Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВИЗНАЧЕННЯ РІВНЯ ГОТОВНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДО АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ
(тема)

Виконав:
студент 4 курсу, групи ІТІНФ-17-1

Полубєхін А.А.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Творошенко І.С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Кобилін О.А.
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2021 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Полубєхіну Антону Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Моделювання інформаційної системи визначення рівня готовності програмного забезпечення до автоматизованого тестування

затверджена наказом університету від 20 травня 2021 року № 663Ст

2. Термін подання студентом роботи до екзаменаційної комісії 26 травня 2021 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, існуючі експертні системи, теоретичні знання про бази даних, теоретичні та практичні знання методів системного аналізу, сервіс Visual-Paradigm, сервіс Typeform.

4. Перелік питань, що потрібно опрацювати в роботі.

1. Аналіз існуючих інформаційних систем врахування рівня розроблення програмного забезпечення в Україні та за кордоном.

2. Розроблення підходу до моделювання інформаційної системи.

3. Розроблення моделі інформаційної системи визначення рівня готовності програмного забезпечення до автоматизованого тестування.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність роботи, постановка задачі, етапи моделювання інформаційної системи визначення рівня готовності програмного забезпечення до автоматизованого тестування.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	12.04.2021	
2	Аналіз завдання, підбір літератури	13.04.21-15.04.21	
3	Аналіз літератури з досліджуваної проблеми	16.04.21-17.04.21	
4	Аналіз методів системного аналізу	18.04.21-24.04.21	
5	Розробка структури	25.04.21-14.05.21	
6	Програмна реалізація	15.05.21-23.05.21	
7	Оформлення пояснювальної записки	24.05.21-26.05.21	
8	Перевірка на плагіат	27.05.21	
9	Рецензування	28.05.21	
10	Підготовка презентації та доповіді	29.05.21-30.05.21	
11	Занесення роботи в електронний архів	02.06.21	
12	Попередній захист кваліфікаційної роботи	02.06.21	

Дата видачі завдання 12 квітня 2021 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Творошенко І.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 70 с., 7 табл., 47 рис., 1 дод., 30 джерел.

МОДЕЛЮВАННЯ, ІНФОРМАЦІЙНА СИСТЕМА, ВИЗНАЧЕННЯ РІВНЯ ГОТОВНОСТІ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, АВТОМАТИЗОВАНЕ ТЕСТУВАННЯ.

Об’єктом роботи є автоматизація тестування.

Метою роботи є розробка структури опитувальної системи і наповнення її бази знань запитаннями, відповіді на які дозволять визначити рівень готовності проєкту, а також отримати базові вказівки з впровадження змін щодо збільшення ступеню підготовки проєкту до введення автоматизації.

Проведено дослідження інформаційних систем врахування орівня розроблення програмного забезпечення в Україні та за кордоном. Проведений аналіз особливостей баз даних інформаційних систем. Досліджено і застосовано методи системного аналізу.

У результаті роботи розроблена модель інформаційної системи визначення рівня готовності програмного забезпечення до автоматизованого тестування.

Результати роботи апробовано у вигляді тез доповіді під час Міжнародного молодіжного форуму «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У XXI СТОЛІТТІ».

MODELING, INFORMATION SYSTEM, DETERMINING THE LEVEL OF READINESS, SOFTWARE, AUTOMATED TESTING.

The object of the research is automation testing.

The research aims to create the structure of the interrogation system and fulfill its knowledge database with questions, passing which helps determine the level of readiness of the project and gives basic instructions of increasing willingness in order to introduce the automation testing.

A study of information systems that take into account the level of the development of the software in Ukraine and abroad. The analysis of features of databases of information systems is carried out. Methods of system analysis are investigated and applied.

As a result the model of the information system for determining the level of readiness of the software for automating testing.

The results of the work were tested in the form of abstracts of reports during the International Youth Forum "RADIO ELECTRONICS AND YOUTH IN THE XXI CENTURY".

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 6
Вступ 7
1 Аналіз існуючих інформаційних систем врахування рівня розроблення програмного забезпечення в Україні та за кордоном..... 8
1.1 Сучасний стан розвитку інформаційних систем врахування рівня розроблення програмного забезпечення в Україні та за кордоном.....8
1.2 Аналіз літературних джерел щодо існуючих підходів моделювання та проєктування інформаційних систем.....14
1.3 Постановка задачі.....16
2 Розроблення підходу до моделювання інформаційної системи 18
2.1 Особливості бази даних інформаційної системи врахування рівня розроблення програмного забезпечення 18
2.2 Застосування методів системного аналізу предметної області.....24
2.3 Моделювання структури та наповнення інформаційної системи....33
3 Розроблення моделі інформаційної системи визначення рівня готовності програмного забезпечення до автоматизованого тестування 42
3.1 Вибір інструментальних засобів для створення моделі 42
3.2 Етапи створення моделі інформаційної системи 45
3.3 Тестування розробленої інформаційної системи та аналіз результатів 57
3.4 Перспективи подальшої роботи..... 62
Висновки..... 63
Перелік джерел посилання 64

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ШІ – штучний інтелект

ПО – програмне забезпечення

IDEF1 – методологія для розв’язання задча моделювання

БД – база даних

СУБД – система управління базою даних

СУРБД – система управління розподільною базою даних

ООП – обьектно-орієнтований підхід

НФ – нормальна форма

СА – системний аналіз

ВСТУП

Штучний інтелект розділяється за трьома основними напрямкам: машинне мислення або когнітивні наукові системи, машине навчання, робототехніка. ШІ почав розвиватись швидкими темпами у сучасному світі. Все більше компаній починають займатися розробкою штучного інтелекту, який можно буде використовувати у різних сферах життя людини.

Одним з підрозділів когнетивних наукових систем є інтелектуальні інформаційні системи. Одним з підрозділів інтелектуально інформаційної системи є експертні системи.

Експертні системи – це системи, які мають знання експертів в деякій галузі і можуть робити логічні висновки, вирішуючи цим специфічні завдання. Експертні системи свого часу знайшли застосування у різних сферах вони стали прешою стадією використання штучного інтелекту. Області застосування – експертних систем сфера медицини, військової справи, IT-сфері. Експертні системи не могли давти більше можливостей і тому ШІ інтелект продовжив свій розвиток, навідміну від експертних систем, які поступово почали виходити з використання у IT-сфері.

Зараз у IT-компаніях, на великих або поступово зростаючих проектах зростає необхідність автоматизації тестування. На таких проектах зростає час на тестування, а через це – потрібно наймати нових тестувальників, що викликає більші витрати або зменшується якість тестування, що приводить до втрати клієнтів через низьку якість ПО.

Використання експертних системи у різних галузях IT-сфери допоможе зменшити час на аналіз і дпоможе зменшити витрати на залучення нових працівників. Автоматизація тестування вимагає аналзіу проекта, перед тим як починати автоматизацію, тому що інколи проекти не готові до введення автомтаизованого тестування через брак інформації.

Актуальність роботи полягає у розробці першої експертної системи для визначення готовності проекта до автоматизації.

1 АНАЛІЗ ІСНУЮЧИХ ІНФОРМАЦІЙНИХ СИСТЕМ ВРАХУВАННЯ РІВНЯ РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В УКРАЇНІ ТА ЗА КОРДОНОМ

1.1 Сучасний стан розвитку інформаційних систем врахування рівня розроблення програмного забезпечення в Україні та за кордоном

Люди завжди хотіли створити роботів, які би могли виконувати команди і вирішувати різні завдання, працювати на небезпечних для людського життя роботах, бути замість солдат і мислити, як людина. Із поширенням штучного інтелекту ці мрії можуть стати реальністю. 2000-й рік ознаменував нову єпоху ШІ. Штучний інтелект захопив майже всі сфери життя, які хоч якимось чином відносислись до нього. Наразі ШІ використовують у спецслужбах, медицині, важкій промисловості і інших сферах. Найбільші ІТ-компанії світу почали розробку власних програм, які пов’язані зі штучним інтелектом та робототехнікою. Майже у кожній великій компанії є відділ, який спеціалізується на розробці штучного інтелекту.

У ШІ виділяють інтелектуальні інформаційні систем, які являються одним з видів автоматизованих інформаційних систем, ще одна назва – система заснована на знаннях. ІІС поділяють на два підрозділи, перший – це експертні системи, куди входять експретні системи і інтерактивні банери, другий підрозділ називається – запитально – відповідальні системи і включає в себе інтелектуальні пошукові системи і віртулаьні співбесідники.

Інтелектуальні інформаційні системи вирішують багато завдань, які включають в себе інтерпретацію даних, діагностику – співідношення об’єкта з класом об’єктів і виявлення відхилення від норми в системі, моніторинг, який допомогає безперервно інтерпритувати дані у реальному масштабі часу, проектування, прогнозування, планування, навчання і керування, а також підтримку прийняття рішень – сукупність процедур, які дають необхідну інформацію особі, яка приймає рішення.

Одним з підрозділів ІІС і штучного інтелекту (рис. 1.1), являються експертні системи. Якщо ШІ з 2000-го року продовжив розвиватись та розширювати свої межі і свій вплив на життя людей, натомість експертні системи у 1990-х роках зупинились у своєму розвитку і надалі термін «експретні системи» був в основному виключений з лексикону ІТ.

Рисунок 1.1 – Підрозділи штучного інтелекту

Експертні системи були першою успішною формою ПО для штучного інтелекту, які використовували і використовують у медичній і автомобільній діагностиках.

Експертні системи складаються з наступних компонентів: бази знань, машини логічного виведення, підсистеми користувача, а також інтерфейсу користувача. Схема експертної системи має наступний вигляд (рис. 1.2).

Рисунок 1.2 – Схема експертної системи

Процесс створення експертної системи описаний у (табл. 1.1).

Таблиця 1.1 – Процес створення експертної системи

Етапи розробки	**Опис етапу**
Ідентифікація проблеми	Виявляється проблема яку потрібно вирішити.
Витягнення знань	Проводиться аналіз предметної області.
Структуризація знань	Моделюється робота системи і структуруються поняття.
Формалізація	База знань наповнюється за допомогою експерта.
Реалізація експертної системи	Створення прототипу.
Етап тестування	Тестується спосіб представлення знань в експертній системі.

За останні роки процес створення експретних систем змінився, через появу нових програмних засобів, складність і час розробки зменшились.

Під час виконання дипломної роботи, я провів аналіз найбільш відомих експертних систем у світі. Найбільш відомі системи були розроблені у 1970-х роках, але використовуються й у наші дні. Наразі більшість експертних систем застосовуються у медичній та промислових галузях.

Однією з найперших і найвідоміших єкспертних систем вважають, MYCIN. Ця експертна система була спроектована для діагностування бактерій, які викликають важкі захворювання крові. Система була спроектована за 6 років у Стенфордському університеті (рис. 1.3).

MYCIN складається з апарату вводу і бази даних, де знаходиться близько 500 запитань (правил). Після тестування цієї системи було виявлено, що у 69% випадків система пропонує дієву терапію, в процентному співвідношенні – це більше ніж у експретів по інфекційним хворобам.

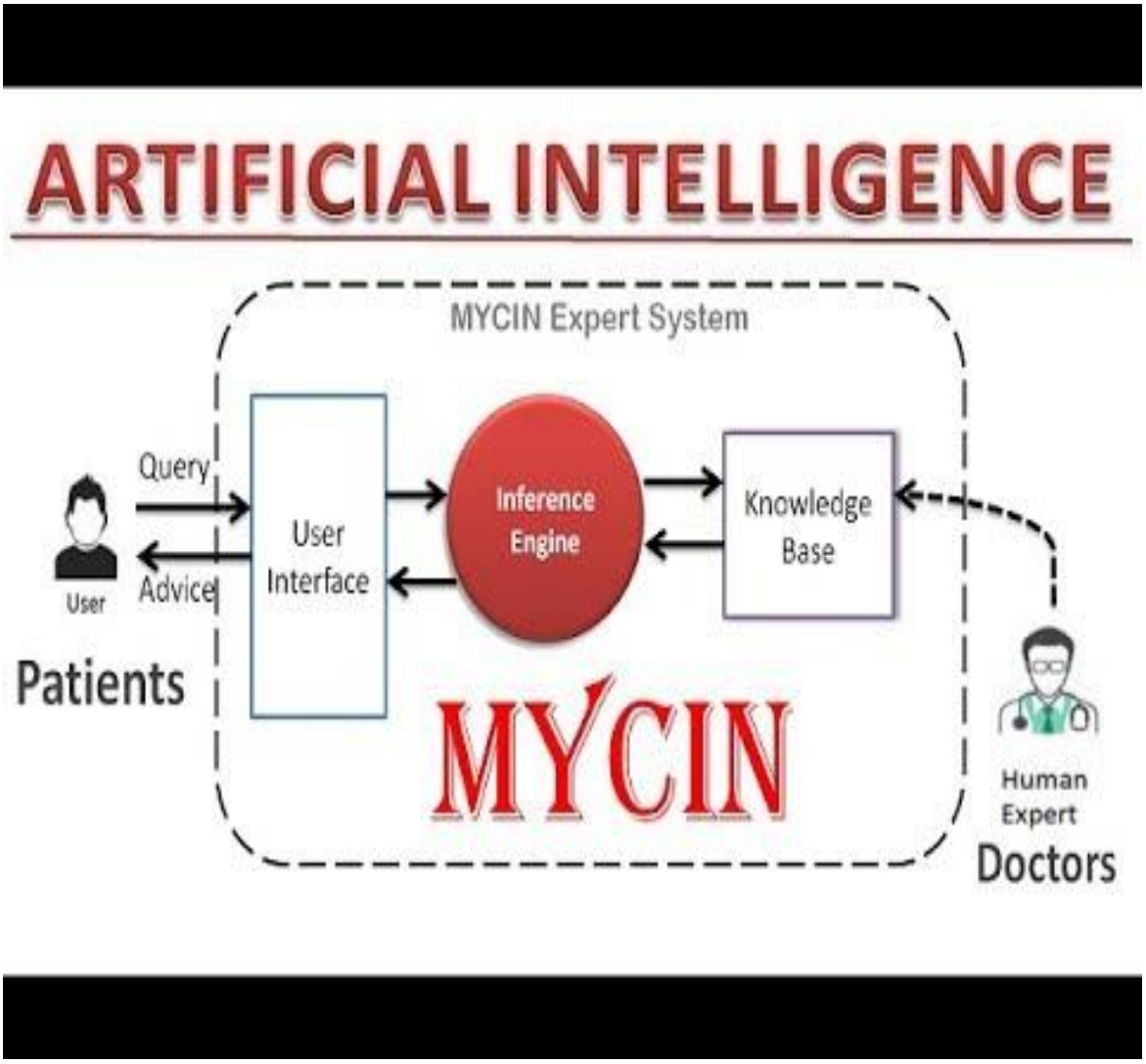

Рисунок 1.3 – Схема експертної системи MYCIN

Ще однією системою, яку використовують у сфері медицини, являється система під назвою STD Wizard. Ця система дає рекомендації та допомогає вибрати, які медчині аналізи рекомендується сдати, виходячи з особистих факторів ризику захворіти на STD. Система має вигляд, безкоштовної онлайн вікторини (рис. 1.4).

Наступною системою є SHINE – SHINE Expert System, ця система є механізмом визначення стану космічного корабля. Вона була розроблена одним з підрозділів в НАСА. Наразі ця система використовується у дослідах штучного інтелекту, а також широко поширена у додатках НАСА, де вона діагностує проблеми в підсистемах Power and Pyro, TELECOM.

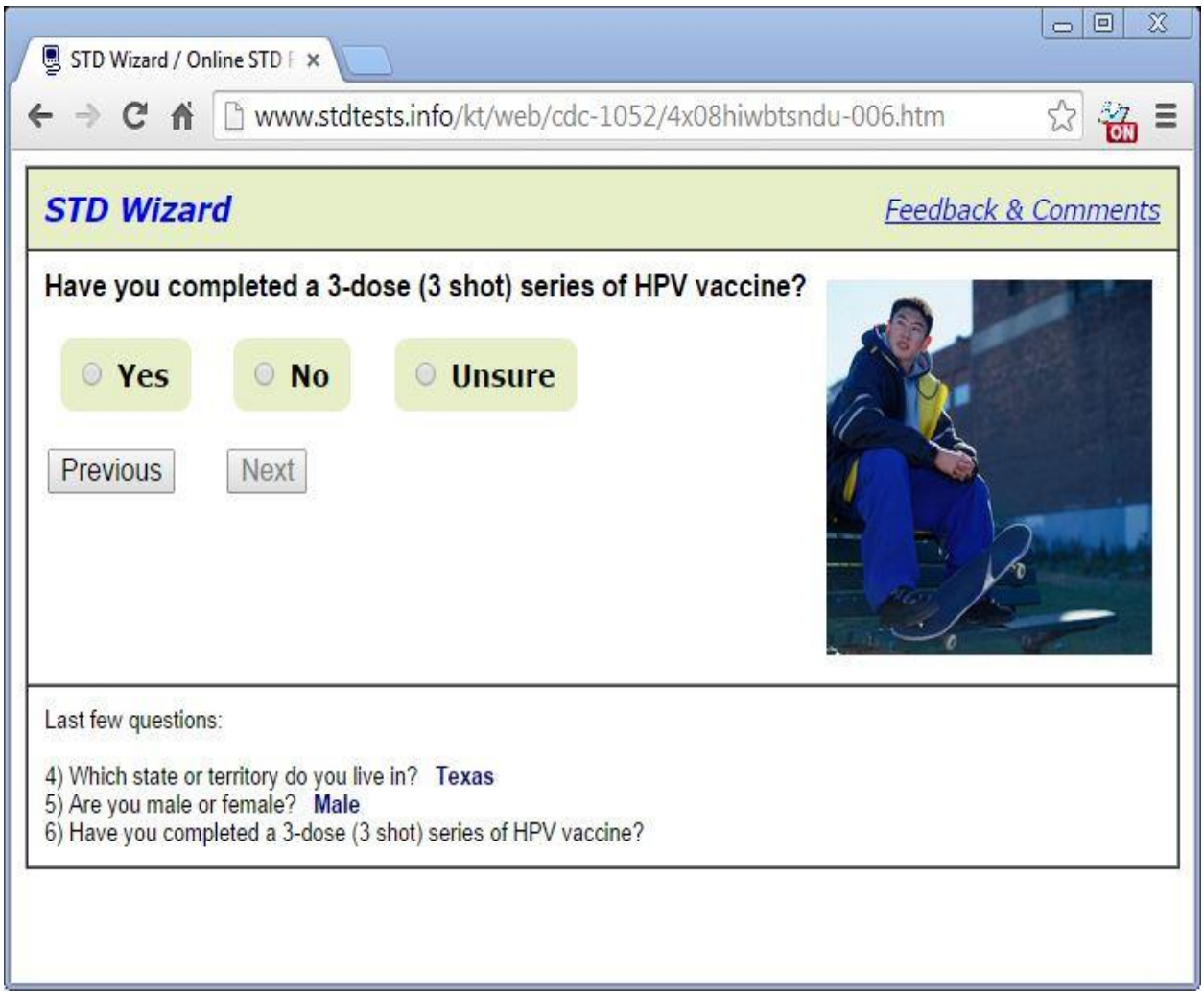

Рисунок 1.4 – Вигляд експертної системи «STD Wizard»

Експертні системи використовуються під час пошуку нафти. Назва такої системи Dipmeter Advisor, вона допомогає аналізувати дані, які були зібрані під час розвідки нафти. Dipmeter Advisor, як і MYCIN складається з бази знань, де знаходиться близько 90 запитань. Але на відміну від MYCIN, Dipmeter Advisor часто був повноцінною рабочою станцією і працював на Xerox 1100 Dolphin Lisp або на одній за машин Xerox (рис. 1.5). Ця система стала однією з найбільш успішних у світі, на рівні MYCIN, а також через успішність, широко використовувалась у підрозділах французької компанії Schlumburg.

