

Akademia Górniczo - Hutnicza im. Stanisława Staszica w Krakowie

Wydział Informatyki, Elektroniki i Telekomunikacji

Katedra Informatyki



Mniam Mniam!

*Aplikacja webowa umożliwiająca przechowywanie, ocenianie, oraz udostępnianie innym
użytkownikom przepisów kulinarnych*

Dokumentacja koncepcyjna

Autorzy: Krzysztof Wrona, Krzysztof Suruło, Piotr Puskarski, Piotr Maciejowski

Spis treści.

[1. Informacje wstępne.](#)

[2. Opis problemu.](#)

[2.1. Kontekst](#)

[2.2. Potrzeby użytkownika:](#)

[3. Wizja rozwiązania.](#)

[3.1. Wizja.](#)

[3.2. Wymagania niefunkcjonalne.](#)

[4. Koncepcja systemu.](#)

[4.1. Opis działania systemu.](#)

[4.2. "Typowy" przebieg użycia aplikacji.](#)

[4.3. Architektura systemu.](#)

[4.3. Struktura modułów.](#)

[4.5. Uwarunkowania technologiczne.](#)

[4.6. Wybrane technologie.](#)

4.6. Wybrane technologie.

1. Informacje wstępne.

Niniejszy projekt powstał w ramach przedmiotu Inżynieria Oprogramowania przeprowadzonego na Akademii Górniczo - Hutniczej im. Stanisława Staszica w Krakowie, Wydział Informatyki, Elektroniki i Telekomunikacji, kierunek Informatyka, rok III, semestr VI, rok akademicki 2016/2017.

Grupa ćwiczeniowa:

Środa, 9.35

Opiekun:

mgr inż. Witold Rakoczy

Skład zespołu:

Piotr Puszkarski

Krzysztof Suruło

Krzysztof Wrona

Piotr Maciejowski

2. Opis problemu.

2.1. Kontekst

Współcześnie coraz więcej ludzi nie wyobraża sobie życia bez urządzeń mobilnych - tablety, smartfony i smartwatche stają się coraz bardziej zaawansowane technologicznie. Tradycyjny papier i zrobione z niego książki i notesy powoli odchodzą do lamusa. Są ciężkie, nieporęczne, zajmują relatywnie dużo miejsca i miejsce w nich kończy się niezwykle szybko.

Młodzi ludzie potrzebują jednak czegoś co zastąpiłoby im wspomniane powyżej przyrządy do notowania najróżniejszych informacji. Jednym z przykładów notesu który powinien zostać zastąpiony jest książka kucharska. Młodzi kucharze zmuszeni są dzisiaj do korzystania z przestarzałych, zniszczonych książek kucharskich sprzed paru lat, albo do długiego przeszukiwania internetu i późniejszego zapisu przepisu w czymś w rodzaju notatnika na komputerze osobistym. Plik w którym zostaną zapisane takie przepisy ma nierzadko niejednorodną strukturę, często brakuje informacji o potrzebnych składnikach czy sposobie przygotowania. Po spróbowaniu danej potrawy nie ma żadnej drogi na wygodne danie oceny danej potrawy i posortowanie istniejących potraw po ich "jakości". Zastanówmy się również co musi zrobić młody człowiek aby podzielić się nowo odkrytym, znakomitym przepisem ze znajomymi. Musi on skorzystać z jakiejś zewnętrznej formy komunikacji, skopiować ze swojego pliku z przepisami daną potrawę i wysłać ją znajomym. Tutaj pojawia się problem związany z tym, że jeśli nasz użytkownik odkryje jakiś sposób, aby poprawić smak danej potrawy i zmodyfikuje przepis, to aby jego znajomi posiadali aktualną wersję musi on powtórzyć procedurę wysyłania. To niewygodnie, zajmuje dużo czasu i zasobów.

