestionValidator.ts
 - UpdateQuestionValidator.ts
- question-router.ts
- repository/
 - PostgresQuestionRepository.ts
- question-category/
 - config/

- factory.ts
- core/
 - model/
 - QuestionCategory.ts
 - port/
 - QuestionCategoryRepository.ts
 - useCase/
 - GetAllQuestionCategoriesUseCase.ts
- question-category.router.ts
- repository/
 - PostgresQuestionCategoryRepository.ts
- step/
 - config/
 - factory.ts
 - core/
 - model/
 - CreateStep.ts
 - PatchPosition.ts
 - Step.ts
 - UpdateStep.ts
 - port/
 - StepRepository.ts
 - useCase/
 - CreateStepUseCase.ts
 - DeleteStepUseCase.ts
 - GetStepByInterviewTemplateUseCase.ts
 - PatchPositionStepUseCase.ts
 - UpdateStepUseCase.ts
 - validator/
 - CreateStepValidator.ts
 - UpdateStepValidator.ts
 - repository/
 - PostgresStepRepository.ts
 - step.router.ts
- step-category/
 - config/
 - factory.ts
 - core/
 - model/

- CreateStepCategory.ts
- StepCategory.ts
- StepCategoryQuestion.ts
- port/
 - StepCategoryRepository.ts
- useCase/
 - CreateStepCategoryUseCase.ts
 - DeleteStepCategoryUseCase.ts
 - GetStepCategoryQuestionUseCase.ts
 - PatchPositionStepCategoryUseCase.ts
- validator/
 - CreateStepCategoryValidator.ts
- repository/
 - mapper/
 - mapStepCategoryQuestion.ts
 - PostgresStepCategoryRepository.ts
- step-category.router.ts
- template/
 - config/
 - factory.ts
 - core/
 - model/
 - CreateInterviewTemplate.ts
 - InterviewTemplate.ts
 - UpdateInterviewTemplate.ts
 - port/
 - InterviewTemplateRepository.ts
 - useCase/
 - CreateInterviewTemplateUseCase.ts
 - DeleteInterviewTemplateUseCase.ts
 - GetInterviewTemplatesByCompanyUseCase.ts
 - UpdateInterviewTemplateUseCase.ts
 - validator/
 - CreateInterviewTemplateValidator.ts
 - UpdateInterviewTemplateValidator.ts
- interview-template.router.ts
- repository/
 - PostgresInterviewTemplateRepository.ts
- swagger-output.json

```
- swagger.js
- test/
  - module/
    - auth/
      - company/
        - core/
          - useCase/
            - CreateCompanyUseCase.test.ts
- tsconfig.json
- types/
  - express/
    - index.d.ts
```

