

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
«ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В ЕКОНОМІЦІ»**

Кафедра інформаційних систем в економіці

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

Форма навчання: денна

КВАЛІФІКАЦІЙНИЙ БАКАЛАВРСЬКИЙ ПРОЕКТ

на тему

**ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ
АДАПТАЦІЄЮ НОВИХ КАДРІВ ПІДПРИЄМСТВА**

Студена Верби Андрія Юрійовича_____

Науковий керівник:

к.е.н., доцент

_____Зінов'єва І.С.

**Кваліфікаційний бакалаврський
проект допущений до захисту в
Екзаменаційній комісії з атестації
здобувачів вищої освіти**

В.о.завідувача кафедри:

к.е.н., доцент.

_____Тішков Б.О.

Київ 2022

АНОТАЦІЯ

кваліфікаційного бакалаврського проекту студента 4-го курсу
Навчально-наукового інституту «Інститут інформаційних технологій в
економіці»

Верби А.Ю., виконаного на тему:

«Проектування інформаційної системи управління адаптацією нових
працівників підприємства»

Київ: кафедра інформаційних систем в економіці, 2021 р.

Метою кваліфікаційного бакалаврського проекту є розробка інформаційної системи управління адаптацією нових кадрів підприємства.

У кваліфікаційному бакалаврському проекті обґрунтовано вибір теми дослідження та доведено її актуальність. Запропоновано варіанти вирішення проблеми, що представленої в проекті, пов'язані з використанням сучасних програмно-технічних засобів.

Кваліфікаційний бакалаврський проект складається з трьох розділів логічно зв'язаних між собою.

Вступ роботи містить обґрунтування вибору теми, її актуальність на сьогоднішній день. Також у вступі сформульовано мету і завдання кваліфікаційного бакалаврського проекту, описано наукову доцільність одержаних результатів в процесі роботи.

Перший розділ містить дослідження характеристики предметної галузі й об'єкта дослідження, аналіз сучасного стану галузі систем, задіяних у сфері роботи відділів кадрів та направлених на управління адаптаційними процесами нових співробітників. Проведений їх порівняльний аналіз.

Другий розділ присвячено моделюванню системи та обґрунтуванню методів проектування системи. В розділі продемонстровано: основний перелік вимог до проекту, а саме бізнес, функціональні та нефункціональні вимоги; діаграму зовнішніх блоків; діаграму класів; діаграму прецедентів; діаграму трасування; інформаційну модель задачі та алгоритм її розв'язання.

Третій розділ наводить технічне, програмне та інформаційне забезпечення системи. В третьому розділі представлено результати роботи над проектом, а саме: розроблену інформаційної систему для управління адаптацією нових кадрів підприємства.

У висновках міститься інформація щодо практичних застосувань системи.

Проект був розроблений за допомогою таких інструментів: Enterprise Architect, мови програмування C/C++ (QT Framework), СУБД SQLite3, за допомогою IDE VS Code та IDE CLion.

РЕФЕРАТ

Кваліфікаційний бакалаврський проект містить **73** сторінки, **11** таблиць, **20** рисунків, список літератури з **37** найменувань, **2** додатки.

ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ АДАПТАЦІЄЮ НОВИХ ПРАЦІВНИКІВ ПІДПРИЄМСТВА

Перелік ключових слів: інформаційна система, стажер, менеджер, система, відділ кадрів, адаптація, користувач, працівник.

Об'єкт дослідження: відділ кадрів організації та інформаційно-аналітичні системи, які використовуються під час роботи відділу.

Мета роботи: розробити робочий прототип інформаційної системи управління адаптацією нових кадрів підприємства.

Область застосувань: мережева система організації.

Використана апаратура та програмне забезпечення: стаціонарний ПК на базі Intel Core i3-6400, ОС Microsoft Windows 10, Excel 2016, QT, СУБД SQLite3, Enterprise Architect, GitHub.

Пропозиції щодо впровадження результатів роботи: результати, отримані в ході розробки кваліфікаційного бакалаврського проекту, можна застосувати для створення системи, що зможе надавати фахівцям відділу кадрів приймати більш точні та правильні дії відносно питань управління адаптацією нових кадрів.

Результати дослідження: спроектовано базу даних предметної області, розроблено робочий прототип інформаційної системи з графічним користувацьким інтерфейсом.

Рік виконання бакалаврського дипломного проекту – **2021**, рік захисту кваліфікаційного бакалаврського проекту – **2022**.

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	7
1.1. Характеристика предметної галузі та об'єкта дослідження.....	7
1.2. Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі.....	11
РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	19
2.1. Аналіз і специфікація вимог до інформаційної системи	19
2.1.1. Бізнес-вимоги до системи	19
2.1.2. Функціональні вимоги до системи.....	20
2.1.3. Нефункціональні вимоги до системи.....	22
2.2. Постановка та алгоритм розв'язання задачі.....	24
2.2.1. Постановка задачі	24
2.3. Моделювання інформаційної системи/підсистеми	29
РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ	35
3.1. Інформаційне забезпечення	35
3.2. Технічне забезпечення.....	40
3.3. Програмне забезпечення	42
3.4. Результати реалізації інформаційної системи/підсистеми	44
ВИСНОВКИ.....	48
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	50
ДОДАТКИ.....	54

ВСТУП

Сьогодні в Україні створення систем і зміна методів управління пов'язані з постійним науковим пошуком оптимальних рішень складних економічних і соціально-політичних проблем, докорінним оновленням усіх рівнів управління, стилів правління. У цьому контексті, особливо гостро постає завдання регулювання корпоративного управління: створення абсолютно нової системи управління або прогресивні зміни в побудові та експлуатації існуючих. Організаційне управління здійснюється шляхом організаційного проектування, реорганізації чи ліквідації одних існуючих систем і створення інших, здатних самостійно й ефективно досягати цілей у ринкових відносинах.

У сфері кадрового управління та корпоративного менеджменту важливою задачею є організація ефективного процесу адаптації нових працівників до існуючих бізнес-процесів в середині організації. Від того, наскільки якісним та продуктивним будуть реалізовані ці процеси залежить, в подальшому, і продуктивність праці майбутніх працівників та рівень кадрового менеджменту на підприємстві в цілому.

Історичний досвід самої України та сучасна зарубіжна практика свідчить, що лише за наукового супроводу інформаційних систем, а також постійного вдосконалення форм програм та методів управління можна досягти ефективності будь-якого управління, у тому числі й кадрового.

Важливість питання та ступінь його вплив на результативність роботи промислового підприємства обумовлюють *актуальність* теми кваліфікаційного бакалаврського проекту. Різноманітні аспекти адаптації персоналу досліджені в спеціалізованій науковій літературі, зокрема, у роботах Балабанової Л. В. [2], Данюк В. М. [3], Макаркої І.А [5], Паратнової О.В. [15] та ін.

Метою дослідження в рамках виконання кваліфікаційного бакалаврського проекту є розробка інформаційної системи управління

адаптацією нових кадрів підприємства. *Завдання дослідження* полягають у наступному:

- надати характеристику предметної галузі та об'єкта дослідження;
- виконати аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі;
- здійснити аналіз й представити специфікацію вимог до інформаційної системи;
- продемонструвати постановку та алгоритм розв'язання задачі;
- змодельовати інформаційну систему;
- представити інформаційне забезпечення;
- відобразити технічне та програмне забезпечення;
- навести результати реалізації інформаційної системи.

Об'єкт дослідження – процеси адаптації нових кадрів підприємства до умов роботи через навчання;

Предмет дослідження – розробка інформаційної системи відділу кадрів промислового підприємства за напрямом управління адаптацією нових кадрів та оцінки рівня їх готовності до виконання функціональних обов'язків.

Апробація результатів досліджень. Основі тези проведеного дослідження були обговорені під час участі у Всеукраїнській науково-практичній конференції «Інформаційні технології та моделювання систем» (м. Житомир, 12 травня 2022 р.), тези на тему «Проектування інформаційної системи управління адаптацією нових працівників підприємства» (додаток Б).

РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1. Характеристика предметної галузі та об'єкта дослідження

У сучасному стані української економіки, через глобальний карантин, зовнішнє середовище суспільства характеризується невизначеністю, мінливістю та невизначеністю. Крім того, така ситуація характерна для внутрішнього середовища компанії. У сьогоднішніх складних умовах дуже важливо гнучко реагувати та адаптуватися до змін, щоб уникнути кризових ситуацій та продовжити діяльність громади [3].

В Україні, як і в інших країнах, відбулося значне скорочення кількості вакантних посад і, водночас, збільшення кількості безробітних. Це посилює конкуренцію за роботу (приблизно 10 безробітних претендують на роботу). За даними Disnaker, станом на 13 травня 2020 року кількість безробітних в обліку Disnaker становила 486 800 осіб, що на 181 тис., або на 59% більше, ніж на відповідну дату попереднього року (станом на 13.2019 р. кількість безробітних становила 305,4 тис.) [4].

Термін «адаптація» походить від лат. адаптувати – пристосовувати. Це позика з біології і означає застосування, пристосування до навколишнього середовища. Адаптація до праці – це соціальний процес розвитку особистості на основі нової робочої ситуації. Вступаючи на роботу, людина активно бере участь у системі професійних і соціально-психологічних відносин певної трудової організації, засвоює нові соціальні ролі, цінності, норми, узгоджує свою особисту позицію з цілями та завданнями організації (трудового колективу), таким чином подання своїх поведінкових приписів [5].

Кваліфікована команда є одним із ключових факторів успіху розвитку будь-якої компанії. Захист людських ресурсів стає все більш актуальним, особливо в період активного зростання та розвитку компанії.

Правильним рішенням проблеми нестачі персоналу може бути конкретна робота, але розвиток співробітників відповідає стратегії компанії.

Сучасна система управління людськими ресурсами – це набір технологій, що автоматизує та полегшує роботу співробітників на всіх рівнях, від оперативного (щоденний облік даних) до стратегічного (рішення щодо розвитку бізнесу). У добре розвиненій системі робота з цією технологією підпорядковується загальній логіці управління людськими ресурсами прийнятої політики, що виражається в автоматичному електронному управлінні записами бізнес-процесів. Система людських ресурсів може ефективно спілкуватися з фінансами, виробництвом та іншими системами, які використовують дані персоналу в процесі [6].

Різні відділи кадрів використовують системи людських ресурсів для виконання таких завдань:

- Бізнес-лідери – як інструмент розвитку бізнесу, що забезпечує підтримку розробки та впровадження ефективної кадрової політики;
- Менеджери з персоналу, керівники кадрів – для ефективного впровадження процесів обліку персоналу та ведення документації, підтримки великих обсягів баз даних, швидкого доступу практично до всіх кадрових звітів;
- Головний бухгалтер та бухгалтер – для прискорення та спрощення методів обліку заробітної плати, встановлення бухгалтерського обліку, забезпечення стандартної звітності, належного обліку витрат на персонал у собівартості продукції;
- Керівник роботи та нарахування заробітної плати – займається всіма організаційними та бухгалтерськими завданнями, розвитком компанії, персоналом, витратами;
- Лінійні керівники – оцінюють рівень розвитку підлеглих, план роботи та заходи щодо розвитку підлеглих, системи стимулювання;
- Самі працівники – аналізуйте правила роботи та порівнюйте їх з іншими вимогами роботи, щоб спланувати збільшення штату співробітників компанії.

Системи управління новим персоналом реалізують такі завдання:

- Організація та управління персоналом (на всіх рівнях: регіональні відділення та інші, їх внутрішня організація);
- Облік заробітної плати;
- Кадровий облік (ведення персональних даних працівників, облік переміщень персоналу, підготовка на основі необхідної системи паперової документації);
- Звіти про робочий час (зайнятість, централізоване та автоматичне виготовлення звітів про погоду);
- План витрат на персонал;
- План роботи та моніторинг прогресу персоналу;
- Робота з копіями кадрів;
- Використання вакансій;
- Навчання персоналу;
- Система сертифікації;
- Управління компетентністю (оцінка та підтримка професійної та особистої компетентності через систему сертифікації та безперервне навчання персоналу).

На рис. 1.1 показана організація типового промислового підприємства, на якому можна побачити взаємозв'язки та зрозуміло хто та як займає місце в роботі організації [7].

В нашій роботі метою стає покращення процесу управління адаптації нових кадрів. Організаційна модель показує, що управління кадрами підпорядковується напрямку вищій ланці керівництва організації, а вже від управління відбувається інші процеси, відбір, оцінка, встановлення трудових відносин та сама адаптація нового кадра. Одним із ключових завдань є правильний підхід до нового кадра компанії, так як саме перших дотик дуже важливий у ставленні людини до організації.

Рішення про звільнення на новому місці приймають близько 80% нових робітників, практика показує, що вони приймають це рішення в перший робочий день, або – в перший тиждень. Тож у компаній насправді є максимум

тиждень, щоб справити позитивне перше враження. Ніколи не буде шансу це виправити [8].



Рисунок 1.1 – Організаційна структура підприємства

Система буде давати змогу підтримувати управління новими кадрами на всіх фазах адаптаційного процесу – починаючи від внесення персональних даних, першого дня роботи та отримання перших завдань, закінчуючи внесенням оцінок роботи людини за встановленні завдання. Коли закінчить своє втручання програма в цей процес вирішить менеджер.

В даному проекті ми охоплюємо різноманітні бізнес-процеси та інші процеси і, на нашу думку, є одним із найважливіших бізнес-процесів для підбору персоналу, який базується на перевірці кваліфікації претендентів відповідно до вимог до роботи, відбору роботи та відбору претендентів. Найбільш підходящий кандидат за навичками, спеціалізацією, особистими якостями та здібностями по відношенню до цілей та інтересів компанії. Слід зазначити, що якісний процес підбору персоналу є дорогим і досить складним процесом, до якого часто залучається багато людей, знайомих з

роботою. Слід також мати на увазі, що набір часто відбувається повільно, оскільки некваліфікованим кандидатам відмовляли на кожному етапі [9].

Одним із методів відбору персоналу є тестування. Цей метод набору використовується останнім часом в основному в західних компаніях. Цей метод дозволяє визначити найбільш точну та достовірну інформацію про професійні характеристики кандидата та здатність виконувати ту чи іншу роботу [10]. Проте наразі іспити використовуються не як незалежний інструмент найму, а як засіб отримання інформації про професійну кваліфікацію заявника. Також слід зазначити, що для проведення експертизи необхідний висновок професійного спеціаліста.

Використання матриці відповідальності «Організаційна структура – бізнес-процеси» у табл. 1.1 вказує, хто в процесі планування бере участь в описаних бізнес-процесах. Ці дані допомагають нам зрозуміти список процесів і відповідального персоналу, тому автоматизація процесів працює добре.

Таблиця 1.1

Матриця відповідальності «Організаційна структура – бізнес-процеси»

	Керівник	Працівник	Менеджер
Подати заявку		X	
Розгляд заявок			X
Запрошення на тестування			X
Проходження тестування		X	
Перегляд кандидатів	X		
Запрошення на співбесіду	X		

1.2. Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі

У різноманітних наукових публікаціях проблему кадрового забезпечення компаній та її вплив на ефективність роботи працівників у різних аспектах враховано по різному. Питанням управління персоналом присвячені

роботи дослідників: М.Д.Виноградського, С.В. Селяєва, Виноградська, О.М. Шканова [11], Л.В. Балабанова [12], про вдосконалення управління персоналом – О.В. Крушельницька, Д.П. Мельничук [13]. Велика увага приділяється також питанням кадрового налаштування. Ці питання обговорюються у наукових працях та наукових публікаціях, таких як: Ю. Конотопцева [14], О. С. Демкович [15], Н. С. Куріна, Н. Д. Дарченко [16], Д.В. Васильєва, В. А. Мирошніченко [17] та інші автори.

Наші дослідження найбільш розповсюджених інформаційних систем з управління та роботи з персоналом переконують у тому, що вони працюють на основі використання елементів традиційних інформаційних технологій, тому важливо знати їх властивості при проектуванні такої системи, а саме:

- необхідність узгодження існуючих програм з новими розробками;
- робота в неоднорідних середовищах на декількох апаратних платформах;
- відсутність різноманітності в рівнях якості та освітніх традиціях при використанні певних інструментів у команді розробників;
- значна тривалість проекту – частково через обмежені можливості команди розробників; з іншого боку, сфера діяльності споживчих організацій і різний рівень готовності окремих підрозділів до впровадження аналітичних інформаційних систем.

