**A blue and black logo

Description automatically generatedWydział Finansów i Zarządzania**

**Kierunek: Informatyka**

**Jakub Czarnecki**

(nr albumu: 81495)

**Nowoczesna rozwiązanie Webowe wspierające pracę położnych środowiskowych i doradców laktacyjnych w sektorze medycznym**

**LUCYCOVER**

#### Inżynierska praca projektowa

**Opiekun merytoryczny**

mgr inż. Wojciech Barczyński

Wrocław 2024

Table of Contents

[WSTĘP 4](#_Toc170292718)

[1. Analiza biznesowa – założenia projektowe aplikacji oraz problemy do rozwiązania 5](#_Toc170292719)

[1.1 Funkcjonalność – w jakim zakresie projekt powinien wspierać użytkownika. 5](#_Toc170292720)

[1.2 Dostępność –Problem różnicy wieku, urządzeń mobilnych oraz niepełnosprawności 5](#_Toc170292721)

[1.3 Bezpieczeństwo – jak przetwarzać oraz jak je przechowywać dane w bezpieczny sposób. 6](#_Toc170292722)

[Podsumowanie 7](#_Toc170292723)

[2. Omówienie wykorzystanych technologii 7](#_Toc170292724)

[2.1 Ogólny opis wykorzystanych technologii w projekcie. 7](#_Toc170292725)

## WSTĘP

W dzisiejszych czasach sektor medyczny stale ewoluuje, wprowadzając nowoczesne technologie w celu poprawy efektywności i bezpieczeństwa opieki nad pacjentami. Jednym z kluczowych wyzwań, przed którymi stoją placówki medyczne, są archiwa papierowe, które nie tylko spowalniają procesy pracy, ale także zwiększają ryzyko utraty danych oraz błędów w dokumentacji.

Celem niniejszej pracy dyplomowej jest zaprojektowanie i implementacja aplikacji dedykowanej dla sektora medycznego, której zadaniem będzie wspieranie pracy położnych środowiskowych oraz doradców laktacyjnych działających na terenie Jeleniej Góry. Ta webowa aplikacja będzie centralnym systemem tworzenia oraz przechowywania dokumentacji medycznej pacjentów, bazą zaleceń wystawionych po każdej wizycie domowej oraz komunikatorem pomiędzy pacjentem a obsługującym go lekarzem/położną.

Obecnie wiele lokalnych przychodni nadal polega na fizycznych archiwach papierowych, co prowadzi do nieefektywności oraz zagrożeń związanych z utratą danych. LucyCover – bo tak nazywa się nasz projekt ma na celu przyspieszenie pracy personelu medycznego, zwiększenie bezpieczeństwa danych oraz ochronę środowiska poprzez zmniejszenie zużycia papieru, tuszu i energii elektrycznej.

Dodatkowo aplikacja ta wesprze personel medyczny w planowaniu codziennej pracy. Dzięki zaimplementowanemu terminarzowi, lekarz/położna będzie mogła szybko i efektywnie zaplanować swój dzień pracy. Planując wizyty poszczególnych pacjentów system automatycznie będzie informował wszystkich interesariuszy o zaplanowanej wizycie co pozwoli zautomatyzować procesy planowania oraz kontaktu z pacjentem. Unikniemy również wielu błędów komunikacyjnych takich jak nie poinformowanie pacjenta o zaplanowanej wizycie lub też jej odwołaniu.

Dodatkowo, dzięki zaawansowanej walidacji danych wprowadzanych do systemu, eliminujemy możliwość błędów w dokumentacji, co jest kluczowe dla zapewnienia wysokiej jakości opieki pacjentom.

LucyCover jako aplikacja webowa działająca w sieci lokalnej placówki będzie również niezwykle bezpiecznym rozwiązaniem. Odizolowanie baz danych oraz aplikacji od sieci publicznej praktycznie uniemożliwi wykradnięcie danych przez zdalnych cyberprzestępców. Dostęp do aplikacji uzyskają jedynie autoryzowane osoby posiadające swoje osobiste dane uwierzytelniania a wszystkie przechowywane w bazie dane będą szyfrowane za pomocą specjalnych algorytmów.

