**Rzeszów, 07.06.2018 r.**

**Wydział: Matematyczno-Przyrodniczy  
Studia: Stacjonarne  
Kierunek: Informatyka  
Rok: III**

**Członkowie zespołu:**

**- Piotr Markowski (Team Lider)  
- Adrian Wilk  
- Tomasz Wysocki  
- Michał Śliwa**

**Przedmiot:   
Programowanie Zespołowe**

Dokumentacja Projektu

**„Aplikacja do zarządzania zadaniami**”

Prowadzący:

Mgr. Inż. Adam Szczur

# Spis treści

[Spis treści 2](#_Toc516066934)

[Cel Projektu 3](#_Toc516066935)

[Zakres projektu 3](#_Toc516066936)

[Przedsięwzięcie 3](#_Toc516066937)

[Użyte technologie 4](#_Toc516066938)

[NetBeans IDE 4](#_Toc516066939)

[Java 4](#_Toc516066940)

[JavaFX Builder 4](#_Toc516066941)

[MySQL (MariaDB) 4](#_Toc516066942)

[GitHub 5](#_Toc516066943)

[Jira 5](#_Toc516066944)

[IText PDF 5](#_Toc516066945)

[CSS – Kaskadowe Arkusze Stylów 5](#_Toc516066946)

[Diagramy 7](#_Toc516066947)

[Diagram Gantta 7](#_Toc516066948)

[Diagram Klas 8](#_Toc516066949)

[Diagram Przypadków Użycia 9](#_Toc516066950)

[Diagram Aktywności 9](#_Toc516066951)

[Diagram Sekwencji 10](#_Toc516066952)

[Baza Danych 11](#_Toc516066953)

[Diagram ERD 12](#_Toc516066954)

[Rodzaje dostępów i funkcjonalności 12](#_Toc516066955)

[Konta w Aplikacji: 12](#_Toc516066956)

[Panele w Aplikacji: 13](#_Toc516066957)

[Możliwości w aplikacji: 18](#_Toc516066958)

[Opis Obsługi Aplikacji 18](#_Toc516066959)

# Cel Projektu

Celem jest stworzenie uniwersalnego systemu do zarządzania zadaniami w przedsiębiorstwie. System posiadać będzie wiele funkcjonalności dając różne uprawnienia użytkownikom na podstawie posiadanego stanowiska i uprawnień w firmie. Aplikacja pozwoli zarządzać czasem, zadaniami, zasobami finansowymi i ludzkimi. Umożliwi również szybką komunikację pomiędzy użytkownikami oraz ułatwi prowadzenie obserwacji nad postępami w poszczególnych zadaniach. Zrealizowanie założeń projektu wymaga stworzenia środowiska i mechanizmów, które pozwolą na szybkie i niezawodne działanie programu. Aplikacja ma być intuicyjna i łatwa obsłudze, o prostym interfejsie, który będzie odpowiadał wymaganiom wielu użytkowników. Baza danych posłuży do przechowywania i modyfikacji wszystkich potrzebnych danych do zapewnienia pełnej funkcjonalności aplikacji.

W dokumentacji zawarte zostaną informacje dotyczące metod realizacji projektu, implementacji oraz użytych technologii.

# Zakres projektu

- Projekt będzie realizowany w 4 osobowej grupie z wykorzystaniem systemów GitHub i Jira

- Opracowanie wizji projektu i przygotowanie środowiska pracy

- Opracowanie diagramów technicznych (przypadków użycia, aktywności, sekwencji)

- Wstępne zaprojektowanie interfejsu użytkownika w Java FX Builder umożliwiające łatwe dodawanie dalszych modułów uwzględniając jednocześnie różne prawa dostępu. Logowanie z ustawionymi na „sztywno” użytkownikami

- Zaprojektowanie schematu bazy danych. Utworzenie diagramu ERD

- Budowanie złożonych kwerend SQL (budowa kwerend, które wybierają z kilku tabel)

- Stworzenie biblioteki generującej raporty w PDF i uruchomienie brakujących modułów

- Wykonanie brakujących elementów (testów, javadoc) oraz instalatora

- Tworzenie pakietów instalacyjnych

- Finalizacja i prezentacja projektu

# Przedsięwzięcie

Projekt będzie realizowany na zasadach metody SCRUM:  
- Samoorganizacja wewnątrz zespołu  
- Odpowiedzialność zbiorowa w ramach grupy  
- 2 tygodniowe Sprinty  
- Utrzymywanie aktualnych diagramów w Jira  
- Realizacja trzech Daily Sprintów na każdy Sprint

