

Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій

Кафедра програмних систем і технологій

УДК 004.8

На правах рукопису

ВИПУСКНА КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

Тема: “Онлайн ідентифікація для двох факторної автентифікації”

Спеціальність: 121 “Інженерія програмного забезпечення”

ПОЯСНЮВАЛЬНА ЗАПИСКА

Студентка

ІПЗ-44 _____ Софія ЄРЕМЕНКО

Науковий керівник

к.т.н, доцент кафедри ПСТ _____ Ксенія ДУХНОВСЬКА

Завідувач кафедри

д.т.н., проф. _____ /Олексій БИЧКОВ/

Київ – 2024

Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій Кафедра програмних систем
і технологій Освітньо-кваліфікаційний рівень бакалавр
Спеціальність 121 “Інженерія програмного забезпечення”

ЗАТВЕРДЖЕНО

Зав. кафедри програмних систем і технологій

_____(Олексій БИЧКОВ)

(підпис) (прізвище та ініціали)

ЗАВДАННЯ**НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ
РОБОТУ СТУДЕНТЦІ**

Єременко Софії Олександрівні

(прізвище, ім'я, по-батькові)

**1. Тема бакалаврської роботи “ Онлайн ідентифікація для двох факторної
автентифікації ”**

керівник проекту (роботи) Духновська Ксенія Костянтинівна, к.т.н, доцент

затверджені наказом вищого навчального закладу від _____ № _____**2. Строк подання студентом роботи _____****3. Вихідні дані до проекту (роботи)** Початкові бізнес вимоги до розроблення
здуманного проекту та теоретичні концепції для розробки**4. Зміст розрахунково - пояснювальної записки(перелік питань, які
потрібно розробити)**

1. Огляд існуючих систем
2. Визначення вимог
3. Розробка програмного забезпечення
4. Тестування розробленої системи

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

1. Три фактори аутентифікації (рис. 1.1, ст. 16)
2. Двофакторна автентифікація (рис. 1.2, ст. 17)
3. Процес оформлення покупки з 2FA для відомого користувача (рис. 2.1, ст. 26)
4. Процес оформлення покупки з 2FA для анонімного користувача (рис. 2.2, ст. 27)
5. Огляд переходів даних в NgRx (рис. 3.1, ст. 36)
6. Логіка перевірки сесії та посвідчення користувача (рис. 3.2, ст. 39)
7. Успішно пройдені тести в локальному середовищі (рис. 4.1, ст. 44)

6. Консультанти з роботи із зазначенням розділів роботи, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Розділ 1	Ксенія ДУХНОВСЬКА	20.01.2024	20.01.2024
Розділ 2	Ксенія ДУХНОВСЬКА	20.01.2024	20.01.2024
Розділ 3	Ксенія ДУХНОВСЬКА	20.01.2024	20.01.2024
Розділ 4	Ксенія ДУХНОВСЬКА	20.01.2024	20.01.2024
Розділ 5	Ксенія ДУХНОВСЬКА	20.01.2024	20.01.2024

7. Дата видачі завдання _____

Керівник _____ (Ксенія ДУХНОВСЬКА)

Завдання прийняла до виконання _____ (Софія ЄРЕМЕНКО)

КАЛЕНДАРНИЙ ПЛАН

	Назва етапів курсової роботи	Термін виконання етапів кваліфікаційної роботи	Відмітка про виконання
1.	Отримання індивідуального завдання	01.02.24-10.02.24	Виконано
2.	Настановча групова співбесіда з курсової роботи	1.02.24	Виконано
3.	Затвердження плану курсової роботи	15.02.24	Виконано
4.	Опрацювання літературних джерел з теми дослідження	15.02.24-15.03.24	Виконано

5.	Проектування і реалізація програмного забезпечення.	15.03.24-15.04.24	Виконано
6.	Тестування та аналіз розробленого програмного забезпечення	15.04.24-10.05.24	Виконано
7.	Підготовка презентації	10.05.24 – 15.05.24	
8.	Подання роботи до захисту	16.05.24	

Студентка – бакалавр _____ (Софія ЄРЕМЕНКО)

Керівник роботи _____ (Ксенія ДУХНОВСЬКА)

АНОТАЦІЯ (укр)

Випускна кваліфікаційна бакалаврська робота: 53 с., 7 рис, 8 джерела.

Тема: Онлайн ідентифікація для двох факторної автентифікації.

Об'єкт дослідження: система ідентифікації та автентифікації користувачів при покупці мобільного тарифу Swisscom, включаючи механізми двофакторної автентифікації (2FA) та управління ідентифікаційними даними користувачів.

Мета роботи: покращення системи оформлення замовлення, імплементація двофакторної автентифікації (2FA) для забезпечення вищого рівня безпеки користувачів та даних.

Предмет дослідження: процеси ідентифікації та автентифікації клієнтів у системі.

Результати дослідження: розроблено сервіс перевірки сесій та документів користувачів, що завдяки двофакторній автентифікації спростило процес покупки. Впроваджена подвійна валідація підвищила безпеку системи, а результати тестування підтвердили швидкість, простоту та безпечність процесу для користувачів.

Висновок

Висновок цього дослідження підтверджує значні переваги впровадження двофакторної автентифікації у системах ідентифікації та автентифікації користувачів. Реалізація цієї технології значно покращила процеси безпеки, швидкості та зручності при покупці мобільних тарифів, демонструючи ефективність у захисті особистих даних користувачів та наданні їм більш легкого доступу до послуг.

ТЕГИ

Двофакторна автентифікація, безпека системи, ідентифікація користувача, оптимізація процесів, Swisscom

АНОТАЦІЯ (англ)

Graduation qualifying bachelor thesis: 53 pp., 7 figures, 8 sources.

Topic: Multi-factor authentication for 2-Factor-authorisation.

Research object: the system for user identification and authentication when purchasing Swisscom mobile tariffs, including two-factor authentication (2FA) mechanisms and the management of user identification data.

The goal of the work: improving the order processing system, implementing two-factor authentication (2FA) to ensure a higher level of security for users and data.

Subject of Research: the processes of client identification and authentication within the system.

Research results: a service for verifying user sessions and documents was developed, which simplified the purchasing process through two-factor authentication. The implementation of double validation enhanced system security, and testing results confirmed the speed, simplicity, and safety of the process for users.

Conclusion:

The conclusion of this study confirms the substantial benefits of implementing two-factor authentication in user identification and authentication systems. The implementation of this technology has significantly improved security, speed, and convenience in purchasing mobile tariffs, demonstrating effectiveness in protecting users' personal data and providing them with easier access to services.

TEGS

Two-factor authentication, system security, user identification, process optimization, Swisscom

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	11
ВСТУП	12
РОЗДІЛ 1	16
Теоретична частина	16
1.1. Традиційні системи автентифікації	16
1.2. Системи двофакторної автентифікації	17
1.3. Переваги та недоліки 2FA	18
1.4. Актуальність використання автентифікації	20
1.4.1. Актуальність використання автентифікації з боку користувача	20
1.4.2. Актуальність використання автентифікації з боку системи	22
РОЗДІЛ 2	24
Аналітична частина	24
2.1. Роз'яснення необхідності розробки	24
2.2. Аналіз вимог до системи	25
2.2.1. Функціональні вимоги	25
2.2.2. Нефункціональні вимоги	25
2.2.3. Опис досвіду користувача	26
2.3. Вибір технологій для розробки	27
2.3.1. Фронтенд розробка	27
2.3.2. Бекенд розробка	28
2.3.3. Тестування API	28
2.3.4. Забезпечення якості та інтеграція	28
РОЗДІЛ 3	29
Проектна частина	29
3.1. Проектування системи	29
3.1.1. Архітектура системи	29
3.1.2. Використання мікросервісів чи монолітної архітектури	30
3.2. Вибір технічних засобів та платформ	30
3.3. Безпека системи	31
3.4. Розробка системи	32
3.4.1. Інтерфейс користувача	32
3.4.2. Огляд сервісу мокування	35
3.4.3. Імплементация функціоналу	36
3.4.4. Логіка перевірки автентифікації	37

	9
3.4.5. Інтеграція з існуючою інфраструктурою	38
РОЗДІЛ 4	40
Експериментальна частина	40
4.1. Тестування системи	40
4.2. Тестування Supress	40
4.3. Сценарії тестування	40
4.3.1. Визначені юз-кейси	40
4.3.2. Сценарії тестування	40
4.4. Опис процедури тестування	42
4.4.1. Спеціальні налаштування для тестування	42
4.5. Аналіз результатів тестування	43
ВИСНОВКИ	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	46

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,**СКОРОЧЕНЬ І ТЕРМІНІВ**

MFA (Багатофакторна автентифікація): Багатофакторна автентифікація - це механізм безпеки, який вимагає від користувачів надання двох або більше факторів верифікації для отримання певного доступу.

2FA (Двофакторна автентифікація): Двофакторна автентифікація є підмножиною багатофакторної автентифікації, яка конкретно вимагає двох різних форм ідентифікації для доступу.

1FA (Однофакторна автентифікація): Однофакторна автентифікація є базовим методом ідентифікації користувача, який вимагає пред'явлення лише одного доказу ідентичності. Цей метод зазвичай ґрунтується на чомусь, що користувач знає, наприклад, паролі або PIN-код.