2.2. Potrzeby użytkownika:

1. Potrzeba zanotowania przepisu
2. Potrzeba udostępnienia przepisu znajomym w łatwy sposób
3. Potrzeba znajdowania nowych lub dobrych przepisów
4. Potrzeba wglądu do przepisów znajomych oraz własnych
5. Potrzeba obliczenia odpowiedniej ilości składników aby przygotować potrawy dla wyznaczonej ilości osób
6. Potrzeba otrzymywania informacji o zmianach w przepisach

2.3. Możliwości użytkownika

- Zapisywanie potrawy w papierowym notesie
- Wysyłanie potrawy używając zewnętrznej komunikacji do znajomych
- Obliczenie ilości potrzebnych składników na kalkulatorze
- Stworzenie rankingu potraw na kartce.
- Noszenie ze sobą papierowego notesu w celu wglądu w przepisy

2.4. Obszary problemu pokryte przez możliwości.

Jak widzimy wszystkie możliwości jakie ma użytkownik są bardzo ograniczone. Co prawda można użyć papieru do zapisu potrawy, ale w notesach może skończyć się miejsce, są one nieporęczne i ciężkie, Dzielenie się z innymi potrawą przez zewnętrzną komunikację wymaga dużego wysiłku i częstego powtarzania w celu aktualizacji przepisu. Do obliczenia ilości potrzebnych składników potrzeba kalkulatora, a stworzenie rankingu potraw wymaga przepisania wszystkich wcześniej posortowanych potraw. Późniejsza aktualizacja tego rankingu wymaga ilości pracy rosnącej wykładniczo. W rozwiązaniu większości (prawdopodobnie bez potrzeby obliczenia ilości potrzebnych na obiad składników) ma pomóc nasza aplikacja.

3. Wizja rozwiązania.

3.1. Wizja.

System pozwoli użytkownikowi na logowanie się za pomocą własnego konta Google, publikowanie własnych przepisów, przeglądanie, pobieranie i ocenianie przepisów opublikowanych przez innych klientów systemu. Przy publikowaniu swojego przepisu, klient podaje składniki, ich ilość oraz procedurę wykonania dania. Użytkownik będzie mógł oceniać inne przepisy za pomocą skali od 1 do 5. Przy przeglądaniu przepisów innych użytkowników ma możliwość użycia filtrów, żeby pozbyć się dań, które go nie interesują. Gdy znajdzie interesujący go przepis, może dodać go lub ocenić.

Po wprowadzeniu systemu na rynek klient będzie miał możliwość użycia go za pomocą przeglądarki stron WWW lub też pobrania go ze sklepu z aplikacjami. Jedynym warunkiem korzystania z systemu jest konieczność posiadania dostępu do internetu oraz przeglądarki lub też urządzenia mobilnego.

Dzięki wprowadzonemu systemowi użytkownik będzie mógł zastąpić tradycyjny sposób dzielenia się przepisami w zdecydowanie bardziej przenośny sposób (wystarczy urządzenie z dostępem do internetu).

Dzięki tej aplikacji użytkownik będzie mógł w każdym miejscu i dowolnym czasie wybrać przepis. System pozwoli na łatwe i długie przechowywanie przepisów wszystkich użytkowników.

Z założenia użytkowanie aplikacji powinno być proste i intuicyjne, dlatego nie ma sensu tworzenie specjalistycznych poradników dotyczących używania systemu. Klientowi powinna wystarczyć dokumentacja użytkowa.

Aplikacja powinna udostępniać prosty interfejs do wprowadzania danych oraz ogólne, proste w zrozumieniu i nieskomplikowane metody przeglądania i oceniania przepisów innych użytkowników tego systemu. Ocena przepisu będzie wyświetlana jako ilość gwiazdek.

Ważnym aspektem jest także użyteczność aplikacji. Należy zadbać aby wyszukiwanie ulubionych przepisów, czy zapisywanie własnego przepisu nie trwało więcej niż zrobienie tego bez użycia aplikacji, a także żeby wyświetlanie przepisu było przejrzyste.

3.2. Wymagania niefunkcjonalne.

1. System powinien składać się z dwóch pomniejszych aplikacji - klienta oraz serwera webowego.

4. Koncepcja systemu.

4.1. Opis działania systemu.

Realizowany system będzie się składał z dwóch części: aplikacji oraz serwera webowego z bazą danych, w której będą zapisywane dane użytkowników.

Aplikacja będzie natychmiast przysyłała dane na serwer poprzez protokół HTTP. W ten sposób będą również pobierane wszystkie dane.

Po uruchomieniu aplikacji użytkownik będzie miał możliwość zalogowania się. Po przejściu do panelu przepisów będzie mógł przeglądać wszystkie przepisy, lub dodać nowy przepis.

