



AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

INŻYNIERIA OPROGRAMOWANIA

DOKUMENT KONCEPCYJNY

Wish List

Autorzy:

Łukasz GAJEWSKI

Grzegorz SIATKA

Patrycja WRONA

2016

Spis treści

1	Opis problemu	2
1.1	Definicja problemu	2
2	Wizja rozwiązania	3
2.1	Usługi	3
3	Koncepcja systemu	5
3.1	Schemat systemu	5
3.2	Moduły	6
3.2.1	Moduł reprezentacji danych	7
3.2.2	Moduł persystencji	7
3.2.3	Moduł autoryzacji użytkownika	7
3.2.4	Moduły komunikacji z użytkownikiem	9
3.2.5	Moduł mailowy	12
3.3	Wymagania jakościowe	13
3.4	Funkcje systemu	13
3.5	Technologie	14
3.5.1	Wymagania systemowe i technologiczne	14
3.5.2	Frontend	14
3.5.3	Backend	15
3.5.4	Persystencja	15
3.5.5	Moduł autoryzacji	16
3.5.6	Moduł mailowy	16

1 Opis problemu

1.1 Definicja problemu

Dobrze znanym faktem jest, że wybór odpowiedniego prezentu może okazać się trudnym zadaniem. Wszyscy mieliśmy okazję obdarowywać, jak i być obdarowywanym. Zdarzają się sytuacje, kiedy prezent okazuje się nietrafiony. Ile razy nie popisaliśmy się znajomością potrzeb i zainteresowań drugiej osoby obdarowując ją czymś, co nie tylko jej się nie spodobało, ale może nawet sprawiło jej przykrość? A ile razy to nam zdarzyło się taki prezent dostać? Otwierasz prezent, a tam wełniane kapcie, perfumy, których zapachu nie znosisz, kolejny egzemplarz wielokrotnie przeczytanej książki? Istnieje również wiele rzadziej spotykanych prezentów, które zamiast okrzyku radości mogą wywołać jedynie cichy jęk zawodu.

Każdy z nas ma inny gust. Z biegiem czasu zmienia się nasze spojrzenie na świat i potrzeby. Czasami po prostu trudno jest trafić w gusta obdarowywanych.

Święta, ślub w rodzinie, urodziny bliskiej osoby to czas nerwowych poszukiwań wymarzonego prezentu. Zastanawiamy się jaki prezent sprawiłby największą radość i często uświadamiamy sobie, że nie mamy pojęcia co tak naprawdę ucieszyłoby osobę obdarowywaną. Często też w natłoku obowiązków po prostu brakuje nam czasu na wybór sensownego upominku.

Robiąc zakupy, musimy pamiętać, że polskie prawo co do zasady nie nakazuje sprzedawcy przyjęcia zwrotu towaru, jeśli został on kupiony w sklepie stacjonarnym. Takie uprawnienie przysługuje tylko i wyłącznie w sytuacji, kiedy zakup jest dokonany na odległość (np. przez internet lub telefon) albo poza lokalem przedsiębiorstwa. Co prawda wiele sklepów wychodzi naprzeciw konsumentom i proponuje wymianę towaru na inny, ewentualnie przyjęcie zwrotu w określonym przez sklep terminie, ale mimo wszystko sprawa wymiany niechcianego prezentu jest kwestią trudną. W jaki sposób powiedzieć to obdarowującemu tak, by go nie urazić? Czy niechciany prezent można sprzedać?

2 Wizja rozwiązania

Celem pracy jest opracowanie aplikacji umożliwiającej użytkownikom stworzenie listy wymarzonych prezentów oraz przeglądanie takich list innych użytkowników. Jej celem będzie wyeliminowanie sytuacji, kiedy osoba obdarowywana dostaje nietrafiony prezent. Program realizujący tę funkcjonalność nie miałby racji bytu tylko na komputerze stacjonarnym, z racji tego, że często podczas wizyty w sklepie bądź innych miejscach nie mielibyśmy możliwości dodania nowego przedmiotu do listy lub sprawdzenia listy innego użytkownika. Znakomicie sprawdzi się natomiast na telefonie komórkowym, który z założenia zawsze mamy przy sobie.