# Użyte technologie

## NetBeans IDE

NetBeans to [zintegrowane środowisko programistyczne](https://pl.wikipedia.org/wiki/Zintegrowane_%C5%9Brodowisko_programistyczne) (IDE) dla języka [Java](https://pl.wikipedia.org/wiki/Java), którego głównym celem jest przyspieszenie budowy aplikacji Java, w tym również [usług sieciowych](https://pl.wikipedia.org/wiki/Us%C5%82uga_internetowa) oraz [aplikacji mobilnych](https://pl.wikipedia.org/wiki/Java_Platform,_Micro_Edition). To również gotowa platforma służąca za fundament dla aplikacji [Rich Client](https://pl.wikipedia.org/wiki/Gruby_klient). Dzięki użyciu gotowej bazy i udostępnianych przez NetBeans Platform gotowych do użycia usług – okna, menu, zarządzanie i przechowywanie konfiguracji, dostęp do plików – można znacząco skrócić proces budowy aplikacji.

Java  
Obiektowy język programowania stworzony przez grupę roboczą pod kierunkiem Jamesa Goslinga z firmy Sun Microsystems. Java jest językiem tworzenia programów źródłowych kompilowanych do kodu bajtowego, czyli postaci wykonywanej przez maszynę wirtualną. Język cechuje się silnym typowaniem. Jego podstawowe koncepcje zostały przejęte z języka [Smalltalk](https://pl.wikipedia.org/wiki/Smalltalk) (maszyna wirtualna,  zarządzanie pamięcią) oraz z języka C++ (duża część składni i słów kluczowych).

JavaFX Builder  
JavaFX rozszerza potencjał platformy Java, pozwalając programistom na używanie bibliotek środowiska Java w aplikacjach JavaFX. W ten sposób programiści mogą poszerzyć swoje możliwości względem platformy Java i skorzystać z technologii prezentacji, jaką oferuje JavaFX, umożliwiając konstruowanie środowisk o atrakcyjnej szacie graficznej. Użytkownicy mogą uruchamiać aplikacje JavaFX w przeglądarce lub przeciągać je na pulpit.

MySQL (MariaDB)  
MariaDB to [baza danych](https://pl.wikipedia.org/wiki/Baza_danych) stworzona przez grupę byłych pracowników MySQL AB, pod przewodnictwem Michaela Wideniusa, współtwórcy [MySQL](https://pl.wikipedia.org/wiki/MySQL). Celem głównym projektu jest współpraca ze społecznością [wolnego oprogramowania](https://pl.wikipedia.org/wiki/Wolne_oprogramowanie) i udostępnianie jej na licencji [GPL](https://pl.wikipedia.org/wiki/GNU_General_Public_License), w przeciwieństwie do niepewnego statusu licencji MySQL, która zależy teraz od firmy [Oracle](https://pl.wikipedia.org/wiki/Oracle_Corporation). Autorzy MariaDB stawiają sobie również za cel utrzymanie kompatybilności z wcześniejszymi wersjami MySQL. [Kod źródłowy](https://pl.wikipedia.org/wiki/Kod_%C5%BAr%C3%B3d%C5%82owy) MariaDB bazuje na wcześniejszej wersji MySQL

GitHubGit jest rozproszonym systemem kontroli wersji. Stanowi wolne oprogramowanie i został opublikowany na licencji GNU GPL. Pierwsza wersja narzędzia Git została wydana 7 kwietnia 2005 roku. Cechuje się dobrym wsparciem dla rozgałęzionego procesu tworzenia oprogramowania, algorytmów łączenia zmian z dwóch gałęzi, dodawaniem własnych algorytmów, pracą off-line tzn. każdy programista posiada własną kopię repozytorium lokalnie, gdzie może zapisywać zamiany na bieżąco. GitHub również wspiera istniejące protokoły sieciowe:  HTTP(S), FTP, rsync, SSH. Umożliwia efektywną pracę z dużymi projektami.