Po kliknięciu na przepis, otwiera się okno przepisu. W oknie tym użytkownik może ocenić przepis, a jeżeli jest to jego przepis to może dokonać go zmodyfikować lub usunąć.

4.2. “Typowy” przebieg użycia aplikacji.

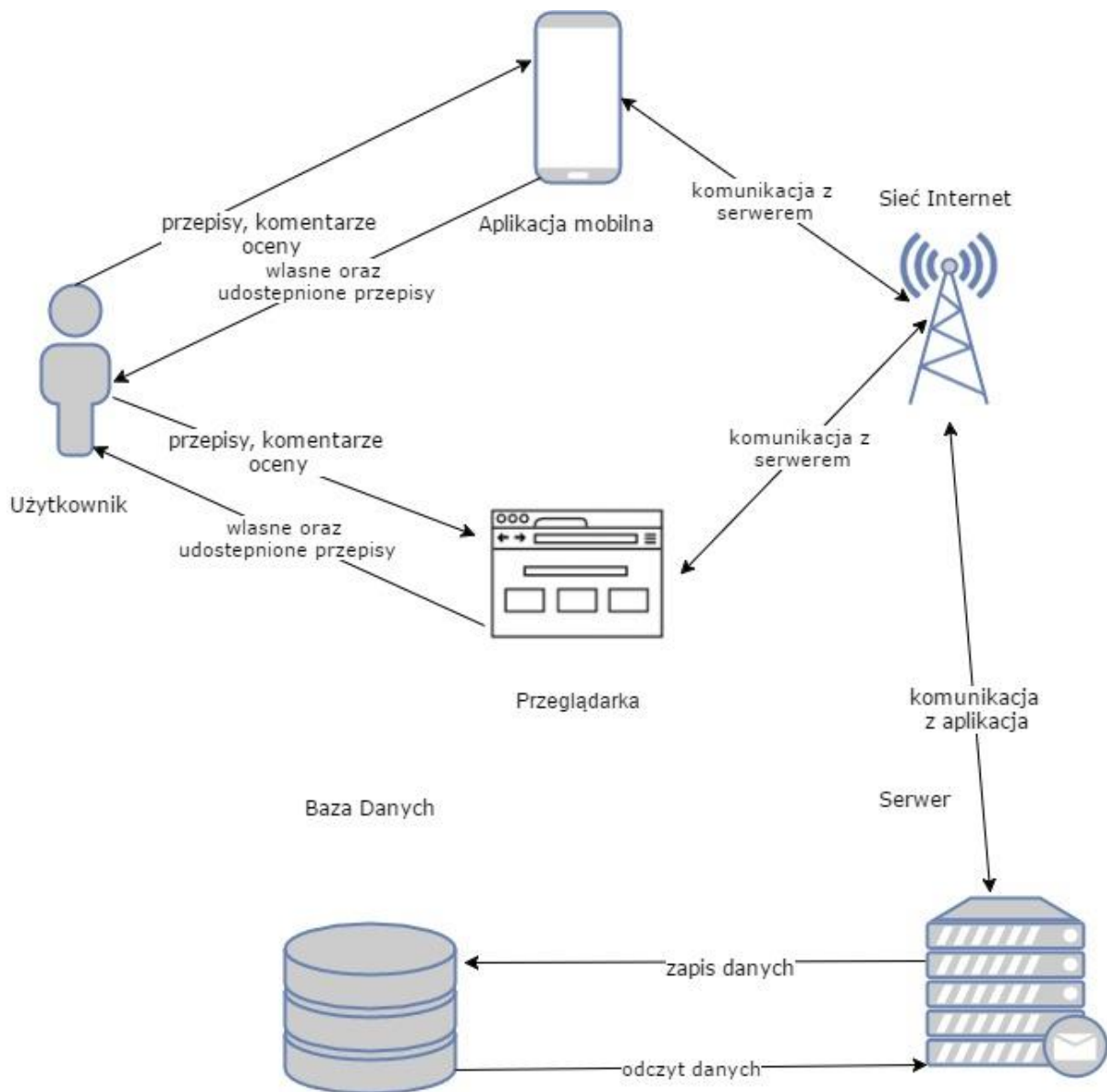
- Logowanie do systemu

- Możliwość skorzystania z funkcji aplikacji:

- * dodania przepisu
- * sprawdzenia przepisów
- * przeglądanie przepisów innych użytkowników
- * ocena potraw innych użytkowników
- * modyfikacja przepisów

- Wylogowanie się z aplikacji

4.3. Architektura systemu.



Rysunek 1. Diagram ideowy architektury systemu.