Zaprojektowana aplikacja powinna, oprócz zarządzania listami prezentów, dostarczać użytkownikowi prosty i przejrzysty interfejs, który umożliwi mu szybkie oraz wygodne posługiwanie się nią, na przykład w czasie zakupów.

Innym zadaniem aplikacji będzie autoryzacja użytkowników. Pozwoli ona na możliwość zarządzania swoim profilem i swoją listą prezentów wszystkim użytkownikom aplikacji.

2.1 Usługi

- **Tworzenie konta dla nowego użytkownika**

System będzie wymagał od użytkownika utworzenia konta. Do rejestracji będzie wymagany e-mail użytkownika, podanie unikalnego loginu oraz hasła do konta. Korzystanie z systemu będzie wymagało od użytkownika początkowego zalogowania się na swoje konto.

- **Uzyskanie nowego hasła dla istniejącego użytkownika**

System pozwoli na odzyskanie swojego konta w przypadku zapomnienia hasła przez użytkownika zarejestrowanego w systemie. Po wprowadzeniu adresu mailowego zostanie wysłana wiadomość e-mail z adresem do strony internetowej, na której będzie możliwe zresetowanie hasła.

- **Dostęp do konta użytkownika**

Konto użytkownika będzie musiało zawierać e-mail, login i hasło oraz możliwe dane dodatkowe takie jak imię i nazwisko. Po zalogowaniu dla użytkownika dostępne będą

informacje o dodanych znajomych oraz o swojej i znajomych listach wymarzonych prezentów.

- **Dodawanie, wyświetlanie, edycja, usuwanie prezentów**

System umożliwi zarządzanie prezentami. Każda utworzony prezent będzie musiał mieć przypisaną inną nazwę. Lista prezentów będzie widoczne jedynie dla znajomych użytkownika. Do listy użytkownik będzie mógł dodawać prezenty wraz z podstawowymi informacjami takimi jak nazwa prezentu oraz opis. Możliwa będzie też edycja i usunięcie wybranego prezentu.

- **Dodawanie, wyświetlanie, edycja, usuwanie szczegółowego opisu do prezentu**

Za pomocą systemu użytkownik będzie mógł dodać, zmienić lub usunąć opis prezentu. W opisie prezentu będzie można umieścić wszelkie dodatkowe szczegóły dotyczące prezentu.

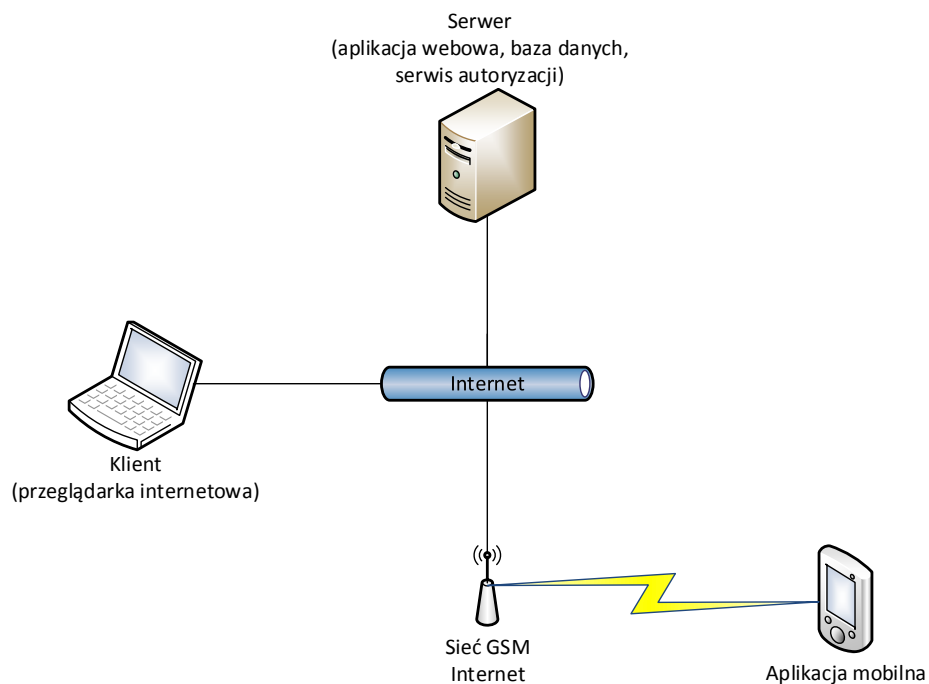
- **Dodawanie, wyświetlanie, usuwanie użytkowników z grona znajomych**

