**Politechnika Krakowska  
im. Tadeusza Kościuszki**

Wydział Fizyki, Matematyki i Informatyki

**Kraków 2018**

**Kamil Wanat**

Numer albumu: 114021

**Aplikacja mobilna wspomagająca zarządzanie małym przedsiębiorstwem**

**Small business management – mobile application**

**Praca inżynierska   
 na kierunku Informatyka**

Praca wykonana pod kierunkiem:  
**dr hab. Zbisława Tabora**

**Uzgodniona ocena:**......................................

...........................................................................podpisy promotora i recenzenta





**Spis treści**

[**1.** **Wstęp** 3](#_Toc502934051)

[**1.1** **Temat i cel pracy** 3](#_Toc502934052)

[**1.2** **Zakres pracy** 3](#_Toc502934053)

[**2.** **Część teoretyczna** 4](#_Toc502934054)

[**2.1** **Przegląd podobnych rozwiązań** 4](#_Toc502934055)

[**2.2** **Opis użytych technologii** 7](#_Toc502934056)

[**2.3** **Opis użytych wzorców projektowych** 10](#_Toc502934057)

[**2.4** **Dostęp do bazy danych** 13](#_Toc502934058)

[**2.5** **Mapowanie relacyjno-obiektowe** 13](#_Toc502934059)

1. **Wstęp**
   1. **Temat i cel pracy**

Tematem mojej pracy inżynierskiej jest aplikacja mobilna wspomagająca zarządzanie małym przedsiębiorstwem. Program nazwałem MMA – Mobile Management Assistant. Do podjęcia tego tematu skłoniła mnie praca w jednym z Krakowskich przedsiębiorstw. Liczba pracowników zatrudnionych w tej działalności oscyluje w okolicach 20 osób. Jest więc to małe przedsiębiorstwo zarządzane bezpośrednio przez właściciela. Podczas pracy w tej firmie zauważyłem pewne trudności jakie napotykali zarówno pracownicy, jak i szefostwo. Każdy z pracowników musi przesyłać do pracodawcy swoją „dostępność” na kolejny miesiąc. Rozpisywane w niej są wszystkie dni miesiąca, wraz z godzinami w których pracownik może się pojawić. Następnie właściciel ustala grafiki na kolejne tygodnie, w których zawiera pracowników oraz ich godziny pracy. Cały proces jednak był chaotyczny. „Dostępności” napływały poprzez różne media, od wiadomości sms, poprzez komunikatory internetowe, aż po wiadomości e-mail oraz wydrukowane tabelki. Postanowiłem usystematyzować proces tworzenia grafików, tak aby mogły być one przechowywane w jednolitej postaci, oraz mogłyby być dostępne w jednym miejscu.

Dodatkowo chciałem usprawnić proces zarządzania magazynem prowadzonym w

przedsiębiorstwie. Nader często występowała sytuacja, gdzie brakowało pewnych artykułów niezbędnych do działania firmy. Dzięki aplikacji możliwe będzie kontrolowanie stanu magazynu.

Aplikacja posiada również inne, mniej istotne moduły takie jak wnioski urlopowe, bądź sekcję news, które zostaną omówione w dalszej części pracy.

* 1. **Zakres pracy**

Zakres niniejszej pracy obejmuje między innymi skontaktowanie się z właścicielem

wybranego przedsiębiorstwa w celu zebrania informacji dotyczących funkcjonowania firmy oraz potrzeb danego pracodawcy. Kolejnym krokiem było zaprojektowanie aplikacji, wybór odpowiednich technologii, oraz dalsze konsultacje z właścicielem przedsiębiorstwa. Dzięki współpracy z potencjalnym klientem aplikacji, mogła ona zostać dopasowana do potrzeb małych przedsiębiorstw. Następnie aplikacja była implementowana oraz testowana.