Mise (мікросервіс): Компонент у мікросервісній архітектурі, який виконує специфічну, обмежену функцію або набір функцій в рамках більшого програмного застосунку.

IDCap: (Id Capture) - сервіс валідації документа та підтвердження особи з камерою.

BüPF: Валідність терміну дії документа та його наявність у системі.

Vega: тестове середовище веб-сайту.

Symphony: середовище з даними клієнтів.

ВСТУП

В епоху цифровізації та інформаційних технологій, безпека користувачів у мережі набуває вирішального значення. Актуальність даної дипломної роботи впливає з постійно зростаючої потреби у забезпеченні надійного захисту особистих даних користувачів в інтернеті. Реалізація мультифакторної автентифікації для користувачів, які вже пройшли двофакторну авторизацію, є важливим кроком у напрямку підвищення безпеки на сайтах та веб-сервісах.

Розробка, що пропонується, є частиною ширшої програми по зручного використання кібербезпеки користувача в телекомунікаційній індустрії Швейцарії. Вона відповідає стратегічним цілям технологічного розвитку та забезпечення безпеки даних, які є актуальними для багатьох країн, включаючи Швейцарію. Проєкт вписується в загальні плани компанії на забезпечення надійного захисту інформації своїх користувачів та відповідає глобальним тенденціям у сфері кібербезпеки.

Мета роботи полягає у розробці та впровадженні функціональної можливості мультифакторної автентифікації для користувачів, що забезпечує вищий рівень безпеки за рахунок використання додаткових методів перевірки особистості. Це дозволить створити ефективний захист від несанкціонованого доступу до облікових записів, зменшуючи ризики витоку конфіденційної інформації.

Ключові ідеї проєкту включають:

- підвищення рівня безпеки та ефективної логізації користувачів шляхом використання додаткових методів автентифікації, які спростять шлях до покупки бажаного продукту;
- використання сучасних технологій для інтеграції мультифакторної автентифікації на веб-сайтах, забезпечуючи зручність та ефективність процесу верифікації;

- дослідження найкращих практик у сфері кібербезпеки та автентифікації для розробки оптимальної стратегії впровадження мультифакторної автентифікації.

Практичне значення роботи полягає у забезпеченні вищого рівня захисту користувачів на сайтах, що не лише підвищує довіру до цих ресурсів, але й сприяє загальному підвищенню безпеки в цифровому просторі. Це також полегшить процес реєстрації при покупці що, позитивно вплине на загальне враження користувача. Впровадження мультифакторної автентифікації може стати стандартом для ресурсів, які прагнуть забезпечити максимальний захист даних своїх користувачів.

Об'єкт дослідження - система ідентифікації та автентифікації користувачів на платформі Symphony, включаючи механізми двофакторної автентифікації (2FA) та управління ідентифікаційними даними користувачів.

Предмет дослідження - методи і технології імплементації мультифакторної автентифікації для користувачів веб-сайтів.

Методи дослідження включають аналіз сучасних підходів до автентифікації, програмування та розробку веб-сервісів, а також використання інформаційних технологій для захисту даних.

Для реалізації цілей роботи були поставлені наступні завдання:

- теоретичний аналіз існуючих технологій 2FA та їх застосування у телекомунікаціях;
- розробка програмної частини для імплементації 2FA на сайті;
- практичні тести системи для оцінки її ефективності та безпеки;
- аналітична робота над отриманими результатами, їх систематизація та оцінка.

Наукова новизна отриманих результатів полягає у розробці комплексного рішення, що дозволяє значно підвищити безпеку користувачів на етапі автентифікації шляхом впровадження додаткових

факторів верифікації. Це сприяє створенню ефективніших механізмів захисту в цифровому просторі. А також це покращує досвід користування сайтом що може в декілька раз підвищити кількість продаж. Розроблена система представляє собою унікальне рішення, адаптоване під специфіку телекомунікаційної індустрії Швейцарії, що дозволяє забезпечити високий рівень безпеки користувачів без значного ускладнення процесу автентифікації.

В рамках виконання випускної кваліфікаційної бакалаврської роботи було розроблено систему двофакторної автентифікації для сайту телекомунікаційної компанії з Швейцарії. Особистий внесок студентки включає ряд ключових аспектів проектування, розробки та впровадження цієї системи, виходячи з вимог та ідей, наданих компанією.

Особистий внесок студентки:

- аналіз вимог та ідей: Студентка провела детальний аналіз загального списку вимог, наданих компанією, а також ідеї дизайну системи, що відповідали загальному напрямку роботи. Було уточнено ключові функціональні та нефункціональні вимоги, які система повинна була задовольнити;
- адаптація концепції системи: Розробка концепції системи була надана системним архітектором та детально проаналізована і погоджена зі студенткою. Студентка активно брала участь у процесі обговорення та погодження концепції, вносячи пропозиції щодо оптимізації та вдосконалення системи з точки зору її ефективності, безпеки та користувацького досвіду;
- імплементація системи: На основі узгодженої концепції, студенткою було виконано програмування основних компонентів системи, включаючи реалізацію механізмів ідентифікації та автентифікації,

інтеграцію з існуючою інфраструктурою сайту та налаштування взаємодії з зовнішніми сервісами для реалізації 2FA;

- тестування та оптимізація: Студенткою було проведено ретельне тестування розробленої системи для виявлення та усунення помилок, оцінки безпеки та загальної ефективності. В результаті тестування були ідентифіковані потреби в корективах та оптимізації, які були успішно реалізовані.

Структура та обсяг роботи: Дипломна робота включає вступ, огляд літератури, опис методології дослідження, практичну частину з розробкою та тестуванням системи, аналіз отриманих результатів, висновки та рекомендації, а також додатки, що містять технічну документацію.

РОЗДІЛ 1

Огляд існуючих систем автентифікації

1.1. Традиційні системи автентифікації

Автентифікація в контексті кібербезпеки – це процес, який дозволяє переконатися в тому, що людина чи система є тією, за кого себе видає. Це ключовий крок у захисті інформаційних систем від несанкціонованого доступу, особливо коли мова йде про доступ до важливих ресурсів.

У фізичному світі ми ідентифікуємо людей через знайомі обличчя та особисте спілкування, але коли ми не можемо здійснити визнання особи безпосередньо, ми часто використовуємо предмети або документи, що підтверджують їхню особу, такі як паспорти чи водійські права. У цифровому світі це прячує аналогічно: ми пред'являємо цифрові креденціали, які можуть бути перевірені системою, аби довести нашу ідентичність.

Ці цифрові креденціали можуть бути паролями, цифровими сертифікатами, або навіть біометричними даними. Важливо, що креденціали, які ми подаємо для автентифікації, містять інформацію, що дозволяє системі переконатися у нашій особистості.

Взагалі, існує три загальноприйняті фактори автентифікації (рис. 1.1):

1. щось, що ви знаєте: це може бути пароль, PIN-код або навіть відповідь на секретне питання;
2. щось, що у вас є: наприклад, фізичний ключ, смарт-карта або мобільний телефон з ОТР (одноразовим паролем);
3. щось, що ви є: біометричні характеристики, такі як відбитки пальців, ретини ока, або голос.





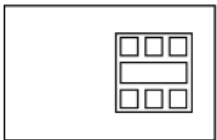

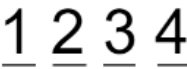


Knowledge Factor (something you know)	Possession Factor (something you have)	Inherence Factor (something you are)
 Password	 Smartphone	 Fingerprint
 Security Question	 Smart Card	 Retina Pattern
 PIN	 Hardware Token	 Face Recognition

Рис. 1.1. – Три фактори аутентифікації

Традиційні системи автентифікації, як правило, покладаються на перший фактор (1FA) – щось, що користувач знає. Найпоширеніший приклад такої системи – це використання статичних паролів. Хоча вони й прості у використанні, вони також несуть у собі ризики: паролі можуть бути відгадані, перехоплені або скомпрометовані через фішинг або інші види кібератак.

Щоб уникнути цих ризиків, системи повинні мати засоби для управління паролями, які включають правила створення складних паролів, регулярну зміну паролів, та застосування додаткових засобів безпеки, таких як 2FA.

Окрім цього, використання cookie-файлів у браузерях може бути прикладом цифрових креденціалів, які дозволяють веб-сайтам впізнавати

користувачів при повторних візитах без необхідності повторного вводу пароля. Хоча такий підхід не є надійною формою автентифікації, він забезпечує певний рівень зручності для користувачів.

1.2. Системи двофакторної автентифікації

Системи 2FA впроваджують додатковий рівень безпеки, використовуючи два різні типи креденціалів для ідентифікації особи. На відміну від традиційних систем, які покладаються тільки на щось, що користувач знає, наприклад, пароль, 2FA вимагає від користувача додаткового фактора — це може бути щось, що він має (рис. 1.2), наприклад, токен, смарт-карта, мобільний телефон, або щось, що є частиною його біометрії, відбиток пальця, сканування сітківки ока.



Рис. 1.2. - Двофакторна автентифікація

Цей підхід значно ускладнює несанкціонований доступ до акаунта, оскільки потенційному нападнику потрібно бути володарем обох факторів, що рідко зустрічається. Наприклад, навіть якщо хтось дізнається ваш пароль, без фізичного пристрою, що генерує одноразові

коди або без вашого біометричного даних, вони не зможуть отримати доступ.

