

Andrzej Chmielewski (L5) - **51408**

Karol Hulka (L4) - **49380**

Dominik Maciński (L4) - **54366**

Konrad Komorowski (L4) - **55256**

Jacek Bryłka (L4) - **54351**

System wspierający pracę serwisu sprzętów mobilnych

Github.com:

[ProjektAEH](#) - [github.com/ProjektAEH](#)

[github.com/orgs/ProjektAEH/projects/3](#)

[github.com/ProjektAEH/51408_49380_54366_55256_54351](#)

Celem niniejszej dokumentacji jest przedstawienie innowacyjnego i nowoczesnego systemu webowego, który usprawnia pracę serwisu sprzętu mobilnego. Aplikacja została zaprojektowana w celu usprawnienia obsługi klientów oraz poprawy jakości usług. Główne funkcje aplikacji obejmują przyjmowanie urządzeń od klientów, wyświetlanie adresów wysyłkowych, tworzenie zleceń serwisowych, zarządzanie naprawami i zleceniami oraz wysyłanie powiadomień o statusie działań.

Jednym z kluczowych aspektów serwisu jest możliwość aktualizacji oprogramowania urządzeń, które nie są już wspierane przez producentów, co przeciwdziała ich porzuceniu i utylizacji. Dzięki aplikacji użytkownicy mogą bezpiecznie i ekologicznie przedłużyć życie swoich urządzeń mobilnych, korzystając z najnowszych wersji oprogramowania, takich jak Android 11, a w przyszłości Android 12. Takie rozwiązanie przyciągnie użytkowników, dla których ochrona środowiska jest priorytetem.

Zalogowani użytkownicy mają dostęp do danych osobowych klientów, przeglądu aktualnie realizowanych zleceń oraz historii zleceń. Mogą również dodawać opisy zgłoszeń i uwagi dotyczące problemów, co umożliwia lepszą komunikację i wgląd w przeprowadzane naprawy.

System został stworzony tak, aby był łatwy w obsłudze, jednocześnie oferując szerokie możliwości przedłużenia użytkowania urządzeń mobilnych. Dzięki przystępnym cenom i profesjonalnej obsłudze, aplikacja ma na celu przyciągnięcie dużej liczby klientów oraz przyczynienie się do zmniejszenia ilości elektrośmieci.

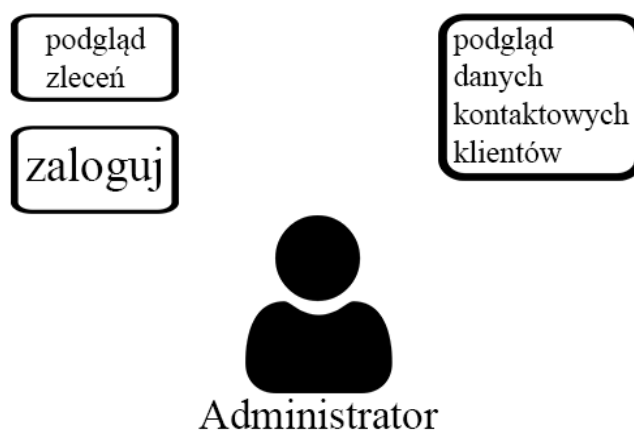
Założenia i wymagania

Po przeglądzie rozwiązań dostępnych na rynku, można określić cechy rozwiązania, które wpasuje się w lukę branżową dotyczącą serwisów urządzeń mobilnych. Głównym atutem ma być szerszy zakres usług dotyczących ingerencji w oprogramowanie urządzeń z systemem Android, np. instalacja na życzenie m.in. systemów bezpieczniejszych bez usług zbierających dane o użytkowniku.

Wymagania funkcjonalne

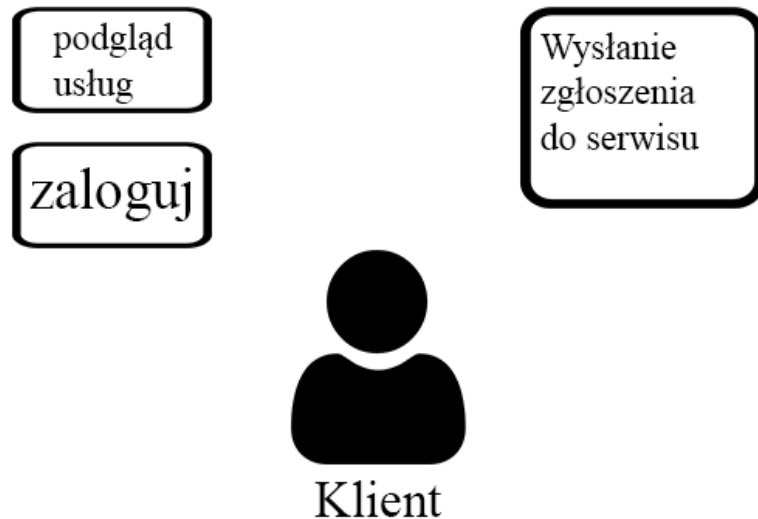
Zobrazowano funkcje, które będą zaimplementowane w aplikacji. Z funkcji tych będzie korzystać administrator systemu oraz użytkownik zewnętrzny (klient).

Administrator jest użytkownikiem, który ma dostęp do obsługi oraz przyjmowania zlecenia serwisowego. Także posiada dostęp do danych kontaktowych klientów. W tym miejscu znajduje się diagram, który prezentuje zestaw funkcji dostępnych dla osoby pełniącej rolę Administratora.



Rysunek. Diagram funkcji Administratora.

Użytkownik korzystający z ograniczonej wersji systemu, nazywany Klientem, ma dostęp tylko do wybranych funkcji i opcji. W ramach swojego konta, Klient może złożyć zgłoszenie serwisowe oraz skorzystać z oferowanych przez system usług. W tym celu może skorzystać z przedstawionych na diagramie przypadków użycia, które specjalnie dla niego zostały przygotowane i umieszczone w Rysunku. W ten sposób, Klient ma możliwość korzystania z funkcji, które są dla niego dostępne, aby jak najlepiej wykorzystać potencjał danego systemu.



Rysunek. Diagram funkcji użycia dla użytkownika typu Klient.

Wymagania niefunkcjonalne

Aby zapewnić, że aplikacja internetowa będzie działała w sposób optymalny i spełni oczekiwania użytkowników, niezbędne było zdefiniowanie wymagań niefunkcjonalnych, które określają cechy i właściwości systemu, takie jak np. wydajność, bezpieczeństwo, łatwość obsługi czy odporność na awarie.

Poniżej przedstawiono listę wymagań niefunkcjonalnych, które powinny być spełnione przez aplikację tworzoną w ramach niniejszej pracy.

Wydajność

Aplikacja powinna działać szybko i płynnie, nawet przy dużej liczbie użytkowników. Czas ładowania stron nie powinien przekraczać kilku sekund.

Dostępność

Aplikacja powinna być dostępna dla użytkowników przez całą dobę, 7 dni w tygodniu. W przypadku konieczności przerwy technicznej, powinna być zapewniona odpowiednia informacja dla użytkowników.

Bezpieczeństwo

Aplikacja powinna być zabezpieczona przed nieautoryzowanym dostępem, atakami złośliwego oprogramowania i kradzieżą danych użytkowników. Powinny być stosowane najnowsze standardy bezpieczeństwa.

Skalowalność

Aplikacja powinna umożliwiać łatwe skalowanie, aby obsłużyć większą liczbę użytkowników w miarę rozwoju biznesu.

Łatwość użytkowania

Aplikacja powinna być prosta i intuicyjna w obsłudze, aby użytkownicy mogli łatwo korzystać z jej funkcjonalności.

Kompatybilność

Aplikacja powinna działać poprawnie na różnych przeglądarkach internetowych i urządzeniach mobilnych, zgodnie z najnowszymi standardami webowymi.

Wsparcie techniczne

Powinno być zapewnione odpowiednie wsparcie techniczne dla użytkowników w przypadku problemów z aplikacją.

Właściwości językowe

Aplikacja powinna umożliwiać obsługę różnych języków i kultur, w zależności od potrzeb użytkowników.

Odporność na awarie

Aplikacja powinna być zbudowana w taki sposób, aby była odporna na awarie i błędy, aby minimalizować wpływ na użytkowników

Wyzwania związane z ruchem

Aplikacja powinna być w stanie obsłużyć przynajmniej 20 użytkowników jednocześnie, zachowując pełną wydajność i stabilność.

Scenariusze przypadków użycia

Tabela. Wyświetlanie danych klienta.

	Wyświetlanie danych klienta	
Priorytet	Wysoki	
	Aktorzy:	Administrator
	Warunek:	Dane klienta znajdują się w bazie danych
Scenariusz:	<p>Aby uzyskać dostęp do zaawansowanych funkcji, użytkownik musi kliknąć zakładkę "zaloguj".</p> <p>Następnie, użytkownik wprowadza swoje dane logowania do konta Administratora.</p> <p>Aplikacja dokonuje weryfikacji poprawności podanych informacji.</p> <p>Po poprawnym zalogowaniu, aplikacja wyświetla listę zakładek dostępnych dla użytkownika z uprawnieniami Administratora.</p> <p>Administrator wybiera zakładkę "zgłoszenia klientów".</p> <p>Aplikacja przenosi Administratora na stronę ze zgłoszeniami.</p> <p>Na stronie ze zgłoszeniami, wyświetlane są szczegóły dotyczące zgłoszeń klientów, wraz z danymi kontaktowymi klientów.</p>	
Scenariusz wyjątku:	<p>Użytkownik podaje nieprawidłowe dane logowania do konta Administratora. Aplikacja wyświetla tylko te elementy, które są dostępne dla zwykłego konta użytkownika.</p> <p>Użytkownik wprowadza dane, których nie ma w bazie użytkowników. Aplikacja wyświetla stosowny komunikat informujący o braku takiego konta.</p>	

Tabela. Wyświetlanie zgłoszeń

	Wyświetlanie zgłoszeń
Priorytet	Wysoki

	Aktorzy: Warunek:	Administrator Zgłoszenie znajduje się w bazie danych
Scenariusz:	<p>Aby uzyskać dostęp do panelu Administratora, użytkownik musi kliknąć zakładkę "Zaloguj".</p> <p>Następnie, użytkownik loguje się na konto Administratora, podając swoje dane logowania.</p> <p>Aplikacja przeprowadza weryfikację podanych danych logowania w celu upewnienia się o ich poprawności.</p> <p>Po pozytywnym zweryfikowaniu danych, aplikacja wyświetla zakładki i funkcjonalności dostępne dla konta Administratora.</p> <p>Administrator klikając w zakładkę "Zgłoszenia klientów" przenosi się na stronę z listą zgłoszeń.</p> <p>Na stronie ze zgłoszeniami, aplikacja wyświetla zgłoszenia klientów, które są dostępne do wglądu dla Administratora.</p> <p>Administrator ma wgląd w zgłoszenia użytkowników oraz ich szczegóły i może podjąć odpowiednie działania w celu ich rozwiązania.</p>	
Scenariusz wyjątku:	<p>Użytkownik podał dane nie należące do konta Administrator. Aplikacja wyświetli elementy dozwolone dla zwykłego konta użytkownika.</p> <p>Użytkownik podał dane, których nie ma w bazie użytkowników. Aplikacja wyświetla informacje o braku takiego konta.</p>	

Tabela. Wylogowanie.

Priorytet	Średni	
	Aktorzy: Warunek wstępny:	Użytkownik Administrator Użytkownik jest zalogowany do systemu
Scenariusz:	<p>Użytkownik klika wybiera przycisk „Wyloguj”.</p> <p>Użytkownik zostaje wylogowany przez aplikację.</p>	

Scenariusz wyjątku:	Brak scenariusza wyjątku w tym przypadku.
--------------------------------	---

Tabela. Wyświetlanie danych kontaktowych serwisu.