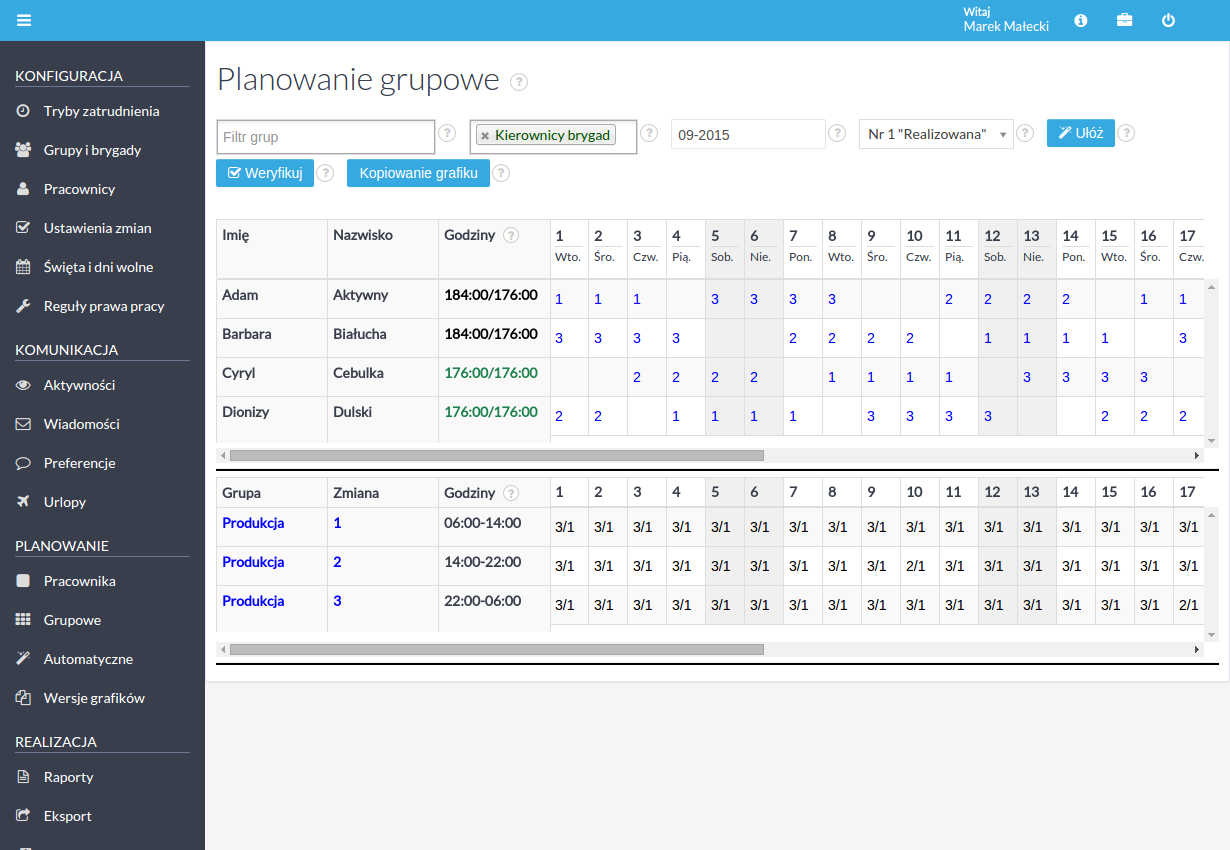
1. **Część teoretyczna**
   1. **Przegląd podobnych rozwiązań**

Na rynku istnieje obecnie niezliczona ilość aplikacji wspomagających zarządzanie nie tylko małymi przedsiębiorstwami. Z łatwością znaleźć można także programy pozwalające na zarządzanie przedsiębiorstwami średnimi oraz systemy które ułatwiają prowadzenie dużych korporacji. Skupiając się jedynie na obszarze w którym znajduję się moja praca inżynierska użyć możemy kilkunastu programów pozwalających na efektywną administrację działalnością gospodarczą. Są to zarówno programy desktopowe, aplikacje mobilne dla różnych systemów mobilnych, a także serwisy internetowe.

Najpopularniejszym obecnie systemem do tworzenia grafików jest serwis sling.is. Pozwala on na łatwe przesyłanie dostępności oraz tworzenie grafików. Jego niewątpliwą zaletą jest wersja mobilna dzięki której system sling możemy mieć zawsze przy sobie, zainstalowany na swoim telefonie.

Kolejnych narzędziem umożliwiającym tworzenie grafików oraz wniosków urlopowych jest program „Grafik Urlopowy”. Za jego pomocą pracodawca może utworzyć plan pracy oraz urlopów na cały rok. Program pozwala na wygenerowanie list pracowniczych oraz ich rozesłanie do pracowników w celu ich sprawdzenia, jednak w celu jednoczesnej pracy kilku użytkowników, jedna z maszyn musi działać w trybie serwera, aby możliwe było połączenie się z bazą danych.

Ostatnim omawianym przeze mnie systemem wspomagającym tworzenie grafików jest serwis „ePlanist” pozwalający na tworzenie grafików w oparciu o preferencje urlopowe pracowników, reguły prawa pracy oraz własne, zdefiniowane przez właściciela reguły. Program pozwala na zautomatyzowanie procesu tworzenia grafików poprzez wbudowane w aplikację algorytmy. Po skończonym procesie pracodawca ma możliwość edycji grafiku.



Rys1. Przykładowy zrzut programu ePlanist.