Попередній досвід у сфері проектування та аналізу інформаційних систем показав, що це складна і трудомістка робота. Повинні бути залучені висококваліфіковані фахівці [18]. Але донедавна проектування такої системи було інтуїтивним, використовуючи неформальні методи, засновані на мистецтві, практичному досвіді, професійному судженні та дорогому експериментальному контролю якості систем аналізу інформації. Інформаційні вимоги до користувачів також можуть бути змінені або уточнені в процесі

проектування та експлуатації, що може ускладнити розробку та обслуговування складних систем.

У сучасних умовах проектування систем інформаційного аналізу управління вищою освітою базується на методичних принципах, що характеризують інформаційні системи (загальні, часткові, організаційно-методичні), закріплених національними та міжнародними стандартами: нові принципи виконання завдання; принцип зворотного зв'язку; принципи інтегрованих систем і стратегій; принцип першого керівника; принцип одноразового введення даних; принцип повної сумісності інформації; принцип незалежності структури системи управління інформаційним аналізом; принцип узгодженості пропускну здатності системи [19].

Враховуючи ключові функції та принципи побудови системи інформаційного аналізу, можна розробити ефективну систему аналізу інформації для управління університетом, що забезпечує гнучкість в управлінні; полегшення умов праці шляхом автоматизації рутинних операцій з обробки великих обсягів даних; посилити узгоджену роботу щодо використання робочого часу працівників університету. Він модернізував технологію управління більшістю їх структурних підрозділів та реалізував усі загальні функції контролю.

Сьогодні використовуються спеціальні програми для організації роботи та управління правильним використанням людських ресурсів в сучасних компаніях, кожна велика компанія зазвичай має свій модуль, далі ми наведемо декілька прикладів таких систем, після їх опису та ознайомлення з ними нами буде проведено аналіз деяких подібних рішень та буде створена порівняльна таблиця цієї системи. Дані отримані цим шляхом дадуть нам змогу розробити найбільш гнучку свою версію, яка помістить в собі найкращі риси аналізованих інформаційних систем [20].

Інформаційна система HRIS. Єдина інформаційна система HRIS для управління персоналом і роботою. Зберігає дані, надає можливість створювати звіти та налаштовувати доступ для співробітників і персоналу. В

арсеналі інформаційної системи більше 100 інструментів, і всіма можна скористатися в єдиній програмі [21].

Інформаційна система персоналу PeopleForce (HRIS) зберігає всю інформацію про персонал в одному хмарному сервісі. Можна сказати, що вона світ, сповнений інструментів, програм і рішень для підприємств, яких немає в жодній іншій системі. Це не просто картка персональних даних, а повна база даних співробітників і кандидатів з налаштуваннями доступу та інтеграцією з популярними месенджерами, такими як Slack.

Програма дає змогу відчувати себе класною ІТ-компанією, яка завжди має доступ до послуг та інформації. Усі дані своєчасно передаються до хмарних сервісів інформаційної системи та підтримуються цілодобово. Можливість безпечно входити в систему з будь-якої точки світу, що є чудовим рішенням для міжнародних команд.

Система автоматично створює аналіз і звіти, а дані систематично оновлюються, щоб ви могли бути впевнені в релевантності даних. Позбавтеся папки з файлами на робочому столі та купи паперу біля ноутбука.

Інформаційна система Oracle E-Business Suite [22]. «Управління людьми» — це повна система організації роботи сучасних підприємств. Це прогресивний інструмент управління, який забезпечує жорсткий контроль витрат, надає достовірну оперативну інформацію для керівництва підприємства, аналізує ефективність людських ресурсів, зокрема:

- Підтримка на всіх етапах роботи: відбір та набір, навчання та розвиток, атестація, рейтингування, мотивація.
- Плануйте економічно ефективні організаційні зміни, включаючи моделювання організаційних структур та ієрархії посад.
- Кадрові книжки працівників та кандидатів із повним обліком їхньої роботи, відомостями про заробітну плату, для створення привабливого компенсаційного пакету, деталями використання робочого часу, кадровою документацією, потребами у навчанні та результатами навчання персоналу.

Розраховувати та сплачувати всі види фіксованих платежів і дозволені законодавством.

- Ведення податкового обліку та статистики за офіційними типовими формами відповідно до вимог трудового законодавства та нестандартного обліку за допомогою Orasle.
- Оперативний аналіз даних та підтримка прийняття рішень на основі технології OLAR.

Інформаційна система Ахартa People Management. Інтегрований модуль системи EQR Ахартa повністю інтегрований в системи обліку та планування компанії. Він забезпечує вирішення проблеми управління мотивацією співробітників. З моменту співбесіди кожен співробітник отримує свій профіль, який постійно контролюється модулем управління персоналом, навіть якщо працівник не працює на підприємстві, а залишається в «кадровому резерві». Крім профілів співробітників, система також надає профілі посад. Управління людськими ресурсами дозволяє автоматично порівнювати критерії різних кандидатів з профілями роботи і тим самим визначати найкращого кандидата на роботу – з точки зору набуття необхідних навичок. Цей модуль забезпечує автоматичне звітування про потреби в навчанні та ресурсах, створення груп, планування, відстеження навчання та автоматичний запис результатів в особистих профілях співробітників.

Модуль для HR Management – це автоматизована система оцінки співробітників – тести, анкети та співбесіди [23]. Має конструктор анкет, інструменти планування (розклад інтерв'ю), можливість збирати та аналізувати інформацію. Особисті результати пов'язані з профілями співробітників, що означає, що вони можуть оцінити розвиток кар'єри та досягти бажаних особистих результатів.

Крім того, можна автоматизувати управління організаційною структурою компанії – персоналом, системою оплати праці, що дозволяє графічно представити структуру компанії у вигляді ієрархічного дерева, матриці або структури проекту.

Блоки зі статистичною та управлінською звітністю відображають історію та динаміку роботи, забезпечуючи графічну інтерпретацію даних – у вигляді графіків та таблиць.

Останньою розглянутою нами стала IC HR MySAP. My SAP Human Resources є одним із модулів інтегрованої системи MySAP Business Suite і є інтегрованим рішенням для ефективного управління персоналом [24]. Система MySAP HR в першу чергу призначена для великих підприємств, для яких актуальними є завдання управління мобільним, національним персоналом.

MySAP HR може значно підвищити ефективність виконання ключових завдань з управління персоналом, зокрема:

- Робота, в тому числі в електронному вигляді;
- Управління витратами на персонал;
- Управління продуктивністю;
- система компенсації;
- Навчання та розвиток персоналу.
- Управління відносинами з персоналом:
- Система внутрішньої комунікації;
- Інструмент самообслуговування для керівників і співробітників;
- Організація заходів;
- Управління персоналом:
- Розрахунок заробітної плати;
- Підтримка аналітичного управління;
- Відповідальність кадрової політики щодо стратегічних цілей;
- Звітність та атестація працівників.

Щоб побачити переваги системи, ми створили таблицю, в якій порівнюються системи за найважливішими критеріями (Таблиця 1.2)

Таблиця 1.2

Таблиця порівняння інформаційних систем

Критерії	HRIS	Oracle E-Business Suite. Управління персоналом	Axapta HR Management	MySAP HR
Проведення тестування	+	+	+	+
Онлайн подача документів	+	+	+	-
Слідкування за роботою нового працівника	+	+	+	+
Управління витратами на персонал	-	+	+	-
Звітність	+	+	+	+
Розрахунок і виплата будь-яких видів оплати	-	+	-	-
Звітність і атестація співробітників	+	-	-	+

Отже, після аналізу існуючих систем, нами отримане розуміння, що однією з складних задач які мають вирішувати інформаційних систем є підвищення ефективності роботи організації, а саме: комплексний вплив на стратегію, мережеву інфраструктуру, організаційну структуру, системи управління і оплати, корпоративну культуру вищої освіти.

Єдиного рішення цієї проблеми не існую, проте автоматизувати діяльність відділу кадрів і створити інформаційне середовище, яке, з одного боку, надає побачити релевантних працівників, послідовну і повну інформацію, а, з іншого – необхідні інструменти для співробітників організації та менеджерів відділу. Це інформаційне середовище має дозволити нам управляти процесами, даними і людьми. Інформаційне середовище можна розглядати як можливість життєзабезпечення відділу і його управління [25].

Сьогодні більшість державних організацій використовують «часткову» автоматизацію, комп'ютеризують не всі, а лише окремі бізнес-процеси. Це призводить до дублювання і великої втрати інформації, нестачі даних для

оперативного аналізу, втрати інтелектуальної власності через обслуговування і підтримки застарілих технологій, великій залежності від людського фактору [26].

Зрозуміло, що всі системи вимагають усунення помилок і обслуговування. Якщо інформаційна система написана сучасною мовою програмування, то таку систему організація може підтримувати і налаштовувати під себе та свої задачі з залученням спеціалістів. Проте, розробка всіх компонентів в ІСУ – дуже складний та тривалий процес.

РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Аналіз і специфікація вимог до інформаційної системи

2.1.1. Бізнес-вимоги до системи

Бізнес-вимоги – це договори, цілі, які показують, навіщо потрібні певні вимоги. Головне питання, яке відповідає вимогам бізнесу – навіщо організації або користувачеві потрібна ця система. Найважливішим фактором вимог до бізнес-інформаційної системи є її призначення.

Щоб задовольнити запити компаній, потрібно не тільки розуміти мету існування інформаційної системи, а також безпосередньо розуміти до яких цілей приведе її існування, а також знайти основні процеси та аспекти самої організації. Для невеликих компаній це вирішується створенням бізнес-моделі, розумінням взаємодії кожного компоненту завантаження, об'єднанням їх у шаблони, на яких можна використовуватися створення стратегій організації. Цю модель зазвичай називають «полотно» [28].

Найпопулярнішою формою введення даних для бізнес цілей є документ. Його мета — визначити, який вихід буде вимагатися від системи, але ефект корони системи можна розвивати без додаткових умов.

Тому цей документ додається до довідкового матеріалу, в ньому описуються технічні характеристики та очікування інфраструктури, включаючи всі професійні вимоги, пов'язані з якістю послуги. До них відносяться продуктивність, адаптивність, технічне обслуговування, ефективність, безпека, надійність і гнучкість.

Як видно із рис. 2.1, для інформаційної системи, яка буде представлена в цьому бакалаврському проекті, були сформовані наступні бізнес вимоги:

- ІС має перевіряти коректність всієї інформацію, що містить;
- Система має полегшувати роботу відділу кадрів організації;

- Система має надавати всю необхідну інформацію, яка необхідна при прийнятті рішення;
- Всі модулі та система в цілому має легко оновлюватися;
- Всі данні які зберігаються та проходять через систему, мають мати надійний захист.

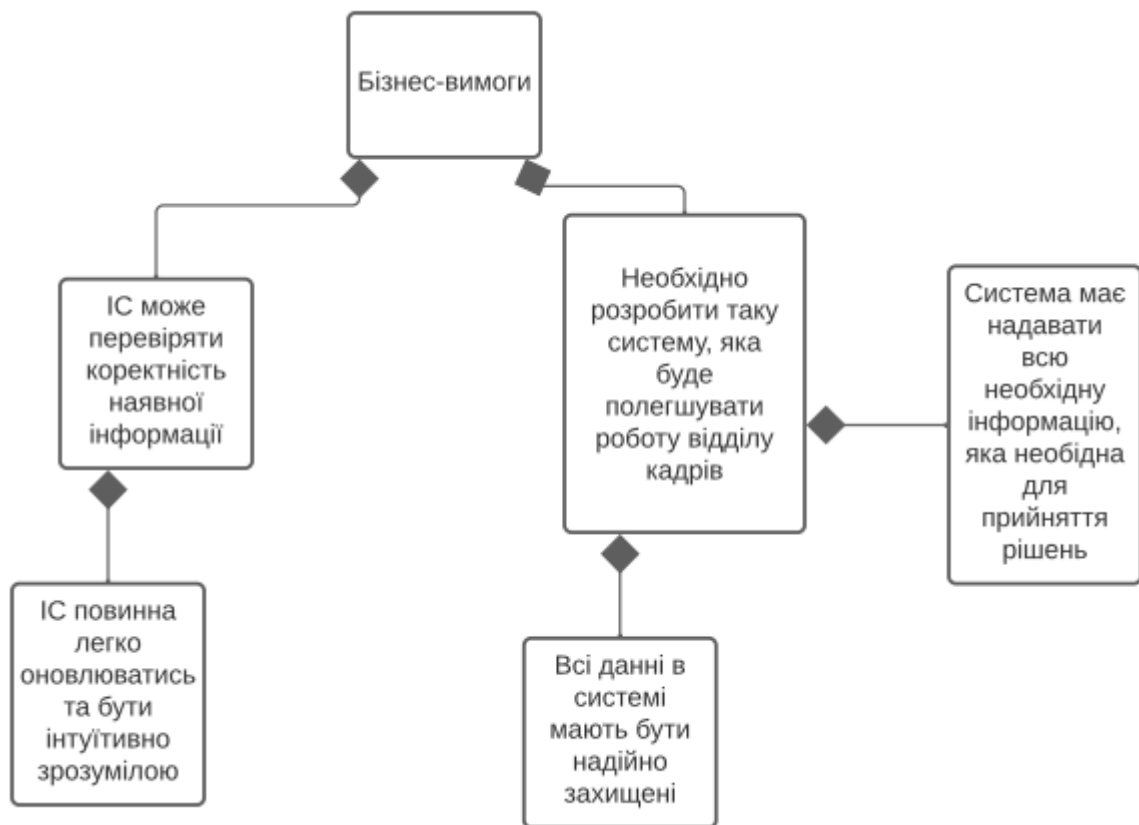


Рисунок 2.1 – Бізнес- вимоги до системи

2.1.2. Функціональні вимоги до системи

Функціональні вимоги визначають можливості програмного забезпечення, які розроблятимуть розробники для вирішення проблем користувачів, які визначені у бізнес-вимогах системи.

Способами документування функціональних вимог є файли та скрипти. Використовується специфікація вимог до програмного забезпечення у проектуванні, тестуванні, забезпеченні якості продукції, управлінні проектами та продуктами, управлінні проектами та стандартних форм реєстрації та скриптів немає, тому вони базуються на власному досвіді та змісті проектів у

програмних компаніях. Характеристики реалізації зазвичай описані сценарієм програми. Можливості використання часто можна визначити за допомогою опитувань бізнесу [29].

До розроблюваної системи були сформовані такі функціональні вимоги:

- Система має демонструвати результати працівників;
- Система може надати права доступу для користувачів;
- Система має надавати доступ до змін;
- В системі має бути два альтернативних вікна програми;
- Система має надавати можливість завантажувати та передивлятися данні;
- Система має мати автоматичне оновлення даних;
- Система повинна виводити аналітику по працівникам;
- Система повинна обробляти дані, та приводити їх до єдиного вигляду.

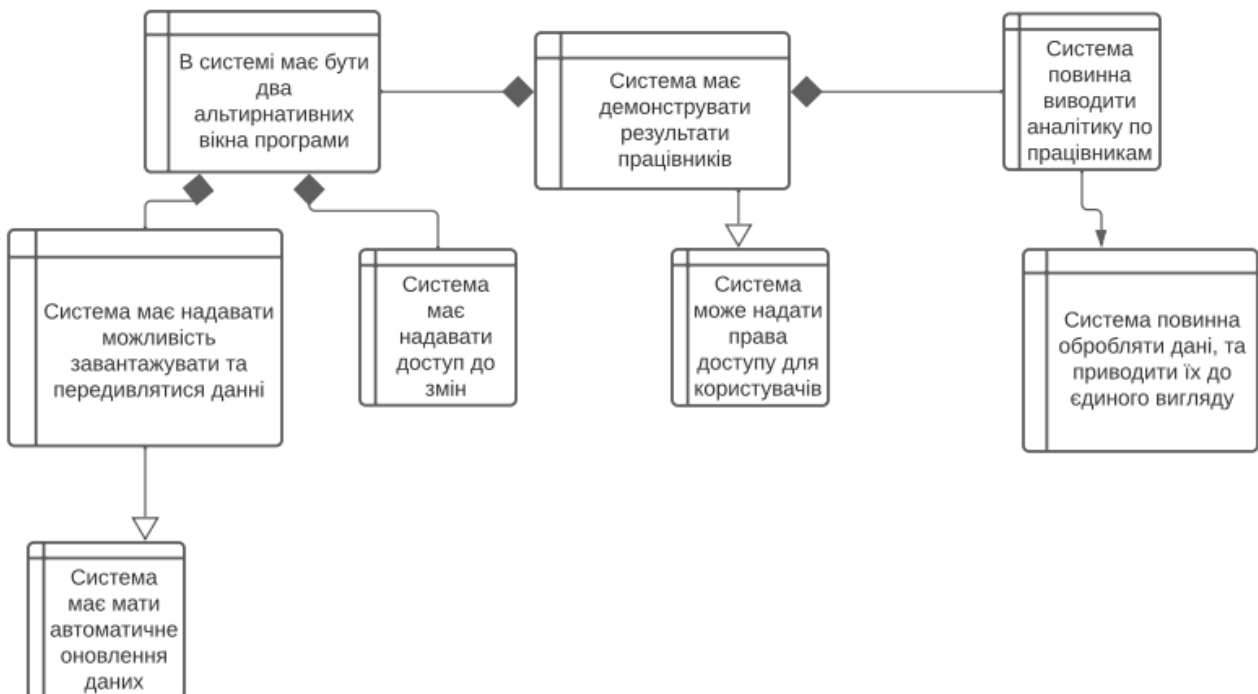


Рисунок 2.2 – Функціональні вимоги до системи

2.1.3. Нефункціональні вимоги до системи

Крім функціональних вимог, існують і нефункціональні вимоги, які описують зовнішній зв'язок між системою і зовнішнім світом, і також обмеження щодо дизайну та реалізації [30]. Нефункціональні вимоги – це основний вид вимог до програмного забезпечення, які визначають критерії які можуть бути використані для оцінки роботи системи. Нефункціональні вимоги є протилежними функціональним вимогам, вони визначають конкретну поведінку або функції системи.

