

Dokumentacja z projektu ZSD 2024

Aplikacja mobilna do zakupu wejściówek

<https://git.pg.edu.pl/p1350715/aplikacja-mobilna-do-zakupu-wejsciwiek-projekt-zsd>

Celem projektu było stworzenie aplikacji mobilnej umożliwiającej zakup wejściówek na skatepark oraz karnetów na zajęcia wykorzystującą funkcję NFC oraz kody QR do sprawdzania i potwierdzania wejść. Drugą częścią projektu jest aplikacja na komputer pokazująca informacje o zakupionych wejściach oraz karnetach.

Zespół:

nr	imię i nazwisko	indeks
1	Maksym Małek	188809
2	Oliwer Lisek	188947
3	Przemek Piątkiewicz	188823
4	Jakub Dobruchowski	188868
5	Katarzyna Łabuńska	188886
6	Maksymilian Mieszkowski	188871
7	Kamil Pipka	188967
8	Kewin Trochowski	188860
9	Sebastian Parzych	188609
10	Michał Zarzycki	184856

Kamienie milowe:

Milestone 1

- Stworzenie szkieletu aplikacji z wykorzystaniem narzędzia flutter umożliwiającej rejestrację/logowanie, przenoszenie do nowych okien aplikacji, podstawowe poruszanie się po aplikacji
- Stworzenie bazy danych do przechowywania informacji o użytkownikach w firebase

Milestone 2

- Stworzenie aplikacji desktopowej w języku python do podstawowej obsługi czytnika nfc
- Dodanie do aplikacji mobilnej i desktopowej funkcji komunikacji za pomocą czytnika nfc
- Generowanie kodów qr unikalnych dla klienta

Milestone 3

- Dodanie do aplikacji funkcji mikropłatności
- Gotowy produkt aplikacji mobilnej na systemy android i ios umożliwiającej rejestrację użytkownika lub logowanie, zakup wejść na skatepark jednorazowych i karnetów na zajęcia.
- Sprawdzanie ważności wejścia za pomocą aplikacji komputerowej używając czytnika nfc (lub za pomocą skanowania kodów qr)
- Dokumentacja i raport z projektu
- Udostępnienie aplikacji na Google Play Store

1. Baza danych


Dane o każdym użytkowniku przechowywane są w bazie danych na platformie Firebase. Używana baza to Firestore database. Przechowywane są dane użytkownika oraz jego aktualny stan wejściówek oraz karnetów. Hasła użytkowników są szyfrowane i nie są widoczne w bazie z poziomu konsoli (Rys. 6). Na potrzeby testowania aplikacji w pliku Rules bazy ustawiony jest publiczny dostęp z kluczem. Aplikacja mobilna oraz desktopowa posiadają klucze dostępu do bazy i wysyłają zapytania oraz modyfikują zawartości w tej bazie danych. Tworzenie kont, wysyłanie maila z potwierdzeniem oraz zabezpieczenie danych zostało zrealizowane przy pomocy biblioteki do Fluttera `firebase_auth` (https://pub.dev/packages/firebase_auth). Treści maili potwierdzającego oraz zmiany hasła są modyfikowalne z poziomu konsoli Firebase.

2. Logowanie, rejestracja oraz resetowanie hasła.

Ekran startowy aplikacji jest oknem logowania. Jeśli użytkownik nie posiada konta, może założyć je klikając w guzik rejestracji widoczny na Rys. 1. Jeśli użytkownik posiada już konto, lecz nie pamięta hasła powinien wpisać swój email, a następnie nacisnąć guzik resetu hasła. Na maila użytkownika zostanie wysłany email z linkiem do resetu hasła (Rys. 4). W przypadku chęci rejestracji użytkownik musi wypełnić wszystkie pola rejestracji i kliknąć guzik „Zarejestruj się”. Mail potwierdzający zostanie wysłany do użytkownika (Rys. 5). Należy potwierdzić maila klikając w link w mailu. Wtedy konto jest ważne i można się zalogować. Nie można używać maila wielokrotnie. Jedynym wyjątkiem jest usunięcie konta. W takim przypadku można założyć na swojego maila nowe konto. Logowanie odbywa się metodą Email/Password. Użytkownik aby poprawnie zalogować się na swoje konto musi wpisać swój adres email oraz poprawne hasło. Dane weryfikowane są z bazą danych i jeśli są poprawne użytkownik zostaje przepuszczony do głównego ekranu. W przeciwnym wypadku pojawia się komunikat o niepoprawnych danych.