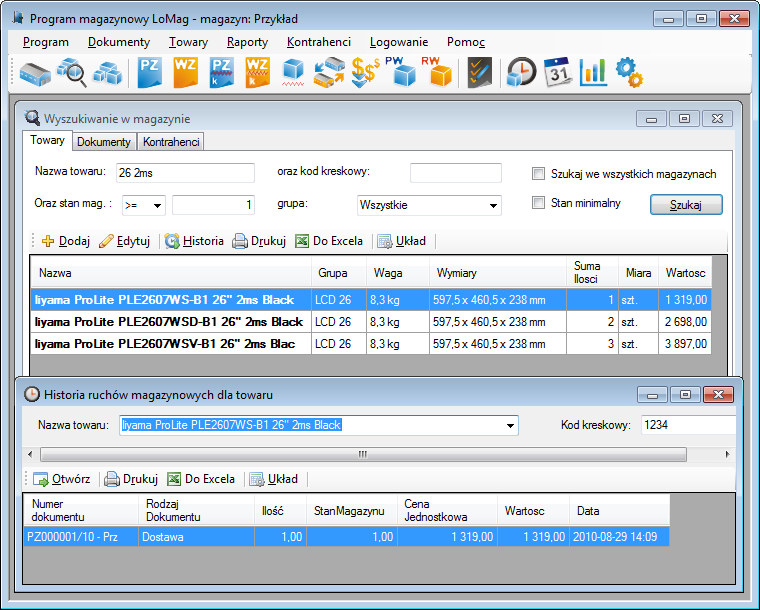
Kolejnymi omawianymi przeze mnie programami są programy pozwalające na prostą obsługę magazynu. Dzięki takim rozwiązaniom przedsiębiorcy mogą w prosty sposób zarządzać swoimi magazynami oraz kontrolować ich stan. Podobnie jak w przypadku systemów ułatwiających tworzenie i przechowywanie grafików dla pracowników, tak i w przypadku programów zarządzających magazynem wybór rozwiązań jest bardzo duży. Na rynku znaleźć można proste darmowe programy pozwalające na zarządzanie małymi magazynami, a także duże komercyjne systemy umożliwiające zarządzanie dużymi magazynami. Omówię teraz kilka rozwiązań.

Weaver WMS to darmowe rozwiązanie przeznaczone dla przedsiębiorców którzy prowadzą średniej wielkości magazyn. Program pozwala na łatwe zarządzanie stanem produktów, wystawianie faktur, tworzenie zamówień od dostawców, a także przygotowywanie list zakupowych dla klientów. System posiada także funkcję planowania magazynu, dzięki czemu użytkownik może w prosty sposób dowiedzieć się w którym miejscu magazynu należy szukać danego produktu. Ponadto oprogramowanie automatycznie tworzy raporty podsumowujące które towary cieszą się największym zainteresowaniem, a które z nich rotują najsłabiej.

Darmowe rozwiązanie ifirma.pl dostarczane jest przez spółkę IFIRMA SA która

tworzy pakiety oprogramowania przeznaczone do zarządzania firmą. Serwis ifirma.pl pozwala na prowadzenie księgowości internetowej, biura rachunkowego, magazynu, a także wystawianie faktur. Część z funkcji jest dodatkowo płatna jednak wykorzystanie samego programu magazynowego jest darmowe. Serwis internetowy do zarządzania magazynem pozwala między innymi na import magazynu z pliku CSV, łatwe zarządzanie stanem magazynu. W ramach jednego konta w systemie firma możliwe jest utworzenie kilku magazynów. Niewątpliwą zaletą jest możliwość zarządzania magazynem z każdego miejsca, ze względu na to, iż jest to serwis internetowy. Jednakże w sytuacji w której brakuje dostępu do sieci Internet, dostęp do zasobów jest niemożliwy.

Ostatnim omówionym tutaj rozwiązaniem jest komercyjny system LoMag dostarczany przez firmę LONGINT. Program został utworzony z wykorzystaniem bibliotek .NET Framework. Baza danych magazynu oparta jest o technologię Microsoft SQL Server Expres. LoMag to program pozwalający na zarządzie magazynem w małym lub średnim przedsiębiorstwie. Dzięki komercyjnemu charakterowi programu, udostępnia on użytkownikowi wiele przydatnych funkcji, ułatwiających kontrolowanie zasobów przedsiębiorstwa. Użytkownik systemu może za jego pomocą tworzyć najważniejsze dokumenty magazynowe takie jak: przyjęcie magazynowy, wydanie z magazynu, dokumenty rozchodowe i przychodowe, podsumowania, raporty a także dokumenty remanentowe. Aplikacja pozwala na wielostanowiskowy dostęp do magazynu a także na tworzenie i zarządzanie wieloma magazynami w ramach jednej aplikacji. Niezawodnym atutem aplikacji jest możliwość tworzenia i drukowania raportów dobowych oraz innych dokumentów bezpośrednio z programu LoMag.