	Wyświetlanie danych kontaktowych serwisu	
Priorytet	Wysoki	
	Aktorzy:	Użytkownik
	Warunek:	Brak warunku w tym przypadku
Scenariusz:	Użytkownik klika w zakładkę dane kontaktowe. Aplikacja wyświetla zakładkę dane kontaktowe. Użytkownik uzyskuje dostęp do danych kontaktowych serwisu	
Scenariusz wyjątku:	Brak scenariusza w tym przypadku.	

Tabela. Dodawanie nowego zgłoszenia serwisowego przez klienta.

	Dodawanie nowego zgłoszenia serwisowego przez klienta	
Priorytet	Wysoki	
	Aktorzy:	Użytkownik
	Warunek:	Klient jest zalogowany w systemie

Scenariusz:	<p>Klient klika w zakładkę "Zgłoś usługę".</p> <p>Aplikacja wyświetla formularz do wypełnienia informacji o zgłoszeniu.</p> <p>Klient wprowadza opis problemu oraz informacje o swoim urządzeniu.</p> <p>Klient wysyła zgłoszenie.</p> <p>Aplikacja dodaje nowe zgłoszenie do bazy danych.</p>
Scenariusz wyjątku:	<p>Klient podał nieprawidłowe dane w formularzu. Aplikacja wyświetla informacje o konieczności poprawienia danych.</p>

Tabela. Wyświetlanie usług serwisu.

	Wyświetlanie usług serwisu	
Priorytet	Wysoki	
	Aktorzy: Warunek:	Administrator Użytkownik Brak warunku w tym przypadku
Scenariusz:	<p>Użytkownik klika w zakładkę "Usługi".</p> <p>Aplikacja wyświetla podstronę z usługami serwisu sprzętu mobilnego.</p>	
Scenariusz wyjątku:	<p>Klient podał nieprawidłowe dane w formularzu. Aplikacja wyświetla informacje o konieczności poprawienia danych.</p>	

Projekt - aplikacji

Architektura systemu

Projekt architektury systemu wspierającego pracę serwisu urządzeń mobilnych zrealizowano w oparciu o kilka kluczowych elementów.

Pierwszym elementem jest zaprojektowanie odpowiedniej struktury aplikacji. W tym celu wykorzystano strukturę katalogową, która umożliwia efektywne zarządzanie kodem. Struktura ta uwzględnia komponenty, które reprezentują różne części interfejsu użytkownika, a także konfigurację Firebase.

Kolejnym krokiem było skonfigurowanie Firebase, tak aby umożliwić aplikacji komunikację z bazą danych oraz autentykację użytkowników. Należało również zapewnić odpowiednie uprawnienia dla użytkowników, aby uniknąć niepożądanych działań w aplikacji.

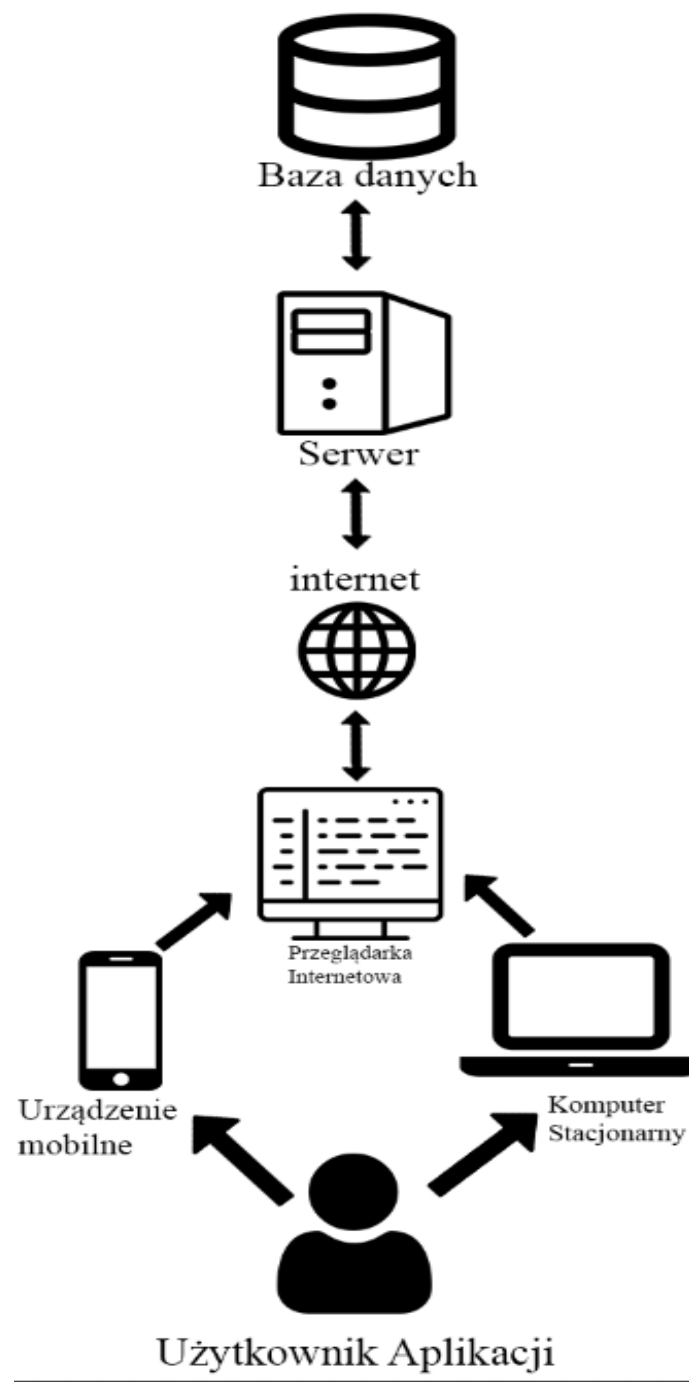
Następnym krokiem było zaprojektowanie komponentów w oparciu o bibliotekę React. W tym celu wykorzystano komponenty funkcyjne, umożliwiające tworzenie funkcjonalnych i łatwych w utrzymaniu komponentów aplikacji. Komponenty te powinny być zaprojektowane zgodnie z zasadą reużywalności, aby umożliwić łatwe korzystanie z nich w różnych częściach aplikacji.

Ważnym elementem projektu architektury systemu wspierającego pracę serwisu urządzeń mobilnych było również odpowiednie zarządzanie stanem aplikacji. W tym celu wykorzystano bibliotekę Redux, która umożliwia łatwe zarządzanie stanem aplikacji. Dzięki zastosowaniu Reduxa można uniknąć problemów związanych z przekazywaniem danych między komponentami i zapewnić spójny stan aplikacji.

Ostatnim etapem projektu architektury systemu opartego na React i Firebase było testowanie aplikacji. Należało przetestować aplikację pod kątem jej funkcjonalności, wydajności oraz bezpieczeństwa. Testowanie powinno być przeprowadzone na różnych urządzeniach i w różnych przeglądarkach, aby upewnić się, że aplikacja działa poprawnie na każdym z nich.

Podsumowując, na poniższym rysunku przedstawiono projekt architektury systemu wspierającego pracę serwisu urządzeń mobilnych – serwisu telefonów opartego na technologiach React i Firebase, wymagającego zaprojektowania odpowiedniej struktury

aplikacji, skonfigurowania Firebase, zaprojektowania komponentów, zarządzania stanem aplikacji oraz przeprowadzenia testów.

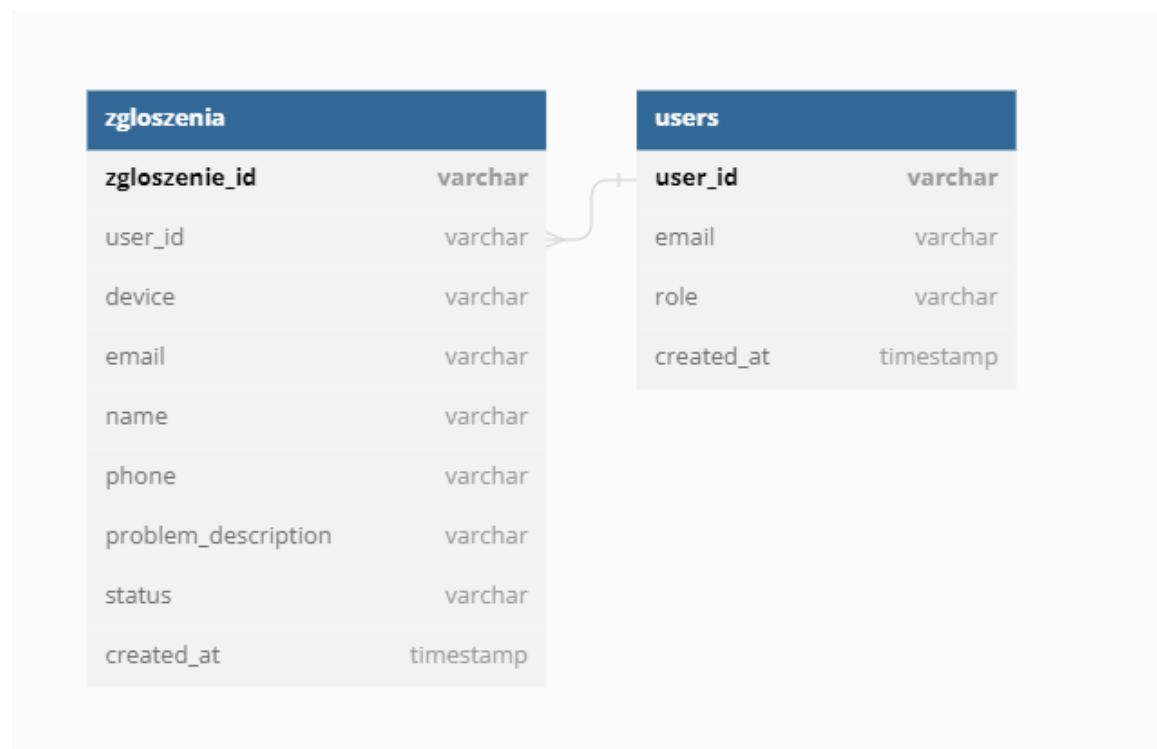


Rysunek. Model architektury systemu.

Projekt bazy danych

W celu zapewnienia prawidłowego i efektywnego działania aplikacji zrealizowano projekt bazy danych w Firebase. Baza danych to kolekcja powiązanych ze sobą danych, uporządkowanych w sposób umożliwiający szybkie i łatwe wyszukiwanie oraz przetwarzanie informacji. W przypadku projektowania bazy danych, kluczowym elementem jest jej struktura, która powinna być dobrze zaprojektowana, aby umożliwić efektywne wyszukiwanie i przetwarzanie danych. Zaproponowana struktura bazy danych opiera się na kluczowych kolekcjach i dokumentach, obejmujących użytkowników, administratorów, dane klientów oraz zgłoszenia.

Założono, że każde zgłoszenie ma unikalny identyfikator `zgloszenie_id`, który jest używany jako klucz główny tabeli `zgloszenia`. Tabela `zgloszenia` zawiera informacje o każdym zgłoszeniu, takie jak `device`, `email`, `name`, `phone`, `problem_description`, `status` i `created_at`, a także `user_id`, który jest kluczem obcym w tabeli `users`. Tabela `users` zawiera informacje o użytkownikach, w tym `user_id`, `email`, `role` i `created_at`.



Rysunek. Projekt Bazy danych.

Powyższy projekt w Firebase umożliwi efektywne zarządzanie użytkownikami, administratorami, danymi klientów oraz zgłoszeniami. Baza ta umożliwi również łatwe

wyszukiwanie i przetwarzanie danych, a także umożliwi automatyczne generowanie raportów i statystyk związanych z działalnością firmy.

Logika aplikacji

Użytkownik typu Klient będzie miał możliwość zarejestrowania się, zalogowania oraz przeglądania dostępnych usług. Będzie mógł również złożyć zamówienie, wprowadzając swoje dane osobowe, informacje dotyczące urządzenia oraz opis problemu. Wszystkie te dane są przechowywane w bazie danych Firebase, stanowiącej integralną częścią aplikacji.