Rys. 1 Startowy ekran logowania.



Rys. 2 Ekran rejestracji.

<input type="checkbox"/>	RollMasters App	Zresetuj hasło do aplikacji RollMasters	Dzisiaj, 20:24
--------------------------	-----------------	---	----------------

Rys. 3 Mail resetujący hasło

← Odebrane

Odpowiedz ▾ → Prześlij dalej ↻ Przenieś 🗑 Usuń 🚫 Zgłoś spam ⋮ ⬅ ➡

Od: RollMasters App,

Do: mnie ramosaso@wp.pl

Temat: Zresetuj hasło do aplikacji RollMasters

Bezpieczeństwo: Szyfrowanie TLS [Więcej informacji](#)

3 cze 2024 20:24 (przed chwilą)

▼

Witamy,
Kliknij ten link, aby zresetować hasło aplikacji RollMasters powiązane z Twoim kontem ramosaso@wp.pl.
https://mhapp-bbd3f.firebaseio.com/__/auth/action?mode=resetPassword&oobCode=MGN1ICRdaHyu0wl2-zhoRiAatRiAeTYvZNeSY2u3ZTkAAAGP31ikvA&apiKey=AlzaSyBoS5z8c_s4ycUDxorZCzXQQMSHQxmHmKU&lang=pl
Jeżeli nie prosisz o zresetowanie hasła, zignoruj tego e-maila.
Pozdrawiamy
Zespół aplikacji RollMasters

Rys. 4 Treść maila resetującego hasło

Witamy,

Kliknij ten link, aby zweryfikować swój adres email.

https://mhapp-bbd3f.firebaseio.com/_/auth/action?mode=verifyEmail&oobCode=R_DWweNhs1moCoonSankSjEesm9OqcU3npyEPwsqWTwAAAGPuvkMeA&apiKey=AlzaSyBoS5z8c_s4ycUDxorZCzXQQMSHQxmHmKU&lang=pl

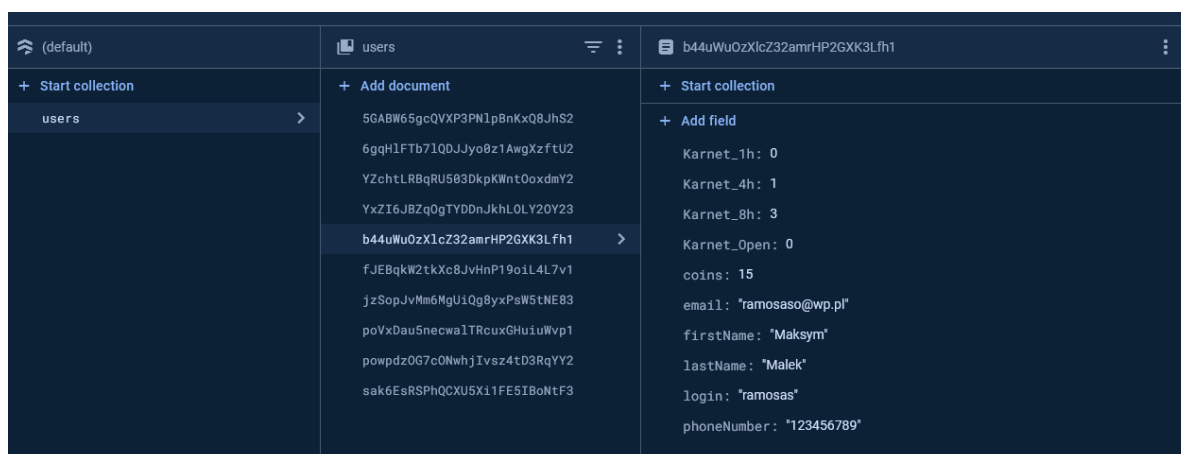
Jeżeli nie prosisz o weryfikację adresu, zignoruj tego e-maila.