JiraJira to [zamknięte oprogramowanie](https://pl.wikipedia.org/wiki/Zamkni%C4%99te_oprogramowanie) firmy [Atlassian](https://pl.wikipedia.org/w/index.php?title=Atlassian&action=edit&redlink=1) służące do [śledzenia błędów](https://pl.wikipedia.org/wiki/Bugtracker) oraz [zarządzania projektami](https://pl.wikipedia.org/wiki/Zarz%C4%85dzanie_projektami). Umożliwia zarządzanie dokumentacją, zasobami firmy, ułatwia wymianę informacji wewnątrz firmy, daje możliwość tworzenia kartotek pracowników i ewidencji zgłoszeń oraz paneli. W głównej mierze pozwala na zarządzanie projektami i zadaniami.

## IText PDF

IText PDF jest darmową biblioteką stworzoną dla języka JAVA oraz C# służącą do dynamicznego tworzenia i generowania dokumentów PDF. Obsługuje obszerną bazę czcionek wraz z różnymi typami kodowania. Umożliwia dzielenie, łączenie i manipulowanie dokumentami PDF. Pozwala również na osadzanie skomplikowanych elementów graficznych w dokumencie. Daje również możliwość edytowania już istniejących dokumentów.

## CSS – Kaskadowe Arkusze Stylów

CSS - jest to specjalny język opracowany tylko w jednym celu: stworzenie możliwości bardziej elastycznego zarządzania sposobem formatowania (wyglądem) elementów znajdujących się w dokumentach elektronicznych. CSS nie może zatem istnieć samodzielnie, gdyż jest ściśle powiązane z językiem opisu struktury dokumentów takim jak (X)HTML czy JAVAFX. CSS daje możliwość globalnego zarządzania formą prezentacji całej witryny oraz aplikacji internetowej.