System umożliwi użytkownikom dodawanie znajomych, dla których mają być widoczne listy prezentów. System zapewni zabezpieczenie wglądu do listy prezentów przed osobami nieznanymi. Każdy użytkownik będzie mógł też usunąć osobę z grona znajomych, jeśli widnieje ona na jego liście znajomych.

3 Koncepcja systemu

3.1 Schemat systemu

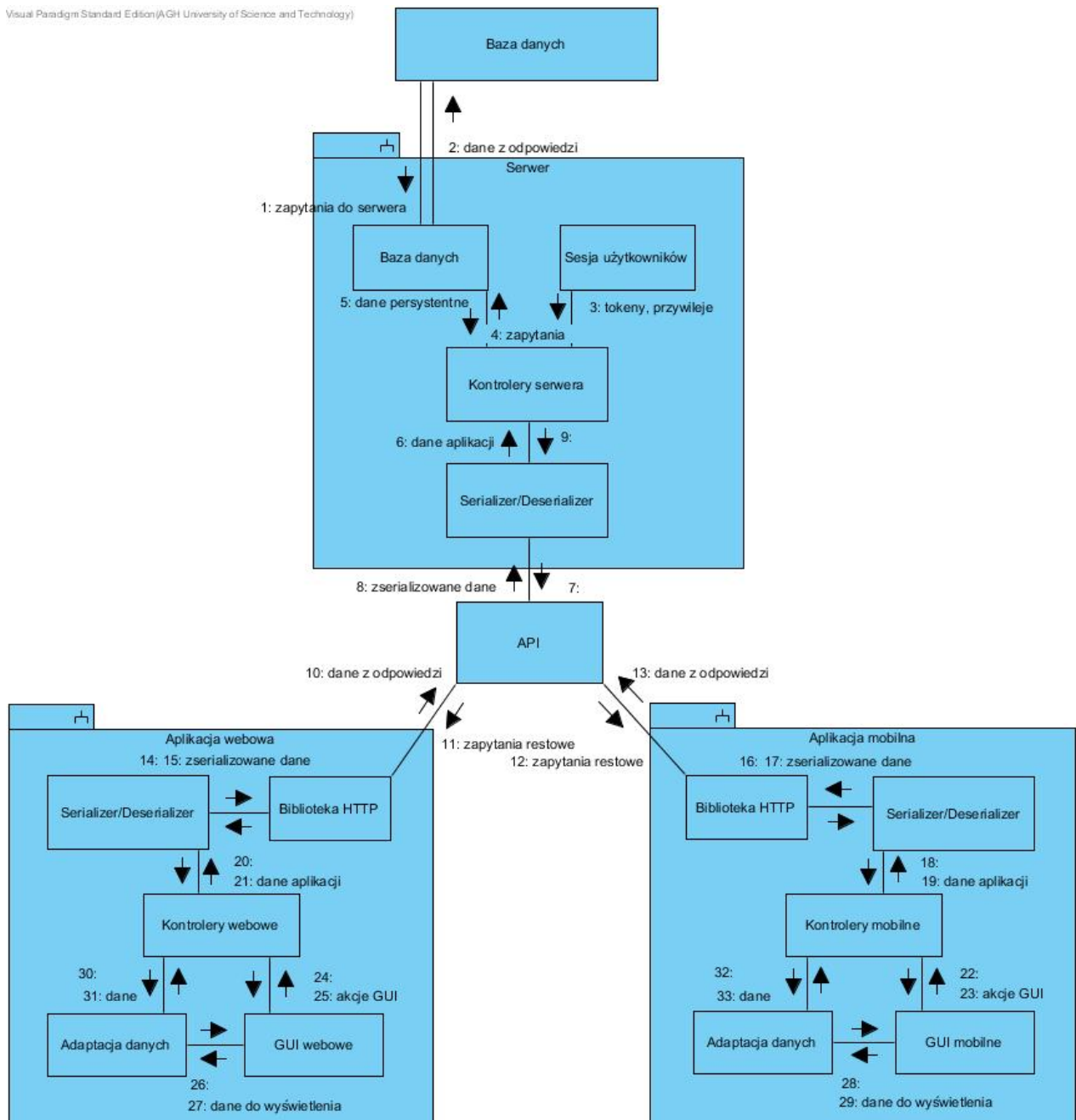
System będzie składał się z aplikacji na urządzenia mobilne (z systemem operacyjnym Android), aplikacji webowej dostępnej dla użytkownika z poziomu przeglądarki internetowej, a także z części serwerowej, na którą będzie składać się serwis autoryzacji użytkowników oraz baza danych przechowująca dane użytkowników, a w szczególności ułożone przez nich listy prezentów.