Pozdrawiamy

Zespół aplikacji mhapp-bbd3f



Rys. 5 Treść maila weryfikacji adresu e-mail.



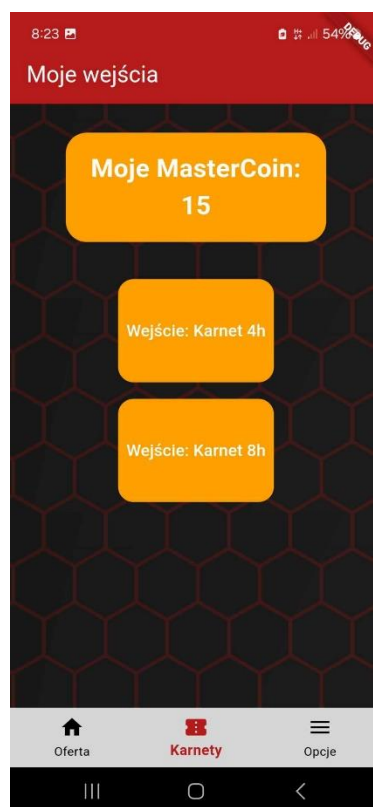
Rys. 6 Przykładowy użytkownik w konsoli bazy Firebase

3. Ekran aplikacji po zalogowaniu

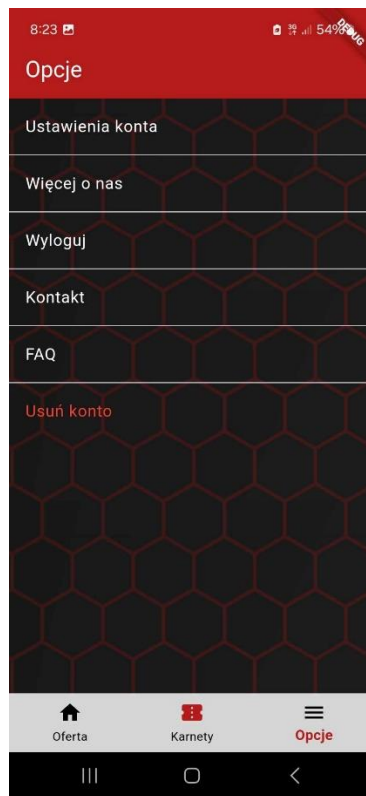
Po udanym zalogowaniu wyświetlony zostaje ekran z trzema klawiszami oraz dolne menu wyboru. W dolnym menu użytkownik może nawigować między trzema głównymi ekranami widocznymi na Rys. 7, Rys. 8, Rys. 9. Są to „Oferta”, „Karnety” i „Opcje”.



Rys. 7 Widok główny po zalogowaniu (Oferta)