Architektura systemu przypomina architekturę klient-serwer. Aplikacja mobilna i serwer webowy komunikują się przez protokół HTTP.

- **Aplikacja mobilna** zapewnia użytkownikowi możliwość prostej interakcji z interfejsem graficznym, który pośredniczy w procesie dodawania przepisów. Za pośrednictwem GUI użytkownik ma możliwość wprowadzenia:

- przepisów
- ocen.

Interfejs graficzny pozwala na zaprezentowanie użytkownikowi:

- Jego przepisów
- Przepisów innych użytkowników.

- Ocen przepisów
- **Serwer webowy** pozwala na operacje:
 - Zalogowania
 - Utworzenia konta
 - Dodawania przepisu
 - Modyfikowania przepisu
 - Usuwania przepisu
 - Pobierania przepisów
 - Dodawania oceny
 - Modyfikowania oceny
 - Usuwania oceny
 - Pobierania ocen
- **Baza danych serwera** służy do przechowywania danych dotyczących użytkowników, przepisów i ocen pochodzących od aplikacji mobilnej.

Przypisanie poszczególnych elementów do węzłów fizycznych:

- *Aplikacja mobilna* znajduje się na urządzeniu mobilnym użytkownika.
- *Serwer webowy* i *Baza danych serwera* znajduje się na infrastrukturze zdalnego serwera dostarczanego i administrowanego przez twórców aplikacji.

4.3. Struktura modułów.

W obszarze klienta przewidziane są następujące moduły:

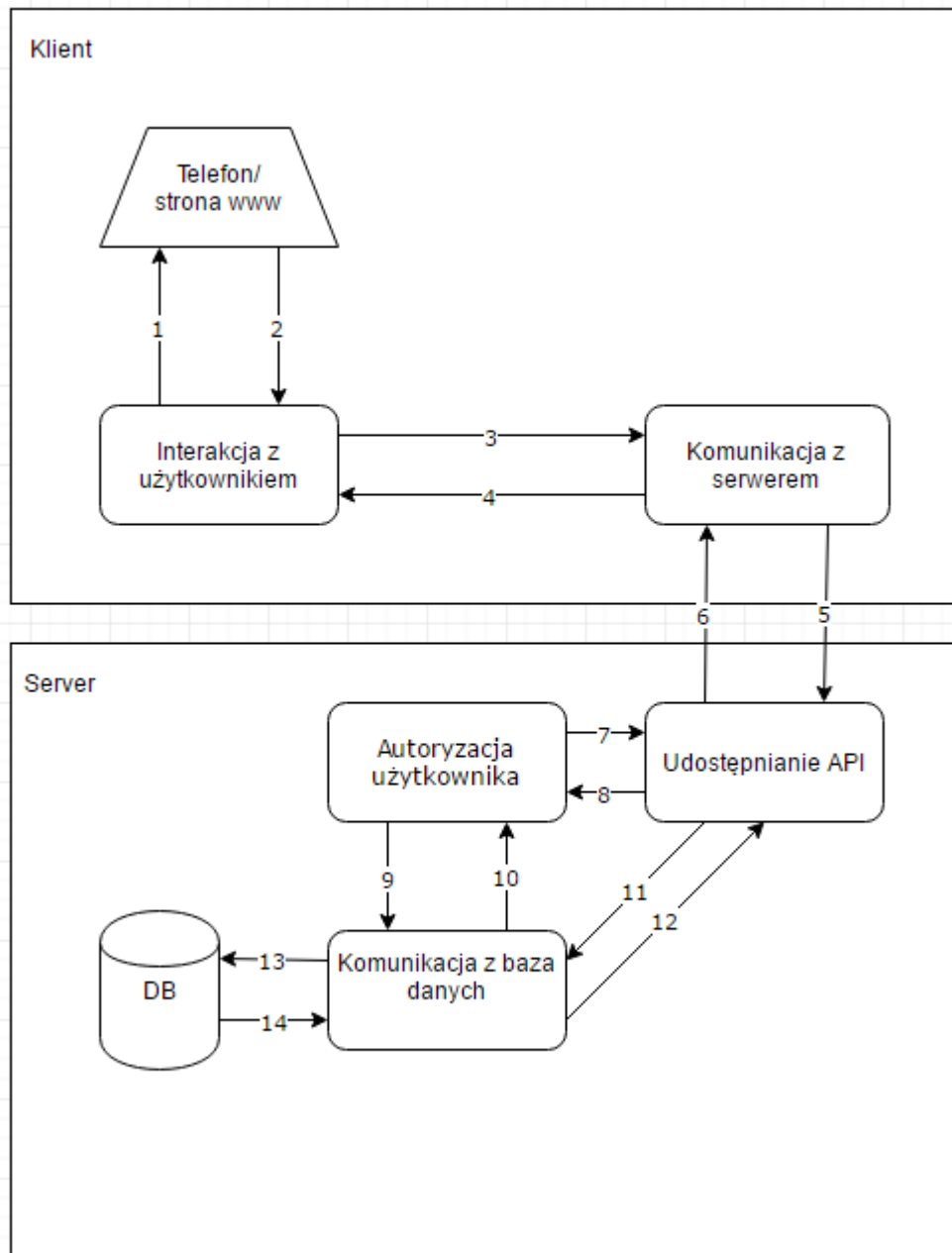
- Moduł komunikacji z zewnętrznym serwerem
 - Dane przesyłane są poprzez protokół HTTP

