Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
(повне найменування вищого навчального закладу)

Факультет *Інфокомунікацій*
(повне найменування інституту, факультету (відділення))

Кафедра *Інформаційно-мережної інженерії*
(повна назва кафедри (предметної, циклової комісії))

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Дослідження і аналіз методів тестування форм авторизації

(тема)

Виконав: студент 2 курсу групи *ІМІм-20-1*
Спеціальність *172. Телекомунікації та радіотехніка*
(код і повна назва спеціальності)
Тип програми *освітньо-професійна*
(освітньо-професійна або освітньо-наукова)
Освітня програма *Інформаційно-мережна інженерія*
(повна назва освітньої програми)
Євченко А.В.
(прізвище та ініціали)

Керівник *ст. викл. Калюжний М.М.*
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри _____ *Безрук В.М.*
(підпис) (прізвище, ініціали)

2021 р.

Не містить відомостей, заборонених
до відкритого публікування

Керівник_____ /*М.М.Калюжний*

Студент _____ / *А.В. Євченко*

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

(повне найменування вищого навчального закладу)

Факультет *Інфокомунікацій*

Кафедра *Інформаційно-мережної інженерії*

Рівень вищої освіти *другий (магістерський)*

Спеціальність *172. Телекомунікації та радіотехніка*

(код і повна назва)

Тип програми *освітньо-професійна*

(освітньо-професійна або освітньо-наукова)

Освітня програма *Інформаційно-мережна інженерія*

(повна назва)

ЗАТВЕРДЖУЮ

Зав. кафедри «ІМІ»

_____ проф. Безрук В.М.

(підпис)

«*29*» *жовтня* 2020 р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові *Євченко Антону Володимировичу*

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження і аналіз методів тестування форм авторизії*

керівник роботи *ст. викл. Калюжний Микола Михайлович*

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

затверджені наказом ВНЗ від «__» _____2021 р. № ___

2. Термін подання студентом роботи до екзаменаційної комісії _. 2021 р.

3. Вихідні дані до роботи авторизація, аутентифікація

4. Перелік запитань, що необхідно опрацювати в роботі

Вступ

1 Тестування та його види

2 Аутентифікація та авторизація

3 Порівняльне тестування довільних форм

4 Порівняльний аналіз форм авторизації

Висновки

5. Перелік графічного матеріалу із зазначенням креслень, схем, плакатів,

комп’ютерних ілюстрацій *слайди презентації в форматі Power Point: 17 слайдів*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз завдання та літературних джерел		
2	Написання розділу 1		
3	Написання розділу 2		
4	Написання розділу 3		
5	Написання розділу 4		
6	Підготовка публікації		
7	Висновки		
8	Оформлення пояснювальної записки		

Дата видачі завдання _____ 2021 р.

Студент _____ Євченко А.В.
(підпис) (прізвище та ініціали)

Керівник роботи _____ ст. викл. Калюжний М.М.
(підпис) (посада, прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка: 67 с., 20 рис., 13 посиланя, 3 додатка.

Мета роботи – дослідження і аналіз методів тестування форм авторизації

Сучасне тестування існує не лише як допоміжна дія при створенні проекту, а окремий великий обсяг робіт в протягом усього циклу життя проекту.

Забезпечення якості — найширше з понять, яке передбачає в собі певні дії, що йдуть паралельно й охоплюють розробку програмного забезпечення, його випуск і подальше використання. Окремою його частиною є контроль якості, його суттю є слідкування за розробкою для визначення чи виконує продукт необхідні норми і перевірка на готовність.

Тестування форм авторизації передбачає визначення готовності проекту забезпечувати ідентифікованому користувачу доступ до інформації після аутентифікації.

ЗАБЕЗПЕЧЕННЯ ЯКОСТІ, КОНТРОЛЬ ЯКОСТІ, АВТОРИЗАЦІЯ, АУТЕНТИФІКАЦІЯ

ABSTRACT

Explanatory note: 67 pp., 20 fig., 13 reference, 3 app.

Object of work – research and analysis of methods of testing forms of authorization.

Modern testing does not exist only as an ancillary action in the creation of the project, but a separate large amount of work throughout the life cycle of the project.

Quality Assurance - an extension with an understanding that has certain actions in them, which in parallel cover the development of software, its release and subsequent use. A separate part of it is Quality Control, its essence is to follow the work to determine the cleaning of the product of the required standard and check for readiness.

Testing of authorization forms determines of project readiness, which provides an identified user, access to information after authentication.

QUALITY ASSURANCE, QUALITY CONTROL, AUTHORIZATION , AUTHENTICATION

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....9
ВСТУП.....10
1 ТЕСТУВАННЯ ТА ЙОГО ВИДИ.....11
1.1 Функціональне тестування.....11
1.2 Нефункціональне тестування.....12
1.3 Тестування пов’язанні з внесенням змін.....14
2 АУТЕНТИФІКАЦІЯ ТА АВТОРИЗАЦІЯ.....15
2.1 Аутентифікація.....16
2.1.1 Аутентифікація з допомогою цифрового підпису.....17
2.1.2 Аутентифікація з допомогою паролю.....17
2.1.3 SMS аутентифікація.....19
2.1.4 Біометрична аутентифікація.....19
2.2 Авторизація.....21
3 ПОРІВНЯЛЬНЕ ТЕСТУВАННЯ ДОВІЛЬНИХ ФОРМ.....25
3.1 Тестування Automationpractice.....25
3.1.1 Створення облікового запису.....25
3.1.2 Авторизація на сайті Automationpractice.....27
3.2 Перевірка сайту Kakao Games.....29
3.2.1 Створення аккаунту.....29
3.2.2 Авторизація на сайті Kakao Games.....32
4 РОЗРОБКА МЕТОДУ ТЕСТУВАНЯ ФОРМ АВТОРИЗАЦІЇ.....37
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....40
ДОДАТОК А – КРОССБРАУЗЕРНІСТЬ.....41
ДОДАТОК Б – ІНТЕРФЕЙС, ЛОКАЛІЗАЦІЯ ТА UI.....43
ДОДАТОК В – СЛАЙДИ ПРЕЗЕНТАЦІЇ.....48
ДОДАТОК Г – АПРОБАЦІЯ РЕЗУЛЬТАТІВ.....48

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

QA	Quality Assurance	Забезпечення якості
QC	Quality Control	Контроль якості
OTP	One Time Password	Одноразовий пароль
IT	Information Technology	Інформаційні технології
SFA	Single-factor authentication	однофакторна аутентифікація
2FA	Two-factor authentication	двофакторна аутентифікація
MFA	Multi-factor authentication	багатофакторна аутентифікація
БД		База даних
ПЗ		Програмне забезпечення

ВСТУП

На сьогоднішній день широкого поширення набуває ІТ сфера й все більше людей намагаються приєднатися до неї. Це пов’язано через швидке поширення персональних пристроїв для доступу до мережі. Такі пристрої дозволяють не лише переглядати доступну інформацію, а й бути носіями критично важливої інформації для власника. Для цього компанії розробляють чисельні програмні та мобільні додатки що дозволяють не тільки зберігати, але й активно використовувати в повсякденному житті електронні документи, що посвідчують особу або документи на авто. Одним з таких додатків є знайома на сьогодні всім «ДІЯ».

Якість кінцевого продукту ІТ – індустрії характеризують не лише програмісти. Тестування програмних додатків на етапах релізу дозволяють не випускати на ринок недопрацьоване ПЗ. Саме тому, серед численних професій що пропонуються, багато студентів та не знайомих з цією індустрією людей обирають QA-інженерів - фахівців, які тестують проект на етапі розробки, щоб швидко виявити помилки та відхилення від заданого курсу та вчасно координувати процес розробки. Адже цей вид діяльності доступний багатьом через досить невеликий поріг входження, проте має безліч перспектив через можливість ознайомлення з повним циклом підтримки проекту від його ідеї до втілення в життя і обслуговування, а також кожен проект має бути перевірено та виправлено й тому компанії активно шукають нових спеціалістів на нові проекти. Слід зазначити, що фінансове забезпечення таких професій також досить велике у порівнянні з іншими професіями на ринку праці.

До зони впливу таких професій можна віднести тестування ПЗ, веб-додатків, а також можна окремою частиною розглянути перевірку процесу авторизації.

Для оцінювання якості створеної форми авторизації до неї можна застосувати функціональне і не функціональне тестування.

Перевірка на локалізацію поля аутентифікації, їх інтерфейс, зручність та інтуїтивну зрозумілість, проте перш за все, звісно нас цікавить функціонал, адже цей процес призначений для захисту певних даних від сторонніх осіб та навпаки, надання повноважень доступу необхідним. Тож нам необхідно бути певним, що користувач матиме змогу зареєструватися, і підтвердити свою особу так, щоб це не було можливим для інших.

В роботі буде розглянуто застосування кількох методів тестування форм авторизації та їх порівняльний аналіз для виявлення недоліків та можливості поєднання основних позитивних елементів. Розглянуто критерії та рівні тестування.

Основною метою проекту є уніфікація процесу тестування шляхом узагальнення основних елементів відомих розглянутих методів, критеріїв та функціональних засобів.

1 ТЕСТУВАННЯ ТА ЙОГО ВИДИ

Якість програмного забезпечення (Software Quality) — є характеристики програмного забезпечення, пов'язані з можливостями виконати визнані та очікувані потреби. До нього відносяться: QA, QC, та тестування ПЗ.

Тестування програмного забезпечення – це процес аналізу програмного забезпечення для виконання наступних завдань: розпізнання помилок та перевірка на якість. Тестування програмного забезпечення також є контролем за відповідністю між фактичною та очікуваною поведінкою програми, воно включає: діяльність з планування роботи, проектування тестів, виконання тестів та аналізу отриманих результатів.

Мета тесту:

1. Збільшити потенціал програми для тестування працювати коректно в «будь-якій ситуації».

2. Збільшити ймовірність того, що ваша програма призначена для тестування відповідає всім описаним вимогам.

3. Виконайте повну перевірку програми протягом зазначеного часу.

Усі види тестування в залежності від мети використання можна поділити на три групи: функціональне, нефункціональне та тестування пов’язане з змінами в ПЗ.

1.1 Функціональне тестування

Функціональне тестування в свою чергу поділяється на: власне функціональне, тестування безпеки та тестування взаємодії.

Власне функціональне розглядає попередньо визначені дії на основі аналізу специфікації компоненту або системи в цілому. Так як це тестування використовує функції системи, то й його виконання можливе протягом усіх етапів. Його перевага — імітація використання ПЗ, проте серед недоліків —

вірогідність надлишкового тестування та можливість пропустити логічні похибки.

Тестування безпеки — використовується як можна здогадатись для виявлення дірок безпеки та рисків пов’язаних з захистом програми, або конфедераційних даних. Серед його принципів розрізняють: конфіденційність(частина інформації може бути доступна лише авторизованим користувачам), цілісність (данні можуть редагуватись лише окремими, визначеними користувачами, а також можливість їх відновлення в випадку ураження) і доступність(надання ресурсів авторизованому користувачу/приладу/об'єкту).

Тестування взаємодії — перевірка ПЗ на сумісність з системою та її компонентами.

