

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
(повна назва)

Кафедра _____ Медіасистем та технологій _____
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____ другий (магістерський) _____
(рівень вищої освіти)

_____ Дослідження засобів автоматизації при заповненні користувацьких форм _____
(тема)

Виконав:
студент 2 курсу, групи ТЕМВм-19-1

_____ Алєєв А.Р. _____

Спеціальності _____ 186 Видавництво та поліграфія _____

Тип програми _____ Освітньо-професійна _____

Освітня програма
_____ Технології електронних мультимедійних видань _____

Керівник _____ доц. Дейнеко Ж.В. _____

Допускається до захисту
Зав. кафедри МСТ

(підпис)

_____ Ткаченко В.П. _____
(прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
Кафедра _____ Медіасистем та технологій _____
Рівень вищої освіти _____ другий (магістерський) _____
Спеціальність _____ 186 Видавництво та поліграфія _____
Тип програми _____ Освітньо-професійна _____
Освітня програма _____ Технології електронних мультимедійних видань _____
(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри МСТ _____
(підпис)

« 26 » жовтня 2020 р.

**ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ**

студентові _____ Алєєву Артему Ренатовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження засобів автоматизації при заповненні користувацьких форм

Затверджена наказом по університету від 23 жовтня 2020р. № 1432 Ст

2. Термін подання студентом роботи 3 грудня 2020 р.

3. Вихідні дані до роботи

Національні та міжнародні стандарти оцінки якості користувацьких форм;
методи та принципи автоматичного заповнення інтерфейсу користувача.

4. Зміст пояснювальної записки (перелік питань, що їх належить розробити)

Огляд сучасних засобів та інструментів для оцінки користувацьких форм; Методи та критерії оцінки користувацьких форм; Опис та проблематика предметної області; Мета роботи та постановка задачі; Планування дослідження; Аналіз критеріїв оцінки користувацьких форм; Планування експерименту і обробки експериментальних даних; Аналіз користувацьких форм; Приклади користувацьких форм; Методи автоматичного заповнення; користувацьких форм; Експериментальна частина; Статистика підпису полів та атрибуту page; Розрахунки для ручного введення даних у форму; Реалізація та застосування методу автоматичного заповнення; Порівняльний аналіз ручного та автоматичного вводу інформації; Економічна частина; Оформлення пояснювальної записки; Оформлення графічної частини; Висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, плакатів)

Слайди: Актуальність та мета роботи; Задачі; Дослідження стану проблеми; Опис предметної області; Вибір та обґрунтування критеріїв оцінки якості користувацької форми; Експериментальна частина; Обробка та аналіз результатів експерименту; Розробка методики автоматичного заповнення користувацької форми; Оцінка ефективності розробленої методики; Економічна частина; Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	доц. Дейнеко Ж.В.		
Економічна частина	проф. Полозова Т.В.		28.11.2020

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд сучасних засобів та інструментів для Оцінки користувацьких форм	1.11.2020	
2	Планування дослідження	3.11.2020	
3	Аналіз користувацьких форм	5.04.2020	
4	Методи автоматичного заповнення користувацьких форм	9.11.2020	
5	Експериментальна частина	12.11.2020	
6	Економічна частина	19.11.2020	
7	Оформлення пояснювальної записки	22.11.2020	
8	Оформлення графічної частини	24.11.2020	

Дата видачі завдання 26 жовтня 2020 р.

Студент _____ Алєєв А.Р.
(підпис)

Керівник роботи _____ доц. Дейнеко Ж.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить 63 сторінки, 19 рисунків, 7 таблиць, 21 використане літературне джерело, 1 додаток.

КОРИСТУВАЦЬКА ФОРМА, ФОРМА РЕЄСТРАЦІЇ, АВТОМАТИЧНЕ ЗАПОВНЕННЯ, АВТОМАТИЗАЦІЯ, WEB-ФОРМА, СТАТИСТИЧНИЙ АНАЛІЗ, СТАТИСТИКА, ПАРСИНГ.

Багато сучасних web-сайтів використовують для організації форм опитування своїх респондентів або майбутніх клієнтів, для оформлення бланків заказу, для анкетування, користувацькі форми. Заповнення цих форм потребує багато часу. Магістерська атестаційна робота присвячена дослідженню сучасних засобів автоматизації заповнення користувацьких форм та розробленню методики власноручного заповнення полів форми з метою зменшення часу та кількості помилок.

В ході роботи були проаналізовані методи обробки даних, різні формати вводу та збереження текстових блоків, отриманих з користувацьких форм. У роботі було використано статистичний аналіз наявних різноманітних користувацьких форм, які застосовуються на сучасних web-сторінках юридичних, громадських та приватних установ. Виконано необхідні економічні розрахунки, що обґрунтовують кошторис доцільності впровадження запропонованої методики.

У результаті досліджень було запропоновано методику обробки користувацьких форм на web-сторінках, що дає змогу зменшити час, витрачений на заповнення цієї форми та виключили помилки заповнення.

РЕФЕРАТ

Пояснительная записка содержит 63 страницы, 19 рисунков, 7 таблиц, 21 использованный литературный источник, 1 приложение.

ПОЛЬЗОВАТЕЛЬСКИЕ ФОРМЫ, ФОРМЫ РЕГИСТРАЦИИ, АВТОМАТИЧЕСКОЕ ЗАПОЛНЕНИЕ, АВТОМАТИЗАЦИЯ WEB-ФОРМА, СТАТИСТИЧЕСКИЙ АНАЛИЗ, СТАТИСТИКА, ПАРСИНГ.

Многие современные web-сайтов используют для организации форм опроса своих респондентов или будущих клиентов, для оформления бланков заказа, для анкетирования, пользовательские формы. Заполнение этих форм требует много времени. Магистерская аттестационная работа посвящена исследованию современных средств автоматизации заполнения пользовательских форм и разработке методики собственноручной заполнения полей формы с целью уменьшения времени и количества ошибок.

В ходе работы были проанализированы методы обработки данных, различные форматы ввода и сохранения текстовых блоков, полученных из пользовательских форм. В работе были использованы статистический анализ имеющихся различных пользовательских форм, применяемых на современных web-страницах юридических, общественных и частных учреждений. Выполнен необходимые экономические расчеты, обосновывающие смету целесообразности внедрения предложенной методики.

В результате исследований было предложена методика обработки пользовательских форм на web-страницах, что позволяет уменьшить время, затраченное на заполнение этой формы и исключили ошибки заполнения.

ABSTRACT

The explanatory note contains 63 pages, 19 pictures, 7 tables, 21 sources, 1 application.

USER FORM, REGISTRATION FORM, AUTOMATIC FILLING, AUTOMATION, WEB-FORM, STATISTICAL ANALYSIS, STATISTICS, PARSING.

Many modern websites use for the organization of forms of interrogation of the respondents or future clients, for registration of forms of the order, for questionnaires, user forms. Completing these forms takes a long time. The master's attestation work is devoted to the research of modern means of automation of filling in customs forms and the development of methods for self-filling form fields in order to reduce the time and number of errors.

In the course of the work, the methods of data processing, different formats of input, and saving of text blocks obtained from user forms were analyzed. The paper used a statistical analysis of the various user forms available on modern web-pages of legal, public, and private institutions. Necessary economic calculations have been performed, which substantiate the estimate of expediency of the introduction of the offered technique.

As a result of research, a method of processing custom forms on web-pages was proposed, which allows reducing the time spent on filling out this form and eliminate filling errors.

ЗМІСТ

	С.
ВСТУП	9
1 ОГЛЯД СУЧАСНИХ ЗАСОБІВ ТА ІНСТРУМЕНТІВ ДЛЯ ОЦІНКИ КОРИСТУВАЦЬКИХ ФОРМ	12
1.1 Оцінка користувацьких форм	12
1.2 Методи та критерії оцінки користувацьких форм	13
1.2.1 GOMS-аналіз юзабіліті інтерфейсу	14
1.2.2 Аналіз XML-дерева	15
1.2.3 Методи експертної оцінки	17
1.2.4 Автоматизація заповнення форм	20
1.3 Опис та проблематика предметної області	21
1.4 Відмінності HTML5 від HTML4 та вплив на користувацькі форми	24
1.5 Мета роботи та постановка задач дослідження	25
2 ПЛАНУВАННЯ ДОСЛІДЖЕННЯ	27
2.1 Аналіз критеріїв оцінки користувацьких форм	27
2.2 Розробка методики проведення експерименту	27
2.3 План проведення експерименту і обробки експериментальних даних	28
3 АНАЛІЗ КОРИСТУВАЦЬКИХ ФОРМ	30
3.1 Приклади надлишкових користувацьких форм	30
3.2 Приклади компактних користувацьких форм	33
4 МЕТОДИ АВТОМАТИЧНОГО ЗАПОВНЕННЯ КОРИСТУВАЦЬКИХ ФОРМ	36
4.1 Парсинг полів по типу	36
4.2 Парсинг DOM дерева	37
4.3 Парсинг полів за атрибутом name	38
5 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА	40
5.1 Парсинг DOM дерева з додатковими параметрами	40
5.2 Способи зберігання користувацьких даних	41

5.2.1 ТХТ файл як сховище приватної інформації	42
5.2.2 Використання у якості сховища XML формат	42
5.2.3 Використання у якості сховища JSON формат	43
5.2.4 Порівняння варіантів сховища	44
5.3 Статистика підпису полів та атрибуту name	44
5.4 Способи автоматичного введення користувацьких даних у форму	46
5.5 Вибір форми для тестування.....	47
5.6 Розрахунки для ручного введення даних у форму	48
5.7 Реалізація та застосування методу автоматичного заповнення	49
5.8 Порівняльний аналіз автоматичного та ручного введення.....	49
6 ЕКОНОМІЧНА ЧАСТИНА	51
6.1 Характеристика науково-дослідної роботи.....	51
6.2 Етапи виконання НДР, їх трудомісткість та заробітна плата.....	52
6.3 Розрахунок одноразових витрат на розробку НДР.....	55
6.4 Оцінка результатів науково-дослідної роботи.....	58
ВИСНОВКИ.....	61
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	62
ДОДАТОК А Код на мові Javascript	64

ВСТУП

На теперішній час все частіше можна почути думку, що пандемія назавжди змінить наше життя у всіх його сферах. І це стосується не тільки економіки, а й ставить під загрозу практично усі сфери нашого життя та ведення бізнесу. Ситуація, що склалася на сьогодні та змусила людей і бізнес сидіти по домівках, явно показує, що бізнес, не представлений в Інтернеті, має величезні ризики в своїй роботі [1].

На сьогоднішній день багато бізнесу переходить в онлайн-режим, відповідно все більше послуг можна здобувати та надавати не виходячи з дому. Такому переходу посприяла пандемія, під час якої 98 % off-line бізнесу не може повноцінно функціонувати, тому в деяких випадках подібний перехід не можливо здійснити швидко, тому що необхідно перестроювати організацію самого бізнесу, але й форми взаємодії із користувачами, змінювати форми спілкування із майбутніми клієнтами.

З моменту заснування Інтернету як загального явища, користувачі почали створювати власні користувацькі сторінки, щоб взаємодіяти один з одним та ділитися інформацією. З ходом розвитку сайти ставали більшими та складнішими, варіанти взаємодії зростали, в свою чергу сайти починали заробляти більше інформації. За весь цей час методи вводу ніяк не змінилися, але форми змінились. Наразі великі сайти можуть вимагати введення великої кількості приватної інформації, що ускладнює взаємодію та вимагає часу та уваги на заповнення.

Під час розроблення сайтів, насамперед видимої його частини – інтерфейса користувача – постає необхідність швидкого, але водночас зваженого вибору способу організації «людино-машинної» та «машино-людинної» взаємодії. Діалог – оперативна взаємодія користувача з комп'ютером у процесі розв'язання задачі – відіграє значну, а іноді і переважну роль в автоматизації проектування. Режим діалогу дає змогу

розширити коло задач, розв'язуваних системами автоматизованого проектування [2].

Користувацька форма – це основний модуль збору особистої інформації, який заповнюється безпосередньо користувачем, шляхом прямого вводу інформації у поля форми. Зазвичай такі форми використовуються під час реєстрації на сайті або сервісі, та під час опитувань.

Користувацькі форми використовуються також у роботі в різних додатках та на різних платформах. На сьогодні неможливо уявити собі роботу з особистими даними без подібних форм. Також можна розглядати користувацькі форми під час процесу вводу для повідомлення в месенджері або соцмережі, у контексті чатингу, але в поточній темі такий підтип розглядатися не буде.

Протягом останнього часу, багато існуючих компаній для взаємодії із потенційними клієнтами використовують різні форми онлайн-спілкування, поширюють можливості Інтернет-ресурсів. І сьогоднішня ситуація не виняток, а тільки підтвердження того факту, що і надалі зростатиме перехід до онлайн спілкування, бо як виявилось велика кількість бізнесу може ефективно функціонувати взагалі без офлайн складової, тим самим зменшивши витрати оренди та комунальні послуги, що в свою чергу, в майбутньому призведе до виникнення підприємств, які повністю перебуватимуть в онлайн режимі.