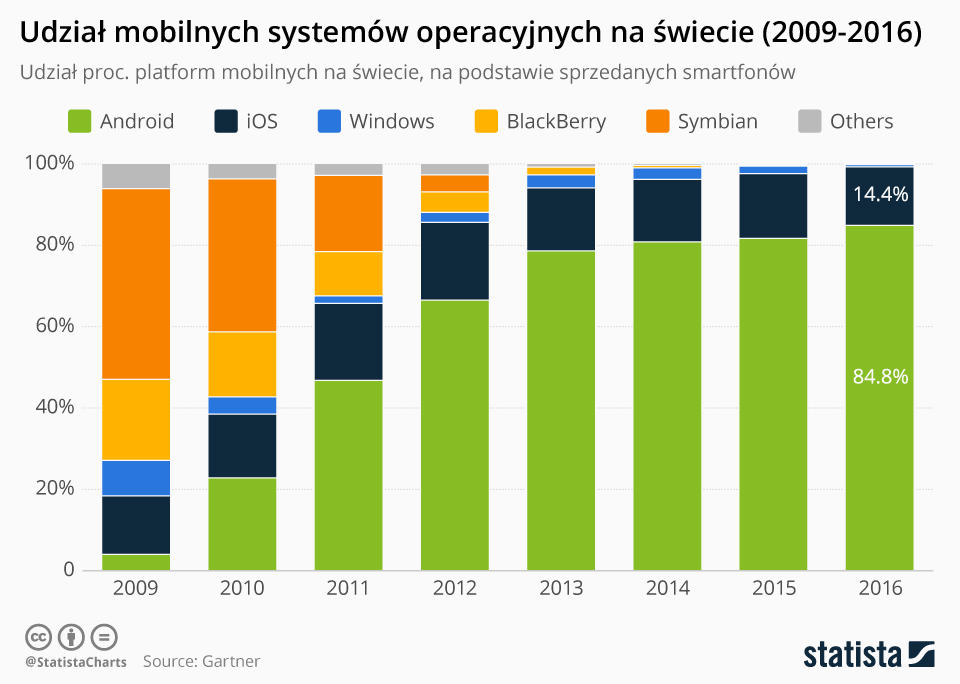
Rys 2. Przykładowy zrzut ekranu systemu LoMag.

* 1. **Opis użytych technologii**

W swojej pracy inżynierskiej wykorzystałem kilka technologii które pozwoliły mi na efektywne zaprojektowanie oraz zaimplementowanie aplikacji. Dzięki wybranym technologiom projekt został poprawnie zaimplementowany, a podczas jego tworzenia możliwy był nadzór nad kolejnymi jego wersjami, dzięki zastosowaniu systemu kontroli wersji.

Android to system operacyjny przeznaczony dla urządzeń mobilnych tj. telefonów

komórkowych, tabletów, smart fonów. Android oparty jest o jądro systemu Linux. Początkowo rozwijany przez firmę Android Inc, obecnie utrzymywany przez Open Handset Alliance. Android od roku 2013 jest najpopularniejszym systemem operacyjnym na urządzenia mobilne, deklasując przy tym system iOS czy Windows Mobile. Prostota tworzenia aplikacji współpracujących z systemem Android sprawiła, że obecnie zrzesza on wokół siebie bardzo dużą społeczność deweloperów. Dzięki temu Android może zostać obecnie rozszerzony przez miliony aplikacji przygotowanych dla użytkowników zarówno przez firmy deweloperskie, jak i samodzielnych deweloperów. Użytkownicy chętnie sięgają po urządzenia z systemem Android ze względu na intuicyjny interfejs użytkownika, stabilne działanie oraz wspomnianą wcześniej liczbę aplikacji rozszerzających działanie systemu. W niniejszej pracy zdecydowałem się na wykorzystanie systemu Android ze względu na jego wielką popularność. Obecnie większość z nas ma dostęp do co najmniej jednego urządzenia z systemem Android. Dzięki temu każdy z pracowników będzie mógł w łatwy sposób korzystać z aplikacji przeznaczonej dla firmy.



Rys 3. Udział mobilnych systemów operacyjnych na świecie.

Java to nowoczesny, obiektowy język programowania stworzony przez firmę

Sun Microsystems. Java została stworzona w latach 90 pod kierunkiem Jamesa Goslinga pod roboczą nazwą OAK. Składnia języka została zaczerpnięta z języków programowania takich jak: C, C++. W roku 1995 została wydana pierwsza publiczna implementacja języka pod znaną nam dziś nazwą Java, wraz z obietnicą, że raz napisany program będzie mógł być uruchomiony na wielu platformach. Możliwe to było dzięki zastosowaniu kompilacji do kodu bajtowego a następnie jego uruchomieniu przez maszynę wirtualną Javy. W 2007 roku została opublikowana całość kodu maszyny wirtualnej Javy na licencji GNU General Public License. Trzy lata później firma Oracle Corporation ogłosiła dalszy rozwój Javy. W pracy inżynierskiej zdecydowałem się na wykorzystanie Javy, ze względu na jej dużą popularność wśród developerów. Posiada on ogromną społeczność programistów wspierającą rozwój języka. Dzięki temu wciąż poszerza się zbiór dostępnych bibliotek rozszerzających funkcjonalność Javy.[1] Jest ona także obecnie najpopularniejszym językiem programowania wykorzystywanym w implementacji programów na systemy Android.