За матеріалами, наданими в скріншотах, розглядається приклад використання челендж-респонс систем, де сервер надсилає випадково згенеровану послідовність символів, і користувач має відповісти на неї, використовуючи попередньо обумовлений алгоритм. Такий метод може включати використання секретного ключа і стати основою для токенів, які генерують одноразові паролі з певною періодичністю, забезпечуючи таким чином двофакторну автентифікацію.

Ці системи можуть бути використані разом з іншими методами автентифікації, що дозволяє створити більш складні і безпечні механізми верифікації. Наприклад, смарт-карти, які містять вбудовані електронні чіпи, можуть зберігати цифрові сертифікати та використовуватися для шифрування та дешифрування даних, забезпечуючи надійну автентифікацію, яка може бути доповнена скануванням відбитків пальців або іншими біометричними даними.

Окрім підвищення безпеки, двофакторна автентифікація також дозволяє зменшити ризики, пов'язані з компрометацією одного з факторів, наприклад, якщо пароль стає відомим нападникам, наявність другого фактора все ще захищає від несанкціонованого доступу. Це робить 2FA не тільки ефективнішою, але й більш надійною системою захисту в порівнянні з традиційними методами автентифікації.

1.3. Переваги та недоліки 2FA

2FA пропонує збалансований підхід до безпеки, вимагаючи два різні методи перевірки, перш ніж надати доступ до акаунта чи ресурсу. Використання 2FA значно підвищує безпеку порівняно з однофакторними методами, адже навіть у випадку, якщо один із факторів (наприклад, пароль) стає відомим нападникам, наявність другого

фактора (такого як фізичний токен чи біометричні дані) все ще захищає від несанкціонованого доступу.

Переваги 2FA:

- підвищена безпека: Додавання другого рівня автентифікації значно ускладнює потенційним зловмисникам використання вкраденої інформації;
- зменшення ризику: Якщо пароль було скомпрометовано, другий фактор автентифікації все ще виступає як бар'єр для несанкціонованого доступу;
- відповідність нормам: Багато регуляторних стандартів та законів вимагають підвищеного рівня захисту, який може бути забезпечений за допомогою 2FA.

Недоліки 2FA:

- незручність для користувача: Введення додаткових кроків може бути сприйнято як незручність, особливо якщо це потребує фізичного носія (наприклад, токена або смарт-карти);
- технічні проблеми: Пристрої для 2FA можуть мати технічні неполадки або потребувати заміни, що може викликати затримки в доступі;
- додаткові витрати: Впровадження 2FA може вимагати інвестицій у нове обладнання або програмне забезпечення;
- керування та обслуговування: Потрібне додаткове адміністрування, наприклад, для управління втратою або заміни токенів.

1.4. Актуальність використання автентифікації

1.4.1. Актуальність використання автентифікації з боку користувача

Автентифікація клієнтів стає невід'ємною частиною сучасного веб-середовища, оскільки все більше особистих та контрольованих за доступом послуг переходить у онлайн. Актуальність використання автентифікації з боку користувача пояснюється кількома ключовими аспектами.

По-перше, наскільки користувачі переходять на цифрові сервіси, постає питання безпеки особистої інформації та доступу до неї. автентифікація дає змогу встановити особу, яка намагається отримати доступ до даних, забезпечуючи тим самим захист від несанкціонованого втручання. Занадто слабкі схеми автентифікації, особливо в не-корпоративних середовищах, можуть призводити до витоків даних через необережне використання автентифікаторів, як це часто буває з веб-кукіс.

По-друге, є певні практичні обмеження веб-автентифікації. Наприклад, автентифікаційні протоколи на веб-сайтах відрізняються від традиційних через обмежений інтерфейс, який пропонує веб. Протоколи автентифікації для всесвітньої мережі не можуть покладатися на технології, що не є широко розповсюдженими. Важливо розробити систему автентифікації, використовуючи загальноприйняті протоколи та технології, доступні в сучасних веб-браузерах та серверах. Це має бути зручно для користувачів і не вимагати від них додаткових дій, як то інсталяція плагінів чи іншого програмного забезпечення.

Третє значне питання — це продуктивність. Сильніші протоколи безпеки, як правило, вимагають більше ресурсів сервера. Слід знайти баланс між надійністю автентифікації та продуктивністю системи, щоб не навантажити сервер надмірною роботою. Особливо це стосується

SSL, який може бути затратним з точки зору обчислювальних потужностей.

Розробка ефективної системи автентифікації повинна також забезпечувати захист від різноманітних загроз, включаючи зловмисників, які здатні адаптивно запитувати веб-сервер.

Задача забезпечення надійної автентифікації клієнтів на веб-сайтах є надзвичайно актуальною в умовах широкого розповсюдження цифрових послуг та необхідності захисту особистих даних користувачів. Не дивлячись на наявність добре вивчених технік автентифікації, багато сайтів продовжують використовувати надзвичайно слабкі автентифікаційні схеми, особливо в не-корпоративних середовищах, таких як інтернет-магазини. Це часто призводить до недбалого використання автентифікаторів в мережі, зокрема через веб-кукіс.

Проблема слабких схем автентифікації часто є результатом недбалого використання автентифікаторів, які зберігаються на клієнті, особливо через кукі. Таке недбале використання може призвести до несанкціонованого доступу та інших проблем безпеки. Це підтверджується тим, що під час неформального дослідження автентифікаційних механізмів на популярних веб-сайтах було виявлено, що автентифікацію клієнтів можна послабити на двох системах, отримати несанкціонований доступ на восьми, та витягнути секретний ключ, використовуваний для створення автентифікаторів, на одному.

Недостатня придатність існуючих механізмів автентифікації HTTP і SSL/TLS для використання в інтернеті в цілому є однією з причин цих проблем. Відсутність централізованої інфраструктури, такої як публічна інфраструктура ключів або єдина система Kerberos, сприяє поширенню слабких схем. Також багато веб-сайтів розробляють власні механізми автентифікації для поліпшення користувацького досвіду, що часто призводить до безпекових прорахунків.

1.4.2. Актуальність використання автентифікації з боку системи

Актуальність використання автентифікації з боку системи полягає в тому, що це є основою для забезпечення безпеки доступу до ресурсів та сервісів в цифровому середовищі. З огляду на стрімке зростання кількості веб-сервісів та інтернет-ресурсів, надійна автентифікація є критично важливою для захисту персональних даних користувачів та обмеження доступу до інформації, яка може бути чутливою або конфіденційною.

Системна автентифікація допомагає впорядковувати доступ до різноманітних веб-сервісів, забезпечуючи, що користувачі, які входять в систему, є дійсно тими, за кого себе видають. Це стає особливо важливим в контексті зберігання та обробки корпоративних даних, де неавторизований доступ може призвести до втрати важливої інформації або навіть до фінансових втрат.

Актуальність автентифікації з боку системи також виявляється у відповідності до регулятивних вимог в ЄС, що вимагають захисту персональних даних. Ці вимоги накладають на системи обов'язок впровадження ефективних механізмів автентифікації для забезпечення прозорості та відповідальності в обробці даних.

Крім того, з огляду на широке розповсюдження кібератак і витоку даних, системна автентифікація виступає як перший рубіж оборони від несанкціонованого доступу та зловмисних дій. Сильні автентифікаційні протоколи та рішення стають обов'язковою умовою для забезпечення цілісності та надійності веб-сервісів, особливо тих, що займаються фінансовими операціями, здоров'ям, освітою чи іншими суттєвими для суспільства сферами.

В кінцевому підсумку, забезпечення надійної системної автентифікації є невід'ємним аспектом у побудові довіри користувачів до

цифрових сервісів і платформ, а також для забезпечення сталого розвитку цифрової економіки.

1.5. Висновок

Після проведення детального аналізу існуючих систем автентифікації, було зрозуміло виявлено, що багатофакторна автентифікація і зокрема двофакторна ідентифікація, виступають як оптимальні рішення для підтвердження особи. Використання двох незалежних факторів автентифікації значно підвищує безпеку системи, роблячи несанкціонований доступ майже неможливим. Це робить двофакторну автентифікацію ідеальною для заміни традиційних методів ідентифікації, таких як перевірка паспортних даних, особливо у контексті нашого кейсу, де потрібно забезпечити високу надійність і безпеку при обробці особистих ідентифікаційних даних.

РОЗДІЛ 2

Аналітична частина

2.1. Роз'яснення необхідності розробки

Згідно із законодавством Швейцарії, кожен користувач, який бажає придбати мобільний тариф, зобов'язаний пройти процедуру підтвердження особи. Раніше, для купівлі мобільних тарифів необхідно було відвідати магазин та мати при собі посвідчення особи. З часом цей процес був оцифрований, що надало можливість проходити валідацію особи онлайн. Для виконання цієї функції був розроблений спеціалізований сервіс IDCap (ID Capture), забезпечуючи автоматизацію процесу ідентифікації. Цей процес включає фотографування документів, що посвідчують особу, та виконання селфі-відео для перевірки "живості" особи. Втім, даний процес може виявитися часомістким і складним для користувачів, які вже пройшли подібну верифікацію раніше.