Administrator powinien mieć dostęp do złożonych zgłoszeń oraz możliwość zmiany ich statusu. Aplikacja zakłada korzystanie z różnych funkcji Firebase, takich jak autoryzacja użytkowników, wysyłanie powiadomień oraz przetwarzanie danych, aby zapewnić płynne działanie aplikacji.

Interfejs graficzny aplikacji będzie oparty na React, który powinien być wykorzystywany wraz z innymi bibliotekami oraz narzędziami po to, aby stworzyć intuicyjny i przyjazny użytkownikowi interfejs. Dzięki połączeniu internetowemu aplikacja może komunikować się z bazą danych Firebase oraz innymi usługami lub API, jeśli są one potrzebne.

Projekt interfejsu użytkownika

Interfejs użytkownika jest jednym z najważniejszych elementów systemu, ponieważ zapewnia interakcję z użytkownikiem. W projekcie systemu wspierającego pracę serwisu urządzeń mobilnych interfejs użytkownika zaprojektowano w sposób, który umożliwia łatwe korzystanie z usług oraz przeprowadzenie procesu logowania i dodawania zgłoszeń. W projekcie uwzględniono funkcjonalności dla klientów oraz dla administratorów.

Strona główna

Strona główna serwisu sprzętów mobilnych składa się z logo firmy oraz przycisków nawigacyjnych umożliwiających szybki dostęp do sekcji takich jak usługi, kontakt i logowanie. Użytkownicy będą mogli zapoznać się z informacją o firmie oraz jej usługach oraz

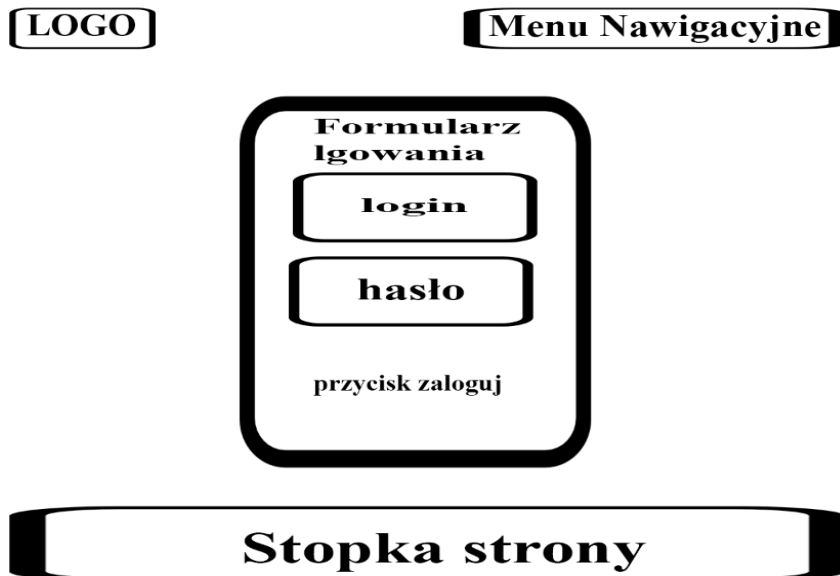
skontaktować się z nią za pomocą sekcji kontaktowej. Na dole strony znajdują się najważniejsze cechy wyróżniające firmę na rynku oraz informacje kontaktowe w stopce.



Rysunek. Projekt Strony głównej.

Logowanie

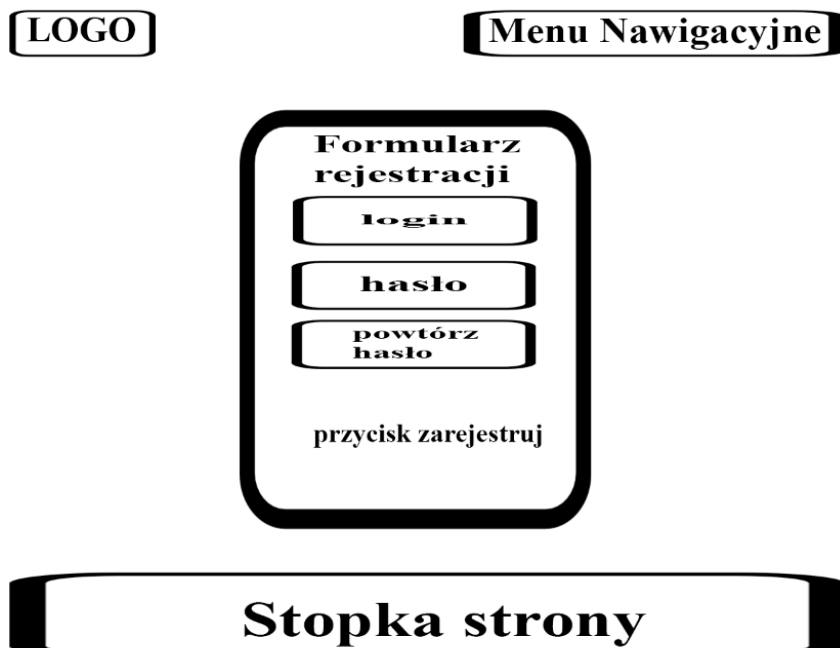
Strona logowania składa się z formularza umożliwiającego użytkownikom wprowadzenie swojego adresu e-mail oraz hasła. Dane te będą następnie wysyłane do serwera Firebase w celu weryfikacji ich poprawności. Po poprawnym zalogowaniu, użytkownik zostanie przekierowany na stronę przeznaczoną dla zalogowanych użytkowników. W przypadku niepoprawnych danych logowania, na stronie pojawi się komunikat błędu "Nieprawidłowy email lub hasło". Dodatkowo, na stronie zostanie umieszczony przycisk, umożliwiający użytkownikom przejście do strony rejestracji.



Rysunek. Projekt formularza logowania.

Rejestracja

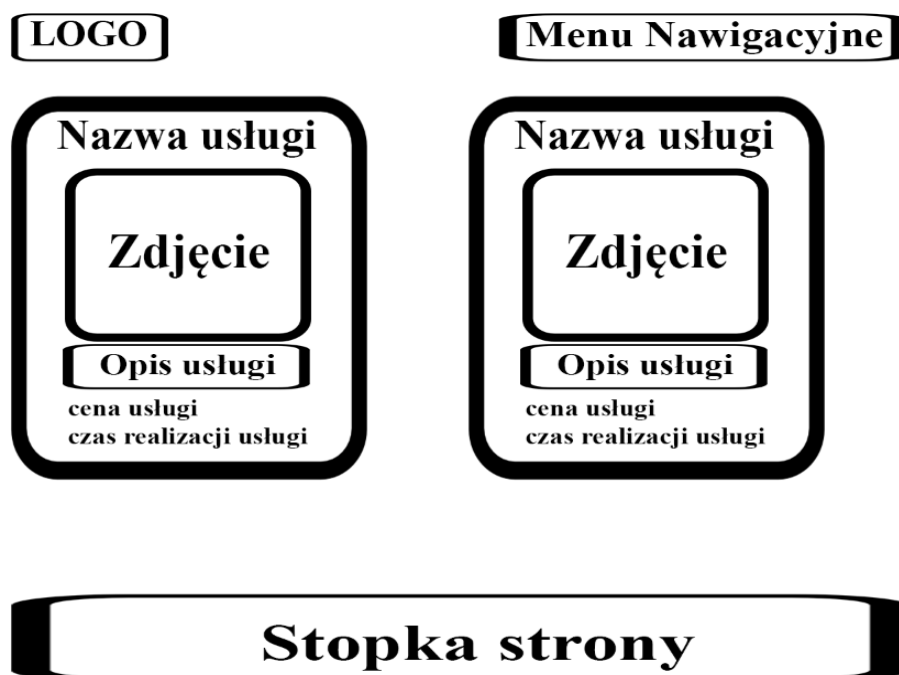
W ramach projektu stworzono komponent rejestracji umożliwiający wprowadzenie adresu e-mail i hasła, a następnie przesłanie ich do serwera Firebase w celu weryfikacji. Po poprawnej rejestracji, użytkownik zostanie przekierowany na stronę logowania.



Rysunek. Projekt formularza rejestracji.

Usługi

Strona usług zapewnia łatwy dostęp do oferty usług, prezentując je w sposób przyciągający wzrok. Na stronie będą wyświetlane osobne bloki z nazwą, opisem, ceną i czasem realizacji każdej usługi. Pod każdym blokiem znajdzie się przycisk "Zadzwoń teraz", który przeniesie użytkownika do strony kontaktowej. Dzięki temu użytkownicy będą mogli łatwo zapoznać się z ofertą, poznać ceny i wybrać najlepszą opcję, a także uzyskać szybką odpowiedź na pytania i zapytania dotyczące oferowanych usług.



Rysunek Projekt strony usług.

Kontakt

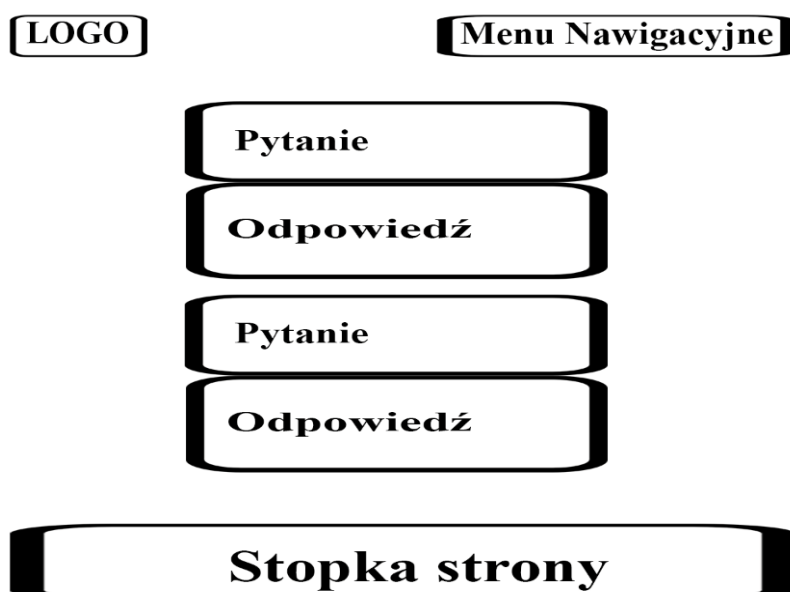
Strona kontaktowa serwisu zawiera informacje kontaktowe, takie jak numer telefonu, adres e-mail oraz adres siedziby. Interaktywna mapa ułatwi odnalezienie siedziby serwisu. Strona będzie prosta i przejrzysta, co pozwoli użytkownikom szybko znaleźć potrzebne informacje i skontaktować się z serwisem w razie potrzeby.



Rysunek. Projekt strony kontaktowej.

Często zadawane pytania (FAQ)

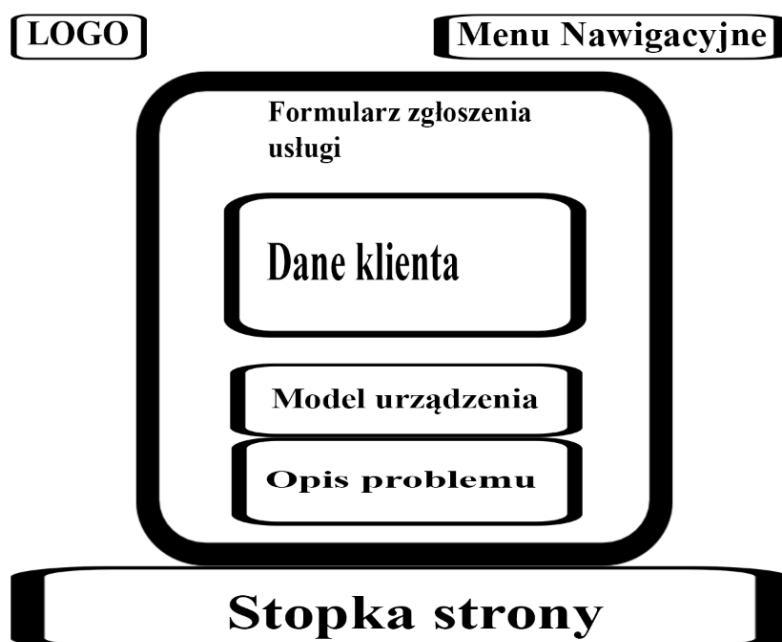
Strona "Często zadawane pytania" zawiera odpowiedzi na typowe pytania, jakie klienci mogą mieć przed skorzystaniem z usług. Użytkownicy będą mogli znaleźć tu informacje dotyczące takich kwestii jak czas naprawy urządzenia, akceptowane formy płatności, gwarancja na wykonane naprawy, możliwość wystawienia faktury za naprawę oraz wygodny sposób składania zamówień. Strona ta będzie zapewniała użytkownikom łatwy dostęp do potrzebnych informacji przed skorzystaniem z usług serwisu mobilnego.