GIT to rozproszony system kontroli wersji. Utworzony został w 2005 roku przez

Linusa Torvaldsa. Git początkowo został wykorzystany przy rozwoju jądra systemu Linux. Git pozwala każdemu użytkownikowi na utrzymywanie lokalnej, pełnej kopii repozytorium oraz pracę na niej. Dzięki temu każdy z użytkowników może pracować z projektem bez wymaganego dostępu do sieci Internet. Po zakończeniu pracy zmiany dokonane na lokalnym repozytorium mogą zostać wysłane na repozytorium zdalne. Dodatkowo git pozwala na tworzenie gałęzi, dzięki którym możliwe jest testowanie pewnych funkcjonalności poza główną gałęzią. Mechanizmy spójności danych w jakie wyposażony jest system Git, pozwalają na bezpieczne jego wykorzystywanie bez ryzyka utraty danych, czy ich zmiany bez reakcji ze strony Git. Do wyznaczania sum kontrolnych Git wykorzystuje funkcję skrótu SHA-1. Git traktuje dane jak zestaw migawek. Po każdym zatwierdzeniu zmian tworzony jest obraz pokazujący jak wyglądają wszystkie pliki a następnie zapisuje referencję do tych obiektów. W celu optymalizacji wydajności git nie zapisuje ponownie pliku które nie został zmieniony, a jedynie referencję do jego wersji poprzedniej. W pracy inżynierskiej został wykorzystany system kontroli wersji Git ze względu na jego bezpieczeństwo. W każdej chwili możliwy jest powrót do poprzednich wersji. Dzięki prostej i intuicyjnej obsłudze system Git jest obecnie jednym z najpopularniejszych systemów kontroli wersji. Niewątpliwą jego zaletą jest także możliwość pracy na lokalnej kopii projektu, a następnie jej wysłanie na repozytorium zdalne.

Rys 4. Schemat działania systemu Git.



MySQL to system służący do zarządzania relacyjnymi bazami danych. Model

relacyjnej bazy danych powstał w roku 1970, a jego twórcą był Frank Edgar Codd. Model relacyjnych baz danych oparty został o tabele – podstawowe struktury przechowujące dane. Pomiędzy tabelami mogą zachodzić pewne powiązania, dzięki którym cała baza danych jest spójna. Wyróżniane są trzy fundamentalne związki pomiędzy tabelami: jeden do jeden, jeden do wielu i wiele do wielu. System MySQL dostępny jest obecnie dla większości systemów i różnych architektur sprzętowych. W swojej pracy wykorzystałem bazę danych MySQL ze względu na jej dużą popularność oraz relacyjny charakter. Taka baza danych dobrze współpracuje z programami napisanymi w języku Java, a szybkość jej działania pozwala na lepszy odbiór aplikacji przez jej użytkowników.

Android Studio to środowisko programistyczne pozwalające na tworzenie aplikacji w

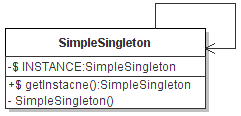
języku Java dedykowanych dla urządzeń mobilnych z systemem Android. Pierwsza wersja środowiska została opublikowana w 2013 roku na konferencji Google. Android Studio oparte zostało na systemie IntelliJ. Środowisko do sprawnego funkcjonowania wymaga jedynie bibliotek JDK. Zaimplementowany został tutaj także menager dzięki któremu użytkownicy mogą zarządzać zainstalowanymi składnikami systemu. W niniejszej pracy zdecydowałem się na wykorzystanie środowiska Android Studio, ze względu na jego stabilną pracę, oraz niezwykle bogatą funkcjonalność. Program udostępnia deweloperom wielu przydatnych narzędzi dzięki którym implementacja aplikacji na system Android staje się o wiele prostsza. Narzędzia te ułatwiają tworzenie kolejnych elementów aplikacji, a graficzne środowisko projektowania pozwala na intuicyjne tworzenie layoutów, co nierzadko jest dużym ułatwieniem dla programistów.

* 1. **Opis użytych wzorców projektowych**

Wzorce projektowe to opisy pewnych problemów często pojawiających się podczas

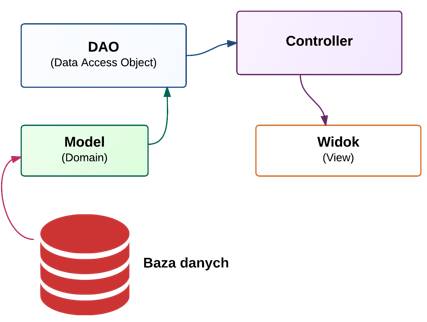
projektowania oraz implementacji oprogramowania, a także sposoby ich rozwiązania. Za pomocą wzorców projektowych programista może w prosty sposób zaimplementować pewne rozwiązanie, które będzie przejrzyste i czytelne dla niego samego oraz późniejszych developerów rozwijających dane oprogramowanie. Przy pomocy wzorców projektowych nazywamy standardowe struktury w projekcie oraz tworzymy ich abstrakcję. Staramy się aby dana struktura była przydatna w tworzeniu projektów i nadawała się do wielokrotnego użycia. We wzorcu projektowym określone są klasy oraz ich role i współdziałanie z innymi klasami, a także podział zadań. Obecnie wzorce projektowe są niezwykle ważne w projektach developerskich, ze względu na ich uniwersalność, oraz …………….