Rys 1. Schemat systemu

3.2 Moduły

Visual Paradigm Standard Edition (AGH University of Science and Technology)



Rys 2. Diagram komunikacji całego systemu

Sekwencja wywołań modułów systemu dla użytkownika

1. Użytkownik klika odpowiedni odnośnik w GUI przeglądarkowym lub mobilnym na telefonie z systemem Android i wysyła tym samym zapytanie do serwera aplikacji o udostępnienie mu danych wystawionych pod określonym URLeM.
2. Zapytanie przechodzi przez moduł serializera (jeżeli trzeba zserializować jakieś obiek-

ty np. na potrzebę HTTP POST).

3. Zapytania są odbierane przez serwer, przez moduł API. Serwer przydziela odpowiedni kontroler do obsługi żądania użytkownika.
4. Kontroler sprawdza przywileje użytkownika, w razie potrzeby prosi o autentykację i decyduje czy może mu zwrócić żądane dane. Opiera się to na komunikacji z modułem Sesji użytkownika.
5. Jeżeli użytkownik posiada odpowiednie przywileje, do bazy danych kierowane jest zapytanie o dane, które kontroler wysyła następnie użytkownikowi prezentując je w odpowiedni i przejrzysty sposób.

3.2.1 Moduł reprezentacji danych

Odpowiedzialny za reprezentację świata rzeczywistego w środowisku wirtualnym. Określa strukturę i zależności między poszczególnymi elementami.

3.2.2 Moduł persystencji

Służy do utrwalania oraz pobierania potrzebnych elementów z bazy danych.

3.2.3 Moduł autoryzacji użytkownika

- Moduł zakładania konta

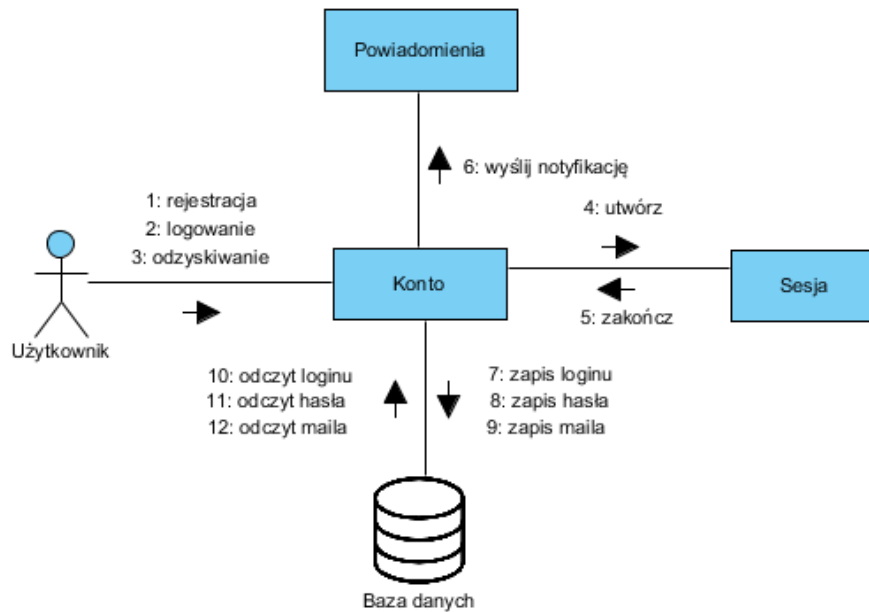
Dzięki niemu użytkownik będzie w stanie założyć konto uprawniające go do korzystania z aplikacji.