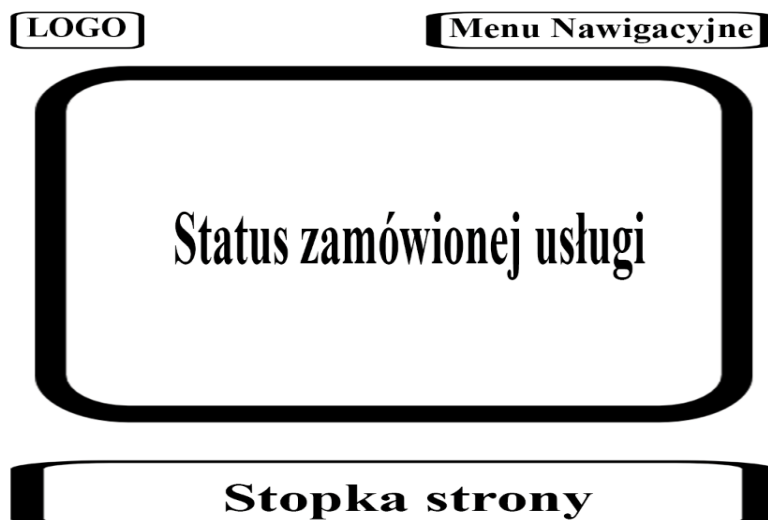
Rysunek. Projekt strony z FAQ.

Widok użytkownika (klient)

Na stronie będzie dostępny formularz zgłaszania usterki, który umożliwi klientom łatwe przesłanie informacji o problemach z ich urządzeniami. Klienci będą mogli wpisać swoje dane kontaktowe oraz opis usterki, co pozwoli na szybką diagnozę problemu i efektywną naprawę. Formularz będzie zintegrowany z bazą danych Firebase, co umożliwi łatwe zarządzanie zgłoszeniami.



Rysunek. Projekt formularza zgłoszeń.

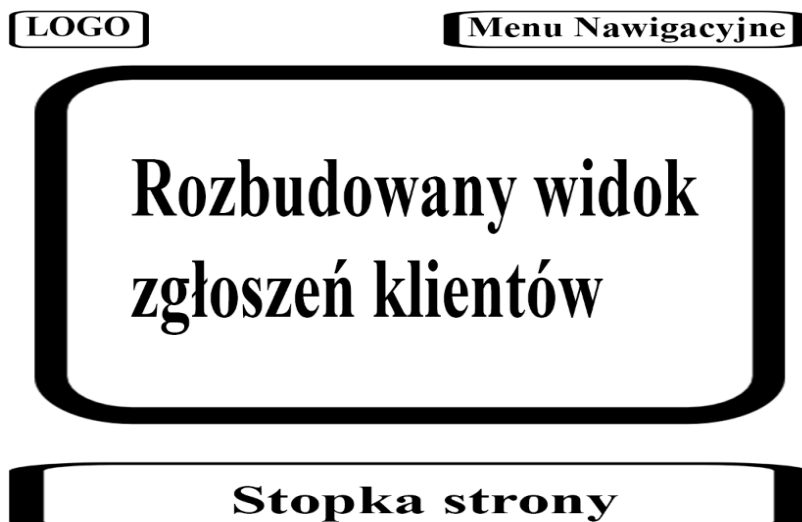


Rysunek. Projekt podglądu statusu usługi.

Użytkownicy będą mieli dostęp jedynie do swoich zgłoszeń co zapewnia prywatność i bezpieczeństwo danych. Dzięki tym narzędziom klienci serwisu mobilnego powinni łatwo zarządzać swoimi zgłoszeniami i śledzić postęp prac.

Widok administratora

Widok zgłoszeń będzie tylko dla autoryzowanych administratorów, którzy będą mieć dostęp do szczegółowych informacji na temat zgłoszeń klientów, w tym danych kontaktowych, opisów problemów oraz informacji o urządzeniach. Administratorzy będą mieli możliwość aktualizacji stanu zgłoszeń, co pozwoli na skuteczne śledzenie postępów prac i informowanie klientów o postępach. Dzięki temu narzędziu zarządzania zgłoszeniami serwisowymi, pracownicy będą w stanie efektywnie zarządzać swoim czasem i poprawić jakość obsługi, co przyczyni się do zwiększenia satysfakcji klientów. Klienci będą mieć również dostęp do narzędzia śledzenia statusu usługi, które pozwoli na monitorowanie postępu prac związanych z naprawą ich urządzeń.



Rysunek. Projekt widoku zgłoszeń dla administratorów.

Implementacja

Wykorzystane technologie i narzędzia

Stworzenie systemu wspierającego pracę serwisu urządzeń mobilnych wymagało użycia różnych narzędzi i technologii. Poniżej przedstawiono wykorzystanie React, Firebase, JavaScript, CSS, Visual Studio Code i GIMP do stworzenia takiego systemu.

React

React to biblioteka JavaScript służąca do tworzenia interfejsów użytkownika. Może być wykorzystana do tworzenia interaktywnych i dynamicznych aplikacji internetowych. W naszym przypadku React został wykorzystany do stworzenia interfejsu użytkownika.

Firebase

Firebase to platforma od Google, która dostarcza usługi takie jak autoryzacja użytkowników, baza danych w czasie rzeczywistym, hosting aplikacji i wiele innych. Firebase może być użyta do przechowywania informacji o użytkownikach, zgłoszeniach, produktach i wielu innych informacji potrzebnych w systemie.

JavaScript

JavaScript to język programowania przeznaczony do tworzenia interaktywnych aplikacji internetowych.

CSS

CSS (ang. Cascading Style Sheets) jest językiem stylów wykorzystywanym do nadawania wyglądu stronom internetowym. Jest to kluczowy element umożliwiający tworzenie atrakcyjnego interfejsu użytkownika.

Visual Studio Code

Visual Studio Code to powszechnie stosowany i bezpłatny edytor kodu, który umożliwia programowanie aplikacji internetowych w różnych językach, w tym w React i JavaScript. Jest to praktyczne narzędzie, które umożliwia tworzenie i debugowanie kodu.

GIMP

GIMP (GNU Image Manipulation Program) to darmowy program do edycji grafiki rastrowej. Może być wykorzystany do tworzenia grafik, logotypów, ikon itp. potrzebnych w systemie.

Wykorzystanie powyższych narzędzi umożliwiło stworzenie systemu wspierającego pracę serwisu urządzeń mobilnych.

Implementacja bazy danych

Kolekcja "Użytkownicy"

Kolekcja "Użytkownicy" zawiera informacje o użytkownikach aplikacji. Każdy dokument w kolekcji reprezentuje jednego użytkownika i zawiera pola takie jak "email", "phone number", "user UID", "created" i "signed in".

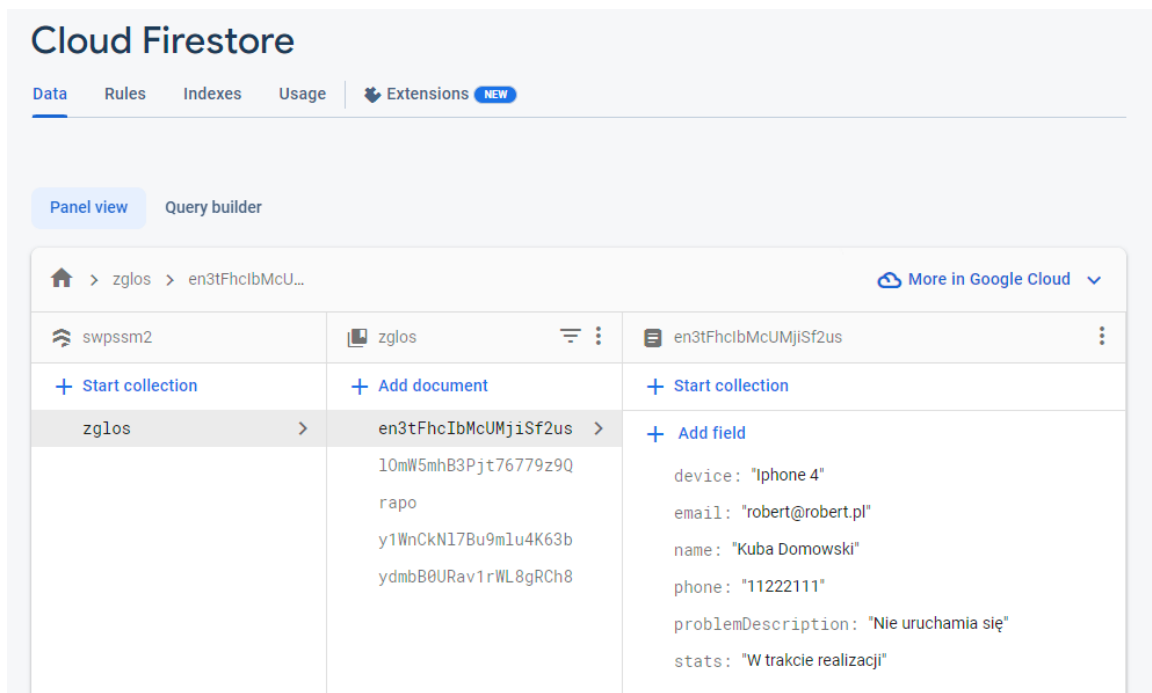
Dokument użytkownika o adresie e-mail "admin@admin.com" jest uznawany za administratora strony, co zostało zdefiniowane w kodzie aplikacji. W kolekcji można wyszukiwać użytkowników po adresie e-mail, numerze telefonu lub UID. Kolekcja umożliwia również wykonywanie różnych akcji na użytkownikach, takich jak blokowanie, usuwanie lub edytowanie informacji. W zakładce "Authentication" można skonfigurować metody logowania dla użytkowników oraz dostosować szablony stron logowania. Kolekcja oferuje również wiele funkcji, takich jak ustawienia i rozszerzenia, które pozwalają dostosować zachowanie kolekcji do potrzeb aplikacji.

Authentication				
Users	Sign-in method	Templates	Usage	Settings
Extensions NEW				
<div> <div> <div> <div> <div></div> <div>Search by email address, phone number, or user UID</div> </div> <div> <div>Add user</div> <div>↺</div> <div>⋮</div> </div> </div> </div> </div>				
Identifier	Providers	Created ↓	Signed In	User UID
adek@gmail.com	✉	Apr 4, 2023	Apr 4, 2023	sZm2mRlqa3MreeNqSnfyXcVRHrh2
robert@robert.pl	✉	Apr 4, 2023	Apr 9, 2023	aJZTppYHAhbelWrq6iuQ9gOIE5P2
admin@admin.com	✉	Apr 4, 2023	Apr 9, 2023	8DbZDQVz9AQRLLb3i1pAX2FFm5...
test@test.pl	✉	Apr 4, 2023	Apr 4, 2023	XKPcdaHpFrT568CT6AFOQzEI3ok1
<div> <div>Rows per page:</div> <div>50</div> <div>1 – 4 of 4</div> <div> <div>⏪</div> <div>⏩</div> </div> </div>				

Rysunek. Kolekcja "Użytkownicy".