- Moduł odpowiedzialny za poprawną interakcję użytkownika z interfejsem
 - Wyświetla pola wprowadzania danych, przyciski, generalny interfejs.
 - Zarządza odpowiednią kolejnością prezentowanych widoków.

W obszarze serwera:

- Moduł odpowiedzialny za komunikację z bazą danych
 - Do komunikacji z bazą wykorzystany zostanie entity framework
 - Pozwala na pobieranie i zapisywanie danych do bazy
- Moduł odpowiedzialny za autentykację użytkowników
 - Pozwala rozpoznać użytkownika za pomocą autoryzacji Google.

Poniżej zamieszczono diagram komunikacji modułów bazujący na powyższym planie rzeczowym



Rysunek 2. Diagram komunikacji modułów.

1.Logowanie

Użytkownik włącza aplikację, a moduł odpowiedzialny za interakcję z interfejsem wyświetla okno logowania do konta(1). Po wpisaniu (2) dane przekazywane są przez moduł komunikacji z serwerem w postaci plików json(3) do serwera, gdzie odbiera je moduł odpowiedzialny za przyjmowanie requestów HTTP(5) i przekazuje do modułu odpowiedzialnego za autentykację użytkowników(8). Ten dokonuje autoryzacji użytkownika, generuje odpowiedni token dzięki któremu system jest w stanie identyfikacji użytkownika i zapisuje go w bazie danych(13). Następnie zwraca tą samą ścieżką do użytkownika informacje (14,10,7,6,4) a moduł odpowiedzialny za interakcję z użytkownikiem aktualizuje widok(1).

2. Obsługa przepisów

Po poprawnym zalogowaniu, użytkownik może korzystać już ze wszystkich funkcjonalności aplikacji takich jak: dodawanie, usuwanie, modyfikowanie, przeglądanie, wyszukiwanie, pobieranie czy ocenianie przepisów

2.1.Dodawanie, usuwanie, modyfikacja przepisów.

Aby dodać przepis, należy kliknąć przycisk "Dodaj przepis"(1), po którym widok przenosi klienta do odpowiedniego okna(2). Klient musi wypełnić odpowiednie pola składników oraz metody przyrządzania, a także wybrać kategorie do których należy przepis oraz ustawić jego prywatność(1). Po zaakceptowaniu moduł interakcji z klientem przekazuje dane jako odpowiednie pliki json do modułu komunikacji z serwerem(3), a ten z kolei do modułu przyjmującego requesty HTTP(5). Gdy informacja trafi na serwer przez moduł do komunikacji z bazą danych(11) tworzone są odpowiednie zapytania SQL'owe, które wpisują do bazy danych informacje o przepisie oraz użytkownika, który go dodał(13). Na podobnej zasadzie działa usuwanie oraz modyfikowanie przepisów, z tą różnicą, że w widoku musimy wybrać odpowiedni przepis, a w module do komunikacji z bazą danych odpowiednio tworzymy zapytania SQL'owe, które odpowiednio usuwają lub zmieniają odpowiednie krotki w tabelach(13).