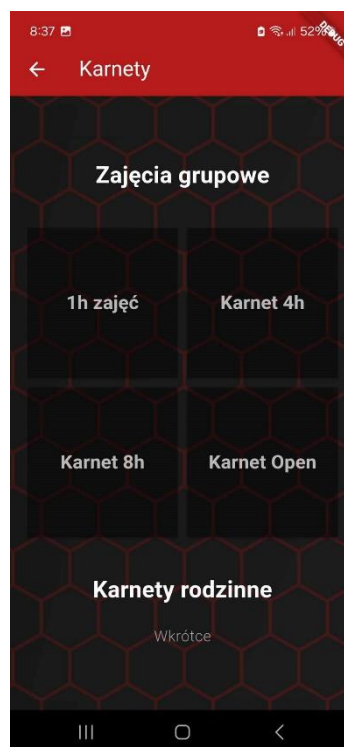


Rys. 8 Środkowy widok karnetów

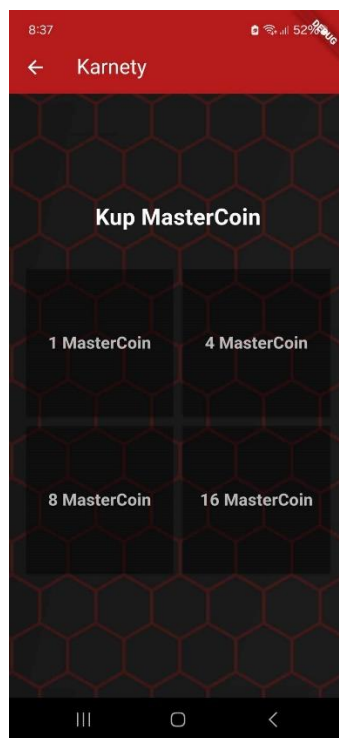


Rys. 9 Trzeci widok (Opcje)

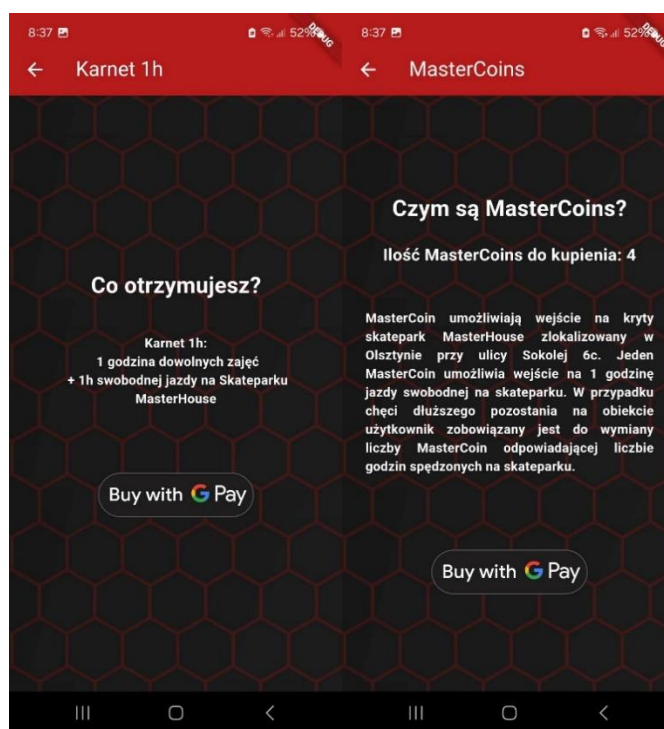
Bedąc w głównym menu możemy zobaczyć dostępne do kupna karnety oraz wejściówki. Rolę wejściówek pełni MasterCoin. Jeden MasterCoin odpowiada wejściu na jedną godzinę swobodnej jazdy na skateparku. Oferta karnetów widoczna jest na Rys. 10, a oferta MasterCoin na Rys. 11.



Rys. 10 Oferta karnetów



Rys. 11 Oferta MasterCoin



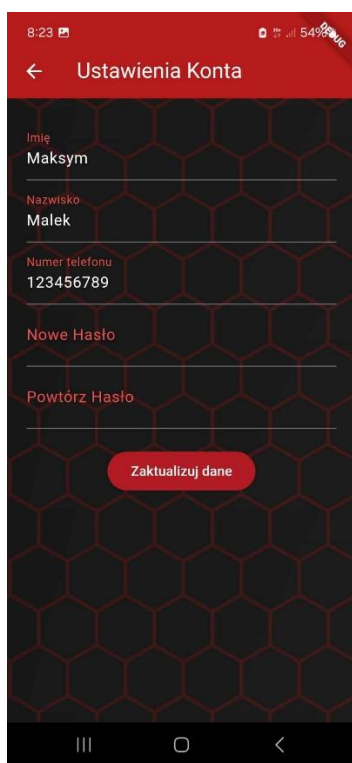
Rys. 12 Widok kupna karnetów oraz MasterCoin

Po wybraniu oferty, którą zainteresowany jest użytkownik, następuje przeniesienie do ekranu informacyjnego. Możliwy jest zakup z Google Pay. W serwisie Google Pay konieczne jest podpięcie działającej karty kredytowej lub debetowej. Na potrzeby projektu wykorzystane jest testowe środowisko zakupu udostępnione przez Google i środki z karty nie są pobierane, a płatność jest jedynie symulowana. W celu komercjalizacji aplikacji do środowiska testowego

należy podłączyć swoje konto firmowe z danymi podatkowymi i kontem bankowym, na które przelewane są środki z transakcji. Pokazanie działania kupna jest niemożliwe ze względu na zabezpieczenie płatności przed udostępnianiem ekranu lub zrzutami ekranu.