Додаток В. Сторінка реєстрації компанії

Create company

Company name

Admin name

Admin email

Admin company role

Password

Repeat password

>

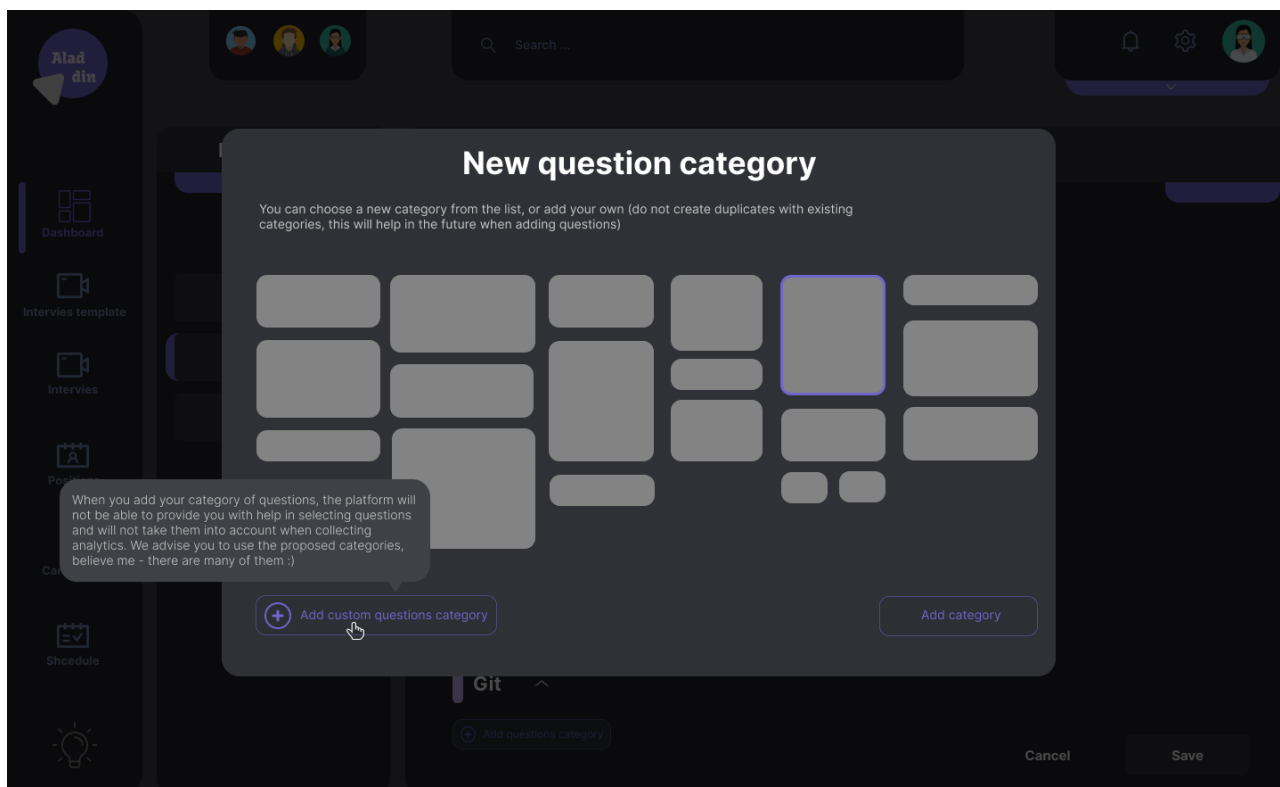
Aladin

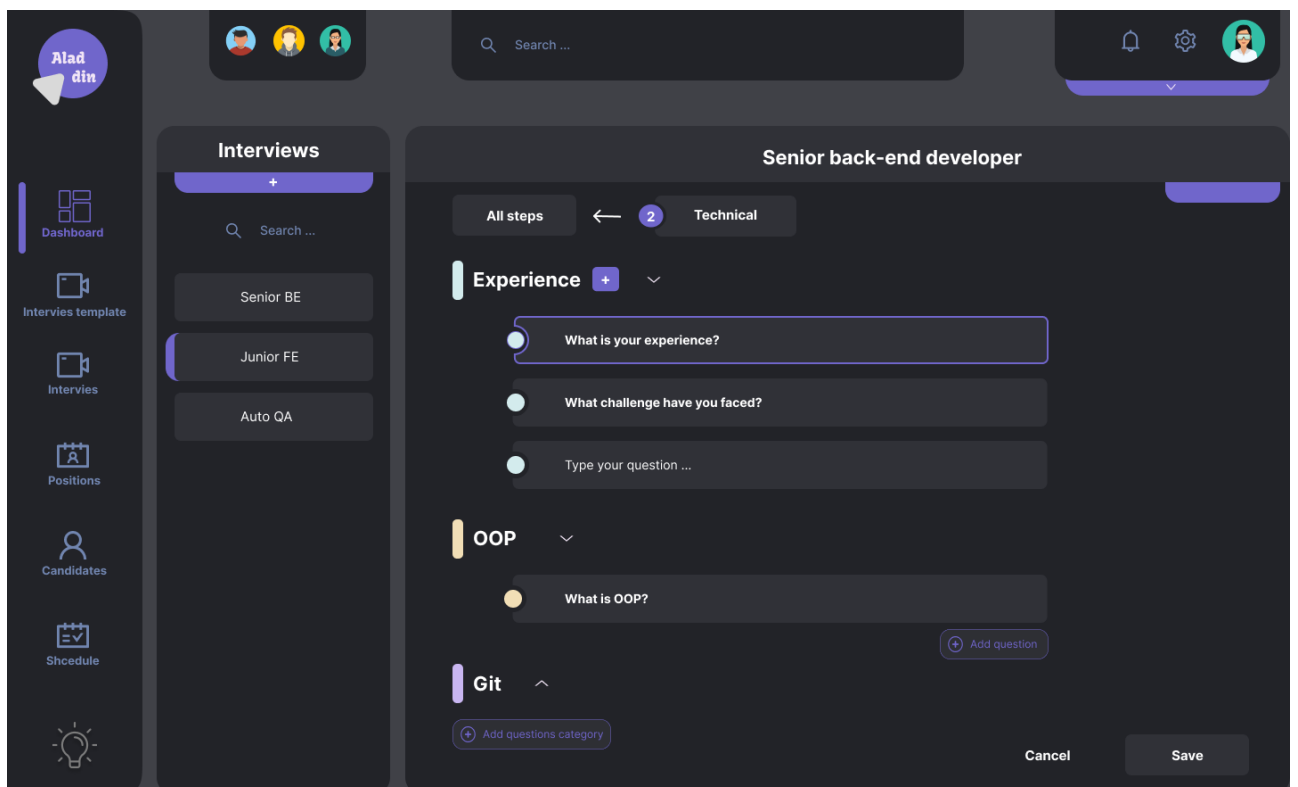