- Moduł logowania

Umożliwia identyfikację użytkownika za pomocą wybranego loginu oraz hasła

- Moduł odzyskiwania

W przypadku utraty lub zapomnienia hasła będzie możliwość odzyskania go przy pomocy adresu mailowego, dla którego utworzono konto.



Rys 3. Diagram komunikacji w module autoryzacji użytkownika

Sekwencja wywołań dla modułu autoryzacji użytkownika

1. Użytkownik w UI modułu zarządzania kontem może wybrać opcję rejestracji.
2. Użytkownik podaje dane w formularzu i klika odpowiedni odnośnik w GUI przeglądarkowym lub mobilnym na telefonie z systemem Android. Do serwera aplikacji zostaje tym samym wysłane prośba o autoryzację i zapytanie o udostępnienie mu wystawionych danych.
3. Informacje podane przez użytkownika zostają przekazane do Konta, które zapisuje je w bazie danych.
4. Użytkownik zarejestrowany może się zalogować za pomocą wpisania loginu i hasła.
5. Dane zostają wysłane do Konta, następnie następuje odczyt z bazy weryfikujący poprawność wprowadzonych danych. W przypadku poprawnych danych następuje przesłanie informacji do Sesji.
6. Zostaje utworzona nowa sesja, a w konsekwencji następuje przekierowanie użytkownika do strony głównej.
7. Jeśli użytkownik nie pamięta hasła może wnieść prośbę o przypomnienie hasła, która zostaje przesłana do konta.

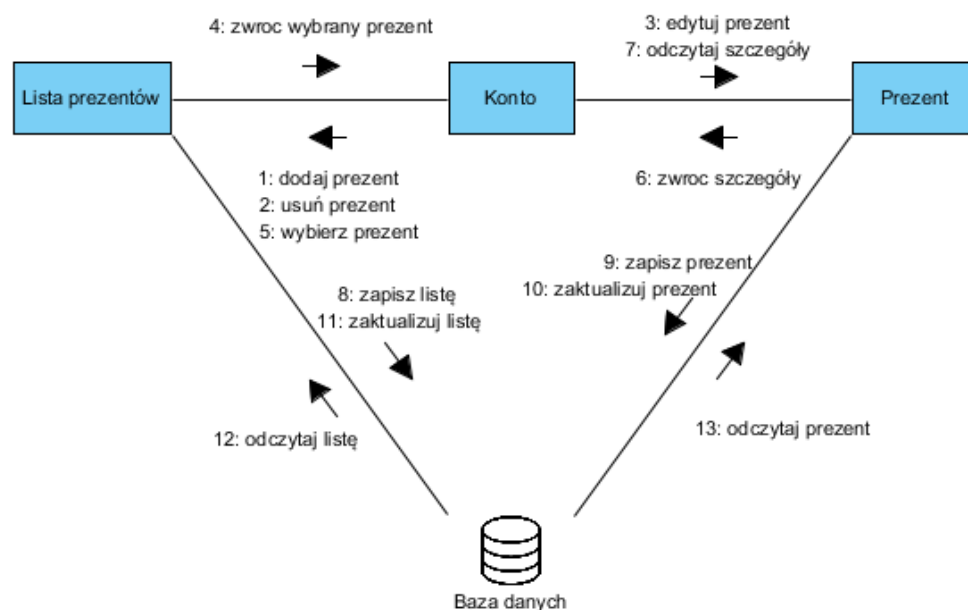
8. Zostaje wygenerowany i zapisany do bazy token zaszyfrowanego hasła, a następnie odczyt e-maila i hasła.
9. Na koniec Konto wysyła do modułu notyfikacji żądanie o wysłaniu e-maila na podany adres w celu ustanowienia nowego hasła.
10. Zalogowany użytkownik poprzez kliknięcie odpowiedniego przycisku w GUI może wylogować się z systemu.
11. Prośba o wylogowanie przekazywana jest do Konta, które następnie przekazuje do Sesji informacje o zakończeniu działań użytkownika.