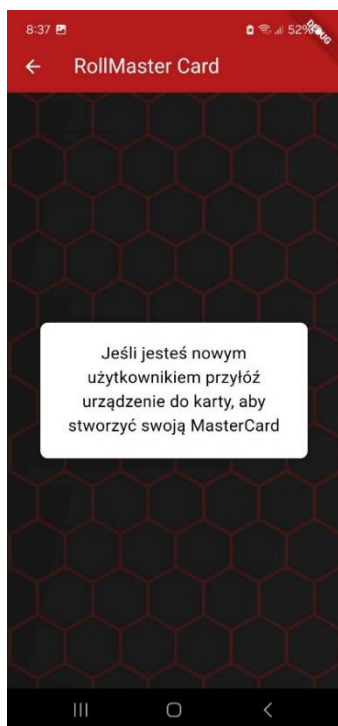
Po kupnie wyświetlany jest komunikat o poprawnym zakupie lub błędzie i nieobciążeniu konta. Efekty zakupu można zobaczyć na ekranie „Karnety”, gdzie po kliknięciu odświeżany jest aktualny stan wejściówek i karnetów użytkownika (Rys. 8).

Ekran „Opcje” (Rys. 9) posiada menu listowe. Znajdują się w nim ustawienia konta (Rys. 13), w których użytkownik może zmienić swoje dane i hasło. Pozycja „Więcej o nas” przenosi do strony internetowej, gdzie znajdują się wszystkie informacje dotyczące skateparku i szkółki. Dostępna jest również opcja wylogowania się ze swojego konta lub usunięcia go. FAQ jest pustym guzikiem pozwalającym na rozbudowanie aplikacji w przyszłości.



Rys. 13 Ustawienia konta

Bedąc w głównym menu (Rys. 7) użytkownik może wybrać opcję RollMaster Card. Przenosi ona do ekranu programowania karty NFC (Mifare1k). Należy przyłożyć kartę do urządzenia z włączoną funkcją NFC, i odczekać aż pojawi się komunikat o pomyślnym zakodowaniu informacji na kartę. Użytkownik posiada wtedy własną kartę, która może w każdym momencie zostać przyłożona do czytnika, gdy otwarta jest aplikacja desktopowa, by uzyskać szczegółowe informacje na temat wejść użytkownika.