Wzorzec projektowy Singleton to jeden z najprostszych wzorców projektowych. Gwarantuje on, że klasa będzie posiadać tylko jeden egzemplarz, i zapewnia globalny dostęp do niego. Wzorzec projektowy Singleton pozwala na zapewnienie kontroli dostępu do swojego jedynego egzemplarza poprzez kapsułkowanie instancji swojej własnej klasy, a także pozwala na zmniejszenie zaśmiecenia przestrzeni nazw przez zmienne globalne przechowujące swoje jedyne instancje. Dodatkowym atutem wzorca jest fakt, iż instancja klasy jest tworzona dopiero przy pierwszym pobraniu instancji Singletona. Dzięki temu instancja tworzona jest dopiero w momencie w którym jest potrzebna. Jeśli nie zajdzie taka potrzeba, instancja klasy Singleton nigdy nie zostanie utworzona.



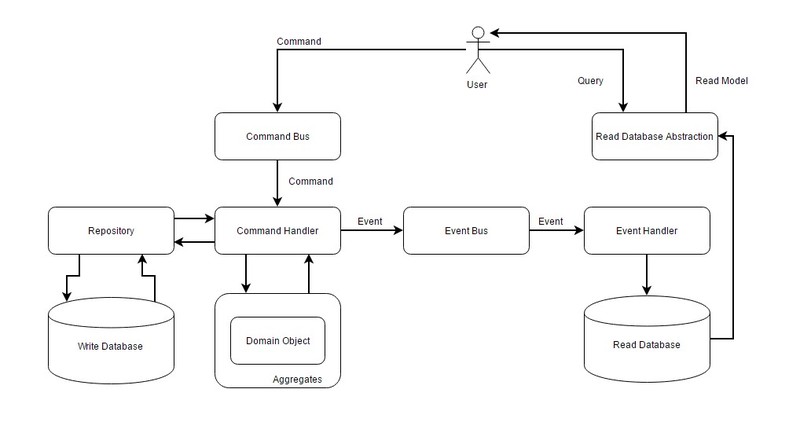
Rys. 5. Struktura wzorca Singleton

Wzorzec projektowy DAO (Data Access Object) pozwala programiście na zapewnienie jednolitego dostępu do różnych źródeł danych. Dzięki DAO możliwe jest uniezależnienie warstwy logiki biznesowej od warstwy dostępu do danych. Wzorzec ten bardzo często wykorzystywany jest przy tworzeniu aplikacji które zapewniają użytkownikowi dostęp do bazy danych. Dzięki DAO aplikacja nie musi znać miejsca składowania swoich danych, a jego zmiana nie pociąga za sobą zmiany kodu źródłowego logiki biznesowej. Wykorzystanie wzorca prowadzi do utworzenia nowej warstwy w architekturze systemu, jednak w aplikacjach, w których wysoka wydajności nie jest krytycznym czynnikiem, DAO może zostać z powodzeniem zaimplementowane, co znacznie uprości pracę przy implementacji oraz późniejszej rozbudowie systemu.



Rys. 6. Sposób wykorzystania DAO w aplikacji.

Kolejnym wzorcem wykorzystanym przeze mnie do zaimplementowania tej aplikacji był CQRS (Command Query Responsibility Separation). Zanim jednak opiszę ten wzorzec chciałbym wspomnieć kilka słów o koncepcie z jakiego się on wywodzi. W roku 1986 Bertrand Mayer przedstawił koncept CQS (Command Query Separation). Zakładał on, że każda metoda w programie musi być zakwalifikowana do jednej z dwóch grup: Command (metody zmieniające coś w systemie, ale nie zwracające nic), oraz Query (metody nie zmieniające systemu, ale zwracające pewne dane). Blisko 20 lat później przedstawiony został następca CQS, czyli CQRS. Wzorzec ten niejako zmusza programistę do rozdzielenia komend i zapytań nie tylko na różne metody, ale także na różne obiekty. Dzięki temu w projekcie powstają obiekty służące do zapisu pewnych informacji, oraz obiekty dzięki którym informacje te są odczytywane. Oczywiście przy takim podejściu ilość kodu jest zwiększona, jednak jego zastosowanie pozwala nam na widoczne rozdzielenie operacji zapisu od odczytu.



Rys. 7. Schemat wykorzystania wzorca CQRS.