Kolekcja "Zgłoszenia"

Kolekcja "zgłoszenia" (Zgłoszenia użytkowników.) w bazie danych Firebase służy do przechowywania informacji o zgłoszeniach użytkowników. W tej kolekcji każde zgłoszenie jest reprezentowane jako dokument, a każdy dokument zawiera informacje o problemie zgłoszonym przez użytkownika, takie jak opis problemu, model urządzenia, imię i nazwisko użytkownika, adres e-mail i numer telefonu. Ponadto, każdy dokument zawiera informacje o statusie zgłoszenia, takie jak czy jest w trakcie realizacji, czy zostało zakończone, a także dodatkowe informacje, takie jak identyfikator zgłoszenia i czas jego utworzenia. W kolekcji "zgłoszenia" można również przechowywać informacje o pracownikach zajmujących się zgłoszeniami oraz o firmie lub serwisie, który oferuje usługi naprawy.



Rysunek. Zgłoszenia użytkowników.

Implementacja logiki działania

Poniżej opisano szczegółowo kilka kluczowych komponentów aplikacji webowej, w tym konfigurację połączenia z bazą danych, komponent logowania, nawigacji oraz zamawiania usługi.

Konfiguracja połączenia z bazą danych

Poniżej zamieszczono listing, na którym przedstawiono parametry dotyczące połączenia pomiędzy aplikacją, a bazą danych.


```

SWPSSM2 > src >  firebasejs > ...
1  import firebase from "firebase/app";
2  import "firebase/auth";
3  import "firebase/firestore";
4
5
6  import { initializeApp } from "firebase/app";
7  import { getAnalytics } from "firebase/analytics";
8
9  const firebaseConfig = {
10     apiKey: "AIzaSyDBaQNblsikvNXFNocPWe18df4zPGIrdnA",
11     authDomain: "swpssm2.firebaseio.com",
12     projectId: "swpssm2",
13     storageBucket: "swpssm2.appspot.com",
14     messagingSenderId: "405521605660",
15     appId: "1:405521605660:web:9a46fed1961ac9507aebd9",
16     measurementId: "G-P1M7J04B00"
17 };
18
19 firebase.initializeApp(firebaseConfig)
20
21 // Initialize Firebase
22 const app = initializeApp(firebaseConfig);
23 const analytics = getAnalytics(app);
24
25 export default firebase;
26

```

Rysunek. Konfiguracja połączenia z bazą danych. – plik firebase.js

Aby ułatwić dostęp do poszczególnych dokumentów znajdujących się w kolekcji i zminimalizować ilość powtarzającego się kodu utworzono funkcję, której zadaniem jest pobranie wszystkich dokumentów z danej kolekcji i zwrócenie ich jako obiektu będącego tablicą. Dzięki temu rozwiązaniu powstanie prostszy dostęp do danych znajdujących się w kolekcji, a także mniej kodu do napisania, co przyspiesza proces tworzenia aplikacji.

Komponent logowania

Komponent Login to klasa funkcyjna napisana w języku JavaScript przy użyciu biblioteki React . Komponent ten odpowiada za logowanie użytkownika w aplikacji webowej.

Wewnątrz komponentu znajduje się stała firebaseConfig, która zawiera informacje potrzebne do konfiguracji połączenia z serwisem Firebase. Następnie, używając tej konfiguracji, inicjalizowana jest aplikacja Firebase za pomocą funkcji initializeApp.

SWPSSM2 > src > Components > Login.jsx > Login

```
1 import React, { useState } from 'react';
2 import '../CSS/Login.css';
3 import { Link, Routes, Route, useNavigate } from 'react-router-dom';
4 import { initializeApp } from 'firebase/app';
5 import { getAuth, signInWithEmailAndPassword } from 'firebase/auth';
6 import Signup from './Signup';
7
8 > const firebaseConfig = { ...
17 };
18
19 const app = initializeApp(firebaseConfig);
20
21 const Login = () => {
22   const [email, setEmail] = useState('');
23   const [password, setPassword] = useState('');
24   const [error, setError] = useState('');
25
26   const handleLogin = (e) => {
27     e.preventDefault();
28     const auth = getAuth();
29     signInWithEmailAndPassword(auth, email, password)
30       .then((userCredential) => {
31         // Logowanie udane
32         console.log('Logowanie udane!');
33         setEmail('');
34         setPassword('');
35         setError('');
36       })
37       .catch((error) => {
38         // Logowanie nieudane
39         setError('Nieprawidłowy email lub hasło.');
```

```
40   });
41 };
42
43 return (
44   <div>
45     <Routes>
46       <Route path="/" element={
47         <div className="login">
48           <form onSubmit={handleLogin}>
49             <h1>Zaloguj się</h1>
50             {error && <div className="error">{error}</div>}
51             <div className="form-group">
52               <label>Email:</label>
53               <input
54                 type="email"
55                 value={email}
56                 onChange={(e) => setEmail(e.target.value)}
57               />
58             </div>
59             <div className="form-group">
60               <label>Hasło:</label>
61               <input
62                 type="password"
63                 value={password}
64                 onChange={(e) => setPassword(e.target.value)}
65               />
66             </div>
67             <button type="submit" className="btn">
68               Zaloguj się
69             </button>
70             <Link to="/signup">
71               <button className="button">Zarejestruj się</button>
72             </Link>
73           </form>
74         </div>
75       } />
76       <Route path="/signup" element={<Signup />} />
77     </Routes>
78   </div>
79 );
80 };
81
82 export default Login;
```

Do you mind takin

Rysunek. Komponent logowania - plik Login.jsx

W Login wykorzystywane są dwa stany: email oraz password, które są aktualizowane za pomocą funkcji `useState`. Są one używane do przechowywania wprowadzonych przez użytkownika danych logowania.

Główną funkcjonalnością komponentu jest metoda `handleLogin`, która obsługuje proces logowania użytkownika. Przechwytuje ona zdarzenie `onSubmit` z formularza logowania, tworzy instancję `auth` z autentykacji `Firebase` i wywołuje metodę `signInWithEmailAndPassword` w celu uwierzytelnienia użytkownika.

W przypadku powodzenia logowania wyświetlany jest komunikat w konsoli, a stan `email` oraz `password` zostają zresetowane do wartości początkowych. W przypadku błędu logowania, ustawiany jest stan `error`, który informuje użytkownika o nieudanej próbie logowania.

Komponent `Login` zawiera również definicje dróg nawigacyjnych za pomocą biblioteki `React Router`. Zdefiniowane są dwie ścieżki: `/` dla strony logowania oraz `/signup` dla strony rejestracji. Odpowiednie komponenty są renderowane w zależności od ścieżki, za pomocą atrybutu `element`.

Komponent nawigacji

Komponent nawigacji jest odpowiedzialny za wyświetlanie sekcji, w której znajdują się linki do innych stron w aplikacji. Składa się z marki, menu nawigacyjnego i przycisku wylogowania.

W sekcji marki znajduje się link prowadzący na stronę główną aplikacji. Menu nawigacyjne zawiera listę linków do innych stron, a przycisk wylogowania jest wyświetlany tylko dla zalogowanych użytkowników.

Komponent ten używa kilku haseł z biblioteki `React`, takich jak `useState`, `useEffect`, oraz `useNavigate` z biblioteki `react-router-dom`, który jest wykorzystywany do nawigowania użytkowników do innych stron w aplikacji.

Komponent również korzysta z biblioteki `Firebase`, aby umożliwić uwierzytelnienie użytkowników i wylogowanie. Zmiana stanu uwierzytelnienia jest wykrywana przez funkcję `onAuthStateChanged` z biblioteki `Firebase`.

Komponent jest dynamiczny, ponieważ listę linków do stron tworzy w oparciu o stan uwierzytelnienia użytkownika. Na przykład gdy użytkownik jest zalogowany jako administrator, pojawia się link do strony zgłoszeń.

```
1 import { Link, useNavigate } from 'react-router-dom';
2 import { useState, useEffect } from 'react';
3 import { getAuth, onAuthStateChanged, signOut } from 'firebase/auth';
4 import { initializeApp } from 'firebase/app';
5 import '../CSS/navbar.css';
6
7 > const firebaseConfig = { ...
16 };
17
18 const app = initializeApp(firebaseConfig);
19 const auth = getAuth(app);
20
21 export default function Navbar() {
22   const [user, setUser] = useState(null);
23   const navigate = useNavigate();
24
25   useEffect(() => {
26     const unsubscribe = onAuthStateChanged(auth, setUser);
27     return unsubscribe;
28   }, []);
29
30   const handleLogout = () => {
31     signOut(auth).then(() => {
32       setUser(null);
33       navigate('/');
34     });
35   };
36
37   const navLinks = [
38     { id: 'home', title: 'Strona Główna' },
39     { id: 'services', title: 'Usługi' },
40     user?.email === 'admin@admin.com' && { id: 'reports', title: 'Zgłoszenia' },
41     user && user.email !== 'admin@admin.com' && { id: 'Order', title: 'Zamów Usługę' },
42     user && user.email !== 'admin@admin.com' && { id: 'Status', title: 'Status Usługi' },
43     { id: 'contacts', title: 'Kontakt' },
44     !user && { id: 'login', title: 'Zaloguj' },
45   ].filter(Boolean);
46
47   return (
48     <nav className="navigation">
49       <Link to="/" className="brand-name">
50         <span className="brand-name__first-letter">S</span>
51         <span className="brand-name__other-letters">WPSUM</span>
52       </Link>
53       <div className="navigation-menu">
54         <ul>
55           {navLinks.map(({ id, title }) => (
56             <li key={id}>
57               <Link to={`/${id}`}>{title}</Link>
58             </li>
59           ))}
60           {user && (
61             <button className="button" onClick={handleLogout}>Wyloguj</button>
62           )}
63         </ul>
64       </div>
65     </nav>
66   );
67 }
```

Rysunek. Komponent nawigacji - plik Navbar.jsx

Komponent zamawiania usługi

Komponent zamawiania usługi napisano w języku JavaScript i służy do obsługi formularza zgłoszenia usterki. Komponent ten korzysta z Firebase, aby zapisywać zgłoszenia użytkowników w bazie danych.

```
25 const [name, setName] = useState('');
26 const [email, setEmail] = useState('');
27 const [phone, setPhone] = useState('');
28 const [device, setDevice] = useState('');
29 const [issue, setIssue] = useState('');
30
31 useEffect(() => {
32
33   onAuthStateChanged(auth, (user) => {
34     if (user) {
35       setEmail(user.email);
36     }
37   });
38 }, [auth]);
39
40 const handleSubmit = async (e) => {
41   e.preventDefault();
42   const order = { name, email, phone, device, issue };
43   await addDoc(ordersRef, order);
44   setName('');
45   setEmail('');
46   setPhone('');
47   setDevice('');
48   setIssue('');
49   dateSubmitted('');
50 };
51
52 return (
53   <div className="order">
54     <form onSubmit={handleSubmit}>
55       <h1 className="title">Zgłoś usterkę</h1>
56       <label className="label">Imię i nazwisko</label>
57       <input
58         className="input"
59         type="text"
60         placeholder="Twoje imię i nazwisko"
61         value={name}
62         onChange={(e) => setName(e.target.value)}
63         required
64       />
65       <label className="label">Adres e-mail</label>
66       <input
67         className="input"
68         type="email"
69         placeholder="Twój adres e-mail"
70         value={email}
71         onChange={(e) => setEmail(e.target.value)}
72         required
```

```

73     />
74     <label className="label">Numer telefonu</label>
75     <input
76         className="input"
77         type="tel"
78         placeholder="Tudj numer telefonu"
79         value={phone}
80         onChange={(e) => setPhone(e.target.value)}
81         required
82     />
83     <label className="label">Urządzenie</label>
84     <input
85         className="input"
86         type="tel"
87         placeholder="Twoje urządzenie"
88         value={device}
89         onChange={(e) => setDevice(e.target.value)}
90         required
91     />
92     <label className="label">Opis usterki</label>
93     <textarea
94         className="textarea"
95         placeholder="Opis usterki"
96         value={issue}
97         onChange={(e) => setIssue(e.target.value)}
98         required
99     />
100     <button className="button" type="submit">
101         Wyślij zgłoszenie
102     </button>
103     <label className="label">Prosimy o przesłanie urządzenia, <br/>na adres podany w stopce</label>
104     </form>
105 </div>
106 };
107
108
109 export default Order;

```

Rysunek. Komponent tworzenia zgłoszeń - plik Order.jsx

W sekcji importów, komponent importuje niezbędne funkcje z Firebase, takie jak `collection`, `addDoc`, `onSnapshot`, `initializeApp`, `getFirestore`, `getAuth` i `onAuthStateChanged`.