Rys. 14 Widok zapisu na RollMasterCard

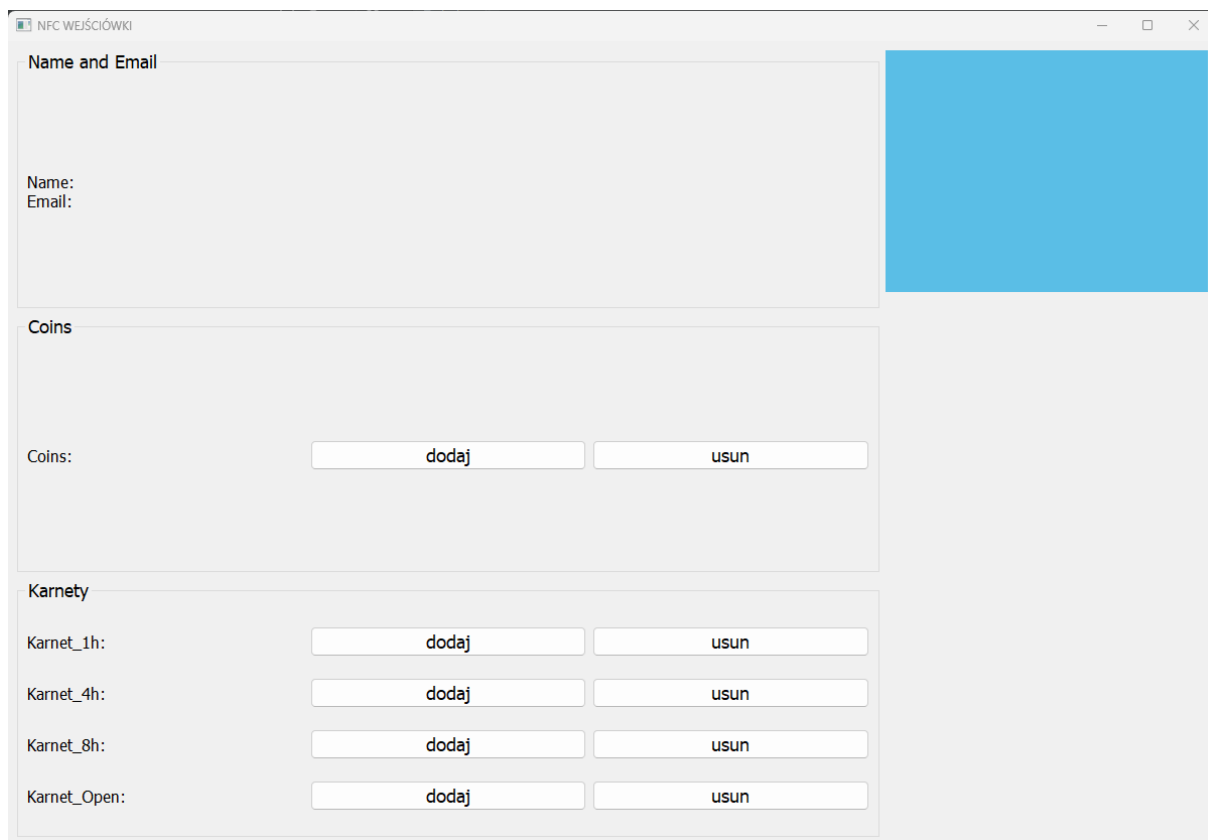
Aby umożliwić udostępnienie aplikacji na Google Play Store założone zostało konto dewelopera i po weryfikacji danych przez Google udostępniony został dostęp do konsoli dewelopera, gdzie po wypełnieniu koniecznych danych i akceptacji warunków możliwe było wypuszczenie aplikacji do wewnętrznych testów. Dostęp do aplikacji realizowany jest poprzez wpisanie adresu mailowego serwisu G-mail w konsoli sklepu Play oraz przyjęciu wygenerowanego zaproszenia.

(<https://play.google.com/apps/internaltest/4699800244431756184>).