* 1. **Dostęp do bazy danych**

W aplikacji MMA-Mobile Management Assistant dostęp do bazy danych

zrealizowany został przy wykorzystaniu wbudowanego w język Java interfejsu programowania JDBC (Java DataBase Connectivity). Interfejs umożliwia aplikacjom napisanym w języku Java (niezależnym od platformy) na porozumiewanie się z lokalnymi a także zdalnymi bazami danych z wykorzystaniem języka SQL. Jest on odpowiednikiem standardu ODBC (Open DataBase Connectivity), niezależnego od języka interfejsu stworzonego przez SQL Access Group. Aby wykorzystać interfejs JDBC w projekcie na system Android konieczne jest pobranie biblioteki JDBC oraz dołączenie jej do projektu.

* 1. **Mapowanie obiektowo-relacyjne**

Mapowanie obiektowo-relacyjne to nowoczesne podejście do zagadnienia współpracy

aplikacji z bazą danych. Problemem jest tutaj sposób komunikacji oprogramowania które bazuje na obiektach z relacyjną bazą danych, która nadal jest najczęściej wykorzystywaną bazą danych pomimo ciągle pojawiających się nowych rozwiązań. Do tej pory nie powstała akceptowana przez wszystkich koncepcja obiektowej bazy danych, choć oczywiście pojawiają różne jej wersje takie jak repozytoria XML czy próba rozszerzania języków programowania w celu przechowywania zserializowanych obiektów. Jednak wciąż najczęściej, oraz najchętniej wykorzystywane są relacyjne bazy danych ze względu na ich szybkość, stabilność oraz zapewniony poziom bezpieczeństwa. Przy wykorzystaniu relacyjnych baz danych pojawia się jednak pewien problem związany z przeniesieniem logicznej struktury obiektowej na strukturę relacyjną bazy danych. Sytuacja odwrotna (wykorzystanie tabelarycznej struktury bazy danych w programie opartym o logiczną strukturę obiektów) także nie jest możliwa. Aby zatem możliwa była współpraca aplikacji z bazą danych wykorzystywane jest mapowanie obiektowo relacyjne. Dzięki temu programista przy zapisie do bazy danych może działać na obiektach, które są mapowane na relacje. Odczytane relacje z bazy danych także są mapowane na obiekty, co pozwala na ich przypisanie do odpowiedniego obiektu.

W mojej pracy wykorzystałem framework OrmLite. Jest to pakiet przeznaczony dla języka Java pozwalający na mapowanie obiektowo relacyjne. Jest to bardzo wygodne narzędzie, dzięki któremu programista może pominąć cały proces tworzenia własnego systemu mapowania obiektowo-relacyjnego. OrmLite pozwala na tworzenie obiektów tabel poprzez annotacje języka Java, co znacznie ułatwia proces ich tworzenia. Zapewnia także podstawową funkcjonalność DAO (Data Access Object) która następnie może być rozszerzona przez użytkownika. Operacje języka SQL takie jak: SELECT, INSERT, UPDATE, czy DELETE są mapowane na konkretne funkcje, dzięki czemu całą kwerendę wybierającą SELECT możemy zapisać w postaci jednej linijki kodu języka Java. Zdecydowałem się na wybranie Framework OrmLite ze względu na jego małą złożoność a także unikanie nadmiarowości pakietów ORM.



Rys. 8. Schemat działania Framework OrmLite.

1. **Projekt aplikacji**
   1. **Model opisowy**

Aplikacja MMA-Mobile Management Assistant pozwala na usprawnienie procesów

biznesowych zachodzących w małym przedsiębiorstwie.

UZUPEŁNIĆ NIE MAM WENY

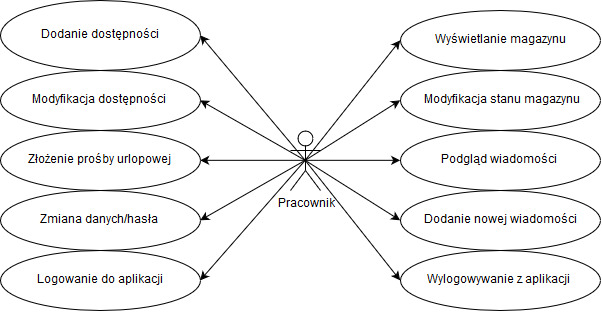
Głównym elementem systemu jest moduł zarządzania grafikami pracowników. Dzięki niemu wszelkie grafiki są przesy

* 1. **Przypadki użycia**

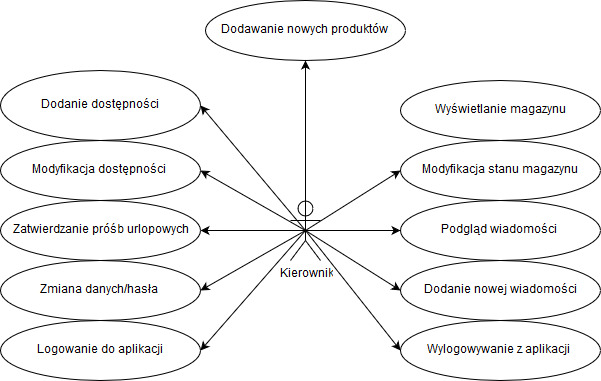
Diagramy przypadków użycia pozwalają na graficzne przedstawienie funkcjonalności

Systemu, w sposób w jaki są one widziane od strony użytkownika aplikacji. Diagramy przypadków użycia składają się z aktorów, oraz przypadków użycia połączonych z aktorami za pomocą asocjacji skierowanych. Diagramy przypadków użycia pomimo swojej prostoty są jednym z ważniejszych narzędzi wykorzystywanych podczas projektowania aplikacji.