1.2 Нефункціональне тестування

До нефункціонального тестування відносяться: тестування продуктивності, встановлення, зручності користування, тестування на відмову та відновлення.

1) Тестування продуктивності перевіряє як ПЗ реагує на різні види навантажень при використанні і в свою чергу поділяються на:

- тестування навантаження;
- стресове;
- тестування стабільності і надійності;
- об'ємне тестування.

2) Тестування встановлення використовується для перевірки, як ПЗ встановлюється/налаштовується, оновлюється і видаляється.

3) Тестування зручності користування необхідно для дослідження на інтуїтивну зрозумілість для збільшення конкурентоспроможності, адже замовнику зручніше використовувати те, що не потребує ознайомлення з самого початку.

Критерії тестування зручності користування:

● продуктивність, результативність (ефективність), тривалість часу необхідного користувачу для виконання базових дій;

● точність – кількість помилок, які користувачі роблять під час роботи використовувати? (Чим менше, тим краще);

● активація в пам’яті (викликання) – наскільки добре користувач запам’ятовує яким чином виконувалося завдання(Після паузи повторне виконання операції має бути швидше ніж у нового користувача).

4) Тестування на відмову та відновлення. Цей тип тестування перевіряє продукт на можливість протидіяти й відновлюватись після збою, виниклих через помилки у ПЗ, приладів, або зв’язку(мережі/струм, тощо.).

5) Конфігураційне тестування використовується для того, щоб визначити конфігурацію оптимального обладнання, яке спроможне надавати характеристики продуктивності і часу реакції на запити.

Для клієнт-серверного додатку тестування поділяють на окремих 2 рівня.

Взаємодія затвердженого програмного забезпечення тестується на першому рівні (сервер).

1. Апаратне забезпечення (тип і кількість процесорів, обсяг пам'яті, властивості мережі / мережевий адаптер тощо).

2. Програмне забезпечення (операційна система, драйвери, бібліотеки, стороннє програмне забезпечення що впливає на роботу.

Основна мета тут — перевірити, щоб визначити конфігурацію пристрою, що відповідає необхідним якісним характеристикам.

На другому рівні (клієнт) програмне забезпечення тестується с точки зору кінцевого користувача: чи відтворюється функціональність на інших конфігураціях та зручність користування. Зазвичай перевіряють різні за типом і розрядністю операційні системи, різні розширення екрану, версії бібліотек та драйверів, а також браузери чи відеоадаптери (при тестуванні веб-додатків чи ігор).

1.3 Тестування пов’язанні з внесенням змін.

При виправленні помилок в роботі, або при оновленні ПЗ зазвичай виконується повторне тестування для перевірки чи не з’явилися нові помилки, а також чи дійсно знешкоджені старі.

Існує декілька видів цього тестування:

1. Димове

Цей вид можна розглядати як швидка перевірка на працездатність нового/виправленого ПЗ. Для цього виконуються основні функції програми з метою знаходження блокуючих, або критичних дефектів. У випадку їх відсутності тестування вважається успішним і програмне забезпечення передається на подальшу, більш глибоку перевірку. В інакшому — на допрацювання.

2. Регресійне тестування.

Цей тип тестування використовується для перевірки чи зберігається вже робочий функціонал після внесення змін в додаток, або оточуюче середовище(зміна веб серверу, баз даних, операційної системи, тощо..)

Метою цього виду тестування є підтвердження того, що:

- помилка була виправлена;
- після зміни в ПЗ старі помилки не почали відтворюватись;
- зміна у коді ПЗ не вплинула на його окремі частини.

Часто регресійне тестування плутають з повторним, проте повторне використовується для підтвердження виправлення помилки, а регресійне для перевірки чи не вплинули зміни у коді негативно на проект загалом.

3. Перевірка виправлень(санітарне).

Санітарне тестування — вузьконаправлене тестування, яке виконується лише з метою перевірки певної функції чи відповідає вона заявленим специфікацією умовам.

2 АУТЕНТИФІКАЦІЯ ТА АВТОРИЗАЦІЯ

Перш ніж перейти до розглядання форм авторизації, необхідно визначити чим відрізняється авторизація, аутентифікація та ідентифікація. Усі три цих процеси використовуються для захисту наших даних від сторонніх людей:

1. Ідентифікація — процес розпізнання особи за її ідентифікатором.
2. Аутентифікація — процедура перевірки справжності, доказ, що користувач саме той, за кого себе видає.
3. Авторизація — керування доступом до певного захищеного ресурсу.

Поняття аутентифікація і авторизація тісно пов’язані один з одним. Після вводу імені користувача та паролю система перевіряє, чи дійсно пароль “*****” відповідає обліковому запису “_____” й таким чином виконується аутентифікація. Якщо пара “****” і “_____” сумісні то система надає користувачу доступ і таким чином відбувається авторизація.

Аутентифікація використовується для підтвердження вже зареєстрованої особи. Якщо ідентифікатор і пароль збігаються із записами, що зберігаються в системній базі даних, доступ надається користувачеві. Якщо дані введені неправильно, програма ініціює попередження безпеки та блокує введення. Якщо спроба не вдається кілька разів, система сама блокує обліковий запис.

Зазвичай коли недостатньо звичайної аутентифікації для забезпечення безпеки входу користувача в систему використовують додаткові категорії факторів:

- однофакторна аутентифікація (SFA) — це основний традиційний метод аутентифікації, який використовує лише одну категорію. Найпоширенішим прикладом SFA є облікові дані, пов’язані з введенням звичайного імені користувача та пароля.

- двофакторна аутентифікація (2FA) — це двохетапний процес перевірки, який враховує два різних типи облікових даних користувача. На додаток до

імен користувачів та паролів, системі може знадобитися спеціальний код, надісланий SMS або електронною поштою, щоб забезпечити більший захист.

- багатофакторна аутентифікація (MFA) — це сучасний метод аутентифікації, який використовує два, три (або більше) рівні безпеки. Щоб усунути слабкі сторони системи, всі рівні категорій повинні бути незалежними один від одного.

Фінансові установи, банки та правоохоронні органи використовують багатофакторну аутентифікацію для захисту своїх даних від потенційних загроз. (як приклад можна розглянути використання банківських карток — необхідна й наявність картки і пароль до неї)

Після аутентифікації при авторизації система перевіряє який обсяг повноважень надати користувачу, також залежно від рівня безпеки можуть бути декілька факторів перевірки на дійсність. Це можна порівняти з списками наявності студентів на парі. Спочатку перевіряють список студентів, що б підтвердити наявність студента в групі, і після того перевіряють чи дійсно студент присутній.

2.1 Аутентифікація

Під час входу в систему користувачі можуть отримати повідомлення про те, що аутентифікація недоступна. У цьому випадку відвідувач повинен знову спробувати пройти автентифікацію, дотримуючись вимог безпеки, відновити забутий пароль або логін, або зв’язатися з технічною підтримкою по телефону або електронною поштою. Повідомлення «Аутентифікація недоступна» надсилається відвідувачам сайту або користувачам сервісу, коли:

- невірно введені дані імені користувача, або паролю;
- користувач був заблокований;
- права користувача на доступ до інформації були анульовані;
- помилка в ПЗ та обладнанні;
- спроба отримати доступ не згідно з встановленим графіком.

Для вашої зручності розроблено різні типи режимів аутентифікації для використання доступних пристроїв і забезпечення відповідності вимогам безпеки. Зазвичай використовуються деякі комбінації цих режимів. Існують такі види:

- за способом доступу: онлайн та офлайн;
- залежить від способу диференціації прав: дискреційний та обов'язковий, залежно від ролі, контексту чи сітки;
- за типом коду: пароль підключення, біометричні дані, електронний ключ, IP-адреса, динамічний пароль, унікальний елемент (паскарт);
- залежно від кількості тестів: один крок і багатокроковий.

Аутентифікація може виконуватися декількома способами, серед них: цифровий підпис, пароль, біометричні дані, певний предмет, або навіть геолокація.

2.1.1 Аутентифікація з допомогою цифрового підпису

Процес віддаленого підпису важливих документів (банківських) став дуже поширеним останнім часом. Аутентифікацію з допомогою цифрового підпису розрізняють на три основні види за процесом підтвердження даних.

1. Простий цифровий підпис, коли з допомогою коду/паролю чи інших речей підтверджується факт підпису певної особи.

2. Некваліфікований цифровий підпис, коли підпис отримано у результаті криптографічних перетворень інформації з допомогою ключа електронного підпису і дозволяє визначити особу, що підписує документ;

3. Кваліфікований цифровий підпис, який відрізняється від некваліфікованого тим, що ключ перевірки підпису вказано у кваліфікаційному сертифікаті.

2.1.2 Аутентифікація з допомогою паролю

Аутентифікацію з допомогою паролю поділяють на два основних типи - одноразові та багаторазові паролі.

Цей вид аутентифікації вже було коротко розглянуто вище. Користувач здійснює вхід на основі зареєстрованих ім’я користувача(логіном) і паролем, система порівнює введені дані з даними у своїх базах й надає доступ в разі відповідності.

Недоліком такої системи є те, що при отриманні порушником паролю, він матиме змогу доступу, доки пароль не буде змінено. Тож для усунення цієї проблеми було вирішено користуватися одноразовими паролями (OTP). Суть цього методу полягає в тому, що пароль дійсний лише для одного підключення до системи і що новий пароль потрібен для кожного наступного запиту на доступ. Механізм одноразової аутентифікації пароля може бути реалізований як апаратним, так і програмним забезпеченням.

Технології одноразового пароля можна класифікувати так:

1. Використовуйте один і той же генератор псевдовипадкових чисел у користувача та в системі.

2. Використання тимчасових міток в поєднанні з універсальною системою часу.

3. Використовуйте базу даних випадкових паролів для уніфікації суб’єкту і системи.

У першому випадку, згенерований користувачем пароль, може бути переданий системі за новим запитом, використовуючи односторонню функцію послідовно або на основі унікальної інформації з попереднього запиту.

Прикладом другого є SecurID. Він заснований на використанні апаратних ключів і синхронізації часу. Аутентифікація заснована на генерації випадкових чисел через певні проміжки часу. Унікальний закритий ключ зберігається лише на системних та апаратних пристроях користувача. Коли людина запитує доступ до системи, їй буде запропоновано ввести PIN-код і випадково

згенерований номер, які будуть відображатися на апаратному пристрої в цей момент. Система порівнює введений PIN-код із приватним ключем тестувальника з бази даних і генерує випадкове число на основі налаштувань приватного ключа бази даних і поточного часу. Потім перевіряється згенерований номер та ідентифікатор введеного номера користувача.

Третій метод використовує спільну базу даних для системи і користувача. Кожен пароль згаданий у БД можна використати лише раз, тож якщо зловмисник здобу пароль користувач нічого не втрачає, адже пароль вже втрачає свою актуальність.

2.1.3 SMS аутентифікація.

Ці методи аутентифікації включають наступні кроки:

- введіть ім'я користувача та пароль;
- відразу після цього PhoneFactor (служба безпеки) надсилає єдиний ключ входу у вигляді SMS;
- отриманий ключ використовується для аутентифікація.