Niniejsza praca dyplomowa tworzona jest z myślą o wykorzystaniu w sieciach lokalnych odizolowanych od sieci Ethernet w celach bezpieczeństwa (jw.). Nie mniej jednak pozostawiona została możliwość wdrożenie aplikacji do działania w sieci VPN w przyszłości. Umożliwi to dostęp do systemu z każdego miejsca na Ziemi, co pozwoli położnym na wygodne i bezpieczne zarządzanie dokumentacją pacjentów bez konieczności przenoszenia ze sobą wielu fizycznych dokumentów. Wymaga to jednak ingerencji specjalistów dot. Cyberbezpieczeństwa w sieciach komputerowych, czego niniejsza praca nie obejmuje.

Praca ta ma na celu szczegółowe omówienie procesu projektowania, implementacji oraz potencjalnych korzyści wynikających z wprowadzenia opisywanej aplikacji dla sektora medycznego. Zaprezentowane rozwiązanie nie tylko zwiększy efektywność pracy personelu medycznego, ale również przyczyni się do podniesienia standardów bezpieczeństwa i ochrony danych w placówkach medycznych.

## Analiza biznesowa – założenia projektowe aplikacji oraz problemy do rozwiązania

W rozdziale dot. Analizy biznesowej zamieściłem wszystkie założenia projektowe jakie zostały ustalone z przedstawicielami sektora medycznego przed rozpoczęciem pracy nad projektem. W sekcji tej można się dowiedzieć jakie problemy należało rozwiązać z punktu widzenia zarówno funkcjonalności, dostępności dla użytkownika oraz bezpieczeństwa przetwarzania i przechowywania danych medycznych.

### Funkcjonalność – w jakim zakresie projekt powinien wspierać użytkownika.

Aplikacja LucyCover została zaprojektowana aby sprostać specyficznym problemom położnych środowiskowych oraz doradców laktacyjnych na terenie Jeleniej Góry. Poniżej przedstawiono kluczowe

funkcjonalności, które zostały uwzględnione w projekcie:

1. Możliwość wprowadzania, przechowywanie oraz modyfikacji danych osobowych pacjentów oraz ich dokumentacji medycznej dot. Stanu zdrowia fizycznego oraz psychicznego zarówno matki jak i noworodka w pierwszych tygodniach życia po opuszczeniu szpitala.
2. Możliwość wprowadzania, przechowywanie oraz modyfikacji dokumentacji medycznej zawierającej informacje o stanach rozwoju noworodka w pierwszych tygodniach życia oraz informacje o stanie matki po porodzie.
3. Wprowadzenie intuicyjnego kanału szybkiej komunikacji między personelem przychodni położniczej a pacjentem. Z poziomu aplikacji położna powinna mieć możliwość wysłania wiadomości email do swojego pacjenta.
4. Aplikacja powinna umożliwić użytkownikowi efektywne planowanie swojego dnia pracy/wizyt. Dostępny powinien być terminarz w, którym będzie można planować wizyty na kolejne dni pracy.
5. Wyżej wspomniany terminarz powinien jednocześnie posiadać opcje, która umożliwi użytkownikowi szybkie wysłanie informacji do wybranego pacjenta na temat zaplanowania wizyty położnej w jego domu lub jej odwołania za pomocą poczty elektronicznej (email)
6. Projekt powinien rozwiązywać problem ręcznego wypisywania zaleceń medycznych dla pacjenta po wizycie. Po każdej wizycie położna będzie mogła w aplikacji szybko wypisać zalecenia dla swojej pacjentki, które zostaną automatycznie sformatowane przez systemu do czytelnej i profesjonalnej formy. Zalecenia te zostaną zapisane na koncie pacjenta do momentu ich ręcznego usunięcia przez użytkownika. Dodatkowo użytkownik ma, możliwość wydrukowania zaleceń aby przekazać je pacjentowi w wydrukowanej formie. Na tą chwilę jest to jedyna forma przekazania pacjentowi zalecań. (Aplikacja nie wysyła zaleceń drogą elektroniczną jednak pozostawia opcje zapisania jej jako plik PDF)

### Dostępność –Problem różnicy wieku, urządzeń mobilnych oraz niepełnosprawności

Projekt został zaprojektowany z myślą o intuicyjności oraz prostocie użytkowania. Aplikacja musi być prosta i dostępna dla osób starszych i mniej zaawansowanych technologicznie jednocześnie nie odstraszać młodszego pokolenia szaro-czarnym wyglądem. Z tego też powodu UI (user interface) [[1]](#footnote-1) zbudowany jest z dużych widocznych bloków oraz wyraźnej czcionki. W kluczowych miejscach aplikacji zastąpiono tekst ikonami lub zdjęciami aby kontent był czytelny oraz intuicyjna a zarazem estetyczny i elegancki. Jako aplikacja o tematyce położniczej dominującym kolorem jest róż i biel nawiązujący do odzieży medycznej noszonej przez położne.