2.2 Przeglądanie, wyszukiwanie przepisów.

Kolejnymi ważnymi funkcjami są możliwość przeglądania lub wyszukiwania przepisów własnych oraz udostępnionych przez innych użytkowników. Po kliknięciu w widoku przycisku przeglądaj przepisy(2), podobnie jak w przypadku dodawania przepisu skierowane zostaje zapytanie do serwera odpowiednio przez ścieżkę (3,5,11,13). Gdy zapytanie trafi do modułu komunikacji z bazą danych tworzone jest zapytanie które zwróci przepisy(14) z odpowiedniej kategorii (własne lub udostępnione). Dodatkowo przy użyciu opcji wyszukaj i podaniu oczekiwanych haseł, zapytanie jest odpowiednio modyfikowane i zwraca tylko te przepisy, które mają je w nazwie(14). Otrzymana lista przepisów wraz z odpowiednimi ID, które umożliwiają zidentyfikowanie przepisu, jest zwracana do użytkownika przez moduł odpowiedzialny za requesty HTTP(6). Następnie po stronie klienta moduł do komunikacji z serwerem odbiera je, przetwarza i przekazuje do widoku(4), który odpowiednio prezentuje listę użytkownikowi.

2.3 Wyświetlanie, pobieranie przepisów

Mając listę przepisów użytkownik może je wyświetlić bądź też pobrać je na swoje urządzenie. Po kliknięciu w dany przepis znów odpytuje serwer, tym razem o wszystkie informacje przepisu o danym ID(2,3,5,11,13). Serwer zwraca je nam, a moduł odpowiedzialny za interakcję z użytkownikiem odpowiednio je prezentuje(14,12,6,4,1). Mając wyświetlony przepis użytkownik może chcieć przenieść go na swoje urządzenie. W tym celu widok udostępnia przycisk "Pobierz przycisk" dzięki czemu moduł Interakcji z użytkownikiem wysyła przepis w odpowiednim formacie do użytkownika(1).

3 Wylogowanie

Po skończonej sesji użytkownik powinien się wylogować. W tym celu moduł do interakcji z użytkownikiem udostępnia przycisk "Wyloguj"(1). Po jego naciśnięciu kierowane jest na serwer odpowiednie żądanie(2,3,5,11). Z serwera usuwany jest tymczasowy token po którym rozpoznawany jest klient(13), dzięki czemu nie może zostać wykonana żadna akcja, dopóki użytkownik nie zaloguje się ponownie. Również, jeśli użytkownik nie będzie aktywny przez określony czas token straci ważność i dojdzie do automatycznego wylogowania.

4.4. Opis interfejsu użytkownika.

Interfejs użytkownika będzie się składał z następujących części części:

- Ekran startowy aplikacji
- Lista przepisów
- Okno przepisu
- Okno dodawania przepisu
- Okno dodawania oceny
- Okno edycji przepisu
- Okno edycji oceny

4.5. Uwarunkowania technologiczne.

Do tworzenia aplikacji mobilnych mogących działać jako aplikacje *standalone* wykorzystywana jest Java w procesie Android Software Development.

Od części serwerowej wymagana jest prostota, szybkość działania. Istnieje wiele technologii spełniających te kryteria. Wybór zostanie podyktowany poziomem znajomości poszczególnych przez członków zespołu.

4.6. Wybrane technologie.

W przypadku aplikacji mobilnej wybór padł na utworzenie aplikacji na system android w języku java.

Technologią wykorzystaną do napisania serwera jest framework asp.net, serwer zostanie napisany w języku C#. Do komunikacji z bazą danych wykorzystany zostanie entity framework.

ASP.NET cieszy się bardzo dużą popularnością, przez co jest on sprawdzony oraz posiada wiele pomocnych przy dewelopowaniu narzędzi.

Ważny aspektem jest również fakt, że zespół projektowy posiada doświadczenie w tej technologii.

Z racji wyboru entity framework do komunikacji z bazą danych, zdecydowaliśmy się na MS SQL.

Komunikacja z serwerem będzie odbywać się przy użyciu protokołu TCP, ponieważ gwarantuje on wyższym warstwom komunikacyjnym dostarczenie wszystkich pakietów w całości, z zachowaniem kolejności i bez duplikatów.