Перевага цього методу полягає в тому, що ключ отримується без проходження каналу по якому здійснюється аутентифікація. Це практично виключає атаки «людина посередині». Запит на введення PIN-коду вашого мобільного пристрою може забезпечити додатковий рівень безпеки. У зв’язку з цим цей метод широко використовується в інтернет-банкінгу.

2.1.4 Біометрична аутентифікація

Методи аутентифікації, засновані на вимірюванні персональних біометричних параметрів, забезпечують майже 100% ідентифікацію та вирішують проблему втрати паролів та персональних ідентифікаторів. Прикладами реалізації цих методів є системи ідентифікації користувачів на основі візерунків райдужної оболонки ока, відбитків долонь, форми вух,

інфрачервоних зображень капілярів, почерку, запахів, мовних тонів і навіть ДНК. Новим напрямком є використання біометричних властивостей у чіпових платіжних картках, токенах доступу та елементах стільникового зв'язку. Наприклад, під час оплати в магазині власник картки прикладає палець до сканера, щоб переконатися, що карта належить йому.

Найпоширеніші біометричні атрибути:

- **відбитки пальців.** Ці сканери невеликі, універсальні та відносно недорогі. Біологічна відтворюваність відбитків пальців становить від 10 до 5%. У зв’язку з великою кількістю призначених архівів електронних відбитків пальців, наразі його просувають правоохоронні органи;
- **форма руки.** Якщо робота сканера пальців утруднена через забруднення або травму, використовується відповідний пристрій. Біологічна відтворюваність форми руки становить приблизно 2%;
- **райдужка ока.** Ці пристрої мають найвищу точність. Теоретична ймовірність того, що обидва іриси збігаються, дорівнює 1078 році.;
- **розпізнавання обличчя.** Системи, засновані на цьому підході, дозволяють за певних умов ідентифікувати людину з похибкою 3% і менше. Залежно від методу впізнати людину можна на відстані від 0,5 метра до кількох десятків метрів. Цей метод практичний, оскільки його можна реалізувати за допомогою стандартних засобів (наприклад, веб-камери). Більш складний процес вимагає більш складного пристрою. Недоліком деяких (але не всіх) методів є заміна. Ідентифікацію можна зробити, замінивши обличчя справжньої людини її фотографією;
- **голос.** Голосові тести корисні для використання в телекомунікаційних програмах. Необхідна 16-розрядна звукова карта і конденсаторний мікрофон коштують менше 25 доларів. Імовірність помилки становить 2-5%. Ця технологія підходить для тестування мови на телефонних каналах зв'язку і є надійнішою, ніж частотний набір для персональних номерів. Сьогодні лідерство розвивається, щоб відрізнити людину від її стану за голосом – вона засмучена, хвора, говорить правду, але не сама;

У той же час біометричність має деякі недоліки:

- Біометрична модель порівнюється не з результатом первинної обробки характеристик користувача, а з тим, що досягло місця порівняння. Під час транспортування може статися багато речей.
- Базу шаблону може змінити зловмисник.
- Слід враховувати відмінності у використанні біометричних даних на контрольованих територіях, під безпечним спостереженням і в «польових» умовах. Наприклад, якщо ви можете піднести манекен до свого пристрою для сканування.
- У зв’язку з зміною біометричних даних деяких людей (через старіння та травми, опіки, порізи, хвороби, відключення тощо), база даних шаблонів потребує регулярного обслуговування користувачів та адміністраторів. Ці два елементи викликають специфічні проблеми.
- Якщо біометричні дані будуть вкрадені або скомпрометовані, вони в принципі залишаться на все життя. Паролі не є надійними, але їх можна змінити в крайньому випадку. Проте ви не можете швидко змінити пальці, очі чи голос.
- Біометрична властивість є унікальним ідентифікатором, але її не можна зберігати в секреті.

Таким чином біометричні показники також не є універсальними показниками ідентифікації.

2.2 Авторизація

У цьому розділі буде розглянуто найпоширеніший варіант з використанням логіну та багаторазового паролю.

На сьогодні багато людей використовують авторизацію щоденно. Інтернет-банкінг, соціальні мережі, робота, тощо. Тож жодному з нас не потрібно нагадувати про те що для цього зазвичай ми використовуємо пошту, або логін та пароль. Для тестування форми авторизації необхідно визначити які

данні можна в неї внести, а також які умови мають виконувати користувачі при створенні аккаунту та його використанні.

На рисунку 2.1 зображена звична нам форма авторизації у популярній соціальній мережі Facebook. Для входу необхідно ввести вже зареєстровані дані пошти/логіну та пароль.

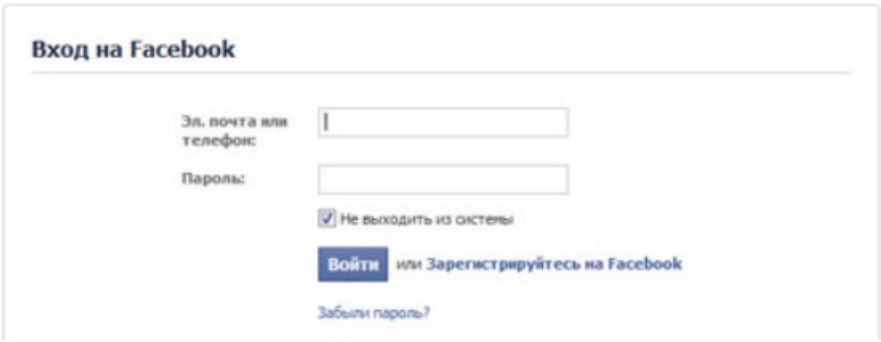

Рисунок 2.1 - Авторизація у соціальній мережі Facebook

Проте, іноді авторизація виконується окремо, наприклад пошта Gmail (рис. 2.2), в якій окремо перевіряється спочатку чи існує пошта, і після цього, якщо запит схвалено, перевіряється чи сумісні введений пароль і пошта у минулому запиті.

Зазвичай використовують перший тип, інколи логіном виступає пошта, інколи у якості логіну — номер телефону, або сам логін вигаданий користувачем.

При створенні логіну можна використовувати літери, цифри та деякі знаки(“.”, “@”, “—”,”_”, тощо) залежно від його форми. Також необхідно звернути увагу, що деякі сайти не чутливу до регістру і не важливо чи великі літери, чи ні, проте досить часто “Login” та “lOGin” два зовсім різних логіну і можлива помилка.

З паролем регістр завжди чутливий, і при створенні аккаунту необхідно звернути увагу на умови.

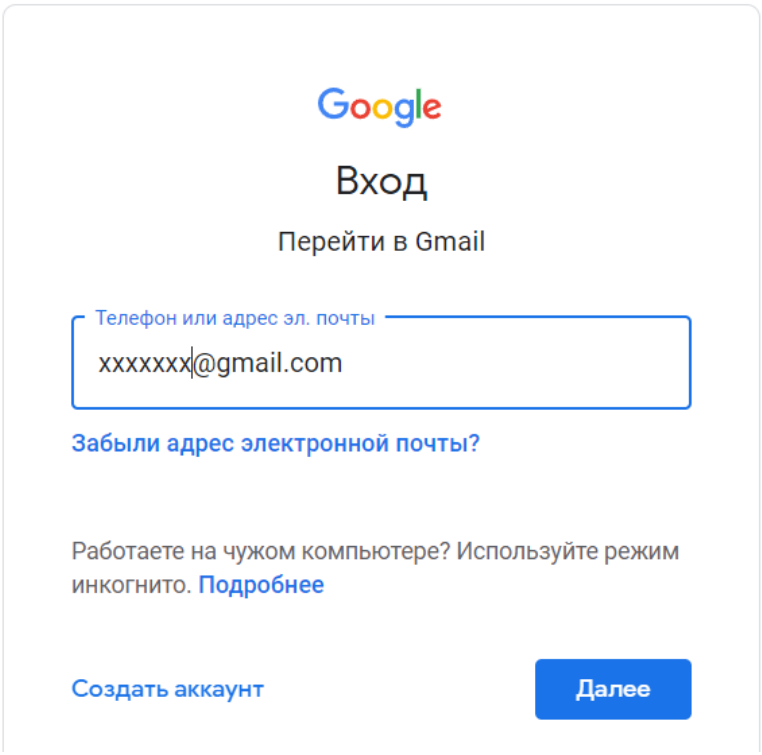

Рисунок 2.2 - Авторизація у поштовій скриньці Gmail

На рисунку 2.3 зображено умови для створення паролю на сайті “Kakao Games”. Серед їх умов:

- від 8 до 16 символів;
- хоча б одна цифра;
- хоча б по одній великій та маленькій літері;
- та хоча б один з вказаних спецсимволів.

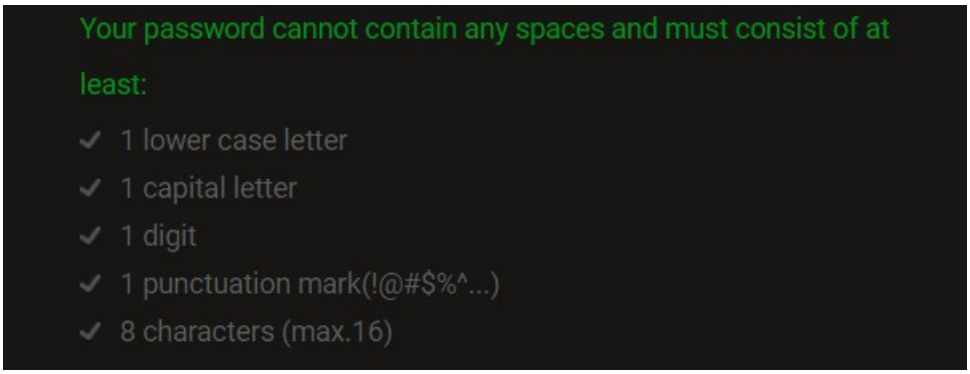

Рисунок 2.3 - Умови для створення паролю

Звичайно не усі сайти вимагають виконання кожного з згаданих пунктів, наприклад використання спецсимволів, або наявність великої літери, проте їх використання надає більше безпеки у зв’язку з збільшенням можливих комбінацій.

Також для тестування необхідно знати стандарти електронних пошт. Пошта може складатися з двух частин — локальної та доменної. Локальна частина може складатися з 64 символів, серед яких дозволено:

- великі і малі літери англійського алфавіту(деякі ресурси, наприклад Яндекс пошта пропонують і російські літери);
- цифри від 0 до 9;
- символи !#$%&'*+-/=?^_`{|}~ (ASCII: 33, 35-39, 42, 43, 45, 47, 61, 63, 94-96, 123-126);
- символ “.” за винятком використання на початку/в кінці пошти, а також двох підряд, якщо вони не зустрічаються в “ ”;
- також такі спецсимволи “ “ и "(),:;<>@[\] за умовою, що перед ними використано “, або \.

Системи, надсилання пошти повинні бути здатні обробляти вихідну пошту всім допустимих адрес. На відміну від відповідних стандартів деякі дефектні системи роблять деякі законні адреси недійсними і не в змозі обслуговувати пошту для цих адрес. Hotmail, наприклад, відмовляється надсилати пошту на будь-яку адресу, що містить будь-який з наступних стандартів допустимих символів: !#$%*/?^`{|}~

На рисунку 2.4 зображені приклади пошт які можна використовувати, і приклад пошти з помилкою.