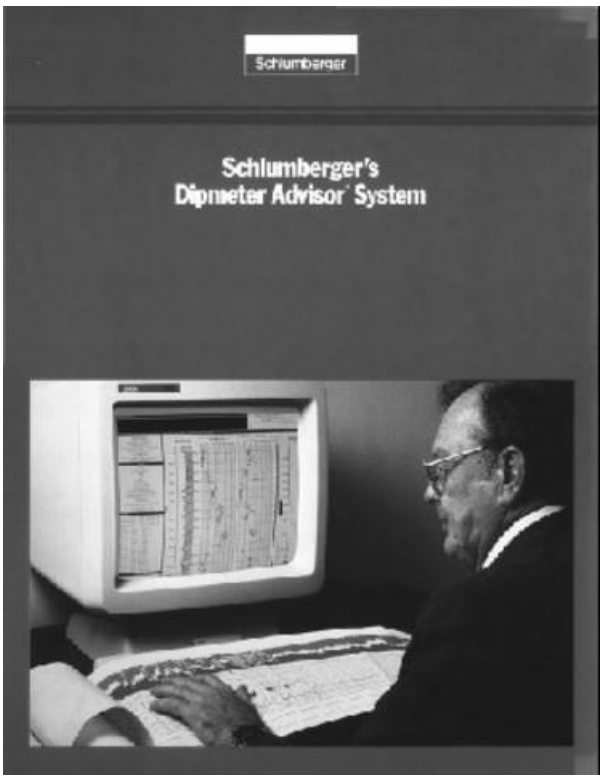

Рисунок 1.5 – Вигляд повноцінної робочої станції Dipmeter Advisor

Успішним прикладом системи в промисловості можна вважати R1 / XCON. Система допомогала розробникам визначити відповідну конфігурацію системи. Визначення виконувалось у два етапи, спочатку додаток перевіряв вимоги до системи, яку планувалось збудувати, перевірка проводилась на основі специфікації до системи. На наступному етапі, додаток визначав конфігурацію, яка б відповідала вимогам, які раніше були описані у специфікації.

У цій системі використовувався метод «запропонуй-і-застосуй». Цей метод складається с наступних кроків, ініціалізація ланцюга, пропозиція, видалення зайвого, вирішення конфліктів, вибір єдиного оператора і його застосування, оцінка мети.

Експертні системи є і в сфері інформатики, така система має назву CODES. Цю систему використовують під час розробки бази даних, коли розробник хоче застосувати підхід IDEF1. Спочатку розробник повинен описати властивості і взаємозв’язки, які повинні бути у базі даних.

На другому етапі система за допомогою своїх знань будує концептуальну схему бази даних. Ця система була реалізована на мові програмування UCI LISP.

Ще одна система, яка працює з базами даних називається RUBRIC. Вона напрвлена на допомогу користувачеві в отриманні доступу до баз даних, в яких знаходяться неформативні тексти. В цій системі взаємовідношення між темами і підтемами визнечені у виді правил, отже, коли користувач вводить якусь тему, система знаходить всі тексти, які пов’язані з цією темою. Багато систем використвоують у військовій справі. Найчастіше такі системи використвоують в аналітичних підрозділах, системи ASTA і I&W допомогають аналітикам визначити тип радару, який послав перехвачений сигнал та передбачити, коли і де буде наступне військове зіткнення відповідно.

Експертні системи вітчизняного виробника не такі відомі, як закордоні і використовуються тільки у вітчізняних компаніях. Такі системи найбільш популярні у сфері адвокатури. Одна з таких систем «STOP кілер» використвується при розслідуванні злочинів в Україні. Головне призначення цієї системи у тому, щоб побудувати версії щодо вбивці та його мотивів.

У деяких сферах, таких, як медицина і військова справа, вони використовуються і приносять користь. Використання таких систем у IT-сфері, а особливо під час аналізу, може принести користь, як компаніям так і замовникам.

1.2 Аналіз літературних джерел щодо існуючих підходів моделювання та проєктування інформаційних систем

Виконуючи цю дипломну роботу, я провів аналіз літератури, яка відноситься до експертних систем.

Першим матерілом, про експертні системи, на який я натрапив стала книга [1]. Ця книга поділяється на два напрямки, перший детально описує

теорію експертних систем, а у другому напрямку описується мова програмування CLIPS за допомогою якої можно запрограмувати експертну систему.

Книга [2], являється вводним курсом у теорію експертних систем. У перших главах описані базові концепції розробки експертних систем. Далі у книзі описуються технічні аспекти розробки експертних систем

У книзі [3], детально описується мова програмування PROLOG і її використання під час розробки експертних систем.

Автор наступної книги [4] детально описує ШІ і його теперешній стан, також зачіпається тема експертних систем. У цій книзі приведені приклади побудови експертних систем.

У книзі [5], були описані найбльш популярні та відомі експертні системи. Також детально описані інструментальні засоби, які використовуються при розробці експретних систем.

Наступним джерелом, яке я проаналізував був [6] навчальний посібник, у якому розписуються етапи розробки експертних систем (рис. 1.6). Також у цьому джерелі описуються динамічні системи і їх порвіняння з статичними системами (рис. 1.7).

Рисунок 1.6 – Етапи розробки експертної системи

У книзі [7], йде обгооврення можливості моделювання людського інтелекту. Приведені відмінності у процессі побудови різних інтелектуальних систем.

У книзі [8], приведені приклади проєктування і реалізації інтелектуальних транспортних систем. Книга призначена для експертів в області проектування інтелектуальних систем.

Проаналізувавши літературу, було набуто знань у проектуванні і реалізації інтелектуальних та експертних систем. Завдяки прикладам розроблених та сформованих систем, було виявлено патерни розробки структури і етапів створення експертних систем.

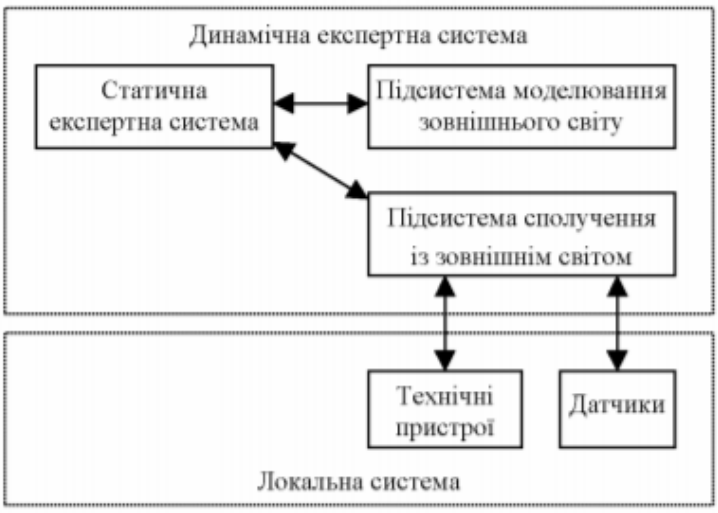

Рисунок 1.7 – Схема динамічної експертної системи

1.3 Постановка задачі

Таким чином, експертні системи були першим вдалим прототипом штучного інтелекту і використовувались у різних сферах життя. Тому ставиться завдання розробки експертної системі, завдяки якій можно визначити ступінь готовності проекту до автоматизації.

Об’єктом роботи є автоматизація тестування.

Метою роботи є розробка структури опитувальної системи і наповнення її бази знань запитаннями, відповіді на які дозволять визначити рівень готовності проєкту, а також отримати базові вказівки з впровадження змін щодо збільшення ступеню підготовки проєкту до введення автоматизації.

Для досягнення мети необхідно вирішити такі завдання:

- провести аналіз існуючих інформаційних систем;
- провести аналіз особливостей баз даних;
- провести аналіз методів системного аналізу;
- розробити структура інформаційної системи;
- наповнити структуру інформаційної системи;
- протестувати розроблену інформаційну систему.

2 РОЗРОБЛЕННЯ ПІДХОДУ ДО МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Особливості бази даних інформаційної системи врахування рівня розроблення програмного забезпечення

Наразі у світі, дані відіграють велику і визначальну роль, у кожному пристрої або програмі зберігаються дані користувача або дані, які у майбутньому будуть використані під час роботи з цією програмою. Ці дані потрібно десь зберігати, і у цьому допомагають бази даних. Зараз у світі використовується велика кількість різних баз даних, як комерційних так і відкритих, із різними моделями даних. Вони використовуються у різних сферах таких, як ігрова індустрія, у мобільних пристроях, під час роботи із вебтехнологіями, у великих корпораціях, де зберігають усі дані про працівників.

Перші бази даних почали розробляти у 1960-ті роки, була розроблена мережева модель даних CODASYL, а також у фірмі North American Rockwell почали розробляти ієрархічну базу даних. У 1970-ті роки, було вперше застосовано термін «реляційні моделі даних». У 1980-ті, почали з’являться комерційні реляційні бази даних від таких компаній, як Oracle та IBM, але наразі через появу великої кількості відкритих баз даних, які можуть конкурувати з комерційними базами від гігантів IT-інудстрії, ці гіганти почали випускати на ринок безкоштовні версії комерційних БД. У 1990-ті роки, почали випускати об’єктно-орієнтовані БД, інженерні і мультимедійні БД, які застосовуються у сферах з комплексними даними. У 2000-ні роки, у БД додали підтримку і можливість застосовувати XML.

Щоб заглибиться у тему баз даних, я вирішив проаналізувати види архітектур баз даних та іх можливості. Архітектурою баз даних називають, систему управління баз даних (СУБД). На сьогоднішній день існує два види органцізації збереження і обробки даних, централізована і розподільна.

Централізовані БД зберігаються в пам’яті однієї обчислювальної системи. Якщо база являється компонентом мережі ЄВМ, то стає можливим розподільний доступ до БД (рис. 2.1). Такі БД використовуються у локальних системах.

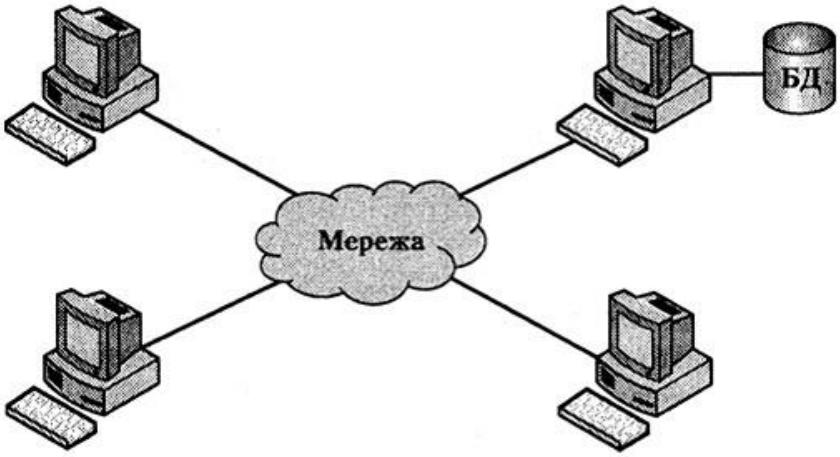

Рисунок 2.1 – Вигляд централізованої БД

Розподільні БД (рис. 2.2) – складаються з декількох частин, які зберігаються у різних ЄВМ, які в свою чергу знаходяться у однії мережі. Робота з такою БД здійснюється за допомогою СУРБД. Усі складові БД розміщуються на окремих ЄВМ, які підключені до неї і цими складовими керують локальні СУБД.

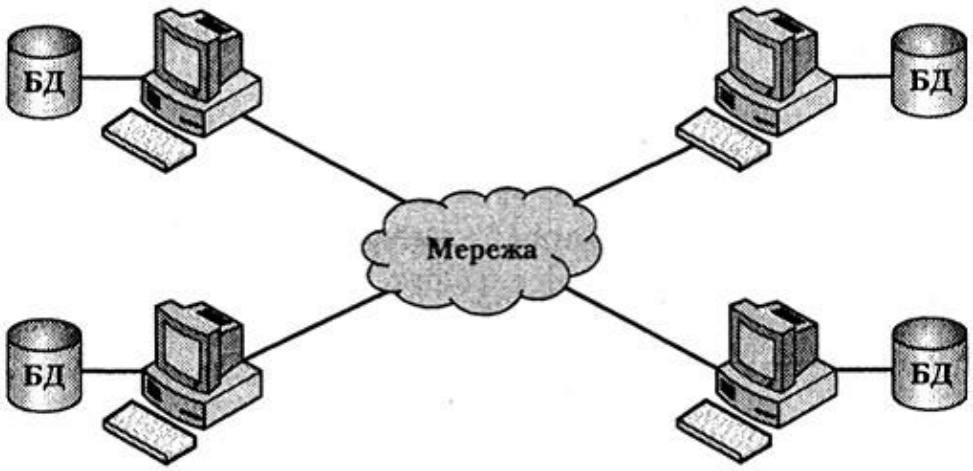

Рисунок 2.2 – Вигляд розподільної БД

Архітектури баз даних розрізняються за способом доступу до інформації, яка знаходиться у сховищі. Існує локальний, «файл-сервер», «клієнт-сервер» доступи.

Основою кожної бази даних, є модель даних. Ієрархічна, реляційна, мережна, об’єктно-орієнтований моделі були популярними у свій час.

Першою з моделей, я проаналізував реляційну модель. Реляційна БД (рис. 2.3) виглядає, як набір взаємозв’язаних відношень.

Рисунок 2.3 – Вигляд реляційної моделі даних

Всі відношення поділяються на об’єктні та зв’язкові і виглядють, як двовимірні таблиці. Інформація про об’єкти предметної області зберігаються в обьектних відношеннях і використовують ключ, який називається первинний. Зв’язок між двома чи більше відношеннями, складає зв’язкове відношення і використовує вторинний ключ. Цілісність являється одним з головних понятть при розробці БД, означає забезпечення достовірності та узгодженності даних в будь-який момент часу.

Наступне важливе поняття у термінології баз даних, це нормалізація схеми бази даних (табл. 2.1) – процесс покрокового розбиття одного відношення на декілька відношень, використовуючи алгоритм нормалізації.

Таблиця 2.1 – Етапи нормалізації схеми БД

Етапи нормалізаціїї схеми БД	**Опис етапів**
Крок 1 (Приведення до 1НФ)	На основі можеді предметної галузі, записуються функціональні залежності, що знайдені
Крок 2 (Приведення до 2НФ)	Проведення декомпозиції відношень, якщо в них є залежність від частини складного ключа.
Крок 3 (Приведення до 3НФ)	Проведення декомпозиції відношень, якщо в них є залежність деяких неключевих атрибутів.

Наступною моделью, яку я проаналізував був об’єктно-орієнтований підхід (рис. 2.4). Під час проєктування бази даних за допомогою ООП, дані представляють у вигляді класів, об‘єктів, атрибутів, методів. Цей підхід зазвичай використовують у БД, коли потрібна високопродуктивна обробка даних.

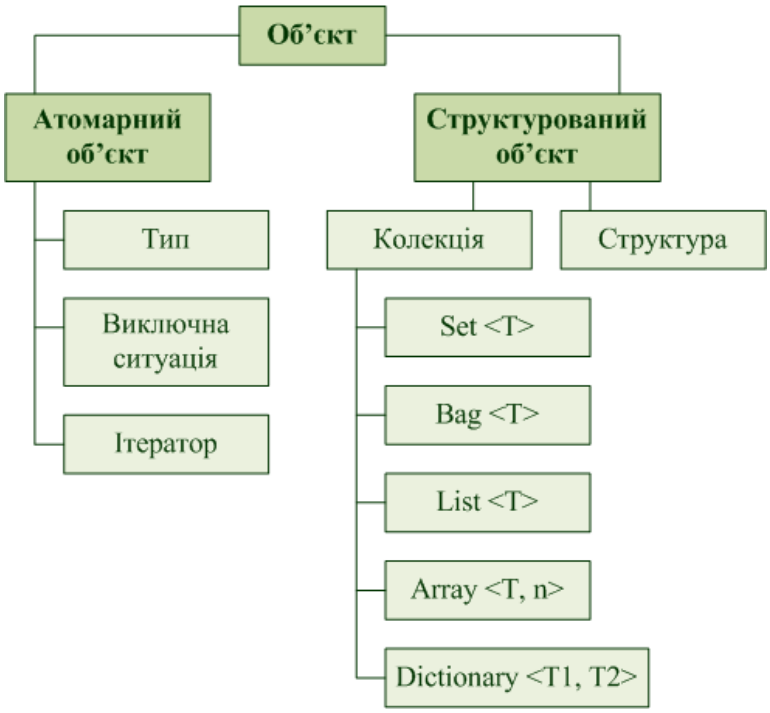

Рисунок 2.4 – Вигляд об’єктно-орієнтованої моделі даних

Одною з найбільш популярних моделей даних, була ієрархічна модель (рис. 2.5). Ієрархічна модель має вигляд деревоподібної структури, де у кожного батьківського елемента існує декілька дочірніх об’єктів.