Саме тому, темою дипломної роботи є дослідження засобів автоматизації при заповненні користувацьких форм. Через те, що подібні форми не є уніфікованими на 100 %, різним сервісам необхідна різна інформація про користувача, наразі не можна створити єдиний шаблон ґрунтуючись на якому можна буде заповнювати інформацію автоматично. Виникає необхідність заповнювати користувацькі форми автоматично або частково автоматично з метою зменшення кількості помилок, економії часу та нервових клітин, бо не тільки під час реєстрації на ресурсах необхідно взаємодіяти з формами, але й під час роботи з тими чи іншими ресурсами.

Також, дуже важливо звернути увагу на категорію людей з обмеженими можливостями, у яких можуть виникнути проблеми під час заповнення користувацьких форм власноруч, посимвольно вводячи інформацію з клавіатури.

Автоматичне заповнення користувацьких форм може знадобитися будь-кому: при заповненні бланку-заяви на подорож до якоїсь країни, при заповненні власної інформації на різноманітних сайтах, від сайтів знайомств до державних муніципальних ресурсів, при отриманні реєстрів надання послуг населенню, при відповіді на питання анкети. І такі дії, пов'язані з заповненням користувацьких форм, у деяких випадках, є необхідними і обов'язковими, кількість полів на цих формах є достатньо великим і доволі специфічним.

1 ОГЛЯД СУЧАСНИХ ЗАСОБІВ ТА ІНСТРУМЕНТІВ ДЛЯ ОЦІНКИ КОРИСТУВАЦЬКИХ ФОРМ

Впродовж усього існування Інтернету у користувацьких формах змінювалась кількість полів, їх складність та способи комбінації [3]. Протягом того, як розвивалися Web-технології почали й відповідно змінюватись користувацькі форми, почали братись до уваги положення полів один від одного, розміри та в цілому було зроблено велику роботу для того, щоб заповнювати форми було зручно.

Користувацькі форми оцінюються у двох випадках:

- на етапі проектування веб-інтерфейсу;
- коли інтерфейс вже створено, та необхідно оцінити його ефективність.

Але цим займаються далеко не всі веб-сайти, деякі структури, такі як державні не дотримуються сучасних тенденцій, а виходячи з специфічних вимог запитують велику кількість приватної інформації у користувачів. Здебільшого це стосується комерційного середовища, де в першу чергу йде бізнес-логіка додатку та технічних сайтів.

1.1 Оцінка користувацьких форм

Оцінка користувацьких форм може знадобитись для того, щоб зрозуміти наскільки вдалою з точки зору положення полів відносно один одного та час що необхідний на коректне заповнення всіх полів форми.

Завдяки оцінці можна отримати таку інформацію як:

- скільки необхідно витратити часу на заповнення;
- кількість необхідних до заповнення полів;
- взаємозв'язок між положенням полів та часом на їх заповнення;
- наскільки зручно заповнювати форму;

- реакція на помилки користувача та необхідність до повторного заповнення полів;
- складні місця для заповнення користувачем.

Володіючи цією інформацією можна використати її для покращення форм, шляхом їх зміни, та для отримання детальної аналітики бо такий елемент може впливати на показники продажів або в цілому популярності Інтернет ресурсу через те, що користуватися таким ресурсом просто неможливо.

Завдяки отриманим даним можна виконати порівняння з новою формою або методами, які були застосовані для прискорення заповнення користувацької форми [4].

Саме поєднання усіх цих методів, дозволить отримати більш розширену аналітичну інформацію, щодо часу, який необхідно витратити на заповнення користувацької форми, зручності для користувача, та багато іншої інформації, що може знадобитися в майбутньому.

1.2 Методи та критерії оцінки користувацьких форм

Оцінка здійснюється на всіх етапах життєвого циклу програмного забезпечення, але в рамках поточної теми, буде здійснюватись оцінка виключно для вже існуючих користувацьких форм без змоги їх хоч якось модифікувати.

Критерії оцінки включають в себе:

- кількість полів;
- види полів;
- відношення кількості полів до їх типу;
- час що необхідний на заповнення форми;
- час який необхідно витратити на заповнення зі здійсненням помилок при згідно існуючої валідації полів;
- статистика щодо співпадіння полів в різних формах.

Ґрунтуючись на отриманих результатах можна отримати детальну інформацію для подальшої обробки та виведення відповідних коефіцієнтів, завдяки чому можна порівняти різні форми між собою та ефективність обраних для заповнення методів.

1.2.1 GOMS-аналіз юзабіліті інтерфейсу

Цей метод дозволяє виміряти скільки часу необхідно витратити досвідченому користувачеві на виконання певної дії у конкретному інтерфейсі. Розраховується він наступним чином, усі дії користувача записуються в послідовності їх виконання, з різними сценаріями завершення дії (рис. 1.1).

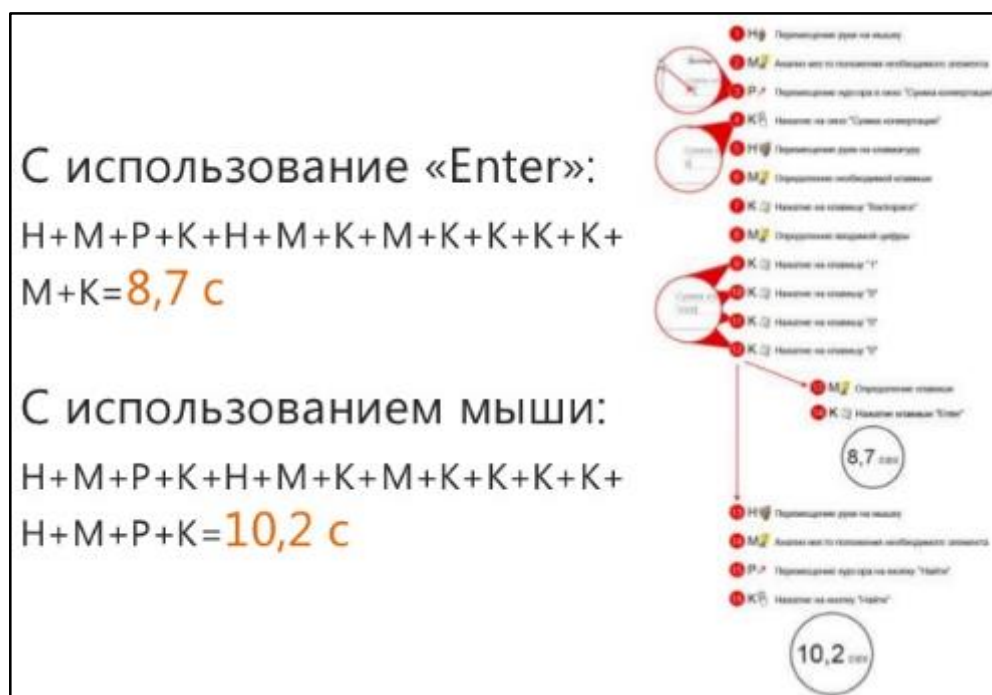


Рисунок 1.1 – Розрахунок часу заповнення користувацької форми (одного сценарію з різним завершенням)

Розглянемо, що означають ці позначення для наведеного вище сценарію:

- Н (перенесення руки на миша) = 0,4 сек;
- К (натискання клавіші клавіатури або миші) = 0,2 сек;

- Р (перенесення курсору до позиції на екрані) = 1,1 сек;
- М (обдумування наступного кроку) = 1,35 сек;
- R (очікування відповіді системи) – час залежить від швидкодії конкретної системи і не бере участі в розрахунках.

Існують правила як розташовувати ці операції:

- початкове розташування усіх М має бути перед кожним К та Р, крім випадків з випадającym меню, в такому випадку ставити перед Р не треба;
- якщо вводиться послідовність символів, будь якого значення, то ставити М перед К не потрібно;
- при умові що введений символ є символом що розділяє дані, оператор М також не застосовується;
- операція М не використовується у випадку коли це кінцевий символ, та застосовується якщо мова йде про перелік даних;
- якщо дія R перекриває собою операцію М то в такому випадку також не треба враховувати операцію М.

Дотримуючись цих правил можна описати будь який інтерфейс, не обов'язково мобільний або інший, залежно від виду інтерфейсу ці правила змінюються та модифікуються під специфіку досліджуємого інтерфейсу.

Всі ці значення можна адаптувати під сенсорний інтерфейс, та виміряти приблизний час що буде витрачено на виконання конкретної дії. В нашому випадку ми використовуємо цей метод для розрахунку необхідного часу на заповнення форми, завдяки чому ми матимемо змогу зробити висновок скільки часу можна зберігати використовуючи методи автоматичного заповнення користувацьких форм.

1.2.2 Аналіз XML-дерева

В основу цього метода закладена ідея того, що чим складніший програмний код для опису інтерфейсу програми, тим складніше ця програма в

використанні. З цим висловлюванням можна посперечатися але просто розглянемо принцип його роботи.

Щоб виміряти наскільки система C є складною, необхідно скористатися наступною формулою:

$$C = -N \sum_{i=1}^n p_i \log_2 p_i, \quad (1.1)$$

де N – кількість об'єктів,

p_i – відносини об'єктів в i -тому класі до всіх об'єктів ($p_i = \frac{n_i}{n}$),

n – кількість класів об'єктів,

n_i – кількість об'єктів i -го класу.

Складність окремого об'єкту можна розрахувати за допомогою формули:

$$C_0 = \frac{c}{N}, \quad (1.2)$$

Також для використання відеодисплейних матеріалів можна розрахувати складність щодо розташування об'єктів, за наступною формулою:

$$C = C_S + C_D, \quad (1.3)$$

де C_S – заснована на класах – розмірах об'єктів,

C_D – на класах – взаємне розташування.

Перевагами цього методу є те, що в його основі лежить розподілення об'єктів по типу, що в даному випадку є доцільним та практичним. Для кожного екрану розраховуються показники складності, а статичні елементи, тобто ті які залишаються незмінними отримують понижуючий коефіцієнт.

Також можна модернізувати цю формулу за допомогою певних коефіцієнтів з метою диференціювати сценарії та групи користувачів.

Коефіцієнт будуть залежати від кількості користувачів певного типу та частоти їх використання програми, а також ціна типу або часу.

$$C_{uk} = \sum_{i=1}^n (C_{tn} \times K_{tn}), \quad (1.4)$$

де C_{uk} – складність системи для k -того типу користувачів;

C_{tn} – складність n -ної задачі;

K_{tn} – коефіцієнт важливості n -ної задачі для користувача.

Перевага цього метода в тому що на відміну від GOMS розраховується складність для реальних користувачів а не абстрактна, завдяки цьому можна розрахувати загальну складність користувацького інтерфейсу. Але як правило застосовується для більш масштабного аналізу додатків.

1.2.3 Методи експертної оцінки

Це частина теорії прийняття рішень, а якщо більш точно то це процедура отримання оцінки на основі думки експертів з метою прийняття конкретного рішення.

Потрібно це для того щоб обрати рішення у випадку надзвичайно складної ситуації, при неможливості математично формалізувати процес рішення, виникає необхідність звернутися до спеціалістів у цьому питанні.

Можна виділити дві групи експертної оцінки це:

- індивідуальна оцінка це така оцінка яка заснована на думці окремих експертів що не залежать один від одного;
- колективна оцінка це така оцінка що ґрунтується на загальній думці експертів (8-12 осіб).

Існують наступні методи вимірювання об'єктів:

- ранжування – дозволяє обрати з досліджуваної сукупності факторів найбільш цінного шляхом сортування за параметром;

– парне порівняння – здійснюється порівняння пар усіх можливих параметрів з метою вибрати кращий або доказати їх рівність;

– безпосередня оцінка – в цьому випадку береться діапазон виміру характеристик об'єктів розбивається на окремі інтервали в діапазоні від 0 до 10.

Оцінки експертів записуються в таблицю для подальшого опрацювання (рис. 1.2).

	1	2	...	j	...	m
1	a_{11}	a_{12}	...	a_{1j}	...	a_{1m}
2	a_{21}	a_{22}	...	a_{2j}	...	a_{2m}
...	...	...	...	...	...	...
i	a_{i1}	a_{i2}	...	a_{ij}	...	a_{im}
...	...	...	...	...	...	...
n	a_{n1}	a_{n2}	...	a_{nj}	...	a_{nm}

Рисунок 1.2 – Таблиця з оцінками експертів

a_{ij} – оцінка параметра експертом,

n – кількість параметрів, m – кількість експертів

Після чого розраховується S_i – середнє значення важливості ознаки.