Для користувачів, які вже мають активовану двофакторну автентифікацію, IDCap може здаватися надмірним обтяженням. Це призводить до необхідності розробки нової функції, яка дозволить пропустити повторну ідентифікацію в рамках процедури купівлі нових послуг або тарифів. Така опція значно спростить користувацький досвід, зберігаючи при цьому високі стандарти безпеки.

Функціонал з двох факторною автентифікацією не лише сприятиме зручності користувачів, але й підвищить ефективність процесу купівлі, зменшуючи час та ресурси, необхідні для обробки запитів. Це також знизить навантаження на сервіси перевірки ідентифікації, дозволяючи компанії оптимізувати свої ресурси та зосередитися на інших аспектах обслуговування клієнтів.

2.2. Аналіз вимог до системи

2.2.1. Функціональні вимоги

Загальні вимоги: Система має забезпечити можливість доступу до послуг без проходження процедури онлайн-ідентифікації (OID) для клієнтів, які відповідають наступним критеріям:

- клієнт активував двофакторну автентифікацію (2FA);
- клієнт уже підтвердив свою особу та пройшов автентифікацію;
- дійсність ID, зареєстрованого в системі Symphony, підтверджена;
- застосовується метод автентифікації "2FA_Online".

Додаткові критерії (вимоги згідно BūPF):

- у системі Symphony для клієнта встановлено прапорець `IdAvailableFlag = true`;
- термін дії документа повинен бути актуальним (тобто `ExpiryDate` має бути більшим або дорівнює поточній даті);
- зазначено, що автентифікація була проведена (`authenticationDone = true`), що вимагає імплементації нового поля в системі Symphony.

2.2.2. Нефункціональні вимоги

- Система має включати засоби для відстеження всіх випадків входу та ідентифікації, щоб забезпечити аналіз та реагування на будь-які виняткові ситуації;
- мають бути розроблені та впроваджені автоматизовані тести Cypress для перевірки всіх аспектів автентифікаційного потоку, забезпечуючи високу якість та стабільність роботи системи.

- ### 2.2.3. Опис досвіду користувача

- якщо 2FA не успішна, запускається стандартний процес OID;
- у разі невідповідності критеріям, запускається стандартний процес OID.

Для клієнтів, що не увійшли в систему (рис. 2.2):

- якщо клієнт відповідає усім критеріям, він повинен пройти вхід до системи з подальшою автентифікацією 2FA;
 - у разі успішної 2FA крок ідентифікації пропускається;
 - якщо 2FA не успішна, запускається стандартний процес OID;

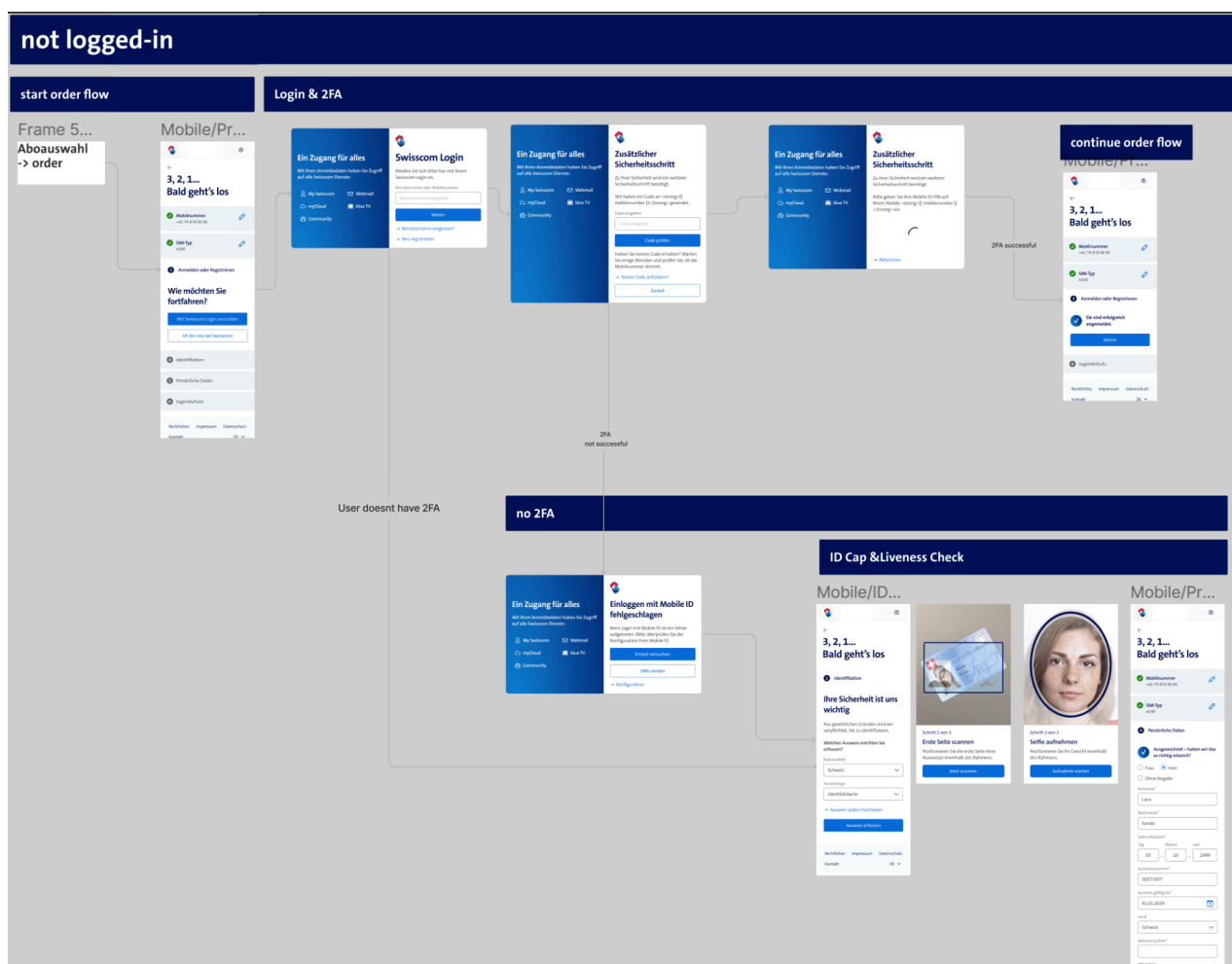


Рис. 2.2. – Процес оформлення покупки з 2FA для анонімного користувача

- якщо клієнт не відповідає критеріям, ініціюється стандартний процес OID.

При поданні замовлення:

- метод автентифікації має бути встановлений як "2FA_Online".

2.3. Вибір технологій для розробки

Під час вибору технологічного стеку для розробки системи двофакторної автентифікації, особливу увагу було приділено існуючим стандартам та практикам, що вже використовуються в компанії. Це дозволило забезпечити високий рівень інтеграції нової системи з наявними рішеннями та спростити процес її впровадження та подальшого супроводження.

2.3.1. Фронтенд розробка

Головна сторінка веб-сайту нашої компанії розроблена за допомогою фреймворку Angular, який використовує мову програмування TypeScript. Angular надає міцну основу для створення інтерактивних веб-інтерфейсів, що дозволяє користувачам легко взаємодіяти з нашим сайтом. Водночас, використання TypeScript підвищує надійність нашого додатку завдяки системі статичної типізації, що мінімізує помилки на етапі розробки та покращує якість нашого коду.

Для підвищення уніфікації та спрощення процесу розробки фронтенду, ми також інтегрували внутрішній інструментарій, схожий на Bootstrap, але спеціалізований під наші корпоративні потреби, який має назву SDX. Цей інструментарій включає велику бібліотеку компонентів зі своїм власним стилем, таких як кнопки, форми, вікна повідомлень та інше, що забезпечує консистентність візуального оформлення наших веб-ресурсів.

Додатково, для управління асинхронними даними та подіями використовується бібліотека RxJS. Це дає змогу ефективно управляти

потоками даних і асинхронною логікою всередині додатку, що є критично важливим для забезпечення швидкодії та відгуку користувацького інтерфейсу.

2.3.2. Бекенд розробка

Вибір Java у поєднанні з фреймворком Spring Boot визначався їхньою здатністю до швидкої розробки надійних і масштабованих мікросервісів, що є ключовим для нашої компанії. Ці технології є стандартом для бекенд-розробки в нашій організації, що забезпечує гладке впровадження та сумісність з іншими внутрішніми системами.

2.3.3. Тестування API

Для тестування API ми використовували Thunder Testing, систему, розроблену нашою компанією, яка є ключовою складовою нашого інструментарію для автоматизації тестування. Цей сервіс особливий тим, що дозволяє заздалегідь визначити шляхи до ендпоінтів, автоматично генеруючи і відправляючи запити з необхідними параметрами для перевірки відповідей сервера. Такий підхід значно підвищує ефективність тестування, забезпечуючи високу точність перевірки і дозволяючи оперативно виявляти та усувати помилки в обробці запитів.

2.3.4. Забезпечення якості та інтеграція

Системи для забезпечення якості коду, такі як SonarQube, та інструменти автоматизації процесів CI/CD, як Jenkins, вже є частиною інфраструктури компанії. Їх застосування в проекті дозволяє підтримувати високі стандарти якості коду та ефективності розробки.