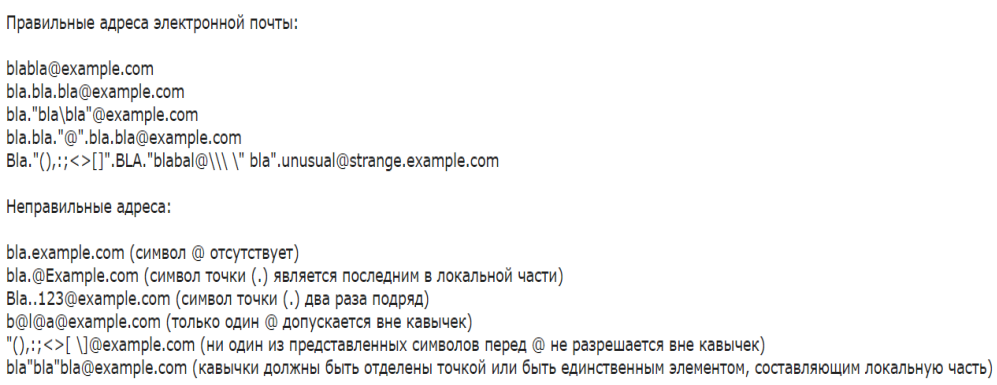

Рисунок 2.4 - Приклад правильних та не правильних поштових адрес

Доменна адреса може містити 256 символів, з урахуванням “@” і може складатися або з адреси сайту, або з IP адреси безпосередньо оточеної [], наприклад “@[192.168.0.1]”.

3 ПОРІВНЯЛЬНЕ ТЕСТУВАННЯ ДОВІЛЬНИХ ФОРМ

Прикладом порівняння тестування форми аутентифікації було обрано Automationpractice, який використовують для практики певних навичок в тестуванні та вже згаданий раніше, відлагоджений kakao games.

3.1 Тестування Automationpractice

Вікно авторизації складається з 2х частин (рис. 3.1) створення аккаунту і безпосередньо авторизації в кабінеті.

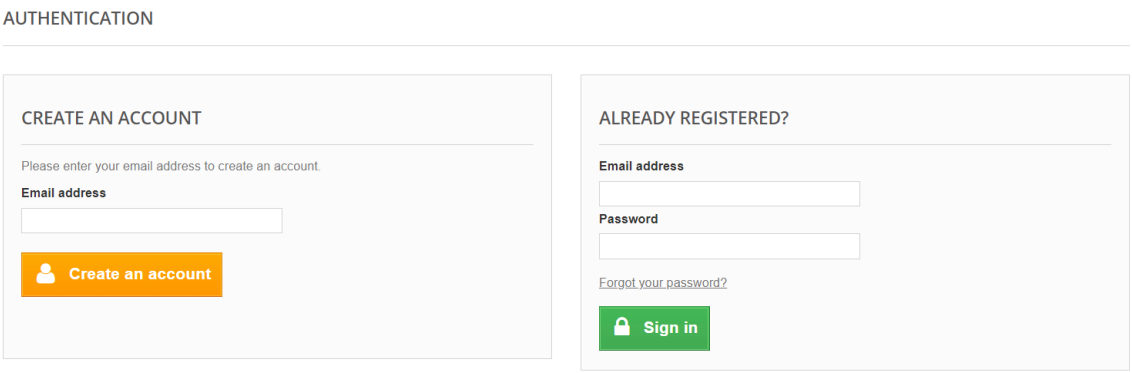

Рисунок 3.1- Вікно аутентифікації

В ході тестування було перевірено кроссбраузерність, інтерфейс, локалізацію і UI, які згадані у додатку А, Б, проте нас цікавлять функціональні тести:

- чи можна авторизуватися без зареєстрованого аккаунту;
- спроба увійти без паролю;
- с хибними даними;
- ввійти в зареєстрований обліковій запис;
- також створити цей самий запис.

3.1.1 Створення облікового запису

Перш за все необхідно перевірити форму створення аккаунту. Для цього ми намагаємося перейти до створення аккаунту використовуючи різні негативні сценарії, такі як пусте поле емейл, змішані мови у почті, або недопустимі символи і маємо отримати сповіщення “не вірна емейл адреса”(рис. 3.2). В ході тестування було знайдено безліч помилок у роботі сайту, наприклад можливість створити аккаунт використовуючи “qwe..qwe@.gmail.com”, що містить 2 “.” підряд, або “йцу@mail.ua”, що складається з кирилиці й латиниці.

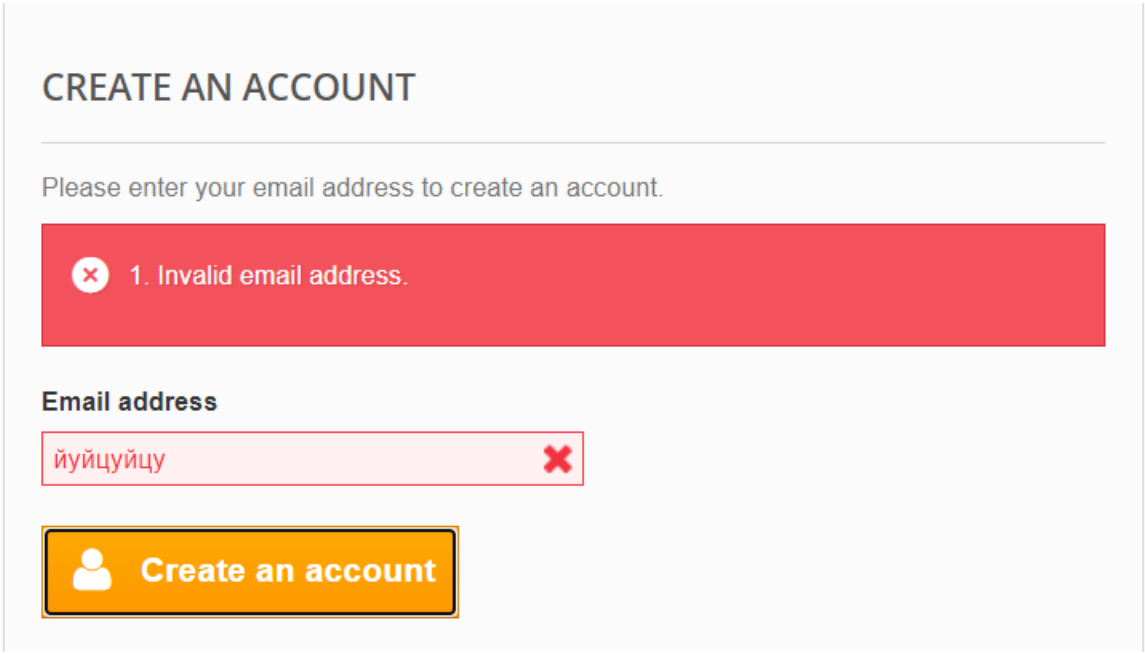

Рисунок 3.2 - Сповіщення “не вірна E-mail адреса”

Після вдалого переходу до наступного вікна у графі пошти вже вказано введені раніше дані, до створення аккаунту лише залишилося заповнити поля. Звичайно має бути неможливим створення облікового запису якщо не заповнювати пароль та інші обов’язкові пункти.

На рисунку 3.3 зображено перелік необхідної інформації для створення аккаунту

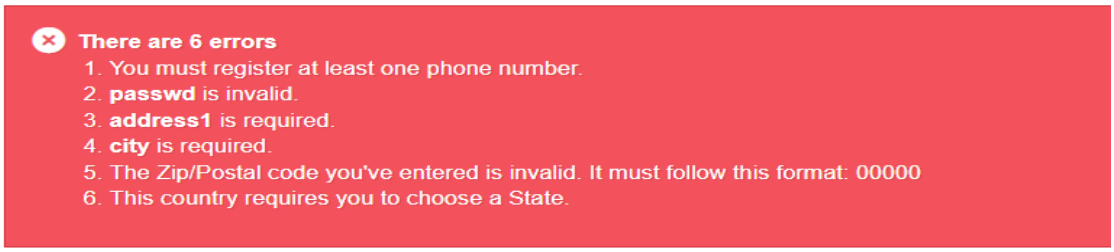

Рисунок 3.3 - Можливі помилки при відсутності інформації

Умовою для створення паролю на цьому сайті є наявність 5 символів, проте відсутня умова про максимальну кількість, що дивно. На сайті автоперевірка поля згодна з будь-якою, більшою за 5, кількістю символів. Вже при підтвердженні створення аккаунту ми отримуємо помилку про наявність завеликої кількості символів у паролю, та граничну точку в 32 знаки.

При вдалому створенні аккаунту ми потрапляємо у вікно особистого кабінету, зображеного на рисунку 3.4, що є останнім кроком створення. У вікні “ My personal information” можна перевірити чи дійсно наші дані використанні у аккаунті і виконати зміни.

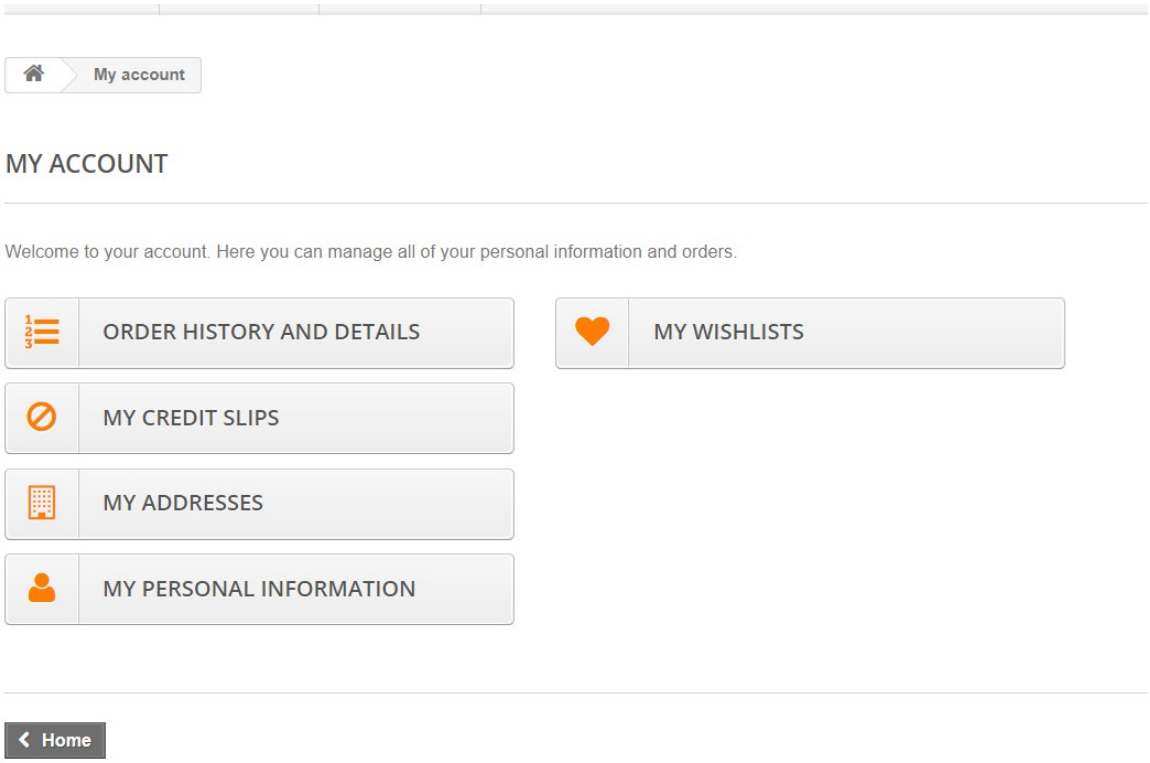

Рисунок 3.4 Особистий кабінет на сайті Automationpractice

3.1.2 Авторизація на сайті Automationpractice

Нарешті можна перейти до наступного етапу тестування, а саме до вікна логіну та паролю.