Залучені експерти можуть висловити свою думку про доцільність використання обраних методів, можливі рішення та перерозподіл акцентів, та інших дій які залежать безпосередньо від проекту в якому вони приймають участь.

Експерти не мають бути ангажовані, тобто вони не мають бути більш схильні до одного з варіантів що їм наданий, також в них не має бути шаблонного ходу думок при виборі якихось варіантів.

Процедура кількісної оцінки якості виглядатиме наступним чином.

1. Обирається категоріальна шкала оцінки характеристик якості (наприклад: цілочисельні коефіцієнти 0 ... 9, де 0 – якість не задовільна, 9 – гранично досяжний рівень на сучасному етапі розвитку галузі).

2. Призначаються кількісні значення вагових коефіцієнтів характеристик якості (вони залежать від потреб користувачів, на яких орієнтоване ПЗ).

3. Визначаються кількісні значення характеристик якості продуктів-аналогів, того ж функціонального призначення, з такими ж основними параметрами, подібної структури і прийнятним в тих же умовах експлуатації.

4. Призначаються кількісні значення базових характеристик якості – вони повинні відповідати сучасному рівню якості і прогнозованому світовому рівню (це може бути середній рівень характеристик якості по продуктам аналогам або більш високе значення, що враховують тенденції розвитку ринку і технологій, через їх вплив).

5. Визначаються (по введеній шкалі) кількісні значення характеристик якості проаналізованого програмного забезпечення.

6. Проводиться розрахунок зваженої суми (інтегрального показника якості):

$$Q_j = \sum a_i K_{ij}, \quad (1.5)$$

$$K_{ij} = X_{ij} \max, X_{ij} \leq 1/ \quad (1.6)$$

7. Якість предмету дослідження визначається шляхом порівняння отриманого значення Q_1 з відповідним базовим значенням Q_0 і індивідуального порівняння з продуктами-аналогами $Q_j, j = 2, 3$.

Рівень якості повинен відповідати базовому і мати потенціал для зростання до гранично досяжного рівня.

Найголовнішою частиною цього є аналіз результатів експертних оцінок, для такого аналізу використовуються методи математичної статистики, не виключається їх комбінація. Надаються ваги для факторів щоб надати певні пріоритети та більш виважено розподілити оцінки.

1.2.4 Автоматизація заповнення форм

Більшості користувачів не подобається заповнювати веб-форми. Особливо часто, і особливо на мобільних пристроях. Полів може бути багато, вони можуть вимагати і перевіряти багато інформації, і неодноразове їх заповнення може сильно дратувати людину. Для полегшення процесу заповнення веб-форм можна використовувати функцію автозаповнення полів від імені користувача [5].

При включеному автозаповненні браузер запропонує користувачеві підставити в поля ті дані, які раніше вже вводилися. Ця можливість зручна, однак для полів, куди вводяться секретні дані (коди, паролі, ID, PIN-коди), функцію автозаповнення відключають. Поля, які обов'язкові для заповнення, необхідно належним чином відзначити. Можна використовувати символ зірочка * для обов'язкових полів або підписати, які поля обов'язкові, а які – ні. Не варто застосовувати атрибут `required` до всіх полів без винятку, позначають тільки дійсно важливі.

Іноді, при розробці сайтів, дизайнери можуть перестаратися з спрощенням веб-форм, прибравши підписи до полів. Тим часом тег `<label>` вельми корисний при розробці зручних форм для мобільних пристроїв з сенсорним екраном [6]. Клік або торкання по `label` – все одно що клік по полю введення / прапорця (`checkbox`). Без тега `label` користувачеві доведеться прицілюватися пальцем і натискати чітко на поле чекбоксу, що може бути особливо незручним для людини з поганим зором. Крім того, вміст тега `label` читається скрінрідером – програмою, яка вимовляє вголос текст на екрані.

Однак, вже існують спеціальні програми для автоматичного заповнення форм. RoboForm – спеціальна програма для автоматизації заповнення форм реєстрації на сайтах, збереження складених для вводу даних.

Використовуючи RoboForm, значно прискорюється процес реєстрації на інтернет-ресурсах. Всі дані, які вимагається запам'ятати, зберігаються в так званій паскарті (скорочене від англ. Pas – пароль і картка – карта), щоб мати

карту паролей. Такі картки для кожного окремого випадку можуть бути складені повністю за умови, що у безкоштовних варіантах використання програм синхронізації вхідних даних передбачено збереження 10 входів. Усі введені дані шифруються та зберігаються залежно від настрою користувача, як на комп'ютерах, так і на серверах. Існує багатофакторна система доступу до даних [7].

Таким чином, програми для автоматичного заповнення форм складаються із двох головних частин. Це своєрідні бази даних, де зберігаються дані, які використовуються при введенні різних форм та власні засоби введення. Зручність використання таких програм полягає у тому, що, поповнивши необхідною для заповнення форми інформацією базу даних, ви отримаєте інформацію про це і в подальшому – дані будуть введені автоматично [8]. Таким чином, форми заповнюються швидко, при цьому вся інформація заздалегідь ретельно перевірена і щодо випадкового проникнення та збереження помилок практично виключається/

1.3 Опис та проблематика предметної області

Потреба з заповненням великої кількості особистої інформації, а іноді і приватної, з'явилася в той самий момент, як була створена перша користувацька форма, та з плином часу проблема ставала все більш гостро бо кількість полів у формах на окремих ресурсах зростала.

На сьогоднішній день існує багато підходів оцінки користувацьких форм з точки зору зручності, складності та того скільки часу необхідно на їх заповнення. Насправді таких критеріїв існує велика кількість, але саме наведені критерії є актуальними в даній роботі.

Було вирішено дослідити державні веб-сайти, бо вони через специфіку своєї роботи записують у користувача багато приватної інформації, під час реєстрації та подібними сайтами мають необхідність користуватися усі

незалежно від фізичної спроможності. Таким чином було досліджено 21 користувацьку форму на різних сайтах (рис. 1.3).

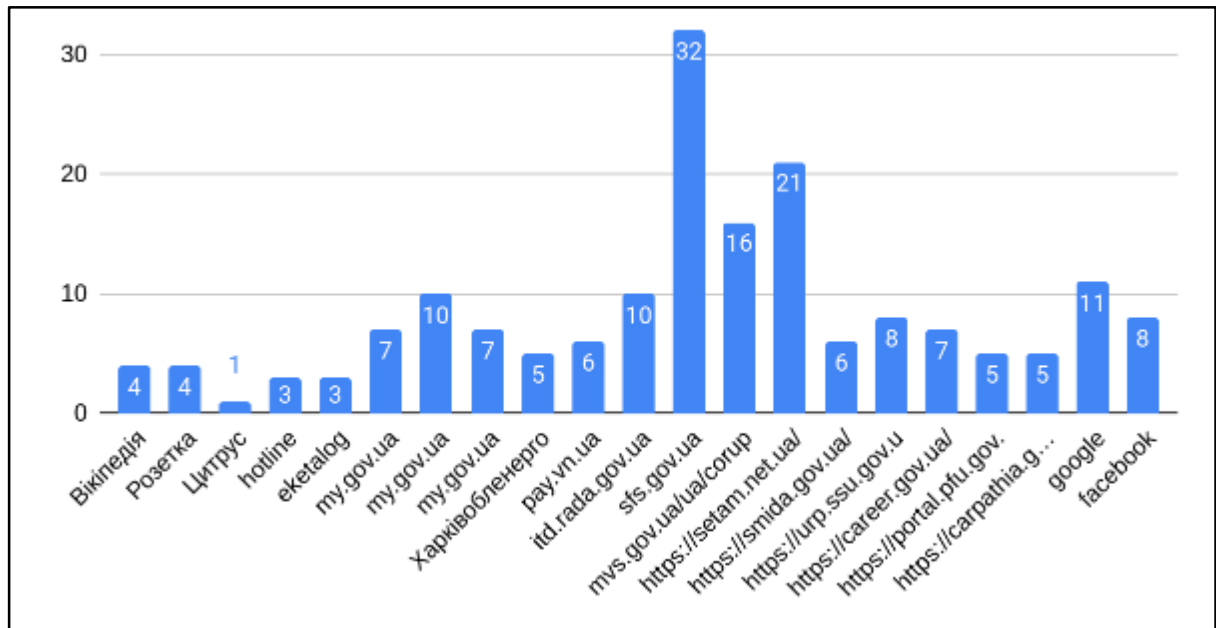


Рисунок 1.3 – Кількість полів в користувацьких формах

Для наочності було також оброблено декілька сайтів, які не є державними і відносяться до інформаційних або до продажу товарів [9]. Результатом зібраної статистичної інформації було розраховано середню кількість полів – 8.52, максимальну – 32 та мінімальну – 1. Загальна кількість полів на 21 веб-ресурсах складає 179 поля користувацької форми.

З метою отримати приблизну інформацію щодо складності полів було пораховано відношення кількості полів до їх типу (рис. 1.4). Проаналізувавши типи використовуваних полів, було побудовано діаграму, яка демонструє наскільки часто вживаються певні типи полів в користувацьких формах.

Завдяки цьому видно співвідношення полів, яке більше нахилене у бік HTML4, через те що навіть на сьогоднішній день, стандарт HTML5 використовується далеко не на всіх веб-ресурсах, в протилежному випадку отримані дані виглядали б інакше [10].

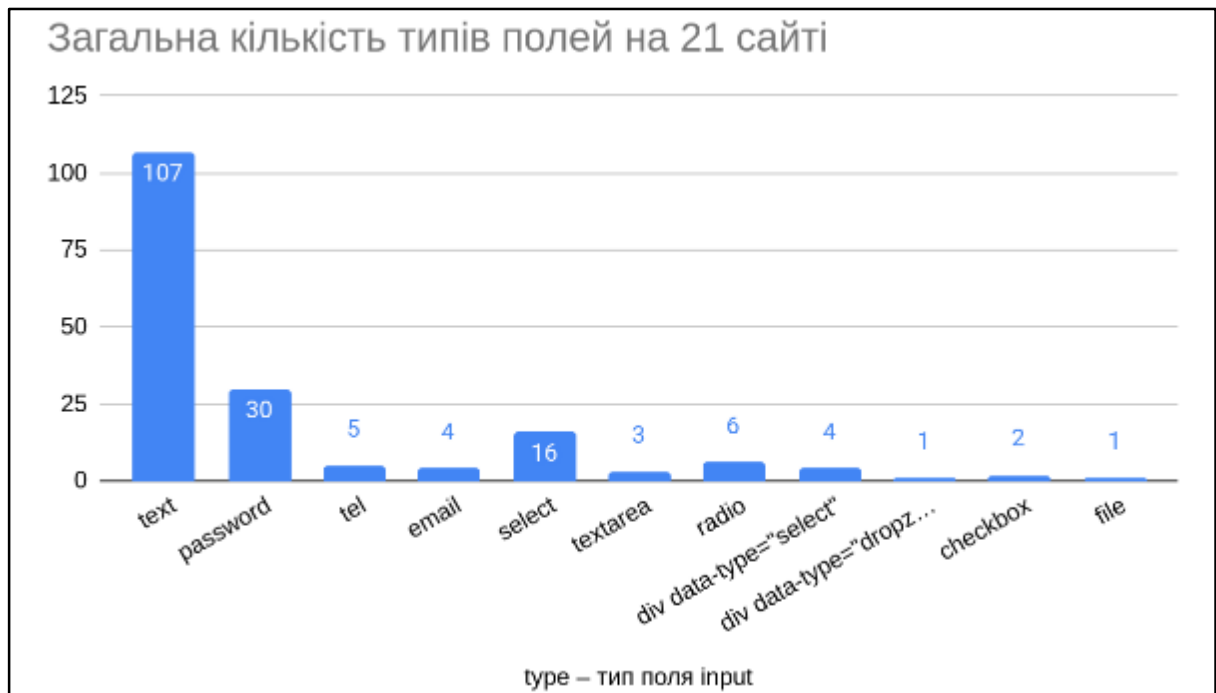


Рисунок 1.4 – Кількість полів певного типу

Підводячи підсумок отриманих статистичних даних, можна побачити, що ТОП-3 типу полів займають:

- text – текстове поле;
- password – поле введення пароллю;
- select – елемент для вибору з випадаючого списку.

Також можна побачити, що у результатах присутні нестандартні типи. Це означає, що на деяких сайтах використовують власні елементи для користувацьких форм, які реалізовані скриптовими мовами програмування, наприклад JavaScript або jQuery, у зв'язку з чим можуть виникнути проблеми під час спроби автоматичної обробки та заповненню користувацьких форм, які потребують додаткової витрати ресурсів та часу на перевірку та обробку [12].

Одним з методів завдяки якому можна навіть в таких випадках оцінити складність форми для обробки. Це метод «Аналіз XML-дерева» оскільки в даній роботі йде робота безпосередньо з HTML DOM-деревом, яке має структуровану форму подання та достатньо уніфікована з точки зору обробки вихідного коду. Даний метод дозволяє математично обґрунтувати отримані результати, та зобразити складність яку вийшло уникнути для користувача.