2.4. Висновок

У рамках аналітичної частини моєї дипломної роботи було проведено глибоке дослідження існуючих систем автентифікації та аналіз вимог, необхідних для впровадження системи двофакторної автентифікації. Основним мотивом для цього стало законодавство Швейцарії, яке вимагає підтвердження особи при купівлі мобільних тарифів. Це дослідження допомогло мені краще зрозуміти, як функціонують традиційні та сучасні методи верифікації особи, їхні сильні та слабкі сторони, а також особливості впровадження двофакторної автентифікації.

Також, під час аналізу було розглянуто технічні аспекти впровадження системи в існуючу інфраструктуру, що включає вибір технологій, розробку фронтенду та бекенду, а також підготовку до тестування системи. Усе це дозволяє сформулювати міцну основу для подальшого проектування та реалізації системи автентифікації, яка буде відповідати як сучасним технологічним вимогам, так і законодавчим регуляціям.

РОЗДІЛ 3

Проектна частина

3.1. Проектування системи

У цьому розділі описується загальний підхід до проектування системи, з акцентом на модульність, масштабованість, та адаптивність до змінних умов експлуатації.

3.1.1. Архітектура системи

Розглянемо технічну структуру системи, включаючи опис логічної, фізичної та компонентної архітектури.

У розробці наявні наступні архітектурні шари:

1. презентаційний шар реалізований за допомогою Angular з TypeScript, який забезпечує розробку інтерактивних веб-інтерфейсів. Ми використовуємо розподілену архітектуру на UI, бізнес логіку та glue layer, який виступає зв'язком між бізнес функціоналом та компонентами. Це дозволяє більш гнучко управляти змінами в користувацькому інтерфейсі та бізнес логіці, оптимізуючи загальну роботу системи;
2. бізнес-логіка реалізована на Java з використанням Spring Boot, що забезпечує потужну платформу для швидкої розробки мікросервісів. Ця частина системи відповідає за обробку даних, виконання бізнес-правил і забезпечення необхідної логіки для взаємодій між компонентами системи;
3. доступ до даних реалізован через множину мікросервісів, розподілених під свої обов'язки, такі як управління інформацією про користувача та оброблення різних запитів.

Це забезпечує високий рівень масштабованості та гнучкості системи, дозволяючи легко адаптуватися до змінних вимог та навантажень.

3.1.2. Використання мікросервісів чи монолітної архітектури

Як вже було зазначено, для цього проекту обрано мікросервісну архітектуру, що дозволяє системі бути більш гнучкою та масштабованою. Мікросервіси сприяють незалежному розгортанню окремих компонентів системи, що поліпшує процеси неперервної інтеграції та доставки (CI/CD) і дозволяє краще управляти завантаженням системи.

Завдяки RESTful API, який реалізовано у нашій системі, взаємодія між різними модулями та з зовнішніми системами є стандартизованою та ефективною. API забезпечує чітке визначення контрактів між сервісами, що підвищує надійність і безпеку даних.

3.2. Вибір технічних засобів та платформ

Для розробки як фронтенду, так і бекенду проекту було обрано інтегроване середовище розробки (IDE) — IntelliJ IDEA. Цей вибір зумовлений рядом переваг, які IntelliJ IDEA надає розробникам, особливо при роботі з Java та JavaScript, що є основними технологіями нашого проекту.

Переваги використання IntelliJ IDEA:

- інтеграція із багатьма мовами програмування та фреймворками: IntelliJ IDEA підтримує не тільки Java, але й інші мови, такі як JavaScript, TypeScript, HTML/CSS, що є ідеальним для розробки мультиплатформних проектів. Це робить її універсальним інструментом для розробки різних частин нашої системи;

- підтримка Spring і Angular: IntelliJ IDEA включає розширену підтримку для Spring і Angular, що значно спрощує конфігурацію проєктів, управління залежностями, відладку та тестування;
- інтегровані інструменти для роботи з базами даних: IDE має вбудовані інструменти для роботи з базами даних, що дозволяє легко здійснювати запити, редагування схеми та управління базою даних без необхідності виходу з середовища розробки;
- широкі можливості для рефакторингу та оптимізації коду: IntelliJ IDEA пропонує потужні інструменти для рефакторингу коду, що допомагає підтримувати код чистим, ефективним і легко підтримуваним.

Недоліки використання IntelliJ IDEA:

- вартість ліцензії: на відміну від деяких інших IDE, таких як Visual Studio Code, IntelliJ IDEA є платною, і її використання може бути витратним для стартапів або індивідуальних розробників без корпоративної підтримки;
- вимоги до ресурсів системи: IntelliJ IDEA може вимагати значних ресурсів системи, особливо RAM, що може призводити до сповільнення роботи на менш потужних машинах.

3.3. Безпека системи

Для захисту ендпоінтів, які взаємодіють із сесіями користувачів, впроваджено квоти на кількість запитів (Rate Limiting) — не більше 5 запитів за одну хвилину на одного користувача. Це обмеження допомагає запобігти атакам типу DDoS та брутфорсу, забезпечуючи стабільність та доступність системи.

Перевірка сесії здійснюється за допомогою JSON Web Token (JWT), що передається у вигляді токена в HTTP-запитах. Використання JWT забезпечує, що жодні чутливі дані користувача не відображаються

та не зберігаються у самому ендпоінті, знижуючи ризик витоку інформації у разі перехоплення трафіку.

Для підвищення безпеки відповідей, які відправляються від мікросервісів до фронтенду, було розроблено три спеціалізовані коди відгуків: TIDC, TOF, TTFA. Ці коди ускладнюють можливість підробки даних або несанкціонованого доступу, оскільки кожен код має певне значення та контекст у системі безпеки.

Навіть у випадку, якщо потенційний зловмисник намагається використовувати один з кодів для обходу автентифікації, система передбачає додаткову перевірку цих кодів під час фіналізації замовлення. Якщо отриманий код вказує на те, що автентифікація могла бути пропущена, система ініціює додаткову перевірку для підтвердження дійсності сесії користувача. Це забезпечує, що кожен етап взаємодії з системою є захищеним і відповідає політикам безпеки.

3.4. Розробка системи

3.4.1. Інтерфейс користувача

У рамках проекту "Онлайн ідентифікація для двофакторної автентифікації" основний функціонал поділяється на фронтед частину, бекенд, і сервіс автентифікації. Сторінка /omni-order/ є сторінкою де необхідно налаштувати всю важливу інформацію для замовлення, таку як бажаний номер телефону, тип сім карти, ідентифікація користувача тощо. В залежності від типу юз-кейсу ці етапи можуть відрізнятись. Сценарії, які ми розглядаємо, стосуються купівлі мобільних тарифів користувачами, які вже залогінені в свій профіль та мають необхідний статус сесії.

Залежно від статусу сесії користувача, система може:

- перед етапом ідентифікації відобразити новий етап "двофакторна автентифікація";
- пропустити всі етапи, пов'язані з ідентифікацією;
- залишити процес без змін.

Ці переходи реалізовані завдяки загальним налаштуванням, які ініціюються під час завантаження сторінки та є частиною стартових запитів.

Фронтендова частина системи розроблена з використанням архітектури "playbook", що визначає хронологію подій, які мають відбуватися послідовно. Кожен "playbook" містить декілька "plays", кожен з яких описує певний етап процесу, його назву та умови для виконання, зокрема, визначення після якого "play" конкретний "play" може бути активований.

Для управління станами та обробки подій у фронтендовій частині застосовується бібліотека ngRx . Вона використовується для управління станом додатку через стори, редюсери та ефекти, забезпечуючи зручне управління станами із чітко визначеними потоками даних та взаємодій між компонентами (рис. 3.1). NgRx дозволяє ефективно інтегрувати реактивне програмування в архітектуру застосунку, забезпечуючи високу продуктивність та масштабованість.

З іншого боку, для більш звичайних асинхронних операцій та обробки потоків даних використовується бібліотека RxJS. Це включає обробку запитів на бекенд та загальну маніпуляцію з даними, де RxJS сприяє створенню гнучких та ефективних потоків даних за допомогою своїх операторів, які дозволяють легко комбінувати, фільтрувати, трансформувати та обробляти асинхронні дії.

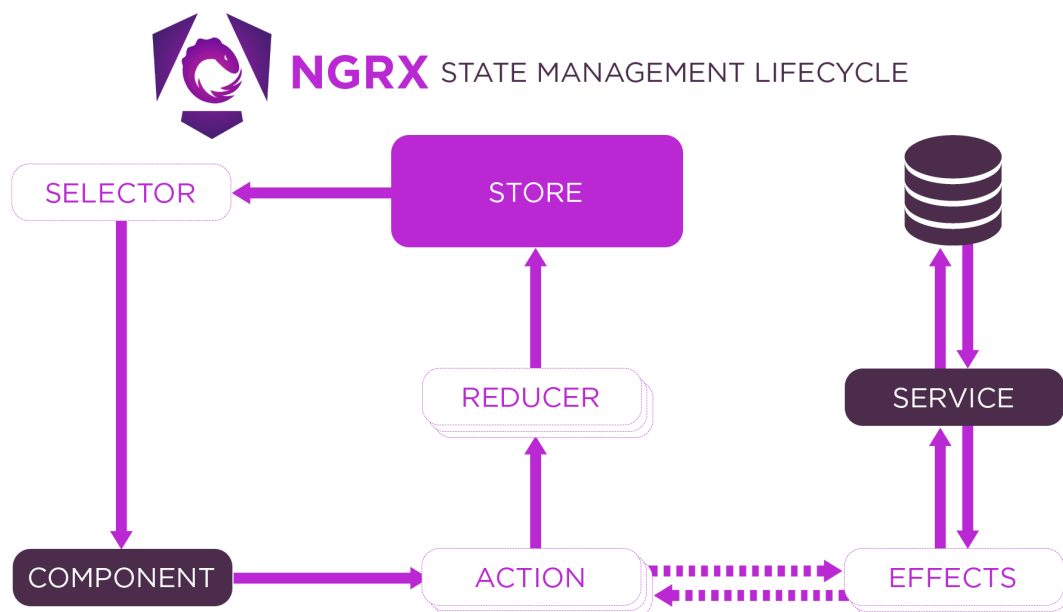


Рис. 3.1. Огляд переходів даних в NgRx

Для кожного тексту в інтерфейсі передбачено локалізацію на чотири мови: німецьку, англійську, італійську, та французьку. Наприклад, для етапу "двофакторна автентифікація", заголовок записаний як "MFA.TITLE": "2-Factor Authentication" у файлах локалізації.