Poza sekcją importów, komponent tworzy obiekt konfiguracyjny Firebase, a następnie inicjalizuje i uzyskuje dostęp do Firestore oraz autentykacji.

W funkcji `useEffect`, komponent nasłuchuje zmian w autentykacji Firebase, aby uzyskać adres e-mail użytkownika i przypisać go do zmiennej `email`.

W funkcji `handleSubmit`, komponent zbiera dane z formularza, tworzy obiekt `order`, który zawiera nazwę, adres e-mail, numer telefonu, urządzenie i opis usterki. Następnie, używając funkcji `addDoc`, komponent zapisuje zgłoszenie w kolekcji Firebase. Po zapisaniu zgłoszenia, wszystkie pola formularza zostają wyczyszczone.

Komponent renderuje formularz z etykietami i polami formularza. Po wypełnieniu formularza, użytkownik może kliknąć przycisk `Wyślij zgłoszenie`, który uruchomi funkcję `handleSubmit`. W stopce formularza, wyświetla się etykieta informująca użytkownika o przesłaniu urządzenia na podany adres.

W skrócie, komponent ten służy do zbierania zgłoszeń użytkowników dotyczących usterki i zapisywania ich w bazie danych Firebase.

Lista eksportowanych komponentów

W kodzie znajduje się lista eksportowanych komponentów z poszczególnych plików. Każdy z tych komponentów jest odpowiedzialny za określone elementy strony:

1. Navbar - komponent renderujący nawigację strony,
2. Login - komponent zawierający formularz logowania,
3. Signup - komponent zawierający formularz rejestracji,
4. Order - komponent zawierający formularz zamówienia,
5. Contacts - komponent zawierający informacje kontaktowe,
6. Services - komponent renderujący listę usług oferowanych przez firmę,
7. Reports - komponent renderujący zgłoszenia,
8. Footer - komponent renderujący stopkę strony,
9. Hero - komponent renderujący stronę główną,
10. Status - komponent renderujący informacje o stanie zamówienia,
11. Faq - komponent renderujący często zadawane pytania,
12. Review - komponent renderujący recenzje i opinie klientów.

```
SWPSSM2 > src > Components > index.js
1  import Hero from "../Hero";
2  import Navbar from "../Navbar";
3  import Services from "../Services";
4  import Order from "../Order";
5  import Contacts from "../Contacts";
6  import Reports from "../Reports";
7  import Footer from "../Footer";
8  import Login from "../Login";
9  import Signup from "../Signup";
10 import Status from "../Status";
11 import Faq from "../Faq";
12 import Review from "../Review";
13
14 export {
15   Navbar,
16   Login,
17   Signup,
18   Order,
19   Contacts,
20   Services,
21   Reports,
22   Footer,
23   Hero,
24   Status,
25   Faq,
26   Review,
27 };
28
```

Rysunek Lista eksportowanych komponentów - plik index.js

Dzięki takiemu eksportowaniu można łatwo importować poszczególne komponenty z tego pliku w innych plikach, co ułatwia zarządzanie kodem i zapewnia jego przejrzystość.

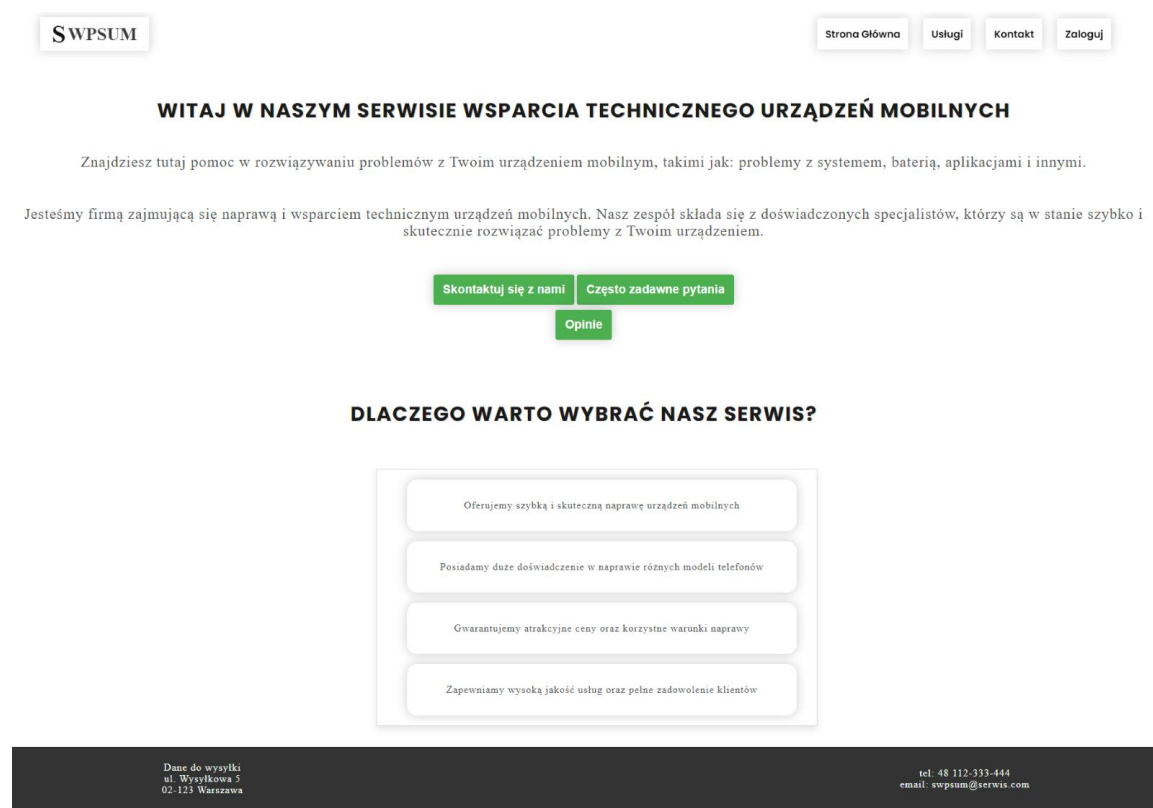
Implementacja interfejsu użytkownika

Projekt interfejsu dla systemu wspierającego pracę serwisu urządzeń mobilnych zaprojektowano w taki sposób, aby umożliwić użytkownikom łatwe korzystanie z usług oraz

przeprowadzenie procesu logowania i dodawania zgłoszeń. Uwzględniono funkcjonalności dla klientów oraz dla administratorów.

Strona główna

Strona główna składa się z nagłówka, który zawiera logo firmy oraz przyciski nawigacyjne, pozwalające na szybkie dostęp do najważniejszych sekcji strony, w tym strony głównej, sekcji usług, kontaktu oraz logowania.



Rysunek. Strona główna systemu.

W środkowej części strony użytkownicy mogą zapoznać się z informacją o firmie oraz o szerokim zakresie usług, jakie świadczą. Znajdują się tu również sekcje pozwalające na skontaktowanie się z firmą, przeczytanie często zadawanych pytań oraz zapoznanie się z opiniami klientów o serwisie.

Poniżej użytkownicy mogą znaleźć informacje o kilku najważniejszych cechach, które wyróżniają tę firmę spośród innych na rynku.

Stopka strony zawiera niezbędne informacje kontaktowe do firmy, takie jak dane do wysyłki czy numery telefonu i adresy e-mail, które pozwalają na szybki i łatwy kontakt z serwisem w przypadku jakichkolwiek pytań lub wątpliwości.

Logowanie

Strona logowania składa się z formularza, który umożliwia użytkownikom wprowadzenie swojego adresu e-mail oraz hasła. Dane te są następnie wysyłane do serwera Firebase w celu weryfikacji ich poprawności. Po poprawnym zalogowaniu, użytkownik zostaje przekierowany na stronę przeznaczoną dla zalogowanych użytkowników. W przypadku niepoprawnych danych logowania, na stronie pojawi się komunikat błędu "Nieprawidłowy email lub hasło". Dodatkowo, na stronie umieszczony jest przycisk, umożliwiający użytkownikom przejście do strony rejestracji.

SWPSUM

Strona Główna Usługi Kontakt Zaloguj

Zaloguj się

Email:

Hasło:

Zaloguj się

Zarejestruj się

Dane do wysyłki
ul. Wysyłkowa 5
02-125 Warszawa

tel. 48 112-333-444
email: swpsum@serwis.com

Rysunek. Ekran logowania do systemu.

Rejestracja

Komponent na zawiera formularz rejestracji, który pozwala użytkownikom na wprowadzenie adresu e-mail i hasła, a następnie ich wysłanie do serwera Firebase w celu weryfikacji poprawności tych danych. Jeśli dane są poprawne, użytkownik zostaje zarejestrowany i przekierowany na stronę logowania. W przeciwnym razie zostanie wyświetlony błąd z informacją o przyczynie niepowodzenia. Komponent zawiera również przycisk umożliwiający przejście do strony logowania oraz obsługę stanów (useState) w celu przechowywania wprowadzonych przez użytkownika danych i błędów.

The screenshot shows a web page for the SWPSUM system. At the top left is the SWPSUM logo. At the top right are navigation links: Strona Główna, Usługi, Kontakt, and Zaloguj. In the center is a registration form titled 'Zarejestruj się'. The form contains three input fields labeled 'Email:', 'Hasło:', and 'Potwierdź hasło:'. Below these fields is a blue button labeled 'Zarejestruj się'. At the bottom of the form, it says 'Posiadasz już konto? Zaloguj się' with a link. At the bottom of the page is a dark footer bar containing contact information: 'Dane do wysyłki ul. Wyszyńska 5 02-123 Warszawa' and 'tel: 48 112-333-444 email: swpsum@serwis.com'.

Rysunek. Ekran rejestracji do systemu.

Usługi

Strona usług to kluczowa podstrona serwisu, która zapewnia użytkownikom łatwy dostęp do oferty usług. Przejrzystość i czytelność prezentacji oferty są jej najważniejszymi cechami.

NASZE USŁUGI

Oferujemy wiele usług związanych z naprawą i wsparciem technicznym Twojego urządzenia mobilnego. Zaloguj się i zgłoś usterkę swojego urządzenia.

**Wymiana ekranu**

Jeśli Twój ekran został uszkodzony, wymienimy go na nowy i sprawimy, że Twój telefon będzie działał jak nowy.

CZAS REALIZACJI: 1-2 TYGODNIE

Cena: 49,99zł

[Zadzwoń teraz](#)

**Aktualizacja oprogramowania**

Jeśli potrzebujesz aktualizacji oprogramowania, pomożemy Ci zaktualizować system operacyjny i aplikacje na Twoim urządzeniu.

CZAS REALIZACJI: 2-3 TYGODNIE

Cena: 79,99zł

[Zadzwoń teraz](#)

**Wymiana baterii**

Jeśli Twój telefon działa zbyt krótko na jednym ładowaniu, wymienimy baterię na nową i przywrócimy pełną funkcjonalność Twojego urządzenia.

CZAS REALIZACJI: 3-4 TYGODNIE

Cena: 129,99zł

[Zadzwoń teraz](#)

**Wodoodporność**

Jeśli chcesz zabezpieczyć swój telefon przed zalaniem lub uszkodzeniem w wyniku kontaktu z wodą, możemy zabezpieczyć go przed wodą.