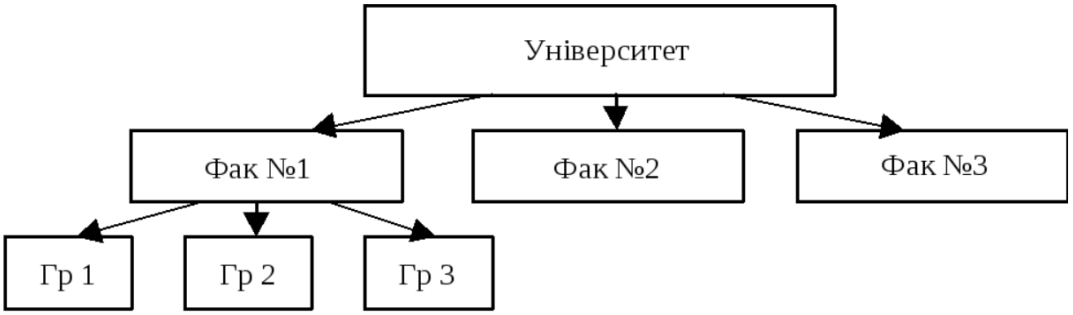

Рисунок 2.5 – Вигляд ієрархічної моделі даних

Зв’язки між об’єктами нагадують генеалогічне дерево і належать до виду «один до декількох». Найбільшою перевагою цієї моделі, є можливість швидкого доступу до заздалегідь визначених зв’язків, але бази даних, які використовують цю модель, є негнучкими і це є недоліком цієї моделі. До недоліків можна віднести і неможливість пошуку по верхнім вузлам з нижніх рівнів.

Ще однією моделлю є мережева (рис. 2.6), яка має вигляд графа. У цій моделі навідміну від ієрархічної, можливі цикли, це є основною відмінністю між цими моделями.

Щоб заглибитись у тему, я проаналізував найбільш популярні системи управління базами даних, комерційні такі, як об’єктно-реляційна система Oracle Database, реляційну систему Microsoft SQL Server і з відкритим кодом MySQL і PostgreSQL.

Спочатку було проведено аналіз комерційної СУБД Oracle Database від IT-гіганта Oracle. Oracle Database використовує термін «база даних» описуючи фізичне сховище, а термін «екземпляр» описуючи програмне забезпечення. Екземпляр виконується на конкретному комп’ютері, а база даних зберігається у фізичному сховищі, такому, як диск, який підключається до комп’ютера або сервера.

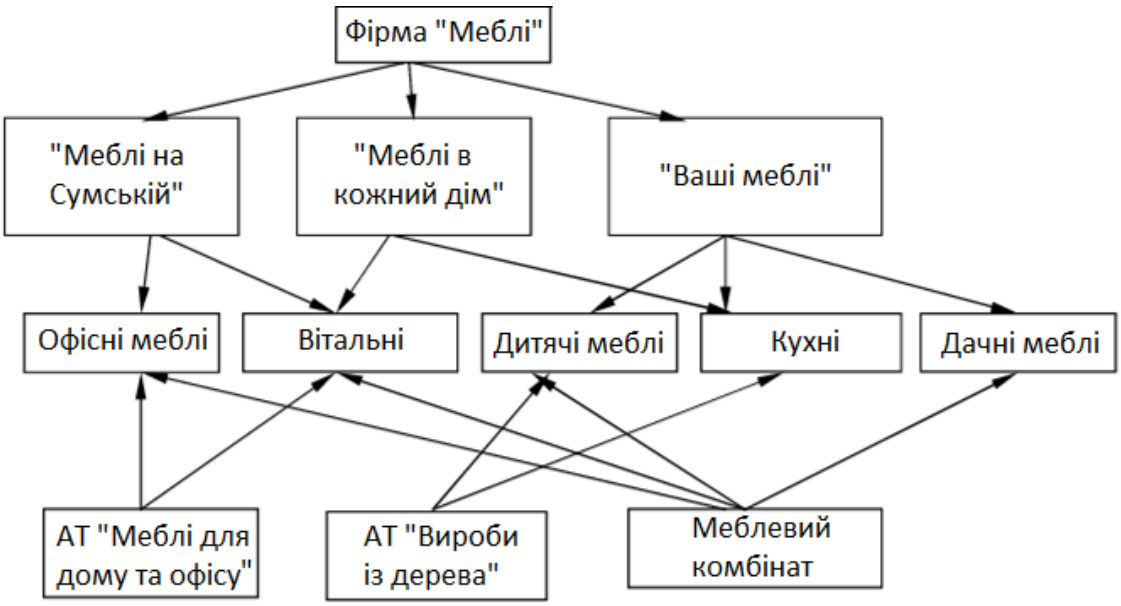

Рисунок 2.6 – Вигляд мережевої моделі даних

База даних Oracle Datbase складається з табличних просторів, які в свою чергу складаються з файлів даних (табличні простори являються логічною структурою, а файли даних – фізичною структурою), у яких зберігаються дані. На рисунку 2.7 табличний простір DATA1 складається з одного фалйа даних data1_01.dbf, натомість табличний простір DATA2 (рис. 2.8) складається з двох файлів даних data2_01.dbf і data2_02.dbf.

Рисунок 2.7 – Табличний простір DATA1

Рисунок 2.8 – Табличний простір DATA2

База даних Oracle складається з трьох файлів: control files, datafiles і redo log files. В control files знаходиться інформація про місцезнаходження інших фізичних файлів, вони вказують на datafiles і redo log files, a redo log files можуть міняти записи в datafiles.

Наступною базою даних є Microsoft SQL Server, ця система управління реляцыйними базами даних розробляється копанією Microsoft. SQL. Бази даних SQL Server збергігаються у файлах і використовують мову запитів Transact-SQL.

MySQL – це система керування реляційними базами даних з відкритим кодом, була створена, як альтернатива комерційними базам даних. MySQL являється багатопотоковим сервером баз даних, який характеризується швидкістю і простотою використання.

Ще однією системою керування базами даних є PostgreSQL, яка створена на основі СУБД Postgres. Ця база даних не контролюється однією компанією, а розвивається і підтримується групою людей і компаній, які використовують цю базу даних.

Функції у PostgreSQL представлені, як блоки коду, які виконуються не на клієнті БД, а на сервері. Вони можуть бути написані на SQL, але щоб реалізувати додаткову логіку, потрібно використовувати мовні розширення.

2.2 Застосування методів системного аналізу предметної області

Системний аналіз – це науковий метод пізнання, який використовують під час проектування, створення і єксплуатації комп’ютерних систем аналізу, прогнозування, управління.

Цей метод являється послідовністю дій з установлення структурних зв’язків між змінними або елементами досліджуваної системи (рис. 2.9). Системний аналіз вперше з’явився під час застосування комп’ютерної техніки у бізнесі та інших сферах.

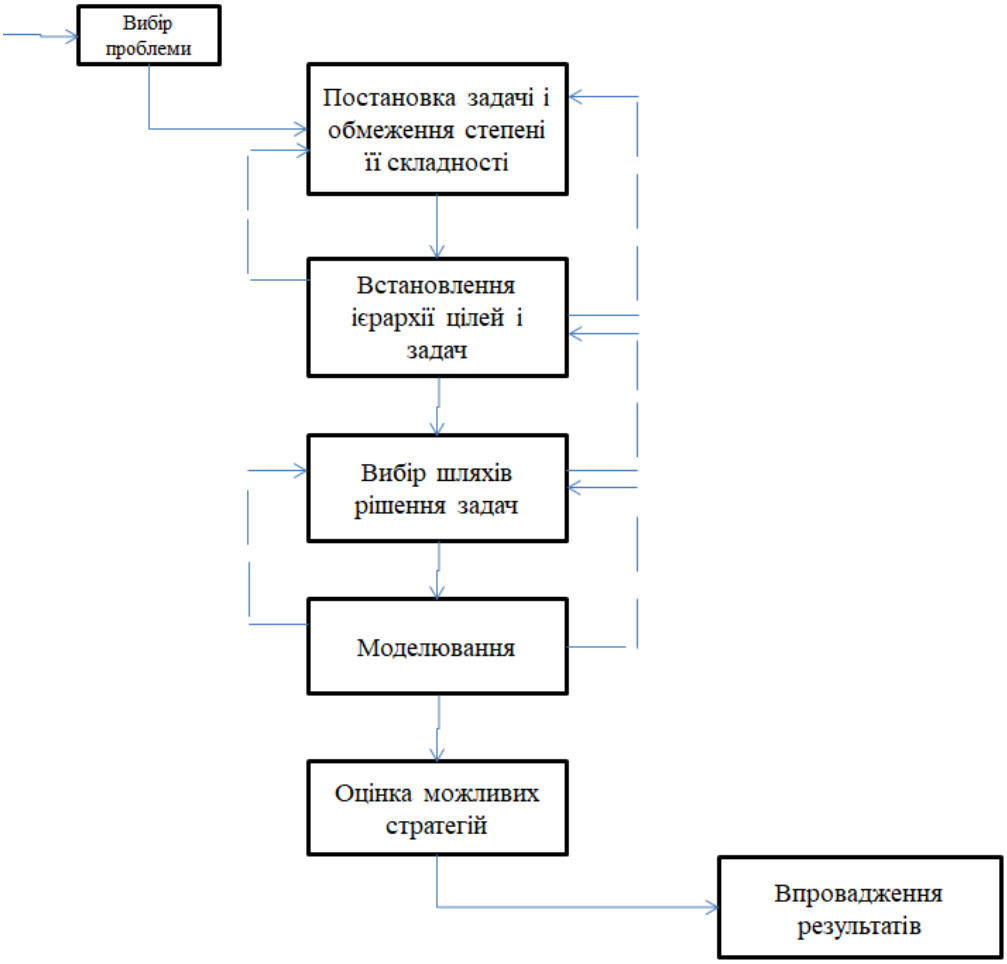

Рисунок 2.9 – Етапи системного аналізу

Системний аналіз розвивається у таких напрямках, як системи і методи прийняття рішень, аналітика даних, консолідована інформація. Системний аналіз наразі бере за основу системні принципи, такі як, принцип оптимальності – вибір варіанту, який є найкращим за показниками для попередньо заданих умов (табл. 2.2).

СА створює основу для послідовного і логічного підходу до проблеми прийняття рішень. Проблеми поділяють на три класи – перший з яких, добре структуровані проблеми, наступний – це якісно виражені або неструктуровані проблеми, третій клас – це слабо структуровані проблеми. Для вирішення проблем, які належать до добре структурованих, будється адекватна математична модель під час використання методології «дослідження операцій».

Таблиця 2.2 – Принципи системного аналізу і їх опис

Принципи системного аналізу	Опис принципів
Принцип оптимальності	Вибір варіанту, який є найкращим за показниками для попередньо заданих умов.
Принцип емерджентності	Чим більша система, тим більше можливість того, що властивості цілого будуть відрізнятися від властивостей частин.
Принцип системності	використовує підхід до об’єкта як до комплексного утворення.
Принцип ієрархії	Тип структурних відносину складніх багаторівненвих системах.
Принцип інтеграції	Дослідження спрямовуються на вивчення інтегративних властивостей.
Принцип формалізації	Направлений на отримання кількісних і комплексних характеристик об’єкта і його елементів.

Під час вирішення слабо структурованих проблем використовують системи підтримки рішень. Процедура прийняття рішень поділяється на наступні етапи – формулювання проблеми, визначення цілей, визначення критеріїв досягнення цілей, побудова моделей обґрунтування рішень, пошук оптимального рішення, узгодження рішення, підготовка рішення до реалізації, затвердження рішення, керування ходом реалізації рішення, перевірка ефективності рішення (тестування).

Методи системного аналізу поділяються на неформальні і формалізовані.

Неформальні методи поділяються на методи «мозкової атаки», метод експертних оцінок, метод «Дельфі», метод сценаріїв, діагностичні методи, морфологічні методи, метод дерева цілей. Формалізовані методи поділяються на графічні, статичні і аналітичні методи. Графічні методи включають в себе мтаричні і мережеві методи. Статичними методами називають математичну статистику, теорію імовірності, теорію масового обслуговування. Аналітичні методи – це методи, як класичної математики, так і матиматичного програмування.

Щоб заглибитись у системний аналіз, я проаналізував методи системного аналізу – їх недоліки і переваги, а також у якій сфері застосовується.

Перший метод, який я проаналізував став метод «мозкової атаки». Цей метод поширився з початку 50-х років після книги [9] і з самого початку був націлений на пошук нових ідей. Цей метод полягає в тому, щоб зібрати групу людей (10 – 15 людей), які входять до певного підрозділу і призначається голова групиі, який має досвід застосування методу «мозкової атаки». Учасники групи повинні висловити найбільшу кількість ідей за встановлений час «Мозковий штурм» поділяється на два етапи, під час першого группа висуває ідеї по заданій темі, всі висловлені ідеї фіксуються.

Під час наступного етапу група оцінює та розвиває ідеї, після чого відбирає кращі ідеї. Під час використання методу «Мозкового штурму» команда дотримується певних правил, перше правило – абсолютна заборона на будь-яку критику, друге правило – заохочення найнеймовірніших та фантастичних ідей, кількість яких не обмежена, третє правило – фіксація ідей без вказівки авторства, четверте правило – відбір найбільш цінних ідей групою експертів після їх генерування [10-15].

Під час аналізу цього методу я виявив наступні недоліки: учасники слідкують за думкою більшості, підтримка ідеї найбільш авторитетного учаника групи і небажання відмовитись від своєї ідеї.

Наступний метод – метод Дельфі, найбільш поширений на Заході. Спочатку він був запропонований, як одна з процедур методу «мозкової атаки», але потім його почали використовувати, як самостійний метод. Метод Дельфі [16-18] відрізняється від методів експертної оцінки, через відмову від групових обговорень. Під час проведення експертизи (рис. 2.10) використовуючи цей метод формується команда експертів і йде їх послідовне опитування, після чого відповіді узагальнюються. Після узагальнення експертам дають додаткову інформацію, після чого вони коректують первісні відповіді.

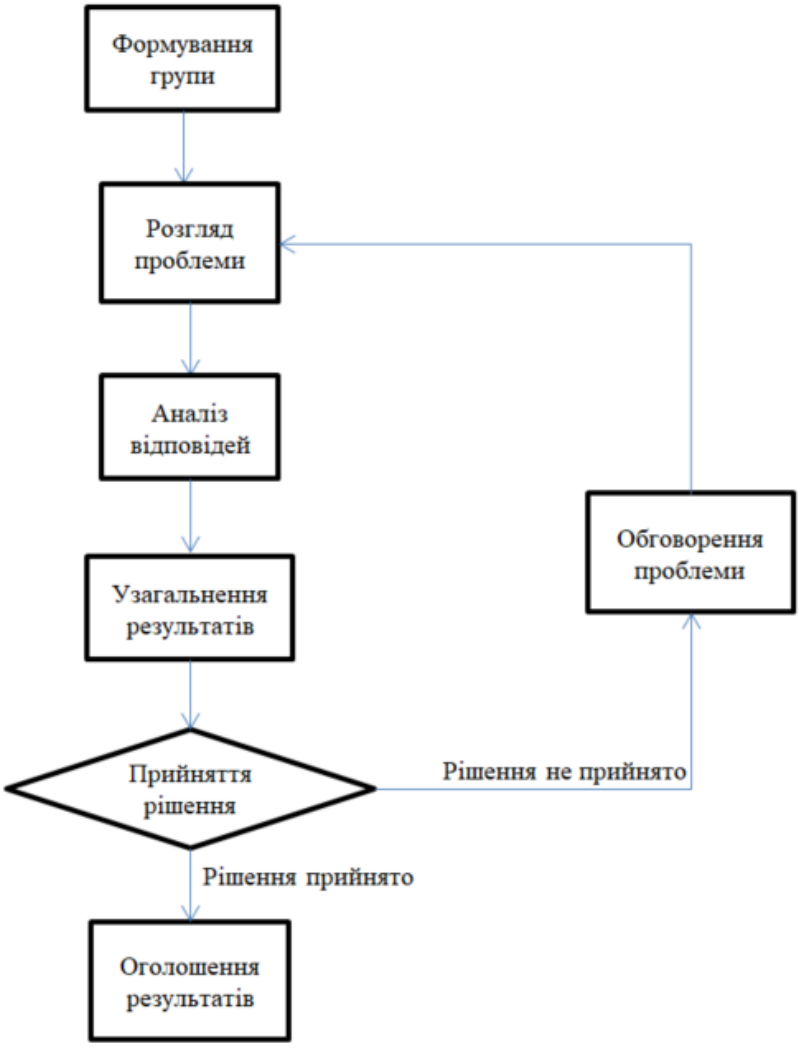

Рисунок 2.10 – Вигляд експертизи за методом Дельфі

Алгоритм методу Дельфі виглядає наступним чином: використовується послідовність циклів «мозкової атаки», на питання в анкетах можно дати кількісну оцінку відповідям експертів, проведення опитуваннння в кілька турів, кожному експерту надають коефіцієнт значимості оснгрунтуючись на їх досвіді. Під час використання цього методу, намагаються зменшити кількість експертів, щоб зменшити витрати і для цього використовують наступну формулу:

$$N_{min} = 0{,}5 \cdot (\frac{3}{b} + 5) ,$$

де b – задана помилка результату прогнозування $0 \leq b \leq 1$.

Через проведення опитування для кожного експерта окремо, цей метод потребує значних часових витрат, а також через неодноразовий перегляд своїх відповідей експертами, може вплинути на їх бажання проходити цю проецедуру.

Наступний метод, який я проаналізував, став метод – експертних оцінок [19]. Метод експертних оцінок складається з індивідуальних і колективних методів (рис. 2.11).

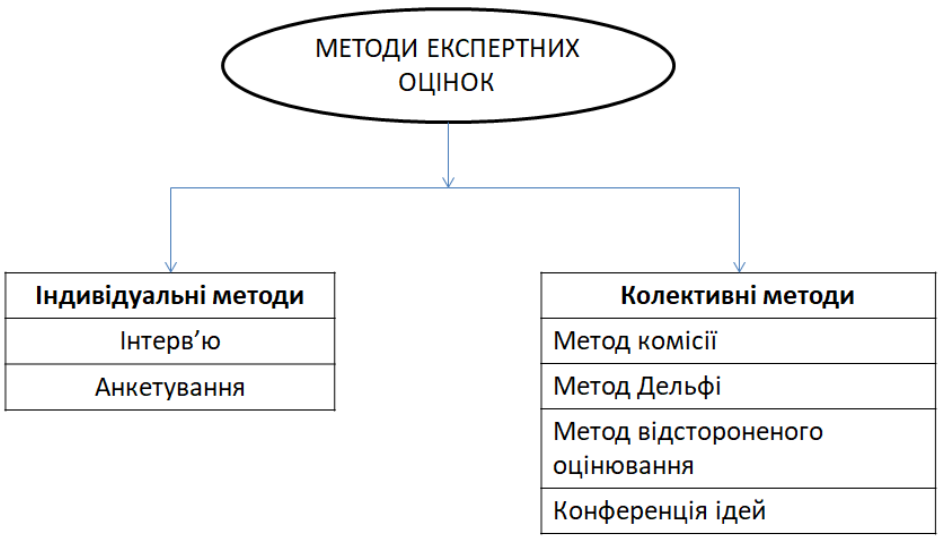

Рисунок 2.11 – Різновиди методу експертних оцінок

Основою цього методу (рис. 2.12) являються різні форми опитувань з оцінками і вибором найкращого варіанту.