Завдяки чому можна якраз і оцінити випадки коли використовуються незвичайні елементи на сторінці, тобто отримані дані будуть більш повноцінними порівняно з іншими методами які можуть обробити лише стандартні користувацькі елементи.

1.4 Відмінності HTML5 від HTML4 та вплив на користувацькі форми

HTML5 приніс багато нового в розробку користувацьких веб-сторінок, включаючи нові типи поля `input`, в форму. Було додано 13 нових типів, що потенційно робить форми більш гнучкими (таблиця 1.1).

Таблиця 1.1 – Нові значення типу поля `type` в `html5`

Тип	Опис
<code>color</code>	Віджет для вибору кольору
<code>date</code>	Поле для вибору календарної дати
<code>datetime</code>	Вибір дати та часу
<code>datetime-local</code>	Вибір місцевої дати та часу
<code>email</code>	Поле для електронної адреси
<code>number</code>	Для вводу чисел
<code>range</code>	Повзунок для вибору значення з діапазону
<code>search</code>	Поле для пошуку
<code>tel</code>	поле для телефонного номеру
<code>time</code>	Поле для введення часу
<code>url</code>	Поле для введення веб адрес
<code>month</code>	Поле для вибору місяця
<code>week</code>	Поле для вибору тижня

Це є важливим, бо надає більшої гнучкості користувацьким веб-формам, та полегшує саму процедуру обробки даних, введених користувачем.

Також це потенційно значно розширює можливості валідації даних та автоматичної обробки полів. Подібний підхід дозволяє зробити форми більш уніфікованими, що в свою чергу дозволяє:

- збільшити охоплення типів користувацьких даних;
- ефективно впровадження легких методів автоматизації заповнення;

- структурування користувацьких даних;
- автоматична перевірка коректності даних на етапі їх введення;
- унікальність типів даних.

1.5 Мета роботи та постановка задач дослідження

Метою даної роботи є створення методики автоматичного заповнення користувацьких форм та вибір критеріїв для оцінки ефективності запропонованого методу.

Запропонована методика побудована на тому що користувач має заздалегідь підготувати свій профайл, що буде містити в собі усю необхідну особисту інформацію, яка необхідна під час заповнення користувацьких форм на різних сайтах.

Створений профайл користувач може, переносити з собою на будь якому накопичувачі та звантажувати його до програмної реалізації запропонованої методики. Програмна реалізація, на основі завантаженого до неї файлу має заповнити будь яку користувацьку форму.

Сформульована наступна гіпотеза, яку необхідно перевірити в ході дослідження: «Якщо використовувати заздалегідь підготовлений профайл користувача під час автоматичного заповнення користувацької форми, то будуть повністю усунуті помилки під час введення, та форма буде заповнена миттєво».

Після чого визначено завдання дослідження, для досягнення мети:

- аналіз стану проблеми оцінки громіздкості користувацьких інтерфейсів та їх складності;
- огляд комерційних, державних та некомерційних веб-сайтів;
- огляд методів і критеріїв оцінки користувацьких форм;
- вибір і обґрунтування методу проведення експерименту;
- розробка методу і складання плану проведення експерименту;
- проведення та обробка результатів експерименту;

- формування методики для оцінювання якості призначеної для користувача користувацької форми;
- проведення порівняльного аналізу результатів за отриманої методики і експерта в області, що вивчається;
- перевірка працездатності методики;
- формування рекомендацій щодо використання методики, що пропонується.

Об'єктом дослідження процес автоматизації заповнення користувацьких форми, заходи які можна застосувати під час автоматизації вводу користувацької інформації.

Предметом дослідження є користувацькі форми їх схожість та відмінність один від одної, опис кожного поля та відмінності у іменах атрибутів полів.

Таким чином, об'єктом дослідження є процес автоматичного заповнення користувацької форми. Предметом дослідження, у свою чергу, є дослідження користувацьких форм на різних сайтах, та пошуку можливості їх уніфікувати.

2 ПЛАНУВАННЯ ДОСЛІДЖЕННЯ

Будь-яке дослідження починається зі складання плану дослідження. Планування дослідження в даній роботі необхідно для обґрунтування вибору критеріїв оцінювання якості призначеної для заповнення користувачем користувацької форми, проведення експерименту та обробки експериментальних даних [13]. План дослідження – це перелік робіт, які включають послідовні дії для досягнення поставленої мети дослідження.

2.1 Аналіз критеріїв оцінки користувацьких форм

Показник оцінювання – це засіб оцінювання, що містить список критеріїв оцінки завдання. Критерій оцінювання розбиває призначену роботу на окремі частини і описує особливості виконання кожної з частин в залежності від рівня складності.

Насамперед необхідно визначити групи критеріїв, за якими можна буде оцінювати користувацьку форму:

- кількість полів;
- види полів;
- відношення кількості полів до їх типу;
- час, що необхідний на заповнення форми;
- час, що необхідно витратити на заповнення з помилками при валідації полів;
- статистика щодо збігів полів в різних формах.

2.2 Розробка методики проведення експерименту

Метод експерименту є одним з основних методів наукового пізнання взагалі, психологічного дослідження зокрема. Відрізняється від спостереження активним втручанням у ситуацію з боку дослідника, що

здійснює планомірне маніпулювання однієї або декількома критеріями і реєстрацію супровідних змін у поведженні досліджуваного об'єкта.

Планування експерименту дозволяє мінімізувати число необхідних випробувань, встановити раціональний порядок і умови проведення досліджень в залежності від їх виду і необхідної точності результатів.

Методологія експерименту – це загальна структура експерименту, тобто постановка і послідовність виконання експериментальних досліджень.

Методологія експерименту включає в себе наступні основні етапи:

- а) розробку плану-програми експерименту;
- б) оцінку вимірювань і вибір засобів для проведення експерименту;
- в) проведення експерименту;
- г) обробку і аналіз експериментальних даних.

2.3 План проведення експерименту і обробки експериментальних даних

Планування експерименту включає ряд етапів.

1. Встановлення мети експерименту (метою даного дослідження є уточнення набору критеріїв для оцінки користувацьких форм).

2. Виявлення і вибір критеріїв оцінювання (докладний опис критеріїв зазначено в п. 2.1).

3. Уточнення умов проведення експерименту (експеримент проводиться відповідно до протоколу, а також видано документ, в якому необхідно вказати бал кожним критерієм).

4. Встановлення точності результатів вимірювань за допомогою надання вагових коефіцієнтів для кожного з критеріїв виділивши їх важливість порівняно з іншими. З урахуванням цієї обставини кожному критерію присуджується свій бал.

5. Складання плану і проведення експерименту – кількість і порядок випробувань, спосіб збору, зберігання і документування даних. Експеримент буде проходити відповідно до схеми (рис. 2.1).



Рисунок 2.1 – Проведення експерименту

6. Статистична обробка результатів експерименту, побудова математичної моделі поведінки досліджуваних характеристик.

7. Пояснення отриманих результатів та формулювання рекомендацій по їх використанню.

3 АНАЛІЗ КОРИСТУВАЦЬКИХ ФОРМ

Веб-форма є одним із основних елементів взаємодії між користувачем та власником Інтернет-ресурсу. За допомогою форм користувачі «спілкуються» із сайтом, здійснюють цільові дії, авторизуються та реєструються, шукають необхідні товари та інше. Користувач вводить власноруч дані, які вимагає форма, після чого дані надсилаються на сервер для подальшої обробки.

З моменту заснування перших сторінок в мережі Інтернет, які вели роботу з користувацькими даними, нічого в механізмі роботи не змінилося, додалися лише варіанти з валідацією даних. Таким чином, це достатньо старий елемент, але який з часом наповнювався та набуває нових особливостей.

Не в останньому їх використовують для оформлення замовлень та продажу товарів, за звучністю їх можна поділити на «компактні» та «надлишкові». Відповідно це еквівалент розміру користувацької форми, якщо форма буде великою, то це незручно для кінцевого користувача.

3.1 Приклади надлишкових користувацьких форм

До цієї категорії відносяться форми, в яких кількість полів можна лічити від 10, в таких формах за один раз необхідно заповнювати занадто велику кількість полів. Іноді їх кількість може не вмішуватись в один екран монітора комп'ютера [14].

Заповнення подібних форм забирає занадто багато часу, та у випадку помилки, можуть втратити усі дані, від подібної особливості користувацьких форм не можна бути захищеним. Відповідно втративши дані які були записані в більш ніж 30 полях, буде не просто неприємно, але й витратить певний час на повторне заповнення інформації.

До надлишкових форм можна віднести такі форми форми, що містять багато полів (рис. 3.1).

Регистрация

Если Вы уже зарегистрированы, перейдите на страницу [входа в систему](#).

1. Основные данные

* Имя, Отчество:

* Фамилия:

* Телефон:

* E-Mail:

2. Ваш адрес

Страна: Российская Федерация

* Регион / Область: --- Выберите ---

* Город:

Адрес:

Индекс:

Компания:

3. Ваш пароль

* Пароль:

* Повторите пароль:

4. Рассылка новостей

Подписка на новости: ☐ Да ☒ Нет

Я прочитал [Политика Безопасности](#) и согласен с условиями ☐

[Продолжить](#)

Рисунок 3.1 – Пример надлишкової форми № 1

Наведений приклад середньостатистичної надлишкової форми, яка містить в собі достатню кількість полів для введення, та особливі елементи в яких треба обрати один з декількох варіантів `radiobutton`, та елемент де необхідно вибрати один варіант із випадаючим списком `select`.

Поточний приклад не є нестерпним, в тому плані що при виникненні помилки, полів не настільки багато, щоб залишити спробу повторно ввести особисту інформацію.

Також існують такі користувацькі форми, які збирають значно більше інформації стосовно користувача, наприклад державні установи або юридичні заклади (рис. 3.2).

Форми подібного роду збирають не просто багато інформації, але ця інформація є занадто конфіденційна та важлива, де ціна помилки є високою. Наприклад при введенні номеру IBANK, або номеру паспорту, ідентифікаційного номеру платника податків.

E-mail * — На этот e-mail будет выслана ссылка активации кабинета
 Логин *
 Пароль *
 Повторите
 IP-адрес [\[Вставить текущий IP\]](#) — ВНИМАНИЕ! Если Вы установите IP, то доступ будет разрешен только с этого компьютера (сети). Что бы пользоваться сервисом с любого компьютера, оставьте поле пустым.

Дополнительная информация

Отправитель SMS — Это поле будет вставляться вместо номера отправителя SMS
 Регион * Новосибирская область
 Юридический адрес
 Фактический адрес *
 Наименование организации *
 Действует на основании
 Контактное лицо *
 ИНН *
 КПП
 Банк *
 БИК *
 р/с *
 к/с *

* * *

Контактная информация

Контактный телефон * — (код города) телефон
 Мобильный телефон — В формате +7ddddddddd
 ICQ
 Я принимаю * ☐ [договор по публичной оферте](#)
 Поля отмеченные звездочкой (*) обязательны для заполнения

Рисунок 3.2 – Приклад надлишкової форми №2

Відповідно це може бути не просто форма реєстрації на сайті, але й форма надання заяви на отримання певних послуг, тоді подібну форму треба буде заповнювати неодноразово, що теж збільшує вірогідність утворення якоїсь механічної помилки.

Людам робота яких зав'язана на подібних формах, також потрібне рішення проблеми пов'язаної з великою кількістю постійного вводу однотипної інформації. Наприклад при обліку товару, або реєстрації нових співробітників, звіти в спеціальних додатках, та міграції клієнтської бази з одного додатку інший (рис. 3.3).