# Diagramy

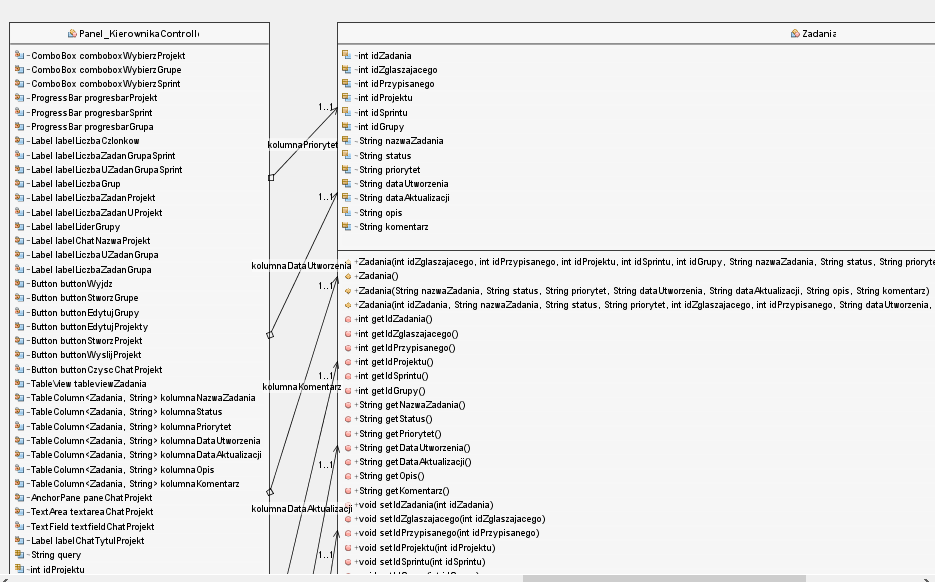
## Diagram Gantta

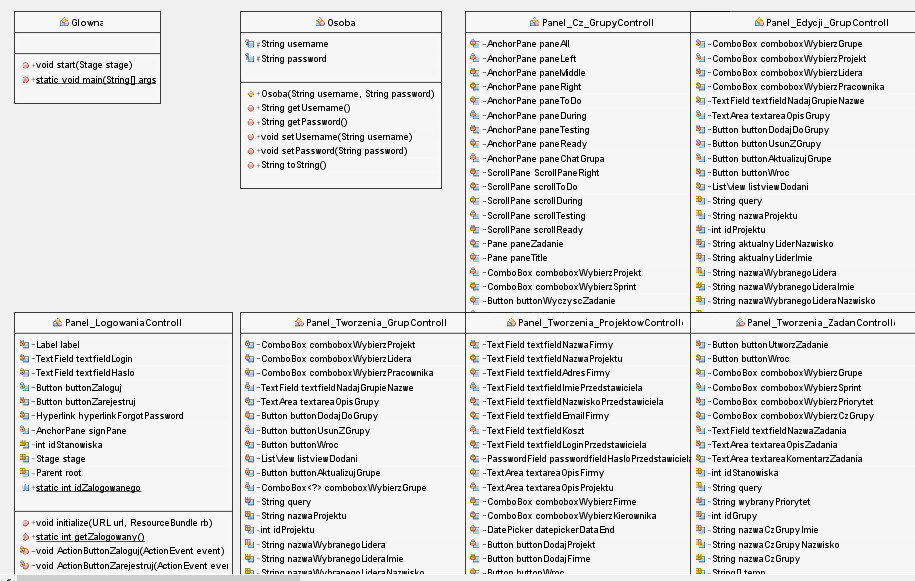
Przedstawia podział projektu na zadania oraz czas, w jakim zostały wykonane.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr** | **Zadanie** | **Początek** | **Trwanie  (w dniach)** | **Zakończenie** |
| **1** | Opracowanie wizji projektu i przygotowanie środowiska pracy | 1 mar | 1 | 2 mar |
| **2** | Działające okna (zakładki) min. 2 modułów | 2 mar | 1 | 3 mar |
| **3** | Zaprojektowanie działającej bazy danych | 3 mar | 7 | 10 mar |
| **4** | Zaprojektowanie wyglądu interfejsu | 10 mar | 7 | 17 mar |
| **5** | Stworzenie bazy danych | 17 mar | 13 | 30 mar |
| **6** | Implementacja struktury projektu | 30 mar | 28 | 27 kwi |
| **7** | Połączenie projektu z bazą danych | 27 kwi | 8 | 5 maj |
| **8** | Budowa złożonych kwerend SQL | 5 maj | 9 | 14 maj |
| **9** | Sporządzenie testów aplikacji i wykonanie brakujących modułów | 14 maj | 14 | 28 maj |
| **10** | Wykonanie dokumentacji | 28 maj | 10 | 7 cze |
| **11** | Oddanie projektu | 7 cze | 1 | 8 cze |