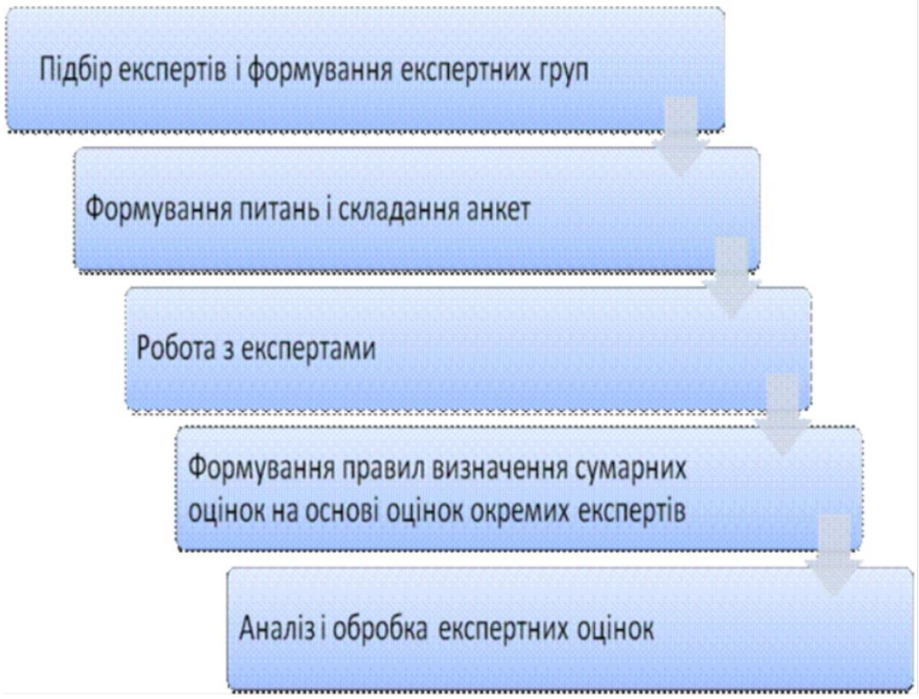

Рисунок 2.12 – Етапи методу експертних оцінок

При обробці експертних оцінок використовують методи теорії рангової кореляції. Використовувана процедура залежить від завдання оцінювання. Найчастіше використовуються процедури ранжування, парне порівняння, множинні порівняння, безпосередня оцінка, Черчмена – Акоффа, метод Терстоуна, метод фон Неймена-Моргенштерна. Під час використання цього методу постає спірний момент – встановлення оцінок експертам.

Ідеєю морфологічного методу являється систематичне знаходження варіантів рішення проблеми, вперше цей підхід застосував Ф. Цвіюси, швейцарський астроном [20-24]. Одним з найбільш відомих різновидів морфологічного методу є метод систематичного покриття поля, який базується на виділенні пунктів знань в області, яку досліджують, а також під час використання методу за допомогою принципів мислення заповнють деякі поля. Найбільш поширений метод – метод морфологічного ящика (рис. 2.13).

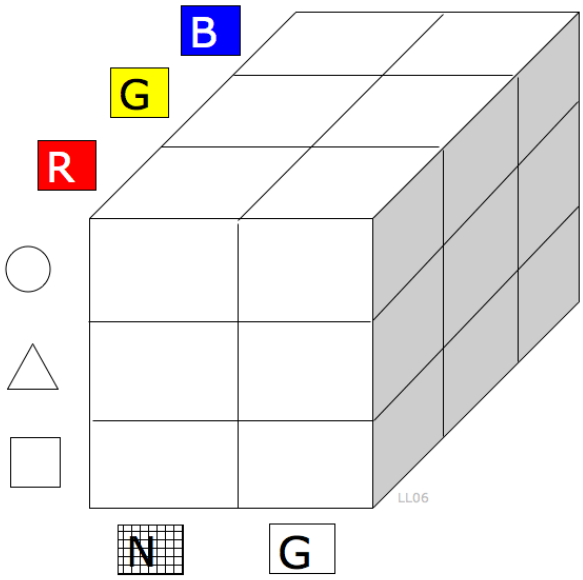

Рисунок 2.13 – Вигляд морфологічного ящика

Під час використання цього методу, визначаються всі парамтери, від яких залежить рішення проблем, вони подаються у вигляді матриць – рядків (рис. 2.14), після чого у цьому «ящику» визначається всі можливі поєднання параметрів по одному з рядка.

Отримані варіанти можуть бути заново оцінені, щоб вибрати найкращий. Ящик буває не тільки двовимірним (рис. 2.15).

№/елемент	Опора	Дошка	Кріплення	Пристосування
1)	А1	Б2	В3	Г3
2)	А2	Б1	В2	Г1
3)	А3	Б2	В1	Г2
4)	А2	Б1	В3	Г1
5)	А3	Б2	В2	Г2

Рисунок 2.14 – Приклад морфологічного ящика

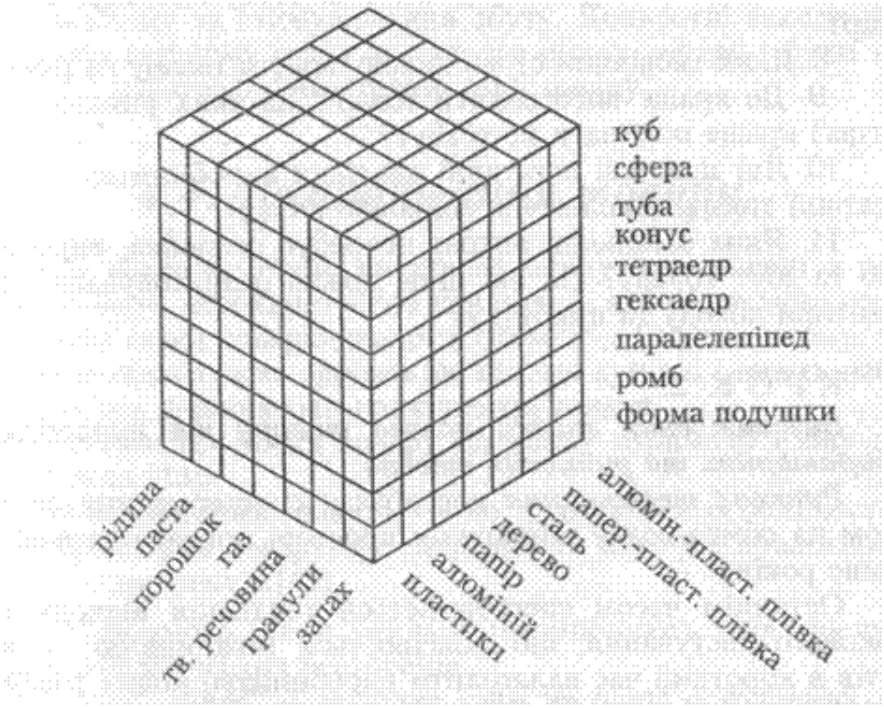

Рисунок 2.15 – Приклад морфологічного ящика

Метод дерева цілей [25-30] – використовує ієрархічну структуру, через розділення основної цілі на підцілі. Основная ідея цього методу – це декомопзиція, розділення системи на декілька частин, після цього розділення частин системи на меньші частини. Дерево цілей (рис. 2.16) виглядає, як зв'язаний граф, вершини інтерпритуються, як цілі, ребра – як зв'язки між цілями. У дереві цілей немає циклів. Цей метод використовується для структуризації і аналізу проблеми, структуризації системи, декомпозиції критеріїв оптимальності.

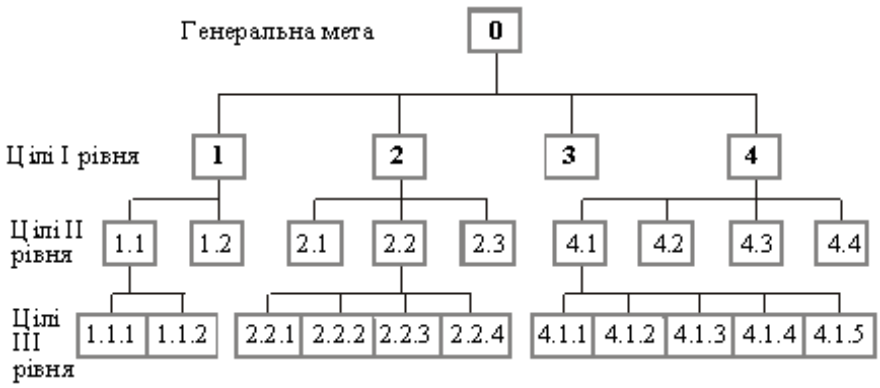

Рисунок 2.16– Вигляд дерева цілей

2.3 Моделювання структури та наповнення інформаційної системи

Моделюючи структуру я використовував сервіс Visual-paradigm, де вибрав схему Block Flow Diagram.

У вигляді квадрату я подаю питання, Р№ – номер питання, перехід між питаннями відображений у вигляді стрілок, вихід з опитувальника позначений за допомогою «End», S№ – номер проміжної форми. Моделюючи структуру, я поділив її на три гілки WEB, Mobile, Desktop (рис. 2.17).

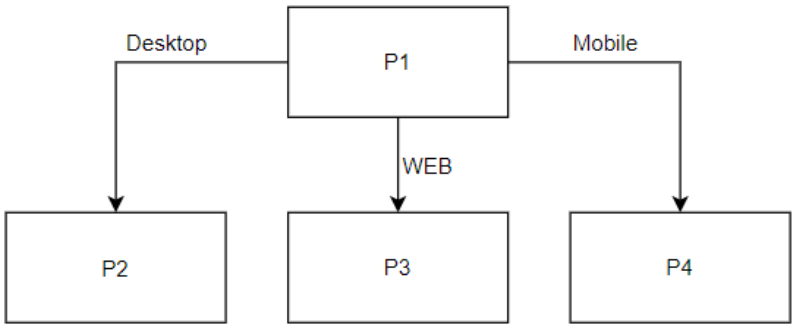

Рисунок 2.17 – Вигляд початкового етапу і розділення його на три гілки

Перша гілка – Desktop поділяється на чотири етапи, перший «етап підготовки» (рис. 2.18), другий етап «етап можливостей» (рис. 2.19), третій етап «етап інфраструктури» (рис. 2.20), четвертий етап «технічний етап» (рис. 2.21).