CZAS REALIZACJI: 1 TYDZIEŃ

Cena: 89,99zł

[Zadzwoń teraz](#)

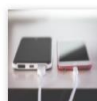
**Usuwanie wirusów i malware**

Jeśli Twój telefon jest zainfekowany wirusami lub malwarem, możemy przeskanować i usunąć złośliwe oprogramowanie.

CZAS REALIZACJI: 2-3 TYGODNIE

Cena: 39,99zł

[Zadzwoń teraz](#)

**Naprawa portu ładowania**

Jeśli Twój telefon nie ładuje się poprawnie lub trzeba przyciskać kabel ładowarki, możemy naprawić port ładowania.

CZAS REALIZACJI: 2 TYGODNIE

Cena: 99,99zł

[Zadzwoń teraz](#)

Rysunek. Ekran usług serwisu.

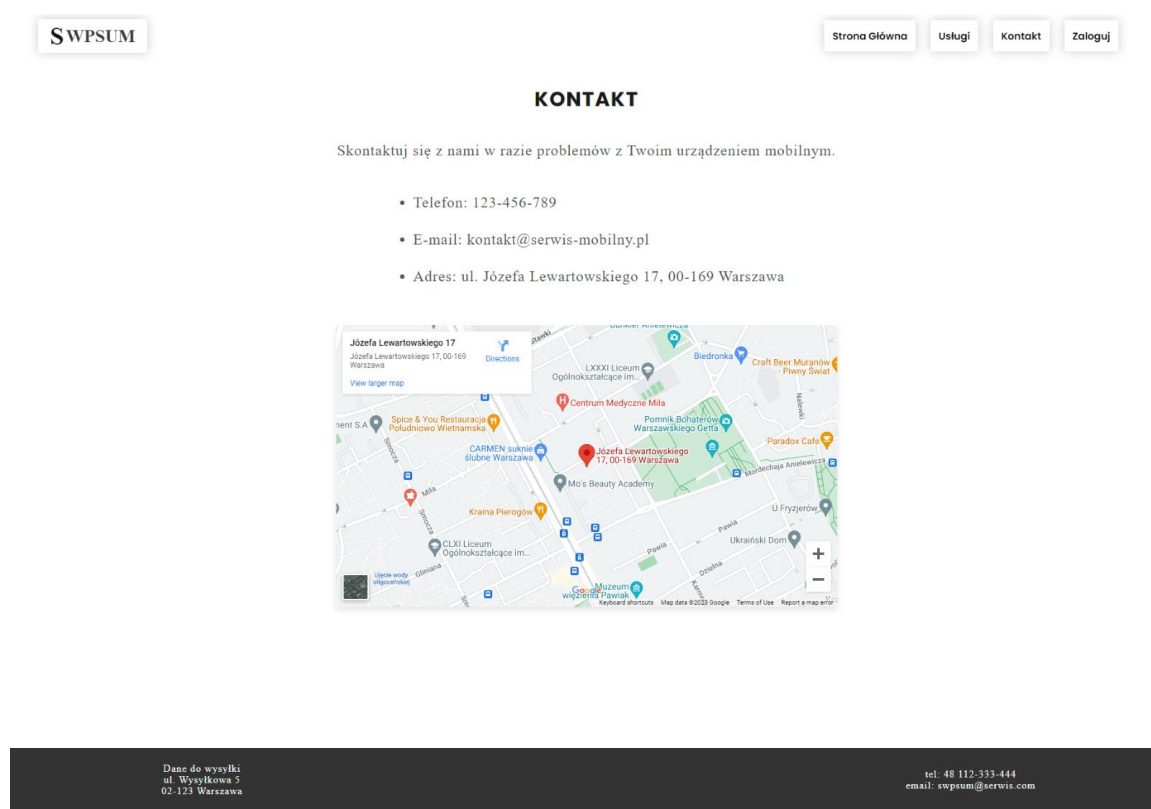
Na stronie wyświetlone są poszczególne usługi, które oferuje serwis, w atrakcyjny i przyciągający wzrok sposób. Każda usługa jest przedstawiona w osobnym bloku, który zawiera nazwę usługi oraz krótki opis. Informacja o cenie usługi i czasie realizacji jest jasno przedstawiona, co pozwala użytkownikom na dokładne zaplanowanie swoich potrzeb i wybór najlepszej dla siebie opcji.

Dodatkowo, pod każdym blokiem usługi umieszczony jest przycisk "Zadzwoń teraz", który przenosi do strony z danymi kontaktowymi z serwisem w celu uzyskania dodatkowych informacji lub zapytań..

Dzięki takiej formie prezentacji usług na stronie, użytkownicy mogą łatwo zapoznać się z ofertą serwisu, poznać ceny i wybrać opcję dla siebie. Dodatkowo, możliwość bezpośredniego kontaktu z serwisem pozwala na szybkie uzyskanie odpowiedzi na pytania i zapytania dotyczące oferowanych usług.

Kontakt

Na tej stronie wyświetlane są informacje kontaktowe serwisu, które umożliwiają użytkownikom szybki i łatwy kontakt w razie problemów z urządzeniem. Na górze strony znajduje się tytuł "Kontakt" oraz podtytuł, który zachęca użytkowników do skontaktowania się z serwisem w przypadku problemów.



Rysunek. Ekran danych kontaktowych serwisu.

Poniżej umieszczone są szczegółowe informacje kontaktowe, takie jak numer telefonu, adres e-mail oraz adres siedziby serwisu. Dodatkowo, na stronie znajduje się interaktywna

mapa, na której widać lokalizację siedziby serwisu sprzętów mobilnych, co pozwala użytkownikom łatwiej odnaleźć miejsce, w którym mogą się zgłosić z problemem. Strona jest prosta i przejrzysta, co pozwala użytkownikom szybko znaleźć potrzebne informacje i skontaktować się z serwisem w razie potrzeby.

Często zadawane pytania (FAQ)

Strona "Często zadawane pytania" (ang. Frequently Asked Questions) zawiera odpowiedzi na typowe pytania, jakie klienci mogą mieć przed skorzystaniem z usług serwisu mobilnego. Tutaj znajdują się informacje dotyczące takich kwestii jak czas naprawy urządzenia, akceptowane formy płatności, gwarancja na wykonane naprawy, możliwość wystawienia faktury za naprawę oraz wygodny sposób składania zamówień.

SWPSUM

Strona Główna

Usługi

Kontakt

Zaloguj

FAQ

Jak długo trwa naprawa sprzętu?

Czas naprawy zależy od rodzaju uszkodzenia i naszego obłożenia. Zazwyczaj staramy się wykonać naprawę jak najszybciej i nie zajmuje to więcej niż kilka dni.

Jakie formy płatności akceptujecie?

Akceptujemy płatności gotówkowe oraz przelewy bankowe. Nie przyjmujemy płatności kartą.

Czy udzielacie gwarancji na naprawy?

Tak, udzielamy gwarancji na wykonane przez nas naprawy. Gwarancja obejmuje okres 3 miesięcy od daty naprawy.

Czy jest możliwość otrzymania faktury za naprawę?

Tak, wystawiamy faktury za wykonane przez nas usługi naprawy. Wystarczy poprosić o fakturę przy składaniu zamówienia.

Czy muszę składać zamówienie online?

Nie, możesz składać zamówienie przez telefon lub osobiście w naszym serwisie. Jednak zalecamy składanie zamówień online, ponieważ jest to wygodniejsze i szybsze.

Dane do wysyłki
ul. Wysyłkowa 5
02-153 Warszawa

tel: 48 112-333-444
email: swpsum@serwis.com

Rysunek. Ekran często zadawanych pytań.

Użytkownik zalogowany (klient)

Zgłoś usterkę

Na stronie znajduje się bardzo użyteczny formularz zgłaszania usterki, który pozwala klientom serwisu na łatwe i szybkie przesłanie informacji o problemach z ich urządzeniami. W formularzu klient może wpisać swoje dane kontaktowe, takie jak imię i nazwisko, adres e-mail oraz numer telefonu, aby pracownicy serwisu mieli możliwość łatwego i szybkiego skontaktowania się z nim.

SWPSUM

Strona Główna Usługi Zamów Usługę Status Usługi Kontakt Wyloguj

ZGŁOŚ USTERKĘ

Imię i nazwisko
Twoje imię i nazwisko

Adres e-mail
robert@robert.pl

Numer telefonu
Twój numer telefonu

Urządzenie
Twoje urządzenie

Opis usterki
Opis usterki

Wyślij zgłoszenie

Prosimy o przesłanie urządzenia,
na adres podany w stopce

Dane do wysyłki
ul. Wyszyńska 5
02-123 Warszawa

tel: 48 112-333-444
email: swpsum@serwis.com

Rysunek. Ekran zgłaszania usterek w systemie.

Dodatkowo, w formularzu znajdują się pola do wpisania nazwy urządzenia oraz opisu usterki, co umożliwia pracownikom serwisu szybszą diagnozę problemu i efektywniejszą naprawę. Formularz ten jest zintegrowany z biblioteką Firebase, która umożliwia łatwe zarządzanie bazą danych oraz uwierzytelnianie użytkowników.

Dzięki temu, gdy klient wypełni formularz i kliknie przycisk "Wyślij zgłoszenie", jego zgłoszenie zostanie automatycznie zapisane w bazie danych Firebase. Taki sposób

przechowywania zgłoszeń umożliwia pracownikom serwisu łatwy i szybki dostęp do informacji o zgłoszonych usterkach oraz efektywne zarządzanie nimi.

Status usługi

Ta strona pozwala klientom na śledzenie statusu usług, które zamówili w serwisie. Dzięki temu użytkownik może na bieżąco sprawdzać, w jakim etapie znajduje się jego zgłoszenie, na przykład czy urządzenie jest już naprawiane czy też zostało dopiero odebrane.

Ważne jest to, że każdy użytkownik ma wgląd tylko do swoich zgłoszeń, co zapewnia prywatność i bezpieczeństwo danych. To proste i wygodne narzędzie pomaga klientom w śledzeniu postępu prac oraz podejmowanych decyzji co do dalszych działań związanych z ich urządzeniami.

SWPSUM		Strona Główna	Usługi	Zamów Usługę	Status Usługi	Kontakt	Wyloguj
TWOJE ZGŁOSZENIA							
IMIĘ I NAZWISKO	TELEFON	E-MAIL	URZĄDZENIE	STATUS USŁUGI			
Kuba Domowski	11222111	robert@robert.pl	Iphone 4	W trakcie realizacji			
Andrew Robertowcy	5554551111	robert@robert.pl	Huawei p20	Sprzęt odebrany od klienta			

Dane do wysyłki
ul. Wysyłkowa 5
02-123 Warszawa

tel: 48 112-333-444
email: swpsum@serwis.com

Rysunek. Ekran statusu usług.

Użytkownik zalogowany (Administrator)

Ta strona jest dostępna wyłącznie dla autoryzowanych administratorów i pozwala im na wyświetlenie szczegółowych informacji dotyczących zgłoszeń klientów, takich jak dane kontaktowe, opisy problemów i informacje o urządzeniach. W ten sposób serwisanci mogą uzyskać pełny wgląd w historię zgłoszeń i podjąć odpowiednie kroki, aby rozwiązać problem klienta.

SWPSUM	Strona Główna	Usługi	Zgłoszenia	Kontakt	Wyloguj
ZGŁOSZENIA SERWISOWE					
STATUS USŁUGI	IMIĘ I NAZWISKO ▲	TELEFON	E-MAIL	URZĄDZENIE	OPIS PROBLEMU
Sprzęt odebrany od klienta	Andrzej Robertowicz	5554551111	robert@robert.pl	Huawei p20	Nie działa telefon :(
Zakończony - sprzęt odebrany przez klienta	Jan Nowak	11-22-55	mail@mail	nexus 4	Proszę o opomoc
W trakcie realizacji	Kuba Domowski	11222111	robert@robert.pl	Iphone 4	Nie uruchamia się
Zakończony - sprzęt odebrany przez klienta	Paweł Saw	555666555	robert@ss.pl	Iphone 5	nie działa cały
Sprzęt odebrany od klienta	Piotr Kowka	8888777443	adsads@ss	adsads	Przyciski się polamały

Dane do wysyłki
ul. Wysyłkowa 5
02-123 Warszawa

tel: 48 112-333-444
email: swpsum@serwis.com

Rysunek. Ekran podglądu zgłoszeń klientów.