Рекомендується зосередитися нефункціональні вимоги в документі, що описує варіант використання для всіх де це можливо. Якщо нефункціональні вимоги загальні, то їх включено в документ «Додатковий опис». Неактивні вимоги можуть включати інформацію про:

- Надійність – вимоги, що описують поведінку системи незалежності обставини;
- Вимоги до зберігання даних у базах даних;
- Розмір системи;
- Вимоги до конфігурації;
- Вимоги до продуктивності;
- Системні обмеження.

Нефункціональні вимоги для нашої системи сформовані наступні:

1. Вимоги до зберігання інформації:
 - Захист має бути цілодобовим
 - Має бути цілісний захист
 - Система повинна робити чек поінти
 - Повинні фіксуватися зміни
2. Вимоги до інтерфейсу програми
 - 1) Має бути можливість налаштування профілю
 - Інтерфейс системи має бути зрозумілим
 - інтерфейс має містити лише потрібні та реалізовані функції

3. Вимоги до даних
 - Данні повинні перевірятися скриптами системи
 - Система має оновлювати данні в реальному часі
4. Вимоги до слідкування дій
 - Всі дії користувачів мають зберігатися
5. Вимоги до програмних компонентів
 - Модулі в системі мають бути зрозумілі
 - Не має бути можливості вводу помилкових даних.

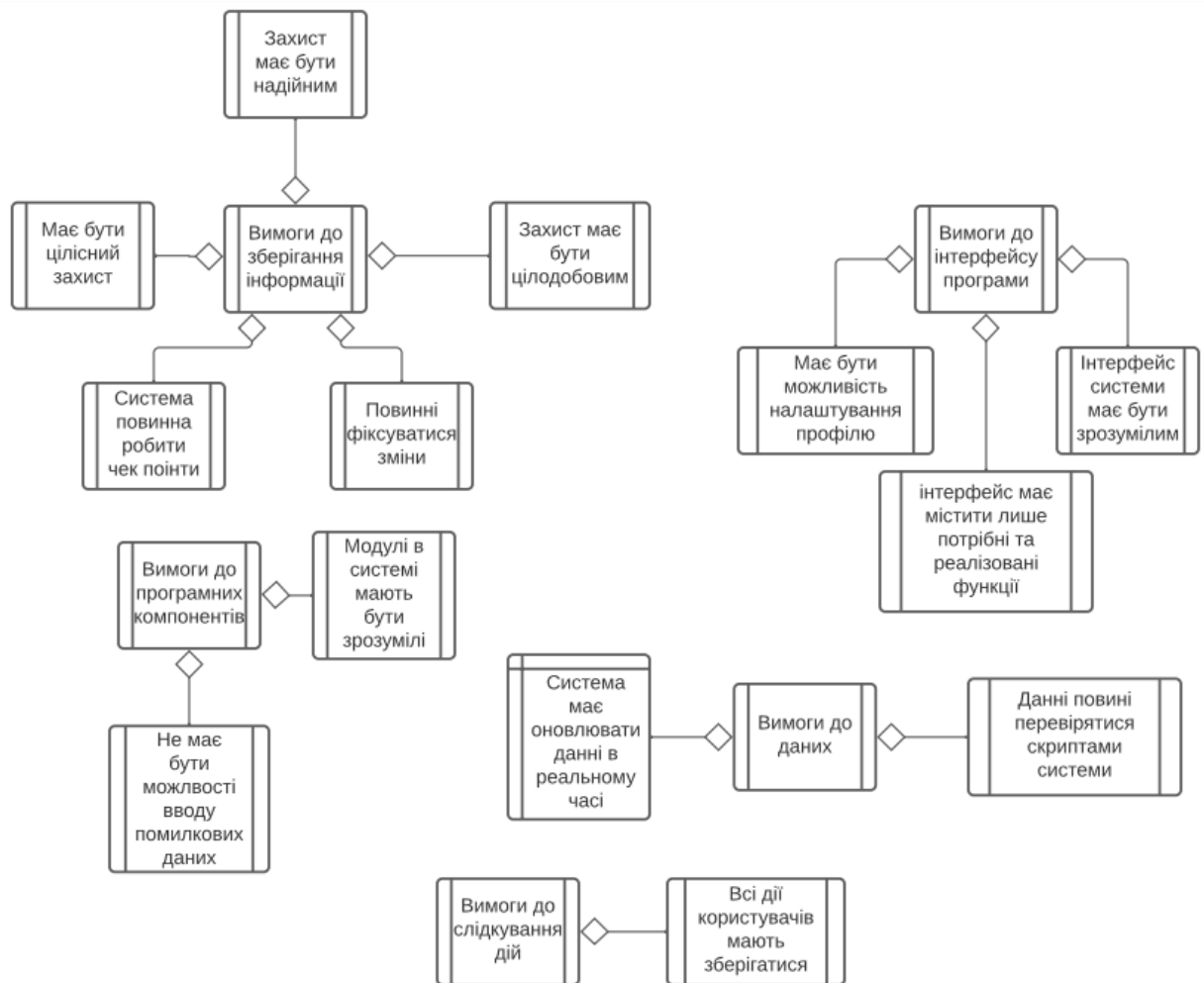


Рисунок 2. 3 – Нефункціональні вимоги до системи

2.2. Постановка та алгоритм розв'язання задачі

2.2.1. Постановка задачі

Задача «Розроблення ІС автоматизації обробки і відбору кандидатів», така програма представляю із себе комплекс рішень та інструментів, які повинні допомагати відповідним фахівцям, в нашому випадку менеджеру відділу кадрів, побачити серед кандидатів найкращого [31]. Таким кандидатом буде та людина, яка найкраще впорається з завантаженими раніше завданнями в систему. Призначення даної системи полягає у спрощені та автоматизації процесу підбору кадрів в організацію.

Реалізація задачі полягає в розробці системи, котра буде складатися з наступних елементів:

- екранних форм для працівника (кандидата);
- екранних форм для адміністратора (менеджера);
- програмної логіки;
- бази даних.

При розв'язанні задачі здійснюється керування наступними об'єктами: вхідні дані компанії, персональних даних працівника, текстуальних завдань які має виконувати новий працівник.

Вихідною інформацією системи є набір даних, який надається користувачу у вигляді екранних форм. В системі одночасно є два типи учасників системи, функціонал для котрих відрізнятиметься. Так для стажер має можливість вносити дані, переглядати дані та відповідати на форму тестів, а менеджер зможе переглядати ці данні, та бачити різницю між усіма стажерами, що допоможе йому приймати більш кращі управлінські рішення на.

Періодичність видання інформації залежить від регулярності запитів менеджера, так як інформація оновлюється постійно, але надається тільки після запиту користувача.

Взаємозв’язок задач їх вхідними та вихідними даними можна прослідкувати на інформаційній моделі задачі (рис. 2.4).

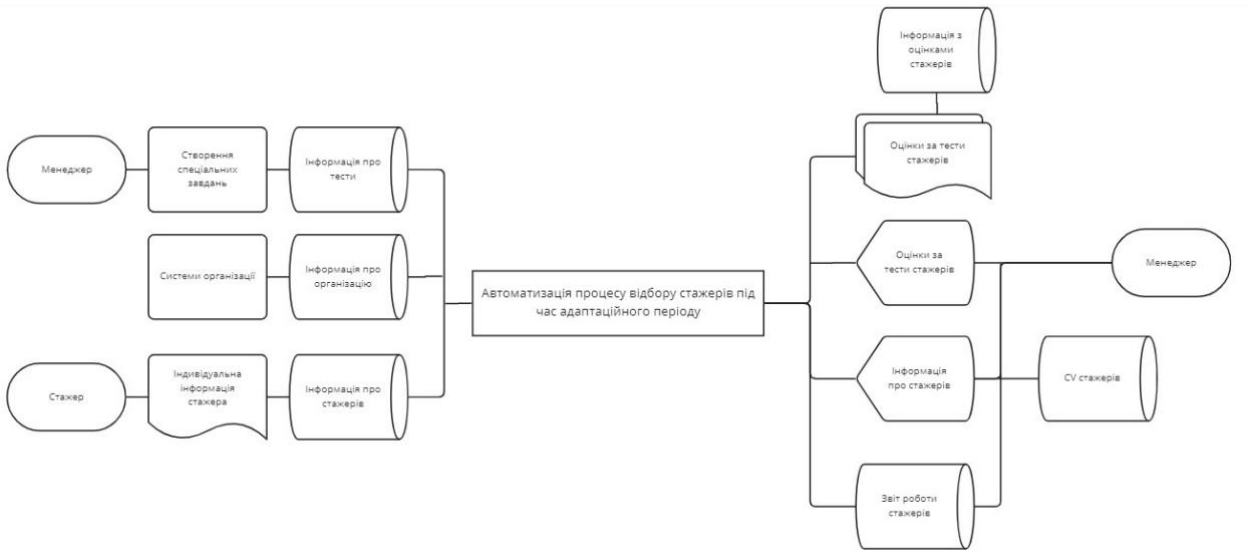


Рисунок 2.4 – Інформаційна модель процесу автоматизації інформаційно-аналітичної діяльності кафедри

Перелік і опис вихідної інформації наведено у табл. 2.1.

Таблица 2.1

Перелік та опис вихідних повідомлень.

Назва вихідного повідомлення	Ідентифікатор	Форма подання і вимоги до неї	Періодичність видання	Термін видання і допустимий час затримки	Користувач інформації
1	2	3	4	5	6
Інформація з оцінками стажерів	test_info_bal	Масив	Після проходження тесту	<1 хв	Менеджер відділу
Оцінки за тести стажерів	test_bal_#	Документ, екранна форма	За запитом користувача	<1 хв	Менеджер відділу
Інформація про стажерів	info_stajer	Документ, масив	За запитом користувача	<1 хв	Менеджер відділу
CV стажерів	cv_stajer	Документ, масив	За запитом користувача	<1 хв	Менеджер відділу
Звіт роботи стажерів	zvit_stajer	Документ, Масив	За запитом користувача	<1 хв	Менеджер відділу

Результати розв'язання задачі використовується менеджерами відділу кадрів організації, для прийняття більш ефективних рішень по персоналу.

Структурними одиницями інформації є: сума набраних балів за виконанні стажерами тестові завдання. Точність числа до сотої.

Вхідним повідомленням для здійснення запиту в системі є масив інформація про стажерів, організацію, тестів та персональних даних працівників.

Перелік і опис вхідних повідомлень наведено у табл. 2.2.

Таблиця 2.2

Перелік та опис вхідних повідомлень.

Назва вхідного повідомлення	Ідентифікатор	Форма подання	Термін і частота надходження	Джерело
Довідник Інформація про тести	Infi_ob_test	масив	За запитом користувача	менеджер
Довідник Інформація про організацію	Info_ob_orga	масив	За запитом користувача	менеджер
Довідник Інформація про стажерів	Info_ob_stajer	масив	За запитом користувача	стажер
Індивідуальна інформація стажера	Ind_info_stajer	документ, екранна форма	За запитом користувача	стажер

Інформаційна система, яка розроблюється в дипломній роботі, функціонує наступним чином: спочатку користувач проходить авторизація в системі, є два види користувачів:

- стажер;
- менеджер;

Стажер відкриває вікно, де зможе побачити інформацію про себе, передивитись завдання для тестування, завантажити звої необхідні документи, пройти тестування та подивитися оцінки за них. Оцінки потрапляють в базу даних з іншими стажерами. Зрозуміло що ця інформація вже знаходиться в профілі менеджера, там він може передавлюватися всі оцінки та інформацію як про всіх так і про кожного окремого стажера. Оцінки будуть допомагати

прийняти кращі управлінські рішення для стажера, тому менеджер має змогу сортувати стажерів по сумі їх балів.

При розв’язанні поставленої задачі в роботі використовується інформація, яка зберігається в масивах даних. Ця інформація зберігається в масивах які являються масивами даних про стажерів, їх персональну інформацію, бали за тести, інформацію завдань тестів, а також побічної інформації, що знаходиться та виходить із системи. Всі масиви інформації використовується системою по мірі необхідності. Перелік масивів, що використовуються в ході розв’язання задачі, продемонстровано в табл. 2.3.

Таблиця 2.3

Перелік масивів, використовуваних у ході розв’язання задачі

Масив	Ідентифікатор	Максимальна кількість записів
Масив персональної інформації стажерів	personal_info	-
Масив професійної інформації про стажерів	kandidat_info	-
Масив інформації про працівників та їх обов’язки	worker_info	-
Масив інформацію про спеціальний №1	test_it_otdel	-
Масив інформацію про спеціальний №2	test_fin_otdel	-
Масив інформацію про спеціальний №3	test_anticor_otdel	-

Результатом розв’язання задачі буде інформаційна система автоматизації обробки і відбору кандидатів. Вихідною інформацією системи є результати тестування кандидатів (стажерів). Результати заносяться до бази даних, частково відображаючи її користувачеві.

Алгоритм розв’язання задачі на комп’ютері представлено на рис. 2.5.