Рисунок 3.3 – Приклад надлишкової форми №3

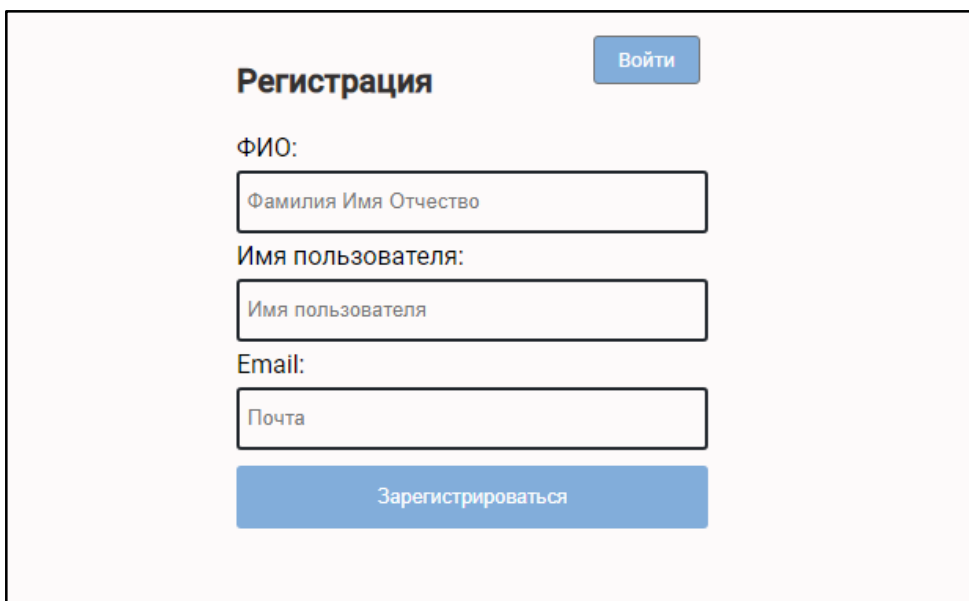
Як можна побачити подібного роду форми не можна назвати еталоном зручності та дружніми до користувача, так і форми люди заповнюють лише тоді, коли немає альтернатив. Зумовлено це тим, що кількість полів є дуже великою – як звичайних так і обов’язкових, що на даний момент можна розцінювати як зневагу до користувача. Час, що необхідно витратити на заповнення подібних форм в деяких випадках можна рахувати хвилинами, також можна побачити, що поля не мають ніяких підказок щодо заповнення полів (формат мобільного номеру або пошти). Такі дрібниці також мають чимале значення, бо збільшують вірогідність виникнення помилки а відповідно час на заповнення форми.

3.2 Приклади компактних користувацьких форм

Якщо користувацька форма при виникненні помилок скидає всю інформацію то вірогідність того, що користувач спробує її заповнити повторно зменшиться на 80%, якщо полів забагато і на 10% якщо полів мало – від двох до чотирьох (рис. 3.4).

Приблизно так виглядають наразі користувацькі форми на більшості сайтів, які не надають юридичних послуг. Збирається мінімальна кількість

користувацьких даних, але з потенційною можливістю отримати необхідні особисті дані користувача пізніше.



The image shows a registration form titled "Регистрация" (Registration) in Russian. In the top right corner, there is a blue button labeled "Войти" (Login). The form contains four input fields: "ФИО:" (Full Name) with placeholder text "Фамилия Имя Отчество", "Имя пользователя:" (Username) with placeholder text "Имя пользователя", "Email:" with placeholder text "Почта", and a large blue button at the bottom labeled "Зарегистрироваться" (Register).

Рисунок 3.4 – Приклад компактной формы №1

Але це також не є правилом, частіше форми складаються з більшої кількості полів, бо необхідні користувацькі дані все таки необхідно отримати, через політику користування сервісом, наприклад дату народження (рис. 3.5).



The image shows a registration form titled "Реєстрація" (Registration) in Ukrainian. It features several input fields: a first name field with "Ivan", a last name field with "Petrenko", an email field with "ivan.petrenko@ukr.net", and two password fields represented by asterisks. Below the passwords is a "Дата народження" (Date of birth) section with dropdown menus for day (2), month (Червень - June), and year (1988), followed by the text "Навіщо вказувати дату народження?" (Why specify date of birth?). There are radio buttons for "Жінка" (Woman) and "Чоловік" (Man), with "Чоловік" selected. At the bottom, there is a small disclaimer: "Нажимая «Создать аккаунт», вы соглашаетесь с нашими Условиями использования и подтверждаете, что ознакомились с нашей Политикой использования данных." (By clicking "Create account", you agree to our Terms of Use and confirm that you have read our Data Usage Policy).

Рисунок 3.5 – Приклад компактной формы №2

Таким чином, можна побачити що кількість полів та їх тип безпосередньо впливають на то, чи повернеться користувач у випадку помилки, та чи взагалі почне заповнювати таку форму.

Обов'язково треба враховувати такі нюанси під час проектування та під час створення механізмів для автоматичного заповнення користувацьких форм. Як можна побачити вони всі різні тож існують певні нюанси для коректного їх заповнення користувачем.

4 МЕТОДИ АВТОМАТИЧНОГО ЗАПОВНЕННЯ КОРИСТУВАЦЬКИХ ФОРМ

Першим, що спадає на думку – це автозаповнення браузером, таких полів як логін та пароль або ж іншими словами форм авторизації, але це лише примітивний варіант, хоча просто ця інформація значно зменшує час на заповнення. Цей приклад демонструє лише те, що паролі бувають доволі складні, лише на їх введення може витратитися від 2-3 секунд якщо пароль доволі коротким, та від 8 секунд якщо пароль дуже складний та великий. Також треба брати до уваги потенційні неправильне введення, які у разі збільшують час на авторизацію у будь якому сервісі [15].

Але це лише один з багатьох випадків, якщо брати загально великі форми які були розглянуті то ручне заповнення таких форм може розраховуватись хвилинами. Та відповідно нівелюється усі недоліки UX, де усі дані або їх певна можуть бути втрачені, шляхом того що всі поля автоматично будуть введені в користувацьку форму в заздалегідь призначені для цього місця [16].

4.1 Парсинг полів по типу

Поля користувацьких форм розподіляються по таким типам полів як: текст, пароль, дата, цифри, час та велике текстове поле. На перший погляд здається що типів багато, і можна легко виділити основні типи полів. З одного боку так і є, проблема виникає тоді коли в одній формі присутній якийсь тип поля більше одного разу.

Тобто дві дати або дев'ять текстових полів, які з точки зору типу поля є однаковими, бо в такому випадку виникають проблеми з тим щоб розпізнати яке поле форми за що відповідає (рис. 4.1).

Рисунок 4.1 – Використання парсингу за типом для заповнення

Як можна побачити, були заповнені всі поля унікального типу, але поле, яке відповідало за прізвище не було заповнене, через те що в концепції цього підходу не закладений варіант з полями однакового типу, обирається лише одне, яке є на думку розробника реалізації найчастіше вживаними.

В окремих випадках лише цього метода буде достатньо, наприклад в формі авторизації, але в загальному випадку великих користувацьких форм це неідеальним. Хоча може зменшити час необхідний на заповнення на певний відсоток.

4.2 Парсинг DOM дерева

Цей метод є різновидом методу «парсинг полів за типом» різниця лише у тому що в даному випадку відстежується зв'язок між усіма полями в формі, їх відношення та комбінації за допомогою парсингу безпосередньо DOM дерева (рис. 4.2), а не просто пошуку по типу поля.

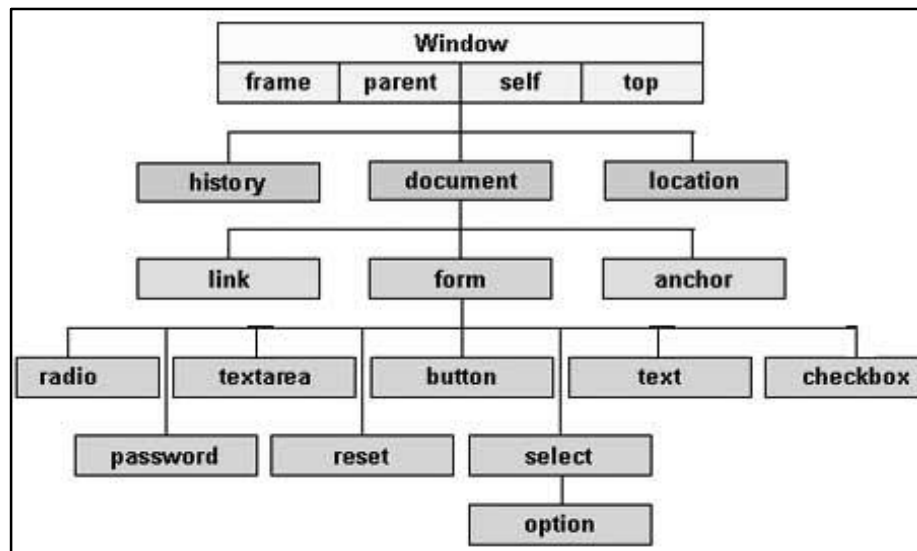


Рисунок 4.2 – DOM дерево та елементи форми

Такий метод є більш складний у виконанні але й більш дієвий бо виключає зайві поля або елементи сторінки/екрану що в свою чергу є більш дієвим методом.

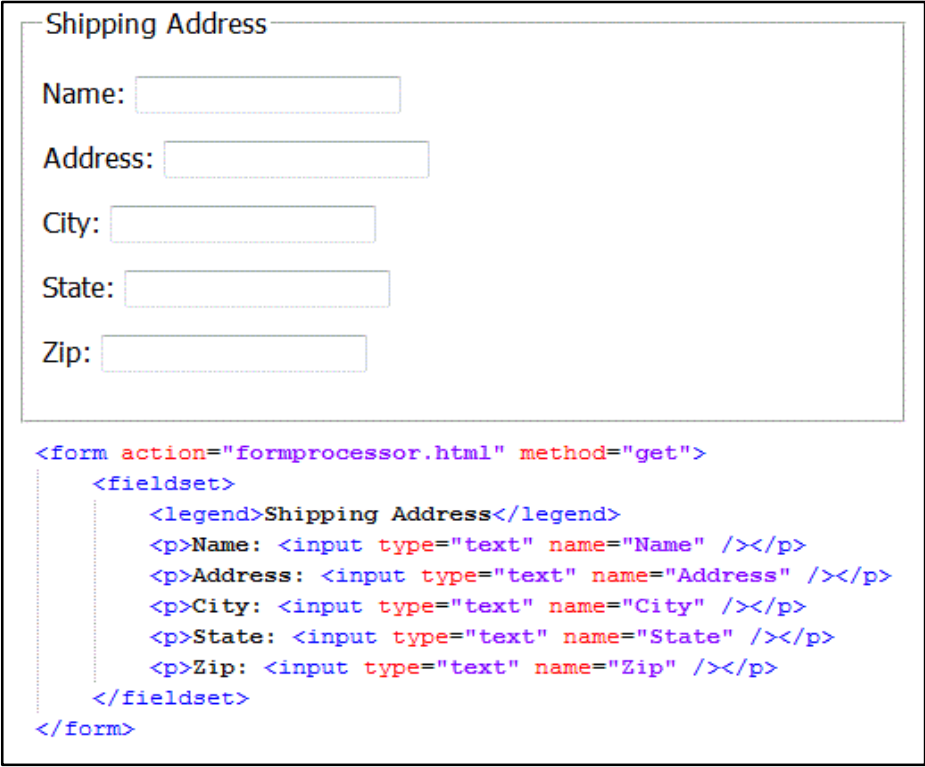
Завдяки цьому методу можна припустити те що, якщо у формі усього 2 текстових поля, та присутнє поле для вводу пошти, то наявні два поля це ім'я та прізвище, в протилежному випадку друге поле призначене для введення пошти користувача.

Саме завдяки цим перевагам метод парсингу DOM дерева є більш переважним ніж метод парсингу по типу поля.

4.3 Парсинг полів за атрибутом name

Частіше більш доцільним та практичним буде використовувати метод розбиття користувацької форми на поля за атрибутами, що дозволяє більш гнучким способом розбити поля вводу користувацької інформації за атрибутом name які в середині форми є унікальними (рис. 4.3).

Можна побачити, що хоча всі поля є однакового типу, вони усі розрізняються атрибутом name, який є унікальним для форми в якій застосовується.



Shipping Address

Name:

Address:

City:

State:

Zip:

```
<form action="formprocessor.html" method="get">
  <fieldset>
    <legend>Shipping Address</legend>
    <p>Name: <input type="text" name="Name" /></p>
    <p>Address: <input type="text" name="Address" /></p>
    <p>City: <input type="text" name="City" /></p>
    <p>State: <input type="text" name="State" /></p>
    <p>Zip: <input type="text" name="Zip" /></p>
  </fieldset>
</form>
```

Рисунок 4.3 – Різниця між полями по атрибуту name

Завдяки цьому можна розрізнити дані та знайти необхідне поле в формі для подальшої роботи з ним.

Такий метод є дуже гнучким особливо якщо порівнювати з методом парсингу полів по типу, але в силу різноманітності сайтів нема уніфікації. Не дивлячись на те що можна точно знайти необхідне поле, можуть виникнути проблеми при спробі повторно використати адаптовану реалізацію для одного сайту, для іншого.

Немає єдиної уніфікації щодо того як називати поля, у якості імені можуть бути обрані як звичні слова на англійській мові, так і просто послідовність символів, які не обов'язково можна зрозуміти прочитавши їх. Лише таке поле як «Ім'я» може бути записано як: name, fname, fullname, murname, та навіть X1psFx тощо. Відповідно треба адаптувати алгоритм таким чином щоб можна було розпізнати більшу частину, тих полів, які можна логічно зв'язати зі звичайними та загально використаними ім'ям.