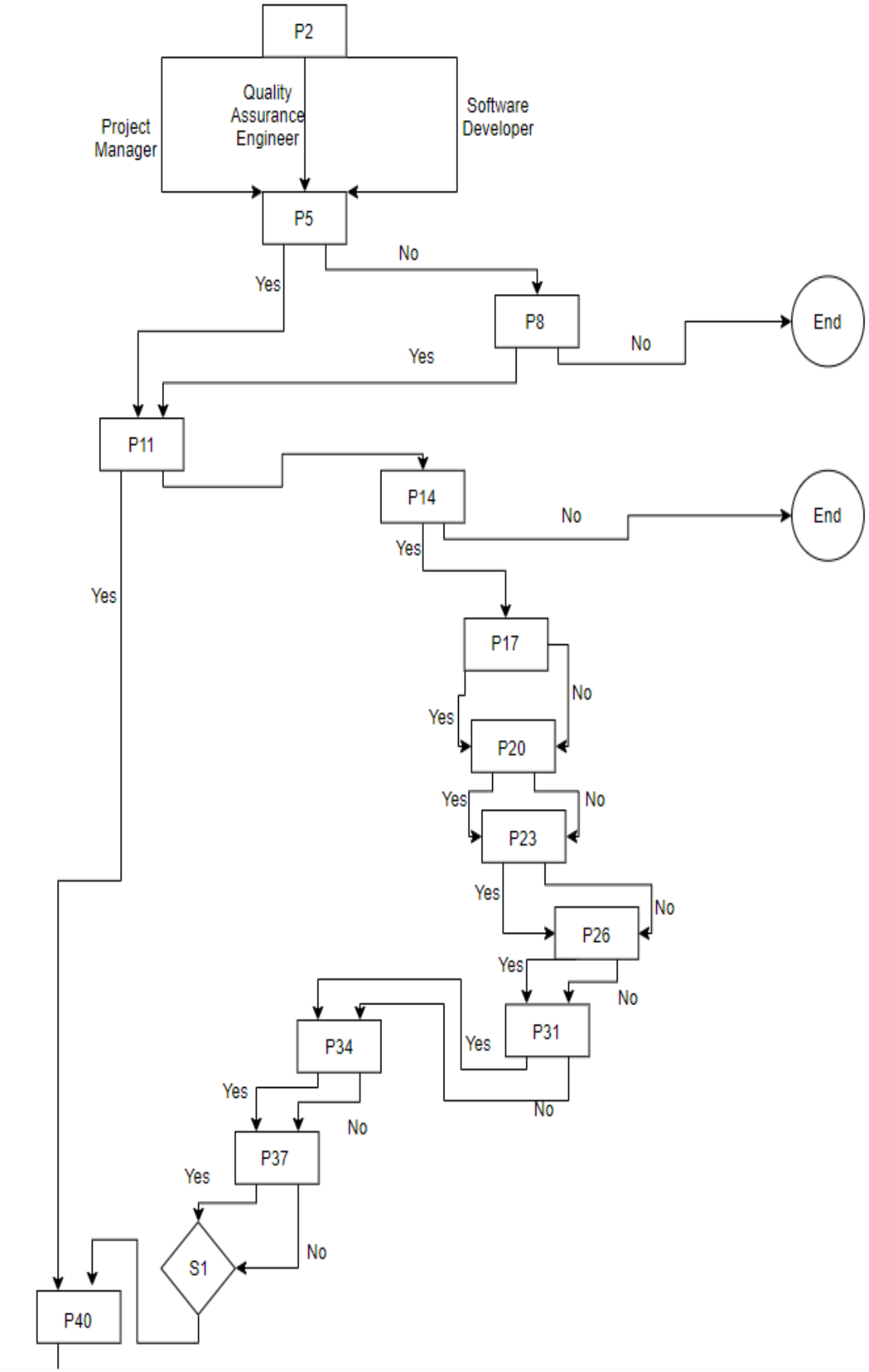

Рисунок 2.18 – Продовження гілки Desktop, перший етап

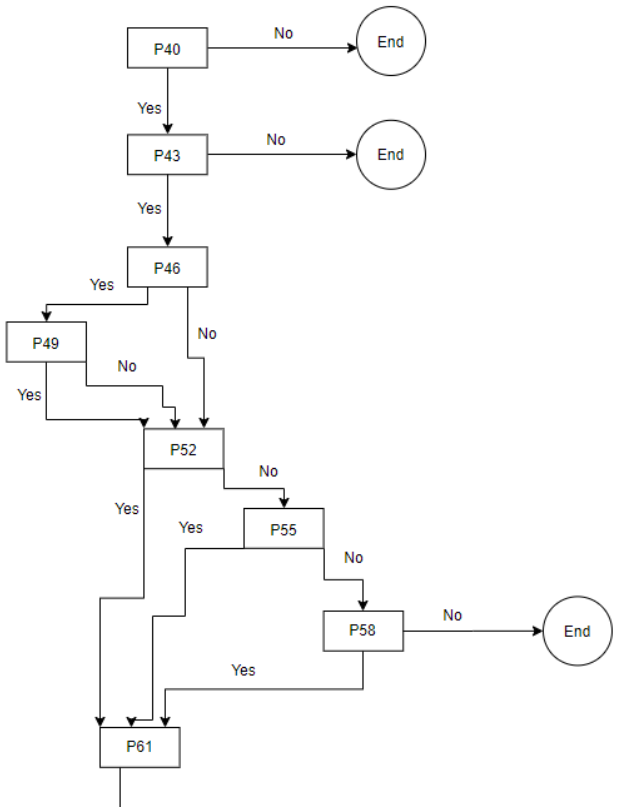

Рисунок 2.19 – Продовження гілки Desktop, другий етап

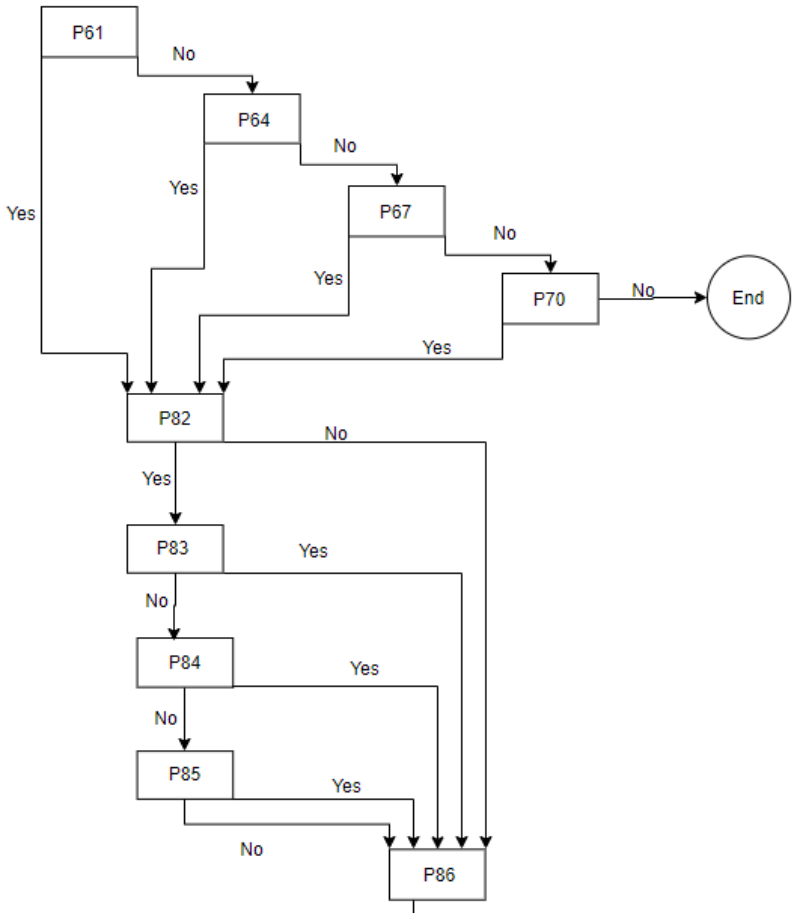

Рисунок 2.20 – Продовження гілки Desktop, третій етап

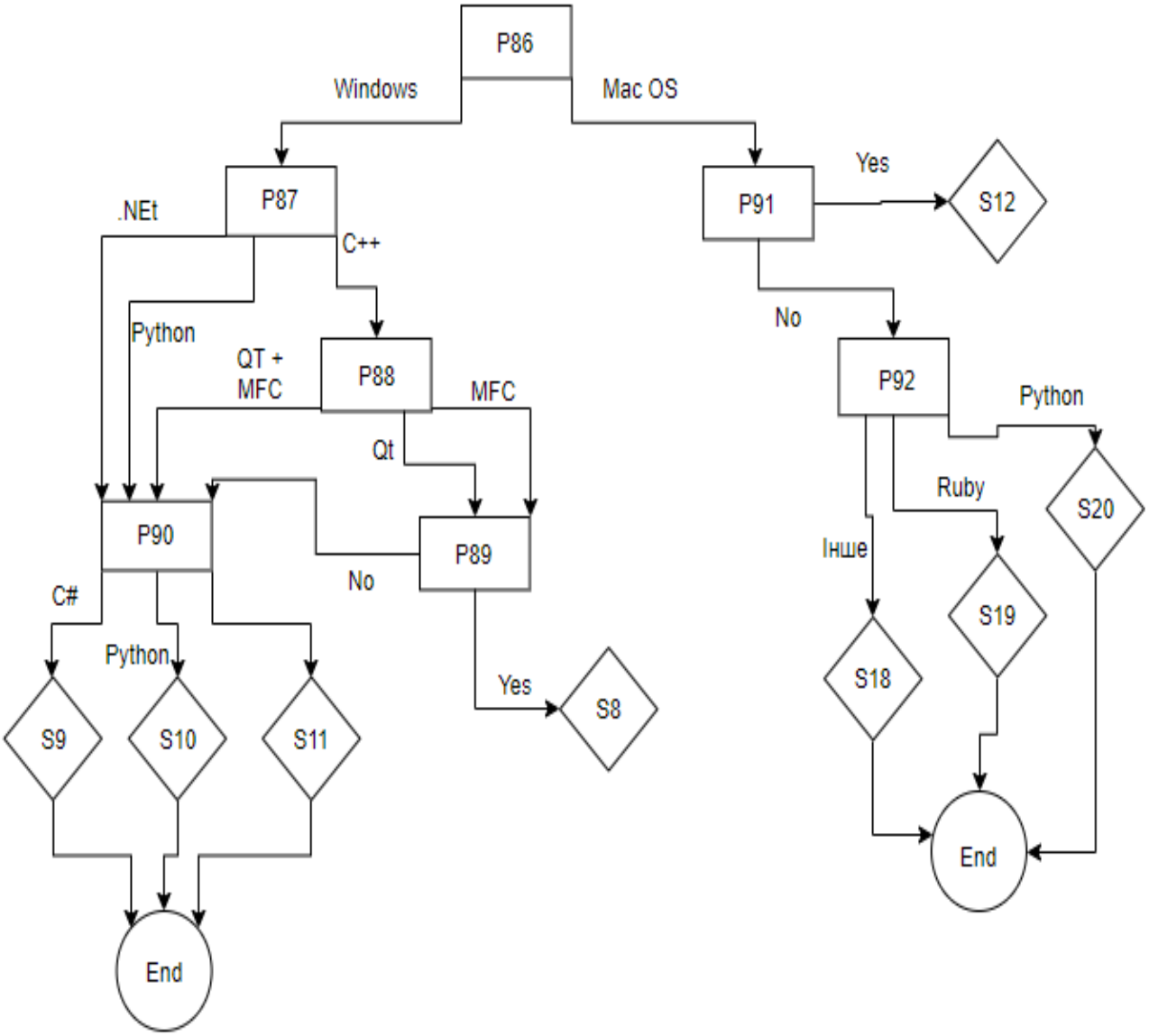

Рисунок 2.21 – Продовження гілки Desktop, четвертий етап

Друга гілка – WEB поділяється на три етапи, перший «етап підготовки» (рис. 2.22), другий етап «етап можливостей» (рис. 2.23), третій етап «етап технічний» (рис. 2.24).

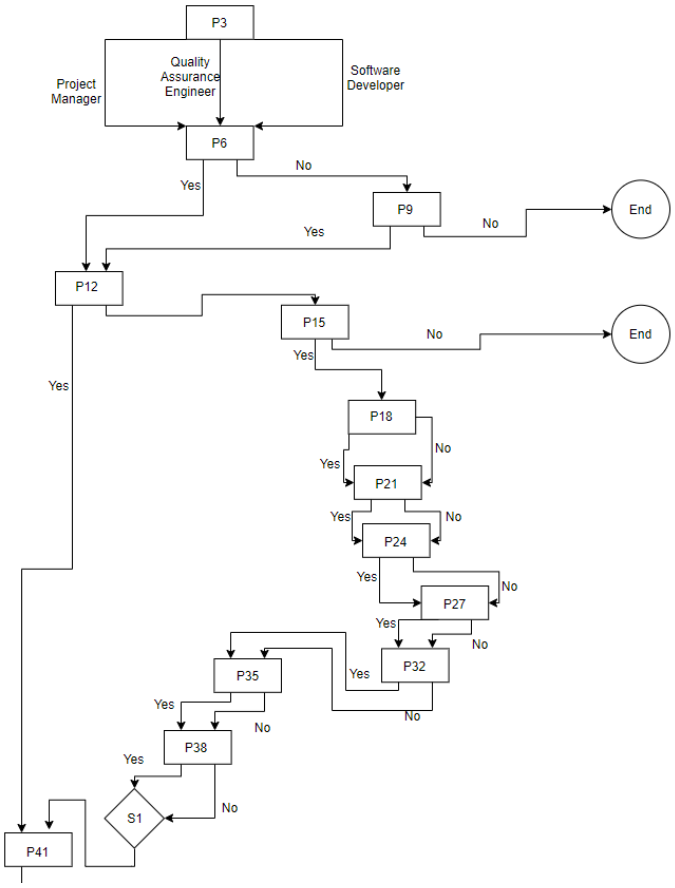

Рисунок 2.22 – Продовження гілки WEB, перший етап

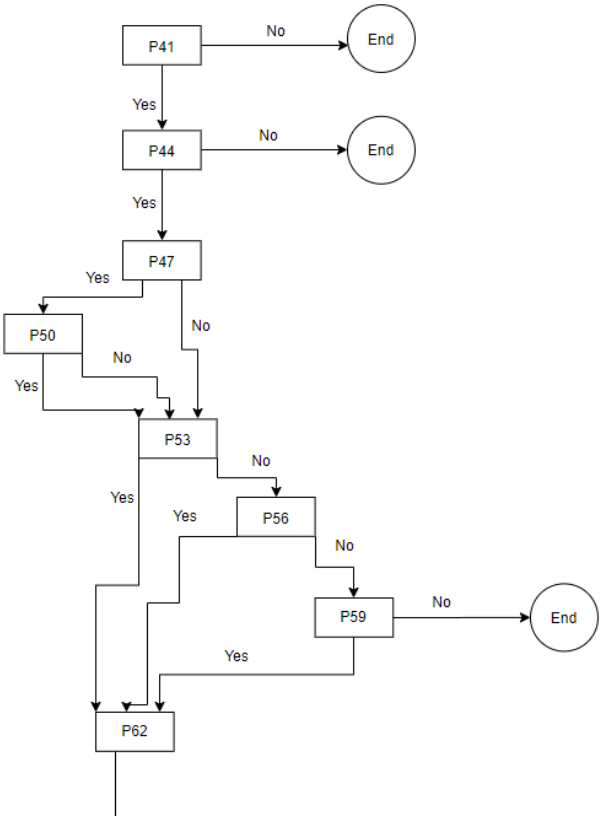

Рисунок 2.23 – Продовження гілки WEB, другий етап

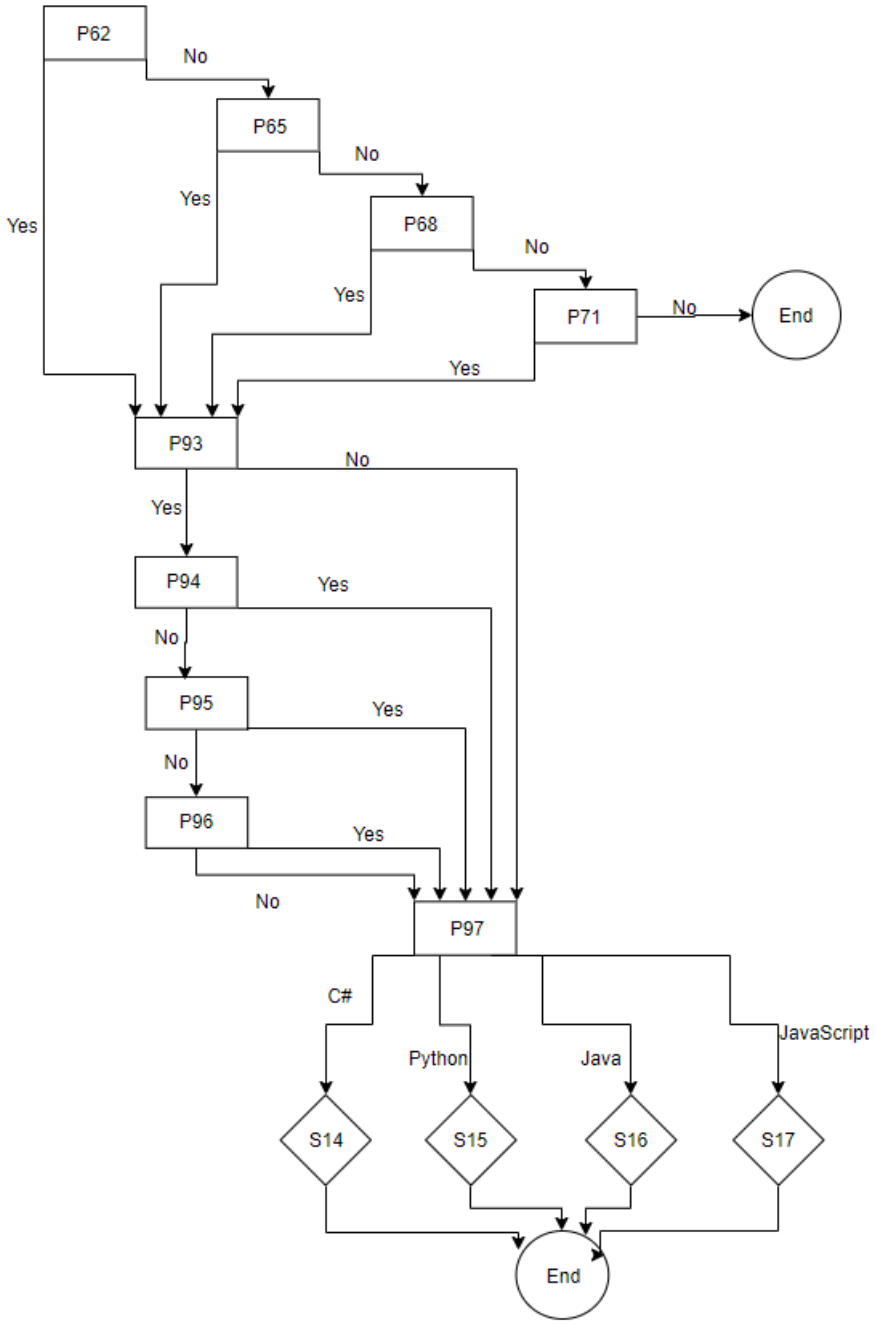

Рисунок 2.24 – Продовження гілки WEB, третій етап

Третя гілка – Mobile поділяється на чотири етапи, перший «етап підготовки» (рис. 2.25), другий етап «етап можливостей» (рис. 2.26), третій етап «етап інфраструктури» (рис. 2.27), четвертий етап «технічний етап» (рис. 2.28).