Якщо користувач вирішує пропустити етап "двофакторна автентифікація", він переходить до наступного кроку. У випадку необхідності переходу до сторінки ідентифікації, система перенаправляє користувача на сервіс ідентифікації, який є частиною іншого проекту. Це робиться шляхом відправлення запиту з посиланням на URL, куди користувач має бути перенаправлений після авторизації. Це також дозволяє користувачам повернутися назад, що було додано як нова функціональність.

Переадресація здійснюється шляхом використання посилання з базовим URL до форми логіну, до якого додаються специфічні параметри. Важливими параметрами є:

- `SNA(serviceName) = myswisscom`, це вказує на оновлений інтерфейс з можливостями вводу двох факторної автентифікації;

- параметр, що визначає що необхідний саме процес двофакторної автентифікації, а не звичайний логін.

Для гарантування зручності користувачів і можливості контролювати свій досвід в системі, було додано параметр `cancelAllowed`, який активує можливість відміни процесу автентифікації. Це зокрема важливо, коли користувач бажає відмовитися від автентифікації після її розпочаття. В разі активації цієї опції система переадресує користувача назад на вказане посилання з поверненням, але з помилкою, яка вказує на те, що процес був свідомо припинений користувачем.

3.4.2. Огляд сервісу мокування

У фронтенд-частині проекту існує спеціалізований сервіс, який виконує мокування (імітацію) запитів для потреб локального тестування. Цей сервіс перехоплює всі вихідні запити, включаючи CRUD-операції та редиректи на інші сторінки, як, наприклад, сторінки авторизації.

При локальному запуску застосунку необхідно також активувати моки. Це дозволяє використовувати вікно налаштувань, де можна вказати різноманітні параметри для симуляції відповідей від сервера на кожен з перехоплених запитів.

До інтерфейсу налаштувань було додано можливість конфігурації відповідей сервера для запитів, які перевіряють статус сесії користувача. Це включає в себе відповіді про те що користувач вже має 2FA сесію, він її не має але може мати, і випадок коли все залишається як є.

Також сервіс моків обладнано здатністю перехоплювати та управляти редиректами на сторінку з двофакторною автентифікацією. Замість реальної сторінки автентифікації, мок-сервіс відображає спрощений інтерфейс із двома кнопками: "Успішна автентифікація" та "Відміна". Ці кнопки дозволяють імітувати успішне або зупинене

проходження процедури автентифікації, що є необхідним для правильного відображення етапу автентифікації.

Ці кнопки імітують поведінку реальної системи, передаючи необхідні параметри для подальшої обробки в системі, що дозволяє перевірити зміну тексту етапу на успішно пройденому автентифікації.

3.4.3. Імплементация функціоналу

У рамках проекту було створено ключовий ендпоінт `/mfa`, призначений для реалізації двофакторної автентифікації. Для його імплементации обрано використання мікросервісу `ose-soe-validation`, що займається валідацією різноманітних аспектів системи, і створено спеціалізований контролер для керування логікою автентифікації.

Для перевірки сесії користувача використовується сервіс `PermissionCheck`. Одною з функцій цього сервісу є перевірка сесій користувачів на основі ідентифікатора, який передається в ендпоінт як параметр. Для перевірки сесії функція потребує спеціальний параметр `"action"`, що визначає набір прав доступу залежно від типу сесії.

Визначено два типи `"action"`:

1. один, який дозволяє доступ із підтвердженою двофакторною автентифікацією;
2. інший для однофакторної автентифікації з потенційною можливістю переходу на двофакторну.

3.4.4. Логіка перевірки автентифікації

Спочатку система перевіряє наявність двофакторної автентифікації. Якщо користувач має двофакторну автентифікацію, подальші перевірки включають валідність наданого документу. Для цього ми перевіряємо, чи відмічений флаг `idAvailable` як `true` та порівнюємо його дату закінчення терміну придатності з датою обробки

запиту (LocalDate.now). Якщо умови не виконуються, система перевіряє можливість виконання однофакторної автентифікації. Це включає сценарій, де первинно зазначено, що потрібна двофакторна автентифікація, але при її неможливості приймається однофакторна. Для перевірок однофакторної автентифікації з можливістю до двофакторної автентифікації, використовується механізм try-catch для ловіння та обробки помилок. Це робиться тому, що у разі якщо користувач не має двофакторну автентифікацію, через певну помилку повернеться код advice. Якщо помилка вказує на конкретну потребу (наприклад, рада щодо двофакторної автентифікації), це означає що користувач має можливість отримати двофакторну (рис 3.2). Інші помилки без “advice” вказують на певний збій або спробу злому і в такому разі логується помилка .

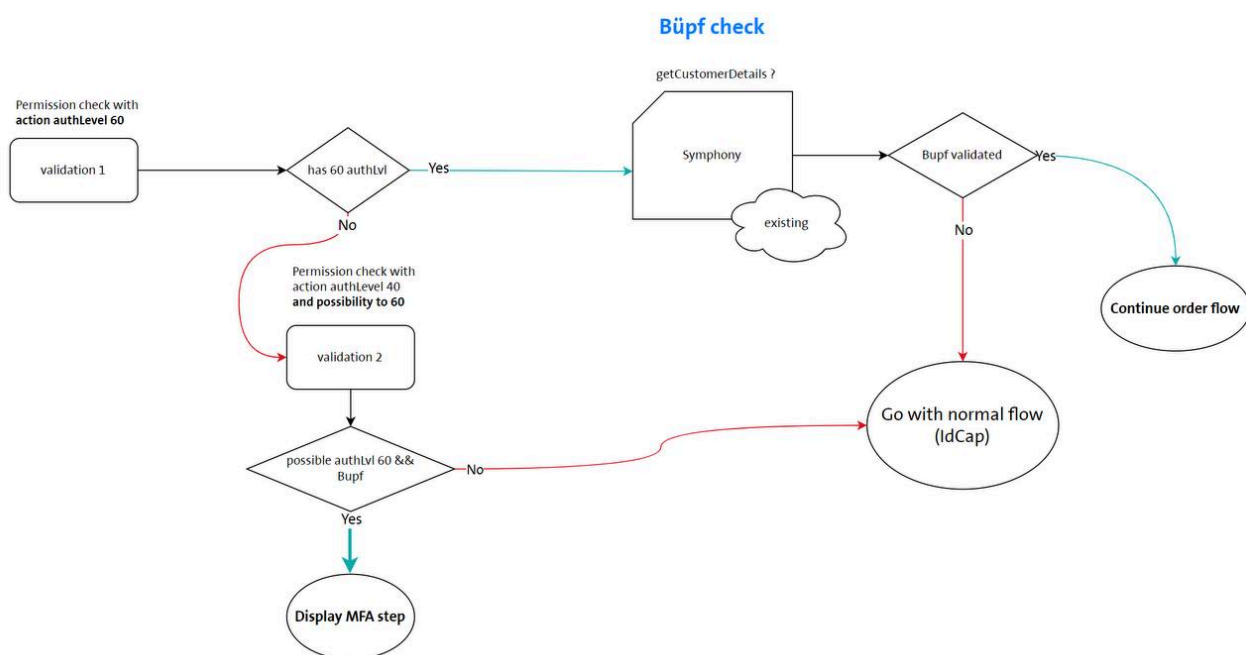


Рис. 3.2 – Логіка перевірки сесії та посвідчення користувача

Для отримання додаткової інформації про документи користувача виконується REST запит до мікросервісу ose-soe-customer, звідки збираються всі відомості про користувача. Цей процес забезпечує

вичерпне розуміння статусу автентифікації користувача та дозволяє налаштувати відповідні відгуки системи залежно від перевірених умов.

3.4.5. Інтеграція з існуючою інфраструктурою

Загалом уся розробка це інтеграція з існуючою інфраструктурою вже розроблених сервісів, але хотілось би виокремити важливий елемент безпеки який було додано до одного з сервісів. А саме виконано запобігання можливості покупки мобільного тарифу без належного підтвердження особи.