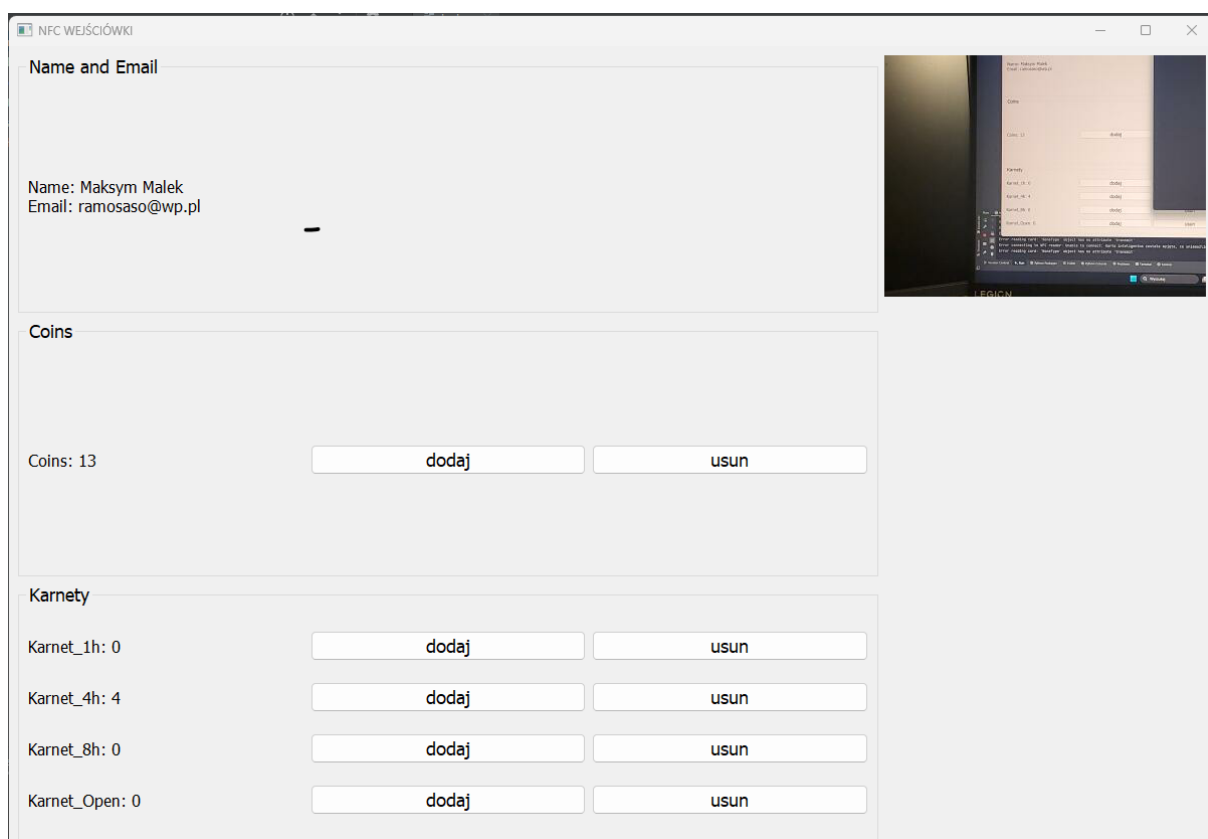
Po przyjęciu zaproszenia (oraz posiadaniu kompatybilnego urządzenia z Android SDK > 21) możliwe jest pobranie aplikacji ze sklepu Play.

4. Aplikacja desktopowa

Aplikacja desktopowa pozwala na odczytywanie danych o użytkowniku oraz dodawaniu lub odejmowaniu wejść i karnetów użytkownikowi. Do korzystania z funkcji odczytu karty NFC konieczny jest czytnik NFC ACR122u oraz konieczne jest zainstalowanie sterowników (<http://www.acs.com.hk/download-driver-unified/9840/ACS-Unified-MSI-4280.rar>). Do korzystania z opcji czytnika kodów QR konieczna jest zewnętrzna aplikacja DroidCam na komputerze oraz telefonie, którego kamera będzie wykorzystywana do skanowania kodu. Po poprawnym przyłożeniu karty użytkownika do czytnika na ekranie pojawią się informacje o użytkowniku (Rys. 16). Jako czytnik kodów QR wykorzystywany jest telefon. Po podłączeniu telefonu przez DroidCam kablem bądź przez Wi-Fi w prawym górnym rogu pojawia się obraz z kamery telefonu. Po zeskanowaniu kodu QR wygenerowanego przez użytkownika na ekranie pojawi się komunikat o poprawnym zeskanowaniu kodu i dane o użytkowniku pokażą się na ekranie. Po zeskanowaniu kodu wygenerowanego z MasterCoin dodatkowo pokaże się komunikat ile MasterCoin pobrać z konta użytkownika. Gdy kod QR jest niepoprawny, a zawiera informacje o koncie (np. screenshot przedawnionego kodu QR), na ekranie pojawi się komunikat o niepoprawnym kodzie dla danego użytkownika.



Rys. 15 Widok pusty aplikacji desktopowej



Rys. 16 Widok aplikacji desktopowej

5. Rozwiązania techniczne

Karty NFC unikalne dla użytkowników wykorzystywane jako RollMaster Card są kartami MiFare o pamięci 1kB. Karty te są szeroko stosowane jako wizytówki, klucze lub gadżety i pozwalają na zaprogramowanie na nich linków, tekstu lub danych kontaktowych. Ich zaletą jest wykonanie z PVC, co umożliwia łatwy wydruk swojego designu lub logo na karcie. Aplikacje desktopowa oraz mobilna oprócz wspierania kart MiFare1k, powinny być kompatybilne z kartami o pamięci 4kB oraz tagami NFC. Nie zostało to jednak na chwilę pisania raportu przetestowane.