Звичайно ми знов намагаємося пройти авторизацію спочатку щойно зареєстрованими даними і потрапляємо у кабінет, після чого разлогінюємося і намагаємося потрапити у кабінет через різноманітні негативні перевірки:

- без пошти;
- без паролю;
- з пустими полями;
- помилка в пошті чи паролі;
- спроба використати пароль та логін від різних облікових записів.

Таким чином ми маємо отримати помилку, зображену на рисунку 3.5.

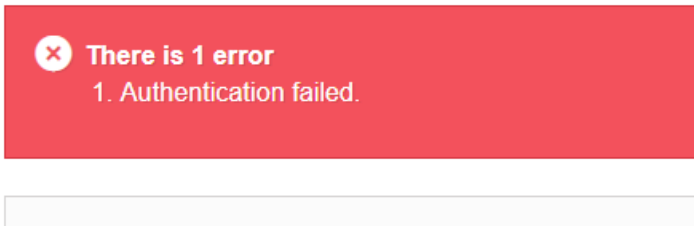

Рисунок 3.5 - Помилка в даних при аутентифікації

Останньою перевіркою в цій формі є вікно відновлення втраченого паролю. Перевіркою в ньому є можливість отримання листа на емейл з інструкціями відновлення доступу.(рис 3.6)

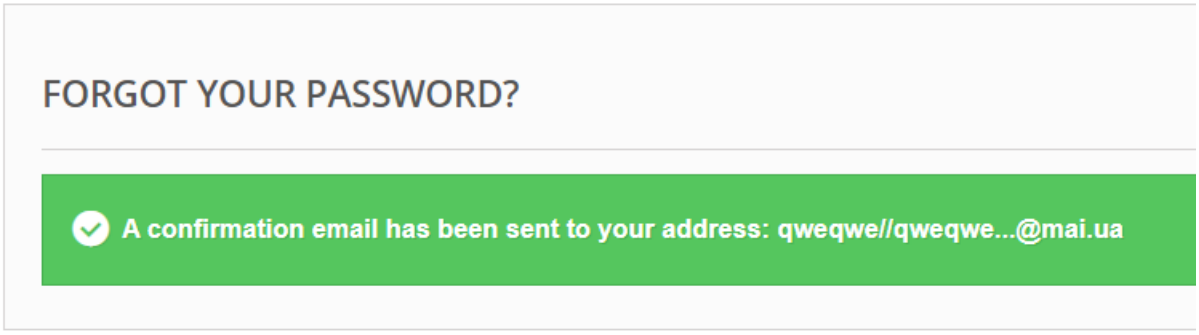

Рисунок 3.6 - Вікно підтвердження відправлення листа з паролем

Також необхідно тестувати можливість надіслати пароль на незареєстрований E-mail. Це позитивний сценарій, на який ми маємо отримати помилку “Такого аккаунту не існує”(рис 3.7)

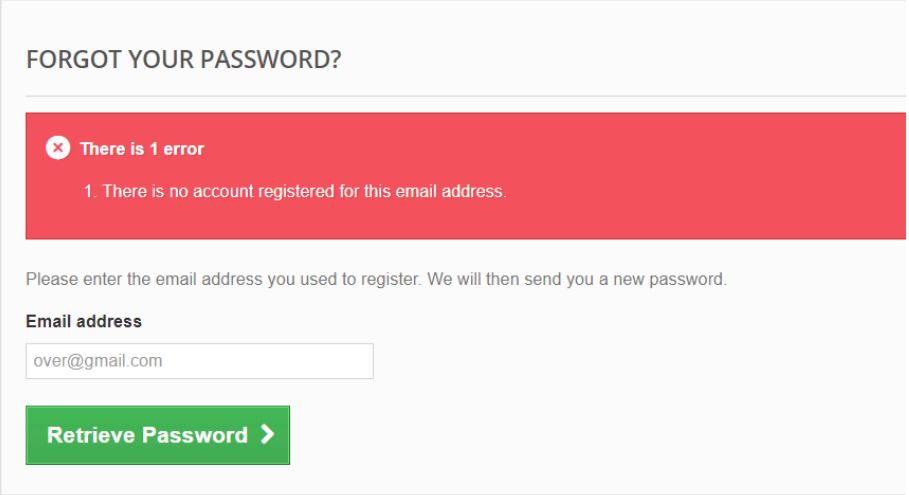

Рисунок 3.7 - Відновлення паролю неіснуючого аккаунту

3.2 Перевірка сайту Kakao Games

Оскільки цей сайт вже широко використовується, то ми проведемо лише поверхове, димове, тестування для ознайомлення з відмінностями. Нами буде створено аккаунт і перевірено чи можливо авторизуватися.

При потраплянні на сайт Kakao Games ми бачимо, що окрім вже безліч раз згаданих логіну і паролю, авторизація можлива й через соціальні мережі, де ми авторизуємося у створений обліковий запис прив’язаним аккаунтом з наприклад Facebook. Сайт розуміє що це дійсно ми, й передає відповідальність авторизації сайту Facebook, який вже має підтвердити нашу особу.

3.2.1 Створення аккаунту

Оскільки сайт пропонує реєстрацію декількома варіантами, то ми повинні перевірити кожен.

При створенні профілю вже зв’язаним аккаунтом Facebook з іншим обліковим запитом ми отримуємо помилку. Зображену на рисунку 3.8 В іншому випадку ми вдало створюємо обліковий запис, де після авторизації з соціальної мережі нам пропонують поєднати його з поштою та паролем, для можливості ввійти використовуючи ці дані.

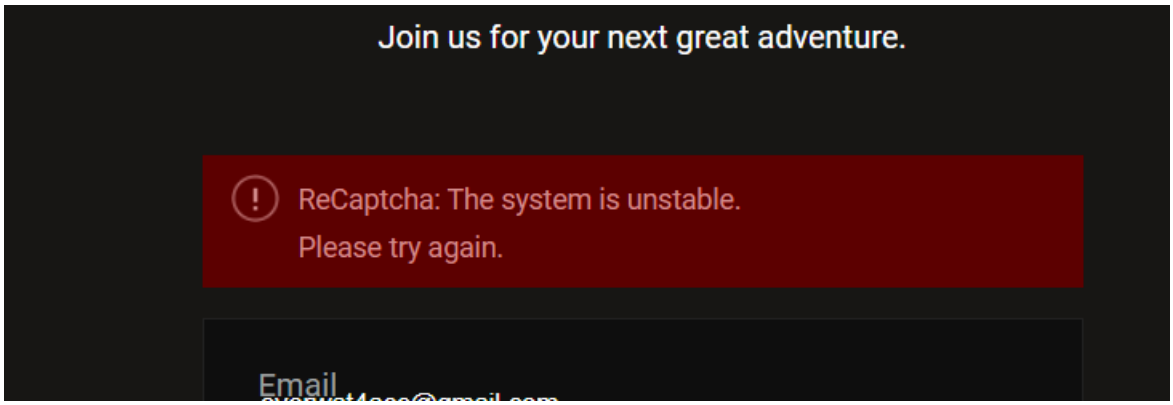

Рисунок 3.8 - Помилка створення аккаунту з допомогою Facebook

Перевірив цю особливість, ми повертаємося до звичного нам процесу. На рисунку ми бачимо що для створення профілю є пункти пошти, паролю, перевірки паролю та дати народження, а також підпункти з флажком.(рис. 3.9)

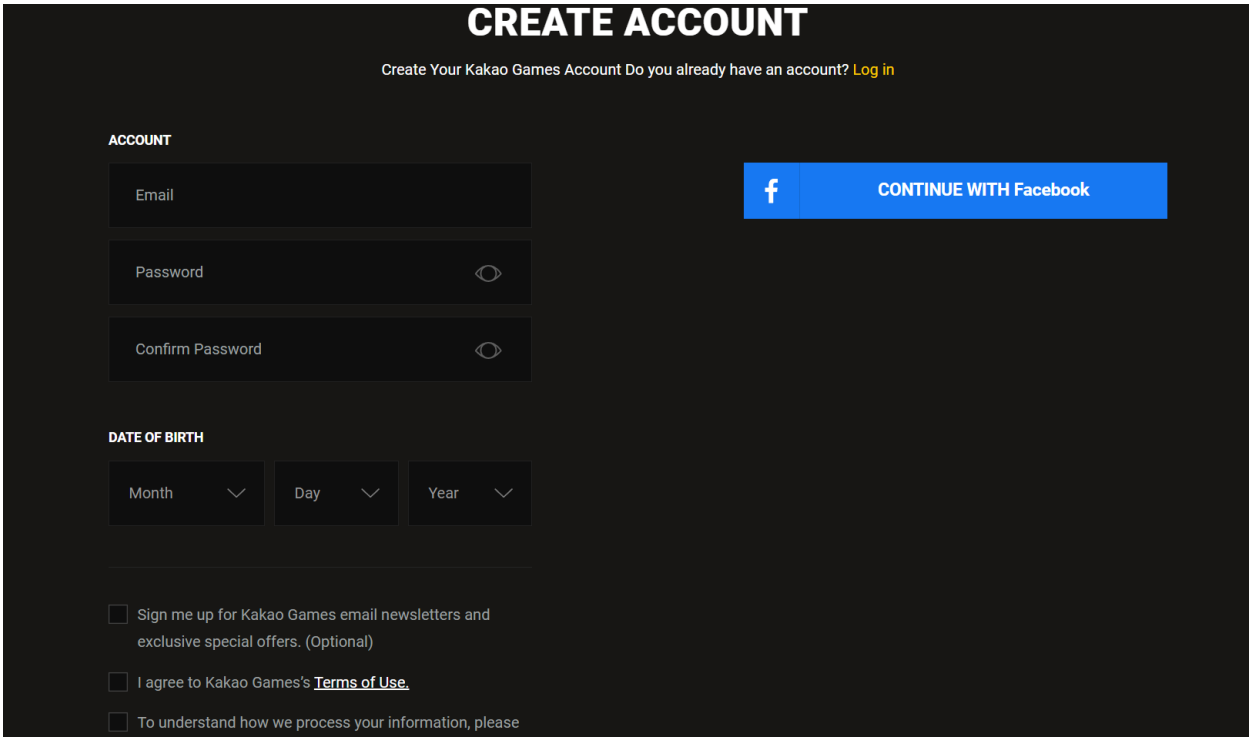

Рисунок 3.9 - Головне вікно створення аккаунту Kakao Games

Нами має бути перевірено кожен пункт із згаданих, тож спочатку ми вдало створюємо обліковий запис, а тоді знову намагаємося отримати помилку створюючи аккаунт тими самими даними, з окремими пустими полями. У такому разі обов'язкові поля підсвічуються нагадуючи про необхідність бути заповненими. Також ми бачимо, що серед чекбоксів можна пропустити перший, про отримання інформації на пошту, як і вказано в тексті самого пункту.