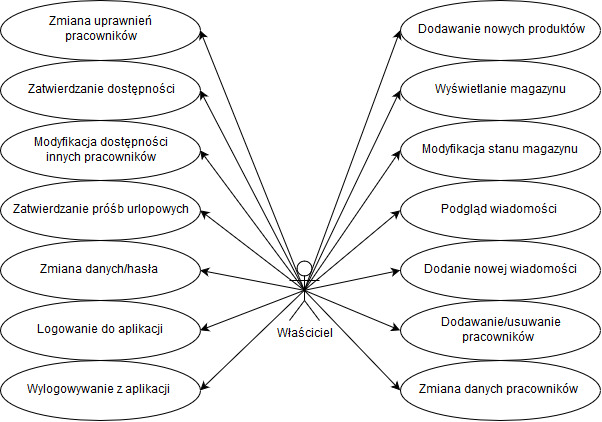
W niniejszej aplikacji można wyróżnić trzech podstawowych aktorów, którymi są: właściciel, kierownik, oraz pracownik. Diagramy przypadków użycia dla tych aktorów zostały przedstawione na poniższych diagramach. Część funkcjonalności jest wspólna dla wszystkich użytkowników aplikacji, jednak dla zachowania przejrzystości zostały one rozdzielone dla poszczególnych aktorów.



Rys. 9 Diagram przypadków użycia dla pracownika.



Rys. 10. Diagram przypadków użycia dla kierownika.



Rys. 11. Diagram przypadków użytkownika dla właściciela.

Na powyższych diagramach łatwo można zauważyć fakt, iż większość funkcji jest taka sama zarówno dla pracownika, kierownika jak i właściciela. Kierownik jest rozszerzeniem pracownika i posiada dodatkową funkcję pozwalającą na zatwierdzanie urlopów innych pracowników. Sam jednak nie musi składać wniosków o urlopy poprzez aplikację, ze względu na fakt, iż w małym przedsiębiorstwie prośby takie kierowane są zazwyczaj bezpośrednio do właściciela firmy. Dodatkowo Kierownik ma możliwość dodawania nowych produktów do magazynu oraz wyświetlania produktów których ilość przekroczyła dopuszczalny stan krytyczny. Największe uprawnienia w aplikacji, a co za tym idzie dostęp do największej liczby funkcji posiada właściciel. Konto właściciela jest swoistym rozszerzeniem konta Kierownika, Właściciel dodatkowo odpowiedzialny jest za zatwierdzenie oraz poprawę grafików. Dodatkowo właściciel jest zwolniony z odpowiedzialności dodawania grafiku dla siebie. Użytkownik z takimi uprawnieniami odpowiedzialny jest również za tworzenie kont dla pozostałych pracowników oraz nadawanie im uprawnień. Jako jedyny również może edytować dane każdego z pracowników a także usuwać użytkowników z systemu MMA-Mobile Management Assistant.

* 1. **Wymagania funkcjonalne i niefunkcjonalne**

Analiza wymagań funkcjonalnych oraz niefunkcjonalnych jest istotnym punktem w

tworzeniu oprogramowania. Pozwala ona na określenie wszelkich funkcjonalności systemu oraz wszelkich standardów systemu.

Wymagania funkcjonalne to konkretne możliwości jakie zapewnia system. Są to

konkretne funkcjonalności tworzonego oprogramowania. Wymagania funkcjonalne nie dotyczą jedynie użytkowników korzystających z danej aplikacji, ale także całego systemu, ze względu na fakt, iż tworzony system sam w sobie może zapewniać pewne funkcjonalności które są niedostępne dla użytkownika. Poniżej przedstawiona została lista wymagań funkcjonalnych systemu.

* Logowanie użytkownika do systemu

Analiza wymagań niefunkcjonalnych pozwala projektantowi aplikacji na określenie

tego, jak system powinien się zachowywać. W odróżnieniu od wymagań niefunkcjonalnych nie definiujemy tutaj funkcjonalności udostępnianej przez aplikację, a jedynie charakterystykę działania danego oprogramowania. Lista wymagań niefunkcjonalnych została przedstawiona poniżej.

* Łatwa rozbudowa aplikacji

Bibliografia

[1]Cay S. Horstmann, Gary Cornell *„Java Podstawy wydanie IX”*, Wydawnictwo HELION, Gliwice 2013