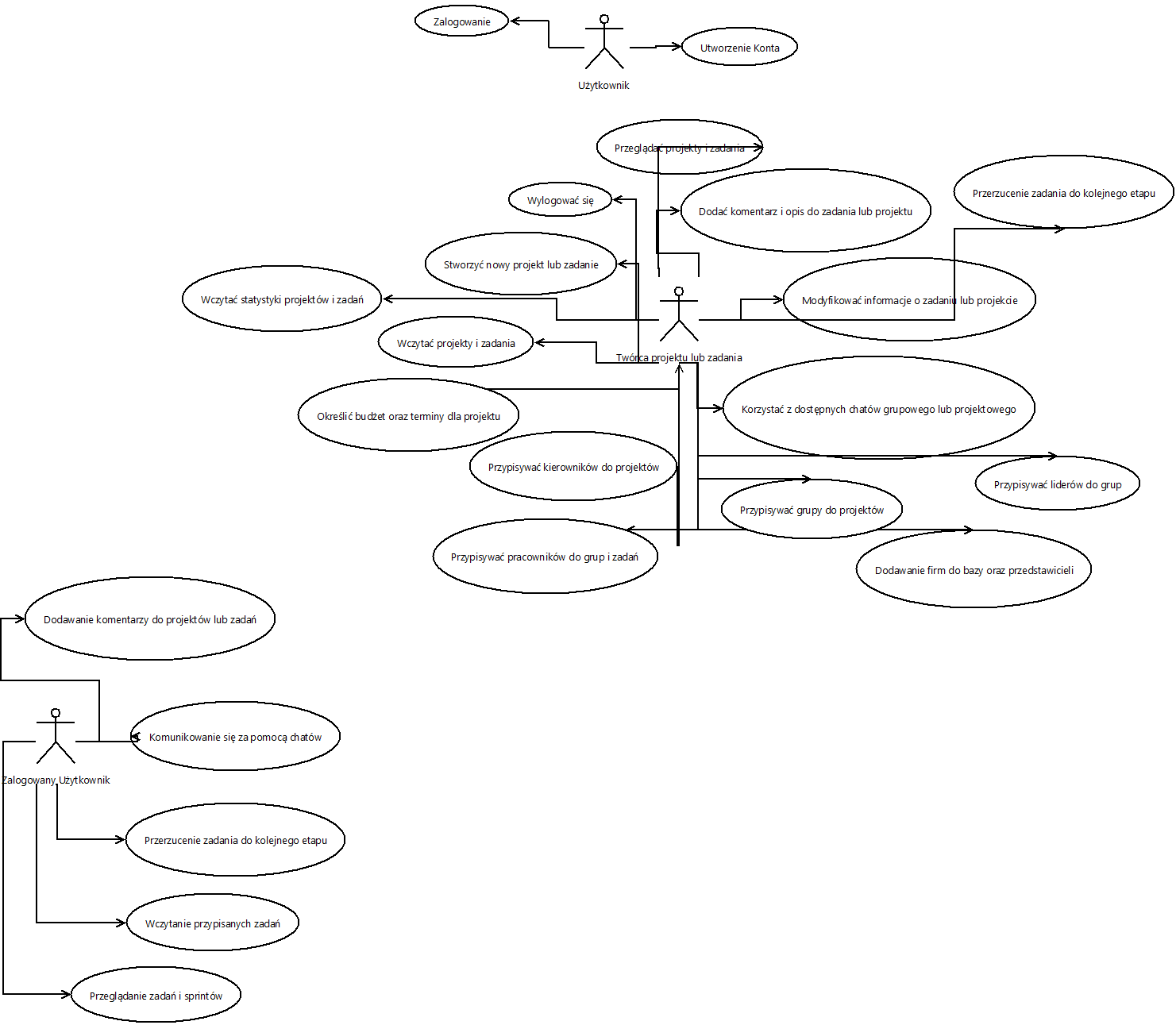
## Diagram Klas

Został on wygenerowany w środowisku NetBeans 8.2 przy użyciu pluginu EasyUML. Poniżej przedstawiony jest fragment:

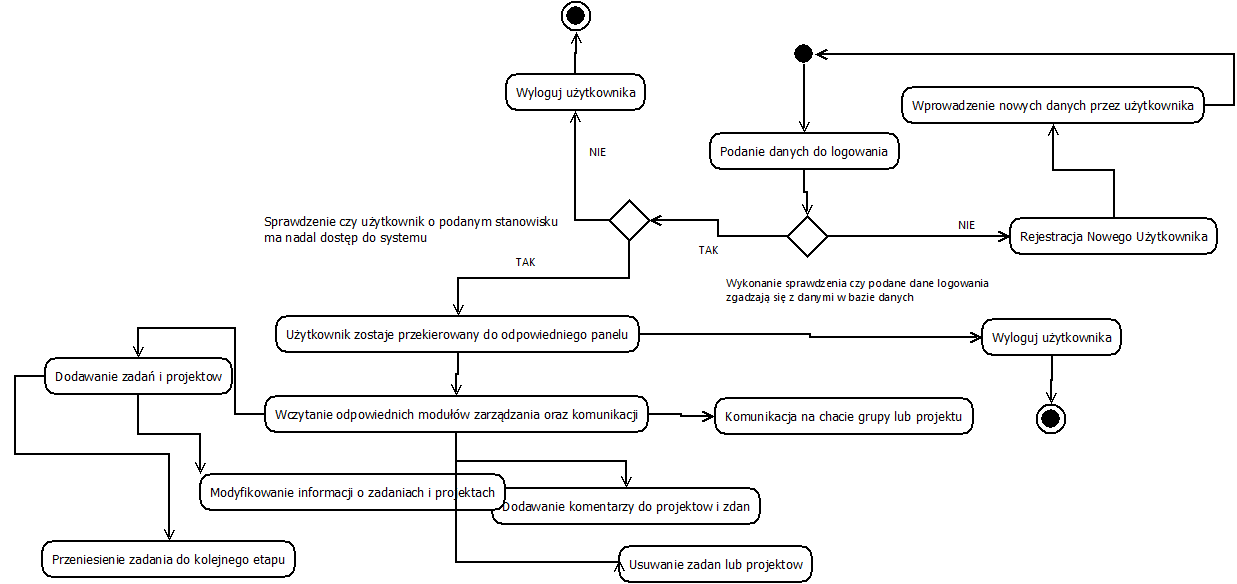




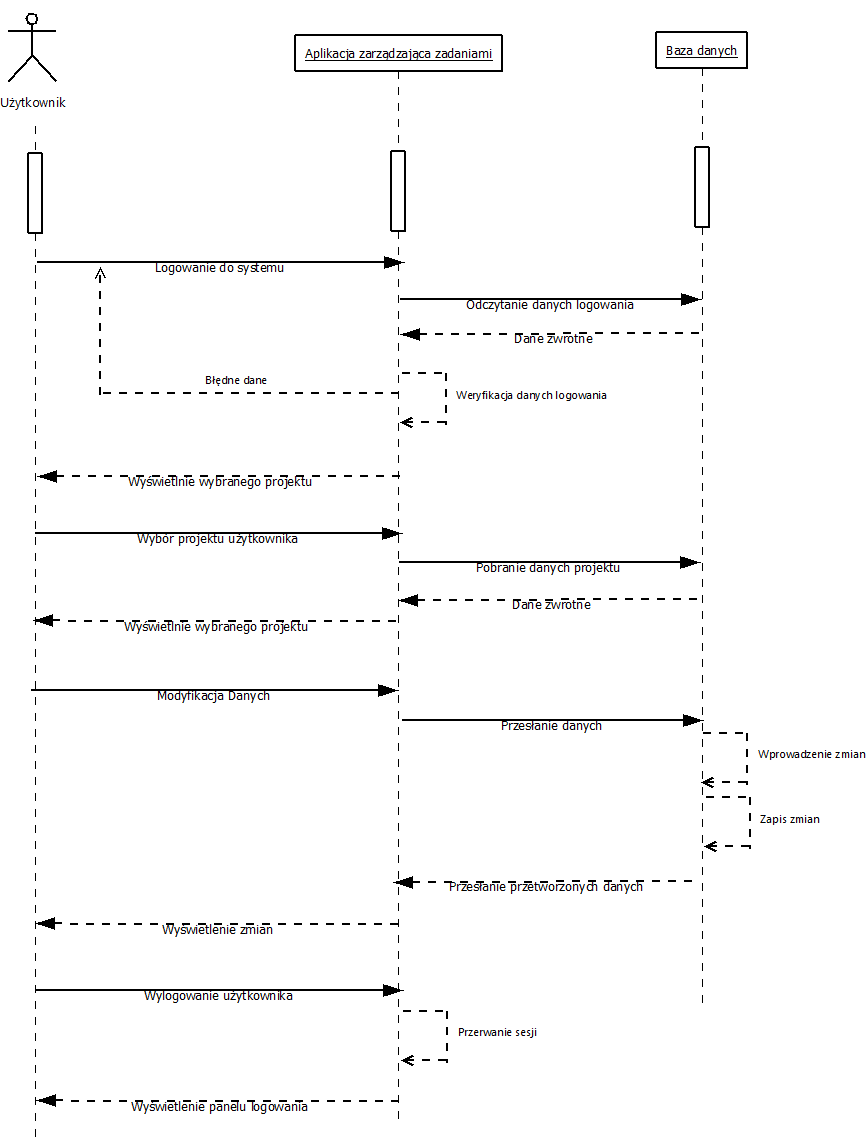
## Diagram Przypadków Użycia



## Diagram Aktywności

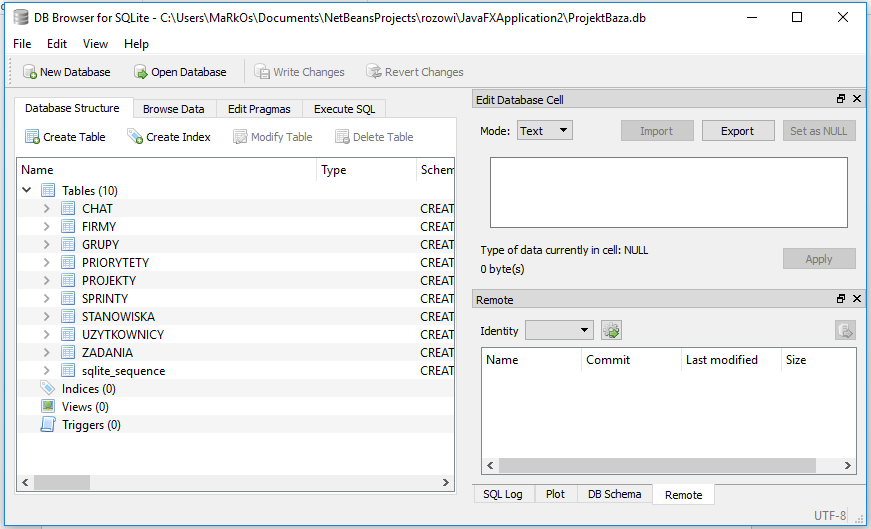


## Diagram Sekwencji



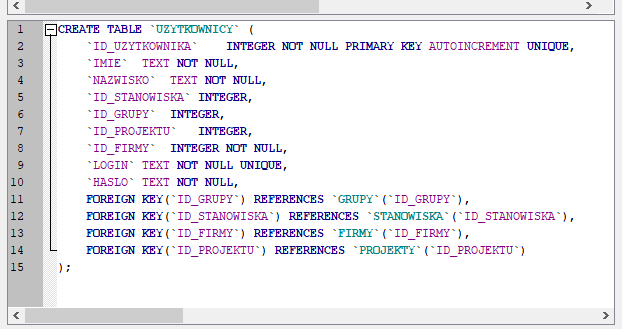
# Baza Danych

Baza danych została wykonana w programie DB Browser for SQLite



Baza danych składa się z 9 tabel: CHAT / FIRMY / GRUPY / PRIORYTETY / PROJEKTY / SPRINTY / STANOWISKA / UZYTKOWNICY / ZADANIA.

Przykładowy kod tworzący tabelę UZYTKOWNICY:



## Skrypt SQL

BEGIN TRANSACTION;

CREATE TABLE IF NOT EXISTS `ZADANIA` (

`ID\_ZADANIA` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,

`NAZWA\_ZADANIA` TEXT NOT NULL UNIQUE,

`STATUS` INTEGER NOT NULL,

`PRIORYTET` TEXT NOT NULL,

`ID\_ZGLASZAJACEGO` INTEGER NOT NULL,

`ID\_PRZYPISANEGO` INTEGER NOT NULL,

`DATA\_UTWORZENIA` TEXT NOT NULL,

`DATA\_AKTUALIZACJI` TEXT NOT NULL,

`OPIS` TEXT NOT NULL,

`KOMENTARZ` TEXT,

`ID\_PROJEKTU` INTEGER NOT NULL,

`ID\_SPRINTU` INTEGER NOT NULL,