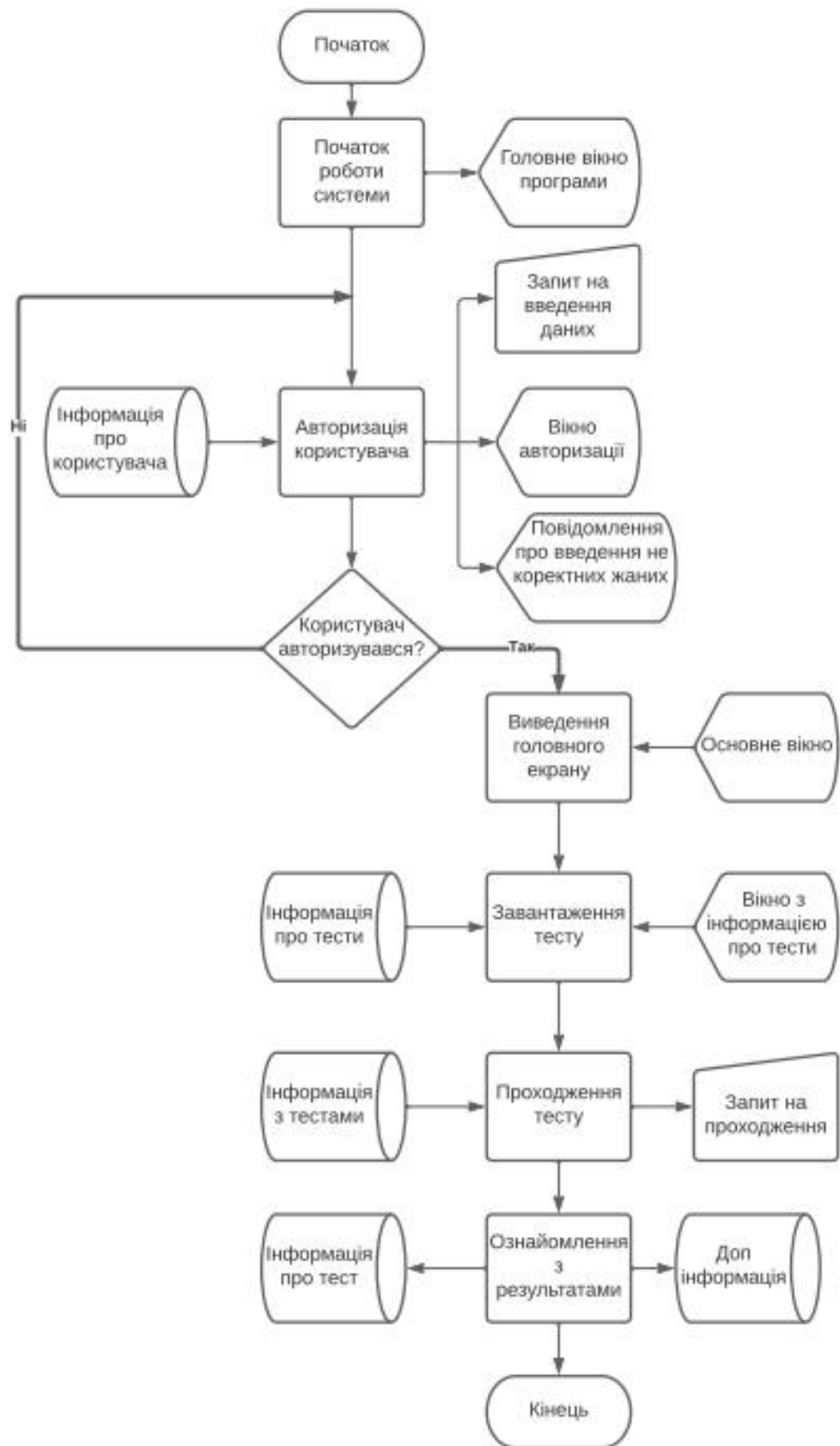


Рисунок 2.5 – Алгоритм розв’язання задачі на комп’ютері

2.3. Моделювання інформаційної системи/підсистеми

Одна з основних діаграм, що застосовуються при розробці логічної моделі інформаційної системи – діаграма прецедентів (use case diagram), її ціль, це побудови моделі концептуального рівня того, як система має працювати в середовищі.

Використання діаграми прецедентів в якості моделі системи, дає змогу зрівняти її з моделлю чорного ящика. Варіанти використання визначаються послідовністю дій, які розроблена система повинна виконати при взаємодії з відповідним актором. У цих діаграмах використовуються два основні типи учасників: варіанти дій та актори, між якими визначаються основні типи відносин.

Робочу модель інформаційної системи для відділу кадрів організації Деснянського РДА зображено в UML – діаграмі прецедентів, яка показує випадки та описує доступні дії актора. На нашій діаграмі є актор: менеджер відділу, та дії які він може робити.

Діаграма прецедентів наочно показує різні сценарії взаємодії між акторами (користувачами) і прецедентами (випадками використання); описати функціональні аспекти системи (бізнес-логіку). Прецедентна діаграма відіграє важливу роль не тільки в спілкуванні між компіляторами мови проекту та потенційними користувачами.

Модель поведінки системи продемонстрована в UML діаграмах прецедентів рис. 2.6, 2.7, які описують можливості стажера та менеджера при взаємодії з системою, та включає в себе:

З боку менеджера:

- Завантажувати тести
- Передивлятися оцінки
- Передивлятися інформацію про кандидатів
- Авторизуватися
- Вносити данні

З боку стажера:

- Завантажувати файли
- Проходити тест
- Вносити данні
- Авторизуватися, Реєструватися
- Передивлятися інформацію про тест



Рисунок 2.6 – Діаграма прецедентів (стажер)

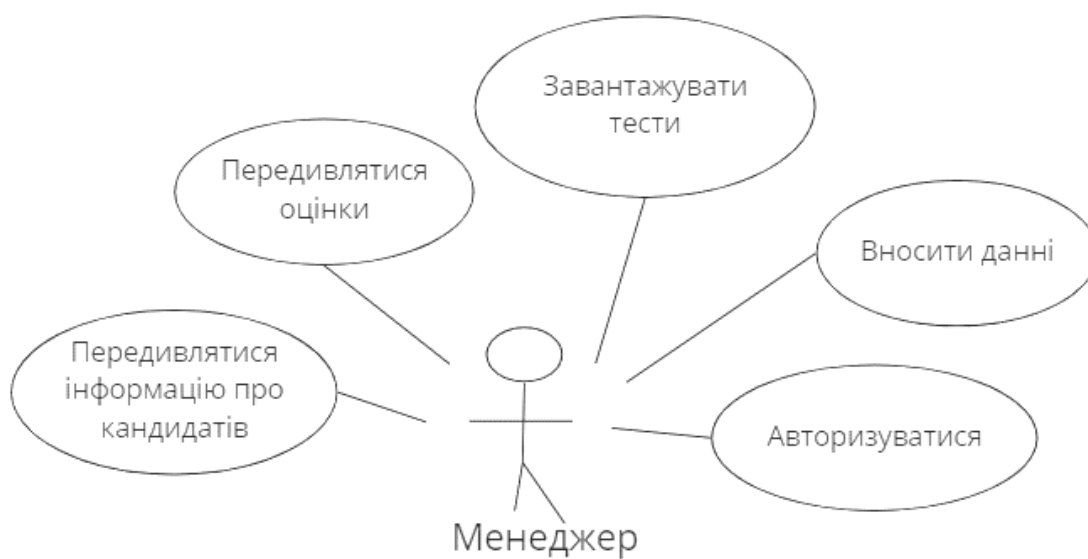


Рисунок 2.7 – Діаграма прецедентів (менеджер)

Діаграми послідовності є способом формалізації сценаріїв використання. Перевага полягає в тому, що на початковій фазі опису сценарію можна визначити склад компонентів, які взаємодіють, і описати потік повідомлень від одного компонента до іншого. Ці компоненти та потоки повідомлень згодом стануть специфічними класами (об'єктами), методами для цих об'єктів. В результаті відразу була прояснена модель системи дій (Actions), яка буде підтримувати та обробляти ці класи (об'єкти).

Діаграми послідовності в контексті UML є взаємодіями між об'єктами і використовуються для визначення потоку подій між об'єктами для заданого результату. Діаграми послідовності є важливим компонентом, який використовується в процесах, пов'язаних з аналізом, проектуванням і документацією. Діаграми послідовності також відомі як діаграми шкали часу, діаграми подій і сценарії подій.

Нами було визначено, що головною особою є менеджер, який оцінює результати тестування стажерів, нами змодельовано більш детальну його поведінку відповідно до сценаріїв виконання. На діаграмі послідовності (рис. 2.8) показана модель системних прецедентів менеджера із інформаційною системою, при аналізі стажерів.

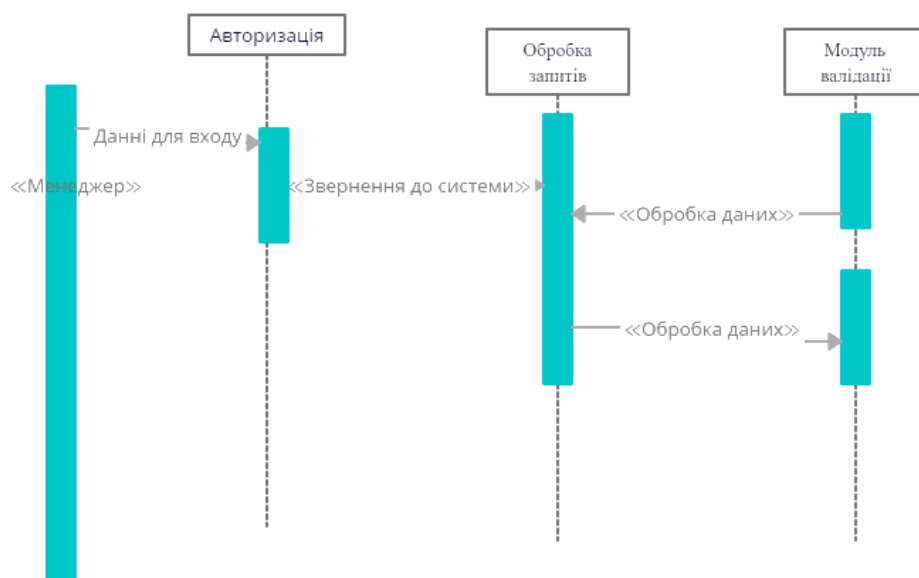


Рисунок 2.8 – Діаграма послідовності процесу авторизації менеджера

Блок-схема визначення блоків, які компоненти присутні в системі та їх взаємозв'язки. На рис. 2.9 представлена блок-схема визначення для нашої системи.

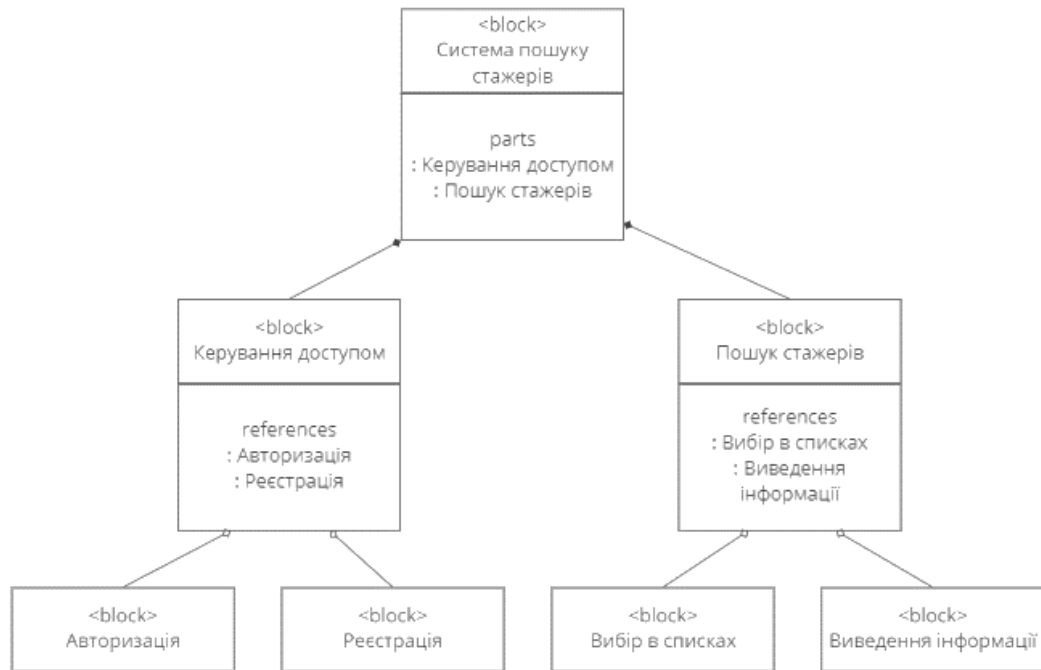


Рисунок 2.9 – Діаграма зовнішніх блоків

Важлива діаграма системи, без якої не можливо побачити внутрішній обмін даними, а саме діаграма класів, використовується для представлення статичної структури моделі системи в термінології класу об'єктно-орієнтованого проектування. Групи діаграм можуть проілюструвати та поділитися різними взаємозв'язками між природою предметної області, якими є об'єкти та підсистеми, а також описати її внутрішню структуру та типи відхилень. Сьогоднішня діаграма не містить інформації про часові аспекти функціонування системи. З цієї точки зору діаграм класів набуває подальшого розвитку концептуальна модель складної системи.

Діаграма класів – це діаграма, вершинами якої є «класифікатор» тип, який асоціюється з різними типами структуризації елементів. Дивлячись на те, які класи графіків також можуть мати окремі інтерфейси, пакети, трафік і

навіть зовнішні елементи, такі як об'єкти та посилання. Мова, що слідує заданій схемі, має можливість враховувати статичну структуру моделі системи, що обробляється. Таким чином, клас діаграм використовує графічно представлені структурні системи пов'язаних логічних моделей, які не є унікальними в часі.

Тобто, діаграма класів демонструє нам класи та їх взаємозв'язки, тим самим представляючи логічний аспект проекту. Аналіз поетапного класу використовується для встановлення загальних ролей та відповідальності за відповідність, забезпечення необхідної поведінки. На стадії формування діаграма класів використовується, щоб представити структуру класів, які формують архітектуру системи.

Фізична модель бази даних системи взаємодіє на СУБД MySQL містять такі таблиці: «personal_info», «kandidat_info», «worker_info», «test_it_otdel», «test_fin_otdel» «test_antikor_otdel», їх взаємозв'язки зображені на рис. 2.10.

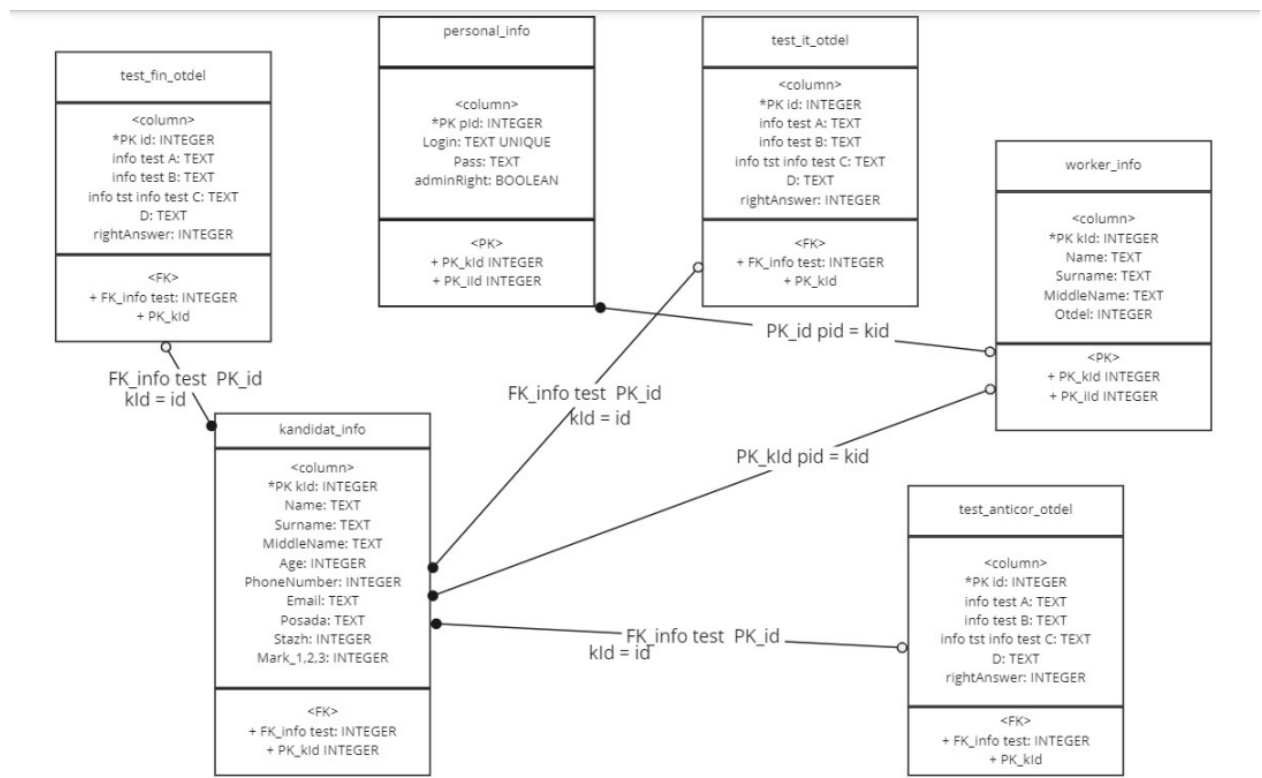


Рисунок 2.10 – Діаграма класів

Етапом який не можливо пропустити при керуванні розробкою системи є трасування вимог, вона представляю із себе діаграму відстеження коректності завдань та реалізації вимог системи під час життєвого циклу системи під час перевірки систем з встановленням вимог.

Під час реалізації діаграми трасування застосовується діаграма функціональних вимог та діаграма прецедентів. Завдяки такому використанню взаємозв'язку, діаграми представляють основні можливості системи. Верифікація створених вимог та їх співвідношення із системними частинами показано на діаграмі трасування (рис. 2.11). Тому на рисунку, діаграма демонструє високо рівневі зв'язки між вимогами та прецедентами.