Після першого кроку наступним є спроба використати декілька неможливих емейлів (наприклад:qweqsobaka@, .seqweobaka@gmail.com, so..ka@gmail.com, і так далі...) результатом чого має бути помилка — неможлива пошта(рис 3.10), також можливе сповіщення (*Email address must be formatted correctly*) при помилці безпосередньо у форматі пошти.

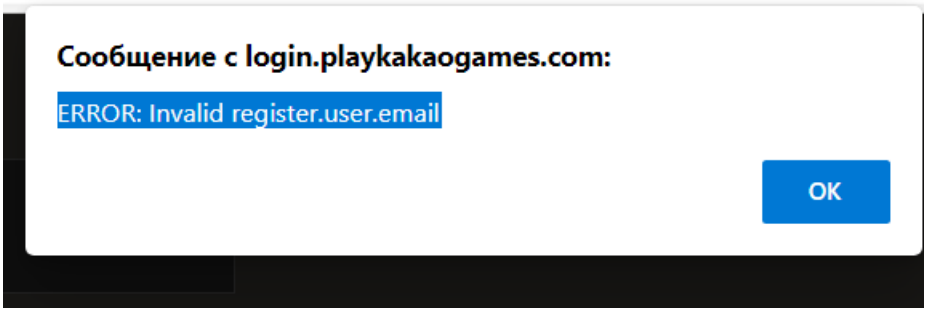

Рисунок 3.10 - Помилка в пошті при регістрації

Коли пункт пошти буде виконано, ми розпочинаємо перевіряти пароль та його підтвердження. Умови паролю вже згадані раніше, це наявність великих/малих літер, певних спецсимволів, цифри та кількість символів від 8 до 16. Оскільки відсутність інформації в полях ми вже перевірили, залишилось лише визначити чи можливо уникнути якусь з умов. Підказка під блоком інформує нас нагадуючи що необхідно додати, тож це поле функціонує вірно. На підтвердження паролю не має необхідності витрачати багато часу, адже у разі будь-яких змін ми отримуємо*(Please enter the same password.)*

Наступним кроком є дата народження, умовою якого *(You must be 16 or older to create an account.)*, тобто наявність будь яких даних пізніше ніж 16 років тому. Цікаво що цілком можливо створити аккаунт наче тобі 114 років, що доречно, проте не необхідно. Також можна зауважити, що поле “день” можна залишити не заповненим, адже це не суттєва інформація і досить часто цей блок не обов’язковий, хоча можливо це баг, необхідно звірятися с специфікацією, доступу до якої на жаль не має.

Повертаючись до вікна вдалого створення аккаунту, після підтвердження інформації ми отримуємо вікно, де нас інформують про необхідність підтвердити пошту,(рис 3.11) звичайно навіть отримав листа ми повинні натиснути “Resend” для перевірки і цього функціоналу.

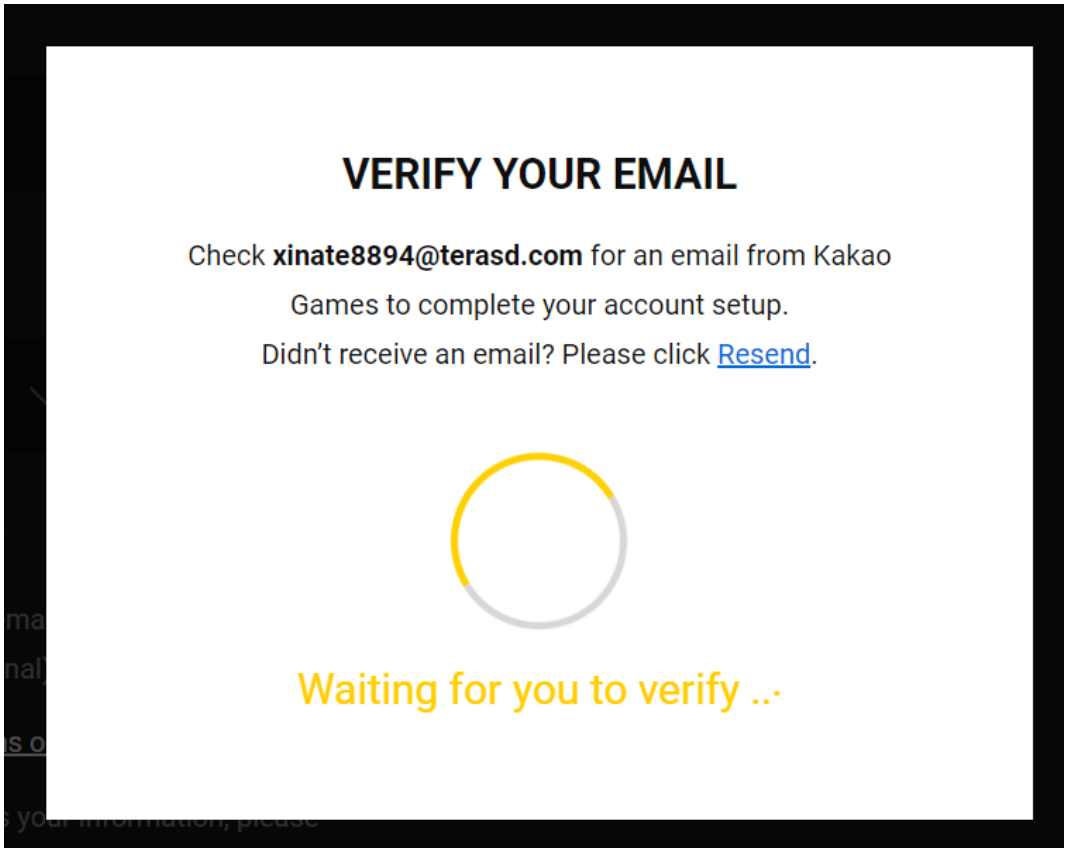

Рисунок 3.11 - Вікно інформування про верифікацію пошти

3.2.2 Авторизація на сайті Kakao Games

Вікно авторизації зображено на рисунку 3.12. Воно має звичні нам логін та пароль, а також кнопки авторизації використовуючи “Steam” та “Facebook”, і “can`t log in?” та кнопку створення аккаунту.

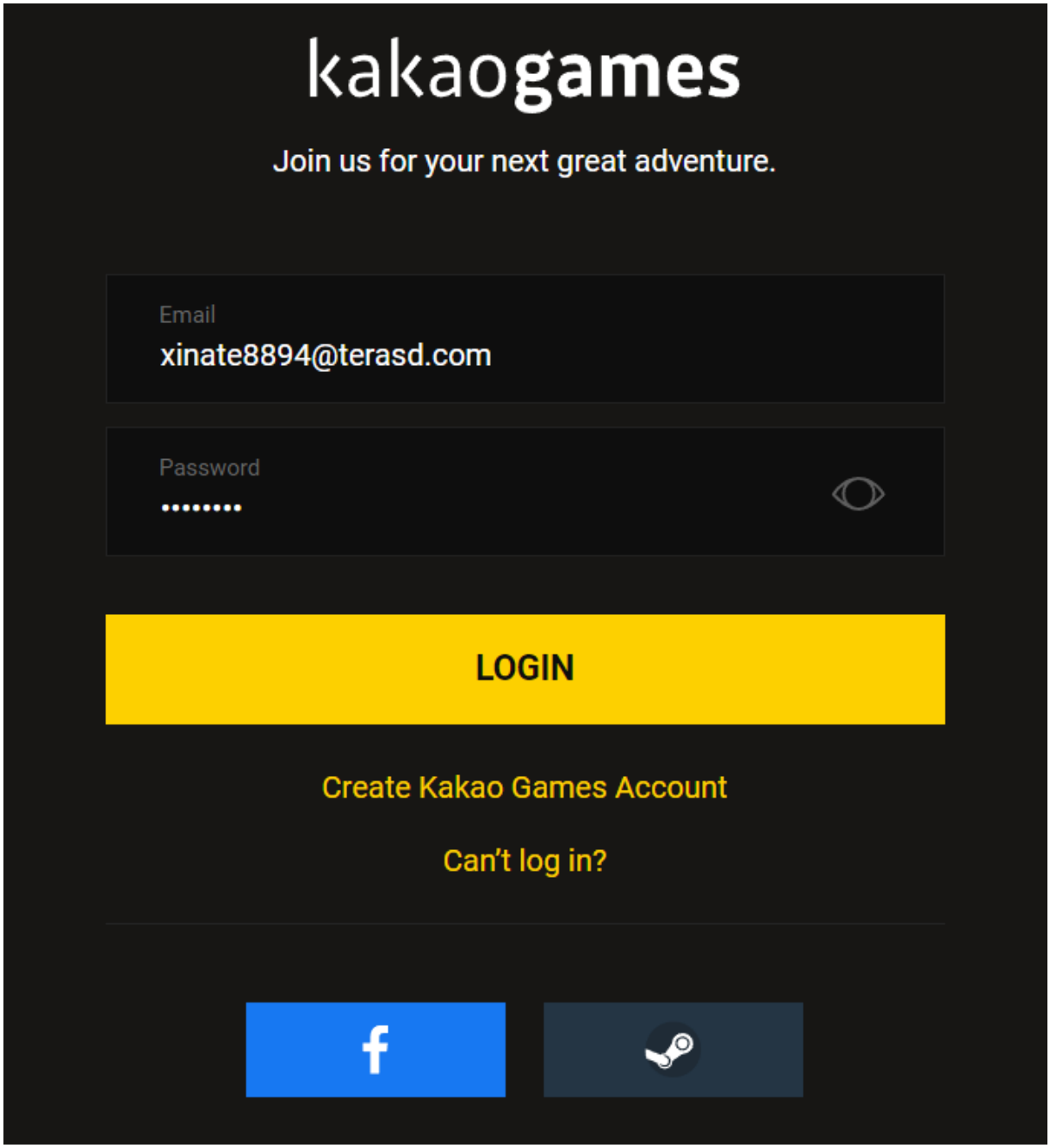

Рисунок 3.12 - Вікно авторизації сайту Kakao Games

Спочатку ми перевіримо успішність авторизації аккаунтом з верифікованою поштою і таким чином потрапимо до особистого кабінету. У випадку використання не верифікованого — знову з’явиться вікно з рисунку 3.11, з нагадуванням про підтвердження пошти.

Якщо намагатися увійти з помилковими даними пошти/паролю, отримуємо помилку з повідомленням *(Email or password information does not match)*, якщо графа пошти чи паролю пуста то кнопка логіну неактивна.