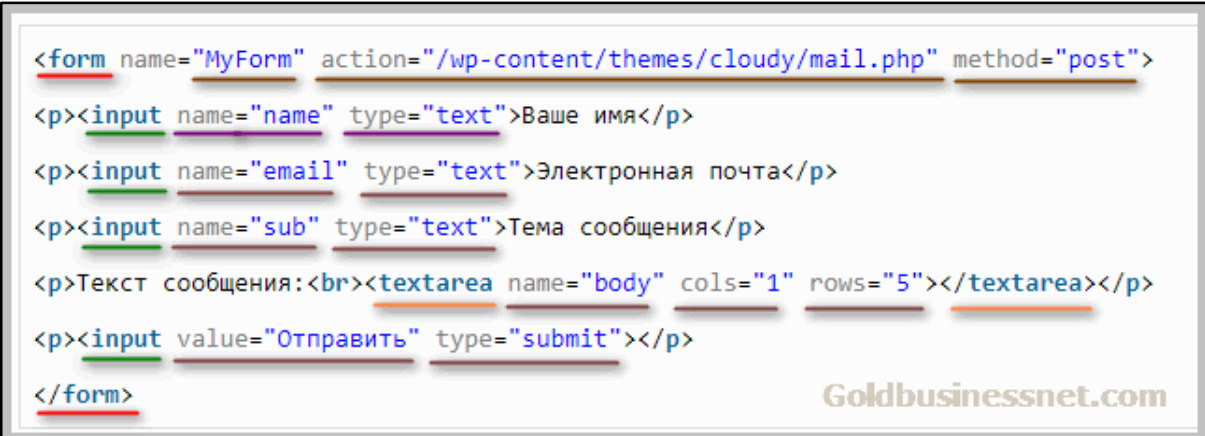
5 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

Наявні методи обробки не є достатньо ефективними в плані охоплення різноманітних випадків, побудови користувацьких форм. Тому була проведена певна модифікація та поєднання існуючих методів, в результаті було отримано альтернативний метод обробки користувацьких форм, який не містить недоліки в аналогів в проекції на універсальність використання над різноманітними користувацькими формами.

5.1 Парсинг DOM дерева з додатковими параметрами

Цей метод є поєднанням методів «Парсинг DOM дерева» та «Парсинг полів за атрибутом» з тією різницею, що немає прив'язки до конкретного типу поля.

Порівняно з іншими є найефективнішим методом, та є дуже гнучким що дозволяє адаптувати під будь яку користувацьку форму. Це досягається шляхом обробки додаткових атрибутів таких як: placeholder, name, type, title, value, data-* та інші (рис. 5.1).



```

<form name="MyForm" action="/wp-content/themes/cloudy/mail.php" method="post">
<p><input name="name" type="text">Ваше имя</p>
<p><input name="email" type="text">Электронная почта</p>
<p><input name="sub" type="text">Тема сообщения</p>
<p>Текст сообщения:<br><textarea name="body" cols="1" rows="5"></textarea></p>
<p><input value="Отправить" type="submit"></p>
</form>
  
```

Goldbusinessnet.com

Рисунок 5.1 – Різниця між полями по атрибуту name

Головна відмінність у тому, що обробляються також обгортки по типу `label` а саме тексту, що в них знаходиться для того, щоб уникнути різноманітностей, які можуть бути в атрибуті `name`. Подібне рішення дозволяє уніфікувати алгоритм під обробку великої кількості різноманітних користувацьких форм.

Таке рішення прийнято на основі того, що підписують поля для людей, і цей текст обробляється користувачами, і на основі нього заповнюється. Таким чином стає значно менше варіантів для порівняння, наприклад для англійської мови поле для електронної адреси матиме лише одну назву – `email` [16].

Для української вже мови, варіантів буде більше: `email`, пошта, електронна пошта, скринька, електронна скринька. В даному випадку всього 2 варіантів для того щоб ідентифікувати для чого поле необхідне, у загальному випадку подібна варіативність складатиме від 3 до 5, у виключних випадках досягатиме 10, бо якщо обробляти велику кількість сайтів, можуть бути зібрані різноманітні варіації.

Те саме й стосується тематичних сайтів, на яких назви полів спеціально змінено, але і в цих випадках поповнивши базу, можна подібну форму легко обробити автоматичним методом заповнення.

5.2 Способи зберігання користувацьких даних

Не кожен наважиться зберігати свої особисті дані, які містять не тільки адресу але й коди паспорта, свідоцтв та банківських карток в посторонніх сервісах, що в певному сенсі значно обмежує портативність рішення.

З метою захищення особистої інформації користувача, було обрано варіант зберігати користувацькі дані в окремому файлі, який можна повторно використовувати на іншій пристрої, за допомогою USB накопичувача.

5.2.1 TXT файл як сховище приватної інформації

Будь який файл не підійде, бо наприклад при зберіганні в звичайному TXT файлі, користувач буде змушений дотримуватись відповідної структури, що в свою чергу не зручно через вірогідність виникнення помилки в цій структурі.

Наприклад зберігання даних через певний розділяючий знак – вимагає дотримуватися заданої послідовності даних, роблячи їх структурованими але залежними від розміщення в послідовності цих самих даних.

Аналогічна ситуація якщо кожен новий інформаційний блок розташовуватиметься у новому рядку, є вірогідність зробити зайвий відступ або забути що зробити, в результаті чого всі дані будуть некоректні, бо вони будуть зсунуті, знаходячись не на своїх місцях.

Подібні умови не можна віднести до комфортних, тому подібний формат може бути доречний якщо інформація не змінюється, або навіть генерується автоматично в подібний файл.

5.2.2 Використання у якості сховища XML формат

Доволі старий варіант, що дозволяє його обробити на будь якій мові програмування, та опрацьовувати у різних системах.

Перевагою порівняно з варіантом зберігання у звичайному файлі, це можливість зберігати дані у структурованому вигляді. Це дозволяє уникнути жорсткої прив'язки до послідовності зберігання даних, тому що дані можна розташовувати у різноманітному порядку, завдяки тому що у кожного блоку даних є своє ім'я (рис. 5.2).

```
<student>
  <name>Christopher Co</name>
  <age>29</age>
  <level>7</level>
  <gender>M</gender>
  <status>good</status>
</student>
```

Рисунок 5.2 – Вигляд даних у XML форматі

Кожна структурна частина, обертається тегом з відповідним ім'ям, що дозволяє візуально легко зрозуміти зо що відповідає той чи інший тег, та яку інформацію.

Але не всі користувачі легко орієнтуються в подібного роду коді.

5.2.3 Використання у якості сховища JSON формат

Це більш сучасний варіант, який використовується при передачі даних між серверами та клієнтами. Також він більш компактний та зрозумілий звичайному користувачеві, через відсутність тегів які багатьох лякають, які треба замикати, слідкувати щоб нічого не пропустити. Дані виглядають дуже звично для звичайної людини, та не містить зайвих елементів (рис. 5.3).

```
{
  "name": "Christopher Co",
  "age": 29,
  "level": 7,
  "gender": "M",
  "status": "good"
}
```

Рисунок 5.3 – Вигляд даних у JSON форматі

Ще однією перевагою є те, що такі файли важать значно менше ніж XML, це може бути важливо, якщо завантажувати файл в якесь онлайн сховище при повільному з'єднанні.

Але найважливішим фактором є те, що дані зберігаються у читаємому форматі, та не є вимогливим для звичайного користувача.

5.2.4 Порівняння варіантів сховища

Після огляду доступних для зберігання форматів, необхідно їх порівняти зваживши позитивні та негативні моменти, та обрати найкращий варіанти для використання (табл. 5.1).

Таблиця 5.1 – Порівняння файлів для сховища

	TXT	XML	JSON
Структурування даних	—	+	+
Читаємість даних	—	—	+
Простота в використанні	—	+	+
Відсутність вимоги до послідовності даних	—	+	+
Компактність	+	—	+
Нативна підтримка обробки іншими системами	—	+	+

В результаті порівняння було обрано JSON, тому що він по усім параметрам при порівнянні є лідером. Але найголовніше що для користувачів які ніколи не працювали з подібного роду даними, він є інтуїтивним, що робить його ідеальним кандидатом для зберігання приватних даних на власному носії або комп'ютері.

5.3 Статистика підпису полів та атрибуту name

Було оброблено користувацькі форми на предмет того що підпис полів є більш стандартизованою для користувацьких форм аніж ім'я що було введене в атрибут name. Було побудовано стовпчасту діаграму, за логарифмічною шкалою для показників атрибуту (рис. 5.4).

Оскільки в випадку з атрибутом, ім'я яке було введено чутливе до регістру, бо можуть бути випадки, коли у полів ідентичні назви, але порядок великих літер різний, що відрізняє їх. Можна побачити що більшість даних зустрічається один раз, а ті що повторюються в середньому складають 2-3 півтори серед усієї вибірки [17].

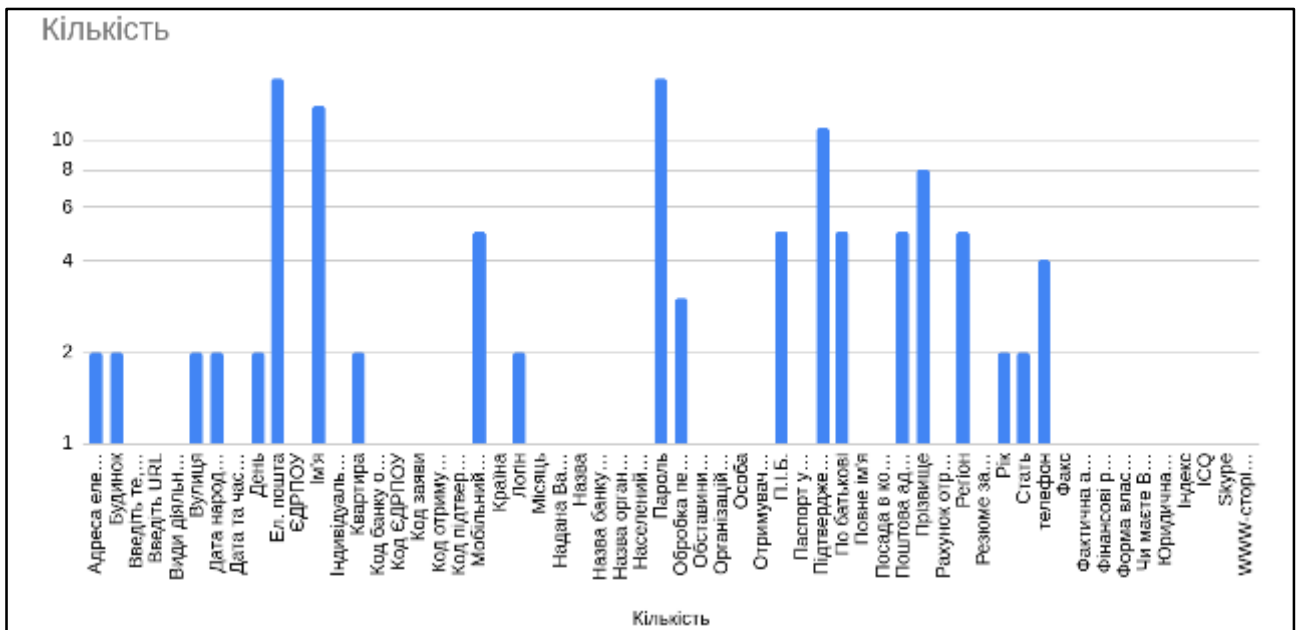


Рисунок 5.6 – Згрупована логарифмічна шкала для підпису полів

Згрупована логарифмічна шкала зображає те що було досягнуто більшу кількість збігів опису полів на різноманітних сайтах, що підтверджує ефективність використання цього опису під час автоматичного заповнення.

Саме тому додаткова обробка, тексту полів, дозволяє ефективно обробити користувацьку форму, завдяки чому програмний додаток зможе коректно відпрацьовувати на різноманітних сайтах.

5.4 Способи автоматичного введення користувацьких даних у форму

Основним способом автоматично ввести дані в користувацьку форму на веб-сторінці, це використати скриптову мову програмування JavaScript. Який є універсальним рішенням для будь якої веб-сторінки, тому що кожен веб-браузер підтримує цю мову.

Таким чином йде обробка DOM дерева, тобто на сторінці знаходиться форма, і кожен елемент всередині, обробляється перевіряється з паттерном, перевіряються сусідні теги. Це дозволяє обробити ситуації коли не вийшло заповнити елемент за ім'ям атрибуту, другим етапом стає аналіз навколишнього оточення тегів, їх вмісту, підвищує ефективність.

5.5 Вибір форми для тестування

Для перевірки та розрахунків, було обрано реєстраційну форму на сайті «ВЕРХОВНА РАДА УКРАЇНИ ЕЛЕКТРОННІ СЕРВІСИ», що знаходиться за посиланням: <https://itd.rada.gov.ua/services/Account/Create>.