На мікросервісі валідації було створено новий ендпоінт, який задіяний у перевірці наявності валідних документів. Він є спрощеною версією ендпоінту який перевіряє сесію та валідний документ, тому що перевіряє тільки валідність документу. Для безпечного з'єднання між мікросервісами надається перевірка `PermissionCheck` яка відповідно створена для базових типів сесій яка саме є у з'єднанні мікросервісів. Цей запит є критично важливим для забезпечення того, що покупка мобільного тарифу можлива тільки при наявності підтверджених документів, що посвідчують особу користувача.

Запит до новоствореного ендпоінту в мікросервісі валідації використовується для перевірки документів в особливо визначених випадках. Цей механізм активується тільки для певних юз-кейсів, де це вважається необхідним, наприклад, при купівлі мобільного тарифу або інтернет-підключенні.

Для керування логікою перевірки документів, система використовує параметри такі як `use-case` і `kycSendLink`:

- `use-case`: цей параметр визначає тип транзакції (наприклад, перша покупка мобільного пакету). Він дозволяє системі ідентифікувати специфічні вимоги для кожної транзакції;

- `kycSendLink`: булевий прапорець, який вказує, чи потрібно відправити користувачу посилання для завантаження документів на пізніший час (`laterCase`).

Під час обробки замовлення, система спочатку категоризує юз-кейси як `Wireless` (мобільні підписки) або `Wireline` (інтернет-підключення). Використовуються асинхронні запити (`CompletableFuture`) до різних мікросервісів для збору потрібної інформації, після чого система очікує завершення всіх цих запитів (`CompletableFuture.allOf`) перед їхньою валідацією.

На етапі валідації, якщо юз-кейс відноситься до категорії `Wireless` і `laterCase` не був активований, система перевіряє наявність валідних документів. Якщо дані про використання сервісу `idcar` відсутні і відповідь з ендпоінту валідації вказує на відсутність валідних паспортних даних, система генерує помилку, що вказує на спробу обходу вимоги про надання документів. У інших випадках, процес вважається успішним, і транзакція продовжується без затримок.

3.5. Висновок

У проектній частині ми розробили систему двофакторної автентифікації, орієнтовану на забезпечення безпеки та гнучкості при взаємодії з користувачем. Використання мікросервісної архітектури дозволило нам забезпечити надійність та масштабованість системи, в той час як інтеграція з сучасними технологіями, такими як `Angular` на фронтенді та `Java` з `Spring Boot` на бекенді, сприяла створенню ефективного і стабільного сервісу. Важливою складовою проекту стала робота над забезпеченням безпеки через використання `JWT` для управління сесіями, що допомогло мінімізувати ризики несанкціонованого доступу.

Розроблена система двофакторної автентифікації не лише відповідає високим вимогам безпеки, але й пропонує гнучкість і масштабованість, що забезпечує можливість її адаптації під змінювані умови використання та потреби користувачів.

РОЗДІЛ 4

Експериментальна частина

4.1. Тестування системи

У рамках проектної частини "Тестування системи", особлива увага була приділена перевірці стабільності та відповідності мікросервісів до заданих специфікацій, з акцентом на мікросервіс `submitorder`, який відіграє вирішальну роль у процесі оформлення покупок. Цей мікросервіс був обраний як критичний елемент системи, що вимагає детального тестування, особливо при інтеграції з іншими компонентами системи.

Під час ранньої стадії випробувань у тестовому середовищі було виявлено, що на етапі інтеграційного тестування виникає помилка, яка не була ідентифікована під час юніт-тестів. Специфічно, помилка проявлялася як неправильно сформовані параметри запиту, що призводило до відмови у обслуговуванні під час спроб оформлення будь-якого виду покупки. Ця ситуація вимагала термінового відкату версії деплойменту для стабілізації сервісу та подальшого аналізу виявленої проблеми.

Після детального вивчення логів та логіки передачі параметрів було зрозуміло, що спосіб передавання параметрів до запиту який я зазначила був некоректним. Надалі для підвищення безпеки та стабільності процесів, запити було обрано конструкцією `try-catch`. У випадку виявлення помилки в реальному часі, система не зупиняється, а видає лог помилки та повертає статус `false`, що дозволяє системі продовжувати роботу, поки проблема обробляється.

4.2. Тестування Cypress

Перед початком процесу тестування, було проведено ретельний аналіз існуючих тестів, щоб забезпечити їх відповідність стандартам і стилю написання коду проекту. Одним з ключових виявлень було те, що селектори елементів інтерфейсу краще винести в окремий файл, щоб підтримувати чистоту і структурованість коду тестів. Це дозволяє легше управляти змінами і швидше адаптуватися до нововведень у проекті.

Тестування було заплановано для двох середовищ: локального і тестового (Vega). У локальному середовищі, для імітації сторінки двофакторної автентифікації, було створено селектори, що відповідають двом кнопкам: "success" та "cancel" з сервісу мокування. Ці кнопки дозволяли імітувати результати автентифікації без необхідності взаємодії з реальним сервісом автентифікації.

Загальна кількість написаних тестів склала п'ять (рис. 4.1), кожен з яких був спрямований на перевірку конкретного аспекту системи.

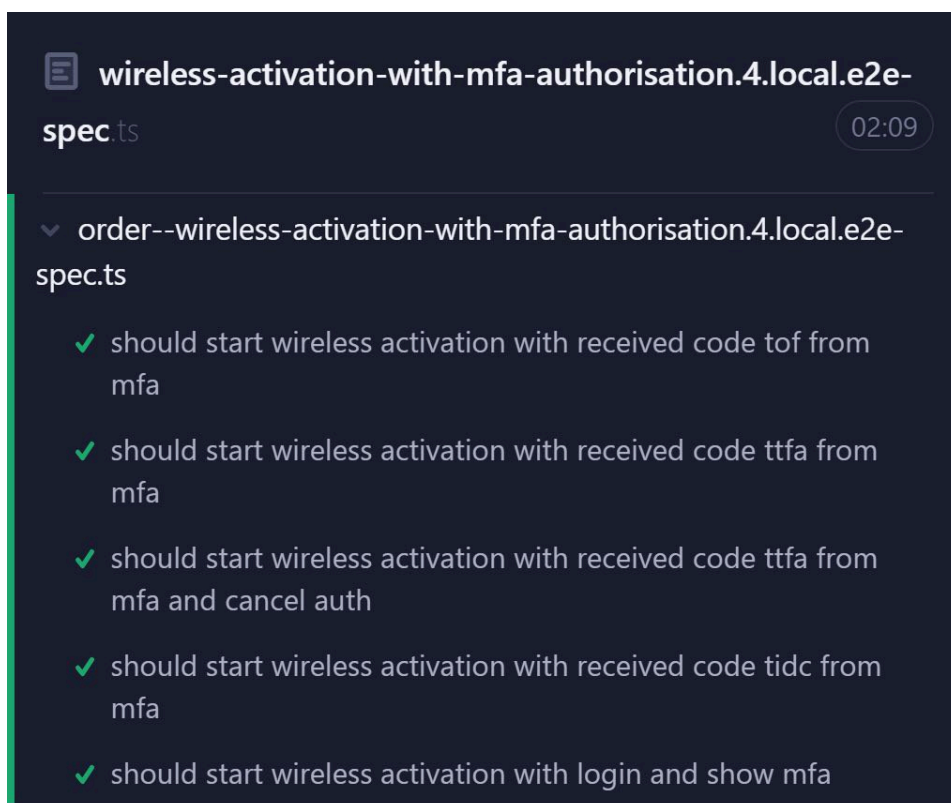


Рис. 4.1. – Успішно пройдені тести в локальному середовищі

Перший тест перевіряє, чи пропускаються всі етапи ідентифікації, включаючи двофакторну, коли система отримує код, який ідентифікує користувача як такого, що перебуває в двофакторній сесії і має валідний документ.

Другий тест перевіряє, чи відображається розроблений етап з двофакторною автентифікацією, якщо користувач налаштував двофакторну автентифікацію і має валідний документ. Цей етап включає перевірку зникнення етапу ідентифікації після натискання на кнопку "success".

Третій тест аналогічний другому, але він включає перевірку залишення етапу автентифікації після натискання на кнопку "cancel".

Четвертий тест перевіряє сценарії, коли етап двофакторної автентифікації має бути пропущений, але процес ідентифікації документів має відображатися, якщо користувач не має можливості перейти на двофакторну автентифікацію або не має валідного документа.

Останній тест перевіряє, що система коректно обробляє вхід неавторизованого користувача під час оформлення замовлення і виконує перевірку його сесії та документа.

Тестування в середовищі Vega проводилося за допомогою реальних даних з тестового акаунта, який було спеціально створено для тестування Cypress. Це дозволило перевірити реалістичність взаємодії користувача з системою та точність відтворення заданих сценаріїв. Усі тести були успішно пройдені, що підтверджує надійність розробленої системи та її готовність до експлуатації в продуктивному середовищі.

4.3. Сценарії тестування

Сценарії тестування для проекту були розроблені для перевірки інтеграції двофакторної автентифікації (2FA) в різні юз-кейси активації мобільних та конвергентних сервісів. Основна мета полягає в тому, щоб

переконалися, що система адекватно обробляє входи користувачів з урахуванням статусу 2FA та інших умов, таких як наявність дійсних документів, увімкненого feature-toggle.