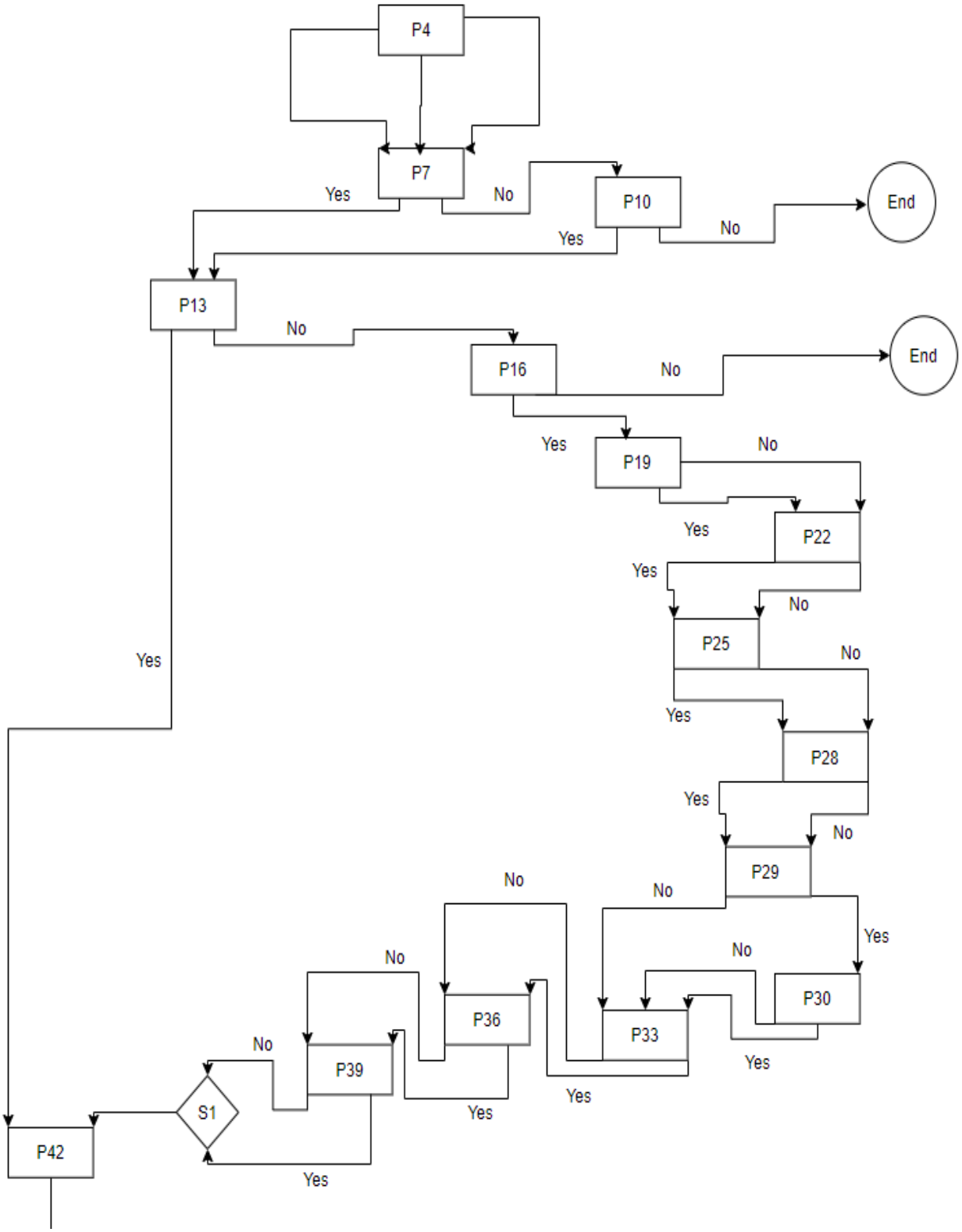

Рисунок 2.25 – Продовження гілки Mobile, перший етап

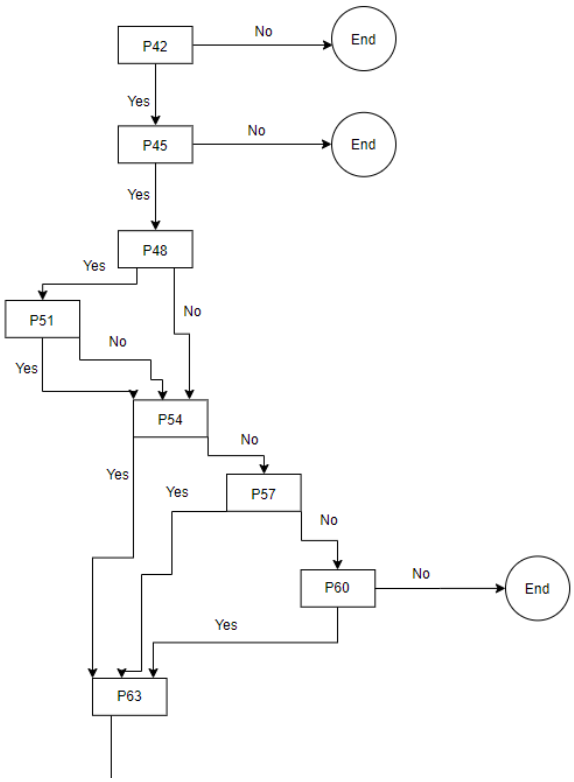

Рисунок 2.26 – Продовження гілки Mobile, другий етап

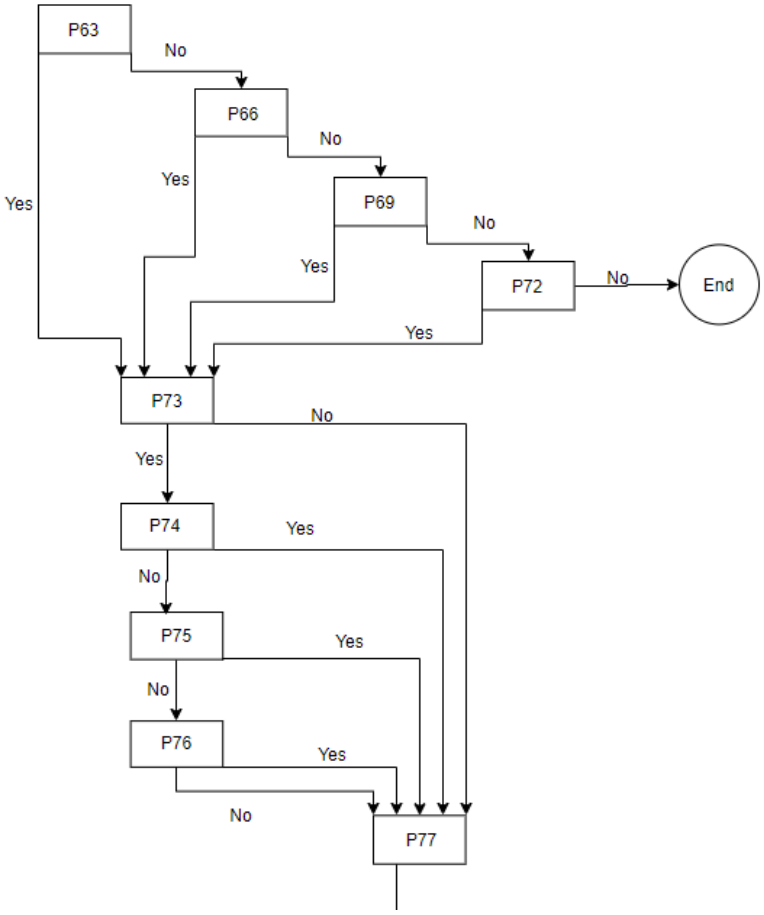

Рисунок 2.27 – Продовження гілки Mobile, третій етап

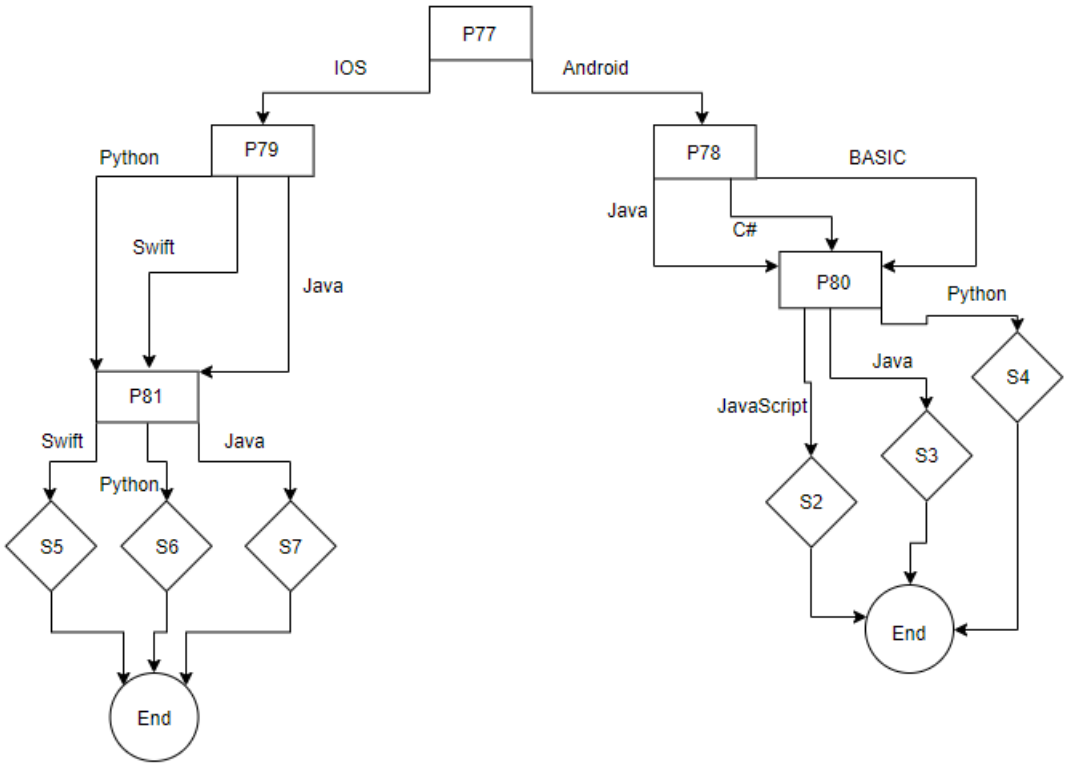

Рисунок 2.28 – Продовження гілки Mobile, четвертий етап

3 РОЗРОБЛЕННЯ МОДЕЛІ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВИЗНАЧЕННЯ РІВНЯ ГОТОВНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДО АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ

3.1 Вибір інструментальних засобів для створення моделі

Проведено аналіз існуючих систем, які дозволяють створити опитувальну систему, щоб вибрати найкращу систему, яка відповідає вимогам.

Першою системою, яка була проаналізована стала Google Forms. Google Forms (рис. 3.1) найбільш популярна форма для створення опитувань. Перевагою цієї системи є можливість використання шаблонів, також важливою функцією є можливість додавання коментарів у реальному часі. Недоліком цієї системи є збір відповідей, неможливо інтегрувати у вебзастосунок. Система являється безкоштовною.

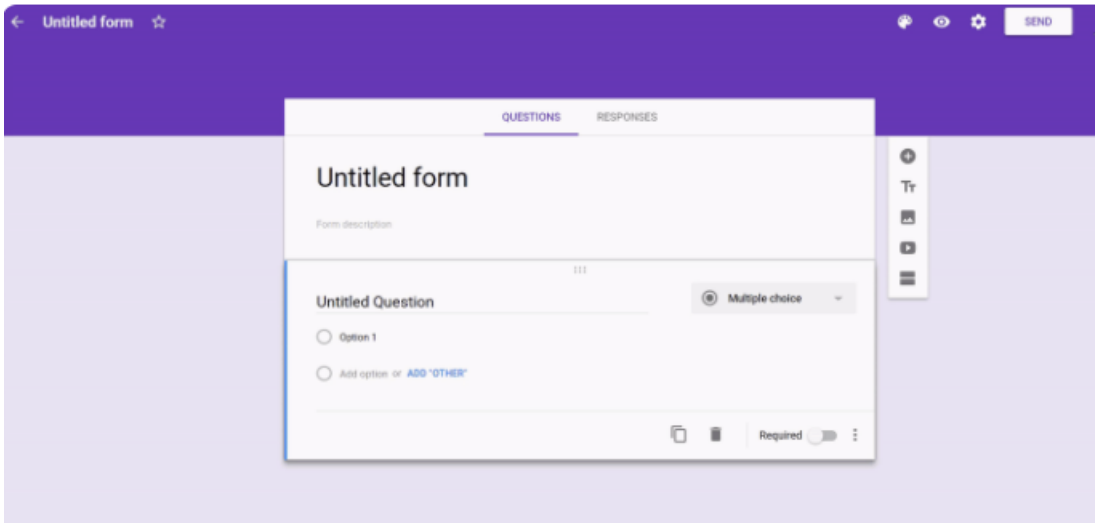

Рисунок 3.1 – Вигляд системи Google Forms

Наступною проаналізованою системою є Survio (рис. 3.2). У системі безкоштовна підписка і платна версія. У безкоштовній версії є ліміт на аналіз відгуків, користувачу доступні тільки перші 100 відгуків. У цій системі є

можливість використовувати шаблони, можна інтегрувати опитувальну систему на вебсайт, а також робити звіти.

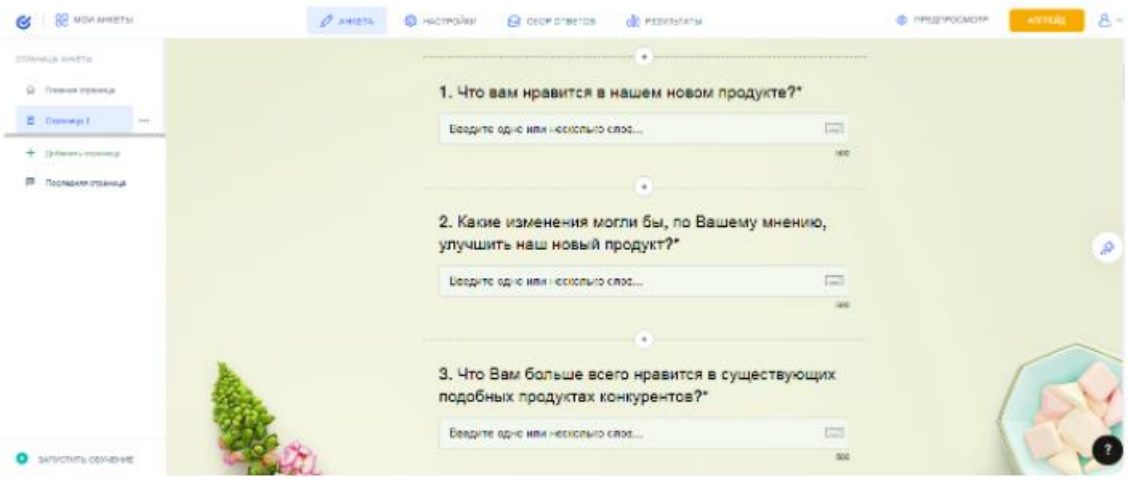

Рисунок 3.2 – Вигляд системи Survio

Одна з найбільш відомих та популярних систем є Survey Monkey (рис. 3.3). За допомогою цієї системи можно провести опитування, але у цій системі немає можливості викионати повторне анкетування. Є можливість користуватись цією системою у мобільному застосунку або браузері. Система має платні ліцензії, а також безкоштовний тариф, де можно розробити одне опитування, в якому може бути не більше 10 запитань.

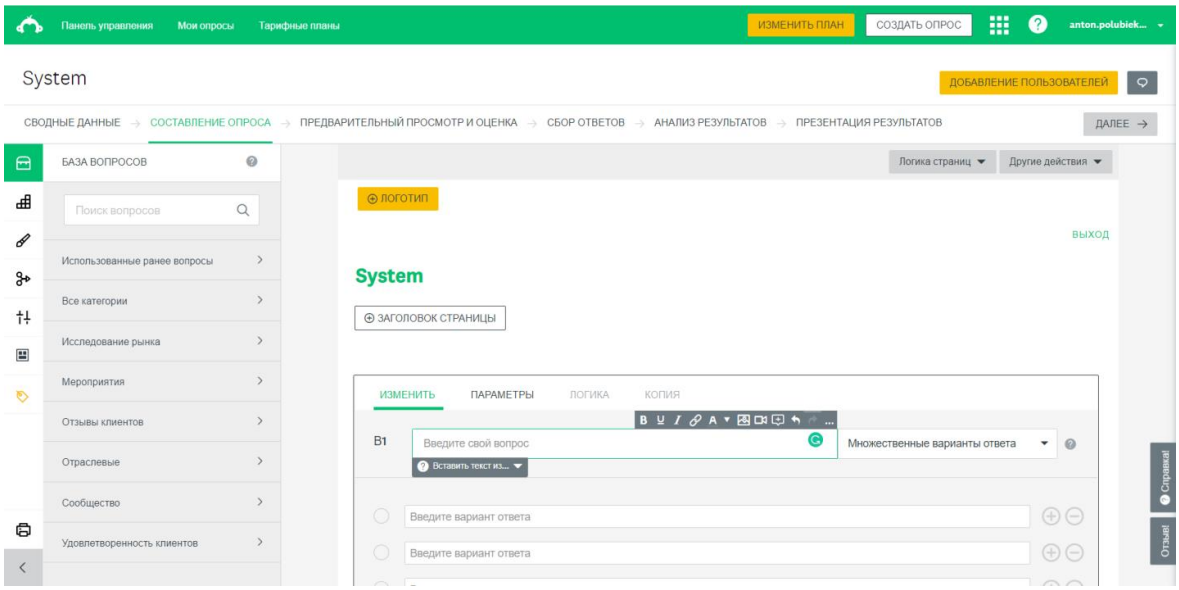

Рисунок 3.3 – Вигляд системи Survey Monkey

Остання система, яка була проаналізована є Typeform. У цій системі є безкоштовна версія, з безлімітними запитаннями. Також можна реалізувати логіку переходів від запитань до запитань (рис. 3.4). Є декілька видів запитань, а також можливість зробити закінчення опитування за допомогою форми «End».

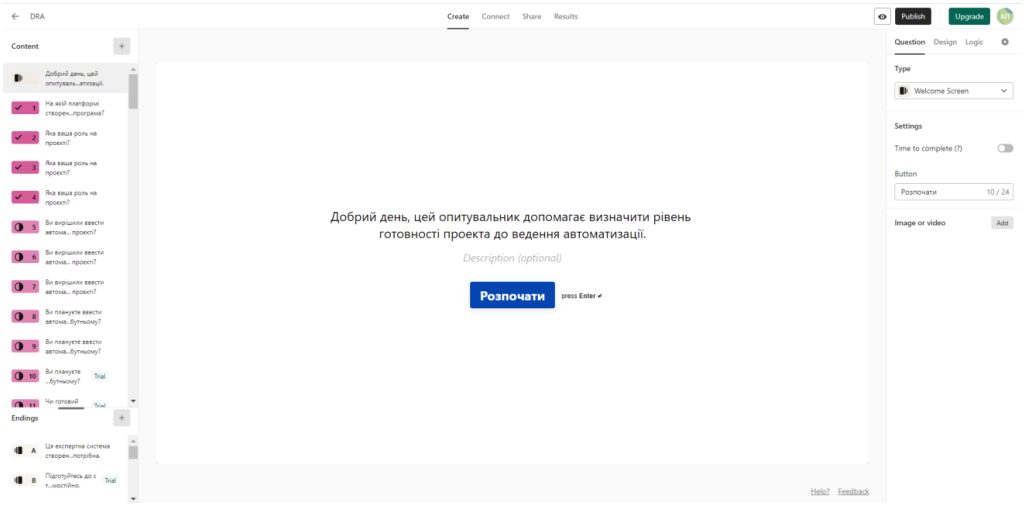

Рисунок 3.4 – Вигляд системи Typeform

Провівши аналіз цих систем, тільки система Typeform, підходить для розробки опитувальної системи, через можливість задавати логіку переходів між питаннями (рис. 3.5), великий вибір запитань, а також можливість робити передчасні виходи з опитувальної системи.

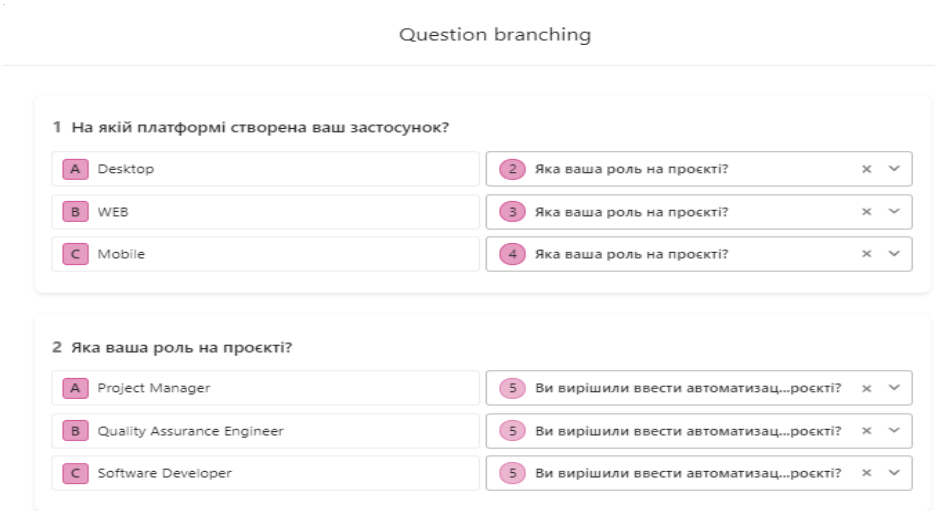

Рисунок 3.5 – Вигляд вікна, де задається логіка переходів між запитаннями

3.2 Етапи створення моделі інформаційної системи

Опитувальна система поділяється на три гілки, Desktop, WEB, Mobile. Після першого питання йде розділення на гілки.

Запитання №1 – На якій платформі створена програма?: відповідь Desktop – перехід до запитання 2, відповідь WEB – перехід до запитання 3, відповідь Mobile – перехід до запитання 4.

Ланцюг – Mobile.

Запитання №4 – Яка ваша роль на проєкті?: відповідь Project Manager – перехід до питання 7, відповідь Quality Assurance Engineer – перехід до питання 7, відповідь Software Developer – перехід до питання 7.

Запитання №7 – Ви вирішили ввести автоматизацію на проєкті?: відповідь Yes – перехід до запитання 13, відповідь No – перехід до запитання 10.

Запитання №10 – Ви плануєте ввести автоматизацію у майбутньому?: відповідь Yes – перехід до запитання 13, відповідь No – вихід з опитувальника.

Запитання №13 – Чи готовий розрахунок економічної доцільності введення автоматизації на проєкті?: відповідь Yes – перехід до запитання 42, відповідь No – перехід до питання 16.

Запитання №16 – Чи готові ви зараз створити розрахунок доцільності за допомогою опитувальної системи?: відповідь Yes – перехід до запитання 19, відповідь No – вихід з опитувальника.

Запитання №19 – Чи відома вартість серверів на 1 рік?: відповідь Yes – перехід до запитання 22, відповідь No – перехід до запитання 22.

Запитання №22 – Чи відома вам вартість необхідного апаратного забезпечення?: відповідь Yes – перехід до запитання 25, відповідь No – перехід до запитання 25.

Запитання №25 – Чи відома вам вартість необхідного програмного забезпечення на 1 рік?: відповідь Yes – перехід до запитання 28, відповідь No – перехід до запитання 28.

Запитання №28 – Чи відома вам кількість загальних тест кейсів?: відповідь Yes – перехід до запитання 29, відповідь No – перехід до запитання 29.

Запитання №29 – Ви тестуєте IOS і Android?: відповідь Yes – перехід до запитання 30, відповідь No – перехід до запитання 33.

Запитання №30 – Чи збігаються у вас тести для IOS і Android?: відповідь Yes – перехід до запитання 33, відповідь No – перехід до запитання 33.

Запитання №33 – Чи відомий вам середній час, який потрібен для проходження тест кейса?: відповідь Yes – перехід до запитання 36, відповідь No – перехід до запитання 36.

Запитання №36 – Чи відомий вам час, який потрібен на покриття кейса, автотестом?: відповідь Yes – перехід до запитання 39, відповідь No – перехід до запитання 39.

Запитання №39 – Чи відомо вам скільки разів за рік виконується цей тест кейс?: відповідь Yes – перехід до запитання 42, відповідь No – перехід до запитання 42.

Проміжна форма: Використайте наступну формулу для розрахунку економічної доцільності введення автоматизації

Запитання №42 – Чи отримано дозвіл від Project Manager?: відповідь Yes – перехід до запитання 45, відповідь No – вихід з опитувальника.

Запитання №45 – Чи отримано дозвіл від замовника?: відповідь Yes – перехід до запитання 48, відповідь No – вихід з опитувальника.

Запитання №48 – Чи є незмінною більша частина програми?: відповідь Yes – перехід до запитання 51, відповідь No – перехід до запитання 54.

Запитання №51 – Чи є у вас часові рамки?: відповідь Yes – перехід до запитання 54, відповідь No – перехід до запитання 54.

Запитання №54 – Чи готова підтримка коду зі сторони розробника?: відповідь Yes – перехід до запитання 63, відповідь No – перехід до запитання 57.

Запитання №57 – Чи є можливість виділити розробника на підтримку коду?: відповідь Yes – перехід до запитання 63, відповідь No – перехід до запитання 60.

Запитання №60 – Чи є можливість найняти розробника для підтримки коду?: відповідь Yes – перехід до запитання 63, відповідь No – вихід з опитувальника.

Запитання №63 – Чи є у QA досвід у автоматизації?: відповідь Yes – перехід до запитання 73, відповідь No – перехід до запитання 66.