3.2.4 Moduły komunikacji z użytkownikiem

3.2.4.1 Moduł webowy

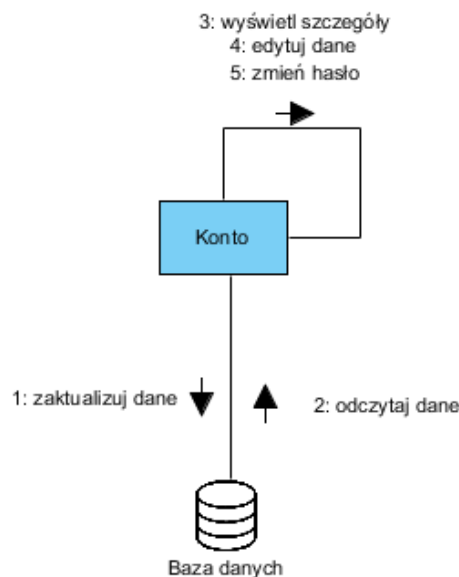
- Moduł zarządzania prezentami

Pozwala na wyświetlanie, dodawanie, edycję, usuwanie prezentów oraz list przypisanych do użytkownika. Ponadto, umożliwia wyświetlanie list prezentów swoich znajomych i przeglądanie ich.



Rys 4. Diagram komunikacji w module zarządzania prezentami
Sekuencja wywołań dla modułu zarządzania prezentami

1. Użytkownik posiada prawa do dodania, edycji bądź usunięcia prezentów.
 2. Po wykonaniu odpowiedniej akcji przez użytkownika w systemie następuje wysłanie informacji do Konta.
 3. Dane potrzebne do dodania lub edycji prezentu zapisywane są w bazie.
 4. Podczas próby usunięcia prezentu konieczne dane wyciągane są z bazy.
 5. Jeśli procedura dodawania, edycji lub usunięcia prezentu przebieganie poprawnie, użytkownik dostaje zwrotną wiadomość o pomyślnym przebiegu akcji.
 6. Jeśli akcja użytkownika zostanie ukończona pomyślnie następuje odświeżenie danych u użytkownika.
- Moduł zarządzania kontem użytkownika
- Umożliwia wyświetlenie oraz zmianę danych powiązanych z kontem użytkownika.

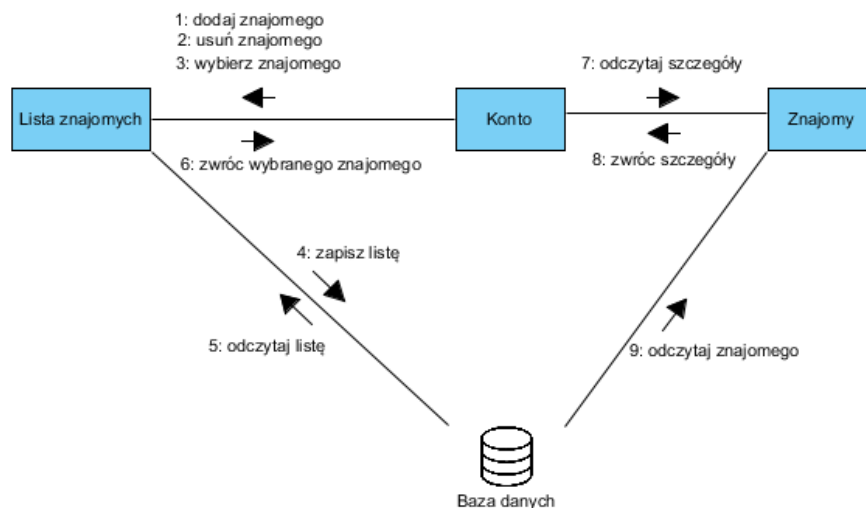


Rys 5. Diagram komunikacji w module zarządzania kontem użytkownika

Sekwencja wywołań dla modułu zarządzania kontem użytkownika

1. Użytkownik w UI modułu zarządzania kontem może wyświetlić szczegóły konta oraz edytować dane.
2. Do serwera aplikacji zostaje wysłane żądanie z prośbą o prezentację szczegółowych danych lub zmianę danych użytkownika.