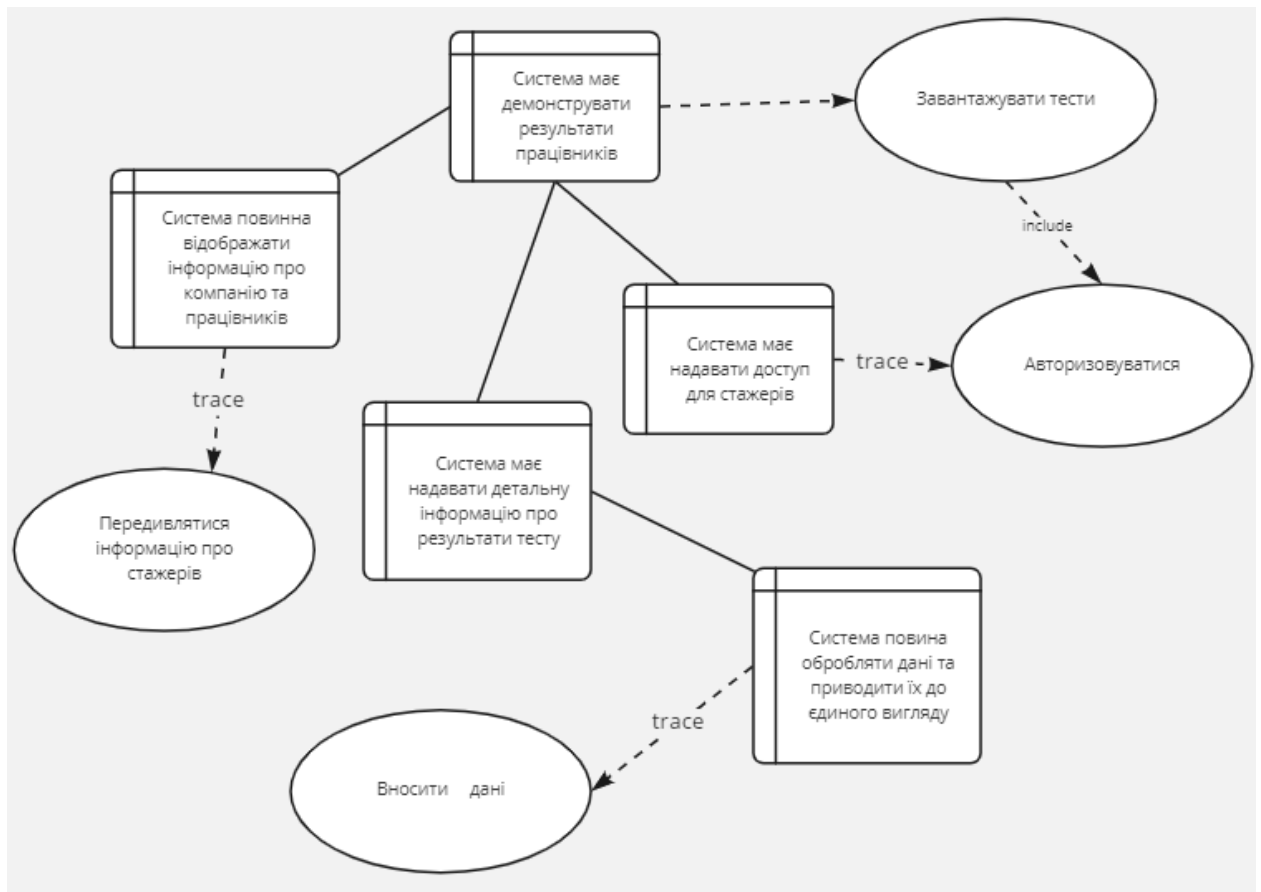


Рисунок 2.11 – Діаграма трасування

РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

3.1. Інформаційне забезпечення

Розглянута в бакалаврській роботі система, фізично має розташування у локальній мережі організації та її пристроях, тому основні носії інформації – це інформація баз даних на фізичних носіях організації. Таке рішення по розміщенню баз даних виходить із вимог до захисту даних, зручності їх витягування, швидкості обробки запитів, а також запитів організації до зовнішніх та внутрішніх факторів (DDos-атак, поломки фізичних носіїв інформації та інше). Також розміщення даних на фізичному носії користувача гарантує доступ до них в будь-який час.

Уся вихідна та вхідна інформація інформаційної системи зберігається у централізованій базі даних, масиви даних передаються та отримуються через мережу локальну мережу з підключенням до Інтернет.

Для реалізації заданої системи класифікації та кодування нами було обрано порядковий метод.

Структура інформаційних масивів системи наведена в таблицях 3.1 – 3.6.

Таблиця 3.1

Масив інформація про користувачів

Опис масиву

«Масив інформація про користувачів»

Назва масиву — масив інформація по користувачів.

Позначення масиву — **personal_info**.

Назва носія інформації — БД.

Обсяг масиву — 4 записи.

Довжина запису — не обмежено.

Метод організації — послідовний.

Ключі впорядкування — ID стажеру, ID тестування.

Ідентифікатор індексного масиву — **personal_info**.

Найменування	Ідентифікатор у програмі	Умове позначення у формулах	Формат	Бізнес-правила				Логічні та систематичні зв'язки
				Первинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
ID користувача	pId	-	int	PK		так		-
Лог-ім'я користувача	Login	-	varchar	-		так		-
Пароль користувача	Pass	-	varchar	-		так		-
Чи являється адміністратором	adminRight	-	bool	-		так		-

Таблиця 3.2

Масив інформація про стажерів

Опис масиву

«Масив інформація стажера»

Назва масиву — масив інформація стажера.

Позначення масиву — **kandidat_info**.

Назва носія інформації — БД.

Обсяг масиву — 10 записів.

Довжина запису — не обмежено.

Метод організації — послідовний.

Ключі впорядкування — ID користувача,.

Ідентифікатор індексного масиву — **kandidat_info**.

Найменування	Ідентифікатор у програмі	Умове позначення у формулах	Формат	Бізнес-правила				Логічні та систематичні зв'язки
				Первинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
ID користувача	kId	-	int	PK		так		-
Ім'я стажера	Name	-	varchar	-		так		-
Прізвище стажера	Surname	-	varchar	-		так		-
По батькові стажера	MiddleName	-	varchar	-		так		-

Найменування	Ідентифікатор у програмі	Умове позначення у формулах	Формат	Бізнес-правила				Логічні та систематичні зв'язки
				Первинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
Вік стажера	Age	-	Int	-		Так		-
Телефон стажера	PhoneNumber	-	Int	-		Так		-
Почта стажера	Email	-	varchar	-		Так		-
Посада стажера	Posada	-	Varchar	-		Так		-
Стаж стажера	Stazh	-	int	-		Так		-
Оцінка тестів	Mark_n	-	int	-		так		-

Таблиця 3.3

Масив інформація про працівників

Опис масиву

«Масив інформація про працівників»

Назва масиву — масив інформація про працівників.

Позначення масиву — **worker_info**.

Назва носія інформації — БД.

Обсяг масиву — 10 записів.

Довжина запису — не обмежено.

Метод організації — послідовний.

Ключі впорядкування — ID користувача,.

Ідентифікатор індексного масиву — **worker_info**.

Найменування	Ідентифікатор у програмі	Умове позначення у формулах	Формат	Бізнес-правила				Логічні та систематичні зв'язки
				Первинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
ID користувача	kId	-	int	PK		так		-
Ім'я	Name	-	varchar	-		так		-
Прізвище	Surname	-	varchar	-		так		-

Найменування	Ідентифікатор у програмі	Умове позначення у формулах	Формат	Бізнес-правила				Логічні та систематичні зв'язки
				Первинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
По батькові	MiddleName	-	varchar	-		так		-
Вік	Age	-	Int	-		Так		-
Телефон	PhoneNumber	-	Int	-		Так		-
Почта	Email	-	varchar	-		Так		-
Посада	Posada	-	Varchar	-		Так		-
Стаж	Stazh	-	int	-		Так		-
Відділ	Otdel	-	Varchar	-		так		-

Таблиця 3.4

Масив інформація про тести

Опис масиву

«Масив інформація про тести»

Назва масиву — масив інформація про тести.

Позначення масиву — **test_it_otdel**.

Назва носія інформації — БД.

Обсяг масиву — 4 записів.

Довжина запису — не обмежено.

Метод організації — послідовний.

Ключі впорядкування — ID тест, користувач.

Ідентифікатор індексного масиву — **test_it_otdel**.

Найменування	Ідентифікатор у програмі	Умове позначення у формулах	Формат	Бізнес-правила				Логічні та систематичні зв'язки
				Первинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
ID тесту	id	-	int	PK		так		-

Найменування	Ідентифікатор у програмі	Умове позначення у формулах	Формат	Бізнес-правила				Логічні та систематичні зв'язки
				Первинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
Інформація про тест №1	Info_test_name_A	-	varchar	-		так		-
Інформація про тест №2	Info_test_name_B	-	varchar	-		так		-
Інформація про тест №3	Info_test_name_C	-	varchar	-		так		-

Таблиця 3.5

Масив інформація про організацію

Опис масиву

«Масив інформація про організацію»

Назва масиву — масив інформація про організацію.

Позначення масиву — **org_info**.

Назва носія інформації — БД.

Обсяг масиву — 5 записів.

Довжина запису — не обмежено.

Метод організації — послідовний.

Ключі впорядкування — ID користувача.

Ідентифікатор індексного масиву — **org_info**.

Найменування	Ідентифікатор у програмі	Умове позначення у формулах	Формат	Бізнес-правила				Логічні та систематичні зв'язки
				Первинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
ID користувача	kId	-	int	PK		так		-
ID Відділ	Id_vid	-	int	-		так		-
Інформація про відділ	Info_vid_org	-	varchar	-		так		-
Керівник	Keriv_vid_org	-	varchar	-		так		-
Кількість працівників	Kilk_vid_org	-	Int	-		Так		-

Таблиця 3.6

Масив звіт стажування

Опис масиву

«Масив інформація звіт стажування»

Назва масиву — масив звіт стажування.

Позначення масиву — **stajer_info**.

Назва носія інформації — БД.

Обсяг масиву — 10 записів.

Довжина запису — не обмежено.

Метод організації — послідовний.

Ключі впорядкування — ID користувача,.

Ідентифікатор індексного масиву — **stajer_info**.

Найменування	Ідентифікатор у програмі	Умове позначення у формулах	Формат	Бізнес-правила				Логічні та систематичні зв'язки
				Первинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
ID стажера	sId	-	int	PK		так		-
Ім'я стажера	Name	-	varchar	-		так		-
Прізвище стажера	Surname	-	varchar	-		так		-
По батькові стажера	MiddleName	-	varchar	-		так		-
Вік стажера	Age	-	Int	-		Так		-
Пропозиція по стажеру	Prop_staj	-	varchar	-		Так		-
CV стажера	Cv_staj	-	Varchar	-		Так		-
Посада стажера	Posada	-	Varchar	-		Так		-
Стаж стажера	Stazh	-	int	-		Так		-
Оцінка тестів	Mark_n	-	int	-		так		-

3.2. Технічне забезпечення

Для реалізації поставленої задачі по розробленні інформаційної системи на технічному рівні нам необхідне таке фізичне забезпечення:

З боку користувача (стажера, менеджера) системи:

- персональний комп'ютер на базі Windows;

- підключення до мережі Інтернету (стандартне мережеве обладнання).
- З боку інформаційної системи:
- підключення до мережі Інтернет.

Система представляється користувачеві надалі менеджер у вигляді інтерфейс-програми. Менеджер може працювати з системою без зайвих доп систем, лише додаток, відкривши програму на пристрої. Таким чином, вимоги до персонального пристрою користувача не дуже великі.

Технічні вимоги до пристрою забезпечуються локальним сервером користувача, для безперебійної роботи будить достатньо наступних характеристик комп'ютера:

- 3-х ядерний процесор;
- 6 GB оперативної пам'яті DDR3;
- 2 GB місця на жорсткому диску;
- пропускна здатність інтернет- з'єднання від 10 –ти Mbit.

Робота інформаційної системи відбувається на локальній машині менеджера. Такий вид роботи системи забезпечить цілісність даних, режим доступу до системи у реальному часі та захист від зовнішніх загроз та ддос атак.

Структура технічного забезпечення наведена на рис. 3.1.

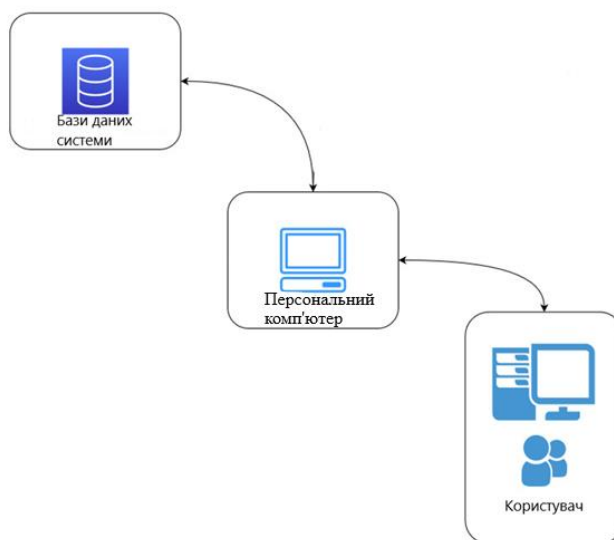


Рисунок – 3.1 Схема технічного забезпечення

3.3. Програмне забезпечення

Для отримання заданої системи, було реалізовано передню частину (візуальні вікна), задню частину (бази даних), для цього були обрані наступні засоби:

Для розробки Backend частини:

- C/C++;
- QT Framework.
- STL.
- UNIX Core.
- Patterns.

Для розробки Frontend частини:

- HTML5;
- CSS3;
- QML;

Для розробки бази даних системи:

- SQLite3;

Загальні інструменти для розробки:

- IDE CLion;
- IDE VSCode;
- Windows 10.
- CMake version 3.17.2;
- GDB.
- QT Creator.
- GitHub

При написанні Backend частини системи використовувався QT Framework –цей фрейворк містить повний інструментарій для написання високо інтуїтивних та модернізованих класів бібліотеки мови C++, а також завантажений API для полегшення розробки додатків [32]. Qt забезпечує інтуїтивний, легко підтримуваний і багатогранний код з високою продуктивністю під час написання виконання та невеликим розміром код, а

також чудово підтримується операційною системою Windows. Було обрано мову C ++ – тому що вона високого рівня програмування з підтримкою багатьох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної. Разом з тим залучено STL (Standard Template Library), Unix Core та їх шаблон коду [33].

Для написання Frontend частини системи використано стандартний стек технологій (HTML5, CSS3, QML) на базі фреймворку QT. За допомогою цих інструментів було побудовано весь інтерфейс нашої програми в досить короткий термін, саме тому нами була обрана дана технологія для залучена в процесі реалізації проекту.

Основними критеріями при виборі БД для реалізації проекту були: швидкодія, стійкість до високих навантажень та сумісність з QT Framework.

Бази даних SQLite3 розроблені для підвищення швидкодії обробки великих та малих баз даних. Вона використовується, для створення будь якого виду додатків, оскільки має сильну підтримку з боку різних мов програмування [34].

Для основних IDE при написанні програми були взяті CLion від компанії JetBrains, вона надає можливість легко і зручно писати і тестувати код, використовуючи вбудовану GDB систему для VScode від компанії Microsoft, тому, за допомогою неї можна відстежити дії в базі даних, переглядати масиви та писати SQL запити [35].

Для розробки інтерфейсів було використано Qt Creator – це між платформне інтегроване середовище (IDE), створене для програмістів з досвідом [36]. Qt Creator функціонує на всіх найвідоміших операційних системах: Windows, Linux та macOS і дозволяє розробникам реалізовувати додатки на ПК, мобільних та веб платформах.

Для якісної розробка додатку було використано систему GitHub – це з найбільший в світі веб-сервіс для спільної розробки програмних продуктів [37]. Оснований на системі керування версіями Git. Даний сервіс є безкоштовним для проектів з відкритим вихідним кодом, надає користувачам

усі свої можливості, а для окремих індивідуальних проектів пропонуються різні платні допоміжні можливості.

3.4. Результати реалізації інформаційної системи/підсистеми

Демонстрація готового прототипу програми наведена на рис 3.2-3.8.