LucyCover stworzone zostało również z myślą o dostępności dla osób niepełnosprawnych. Wielkość i kolor czcionki oraz tło na jakim się znajduje dobrane zostało również aby ułatwić pracę osobą mający problem ze wzrokiem. Wielkość ikon oraz formularzy do wprowadzania danych również została powiększona dla osób mających problemy z płynną obsługą myszki komputerowej.

Premiera iPhone’a w 2007 roku znacząco wpłynęła na rozwój aplikacji mobilnych[[2]](#footnote-2).Zdaniem firmy Research.com statystyki pokazują, że w dzisiejszych czasach ponad połowa osób korzysta z aplikacji webowych z poziomu urządzeń mobilnych takich jak smartfony, tablety. LucyCover wychodzi naprzeciw tym oczekiwania i zostało stworzona w taki sposób aby być dostępnym zarówno dla urządzeń mobilnych jak i desktopowych. Dzięki temu aplikacja po zainstalowaniu na serwerze i uruchomieniu w sieci lokalnej będzie mogła być używana bezpośrednio podczas wizyty w domu pacjenta za pomocą sieci VPN.

A graph of a bar chart

Description automatically generated with medium confidence[[3]](#footnote-3)

### Bezpieczeństwo – jak przetwarzać oraz jak je przechowywać dane w bezpieczny sposób.

Wraz z rozwojem sieci internetowej oraz korzystających z niej aplikacji webowych pojawił się problem związany z dostępem do zasobów. Aplikacje udostępnione w sieci narażone są na bezpośrednie ataki ze strony cyberprzestępców. Omawiana przez nas aplikacja jako że dotyczy sektora medycznego będzie miała styczność z danymi, które są poufne i dostęp do nich powinien być szczególnie chroniony. W związku z tym ustalone zostało, że aplikacja zostanie uruchomiona i dostępna jedynie w sieci lokalnej. W związku z koniecznością stałego połączenia aplikacji z siecią zewnętrzną do celów obsługi protokołów poczty elektronicznej koniecznym było staranne zaprojektowanie strategii zabezpieczenia i ochrony składowanych danych w bazie danych oraz ich przetwarzania. W związku z tym zaplanowano następujące instrukcje:

1. Komunikacja między warstwą prezentacji oraz warstwą logiki biznesowej powinna być nawiązana z wykorzystaniem protokołu HTTPS[[4]](#footnote-4) (szyfrowanie danych podczas transmisji)
2. Przetwarzane dane powinny być odpowiednio walidowane oraz szyfrowane przed zapisem do bazy danych
3. Baza danych powinna być oddzielną zabezpieczoną strukturą.
4. Wszystkie dane w bazie danych powinny być zaszyfrowane i niezrozumiałe dla człowieka oraz sztucznej inteligencji
5. Dostęp do wszystkich interfejsów API powinien być poprzedzony autentykacją oraz autoryzacją.
6. Aplikacja powinna wymagać od użytkownika planowania oraz wykonywania regularnych kopi zapasowych (backup) z wykorzystaniem reguły (3-2-1)[[5]](#footnote-5)
7. Warstwa aplikacji powinna zapewniać pierwszą linię obrony poprzez walidacje i sprawdzenie wprowadzanych danych przez użytkownika zanim zostaną one wysłane na poszczególne interfejsy końcowe warstwy logiki biznesowej.