3. W przypadku edycji konta informacje podane przez użytkownika zostają zweryfikowane pod względem poprawności a następnie przekazane do Konta, które zapisuje je w bazie danych.
 4. Jeśli użytkownik chce wyświetlić szczegóły swojego konta może wnieść prośbę, która zostaje przesłana do konta.
 5. Pobrane zostają dane z bazy i w czytelny i przejrzysty sposób zaprezentowane użytkownikowi.
- Moduł zarządzania znajomymi
- Wyświetla listę znajomych oraz zezwala na dodanie nowych użytkowników do swojej listy znajomych.



Rys 6. Diagram komunikacji w module zarządzania znajomymi

Sekwencja wywołań dla modułu zarządzania znajomymi

1. Użytkownik może dodawać i usuwać innych znajomych ze swojego grona znajomych.
2. Poprzez wybranie odpowiedniej opcji w aplikacji użytkownik wysyła żądanie do Konta.
3. W przypadku dodawania użytkownika do listy znajomych konieczne dane zapisywane są w bazie.
4. Podczas wykonania akcji usuwania użytkownika z listy znajomych informacje o nim pobierane są z bazy.

5. Jeśli procedura dodawania lub usunięcia znajomego przebiegnie poprawnie, użytkownik dostaje zwrotną wiadomość o pomyślnym przebiegu akcji.
6. Jeśli akcja użytkownika zostanie ukończona pomyślnie następuje odświeżenie danych u użytkownika.
7. Użytkownik ma również prawo do wyświetlenia szczegółów dotyczących swojego znajomego.
8. Aby mieć dostęp do jego publicznych danych klika odnośnik w GUI i Konto wysyła zapytanie do bazy o zwrócenie danych dotyczących znajomego.
9. Następuje wyświetlenie danych użytkownikowi.

- Moduł powiadomień

Odpowiada za informowanie poszczególnych modułów o powstałych zdarzeniach.

- Moduł iteracji

Dzięki niemu mamy możliwość wykonywania tych samych bloków instrukcji na grupie elementów.

- Moduł wykonawczy

Realizuje określone czynności w zależności od dostarczonego bytu.

3.2.4.2 Moduł mobilny (funkcjonalność poszczególnych podmodułów taka sama jak w module webowym)

- Moduł zarządzania prezentami
- Moduł zarządzania kontem użytkownika
- Moduł zarządzania znajomymi
- Moduł powiadomień
- Moduł wykonawczy

3.2.5 Moduł mailowy

Zapewnia możliwość wysyłania powiadomień dotyczących m. in. utworzenia konta, czy resetowania hasła na adres mailowy zarejestrowany przez użytkownika.

3.3 Wymagania jakościowe

- *Niezawodność, dostępność:* Moduł powinien być dostępny w każdy dzień tygodnia 24h na dobę i działać na każdym popularnym systemie operacyjnym.
- *Wydajność:* System powinien być na tyle wydajny, aby umożliwić użytkownikom swobodne korzystanie z aplikacji. Synchronizacja danych z serwerem powinna przebiegać płynnie dla wszystkich użytkowników podłączonych jednocześnie do systemu.
- *Zajętość pamięci oraz zasobów:* Dane buforowane na urządzeniach mobilnych nie powinny utrudniać pracy (są to dane tekstowe) przez zajęcie zbyt dużej ilości miejsca. Natomiast baza danych na serwerze ma za zadanie przechowywać dane dotyczące wszystkich użytkowników systemu.
- *Intuicyjność:* Użytkowanie systemu powinno być intuicyjne i przejrzyste dla użytkownika. Aplikacja włączania jest na życzenie użytkownika.
- *Zrozumiałość:* Konstrukcja systemu powinna być wykonana w ten sposób, że po przeczytaniu instrukcji, każdy użytkownik nie miał problemów w jego skonfigurowaniu i późniejszej obsłudze. Główną ideą systemu ma być jego prostota obsługi.