При натисканні кнопки створити акаунт ми потрапляємо на сторінку, яку розглянуто раніше зі створення профілю, то ж нам лише необхідно перевірити чи дійсно ми потрапляємо на неї.

Далі нас цікавить напис “can`t log in?”, натискаючи на яку ми потрапляємо на окрему сторінку в якій міститься:

1. Забули свою поштову адресу?
2. Забули свій пароль?
3. Чи ваш аккаунт заблоковано?
4. Проблеми з доступом через аутентифікатор ?

Перший і четвертий пункт пропонують зв’язатися з технічною підтримкою, роботу якої немає нам необхідності перевіряти, лише чи працює посилання.

В другому пункті перевіряємо чи надсилає сайт нам перевірочний лист з кодом для відновлення пароля, а також чи можна отримати код на незареєстровану, або неможливу пошту.

Останній, тобто третій пункт має інформацію про можливі причини блокування, та посилання на розблокування.

Нарешті закінчивши з “can`t log in?” ми бачимо можливість авторизації через “Steam” та “Facebook”. Натиснувши на “Facebook” ми бачимо помилку, зображену на русинку 3.13, адже відв’язали його від нашого профілю.

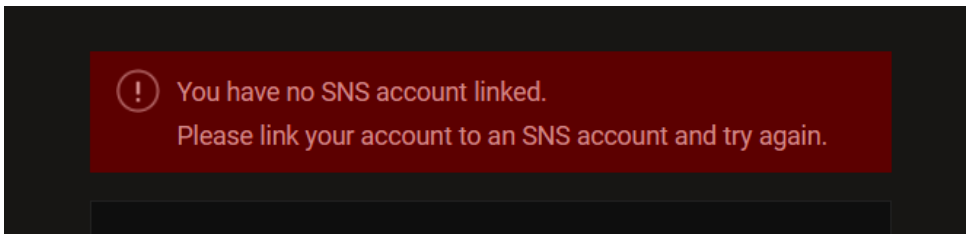

Рисунок 3.13 - Помилка при використані не зв’язаної соціальної мережі

Для подальшого тестування відвідаємо профіль звичайним способом і знову прив’яжемо “Facebook” у нашому особистому кабінеті. Виходимо з нього і перевіряємо можливість увійти через кнопку соціальної мережі. Тепер процес виконано вдало. Знову залишаємо кабінет, та переходимо до тестування авторизації через “Steam”. Нас це цікавить не лише через необхідність перевірки, а й у зв’язку з використанням двофакторної авторизації Steam, але на жаль отримуємо повідомлення, зображене на рисунку 3.14, що даний метод більше не використовується.

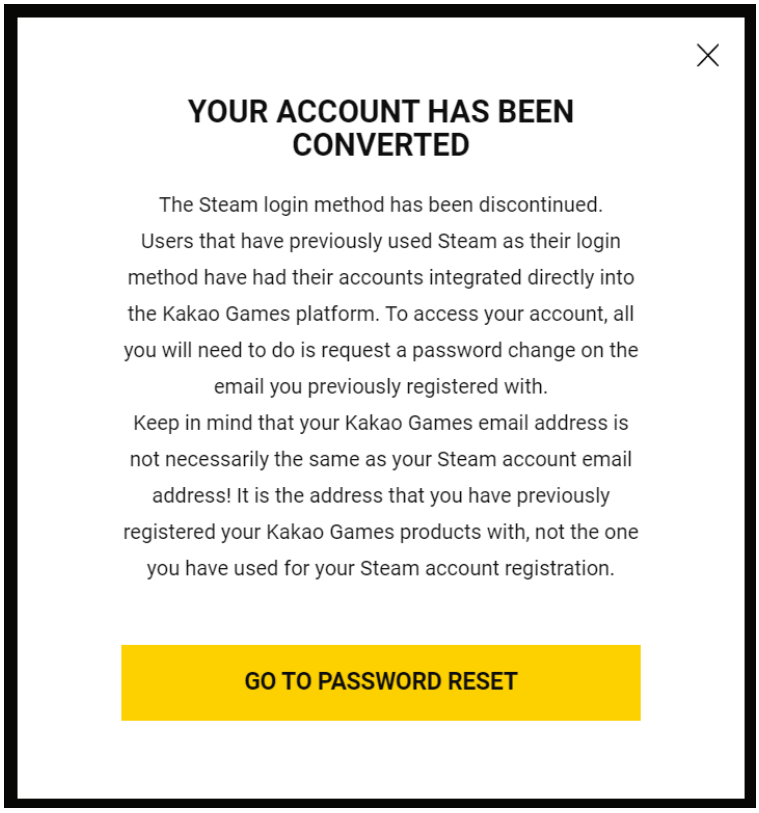

Рисунок 3.14 - Сповіщення про призупинення підтримки авторизації через “Steam”

Оскільки зміни впроваджено лише нещодавно, тож інформація, як відбувалася авторизація раніше в нас присутня, і буде доцільним її використати як демонстрацію двофакторної авторизації. Після введення дійсних логіну та паролю, з’являється вікно, що чекає на код підтвердження з нашого мобільного додатку(рис. 3.15), або пошти, залежно від налаштування

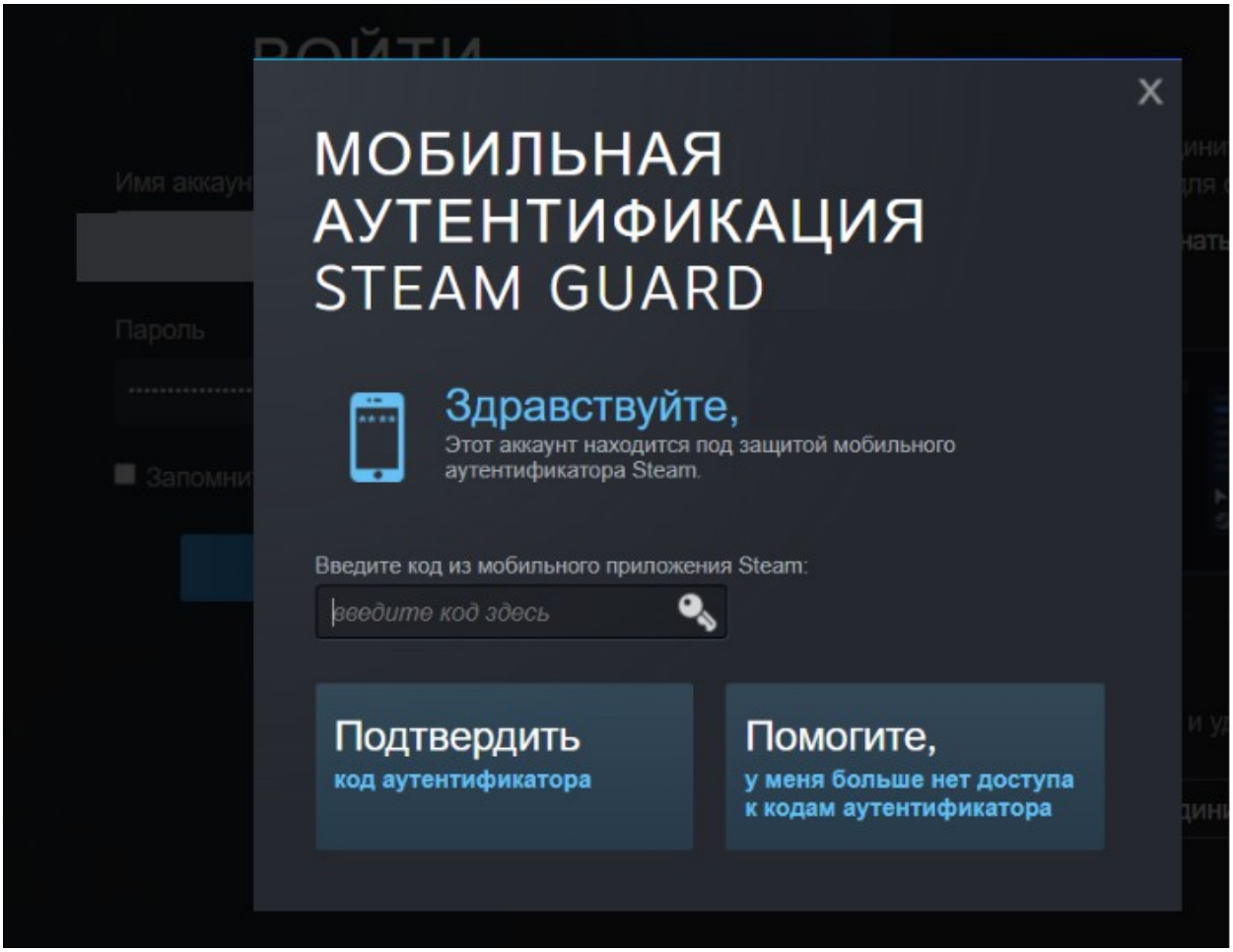

Рисунок 3.15 - Вікно аутентифікації “ Steam”

Після підтвердження коду доступу ми повертаємося на сайт і потрапляємо до особистого кабінету. Також можна відмовитись від даного випадку аутентифікації, в разу втрати доступу до мобільного додатка, й підтвердити свою особу за номером телефона, або поштою.

4 ПОРІВНЯЛЬНИЙ АНАЛІЗ ФОРМ АВТОРИЗАЦІЇ

Після дослідження двох окремих різних форм авторизації на сайті, зрозуміло, що алгоритми для їх тестування мають деякі особливості пов'язані з їх напрямком використання. У першому випадку ми маємо сторінку, що містить безліч наглядних помилок, але має широку можливість для дослідження особливостей тестування, адже і має більший спектр дозволених змін у введеній інформації. Друга ж форма — чітко вивірена. Вона призначена для використання у Європі та Америці, тож не розпізнає пошту кирилицею, але в той самий час через те що вона не є навчальним засобом, а реальний проект, що має зберігати кошти користувачів і їх інформацію, в ній більш детально розглянуто питання безпеки паролю, та методів доступу до інформації.

Під час аналізу цих форм ми бачимо, що єдиний стандарт відсутній через різноманітність виконуваних дій у процесі аутентифікації. Тож єдиний метод не може бути створено. Проте можна виділити ключові кроки в процесі тестування.

- необхідно аналізувати не окремі частини, а аутентифікації в цілому;
- обов'язково перевіряємо працездатність нашої форми на правильних даних;
- необхідно охопити як змога більше негативних сценаріїв й для логіну, й паролю, від відсутності інформації до помилкової, або перебільшень;
- у випадку можливості авторизації іншими способами необхідно перевірити і їх.

Пошта може бути: як кирилицею, так і латиницею, без поєднань; мати 64 символи у локальній частині і 256 з "@"; необхідно перевіряти використання дозволених і не дозволених символів.

Після аналізу цих двох форм було виявлено, що для використання отриманої інформації в подальшому доцільно поєднувати обидва варіанти.