Запитання №66 – Чи є можливість виділити розробника для навчання QA?: відповідь Yes – перехід до запитання 73, відповідь No – перехід до запитання 69.

Запитання №69 – Чи є можливість найняти QA з досвідом автоматизації?: відповідь Yes – перехід до запитання 73, відповідь No – перехід до запитання 72.

Запитання №72 – Чи є можливість виділити для QA час на самостійне навчання?: відповідь Yes – перехід до запитання 73, відповідь No – вихід з опитувальника.

Запитання №73 – Чи є на проєкті система безперервної інтеграції?: відповідь Yes – перехід до запитання 74, відповідь No – перехід до запитання 77.

Запитання №74 – Чи є на проєкті DevOps?: відповідь Yes – перехід до запитання 77, відповідь No – перехід до запитання 75.

Запитання №75 – Чи є можливість найняти DevOps?: відповідь Yes – перехід до запитання 77, відповідь No – перехід до запитання 76.

Запитання №76 – Чи є можливість у компанії виділити DevOps на проєкт?: відповідь Yes – перехід до запитання 77, відповідь No – перехід до запитання 77.

Запитання №77 – На якій платформі у вас програма?: відповідь IOS – перехід до запитання 79, відповідь Android – перехід до запитання 78.

Запитання №78 – На якій мові розробляється проєкт?: відповідь Java – перехід до запитання 80, відповідь C# – перехід до запитання 80, відповідь BASIC – перехід до запитання 80.

Запитання №79 – На якій мові розробляється проєкт?: відповідь Python – перехід до запитання 81, відповідь Swift – перехід до запитання 81, відповідь Java – перехід до запитання 81.

Запитання №80 – Яка мова програмування буде використана під час автоматизації?: відповідь Python – перехід до проміжної форми S2, відповідь Java – перехід до проміжної форми S3, відповідь JavaScript – перехід до проміжної форми S4.

Запитання №81 – Яка мова програмування буде використана під час автоматизації?: відповідь Swift – перехід до проміжної форми S5, відповідь Python – перехід до проміжної форми S6, відповідь Java – перехід до проміжної форми S7.

Вихід з опитувальника – отримання результату.

Ланцюг – Desktop.

Запитання №2 – Яка ваша роль на проєкті?: відповідь Project Manager – перехід до питання 5, відповідь Quality Assurance Engineer – перехід до питання 5, відповідь Software Developer – перехід до питання 5.

Запитання №5 – Ви вирішили ввести автоматизацію на проєкті?: відповідь Yes – перехід до запитання 11, відповідь No – перехід до запитання 8.

Запитання №8 – Ви плануєте ввести автоматизацію у майбутньому?: відповідь Yes – перехід до запитання 11, відповідь No – вихід з опитувальника.

Запитання №11 – Чи готовий розрахунок економічної доцільності введення автоматизації на проєкті?: відповідь Yes – перехід до запитання 40, відповідь No – перехід до питання 14.

Запитання №14 – Чи готові ви зараз створити розрахунок доцільності за допомогою опитувальної системи?: відповідь Yes – перехід до запитання 17, відповідь No – вихід з опитувальника.

Запитання №17 – Чи відома вартість серверів на 1 рік?: відповідь Yes – перехід до запитання 20, відповідь No – перехід до запитання 20.

Запитання №20 – Чи відома вам вартість необхідного апаратного забезпечення?: відповідь Yes – перехід до запитання 23, відповідь No – перехід до запитання 23.

Запитання №23 – Чи відома вам вартість необхідного програмного забезпечення на 1 рік?: відповідь Yes – перехід до запитання 26, відповідь No – перехід до запитання 26.

Запитання №26 – Чи відома вам кількість загальних тест кейсів?: відповідь Yes – перехід до запитання 31, відповідь No – перехід до запитання 31.

Запитання №31 – Чи відомий вам середній час, який потрібен для проходження тест кейса?: відповідь Yes – перехід до запитання 34, відповідь No – перехід до запитання 34.

Запитання №34 – Чи відомий вам час, який потрібен на покриття кейса, автотестом?: відповідь Yes – перехід до запитання 37, відповідь No – перехід до запитання 37.

Запитання №37 – Чи відомо вам скільки разів за рік виконується цей тест кейс?: відповідь Yes – перехід до запитання 40, відповідь No – перехід до запитання 40.

Проміжна форма: Використайте наступну формулу для розрахунку економічної доцільності введення автоматизації

Запитання №40 – Чи отримано дозвіл від Project Manager?: відповідь Yes – перехід до запитання 43, відповідь No – вихід з опитувальника.

Запитання №43 – Чи отримано дозвіл від замовника?: відповідь Yes – перехід до запитання 46, відповідь No – вихід з опитувальника.

Запитання №46 – Чи є незмінною більша частина програми?: відповідь Yes – перехід до запитання 49, відповідь No – перехід до запитання 52.

Запитання №49 – Чи є у вас часові рамки?: відповідь Yes – перехід до запитання 52, відповідь No – перехід до запитання 52.

Запитання №52 – Чи готова підтримка коду зі сторони розробника?: відповідь Yes – перехід до запитання 61, відповідь No – перехід до запитання 55.

Запитання №55 – Чи є можливість виділити розробника на підтримку коду?: відповідь Yes – перехід до запитання 61, відповідь No – перехід до запитання 58.

Запитання №58 – Чи є можливість найняти розробника для підтримки коду?: відповідь Yes – перехід до запитання 61, відповідь No – вихід з опитувальника.

Запитання №61 – Чи є у QA досвід у автоматизації?: відповідь Yes – перехід до запитання 82, відповідь No – перехід до запитання 64.

Запитання №64 – Чи є можливість виділити розробника для навчання QA?: відповідь Yes – перехід до запитання 82, відповідь No – перехід до запитання 67.

Запитання №67 – Чи є можливість найняти QA з досвідом автоматизації?: відповідь Yes – перехід до запитання 82, відповідь No – перехід до запитання 70.

Запитання №70 – Чи є можливість виділити для QA час на самостійне навчання?: відповідь Yes – перехід до запитання 82, відповідь No – вихід з опитувальника.

Запитання №82 – Чи є на проєкті система безперервної інтеграції?: відповідь Yes – перехід до запитання 83, відповідь No – перехід до запитання 86.

Запитання №83 – Чи є на проєкті DevOps?: відповідь Yes – перехід до запитання 86, відповідь No – перехід до запитання 84.

Запитання №84 – Чи є можливість найняти DevOps?: відповідь Yes – перехід до запитання 86, відповідь No – перехід до запитання 85.

Запитання №85 – Чи є можливість у компанії виділити DevOps на проєкт?: відповідь Yes – перехід до запитання 86, відповідь No – перехід до запитання 86.

Запитання №86 – На якій платформі у вас програма?: відповідь Windows – перехід до запитання 87, відповідь Mac OS – перехід до запитання 91.

Запитання №87 – На якій мові розробляється проєкт?: відповідь C++ – перехід до запитання 88, відповідь .NET – перехід до запитання 90, відповідь Python – перехід до запитання 90.

Запитання №88 – Яка IDE використовується?: відповідь QT – перехід до запитання 89, відповідь MFC – перехід до запитання 89, відповідь QT+MFC – перехід до запитання 90.

Запитання №89 – Чи є можливість використовувати комерціний застосунок для автоматизації?: відповідь Yes – перехід до проміжної форми S8, відповідь No – перехід до запитання 90.

Запитання №90 – Яка мова програмування буде використана під час автоматизації?: відповідь C# – перехід до проміжної форми S9, відповідь Python – перехід до проміжної форми S10, відповідь Java – перехід до проміжної форми S11.

Запитання №91 – Чи є можливість використовувати комерціний застосунок для автоматизації?: відповідь Yes – перехід до проміжної форми S12, відповідь No – перехід до запитання 92.

Проміжна форма S12: PopClip застосунок один з накращих варіантів для застосування під час автоматизації.

Запитання №92 – Яка мова програмування буде використана під час автоматизації?: відповідь Python – перехід до проміжної форми S15, відповідь Ruby – перехід до проміжної форми S14, відповідь Інше– перехід до проміжної форми S13.

Вихід з опитувальника – отримання результату.

Ланцюг – WEB.

Запитання №3 – Яка ваша роль на проєкті?: відповідь Project Manager – перехід до питання 6, відповідь Quality Assurance Engineer – перехід до питання 6, відповідь Software Developer – перехід до питання 6.

Запитання №6 – Ви вирішили ввести автоматизацію на проєкті?: відповідь Yes – перехід до запитання 12, відповідь No – перехід до запитання 9.

Запитання №9 – Ви плануєте ввести автоматизацію у майбутньому?: відповідь Yes – перехід до запитання 12, відповідь No – вихід з опитувальника.

Запитання №12 – Чи готовий розрахунок економічної доцільності введення автоматизації на проєкті?: відповідь Yes – перехід до запитання 41, відповідь No – перехід до питання 15.

Запитання №15 – Чи готові ви зараз створити розрахунок доцільності за допомогою опитувальної системи?: відповідь Yes – перехід до запитання 18, відповідь No – вихід з опитувальника.

Запитання №18 – Чи відома вартість серверів на 1 рік?: відповідь Yes – перехід до запитання 21, відповідь No – перехід до запитання 21.

Запитання №21 – Чи відома вам вартість необхідного апаратного забезпечення?: відповідь Yes – перехід до запитання 24, відповідь No – перехід до запитання 24.

Запитання №24 – Чи відома вам вартість необхідного програмного забезпечення на 1 рік?: відповідь Yes – перехід до запитання 27, відповідь No – перехід до запитання 27.

Запитання №27 – Чи відома вам кількість загальних тест кейсів?: відповідь Yes – перехід до запитання 32, відповідь No – перехід до запитання 32.

Запитання №32 – Чи відомий вам середній час, який потрібен для проходження тест кейса?: відповідь Yes – перехід до запитання 35, відповідь No – перехід до запитання 35.

Запитання №35 – Чи відомий вам час, який потрібен на покриття кейса, автотестом?: відповідь Yes – перехід до запитання 38, відповідь No – перехід до запитання 38.

Запитання №38 – Чи відомо вам скільки разів за рік виконується цей тест кейс?: відповідь Yes – перехід до запитання 41, відповідь No – перехід до запитання 41.

Проміжна форма: Використайте наступну формулу для розрахунку економічної доцільності введення автоматизації

Запитання №41 – Чи отримано дозвіл від Project Manager?: відповідь Yes – перехід до запитання 44, відповідь No – вихід з опитувальника.

Запитання №44 – Чи отримано дозвіл від замовника?: відповідь Yes – перехід до запитання 47, відповідь No – вихід з опитувальника.

Запитання №47 – Чи є незмінною більша частина програми?: відповідь Yes – перехід до запитання 50, відповідь No – перехід до запитання 53.

Запитання №50 – Чи є у вас часові рамки?: відповідь Yes – перехід до запитання 53, відповідь No – перехід до запитання 53.

Запитання №53 – Чи готова підтримка коду зі сторони розробника?: відповідь Yes – перехід до запитання 62, відповідь No – перехід до запитання 56.

Запитання №56 – Чи є можливість виділити розробника на підтримку коду?: відповідь Yes – перехід до запитання 62, відповідь No – перехід до запитання 59.

Запитання №59 – Чи є можливість найняти розробника для підтримки коду?: відповідь Yes – перехід до запитання 62, відповідь No – вихід з опитувальника.

Запитання №62 – Чи є у QA досвід у автоматизації?: відповідь Yes – перехід до запитання 93, відповідь No – перехід до запитання 65.

Запитання №65 – Чи є можливість виділити розробника для навчання QA?: відповідь Yes – перехід до запитання 93, відповідь No – перехід до запитання 68.

Запитання №68 – Чи є можливість найняти QA з досвідом автоматизації?: відповідь Yes – перехід до запитання 93, відповідь No – перехід до запитання 71.

Запитання №71 – Чи є можливість виділити для QA час на самостійне навчання?: відповідь Yes – перехід до запитання 93, відповідь No – вихід з опитувальника.

Запитання №93 – Чи є на проєкті система безперервної інтеграції?: відповідь Yes – перехід до запитання 94, відповідь No – перехід до запитання 96.

Запитання №94 – Чи є на проєкті DevOps?: відповідь Yes – перехід до запитання 97, відповідь No – перехід до запитання 95.

Запитання №95 – Чи є можливість найняти DevOps?: відповідь Yes – перехід до запитання 97, відповідь No – перехід до запитання 96.

Запитання №96 – Чи є можливість у компанії виділити DevOps на проєкт?: відповідь Yes – перехід до запитання 97, відповідь No – перехід до запитання 97.

Запитання №97 – Яка мова програмування буде використана під час автоматизації?: відповідь C# – перехід до проміжної форми S5, відповідь Python – перехід до проміжної форми S16, відповідь Java – перехід до проміжної форми S17, відповідь JavaScript – перехід до проміжної форми S18.

Вихід з опитувальника – отримання результату.

Запитання поділяються на вагові і проміжні – ті, завдяки яким коритувач дістається до вагових. Перелік запитань для гілки Mobile і їх коєфіцієнт наведені у (табл. 3.1). Завдяки цим коєфіцієнтам можна розрахувати ступінь готовності програми до введення автоматизації. Загальна сумма дорівнює 100 відсотків.

Щоб порахувати розрахунок економічної доцільності введення автоматизації на проєкті – потрібно вкиористати наступну формулу:

$$1QA - (m \cdot t) \cdot z + v = 1automation\ QA - (m \cdot T) \cdot z + a + b + c + V,$$

де m – кількість загальних тест кейсів;

t – середній час, який потрібен для проходження тест кейса;

z – скільки разів за рік виконується цей тест кейс;

T – час, який потрібен на покриття кейса, автотестом;

a – вартість серверів на 1 рік;

b – вартість необхідного апаратного забезпечення;

c – вартість необхідного програмного забезпечення на 1 рік;

v – заробітна плата QA;

V – заробітна плата automation QA.

Таблиця 3.1 – Запитання, що мають вагові коефіцієнти

Запитання №13 – Чи готовий розрахунок економічної доцільності введення автоматизації на проєкті?	Yes – 28 \ No – 0
Запитання №19 – Чи відома вартість серверів на 1 рік?	Yes – 6 \ No – 0
Запитання №22 – Чи відома вам вартість необхідного апаратного забезпечення?	Yes – 6 \ No – 0
Запитання №25 – Чи відома вам вартість необхідного програмного забезпечення на 1 рік?	Yes – 6 \ No – 0
Запитання №28 – Чи відома вам кількість загальних тест кейсів?	Yes – 2 \ No – 0
Запитання №33 – Чи відомий вам середній час, який потрібен для проходження тест кейса?	Yes – 1,5 \ No – 0
Запитання №36 – Чи відомий вам час, який потрібен на покриття кейса, автотестом?	Yes – 3,5 \ No – 0

Кінець таблиці 3.1

Запитання №39 – Чи відомо вам скільки разів за рік виконується цей тест кейс?	Yes – 3 \ No – 0
Запитання №48 – Чи є незмінною більша частина програми?	Yes – 20 \ No – 3,5
Запитання №54 – Чи готова підтримка коду зі сторони розробника?	Yes – 15 \ No – 0
Запитання №57 – Чи є можливість виділити розробника на підтримку коду?	Yes – 10 \ No – 0
Запитання №60 – Чи є можливість найняти розробника для підтримки коду?	Yes –5 \ No – 0
Запитання №63 – Чи є у QA досвід у автоматизації?	Yes – 15 \ No – 0
Запитання №66 – Чи є можливість виділити розробника для навчання QA?	Yes – 10 \ No – 0
Запитання №69 – Чи є можливість найняти QA з досвідом автоматизації?	Yes – 5 \ No – 0
Запитання №72 – Чи є можливість виділити для QA час на самостійне навчання?	Yes – 2,5 \ No – 0
Запитання №73 – Чи є на проєкті система безперервної інтеграції?	Yes – 11 \ No – 0
Запитання №74 – Чи є на проєкті DevOps?	Yes – 11 \ No – 0
Запитання №75 – Чи є можливість найняти DevOps?	Yes – 5 \ No – 0
Запитання №76 – Чи є можливість у компанії виділити DevOps на проєкт?	Yes – 2,5 \ No – 0
	Загальний відсоток = 100%

3.3 Тестування розробленої інформаційної системи та аналіз результатів

Щоб протестувати розроблену інформаційну системи, було запропоновано пройти цей опитувальник двом розробникам, двом тестувальникам і менеджеру проєкта. Кожний з них працює у різних компаніях, на різних проєктах, які включають в себе мобільні, веб- і десктопні застосунки.

1. Middle Automation QA Engineer, мобільний застосунок.

Пройшовши опитування за наступним ланцюгом: Запитання №1 – Запитання №4 – Запитання №7 – Запитання №10 – Запитання №13 – Запитання №16 – Запитання №19 – Запитання №22 – Запитання №25 – Запитання №28 – Запитання №29 – Запитання №33 – Запитання №36 – Запитання №39 – Запитання №42 – Запитання №45 – Запитання №48 – Запитання №51 – Запитання №54 – Запитання №57 – Запитання №60 – Запитання №63 – Запитання №73 – Запитання №74 – Запитання №75 – Запитання №76 – Запитання №77 – Запитання №79 – Запитання №81 – Форма S6 – Вихід з опитувальника.

Був отриманий наступний результат – 72% (рис. 3.6), готовність проєкта до введення автоматизації. Вагові коефіцієнти розписані у (табл. 3.2).