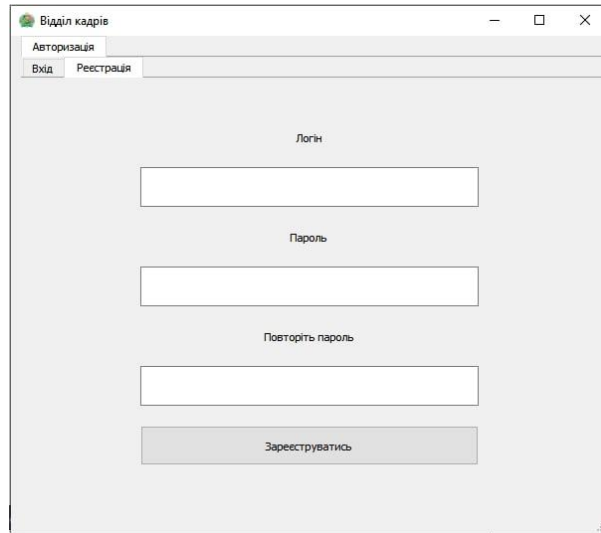


Рис 3.2. – Вікно реєстрації нового користувача

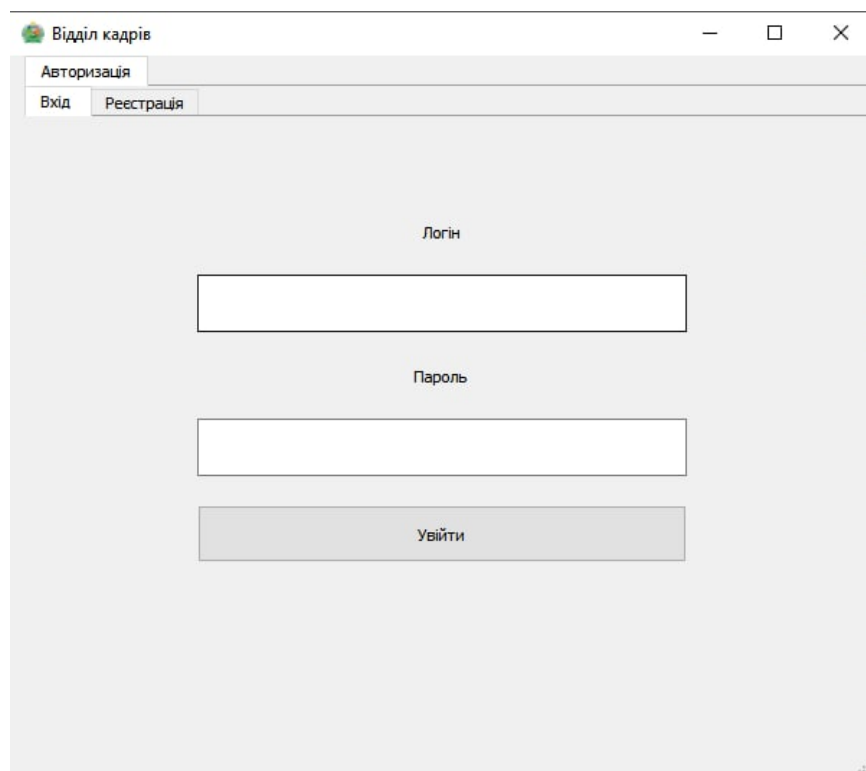


Рис 3.3 – Вікно авторизації існуючого користувача

Відділ кадрів

Залишити заявку

Ім'я	Телефон	Резюме
Олег	380961212133	Прикріпити
По-батькові	Email	Коментарі
Юрійович	oleg@gmail.com	
Прізвище	Посада	
Приходько	Розробник	
Вік	Стаж роботи	
22 років	5 років	
Подать заявку		

Рис 3.4 – Головне вікно користувача (стажер)

Відділ кадрів

Список кандидатів

Кандидати	Деталі кандидата
Олег Сергійович Сайгак	ПІБ
Аркадій Володимирович Сайгак	Олег Юрійович Приходько
Сергій Георгійович Сайгак	Посада та стаж
Георгій Дмитрович Сайгак	Розробник, 5 роки
Дмитро Олегович Сайгак	Коментар від стажера
Олег Юрійович Приходько	
	Переглянути заявку

Рис 3.5 – Головне вікно користувача (менеджер)

Відділ кадрів

Інформація про оцінку

ПІБ

Олег Юрійович Приходько

Посада	Телефон
Розробник	380961212133
Стаж	Email
5 років	oleg@gmail.com
Оцінки за тест	Резюме

Загальна = 0 балів

Завантажити

Прийнято

Відхилено

Рис 3.6 – Вікно з даними стажера (менеджер)

Відділ кадрів

Тестування

Тест №1

Питання №1

В новинах написали, що НБУ знову знизив облікову ставку. Що це означає?

A ☒ Відсотки за кредитами стануть меншими

B ☐ В Україні знизяться ціни

C ☐ Кешбек на покупки збільшиться

D ☐ Ставки за депозитами зростуть

Старт

Завершити тестування

Рис 3.7 – Вікно з тестуванням (стажер)

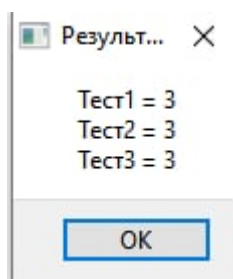


Рис 3.8 – Вікно з оцінками за тест (менеджер)

ВИСНОВКИ

У сучасному світі в функціонуванні підприємств відбуваються істотні зміни загальної парадигми управління. Сьогодні персонал представляє із себе один з головніших ресурсів будь-якої організації, саме на цьому ресурсі базують увесь економічний успіх всієї організації. Тому, персоналом треба вміти правильно та раціонально вправляти, створювати для нього необхідні умови, що покращить його розвиток та вкладати зусилля у цей розвиток.

Для того, щоб організація працювала більш результативніше, необхідно вдосконалити систему управління персоналом. Як нами було визначено, управління персоналом – це система взаємозалежних, організаційно-економічних і соціальних заходів по створенню умов для нормального функціонування, розвитку й ефективного використання потенціалу живої сили на рівні організації.

Головними принципами побудови сучасної інформаційної системи для управління персоналом є: ефективність відбору і розміщення працівників; справедливість їх оцінювання та мотивації їх інтересів. професійний ріст працівників відповідно до їх результатів, підвищення кваліфікації, здібностей, особистих інтересів. Швидке й ефективне розв'язання особистих проблем також займають велику частку в побудові взаємовідносин..

В ході робіт над кваліфікаційним бакалаврським проектом, було досліджено предметну область теми та здійснено повний цикл розробки інформаційної системи автоматизації управління адаптацією нових кадрів підприємства, від моделювання та постановки задачі до фактичної реалізації програмного продукту.

Розроблена інформаційна система орієнтована на задачі управління адаптацією персоналу. Порівнюючи нашу систему з аналогічними системами на ринку, можна виділити декілька переваг які ми спроектували:

По-перше, це – інтуїтивний та мінімалістичний інтерфейс системи. Ця, на перший погляд, незначна відмінність є досить суттєвою перевагою для непрофесійних користувачів комп'ютерів та спеціального ПО.

По-друге, дана система є однією із перших представників даної ланки на державному рівні, для нас це стало досить гарним колодязем опиту, можливо в майбутньому підтримка та популяризація таких інформаційних систем в державі призведе до зростання рівня залучення іт – сектору.

Отже, спроектована в результаті дипломного бакалаврського проекту система має конкретну задачу та широкий потенціал для розширення роботи таких систем в сфері управління нових кадрів в організаціях.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Конституція України: [зі змін. та допов., внесеними Законом України від 1 лют. 2011 р. № 2952-VI]. – Харків : Фактор, 2011. – 118 с.
2. Господарський кодекс України // Офіційний вісник України. – 2003. – № 11. – Ст. 462.
2. Балабанова Л.В. Стратегічне управління персоналом.
3. Данюк В.М., Петюх В.М., Цимбалюк С.О. Менеджмент персоналу, навч. посіб. – К.: КНЕУ, 2015. – 398 с.
4. Ofitsiynyysayt «Derzhavnyy tsentr zaynyatosti» [Official site of the State Employment Center]. Elektronnyy resurs [Electronic resource]. Available at: <https://www.dcz.gov.ua/novyna/nelegkyy-poshuk-komu-vdayetsya-znayty-robotu-na-karantyni> (accessed 26 June 2020).
5. Стратегія виробничої адаптації персоналу підприємства в сучасних умовах розвитку економіки / І.А. Маркіна, В.В. Шкурко // Економіка України: реалії перспективи розвитку ринкових відносин. Матеріали IV Всеукраїнської наукової конференції.
6. Старобинський Є.Є. Як управляти персоналом. — М: ИнтелСинтез, 2019. — 338 с.
7. Грачев М.В. Суперкадры / М.В. Грачев. — М.: Финпресс, 2021. — 291 с.
8. Василенко В.А. Теорія і практика розробки управлінських рішень: Навч. посіб. / В.А. Василенко. – К., 2012. – С. 37-44.
9. Демкович О.С. Сутність та основні чинники адаптації персоналу підприємств у ринкових умовах господарювання. Вісник Нац. ун-ту “Львівська політехніка”. № 727. С. 46–48.
10. Гризовська Л.О. Актуальні проблеми управління розвитком персоналу / Л.О. Гризовська // Вісник Хмельницького національного університету. – 2011. – № 5. – Т. 2. – С. 192–196

11. Оптимальна складність управлінських структур / М.Райс // Проблеми теорії і практики управління. – 2014. – №5. – С. 14-16.
12. Талюпа Н. Сучасні підходи до удосконалення технології управління / Н. Талюпа // Інвестиції: практика та досвід. – 2015. – №8. – С. 49-50.
13. Одегов Ю. Г., Карташова Л. В. Управління персоналом, оцінка ефективності – Навч. посібник для вузів. – М: Іспит, 2018.- 218 с.
14. Одинцов, А.А. Менеджмент організації. Введення в спеціальність: Навч. посібник- М .: Іспит, 2019.- 320 с.
15. Паратнова О. Успішна адаптація: критерії та «підводні камені» / О. Паратнова // Управління людським потенціалом. – 2017. – №3. – С. 198 – 203.
16. Поршнева А.Г., Румянцева З.П., Саломатина Н.А., Кібанов А.Я. – Управління організацією. – М .: Іспит, 2020.-318с.
17. Рутицька В.В. Процес адаптації співробітника // Менеджер по персоналу. – 2011. – №10. – С. 30-37.
18. Сатієв, С.В. Хто на новенького, або наставництво з перших днів / С.В. Сатієв, Ю.Ф. Лаврентьєва // Довідник кадровика. 2019. – № 9. – С.107 – 111.
19. Співак В. А. Особистість і підприємство: фактори успішної взаємодії (поведінкові основи управління персоналом). – СПб, 2019. – 252 с.
20. Хміль Ф. Управління персоналом: Підручник – М .: Академвидав, 2018.- 487 с.
21. Юрасов, І.Д. Моніторинг адаптації «синіх комірців» / І.Д. Юрасов // Довідник з управління персоналом. 2018.- № 9. С. 21-27.
22. Понаморенко В. С. Проектування інформаційних систем / В. С. Понаморенко., 2017. – 432 с.
23. Макаренко М.В. Формування механізму управління ефективним функціонуванням підприємства // Актуальні проблеми економіки. – 2015. – №1. – С. 126-135.

24. Eden Dahlstrom, Stephen di Filippo, Consumerization of Information Technology/BYOD, EDUCAUSE
25. Galliers. R.D., Patterson, E.M. and Reponen, T. (2018) Strategic information systems workshops: lessons from three cases. International Journal of Information Management, 14, 51-66.
26. SySML Language [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <http://www.omgsysml.org/index.htm>
27. Enterprise Architect UML tutorial [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://sparxsystems.com/resources/tutorials/uml/part1.html>
28. Visual Paradigm [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://www.visual-paradigm.com/solution/freeumltool/>
29. Espresif офіційний сайт [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://www.espressif.com/>
30. Intel's incredible Edison PC is the size of an SD card [Електронний ресурс] // Stuff. – 2014. – Режим доступу до ресурсу: <https://www.stuff.tv/news/intels-incredible-edison-pc-size-sd-card>
31. Прайс, Джейсон Oracle Database 11g SQL. Оператори SQL і програми PL / SQL / Джейсон Прайс. – М .: ЛОРИ, 2014. – 688 с.
32. Прайс, Джейсон Oracle Database 11g: SQL. Оператори SQL і програми PL / SQL / Джейсон Прайс. – М .: ЛОРИ, 2016. – 660 с.
33. Пржиялковского, В. В. Введения в Oracle SQL / В.В. Пржиялковского. – М .: Біном. Лабораторія знань, Інтернет-університет інформаційних технологій, 2013. – 320 с.
34. Прибув, Білл Oracle PL / SQL. Для професіоналів / Білл Прибув. – М .: Питер, 2014. – 725 с.
35. Проектування і реалізація баз даних Microsoft SQL Server 2000. Навчальний курс MCSE (+ CD-ROM). – М .: Російська Редакція, 2013. – 664 с.
36. Саураб, Гупта Oracle PL / SQL. Керівництво для розробників / Гупта Саураб. – М .: ЛОРИ, 2014. – 978 с.

37. Мартишин, С. А. Проектування і реалізація баз даних в СУБД MySQL з використанням MySQL Workbench. Навчальний посібник / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. – М .: Форум, Инфра-М, 2015. – 160 с.

ДОДАТКИ

Фрагмент лістингу програмного коду розробленої інформаційної системи

```

#include "mainwindow.h"
#include "ui_mainwindow.h"

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow),
    m_db(QSqlDatabase::addDatabase("QSQLITE"))
{
    ui->setupUi(this);

    m_db.setHostName("localhost");
    m_db.setDatabaseName("otdel_kadrov_DB");
    bool ok = m_db.open();

    if (ok)
    {
        creationOfTables();
        //      insertIntoGameInfo("CS GO", "Vavle", "shooter", 2012, 10, 25, 20000, "Counter-
Strike: Global Offensive (CS: GO) expands upon the team-based action gameplay that it pioneered
when it was launched 19 years ago. CS: GO features new maps, characters, weapons, and game
modes, and delivers updated versions of the classic CS content (de_dust2, etc.).");
    }
    QPixmap pix("./kadrovik/app/resources/logo.png");
    int w = ui->logo->width();
    int h = ui->logo->height();
    ui->logo->setPixmap(pix.scaled(w, h, Qt::KeepAspectRatio));

    QPixmap pix1("./kadrovik/app/resources/download.png");
    int w1 = ui->download_1->width();
    int h1 = ui->download_1->height();
    ui->download_1->setPixmap(pix1.scaled(w1, h1, Qt::KeepAspectRatio));

    setWindowIcon(QIcon("./kadrovik/app/resources/logo.png"));

    ui->tabWidget->setTabVisible(0, 1);
    ui->tabWidget->setTabVisible(1, 0);
    ui->tabWidget->setTabVisible(2, 0);
    ui->tabWidget->setTabVisible(3, 0);
    ui->tabWidget->setTabVisible(4, 0);
    resize(600, 500);

```

```

        m_popupButtonGroup = new QButtonGroup(new QMenu(this));
        m_popupButtonGroup->addButton(ui->a);
        m_popupButtonGroup->addButton(ui->b);
        m_popupButtonGroup->addButton(ui->c);
        m_popupButtonGroup->addButton(ui->d);
        m_popupButtonGroup->setId(ui->a, 0);
        m_popupButtonGroup->setId(ui->b, 1);
        m_popupButtonGroup->setId(ui->c, 2);
        m_popupButtonGroup->setId(ui->d, 3);
    }

MainWindow::~MainWindow()
{
    m_db.close();
    delete ui;
}

void MainWindow::creationOfTables() {
    QSqlQuery query;
    query.exec("CREATE TABLE personal_info (pId INTEGER PRIMARY KEY AUTOINCREMENT, Login
TEXT UNIQUE, Pass TEXT, adminRight BOOLEAN);");
    query.exec("CREATE TABLE kandidat_info (kId INTEGER PRIMARY KEY AUTOINCREMENT, Name
TEXT, Surname TEXT, MiddleName TEXT, Age INTEGER, PhoneNumber INTEGER, Email TEXT, Posada TEXT,
Stazh INTEGER, Description TEXT, Mark_1 INTEGER, Mark_2 INTEGER, Mark_3 INTEGER, filepath TEXT,
accept INTEGER);");
    query.exec("CREATE TABLE worker_info (kId INTEGER PRIMARY KEY AUTOINCREMENT, Name
TEXT, Surname TEXT, MiddleName TEXT, Otdel INTEGER);");
    query.exec("CREATE TABLE test_it_otdel (id INTEGER PRIMARY KEY AUTOINCREMENT,
Question TEXT, A TEXT, B TEXT, C TEXT, D TEXT, rightAnswer INTEGER);");
    query.exec("CREATE TABLE test_fin_otdel (id INTEGER PRIMARY KEY AUTOINCREMENT,
Question TEXT, A TEXT, B TEXT, C TEXT, D TEXT, rightAnswer INTEGER);");
    query.exec("CREATE TABLE test_antikor_otdel (id INTEGER PRIMARY KEY AUTOINCREMENT,
Question TEXT, A TEXT, B TEXT, C TEXT, D TEXT, rightAnswer INTEGER);");
}

//void MainWindow::insertIntoGameInfo(const QString& title, const QString& author, const
QString& genre, int year, int rating, int price, int placeOnDevice, const QString& desc) {
    //    QSqlQuery query;
    //    query.prepare("INSERT INTO game_info (Title, Author, Genre, DateOfBirth, Rating,
Price, PlaceOnDevice, Description)"
    //
    //        "VALUES (?, ?, ?, ?, ?, ?, ?, ?)");
    //    query.bindValue(0, title);
    //    query.bindValue(1, author);
    //    query.bindValue(2, genre);
    //    query.bindValue(3, year);