Користувачка форма складається з 10 полів, серед яких значна кількість приватної інформації, а саме мобільний телефон та декілька полів для визначення адреси людини що реєструється на ресурсі (рис. 5.7).

Електронні сервіси

ВЕРХОВНА РАДА УКРАЇНИ
ЕЛЕКТРОННІ СЕРВІСИ

СЕРВІСИ ПРОФІЛЬ КОРИСТУВАЧА ЕЛЕКТРОННІ ПЕТИЦІЇ ОБГОВОРЕННЯ ЗАКОНОПРОЄКТІВ

Створення облікового запису користувача

*Прізвище:

*Ім'я:

По батькові:

*Електронна пошта:

*Мобільний телефон: +380

*Регіон: не визначено

*Населений пункт, район:

*Вулиця:

*Будинок:

Квартира:

**Увага! Служба мобільного зв'язку Інтернетком (096) не приймає СМС-повідомлень від з'явування підтримки в зв'язку з власними технічними обмеженнями.
***Створення облікового запису, я надаю згоду на оброблення своїх персональних даних за допомогою інформаційно-телекомунікаційної системи Верховної Ради України з метою ідентифікації користувача сервісу Верховної Ради України.

Створити обліковий запис

Рисунок 5.7 – Сторінка та форма реєстрації на ресурсі

Для того щоб заповнити подібну форму, необхідно заздалегідь мати необхідну інформацію, виглядати вона буде наступним чином.

```
{
  "FIRSTNAME": "Артем",
  "LASTNAME": "Алеєв",
  "MIDDLENAME": "Ренатович",
  "MAIL": "artem.alieiev@nure.ua",
  "PHONE": "+380123456789",
```

```

"REGION": "Харківська область",
"SETTLEMENT": "Харків",
"STREET": "вул. Леніна",
"BUILDING": "3",
"HOUSE": "13"
}

```

Відповідно до вимоги, дані структуровані, та мають унікальні ключі, для того щоб їх було легко ідентифікувати, та заповнити власноруч.

5.6 Розрахунки для ручного введення даних у форму

Буде проведено 2 заміри:

- мануальний, що означає ручне ввімкнення секундоміру, заповнення форми та зупинення секундоміру;
- розрахований за допомогою методу GOMS.

На ручний ввід з помірною швидкістю знадобилося 45 секунд. Це час під час заповнення якого не робилися помилки, в протилежному випадку.

Такою був проведений замір для заповнення користувацької форми з допущенням помилок та їх виправленням, час склав 63 секунди.

Згідно з методом GOMS було розраховано оцінювання часу, який має бути витрачено на автоматичне заповнення:

Н+М+Р+К+К+К+К+К+Н+Р+К+К+К+К+К+Н+Р+К+К+К+К+К+К+К+
 К+К+Н+Р+К+К+К+К+К+К+К+К+Н+Р+К+М+К+К+К+К+М+Р+К+К+К+
 К+К+М+Р+К+К+К+К+К+К+К+К+К+К+М+Р+К+М+Р+К+К

Цей запис у цифровому записі виглядає наступним чином:

$$\begin{aligned}
 &0,4+1,35+1,1+0,2*5+0,4+1,1+0,2*5+0,4+1,1+0,2*9+0,4+1,1+0,2*28+0,4+ \\
 &+1,1+0,2+1,35+0,2*4+1,35+1,1+0,2*4+1,35+1,1+0,2*11+1,35+1,1+0,2+ \\
 &+1,35+1,1+0,2*2=34.
 \end{aligned}$$

Завдяки методу GOMS було розраховано час, який св середньому має витратити користувач для ручного заповнення своєї інформації до форми, цей час становить 34 секунди.

5.7 Реалізація та застосування методу автоматичного заповнення

Програмна реалізація була виконана з використанням скриптової мови програмування JavaScript, Саме через те, що вона за замовчуванням підтримується всіма браузерами. Також на JavaScript можна створити розширення для браузера, що відображає суть ідеї того, як має бути використаний подібний підхід.

Приклад простої програмної реалізації автоматичного заповнення користувацьких форм, можна подивитись у Додаток А.

5.8 Порівняльний аналіз автоматичного та ручного введення

В результаті проведеного експерименту, було отримано 4 заміри часу що було витрачено на заповнення:

Таблиця 5.2 – Результати замірів при заповненні форми

Метод вводу	Час на заповнення
Ручне введення користувацької інформації	45 сек
Ручне введення користувацької інформації з допущенням помилок та їх подальшим виправленням	63 сек
Розрахунок методом GOMS	34 сек
Автоматичне заповнення користувацької форми	0.1 сек

Підвівши підсумки можна зробити наступні висновки: люди абсолютно точно роблять помилки, викликане то може бути через втому, неуважність чи просто механічні помилки. На заповнення доволі простої форми, яка не містить в собі коди паспортів або особистих рахунків, було витрачено 63

секунди, якщо зробити помилку, та фактично витрачений час склав 45 секунд, що на 18 секунд швидше.

Простий розрахунок який визначає середнє значення на заповнення форми склав 34 секунди, що на 11 секунд менше ніж фактичне значення. Це є приблизним часом, тобто можна заповнити як швидше, так і довше. В поточному експерименті, цей час був перевищений, через помірну інтенсивність введення інформації.

Фінальним результатом стало автоматичне заповнення, якому був номінально наданий час в 100 мілісекунд, бо уся форма заповнюється миттєво. Можна сказати що прискорення роботи склало 99.9% та повністю виключена вірогідність допущення помилок. Подібне рішення дуже допоможе людям з фізичними вадами, яким необхідно в рази більше часу на заповнення форми.

В ході проведення дослідження була зібрана та оброблена статистична інформація щодо використання тегів у користувацьких формах, імен, що записані у атрибут name та текст, який описує призначення кожного з полів. В результаті аналізу було доведено, що найбільш ефективним є оброблення тексту у формі, через однакові поля, підписані у різних формах і, як наслідок, ці дані людина сприймає однаково.

На основі отриманих даних була запропонована методика, яка дозволяє ефективно заповнювати користувацькі форми автоматично, та структурований користувацький профайл, що містить у собі заздалегідь внесену користувачем особисту інформацію.

Повноцінна програмна реалізація методики використовує профайл користувача, та на основі тих даних, що містяться у файлі заповнюється обрана користувачем форма. Користувацькі дані заповнюються миттєво та не містять жодних помилок, що дозволяє без побоювань вводити конфіденційні банківські та інші юридичні дані.

6 ЕКОНОМІЧНА ЧАСТИНА

6.1 Характеристика науково-дослідної роботи

В економічній частині атестаційної роботи обґрунтовано економічну доцільність впровадження методу автоматичного заповнення користувацьких форм у повсякденному робочому процесі [18].

Досліджено процес автоматизації для існуючого та функціонуючого сайту в мережі інтернет, склад команди, яка бере участь у побудові та реалізації методу автоматичного заповнення існуючого сайту, проведено розрахунок економічної ефективності впровадження.

Впровадження розробленої методики передбачає:

- аналіз предметної галузі;
- створення алгоритму реалізації методу;
- дослідження методів парсингу користувацьких форм;
- визначення методів для проведення експерименту;
- створення методики автоматизації заповнення користувацьких форм;
- реалізація прототипу методики;
- впровадження методики.

Реалізація розробленої методики дозволяє:

- визначити загальні проблеми під час ручного заповнення форм;
- отримати якісну оцінку результатів впровадження методики;
- зменшити час, необхідний для заповнення користувацьких форм особистими даними;
- зменшити кількість помилок та навіть їх повне усунення у випадку якщо метод було правильно використано;
- розробити універсальні рішення для зберігання особистих даних та їх подальшого використання.

6.2 Етапи виконання НДР, їх трудомісткість та заробітна плата

Під час виконання науково-дослідницької роботи було розглянуто існуючі теоретичні підходи в межах предметної області, досліджені основні методи та алгоритми парсингу користувацьких форм та їх обробку. На основі обраних методик було створено гнучку методику для автоматизації заповнення користувацьких форм на web-ресурсах.

Науково-дослідну роботу (НДР) можна умовно розділити на наступні три етапи:

- підготовчий;
- основний;
- заключний.

На стадії підготовчого етапу були виконані підбір критеріїв оцінки ефективності методу та розробка прототипу. Проведено пошук інформації в Internet та у відповідній літературі [19].

Проаналізовано спеціалізовані сайти, які вимагають від користувача введення великої кількості інформації.

На етапі виконання основної частини НДР були виконані такі роботи:

- розглянуто методи парсингу користувацьких форм;
- визначено недоліки та переваги розглянутих методів;
- запропоновано новий метод, що нівелює усі недоліки;
- створено прототип запропонованої методики;
- виконання замірів часу заповнення форми з використанням методу, та без нього в ручному режимі;
- оцінка якості розробленого прототипу;
- обробка та аналіз результатів експерименту.

У заключній частині здійснюється оцінка ефективності виконання НДР, складання звіту по НДР, захист звіту.

Найбільш складною та відповідальною частиною при плануванні НДР є розрахунок трудомісткості робіт, тому що трудові витрати часто становлять

основну частину вартості науково-дослідних робіт і безпосередньо впливають на строки розробки.

Для виконання роботи було залучено 2 фахівця: front end розробник, та бізнес аналітик.

Середня заробітна плата для front end розробника, за даними з сайту dou.ua, складає 25 000 грн./міс. а бізнес аналітика 23 000 грн./міс. відповідно до рівня спеціаліста який потрібен для даної задачі.

Середньоденна заробітна плата виконавця робіт (Зср.дн.) розраховується за формулою:

$$З_{ср.дн.} = \frac{З_{ср.міс.}}{n}, \quad (6.1)$$

де $З_{ср.міс.}$ – середньомісячна зарплата виконавця роботи;

n – число робочих днів у місяці, ($n = 22$).

Середньоденна заробітна плата front end розробника складає:

$$З_{ср.міс} = \frac{25000}{22} = 1136,36 \text{ (грн.)}.$$

Середньоденна заробітна плата бізнес аналітика складає:

$$З_{ср.міс} = \frac{23000}{22} = 1045,45 \text{ (грн.)}.$$

Етапи виконання НДР, перелік і зміст робіт, трудомісткість їх виконання, заробітна плата виконавців робіт представлені в табл. 6.1.

Таблиця 6.1 – Розрахунок трудовитрат і заробітної плати виконавців робіт

Перелік робіт	Кіль- кість вико- навців	Посада виконавця	Трудоміст- кість робіт, люд. днів	Середньо- денна заробіт- на плата, грн.	Сума заробітної плати, грн.
1	2	3	4	5	6
1. Підготовчий етап					
1.1. Розробка та затвердження ТЗ	1	Бізнес аналітик	2	1045,45	2090,9
1.2 Підготовка довідкових матеріалів та даних для виконання НДР	1	Бізнес аналітик	3	1045,45	3136,35
2. Основний етап					
2.1 Постановка задачі	1	Бізнес аналітик	1	1045,45	1045,45
2.2 Розгляд сучасних методів рендерингу веб форм	2	front end розробник Бізнес аналітик	3	1136,36 1045,45	6545,43
2.3 Аналіз розглянутих методів та відбір за критеріями	2	front end розробник Бізнес аналітик	4	1136,36 1045,45	8727,24
2.4. Створення методики парсингу	1	front end розробник	3	1136,36	3409,08
3 Тестування					
3.1 Відбір тестових користувацьких форм	1	Бізнес аналітик	1	1045,45	1045,45
3.2 Заповнення форм за методикою	1	Бізнес аналітик	1	1045,45	1045,45
3.3 Підтвердження результату	1	Бізнес аналітик	1	1045,45	1045,45
4. Заключний етап					
4.1 Аналіз результатів проведення роботи	1	Бізнес аналітик	2	1045,45	2090,9
4.2 Формування висновків та пропозицій за темою дослідження	1	Бізнес аналітик	1	1045,45	1045,45
4.3 Технічне оформлення звіту виконання НДР	1	Бізнес аналітик	2	1045,45	2090,9
Всього			24		33 318,05

6.3 Розрахунок одноразових витрат на розробку НДР

Собівартість розраховується відповідно до існуючих нормативних актів України. До складу калькуляції входять такі статті витрат:

- матеріальні витрати;
- витрати на оплату праці;
- єдиний соціальний внесок;
- амортизація основних засобів (вартість машинного часу);
- витрати на спожиту електроенергію;
- інші витрати.

До інших витрат відносяться адміністративні витрати (водопостачання, водовідведення, опалення, освітлення) та вартість послуг зв'язку.

Матеріальні витрати – це витрати на матеріали, визначені їх потребою для виконання робіт, і цін, що діють на момент складання калькуляції [20]. Для проведення НДР потрібно: 1 шт. шарикова ручка та 1 шт. блокнот. Наведені матеріальні витрати потрібні для бізнес аналітика.