Czytnik ACR122u jest czytnikiem obsługującym karty w formacie MiFare, ISO oraz tagi NFC. Aplikacja desktopowa wspiera jedynie ten model czytnika, a komendy i funkcje napisane w kodzie wykorzystują komendy jedynie obsługujące ten model.

Czytanie i rozszyfrowywanie kodów QR zrealizowane jest na poziomie aplikacji desktopowej.

Wykorzystane zostały do tego biblioteki OpeCV-python oraz pyzbar

(<https://pypi.org/project/opencv-python/> <https://pypi.org/project/pyzbar/>).

Aby poprawnie przesłać obraz z telefonu wykorzystywanego jako czytnik kodów QR użyta została zewnętrzna darmowa aplikacja DroidCam.

Do generacji unikalnych kodów QR z id użytkownika i typem wejściówki została wykorzystana biblioteka do Fluttera qr_flutter (https://pub.dev/packages/qr_flutter).

6. Problemy napotkane podczas realizacji aplikacji mobilnej

Pierwszym problemem napotkanym podczas tworzenia aplikacji mobilnej był brak dynamicznej zmiany rozmiarów części front-endowej aplikacji. Tło oraz przyciski miały stałe wymiary, co często prowadziło do błędów typu overflow i nieprawidłowego widoku na urządzeniach innych wymiarów. Po zmianie wymiarów na dynamicznie dostosowujące się do wymiarów urządzenia oraz zmianie widoku na scrollview powodującego konieczność przesuwania palcem po ekranie w przypadku dłuższej listy pionowej problemy tego typu pojawiały się sporadycznie i były szybko naprawiane.

Kolejnym problemem była praca na dużej ilości plików .dart. Każdy ekran aplikacji wymagał stworzenia oddzielnej klasy, które rozdzielone po plikach utrudniały często wprowadzenie nowego rozwiązania, które musiało być wprowadzane na wielu plikach.

Trudnością napotkaną na końcowym etapie prac było niepoprawne działanie usuwania konta. W środowisku symulacyjnym Android Studio nie było problemów z usuwaniem konta, lecz po zbudowaniu aplikacji na telefon pojawiał się problem, że konta usuwały się jedynie pozornie. Problemem był wymóg technologii firebase_auth do posiadania statusu „recent login”. Po dodaniu konieczności podania hasła przed usunięciem konta błąd przestał występować.

7. Problemy napotkane podczas realizacji aplikacji komputerowej

Głównym problemem napotkanym podczas pisania aplikacji był brak szczegółowych informacji dostępnych w dokumentacji czytnika ACR122u. Brak dostępu do wiedzy spowodował trudności w wysyłaniu prawidłowych komend do czytnika, a tym samym odczytywaniu danych z określonych sektorów pamięci kart MiFare1k. Podstawowym działaniem czytnika jest odczytywanie stałego Uid karty, lecz w projekcie kluczowy był zapis i odczyt danych z sektorów pamięci karty, by umożliwić zapisanie id użytkownika na karcie. Uid

karty oraz id użytkownika nie mogły być takie same. Rozwiązaniem problemu było zgłębienie wiedzy na temat działania czytnika oraz tagów NFC różnego rodzaju. Dużym rozszerzeniem posiadanej wiedzy okazała się być biblioteka <https://github.com/Flowtter/py-acr122u> do języka Python.

Innym problemem był brak posiadania czytnika kodów QR. Stosunkowo prostym rozwiązaniem było w tym przypadku streamowanie obrazu z kamery telefonu do aplikacji, gdzie obraz jest analizowany pod kątem znajdowania kodu QR. Plusem takiego obejścia konieczności posiadania czytnika jest bezpłatność rozwiązania. Minusem konieczność zainstalowania zewnętrznej aplikacji DroidCam oraz na komputer oraz na urządzenie mobilne mające służyć jako czytnik.