```

```

//    query.bindValue(4, rating);
//    query.bindValue(5, price);
//    query.bindValue(6, placeOnDevice);
//    query.bindValue(7, desc);
//    query.exec();
//}

void MainWindow::dataRecovery() {
    QSqlQueryModel model;
    model.setQuery("SELECT kId, Name, Surname, MiddleName, accept FROM kandidat_info");
    for (int i = 0; i < model.rowCount(); ++i) {
        int kId = model.record(i).value("kId").toInt();
        int accept = model.record(i).value("accept").toInt();
        QString Name = model.record(i).value("Name").toString();
        QString Surname = model.record(i).value("Surname").toString();
        QString MiddleName = model.record(i).value("MiddleName").toString();
        if(accept == 0)
        {
            QListWidgetItem *new_item = new QListWidgetItem;
            new_item->setText(Name + " " + Surname + " " + MiddleName);
            new_item->setWhatsThis(QString::number(kId));
            ui->myCandidates->addItem(new_item);
        }
    }
}

void MainWindow::kandidatProfileFilling(const QString& kId) {
    QSqlQueryModel model;
    QString buff = "SELECT Name, Surname, MiddleName, Age, PhoneNumber, Email, Posada,
Stazh, Mark_1, Mark_2, Mark_3, filepath FROM kandidat_info WHERE kId = '";
    buff += kId;
    buff += "'";

    model.setQuery(buff);

    QString name = model.record(model.rowCount() - 1).value("Name").toString();
    QString surname = model.record(model.rowCount() - 1).value("Surname").toString();
    QString      middleName      =      model.record(model.rowCount() -
1).value("MiddleName").toString();
    ui->pib->setText(name + " " + surname + " " + middleName);

    QString email = model.record(model.rowCount() - 1).value("Email").toString();
    QString phone = model.record(model.rowCount() - 1).value("PhoneNumber").toString();
    QString posada = model.record(model.rowCount() - 1).value("Posada").toString();
    int stazh = model.record(model.rowCount() - 1).value("Stazh").toInt();

```

```

ui->email_2->setText(email);
ui->phone_2->setText(phone);
ui->stazh_2->setText(QString::number(stazh) + " років");
ui->posada_2->setText(posada);

m_filepath = model.record(model.rowCount() - 1).value("filepath").toString();

int mark1 = model.record(model.rowCount() - 1).value("Mark_1").toInt();
int mark2 = model.record(model.rowCount() - 1).value("Mark_2").toInt();
int mark3 = model.record(model.rowCount() - 1).value("Mark_3").toInt();

ui->kandidatMark->addItem("Загальна = " + QString::number(mark1 + mark2 + mark3) + "
балів");
ui->kandidatMark->addItem("Тест 1 = " + QString::number(mark1) + " балів");
ui->kandidatMark->addItem("Тест 2 = " + QString::number(mark2) + " балів");
ui->kandidatMark->addItem("Тест 3 = " + QString::number(mark3) + " балів");
ui->kandidatMark->setCurrentIndex(0);
}

void MainWindow::on_signInButton_clicked() {
//    //sign_in
    QString inputLogin = ui->loginEdit->text();
    QString inputPass = ui->passEdit->text();

    QSqlQueryModel model;
    QString buff = "SELECT Pass, adminRight FROM personal_info WHERE Login = '";
    buff += inputLogin;
    buff += "'";
    model.setQuery(buff);
    QByteArray pass;
    pass.append(inputPass.toStdString().c_str());
    bool admin = model.record(model.rowCount() - 1).value("adminRight").toBool();
    if (model.record(model.rowCount() - 1).value("Pass").toString() == pass.toBase64()
        && !model.record(model.rowCount() - 1).isNull("Pass")) {
        m_login = inputLogin;
        ui->loginEdit->setText("");
        ui->passEdit->setText("");
        ui->tabWidget->setTabVisible(0, 0);
        resize(1145, 610);
        if(admin)
        {
            dataRecovery();

```

```

        ui->tabWidget->setTabVisible(1, 1);
    }
    else
    {
        ui->tabWidget->setTabVisible(3, 1);
    }
}
else {
    ui->statusBar->showMessage("Не правильний логін чи пароль.", 5000);
}
}

void MainWindow::on_signUpButton_clicked() {
    //sign up
    QString inputLogin = ui->loginSignUp->text();
    QString inputPass = ui->passSignUp->text();
    QString inputPass_2 = ui->passSignUp_1->text();

    QByteArray ba;
    ba.append(inputPass.toStdString().c_str());
    // qDebug() << "create = " << ba.toBase64();

    QSqlQueryModel model;
    QString buff = "SELECT Login FROM personal_info WHERE Login = '";
    buff += inputLogin;
    buff += "'";
    model.setQuery(buff);
    // qDebug() << "Login From DB = " << model.record(model.rowCount() -
1).value("Pass").toString();
    if (inputPass != inputPass_2)
        ui->statusBar->showMessage("Паролі не співпадають.", 5000);
    else if (model.record(model.rowCount() - 1).isNull("Login")) {
        QSqlQuery query;
        query.prepare("INSERT INTO personal_info (Login, Pass)"
            "VALUES (?, ?)");
        query.bindValue(0, inputLogin);
        query.bindValue(1, ba.toBase64());
        query.exec();
        ui->statusBar->showMessage("Користувач успішно зареєстрований!", 5000);
    }
    else
        ui->statusBar->showMessage("Такий логін уже існує.", 5000);
    ui->loginSignUp->setText("");
    ui->passSignUp->setText("");
    ui->passSignUp_1->setText("");
}

```

```

}

void MainWindow::on_myCandidates_itemDoubleClicked(QListWidgetItem *item)
{
    active_kId = item->whatsThis();

    QSqlQueryModel model;
    QString buff = "SELECT Stazh, Posada, Description FROM kandidat_info WHERE kId = '";
    buff += active_kId;
    buff += "'";
    model.setQuery(buff);
    int stazh = model.record(model.rowCount() - 1).value("Stazh").toInt();
    QString posada = model.record(model.rowCount() - 1).value("Posada").toString();
    QString description = model.record(model.rowCount() -
1).value("Description").toString();

    ui->mini_pib->setText(item->text());
    ui->mini_posada->setText(posada + ", " + QString::number(stazh) + " поки");
    ui->mini_desc->setText(description);
}

void MainWindow::on_check_clicked()
{
    if(active_kId != "-1")
    {
        kandidatProfileFilling(active_kId);
        ui->tabWidget->setTabVisible(1, 0);
        ui->tabWidget->setTabVisible(2, 1);
    }
}

void MainWindow::on_accept_clicked()
{
    //prinyato
    QSqlQuery query;
    query.prepare("UPDATE kandidat_info SET accept = :accept WHERE kId = '" + active_kId
+ "';");
    query.bindValue(":accept", 1);
    if (query.exec()) {
        qDebug() << "success";
    }
    else {
        qDebug() << "fail";
    }
}

```

```

    }

    ui->myCandidates->clear();

    QSqlQueryModel model;
    model.setQuery("SELECT kId, Name, Surname, MiddleName, accept FROM kandidat_info");
    for (int i = 0; i < model.rowCount(); ++i) {
        int kId = model.record(i).value("kId").toInt();
        int accept = model.record(i).value("accept").toInt();
        QString Name = model.record(i).value("Name").toString();
        QString Surname = model.record(i).value("Surname").toString();
        QString MiddleName = model.record(i).value("MiddleName").toString();
        if(accept == 0)
        {
            QListWidgetItem *new_item = new QListWidgetItem;
            new_item->setText(Name + " " + Surname + " " + MiddleName);
            new_item->setWhatsThis(QString::number(kId));
            ui->myCandidates->addItem(new_item);
        }
    }

    ui->mini_pib->setText("");
    ui->mini_posada->setText("");
    ui->mini_desc->setText("");

    ui->tabWidget->setTabVisible(1, 1);
    ui->tabWidget->setTabVisible(2, 0);
    active_kId = "-1";
}

void MainWindow::on_refused_clicked()
{
    //vidhyleno

    QSqlQuery query;
    query.prepare("UPDATE kandidat_info SET accept = :accept WHERE kId = '" + active_kId
+ "';");
    query.bindValue(":accept", 2);
    if (query.exec()) {
        qDebug() << "success";
    }
    else {
        qDebug() << "fail";
    }
}

```

```

ui->myCandidates->clear();

 QSqlQueryModel model;
 model.setQuery("SELECT kId, Name, Surname, MiddleName, accept FROM kandidat_info");
 for (int i = 0; i < model.rowCount(); ++i) {
     int kId = model.record(i).value("kId").toInt();
     int accept = model.record(i).value("accept").toInt();
     QString Name = model.record(i).value("Name").toString();
     QString Surname = model.record(i).value("Surname").toString();
     QString MiddleName = model.record(i).value("MiddleName").toString();
     if(accept == 0)
     {
         QListWidgetItem *new_item = new QListWidgetItem;
         new_item->setText(Name + " " + Surname + " " + MiddleName);
         new_item->setWhatsThis(QString::number(kId));
         ui->myCandidates->addItem(new_item);
     }
 }

 ui->mini_pib->setText("");
 ui->mini_posada->setText("");
 ui->mini_desc->setText("");

 ui->tabWidget->setTabVisible(1, 1);
 ui->tabWidget->setTabVisible(2, 0);
 active_kId = "-1";
}

//download cv
void MainWindow::on_pushButton_clicked()
{
    QString fileName = QFileDialog::getSaveFileName(this, tr("Save File"), m_filepath,
tr("*.pdf"), nullptr, QFileDialog::ShowDirsOnly);
    QFile file(fileName);
    if (!file.open(QIODevice::WriteOnly | QIODevice::Text)){
        qDebug() << file.errorString();
        return;
    }
    //    QTextStream out(&file);
    //    out << "The magic number is: " << 49 << "\n";
    ui->statusBar->showMessage("Резюме збережено за шляхом = " + fileName, 4000);
}

```

```

//upload cv
void MainWindow::on_pushButton_2_clicked()
{
    QString filename = QFileDialog::getOpenFileName(this, tr("Open File"), "/",
tr("*.pdf"));
    qDebug() << "filename = " << filename;
    active_filename = filename;
}

void MainWindow::on_nextQuestion_clicked()
{
    QVector < QPair < QVector < QString > , uint16_t >> test;

    if (m_test == 1) {
        test = test1;
    }
    else if (m_test == 2) {
        test = test2;
    }
    else if (m_test == 3) {
        test = test3;
    }

    if(!m_test_started) {
        m_test_started = true;
        ui->nextQuestion->setText("Наступне питання");
    }
    else {
        qDebug() << "m_test = " << m_test;
        qDebug() << "m_iter = " << m_iter;
        if (m_right_answer == m_popupButtonGroup->checkedId() + 1) {
            m_mark++;
            qDebug() << "m_mark = " << m_mark;
        }

        ui->label_3->setText("Питання №" + QString::number(m_iter + 1));
        ui->test_1->setText("Тест №" + QString::number(m_test));
        ui->question->setText(test.at(m_iter).first.at(0));
        ui->a->setText(test.at(m_iter).first.at(1));
        ui->b->setText(test.at(m_iter).first.at(2));
        ui->c->setText(test.at(m_iter).first.at(3));
        ui->d->setText(test.at(m_iter).first.at(4));
        m_right_answer = test.at(m_iter).second;
    }
}

```

```

        if (m_test == 1) {
            m_mark1 = m_mark;
        }
        else if (m_test == 2) {
            m_mark2 = m_mark;
        }
        else if (m_test == 3) {
            m_mark3 = m_mark;
        }

        m_iter++;
        if (m_iter >= 3) {
            m_test++;
            m_iter = 0;
            m_mark = 0;
            if (m_test > 3)
                ui->nextQuestion->setDisabled(true);
        }
    }
}

void MainWindow::on_endTest_clicked()
{
    ui->statusBar->showMessage("Відповідь збережена", 2000);
    m_test_finished = true;
    ui->wantJob->setText("Подать заявку");

    QMessageBox msgBox;
    msgBox.setText("Тест1 = " + QString::number(m_mark1) + "\nТест2 = " +
        QString::number(m_mark2) + "\nТест3 = " + QString::number(m_mark3));
    msgBox.setWindowTitle("Результати");
    msgBox.exec();
    ui->tabWidget->setTabVisible(4, 0);
    resize(1145, 610);
    ui->tabWidget->setTabVisible(3, 1);
}

void MainWindow::on_wantJob_clicked()
{
    if(!m_test_finished)
    {
        QSqlQueryModel model;
    }
}

```

```

model.setQuery("SELECT A, B, C, D, rightAnswer FROM test_antikor_otdel");
for (int i = 0; i < model.rowCount(); ++i) {
    QPair<QVector<QString>, uint16_t> pair;
    pair.second = model.record(i).value("rightAnswer").toInt();
    pair.first.push_back(model.record(i).value("Question").toString());
    pair.first.push_back(model.record(i).value("A").toString());
    pair.first.push_back(model.record(i).value("B").toString());
    pair.first.push_back(model.record(i).value("C").toString());
    pair.first.push_back(model.record(i).value("D").toString());
    test1.push_back(pair);
}
model.clear();

model.setQuery("SELECT A, B, C, D, rightAnswer FROM test_fin_otdel");
for (int i = 0; i < model.rowCount(); ++i) {
    QPair<QVector<QString>, uint16_t> pair;
    pair.second = model.record(i).value("rightAnswer").toInt();
    pair.first.push_back(model.record(i).value("Question").toString());
    pair.first.push_back(model.record(i).value("A").toString());
    pair.first.push_back(model.record(i).value("B").toString());
    pair.first.push_back(model.record(i).value("C").toString());
    pair.first.push_back(model.record(i).value("D").toString());
    test2.push_back(pair);
}
model.clear();

model.setQuery("SELECT A, B, C, D, rightAnswer FROM test_it_otdel");
for (int i = 0; i < model.rowCount(); ++i) {
    QPair<QVector<QString>, uint16_t> pair;
    pair.second = model.record(i).value("rightAnswer").toInt();
    pair.first.push_back(model.record(i).value("Question").toString());
    pair.first.push_back(model.record(i).value("A").toString());
    pair.first.push_back(model.record(i).value("B").toString());
    pair.first.push_back(model.record(i).value("C").toString());
    pair.first.push_back(model.record(i).value("D").toString());
    test3.push_back(pair);
}

ui->label_3->setText("Питання №" + QString::number(m_iter + 1));
ui->test_1->setText("Тест №" + QString::number(m_test));
ui->question->setText(test1.at(m_iter).first.at(0));
ui->a->setText(test1.at(m_iter).first.at(1));
ui->b->setText(test1.at(m_iter).first.at(2));
ui->c->setText(test1.at(m_iter).first.at(3));
ui->d->setText(test1.at(m_iter).first.at(4));