Administratorzy mają możliwość ustawienia aktualnego stanu każdego zgłoszenia, co umożliwia śledzenie postępu prac i informowanie klientów o postępach. Dzięki temu systemowi zarządzania zgłoszeniami serwisowymi, pracownicy mogą efektywnie zarządzać swoim czasem i zwiększyć satysfakcję klientów, poprawiając jakość obsługi.

Uruchamianie i testowanie

W celu zapewnienia doskonałej jakości utworzonego systemu, przeprowadzono liczne testy, które miały za zadanie zweryfikowanie poprawności działania interfejsu użytkownika, użyteczności aplikacji oraz komunikacji z bazą danych.

Dzięki tym testom, uzyskano szczegółowe informacje dotyczące funkcjonalności systemu oraz jego efektywności. Sprawdzono, czy interfejs użytkownika działa bez zarzutu, czy użytkownicy mogą łatwo korzystać z funkcji systemu, a także, czy dane przechowywane w bazie danych są dostępne i poprawnie wykorzystywane przez system

Testy funkcjonalne

Dzięki korzystaniu z literatury "Testowanie oprogramowania. Podręcznik dla początkujących", udało się wdrożyć odpowiednie narzędzia i techniki, które umożliwiły szczegółową weryfikację poprawności działania systemu internetowego oraz wykrycie i naprawę błędów. Wdrożone testy funkcjonalne (zestawione w tabelach) były kluczowe w procesie weryfikacji, ponieważ umożliwiły dokładną analizę funkcjonalności systemu, takich jak logowanie, rejestracja użytkowników, dodawanie zleceń serwisowych oraz interfejs użytkownika. Każdy test składał się z precyzyjnie określonej sekwencji działań oraz oczekiwanych i faktycznych rezultatów. W wyniku tych działań, udało się wcześniej wykryć potencjalne problemy i błędy w systemie, co pozwoliło na ich skuteczną naprawę przed wystąpieniem u użytkowników końcowych.

Tabela. Przeprowadzonych testów logowania

Nr. testu	Wykonane działanie	Oczekiwany rezultat	Rezultat
1	Próba zalogowania bez podania danych w formularzu logowania	Wyświetlenie komunikatu o konieczności podania danych logowania	Pozytywny
2	Próba zalogowania bez podania hasła w formularzu logowania	Wyświetlenie komunikatu o konieczności podania hasła	Pozytywny

3	Próba zalogowania bez podania loginu w formularzu logowania	Wyświetlenie komunikatu o konieczności podania loginu	Pozytywny
4	Podanie niepoprawnego formatu loginu przy logowaniu	Wyświetlenie komunikatu o konieczności podania poprawnego formatu	Pozytywny
5	Poprawne podanie danych logowania w formularzu	Autoryzacja dostępu do systemu	Pozytywny
6	Wylogowanie z systemu poprzez kliknięcie przycisku "Wyloguj"	Wylogowanie użytkownika z systemu	Pozytywny

Tabela. Przeprowadzonych testów dodawania zleceń serwisowych.

Nr. testu	Wykonane działanie	Oczekiwany rezultat	Rezultat
1	Dodanie nowego zlecenia serwisowego po wypełnieniu wymaganych pól formularza	System rejestruje nowe zlecenie serwisowe w bazie danych	Pozytywny
2	Próba dodania nowego zgłoszenia bez podania wymaganych pól formularza	System wyświetla komunikat informujący o konieczności wypełnienia wszystkich wymaganych pól formularza	Pozytywny

Tabela. Przeprowadzonych testów rejestracji użytkowników.

Nr. testu	Wykonane działanie	Oczekiwany rezultat	Rezultat
1	Próba rejestracji użytkownika z istniejącym już adresem e-mail	System wyświetla komunikat informujący o konieczności podania unikalnego adresu e-mail	Pozytywny

2	Próba rejestracji użytkownika bez podania adresu e-mail	System wyświetla komunikat informujący o konieczności podania adresu e-mail	Pozytywny
3	Próba rejestracji użytkownika bez podania hasła	System wyświetla komunikat informujący o konieczności podania hasła	Pozytywny
4	Próba rejestracji użytkownika bez potwierdzenia hasła	System wyświetla komunikat informujący o konieczności potwierdzenia hasła	Pozytywny
5	Próba rejestracji użytkownika z niezgodnymi hasłami	System wyświetla komunikat informujący o niezgodności haseł	Pozytywny
6	Próba rejestracji użytkownika z niepoprawnym adresem e-mail	System wyświetla komunikat informujący o niepoprawnym formacie adresu e-mail	Pozytywny
7	Próba rejestracji użytkownika z unikalnym adresem e-mail	System przeprowadza rejestrację użytkownika	Pozytywny
9	Próba rejestracji użytkownika z nieunikalnym adresem e-mail	System wyświetla komunikat informujący o konieczności podania unikalnego adresu e-mail	Pozytywny

Tabela. Przeprowadzonych testów interfejsu użytkownika.

Nr. testu	Wykonane działanie	Oczekiwany rezultat	Rezultat
1	Kliknięcie przycisku "Zarejestruj się" na stronie logowania	Przekierowanie do formularza rejestracji użytkownika	Pozytywny
2	Kliknięcie przycisku "Zaloguj się" na stronie głównej	Przekierowanie do formularza logowania	Pozytywny
3	Kliknięcie przycisku "Wyloguj" w panelu użytkownika	Wylogowanie użytkownika i przekierowanie na stronę główną	Pozytywny

4	Kliknięcie przycisku "Usługi" na stronie	Przekierowanie do panelu usług	Pozytywny
---	--	--------------------------------	-----------

Testy wydajnościowe

Przeprowadzono również testy wydajnościowe, po zgłębieniu wiedzy na temat sposobu działania narzędzia Lighthouse i jego zastosowania w testach wydajnościowych. przy użyciu narzędzia Lighthouse w wersji 10.0.1. Testy przeprowadzono dla strony internetowej pod adresem <http://localhost:5173/>. Wyniki testów były następujące:

- 13. Wynik wydajności: 66/100,
- 14. Dostępność: 83/100,
- 15. Najlepsze praktyki: 100/100,
- 16. SEO: 78/100.

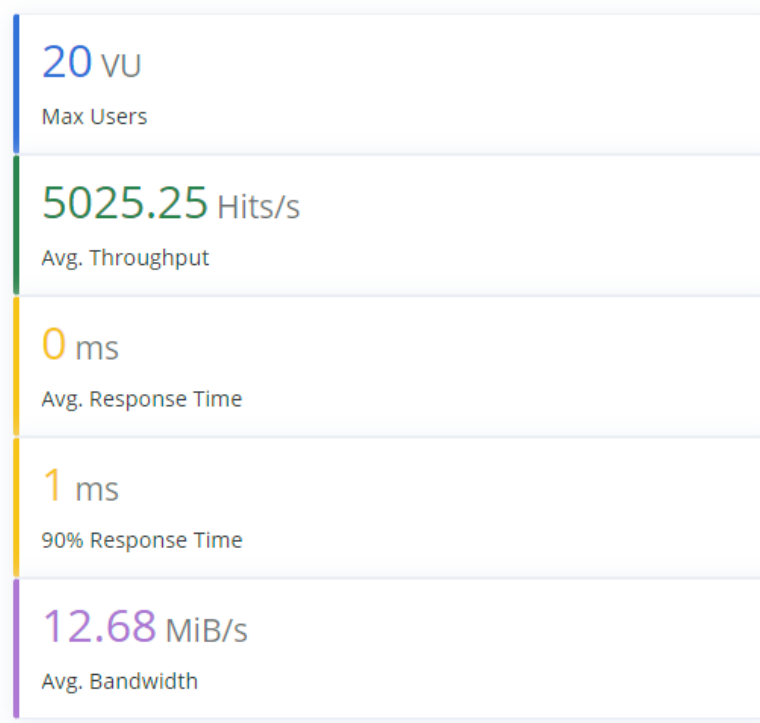


Rysunek. Testy wydajnościowe – Lighthouse.

W raporcie Lighthouse znaleziono wiele sugestii dotyczących optymalizacji utworzonej strony internetowej, takich jak umożliwienie kompresji tekstu, minimalizacja kodu JavaScriptu i wiele innych. Ponadto, raport zawiera informacje na temat zgodności strony z najlepszymi praktykami dotyczącymi dostępności SEO.

Testy zostały przeprowadzone na stronie internetowej przy użyciu emulatora Lighthouse na poziomie urządzenia stacjonarnego, a wyniki testów zostały wygenerowane przez Lighthouse o godzinie 1:54 PM GMT+2 dnia 12 maja 2024 roku.

Oprócz testów wydajnościowych przy użyciu narzędzia Lighthouse, przeprowadzono również test obciążeniowy za pomocą platformy BlazeMeter. W teście obciążeniowym wykorzystano 20 użytkowników jednocześnie, co przyniosło wynik maksymalnego obciążenia wynoszącego 5025.25 żądań na sekundę. Średnia przepustowość wyniosła 100%, co oznacza, że serwer był w stanie obsłużyć wszystkie żądania bez przeciążenia. Średni czas odpowiedzi wynosił 0ms, a 90% odpowiedzi na żądania miało czas nie przekraczający 1ms. Średnia przepustowość wyniosła 12.68MiB/s.



Rysunek. Wynik testu obciążeniowego.

Po przeprowadzeniu szczegółowych testów funkcjonalnych dokonano korekt w systemie, aby zoptymalizować jego funkcjonalność i zapewnić niezwykle wysoką jakość działania.

Dzięki tym korektom, otrzymany produkt jest w pełni zgodny z wymaganiami użytkowników. Wykonana praca przyczyniła się do stworzenia zaawansowanego i kompleksowego produktu, który spełnia współczesne standardy jakości i funkcjonalności.

Podsumowanie

Celem niniejszego projektu grupowego było stworzenie nowoczesnego i funkcjonalnego systemu wspierającego pracę serwisu urządzeń mobilnych. Postanowiono wykorzystać najnowsze narzędzia i technologie, aby zapewnić systemowi wysoką jakość i funkcjonalność.

Zaimplementowano aplikację webową, która umożliwia użytkownikom składanie zgłoszeń oraz dostęp do wszelkich usług oferowanych przez serwis. Dla administratorów stworzono interfejs umożliwiający szybkie i łatwe przeglądanie informacji o klientach oraz ich zgłoszeniach.

Najważniejszą cechą systemu jest wykorzystanie interfejsów programowania aplikacji (API) dostarczanych przez Firebase, dzięki czemu zapewniono szybką oraz niezawodną obsługę danych. Zastosowanie technologii React umożliwiło stworzenie natomiast intuicyjnego interfejsu, który ułatwia użytkownikom korzystanie z systemu i zapewnia odpowiednią jakość obsługi klienta.

W efekcie powstał system, który zapewnia zwiększenie efektywności oraz szybkości obsługi zgłoszeń serwisowych oraz monitorowania stanu urządzeń mobilnych. Charakteryzuje się on także wysoką wydajnością oraz skalowalnością, co pozwala na łatwą rozbudowę i dostosowanie do zmieniających się potrzeb biznesowych.

Wdrożenie utworzonego systemu przyczyni się do usprawnienia procesów w firmie oraz zwiększenia konkurencyjności na rynku. Podniesienie jakości obsługi klienta zaś pozwoli na budowanie pozytywnego wizerunku firmy oraz utrzymywanie długotrwałych relacji z klientami.