При створенні нових форм для проектів бажано було взяти більш детальну форму реєстрації з першого варіанту, таким чином для ідентифікації

користувача буде більше інформації (контактні дані, адресу, і т.д.) також одразу було б у нагоді додати можливість вказати додаткову пошту, або телефон для більш різноманітних способів відновити доступ. Звичайно чим більше способів увійти тим гірший захист, тож потрібно зробити можливість вільно керувати цими даними через кабінет і змінювати їх з підтвердженням, наприклад через SMS та телефон користувача.

Від другої можна взяти більш детальні вимоги до створення паролю, авторизацію сторонніми ресурсами, для зручності, а також у випадку з необхідністю більшого захисту створити можливість використання двофакторної аутентифікації через спеціальний мобільний додаток, або Google Authenticator. Також на відміну від першого варіанту, необхідно збільшити можливості з відновлення паролю. Як вже запропоновано вище можна додати декілька пошт, або мобільних номерів, проте також можна створити невеликого чат-боту з інструкцією як здійснити вхід при наявності проблем, наприклад втрати телефону.

Доречною була б можливість змінювати мову форми на загальновизнану англійську та повертати на українську. Під час тестування в подальшому необхідно перевірити і обідві локалізації, і функціонал зміни мови.

Рекомендації до тестування форм такого виду є розділення роботи на перевірку створення аккаунту і авторизація в ньому як і було продемонстровано вище. Спочатку необхідно буде перевірити усі можливі способи використання пошт, позитивні і негативні сценарії. Пошта латинецею, кирилецею, пошта з різними доменами, можливість використати IP адресу у якості пошти, відсутня пошта, тимчасова, використання пошти з помилками, наявність повідомлень с помилками і рекомендацій до їх усунення. Перевірити наявність поля “пароль” та “перевірка паролю”, наявність відображення помилок у разі різних паролей у цих полях, або якщо пароль не задовольняє умовам. Перевірити наявність інших полей с контактною інформацією, застережень про необхідність заповнити усі обов’язкові поля, тощо.

Для форми з безпосередньо авторизацією ми використаємо такі ж основні перевірки згадані раніше, авторизація з верифікованими даними, без логіну чи паролю, з помилковими даними, тощо. Додатково необхідно перевірити чи можна авторизуватись довільними даними аккаунту, який зареєстровано, а не лише логіном(чи поштою, якщо це обумовлено)

Обов'язковим для цього вікна є стрес-тестування і тестування навантаження, для запобігання проблем з непрацездатністю форми під час напливу нових користувачів. Кількість спроб одночасного використання форми обумовлюється місцем використання. Звичайно, авторизація на сайті інтернет-магазину або соціальної мережі, наплив користувачів буде більший, ніж у сторінки навчального закладу.

На рисунку 4.1 зображено блок-схему розробленої на основі аналізу форми тестування.

Розглянемо більш детально покроковій алгоритм згідно запропонованої схеми. На першому рівні ми перевіряємо чи створено обліковий запис. Надалі йде розмежування в залежності від наявності в системі облікового запису.

1. Обліковий запис відсутній. Наслідком такого твердження є потреба його створити. Далі як зображено на схемі (рис. 4.1) відбувається позитивна та негативна перевірка даних в наслідок яких ми отримуємо або повідомлення про помилку або створення запису використовуючи невалідні дані.

2. Обліковий запис було створено раніше. В такому випадку відбувається перевірка авторизації шляхом негативного тестування та валідних даних. Наслідком введення невалідних даних або взагалі їх відсутності система повинна видати повідомлення про помилку. Внаслідок використання валідних даних система тестування поверне нас на початок авторизації та видасть повідомлення про те, що тестування пройдено.

Таки алгоритм тестування покаже недоліки та прогалини розробленої системи ідентифікації для ПЗ різного призначення та архітектурного рівня реалізації (мобільні додатки, інтернет магазин, інформаційні сайти, персональні системи).

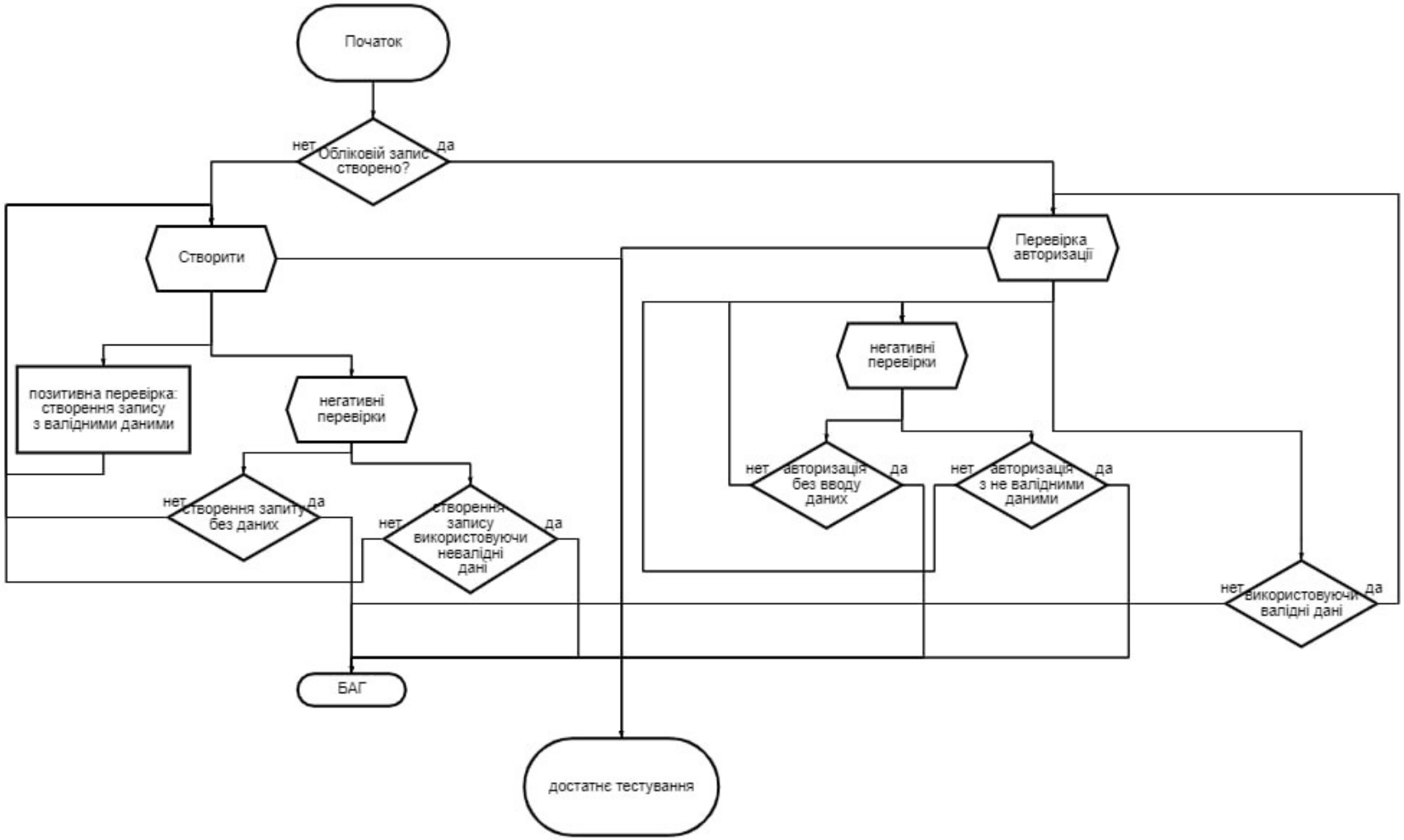

Рисунок 4.1 - Блок-схема розроблена на основі аналізу форми тестування

Перевірки кроссбраузерності і кроссплатформенності для визначення чи можливо комфортно користуватися з довільних браузерів/платформ. Також одразу можна перевірити й локалізацію з інтерфейсом на довільних платформах.

Таким чином можна зробити висновок за розділом, що уніфікувати алгоритм тестування різних форм авторизації неможливо, але спроба поєднати окремі елементи різних тесту вальних алгоритмів дозволила розробити наближену до оптимальної блок схему тестування.

ВИСНОВКИ

В роботі була досягнута основна мета аналізу методів тестування форм авторизації результатом якої стала розроблена узагальнена форма тестування різнотипних форм авторизації.

У розділі 1 було розглянуто види тестування та їх функціональні можливості. Усі види тестування в залежності від мети використання можна поділити на три групи: функціональне, нефункціональне та тестування пов’язане з змінами в ПЗ.

У розділі 2 було розглянуто основні поняття авторизації та аутентифікації. Проведено докладний аналіз видів аутентифікації. Виявлено основні аспекти кожного з них. Розглянуто розгорнуту структуру поняття авторизація та проведено порівняльний аналіз цих двох понять для подальшого розуміння.

У розділі 3 було проведено порівняльне тестування довільних форм на основі авторизації на сайтах Automationpractice та Kakao Games. Це дозволило виявити основні елементи для аналізу авторизації і подальшої розробки уніфікованої форми тестування.

На даний час після проведеного дослідження було виконано повне тестування форми авторизації обраного ресурсу та димові для іншого. Встановлено, що виконання однакових дій при довільних початкових даних не є доцільним, адже процес тестування виконувався вже на кінцевому продукті, а не впродовж його циклу життя. Після аналізу зібраної інформації було запропоновано певний алгоритм дій для форм авторизації з використанням паролю та логіну.

Розділ 4 пояснювальної записки присвячено порівнянню результатів за розділом 3, а саме результатів тестування двох незалежних різнотипних систем авторизації. Це дозволило розробити блок схему алгоритму тестування з

урахуванням можливих недоліків виявлених на етапі аналізу існуючих форм авторизації та їх тестування.

Доречно додати рекомендацію розпочинати тестувати проект паралельно його створенню, для швидкого реагування на зміни й подальшою підтримкою регресійного тестування, задля не допущення повторної появи виправлених помилок. Закінчувати ж роботи необхідно лише після виведення проекту з експлуатації.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Software Testing and Continuous Quality Improvement 2nd edition./ William E.Lewis 2005. – 561 с.

2. The art of software testing 2nd edition./ Glenford J. Myers. 2004. – 255 с.

3. Automationpractice [Електронний ресурс] – Режим доступу до ресурсу: https://automationpractice_.com/.

4. Аутентифікація [Електронний ресурс] – Режим доступу до ресурсу: https://www.unisender.com/ru/support/about/glossary/chto-takoe-email-autentifikaciya/

5. Двофакторна атентифікація [Електронний ресурс] – Режим доступу до ресурсу: https://www.aladdin-rd.ru/catalog/two-factor_authentication

6. Авторизація [Електронний ресурс] – Режим доступу до ресурсу: https://www.sravni.ru/enciklopediya/info/chto-takoe-avtorizacija/

7. Тестування форм входу [Електронний ресурс] – Режим доступу до ресурсу: http://marshrut-testirovshika.ru/forma_vhoda/

8. Тестування логін/пароль [Електронний ресурс] – Режим доступу до ресурсу: https://testitquickly.com/2009/09/09/vvodeste-loginu-la-adnaklassni6i/

9. Чек-лист юзабіліті [Електронний ресурс] – Режим доступу до ресурсу: https://testmatick.com/ru/aktualnyj-chek-list-dlya-provedeniya-yuzabiliti-testirovaniya/