```

```

        m_right_answer = test1.at(m_iter).second;
        m_mark1 = 0;
        m_mark2 = 0;
        m_mark3 = 0;

        ui->tabWidget->setTabVisible(3, 0);
        resize(650, 550);
        ui->tabWidget->setTabVisible(4, 1);
    }
    else
    {
        QString name = ui->name->toPlainText();
        QString middle = ui->middle->toPlainText();
        QString surname = ui->surname_2->toPlainText();
        QString phone = ui->phone->toPlainText();
        QString email = ui->email->toPlainText();
        QString posada = ui->posada->toPlainText();
        QString desc = ui->desc->toPlainText();
        int age = ui->age->value();
        int stazh = ui->stazh->value();

        if(name.isEmpty() || middle.isEmpty() || surname.isEmpty() || phone.isEmpty() ||
email.isEmpty() || posada.isEmpty() || age == 0 || stazh == 0)
        {
            QMessageBox msgBox;
            msgBox.setText("Заповніть всі поля.");
            msgBox.setWindowTitle("УВАГА!");
            msgBox.exec();
        }
        else
        {
            QSqlQuery query;
            query.prepare("INSERT INTO kandidat_info (Name, Surname, MiddleName, Age,
PhoneNumber, Email, Posada, Stazh, Description, Mark_1, Mark_2, Mark_3, filepath, accept)"
                        "VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");
            query.bindValue(0, name);
            query.bindValue(1, middle);
            query.bindValue(2, surname);
            query.bindValue(3, age);
            query.bindValue(4, phone);
            query.bindValue(5, email);
            query.bindValue(6, posada);
            query.bindValue(7, stazh);
            query.bindValue(8, desc);
            query.bindValue(9, m_mark1);

```

```

        query.bindValue(10, m_mark2);
        query.bindValue(11, m_mark3);
        query.bindValue(12, active_filename);
        query.bindValue(13, 0);
        query.exec();

        m_test_finished = false;
        m_test_started = false;
        m_iter = 0;
        m_test = 1;
        m_mark = 0;
        m_right_answer = -1;
        m_mark1 = -1;
        m_mark2 = -1;
        m_mark3 = -1;
        active_kId = "-1";
        active_filename = "-1";

        QMessageBox msgBox;
        msgBox.setText("Заявка успішно подана!");
        msgBox.setWindowTitle("-----");
        msgBox.exec();
        close();
    }
}
}

```

```
#pragma once
```

```

#include <QMainWindow>
#include <QMenuBar>
#include <QFileDialog>
#include <QTextStream>
#include <QFileSystemModel>
#include <QApplication>
#include <QMessageBox>
#include <QWidget>
#include <QStringList>
#include <QDebug>
#include <QFileInfo>
#include <QPlainTextEdit>
#include <QPalette>
#include <QLabel>

```

```

#include <QtSql>
#include <QAbstractItemView>
#include <QSqlTableModel>
#include <QListWidgetItem>
#include <QButtonGroup>

#include <iostream>
#include <fstream>
#include <string>
#include <algorithm>
#include <vector>
#include <QVector>

namespace Ui {
    class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

    //databases part
    void creationOfTables();

    //    void insertIntoGameInfo(const QString& title, const QString& author, const QString&
genre, int year, int rating, int price, int placeOnDevice, const QString& desc);

    void dataRecovery();

    void kandidatProfileFilling(const QString& kId);

private slots:
    //    void on_find_clicked();

    void on_signInButton_clicked();

    void on_signUpButton_clicked();

    //    void on_saveProfile_clicked();

    void on_myCandidates_itemDoubleClicked(QListWidgetItem *item);

```

```

void on_check_clicked();

void on_accept_clicked();

void on_refused_clicked();

void on_pushButton_clicked();

void on_pushButton_2_clicked();

void on_nextQuestion_clicked();

void on_endTest_clicked();

void on_wantJob_clicked();

private:
    bool m_test_finished = false;
    bool m_test_started = false;
    int m_iter = 0;
    int m_test = 1;
    int m_mark = 0;
    int m_right_answer;
    int m_mark1 = -1;
    int m_mark2 = -1;
    int m_mark3 = -1;
    QString active_kId = "-1";
    QString active_filename = "-1";
    QVector<QPair<QVector<QString>, uint16_t>> test1;
    QVector<QPair<QVector<QString>, uint16_t>> test2;
    QVector<QPair<QVector<QString>, uint16_t>> test3;

    QString m_filepath;
    QString m_login;
    QButtonGroup *m_popupButtonGroup;
    Ui::MainWindow *ui;
    QSqlDatabase m_db;
};

```

Матеріали апробації основних тез проведеного дослідження

Збірник праць учасників Всеукраїнської науково-практичної конференції
«Інформаційні технології та моделювання систем»

гарантій інформаційно-телекомунікаційних систем. Застосунок дозволяє експертові витратити менше часу для оцінки інформаційно-телекомунікаційної системи щодо захисту інформації, яка циркулює у системі. Допомогає виявити застосований функціональний профіль захищеності та рівень гарантій оброблюваної інформації від несанкціонованого доступу. У випадку виявлення профілю захищеності, програма допомагає перевірити його на співпадіння із стандартним, такі ж дії виконуються із рівнем гарантій. [2,3].

Розроблений програмний додаток надає можливість автоматизовано визначати функціональні профілі захищеності та рівні гарантій оброблюваної інформації від несанкціонованого доступу та підтвердити стандартний або визначити інший рівень гарантій. В умовах війни даний додаток дозволить пришвидшити визначення вразливих місць інформаційно-телекомунікаційних систем та оперативно їх усувати адміністраторам безпеки.

Список використаних джерел

1. Юдін О. К. Теоретичні основи визначення стандартних функціональних профілів захищеності автоматизованої системи від несанкціонованого доступу / О. К. Юдін, С. С. Бучик, С. В. Мельник // Наукоємні технології. – 2016. – № 2 (30). – С.195 – 205, doi.org/10.18372/2310-5461.30.10564.
2. Бучик С. С. Реалізація програмного забезпечення визначення функціональних профілів та рівня гарантій автоматизованих систем від несанкціонованого доступу / С. С. Бучик, Р. В. Нетребко // Наукоємні технології № 4 (36), 2017. – С. 309-315. DOI: 10.18372/2310-5461.36.12228.
3. Комп'ютерна програма. Інформаційна система визначення функціонального профілю захищеності та рівня гарантій автоматизованої системи від несанкціонованого доступу (ОФПАС 2.0) / С. С. Бучик, Р. В. Нетребко (Україна). – №75050. – заявл. 23.10.2017. – С. 142 – 143.

Верба А. Ю., студент
Київський національний економічний університет
імені Вадима Гетьмана
Науковий керівник – к. е. н., доцент Зінов'єва І. С.

ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ АДАПТАЦІЄЮ
НОВИХ КАДРІВ ПІДПРИЄМСТВА

Сьогодні в Україні створення систем і зміна методів управління пов'язані з постійним науковим пошуком оптимальних рішень складних економічних і соціально-політичних проблем, докорінним оновленням усіх рівнів управління, стилів правління. У сфері кадрового управління та корпоративного менеджменту важливою задачею є організація ефективного процесу адаптації нових працівників до існуючих бізнес-процесів в середині організації. Від того, наскільки якісним та продуктивним будуть реалізовані ці процеси залежить, надалі, і продуктивність праці майбутніх працівників та рівень кадрового менеджменту на підприємстві в цілому.

Термін «адаптація» походить від лат. адаптувати – пристосовувати. Адаптація до праці – це соціальний процес розвитку особистості на основі нової робочої ситуації. Вступаючи на роботу, людина активно бере участь у системі професійних і соціально-психологічних відносин певної трудової організації, засвоює нові соціальні ролі, цінності, норми, узгоджує свою особисту позицію з цілями та завданнями організації (трудового колективу), таким чином подання своїх поведінкових приписів [1]. Кваліфікована команда є одним із ключових

факторів успіху розвитку будь-якої компанії. Захист людських ресурсів стає все більш актуальним, особливо в період активного зростання та розвитку компанії.

Проведені дослідження найбільш розповсюджених інформаційних систем з управління та роботи з персоналом дали підстави стверджувати, що ці системи працюють на основі використання елементів традиційних ІТ [2].

Існуюче програмне забезпечення у сфері управління персоналом можна умовно поділити на дві групи:

- спеціалізоване програмне забезпечення, яке покликано розв'язувати управлінські задачі: підбір персоналу, оцінка настрою, навчання, тестування тощо (наприклад, HRIS, Achart People Management та ін);
- окремий модуль / підсистема в рамках глобальної корпоративної ERP (наприклад, ERP ISPro, Oracle E-Business Suite, MySAP Business Suite та ін).

Питання підтримки процесів адаптації персоналу в розглянутих системах та програмних продуктах реалізовано досить «поверхнево», немає реалізованої логіки бізнес-процесу управління адаптацією нових кадрів підприємства.

З метою заповнення даної прогалини було вирішено спроектувати та розробити прототип інформаційної системи, яка б забезпечувала управління адаптацією нових кадрів на підприємстві.

На рис. 1 сформовано ключові бізнес-вимоги до проектуваної системи, а на рис. 2 – функціональні вимоги, до яких віднесено демонстрацію результатів навчання працівників, управління правами доступу для користувачів, можливість завантаження та вивантаження даних із системи та їх оновлення, ведення аналітики за результатами адаптаційного періоду.

Для розробки Backend частини інформаційної системи було використано: мову програмування C++, фреймворк – QT Framework, STL, UNIX Core.

Для розробки Frontend частини: HTML5, CSS3, QML.

Дані, якими оперуватиме інформаційна система зберігаються в базі даних, для її розробки було використано SQLite3.

Загальні інструменти, що були використані для розробки інформаційної системи: IDE CLion, IDE VSCode, Windows 10, CMake version 3.17.2, GDB, QT Creator, GitHub.

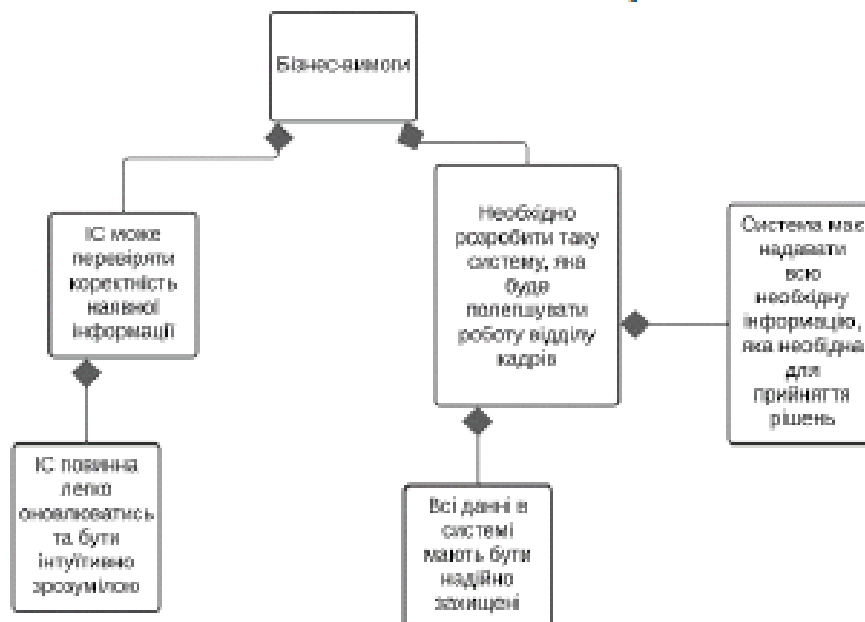


Рис. 1. Бізнес-вимоги до інформаційної системи управління адаптацією нових кадрів підприємства

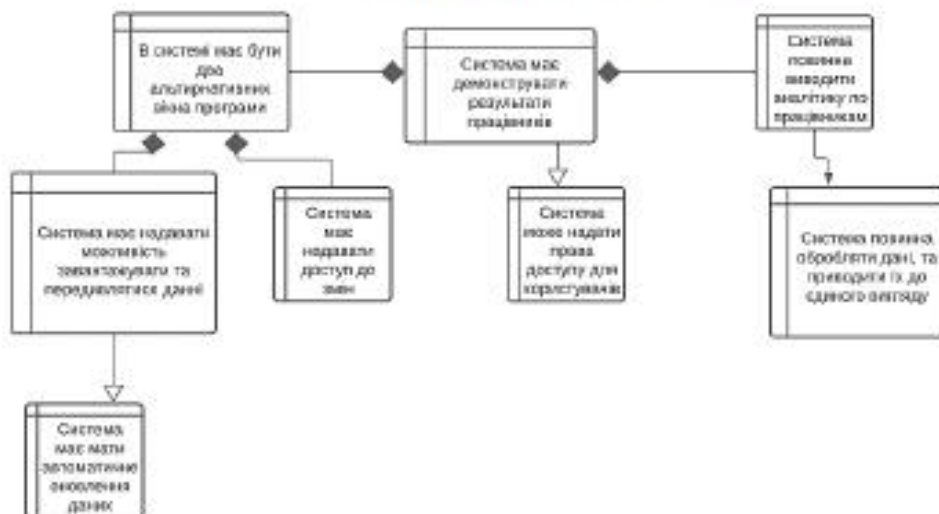


Рис. 2. Функціональні вимоги до інформаційної системи управління адаптацією нових кадрів підприємства

У результаті виконання проєктних робіт, було створено робочий прототип інформаційної системи управління адаптацією нових кадрів підприємства, який містить функціонал для:

- стажера – нового працівника підприємства, який проходить адаптацію та навчання на робочому місці (рис. 3);
- менеджера, який виконує функції куратора стажера і контролює результати адаптації нового працівника на підприємстві;
- адміністратора системи.



Рис. 3. Головне вікно користувача (стажер)

У ході дослідження реалізовано роботи повного циклу розробки інформаційної системи автоматизації управління адаптацією нових кадрів підприємства, від моделювання та постановки задачі до фактичної реалізації програмного продукту.

Порівнюючи розроблену систему з аналогічними на ринку, можна виділити декілька переваг які ми спроектували: по-перше, це – інтуїтивний та мінімалістичний інтерфейс системи, по-друге – реалізований функціонал є, порівняно новим, але затребуваним на ринку корпоративного управління, по-третє – розроблена система має мінімальні вимоги до апаратного та програмного забезпечення.

Список використаних джерел

1. Паратнова О. Успішна адаптація: критерії та «спідводні камені» / О. Паратнова // Управління людським потенціалом. – 2017. – №3. – С. 198-203.
2. Понаморенко В. С. Проектування інформаційних систем / В. С. Понаморенко., 2017. – 432 с.

Балейко А. С., студент
Дніпровський національний університет імені Олеся Гончара
Науковий керівник – к. ф.-м. н. Микальчук Г. Й

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ОПТИМІЗАЦІЇ РОБОТИ СЛУЖБИ ДОСТАВКИ

Важко переоцінити важливість роботи служб доставки у сучасному світі. Оптимально налагоджена логістика дозволяє значно покращити якість сервісу доставки і зекономити велику кількість грошей на транспортні витрати.

Для оптимізації роботи служби доставки було створено програмне забезпечення яке дозволяє на основі вхідних даних про клієнтів побудувати найбільш оптимальну мережу маршрутів. Вхідною інформацією є список запитів на доставку. Кожен запит містить адресу клієнта, інформацію про товар, часовий проміжок у який клієнт повинен бути відвіданий. Кількість транспортних засобів обмежена. Кожен транспортний засіб має максимальну вантажопідйомність та може працювати у заданий проміжок часу. Кожен маршрут обов'язково повинен починатись і закінчуватись у депо.

Побудова оптимальної мережі маршрутів відбувається у два етапи. На першому етапі будується початкове рішення. На другому етапі початкове рішення покращується з метою побудови більш оптимальної мережі маршрутів. Для побудови початкового рішення користувач може використати один з таких запропонованих алгоритмів:

- алгоритм найближчого сусіда;
- алгоритм найбільш віддаленого сусіда;
- модифікований алгоритм Кларка-Райта.

Для покращення початкового рішення використовуються алгоритми:

- 2-opt;
- модифікований алгоритм Swap.

Огляд використаних евристичних алгоритмів наведений у роботах [1, с. 6], [2, с. 88].

Користувач самостійно обирає алгоритми та параметри до них для обробки власних даних. Це дозволяє віднайти найбільш ефективну комбінацію алгоритмів для обраного набору даних. Алгоритми покращення стають доступними для виконання тільки після створення початкового рішення. Також, після побудови початкового рішення, користувачу стає доступною інформація про кожен маршрут (довжина, вартість, кількість відвіданих клієнтів, сумарний об'єм товару). Отримана мережа маршрутів зображується на мапі. Кожен маршрут виділяється одним з трьох кольорів:

- зелений – враховано всі обмеження задачі;