Матеріальні витрати розраховуються за такою формулою:

$$M = \sum_{j=1}^n Q_j \times C_j, \quad (6.2)$$

де M – сумарні витрати на матеріали, в тому числі малоцінні предмети, що швидко зношуються (носії, папір, канцелярське приладдя тощо), або на літературу, яка необхідна для проведення роботи, тощо;

Q_j – кількість використаних одиниць j -го виду матеріалів, $j = (1n)$;

C_j – ціна одиниці j -го виду матеріалів.

Розрахунок матеріальних витрат представлено в таблиці 6.2.

Витрати на оплату праці розраховуються виходячи з необхідного для виконання робіт складу й кількості працівників, а також із середньомісячної заробітної плати.

Таблиця 6.2 – Розрахунок матеріальних витрат

Найменування	Одиниці вимірювання	Кількість використаних	Ціна за одиницю, грн	Сумарні витрати, грн
Кулькова ручка	шт.	1	8,00	8,00
Блокнот	шт.	1	60,00	60,00
Усього				68,00

Відповідно до проведених розрахунків витрати на оплату праці виконавців роботи дорівнюють 33 318,05 грн.

Єдиний внесок на загальнодержавне соціальне страхування (ЄСВ) – консолідований страховий внесок, збір якого здійснюється в систему загальнообов’язкового державного соціального страхування в обов’язковому порядку і на регулярній основі з метою забезпечення захисту у випадках, передбачених законодавством, прав застрахованих осіб і членів їх сімей на отримання страхових виплат (послуг) за діючими видами загальнообов’язкового державного соціального страхування.

Для об’єкта дослідження ставка єдиного соціального внеску дорівнює 22% від витрат на оплату праці, тобто розмір ЄСВ дорівнює 7 329,97 грн.

При виконанні НДР застосовувалось наступне обладнання: комп’ютер 2шт. вартістю 17000 грн.

Вищенаведене устаткування є власністю організації виконавця, тому доцільно розрахувати суму амортизаційних відрахувань на період виконання НДР. Амортизація основних засобів розраховується за формулою:

$$AB = \sum_{k=1}^L \frac{BO_k}{TE_k} \times T, \quad (6.3)$$

де АВ – сума амортизаційних відрахувань, нарахованих під час проведення науково-дослідницької роботи;

BO_k – вартість основних засобів k-го виду;

TE_k – термін експлуатації основних засобів k-го виду, днів;

T – термін науково-дослідницької роботи, днів;

L – кількість видів обладнання.

Підставивши відомі значення у формулу (6.3), визначимо величину амортизаційних відрахувань. Отже маємо:

$$AB = \frac{17000 \times 21}{540} + \frac{17000 \times 10}{540} = 975,92(\text{грн}).$$

Витрати на використану обладнанням електроенергію розраховуються за формулою:

$$Z_e = M \cdot t \cdot T_{kBm}, \quad (6.4)$$

де M – потужність устаткування, тобто кількість енергії, споживаної за одиницю часу (кВт/година);

t – кількість годин використання устаткування за період проведення науково-дослідницької роботи;

T_{kBm} – тариф, тобто вартість використання 1 кВт електроенергії.

Споживна потужність комп'ютера складає 0,6 кВт за годину. Тариф споживачів за першим класом напруги, тобто 35 кВт та більше), складає 1,96 грн./кВтгодин (без ПДВ). Підставивши значення у формулу (6.4), визначимо величину витрат на спожиту електроенергію:

$$Z_e = 0,6 \times 21 \times 8 \times 1,96 + 0,6 \times 10 \times 8 \times 1,96 = 291,64 \text{ грн.}$$

До інших статей витрат відносяться такі:

- адміністративні витрати: (водопостачання, водовідведення, освітлення, опалення), які прийнято у розмірі 20% від витрат на оплату праці;
- вартість оплати послуг зв'язку.

Вартість оплати послуг зв'язку становитиме:

а) Інтернет – із розрахунку 180 грн. на місяць (безлімітний пакет); всього 180 грн. за 22 дні виконання НДР;

б) телефон – із розрахунку 160 грн на місяць; всього 160 грн. за 22 дні.

За час виконання НДР витрати на відрядження, аутсорсинг, інформаційні послуги та маркетингові заходи не мали місця.

Для виконання НДР використовувалося безкоштовне ПО, наприклад, для розробки прототипів – Sublime Text 3, багатофункціональний текстовий редактор, який має безкоштовну версію. Результати розрахунку кошторису витрат, тобто одноразових витрат, на виконання НДР наведені в табл. 6.3.

Таблиця 6.3 – Кошторис витрат на розробку НДР

№ з/п	Стаття витрат	Сума, грн.
1	Заробітна плата	33 318,05
2	Єдиний соціальний внесок (22,0 % від п.1)	7 329,97
3	Матеріальні витрати	68,00
4	Амортизація основних засобів	975,92
5	Витрати на спожиту електроенергію	291,64
6	Адміністративні витрати (20,0 % від п.1)	6663,61
7	Вартість послуг зв'язку	340
8	Загальні витрати	48987,19

Таким чином, кошторис витрат на виконання даної НДР відбиває сумарні витрати за статтями складає 48987,19 грн.

6.4 Оцінка результатів науково-дослідної роботи

Результат – це завершальний наслідок послідовності дій, виражений якісно або кількісно. В загальному випадку оцінка результатів НДР – це визначення ефективності отриманих рішень порівняно з сучасним науково-технічним рівнем.

Відповідно до теми даної роботи можна зробити висновок про те, що у якості результату впровадження НДР є зменшення часу на заповнення користувацької форми та усунення вірогідності здійснення помилок, що має досить велике значення для користувачів [22-24].

Результат від впровадження НДР визначається за такою формулою:

$$\Delta P_j = |X_{б_j} - X_{н_j}|, \quad (6.5)$$

де ΔP_j – покращення j -ої характеристики досліджуваного процесу за рахунок впровадження результатів НДР ($j=1,m$);

m – кількість досліджуваних характеристик;

$X_{б_j}$ – базове значення j -ої характеристики, тобто до впровадження результатів НДР;

$X_{н_j}$ – нове значення j -ої характеристики після впровадження пропонованих рішень.

У якості досліджуваної характеристики виступає час, який необхідно витратити на повне заповнення користувацької форми. До впровадження методики, на заповнення реєстраційної форми необхідно було витратити від 34 секунд до 63 секунд (якщо допустити помилку і виправити її) [21].

Підставивши відповідні значення часу на заповнення користувацької форми в формулу (6.5), визначимо результат від впровадження НДР у чисельному вигляді.

Для часу, що було розраховано завдяки методології GOMS:

$$\Delta P_j = |34 - 0.1| = 33.9 \text{ (сек)}.$$

Для фактичного часу, що було виміряно секундоміром під час ручного заповнення форми:

$$\Delta P_j = |45 - 0.1| = 44.9 \text{ (сек)}.$$

Для часу, який необхідно було витратити у випадку допущення помилки та її виправленням:

$$\Delta P_j = |63 - 0.1| = 62.9 \text{ (сек).}$$

Розрахунки були оброблені та їх результати наведені у таблиці 6.4.

Таблиця 6.4 – Результат від впровадження НДР

Критерій	Час розрахований за методологією GOMS			Фактичний час для ручного введення			Час з допущенням та виправл. помилок		
	До	Після	Різниця	До	Після	Різниця	До	Після	Різниця
Час на заповнення, сек	34	0.1	33.9	45	0.1	44.9	63	0.1	62.9

Таким чином, можна зробити такий висновок про те, що завдяки результату НДР користувацькі форми на сайтах оптимізовано з точки зору заповнення для кінцевого користувача. Процес реєстрації на сайті або постійної роботи з користувацькими формами стає значно меншим та повністю виключає виникнення помилок. Роботу у цілому можна враховувати ефективною або такою, що має високий науковий та технічний рівень.

ВИСНОВКИ

У даній роботі було досліджено методи автоматичного заповнення користувацьких форм, що дозволило більш детально розглянути поточну проблему з технічної точки зору. Бо головним питанням є ефективність та доречність застосованих методів з ефективним введенням у використання.

Був розроблений та застосований метод для автоматизації заповнення користувацької форми, що надало змогу порівняти різні підходи щодо заповнення форм користувацького інтерфейсу.

Покладена в основу дослідження гіпотеза, була підтверджена.

Для проведення дослідження було вивчено необхідний теоретичний матеріал, була зібрана статистика стосовно існуючих користувацьких форм, зібрана статистика щодо використання різних типів полів, того які ім'я надаються атрибутам та аналіз тексту, яким підписані поля, що довело думку про те, що самі поля в формах підписують більш шаблонно та подібно одне одному. Була доведена економічна ефективність впровадження даного методу в робочі процеси. Економічні переваги були розраховані та підтверджені.

В результаті атестаційної роботи, було запропоновано метод, який охоплює абсолютно всі можливі комбінації імені атрибуту name та оточуючих елементів які є підписом поля, що призначений на прочитання цього тексту користувачем та створення асоціативного зв'язку з тим, які дані необхідно ввести в конкретне поле.

Запропонований метод повністю виключає людський фактор з процесу заповнення форми, що в свою чергу не допускає утворення помилок, та відповідно наслідків цього.

На основі економічних розрахунків можна сказати, що дана магістерська дослідницька робота позитивно впливає на економічну складову підприємства, на якому вона може бути впроваджена.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні рекомендації щодо організації, структури та змісту навчальних занять / Н.С. Лесна, В.В. Логвин, І.О. Милотченко. Харків: ХНУРЕ, 2009. 42 с.
2. Раскин Дж. Интерфейс: новые направления в проектировании компьютерных систем. 2015. 272 с.
3. Гнатієнко Г.М., Снитюк В.Є. Експертні технології прийняття рішень. Київ: Знання, 2008. 444 с.
4. Стейнер К. Тотальна автоматизація. Як комп'ютерні алгоритми змінюють світ. Київ: Знання, 2018. 280 с.
5. Зайцев Г.М., Лапшин А.Б. Организация планирования научно-технического прогресса. М., 1970.
6. ГОСТ 28195-89. Оценка качества программных средств. Общие положения. Введ. 28.07.1989. М.: Изд-во стандартов, 1989. 26 с.
7. Коутс Р., Влейминк И. Интерфейс «человек-компьютер». М.: Мир, 1990. 501 с.
8. Гарретт Д. Веб-дизайн: книга Джесса Гарретта. Элементы опыта взаимодействия. Символ-Плюс, 2008. 192 с.
9. Кита Дж. HTML5 для веб-дизайнеров. М.: ЛКИ, 2012. 272 с.
10. Четыркин Е.М. Статистические методы прогнозирования. 2-е изд. М.: Статистика, 2007. 145 с.
11. Айвазян С.А., Енюков И.С., Мешалкин Л.Д. Прикладная статистика: основы моделирования и первичная обработка данных. М.: Финансы и статистика, 1983. 124 с.
12. Тоценко В.Г. Методы и системы поддержки принятия решений. Алгоритмический аспект. К.: Наукова думка, 2002. 381 с.
13. Поліграфічні, мультимедійні та web-технології: матеріали Молодіжної школи-семінару V Міжнародної науково-технічної конференції (3

листопада 2020, м. Харків) / В.П. Ткаченко, І.Б. Чеботарьова, О.В. Вовк, Г.В. Бельчева. Харків: ХНУРЕ, 2020. Т2. 132 с.

14. Глотов В.А., Гречко В.М., Павельев В.В. Многокритериальные задачи принятия решений. М.: Машиностроение, 1978. 180 с.

15. Методические рекомендации к выполнению экономической части дипломных проектов, работ для студентов дневной и заочной форм обучения всех специальностей / Л.В. Соколова, О.І. Горбач, С.В. Гришко, Є.В. Диденко, Л.В. Левченко, Г.М. Путятин, В.Г. Харченко. ХНУРЕ. 2010.

16. Райзберг Б.А., Лозовский Л.Ш., Стародубцева Е.Б. Современный экономический словарь. 2-е изд. М.: ИНФРА-М, 1999. 479 с.

17. Продиус Ю.И. Экономика предприятия: учеб. Пособие. Харьков: Одиссей, 2007. 416 с.

18. Волков В.П., Павлов І.Д., Павлов Ф.І. Аналіз і оцінка ефективності реалізації складних проектів: монографія. Запоріжжя: Запорізь. нац. ун-т, 2012. 315 с.

19. Левинсон А.Г., Стучевская О.И. Фокус-группы: эволюция метода (обзор дискуссии на конференции ESOMAR) // Мониторинг общественного мнения: экономические и социальные перемены. 2003. № 1 (63). С. 46-55.

20. Большой экономический словарь / под ред. А. Н. Азрилияна. 2-е изд. М.: Институт новой экономики. М.: Планета, 1997. 231 с.

21. Экономика: учебник / под ред. А.С. Булатова. 2-е изд. М.: БЕК, 1997. 196 с.