### Podsumowanie

W ramach analizy biznesowej aplikacji omówiono założenia projektowe, uwzględniając potrzeby funkcjonalne, dostępność oraz bezpieczeństwo danych medycznych. Aplikacja ma wspierać położne i doradców laktacyjnych, oferując m.in. możliwość zarządzania danymi pacjentów, intuicyjną komunikację oraz efektywne planowanie wizyt. Projekt zakłada prosty i dostępny interfejs dla użytkowników w różnym wieku i o różnym stopniu zaawansowania technologicznego, a także zapewnienie bezpieczeństwa poprzez odpowiednie szyfrowanie, walidację danych oraz dbałość o reguły bezpiecznego tworzenia aplikacji webowych. Projekt będzie działać w sieci lokalnej, co ma minimalizować ryzyko ataków cybernetycznych. Jednak nie wyklucza oraz nie zamyka furtki na udostępnienie projektu w sieci z wykorzystaniem sieci VPN.

## Omówienie wykorzystanych technologii

Rozdział ten szczegółowo tłumaczy jakie problemy rozwiązuje projekt oraz jakie wytyczne zostały ustalone podczas analizy biznesowej obsługiwanego sektora.

Rozdział ten szczegółowo opisuje jakie technologie zostały wykorzystane w projekcie i w jaki sposób zrealizowana została komunikacja między nimi. Wytłumaczone zostaje również dlaczego zostały podjęte decyzje o wykorzystaniu konkretnych technologii oraz w jaki sposób zostały zaimplementowane.

### Ogólny opis wykorzystanych technologii w projekcie.

Projekt LucyCover wykorzystuje nowoczesne technologie, które zapewniają wysoką wydajność, skalowalność oraz bezpieczeństwo. Aby w pełni odizolować obszary odpowiedzialności projekt został podzielony na 3 oddzielne aplikacje tworząc tak zwaną **architekturę trójwarstwową[[6]](#footnote-6).** Każda z tych warstw pełni specyficzne funkcje i odpowiada za inne aspekty działania aplikacji.

Pierwsza warstwa **FROTNEND (warstwa prezentacji)**odpowiedzialna jest za interakcje użytkownika z aplikacją. Została ona utworzona przy wykorzystaniu języka programowania JavaScript, który swoja szybkością i wysoką dostępnością rozwiązań webowych idealnie sprawdził się w tej roli. Niestety tworzenie zaawansowanych aplikacji frontendowych przy użyciu jedynie tego języka programowania byłoby bardzo czasochłonne i nieefektywne. Naprzeciw tym problemom wychodzi biblioteka REACT w wersji 18. To w tej technologii i jej towarzyszących została utworzona cała wizualna warstwa projektu. Drugą warstwą tego rozwiązania jest **Backend (warstwa logiki biznesowej).** Jako, że projekt wspiera sektor medyczny istnieje wiele krytycznych aspektów które muszą zostać pokryte aby zachować skalowalność, wydajność a przede wszystkim bezpieczeństwo. To właśnie backend odpowiedzialny jest za wykonywanie wszystkich założeń logiki biznesowej aplikacji taki jak przyjmowanie danych, walidacje, autoryzacje oraz autentykacje, odpowiednie przygotowanie danych do zapisu w bazie danych oraz ich pobrania. Do obsługi tego ważnego zadania wykorzystałem język programowania utworzony przez firmę Microsoft – C#. Język ten wspiera wiele bardzo zaawansowanych technologii do obsługi aplikacji po stronie serwera. Dodatkowym powodem wykorzystania tej technologii jest świetna współpraca tego języka z serwisami dostępnymi na chmurze obliczeniowej Azure (LucyCover pozostawia możliwość uruchomienia aplikacji w sieci). W języku C# stworzona została technologia ASP .NET CORE, która wychodzi naprzeciw problemom stawianym przez nasz projekt. To właśnie ta technologia została wykorzystana do przygotowania warstwy logiki biznesowej mojej aplikacji. Ostatnią warstwą naszego projektu jest **warstwa danych (czyt. Baza danych)**. MS SQL odpowiedzialny jest za przechowywanie i zarządzanie danymi aplikacji. To tutaj dane są zapisywane, odczytywane, aktualizowane i usuwane.

W architekturze trójwarstwowej każda warstwa jest od siebie niezależna, co pozwala na większą elastyczność, skalowalność oraz łatwiejsze zarządzanie i rozwój aplikacji.