3.4 Funkcje systemu

- Tworzenie konta
- Logowanie użytkownika
- Odzyskiwanie hasła
- Zmiana hasła
- Dodawanie prezentu do listy
- Usuwanie prezentu z listy
- Edytowanie wybranego prezentu
- Wyświetlanie szczegółowego opisu prezentu
- Dodawanie znajomego do listy znajomych

- Usuwanie znajomego z listy znajomych
- Wysyłanie powiadomień na adres mailowy użytkownika
- Wyświetlanie szczegółów konta użytkownika
- Wyświetlanie informacji o znajomym
- Wyświetlanie listy prezentów znajomego

3.5 Technologie

3.5.1 Wymagania systemowe i technologiczne

- **Serwer z dostępem do sieci**

Ze względu na wielką powszechność i dostępność sieci Internet serwer udostępniający usługi innym aplikacjom powinien mieć połączenie z siecią. Zapewni to doskonałą skalowalność naszego rozwiązania oraz pozwoli na dostęp do danych z prawie każdego miejsca na świecie.

- **Implementacja w języku Java**

Obiektowy język programowania, cechujący się niezależnością od architektury, niezawodnością i bezpieczeństwem.

- **MongoDB** - nierelacyjny system zarządzania bazą danych, wspierający wiele typów danych, obsługujący kursory, indeksowanie oraz posiadający wsparcie dla metod pobierających dane.

3.5.2 Frontend

- **Bootstrap**

Do stworzenia reaktywnego interfejsu użytkownika zostanie zastosowana technologia Bootstrap. Bootstrap dostarcza zbioru szablonów formularzy, przycisków, czcionek i innych komponentów wyświetlanych na stronach WWW. Jest napisany głównie przy użyciu języka HTML i CSS, ale używa też języka JavaScript, dzięki czemu można rozszerzać funkcjonalność komponentów. Komponenty udostępniane przez tę technologię pozwolą na dostarczenie funkcjonalności w stabilny sposób.

- **CSS**

W celu odseparowania struktury dokumentu od formy jego prezentacji użyjemy kaskadowych arkuszy stylów CSS. Dokumenty pisane z wykorzystaniem arkuszy stylów są zwykle bardziej przejrzyste i krótsze. Style pozwalają w łatwy i wygodny sposób zarządzać całą serią dokumentów, poprzez stosowanie zewnętrznych arkuszy stylów.

- **Angular**

Do stworzenia aplikacji webowej wybraliśmy framework AngularJS. Pozwala on na rozszerzenie możliwości zwykłego HTML o nowe funkcje, które umożliwiają utrzymanie ciągłej synchronizacji danych między widokiem a modelem. Dzięki niemu można szybko zbudować aplikacje typu single page.

3.5.3 Backend

- **Aplikacja webowa**

Aplikacja webowa zostanie napisana przy użyciu języka Java. Wykorzystanie tego popularnego języka gwarantuje, że aplikacja będzie działać i wyglądać tak samo na różnych urządzeniach. Java posiada ogromną ilość gotowych bibliotek, które przyspieszą tworzenie aplikacji. Świetnie przemyślane i zaprojektowane biblioteki oraz narzędzia ułatwiają programistom wygodne tworzenie interfejsów graficznych aplikacji.

- **Aplikacja mobilna**

Aplikacja przeznaczona jest do wykorzystania na telefonach komórkowych typu smartphone z zainstalowanym mobilnym systemem operacyjnym Android. Technologia Android została wybrana ze względu na jej popularność oraz bogatą gamę narzędzi programistycznych.

3.5.4 Persystencja

Z uwagi na to, że aplikacja serwerowa jest niewielka, a jej model jest nieco ubogi (nie mamy skomplikowanych relacji pomiędzy encjami) zdecydowaliśmy użyć dokumentowej bazy danych MongoDB. Pomimo istnienia w projekcie kilku encji całość persystencji została zamknięta w trzech kolekcjach dokumentów - 'users', 'userDetails' oraz 'gifts'.

3.5.5 Moduł autoryzacji

Do wdrożenia mechanizmów uwierzytelniania i autoryzacji w naszym projekcie wybraliśmy bibliotekę Spring Security. Spring Security pozwala na bardzo precyzyjne i wygodne zarządzanie bezpieczeństwem aplikacji ze względu na bardzo elastyczną konfigurację.

3.5.6 Moduł mailowy

Moduł mailowy zostanie napisany w języku Java. Przemawia za tym wysoka stabilność serwerów, które pozwolą na zaawansowaną komunikację z klientami w późniejszej fazie rozwoju produktu.