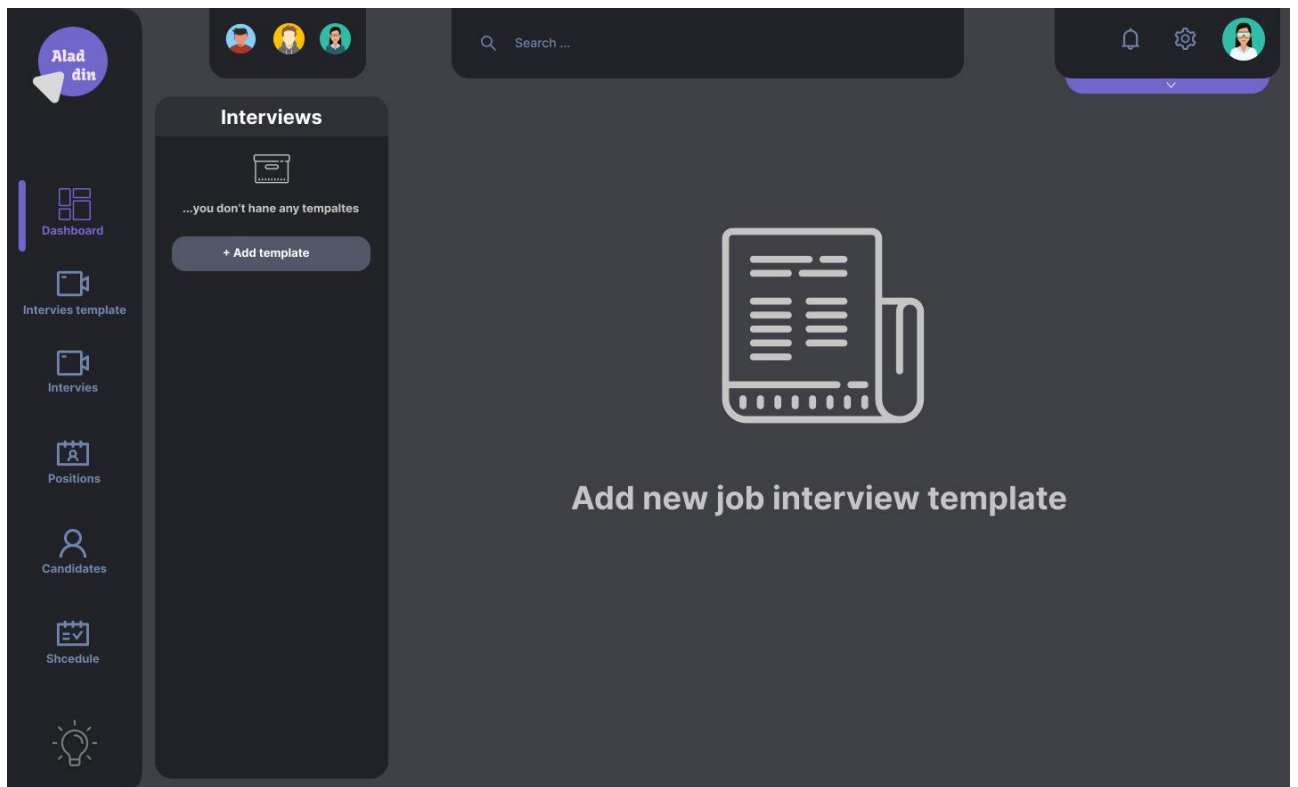
Do you have an account? [Login](#)

Додаток Г. Сторінка автентифікації



Додаток Г. Сторінки додавання шаблонів інтерв'ю





Додаток Д. Інформаційна панель