4.3.1. Визначені юз-кейси

- wireless-prepaid-activation – активація передплатених мобільних тарифів;
- wireless-pre2post – перехід з передплати на постплату;
- convergent-activation-pre2post – перехід конвергентних сервісів з передплати на постплату;
- wireless-activation – активація мобільного зв'язку;
- wireless-activation-blitz – швидка активація мобільного зв'язку;
- convergent-activation – активація конвергентних сервісів.

4.3.2. Сценарії тестування

1. Wireless Activation: Користувач уже увійшов у систему, успішна автентифікація 2FA.
 - Перевіряється, що якщо автентифікація 2FA пройшла успішно, процес ідентифікації (IDCap) пропускається.
2. Wireless Activation: Користувач уже увійшов, виконано умови BūPF, успішна автентифікація 2FA.
 - Перевіряється, що процес IDCap пропускається при виконанні умов BūPF після успішної автентифікації 2FA.
3. Wireless Activation: Користувач увійшов, автентифікація 2FA не успішна.
 - Якщо автентифікація 2FA не пройшла успішно, система продовжує звичайний процес IDCap.
4. Wireless Activation: Користувач увійшов, доступна активна 2FA, виконано умови BūPF, успішне виконання 2FA.

- Тестує можливість пропуску IDCap після успішного виконання 2FA при виконанні умов BūPF.
5. Wireless Activation: Користувач увійшов, 2FA не активна.
- Процес IDCap виконується як зазвичай, оскільки 2FA не активована.
6. Wireless Activation: Анонімний вхід, без 2FA.
- Процес IDCap виконується як зазвичай для анонімного користувача без активної 2FA.
7. Wireless Activation: Анонімний до 2FA входу, успішна валідація за допомогою коду, виконано умови BūPF.
- При успішній валідації та виконанні умов BūPF процес IDCap пропускається.
8. Wireless Activation: Анонімний до 2FA входу, валідація за допомогою коду не успішна.
- Продовження звичайного процесу IDCap у разі невдалої валідації коду.
9. Перевірка AuthMethod для 2FA успішного замовлення в CPQ - Symphony встановлено на 2FA_Online.
- Перевіряється, чи правильно встановлено статус автентифікації у системі Symphony для успішних 2FA замовлень.
10. Перевірка AuthMethod для замовлень 2FA успішно без 2FA - Symphony встановлено на Online.
- Перевіряється відповідність методу автентифікації у системі Symphony для замовлень, де 2FA не була активована.
11. Перевірка статусу автентифікації (нове поле) в Symphony для замовлень 2FA.
- Виконується перевірка нового поля статусу автентифікації у Symphony для замовлень з активованою 2FA.

4.4. Опис процедури тестування

Тестування системи перед запуском у продакшн відбувається у спеціалізованому тестовому середовищі під назвою Vega. Це середовище реплікує поведінку продакшн-системи з декількома специфічними відмінностями, які адаптовані для потреб тестування.

Середовище Vega має власні бази даних і набори користувачів, які ізольовані від продакшн-середовища. Це дозволяє проводити тести без ризику для реальних даних користувачів і бізнес-операцій. Система налаштована таким чином, щоб всі процеси та функції, які працюють у Vega, ідентично відтворювались на продакшн, з кількома винятками для специфіки тестування.

4.4.1. Спеціальні налаштування для тестування

- Мобільні підтвердження: для підтвердження через мобільний телефон використовується стандартний код (наприклад, 513513), що дозволяє тестувати функціональність без використання реальних SMS-повідомлень;
- Фізичні листи: листи, які мали б відправлятися для підтвердження особи, не відправляються. Натомість тестування проводиться в умовах, де ці процедури підтвердження можуть бути емульовані або відмінені.

Для тестування розробки було створено спеціальний тестовий аккаунт і налаштовано сервісом MFA (Мультифакторної автентифікації). Для цього було змінено ім'я користувача на спеціальний формат, який в сервісі автентифікації розглядається як виняток і обробляється по особливому. Після цього до аккаунту було додано спеціальний замокований номер телефону який автоматично вводить пароль з двох факторної автентифікації.

4.5. Аналіз результатів тестування

Під час проектної частини "Тестування системи", особлива увага була приділена перевірці стабільності та відповідності мікросервісів до заданих специфікацій, з акцентом на мікросервіс `submitorder`, який відіграє вирішальну роль у процесі оформлення покупок. Цей мікросервіс був обраний як критичний елемент системи, що вимагає детального тестування, особливо при інтеграції з іншими компонентами системи.

На ранніх стадіях випробувань у тестовому середовищі виявлено помилку неправильно сформованих параметрів запиту, що спричинило відмову у обслуговуванні під час спроб оформлення будь-якого виду покупки. Детальний аналіз логів і параметрів запитів показав, що основною причиною була некоректна передача параметрів, що не була помічена на етапах раннього кодування. Помилку було виправлено, а надалі запити було обрано конструкцією `try-catch`, що дозволить в майбутньому системі автоматично обробляти помилки і продовжувати роботу, підвищуючи її безпеку і стабільність.

Додатково, під час кінцевого етапу `end-to-end (E2E)` тестування було виявлено, що флаг `authMethod` не змінювався відповідно до вимог, залишаючись неактивним, тобто жоден параметр не був записаний під цей флаг. Після детального аналізу з'ясувалося, що зміна цього флагу до необхідного значення `"2FA_Online"` тимчасово неможлива через необхідність змін в коді, які повинна була внести інша команда розробників. Тимчасовим рішенням стало призначення `authMethod` як `"Online"`, що дозволило продовжити тестування та використання системи, поки не будуть внесені відповідні зміни.

Ці випробування наголосили на важливості всебічного тестування та готовності до швидкого виявлення та усунення помилок. Вони також

показали необхідність тісної співпраці між командами для забезпечення відповідності системи встановленим стандартам і вимогам.

ВИСНОВКИ

У рамках даної дипломної роботи були успішно виконані завдання, які стосувалися підключення системи автентифікації до існуючого процесу покупки, визначення вимог до системи, а також реалізація програмного забезпечення та його тестування.

Огляд алгоритмів автентифікації.

Робота розпочалась з детального огляду існуючих алгоритмів автентифікації, що дозволило зрозуміти технології автентифікації для інтеграції у систему покупки. Вибір був зосереджений на методах, які забезпечують високий рівень безпеки та відповідність сучасним вимогам, таких як двофакторна автентифікація (2FA).

Визначення вимог до системи та побудова структури.

На базі огляду алгоритмів було встановлено технічні та функціональні вимоги до системи. Це включало розробку специфікацій для інтеграції системи автентифікації з існуючими процесами покупки, врахування необхідності валідації типів сесій користувачів та перевірки валідності документів. Було визначено необхідність повторною валідації під час оформлення замовлення для підвищення безпеки. Було побудовано структуру системи, яка забезпечує гнучкість, масштабованість та високу інтеграційну спроможність.

Реалізація програмного забезпечення.

Програмне забезпечення було розроблено з використанням сучасних технологій та мов програмування. Реалізація включала модифікацію існуючого інтерфейсу користувача для оброблення автентифікації та введення змін, що дозволяють зберігати інформацію про тип автентифікації у системі разом з валідацією параметрів для підвищення безпеки.

Проведення прикладного тестування.

Тестування системи виконувалось у спеціалізованому тестовому середовищі Vega, що дозволило імітувати реальні умови експлуатації. Тестування підтвердило ефективність реалізації нових функцій, стабільність роботи системи під навантаженням та її здатність відповідати встановленим вимогам безпеки.

Загальні висновки.

В результаті було успішно інтегровано розширені функції автентифікації у процес онлайн-покупок. Система наділена новим процесом оформлення замовлень, що характеризується підвищеним рівнем безпеки та зручністю використання. Ці зміни не лише підвищують загальну безпеку процесу покупки, але й сприяють зростанню довіри та задоволення користувачів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ 3008-2015: Державний стандарт України «Документація. Звіти у сфері науки і техніки. Структура і правила оформлення». – Київ:ДП «УкрНПНЦ», 2016. – 25 с.
2. Windley P. Digital Identity. O'Reilly Media, Inc., 2005. 254 p..
3. Види автентифікації. – [Електронний ресурс]. – Режим доступу: <https://rublon.com/blog/what-are-the-three-authentication-factors/> (Дата звернення: 19.04.2024).
4. RxJs - Реактивне програмування. – [Електронний ресурс]. – Режим доступу: <https://rxjs.dev/guide/operators> (Дата звернення: 28.04.2024).
5. NgRx - Концепція NgRx. – [Електронний ресурс]. – Режим доступу: <https://ngrx.io/guide/store> (Дата звернення: 28.04.2024).
6. LocalDate (Java Platform SE 8). - Документація Oracle Java. – [Електронний ресурс]. – Режим доступу: <https://docs.oracle.com/javase/8/docs/api/java/time/LocalDate.html> (Дата звернення: 30.04.2024).
7. CompletableFuture (Java Platform SE 8). – Документація Oracle Java. – [Електронний ресурс]. – <https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/CompletableFuture.html> (Дата звернення: 10.05.2024).
8. ДСТУ 8302:2015: Державний стандарт України «Інформація та документація. БІБЛІОГРАФІЧНЕ ПОСИЛАННЯ. Загальні положення та правила складання». – Київ: ДП «УкрНДНЦ», 2016. – 16 с.