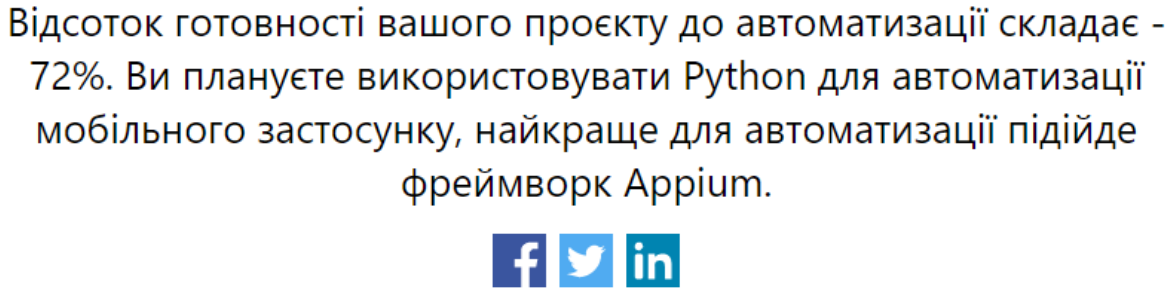

Рисунок 3.6 – Отриманий результат під час тесту №1

Таблиця 3.2 – Результат вагових коефіцієнтів тесту №1

Запитання №13 – Чи готовий розрахунок економічної доцільності введення автоматизації на проєкті?	No – 0
Запитання №19 – Чи відома вартість серверів на 1 рік?	Yes – 6
Запитання №22 – Чи відома вам вартість необхідного апаратного забезпечення?	No – 0
Запитання №25 – Чи відома вам вартість необхідного програмного забезпечення на 1 рік?	Yes – 6
Запитання №28 – Чи відома вам кількість загальних тест кейсів?	Yes – 2
Запитання №33 – Чи відомий вам середній час, який потрібен для проходження тест кейса?	Yes – 1,5
Запитання №36 – Чи відомий вам час, який потрібен на покриття кейса, автотестом?	No – 0
Запитання №39 – Чи відомо вам скільки разів за рік виконується цей тест кейс?	Yes – 3
Запитання №48 – Чи є незмінною більша частина програми?	Yes – 20
Запитання №54 – Чи готова підтримка коду зі сторони розробника?	No – 0
Запитання №57 – Чи є можливість виділити розробника на підтримку коду?	No – 0
Запитання №60 – Чи є можливість найняти розробника для підтримки коду?	Yes –5
Запитання №63 – Чи є у QA досвід у автоматизації?	Yes – 15
Запитання №73 – Чи є на проєкті система безперервної інтеграції?	Yes – 11
Запитання №74 – Чи є на проєкті DevOps?	No – 0
Запитання №75 – Чи є можливість найняти DevOps?	No – 0
Запитання №76 – Чи є можливість у компанії виділити DevOps на проєкт?	Yes – 2,5
	72%

2. Junior Project Manager, мобільний застосунок.

Пройшовши опитування за наступним ланцюгом: Запитання №1 – Запитання №4 – Запитання №7 – Запитання №13 – Запитання №42 – Запитання №45 – Запитання №48 – Запитання №54 – Запитання №57 – Запитання №63 – Запитання №66 – Запитання №69 – Запитання №72 – Запитання №73 – Запитання №77 – Запитання №79 – Запитання №81 – Форма S5 – Вихід з опитувальника.

Був отриманий наступний результат – 52.5% (рис. 3.7), готовність проєкта до введення автоматизації. Вагові коефіцієнти розписані у (табл. 3.3).

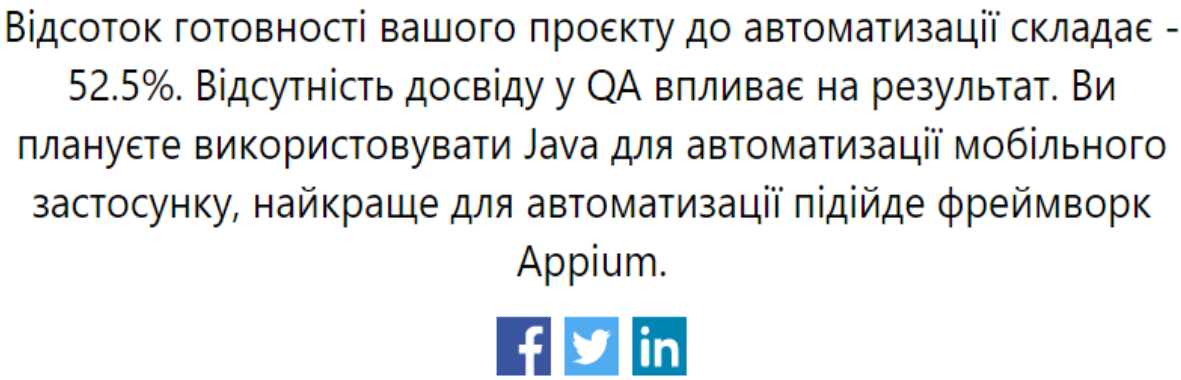

Рисунок 3.7 – Отриманий результат під час тесту №2

Таблиця 3.3 – Результат вагових коефіцієнтів тесту №2

Запитання №13 – Чи готовий розрахунок економічної доцільності введення автоматизації на проєкті?	No – 20
Запитання №48 – Чи є незмінною більша частина програми?	No – 20
Запитання №54 – Чи готова підтримка коду зі сторони розробника?	No – 0
Запитання №57 – Чи є можливість виділити розробника на підтримку коду?	Yes –10

Кінець таблиці 3.3

Запитання №63 – Чи є у QA досвід у автоматизації?	No – 0
Запитання №66 – Чи є можливість виділити розробника для навчання QA?	No – 0
Запитання №69 – Чи є можливість найняти QA з досвідом автоматизації?	No – 0
Запитання №72 – Чи є можливість виділити для QA час на самостійне навчання?	Yes – 2,5
Запитання №73 – Чи є на проєкті система безперервної інтеграції?	No – 0
	52,5%

3. Senior Project Manager, десктоп застосунок.

Пройшовши опитування за наступним ланцюгом: Запитання №1 – Запитання №2 – Запитання №5 – Запитання №8 – Запитання №11 – Запитання №40 – Запитання №43 – Запитання №46 – Запитання №49 – Запитання №52 – Запитання №61 – Запитання №82 – Запитання №83 – Запитання №86 – Запитання №87 – Запитання №88 – Запитання №89 – Форма S8 – Вихід з опитувальника.

Був отриманий наступний результат – 100% (рис. 3.8), готовність проєкта до введення автоматизації. Вагові коефіцієнти розписані у (табл. 3.4).

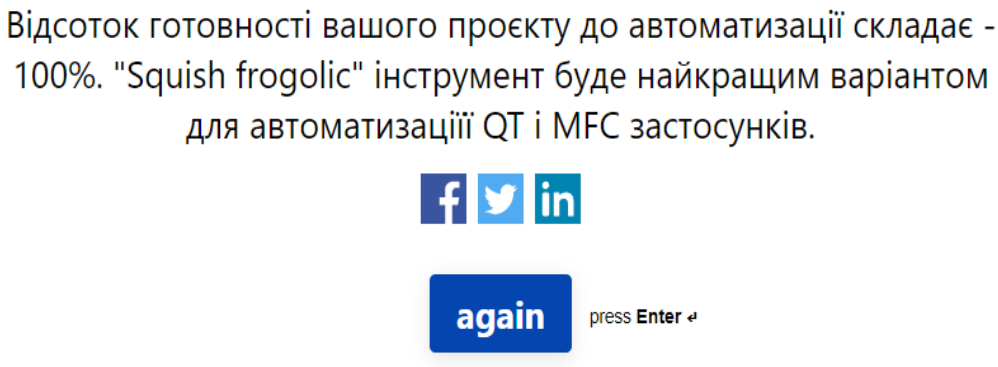

Рисунок 3.8 – Отриманий результат під час тесту №3

Таблиця 3.4 – Результат вагових коефіцієнтів тесту №3

Запитання №13 – Чи готовий розрахунок економічної доцільності введення автоматизації на проєкті?	Yes – 28
Запитання №48 – Чи є незмінною більша частина програми?	Yes – 20
Запитання №54 – Чи готова підтримка коду зі сторони розробника?	Yes – 15
Запитання №63 – Чи є у QA досвід у автоматизації?	Yes – 15
Запитання №73 – Чи є на проєкті система безперервної інтеграції?	Yes – 11
Запитання №74 – Чи є на проєкті DevOps?	Yes – 11
	100%

4. Middle Software Developer, вебзастосунок.

Пройшовши опитування за наступним ланцюгом: Запитання №1 – Запитання №3 – Запитання №6 – Запитання №9 – Запитання №12 – Запитання №41 – Запитання №44 – Вихід з опитувальника.

Був отриманий наступний результат – через відсутність дозволу від замовника, введення автоматизації неможливе (рис. 3.9).

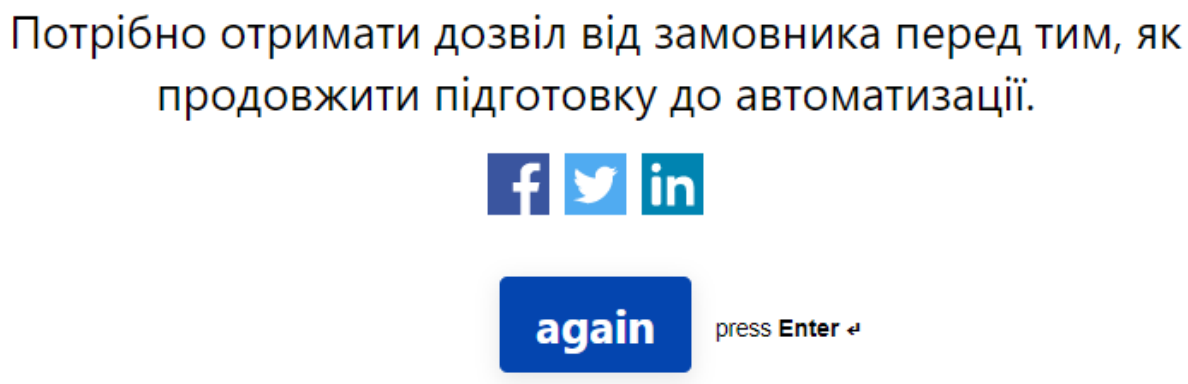

Рисунок 3.9 – Отриманий результат під час тесту №4

Провівши аналіз отриманих результатів і порівнявши їх з ситуацією на проєктах, 3 з 4 тестів виявились вдалими і відповідали стану справ на проєктах. Тест №4 виявився невдалим, через те, що остаточне рішення про введення автоматизації приймає менеджер. Після тестування цієї системи було виявлено, що у 75% випадків система показує правильний результат.

3.4 Перспективи подальшої роботи

Під час виконання цієї роботи, було проведено аналіз методів системного аналізу, які були використані під час спілкування з експертами зі сфери ІТ. Під час спілкування була проведена бесіда з розробниками, тестувальниками, менеджерами. Ці експерти працюють у різних компаніях, на різних проектах і з різними технологіями.

Була проведена презентація розробленної опитувальної системи, пояснені можливості системи, її задачі і перспективи. Після презентаціїї, кожен з експертів протестував систему і зробив висновок, а також вніс пропозиції по вдосконаленю цієї системи.

Був зроблений аналіз відгуків і пропозицій, було визначено пріорітетні напрямки розвитку цієї системи, такі як – збільшення бази знань системи, зміна системи для розробки опитувальних системи на самостійний застосунок, показ усіх пройдених запитань з відповідями, після закінчення опитування.

Після вдосконалення системи, вона може використовуватись у ІТ-компаніях менеджерами проєктів з метою збору і оцінки інформації про можливість введення автомтиазації.

У подальшому планується продовжити вдосконалювати систему, з метою її подальшого впровадження у ІТ-компаніях.

ВИСНОВКИ

У рамках кваліфікаційної роботи була розроблена і реалізована модель інформаційної системи визначення рівня готовності програмного забезпечення до автоматизованого тестування.

Під час розробки було проведено дослідження сучасних інформаційних систем і їх теперешній стан. Під час дослідження було виявлено патерни, за якими проходить розробка інформаційних систем.

Проведено аналіз особливостей баз даних і їх компонентів, здійснено дослідження найбільш поширених баз даних на сьогоднішній день.

Щоб наповнити базу знань, проаналізовано і визначено методи системного аналізу, які в подальшому були використані під час спілкування з експертами.

Під час розробки структури використано знання експертів у сфері розробки програмного забезпечення.

Структура і наповнення реалізовано у системі, яка допомогає розробляти опитувальні системи.

Розробивши і наповнивши систему запитаннями, її протестовано за допомогою працівників різних IT-компаній, які пройшли систему спираючись на проекти, на яких вони працюють, а також дали свої відгуки.

Завдяки невеликій IT-компанії, яка дозволила запровадити цю систему у себе, отримано відгуки від людей, які використовували систему під час своєї роботи, це допомогло виділити основні напрямки, у яких варто розвивати систему.

Мету роботи досягнуто.

Результати роботи апробовано у вигляді тез доповіді під час Міжнародного молодіжного форуму «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У XXI СТОЛІТТІ» [31].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Джозефа Джарратано, Гари Райли (2006) Експертні системи: принципи розробки та програмування, 4-е видання», 2006. 1152 с.

2. Питер Джексон (2001) Структурный анализ и интеллектуальная обработка данных в компьютерном зрении, 624 с.

3. Іван Братко (2004) «Алгоритмы искусственного интеллекта на языке PROLOG», 640 с.

4. Джордж Ф. Люгер (2016) «Искусственный интеллект. Стратегии и методы решения сложных проблем», 864 с.

5. Филиппович Ю.Н., Филиппович А.Ю., (2003) «Экспертные системы», 312 с.

6. Субботінім С.О., (2008) «Подання і обробка знань у системах штучного інтелекту та підтримки прийняття рішень», 332 с.

7. Владимир Евменов (2016) Интеллектуальные системы управления. Превосходство искусственного интеллекта над естественным интеллектом?, 304 с.

8. Роман Душкін (2020) Интелектуальные транспортные системы, 280 с.

9. Alex F. Osborn. Applied imagination principles and procedures of creative thinking, 1953. 317c.

10. M. Ayaz Ahmad, Irina Tvoroshenko, Jalal Hasan Baker, and Vyacheslav Lyashenko (2019) Modeling the Structure of Intellectual Means of Decision-Making Using a System-Oriented NFO Approach, *International Journal of Emerging Trends in Engineering Research*, 7(11), pp. 460-465.

11. Yousef Ibrahim Daradkeh, and Iryna Tvoroshenko (2020) Application of an Improved Formal Model of the Hybrid Development of Ontologies in Complex Information Systems, *Applied Sciences*, 10(19). p. 6777.

12. Matarneh Rami, Tvoroshenko Irina, and Lyashenko Vyacheslav (2019) Improving Fuzzy Network Models For the Analysis of Dynamic Interacting Processes in the State Space, International Journal of Recent Technology and Engineering, 8(4), pp. 1687-1693.

13. Tvoroshenko I.S., and Kramarenko O.O. (2019) Software determination of the optimal route by geoinformation technologies, *Radio Electronics Computer Science Control*, 3, pp. 131-142.

14. Tvoroshenko, I. S. (2004) Structure and functions of intelligent decision-making tools in complex systems. *Artificial Intelligence*, *4*, 462-470

15. Творошенко І.С. (2018) Особливості застосування сучасних принципів штучного інтелекту до розробки ефективних механізмів моделювання складних систем. *Science and Technology of the Present Time: Priority Development Directions of Ukraine and Poland*: International Multidisciplinary Conference (Wolomin, Republic of Poland, 19–20 October 2018). Wolomin: Izdevnieciba «Baltija Publishing». Volume 4. pp. 118-121.

16. Tvoroshenko Iryna (2019) Development of models of spatial analysis of status of interactive processes of complex systems. International Scientific Conference *Scientific Development of New Eastern Europe*: Conference Proceedings (Riga, April 6th, 2019). Latvia: Baltija Publishing. Part II. pp. 152-156.

17. Творошенко І.С. Технології прийняття рішень в інформаційних системах: навч. посібник. Харків: ХНУРЕ, 2021. 120 с.

18. Гороховатський В.О., Творошенко І.С. Методи інтелектуального аналізу та оброблення даних: навч. посібник. Харків: ХНУРЕ, 2021. 92 с.

19. Творошенко, И. С. (2010). Анализ процессов принятия решений в интеллектуальных системах. *Системи обробки інформації*, (2), 248-253.

20. Kucherenko, Y. I., Filatov, V. A., Tvoroshenko, I. S., & Baidan, R. N. (2005). Intellectual technologies in decision-making technological complexes based on fuzzy interval logic. *East European Journal of Advanced Technologies*, *2*, 92-96.

21. Кучеренко, Е. И., Творошенко, И. С. (2003) Процессы принятия решений в сложных системах на основе нечетких интервальных представлений. *Вісник Національного технічного університету «ХПІ». Тематичний випуск: Системний аналіз, управління та інформаційні технології. – Х.: НТУ «ХПІ», 1*(7), 79-86.

22. Творошенко І.С., Федченко М.А. До питання обробки колективних рішень за допомогою програмного застосунку виборчої системи. *Сучасний рух науки*: тези доповідей VII міжнародної науково-практичної інтернет-конференції (Дніпро, 6–7 червня 2019 р.). Дніпро, 2019. С. 1664-1669.

23. I. Tvoroshenko Information technologies for decision-making on the conditions of spatially distributed objects, in Abstracts of I International Scientific and Practical Conference. Problems and perspectives of modern science and practice, Austria, 2020. pp. 45-50.

24. Творошенко, И. С., Дехтярь, А. П. (2005, June) Информационные технологии в задачах компьютерной диагностики с использованием интеллектуальных систем. In *Клиническая информатика и Телемедицина. Компьютерная Медицина–2005: материалы междунар. научн.-технич. конф., Харьков* (p. 138).

25. Кучеренко, Е. И., Творошенко, И. С. (2010) Прикладные аспекты моделирования нечетких процессов в сложных системах. *Збірник наукових праць Харківського університету Повітряних сил*, (1), 127-131.

26. Кучеренко, Е. И., Корниловский, А. В., Творошенко, И. С. (2010) О методах настройки функций принадлежности в нечетких системах. *Системы управления, навигации и связи*, (1), 13.

27. Daradkeh Y.I., and Tvoroshenko I. (2020) Technologies for Making Reliable Decisions on a Variety of Effective Factors using Fuzzy Logic, *International Journal of Advanced Computer Science and Applications*, 11(5), pp. 43-50.

28. Ahmad M. Ayaz, Tvoroshenko Irina, Baker Jalal Hasan, and Lyashenko Vyacheslav (2019) Computational Complexity of the Accessory Function Setting Mechanism in Fuzzy Intellectual Systems, *International Journal of Advanced Trends in Computer Science and Engineering*, 8(5), pp. 2370-2377.

29. Tvoroshenko I.S., and Gorokhovatsky V.O. (2019) Modification of the branch and bound method to determine the extremes of membership functions in fuzzy intelligent systems, Telecommunications and Radio Engineering, 78(20), pp. 1857-1868.

30. Asaad Ma. Babker, Abd Elgadir A. Altoum, Irina Tvoroshenko, and Vyacheslav Lyashenko (2019) Information Technologies of the Processing of the Spaces of the States of a Complex Biophysical Object in the Intellectual Medical System HEALTH, *International Journal of Advanced Trends in Computer Science and Engineering*, 8(6), pp. 3221-3227.

31. Полубєхін А.А. Про особливості оброблення даних для подальшого моделювання конфігурації організаційної системи. *Радіоелектроніка та молодь у XXI столітті*: тези доповідей 25-го Міжнародного молодіжного форуму (Харків, 20–21 квітня 2021 р.). Харків: ХНУРЕ, 2021. Т. 7, 10. С. 40-41.