Każda warstwa projektu korzysta ze swojej głównej technologii oraz wielu bibliotek wspomagających oraz usprawniających ich pracę. Aby LucyCover mogła wydajnie sprostać wszystkim powierzonym jej zadaniom wykorzystałem do jej utworzenia następujące technologie:

* C#[[7]](#footnote-7) 11
* .Net 7[[8]](#footnote-8)
* Entity Framework [[9]](#footnote-9)
* ASP .NET CORE – REST API[[10]](#footnote-10)
* AutoMapper[[11]](#footnote-11)
* FluentValidator[[12]](#footnote-12)
* Nlog[[13]](#footnote-13)
* JwtBearer[[14]](#footnote-14)
* Swashbuckle/swawgger[[15]](#footnote-15)
* xUnit[[16]](#footnote-16)
* FluentAssertions[[17]](#footnote-17)
* HTML 5 / CSS3[[18]](#footnote-18)
* JavaScript[[19]](#footnote-19)
* React.js[[20]](#footnote-20)
* React Redux[[21]](#footnote-21)
* React Router[[22]](#footnote-22)
* React @Tanstack Query[[23]](#footnote-23)
* Vite[[24]](#footnote-24)
* Node[[25]](#footnote-25)

1. https://ks.pl/slownik/co-to-jest-ui [↑](#footnote-ref-1)
2. https://www.dbc.wroc.pl/Content/39276/Ratalewska\_Rozwoj\_Rynku\_Aplikacji\_Mobilnych\_2017.pdf [↑](#footnote-ref-2)
3. https://research.com/software/mobile-vs-desktop-usage#4 [↑](#footnote-ref-3)
4. https://poradnikprzedsiebiorcy.pl/-roznice-miedzy-http-a-https-i-ich-wplyw-na-pozycje-strony [↑](#footnote-ref-4)
5. https://www.google.com/search?q=technologia+backup+3-2-1&rlz=1C1GCEJ\_enPL1061PL1061&oq=technologia+backup+3-2-1&gs\_lcrp=EgZjaHJvbWUyBggAEEUYOTIICAEQABgWGB4yCAgCEAAYFhgeMggIAxAAGBYYHjIICAQQABgWGB4yCAgFEAAYFhgeMggIBhAAGBYYHjIICAcQABgWGB4yCAgIEAAYFhgeMggICRAAGBYYHtIBCDUzNTBqMGo0qAIAsAIA&sourceid=chrome&ie=UTF-8 [↑](#footnote-ref-5)
6. https://www.ibm.com/docs/pl/was/9.0.5?topic=overview-three-tier-architectures [↑](#footnote-ref-6)
7. https://en.wikipedia.org/wiki/C\_Sharp\_(programming\_language) [↑](#footnote-ref-7)
8. https://vavatech.pl/technologie/frameworki/.net [↑](#footnote-ref-8)
9. https://learn.microsoft.com/pl-pl/dotnet/framework/data/adonet/ef/overview [↑](#footnote-ref-9)
10. https://www.plukasiewicz.net/Artykuly/AspNetCoreFeatures [↑](#footnote-ref-10)
11. https://cezarywalenciuk.pl/blog/programing/automapper-z-aspnet-core [↑](#footnote-ref-11)
12. https://docs.fluentvalidation.net/en/latest/ [↑](#footnote-ref-12)
13. https://nlog-project.org/ [↑](#footnote-ref-13)
14. https://sandrino.dev/blog/aspnet-core-5-jwt-authorization [↑](#footnote-ref-14)
15. https://learn.microsoft.com/pl-pl/aspnet/core/tutorials/getting-started-with-swashbuckle?view=aspnetcore-8.0&tabs=visual-studio [↑](#footnote-ref-15)
16. https://boringowl.io/blog/zwiekszenie-wydajnosci-testowania-kodu-za-pomoca-xunit [↑](#footnote-ref-16)
17. https://fluentassertions.com/ [↑](#footnote-ref-17)
18. https://medium.com/@ojbrot/everything-you-need-to-know-about-html5-and-css3-e83c80b5bb8b [↑](#footnote-ref-18)
19. https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\_steps/What\_is\_JavaScript [↑](#footnote-ref-19)
20. https://react.dev/ [↑](#footnote-ref-20)
21. https://redux.js.org/ [↑](#footnote-ref-21)
22. https://reactrouter.com/en/main [↑](#footnote-ref-22)
23. https://tanstack.com/query/latest [↑](#footnote-ref-23)
24. https://vitejs.dev/ [↑](#footnote-ref-24)
25. https://nodejs.org/en [↑](#footnote-ref-25)