`ID\_GRUPY` INTEGER,

FOREIGN KEY(`ID\_GRUPY`) REFERENCES `GRUPY`(`ID\_GRUPY`),

FOREIGN KEY(`ID\_ZGLASZAJACEGO`) REFERENCES `UZYTKOWNICY`(`ID\_UZYTKOWNIKA`),

FOREIGN KEY(`ID\_PRZYPISANEGO`) REFERENCES `UZYTKOWNICY`(`ID\_UZYTKOWNIKA`),

FOREIGN KEY(`ID\_SPRINTU`) REFERENCES `SPRINTY`(`ID\_SPRINTU`),

FOREIGN KEY(`ID\_PROJEKTU`) REFERENCES `PROJEKTY`(`ID\_PROJEKTU`)

);

CREATE TABLE IF NOT EXISTS `UZYTKOWNICY` (

`ID\_UZYTKOWNIKA` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,

`IMIE` TEXT NOT NULL,

`NAZWISKO` TEXT NOT NULL,

`ID\_STANOWISKA` INTEGER,

`ID\_GRUPY` INTEGER,

`ID\_PROJEKTU` INTEGER,

`ID\_FIRMY` INTEGER NOT NULL,

`LOGIN` TEXT NOT NULL UNIQUE,

`HASLO` TEXT NOT NULL,

FOREIGN KEY(`ID\_PROJEKTU`) REFERENCES `PROJEKTY`(`ID\_PROJEKTU`),

FOREIGN KEY(`ID\_GRUPY`) REFERENCES `GRUPY`(`ID\_GRUPY`),

FOREIGN KEY(`ID\_FIRMY`) REFERENCES `FIRMY`(`ID\_FIRMY`),

FOREIGN KEY(`ID\_STANOWISKA`) REFERENCES `STANOWISKA`(`ID\_STANOWISKA`)

);

CREATE TABLE IF NOT EXISTS `STANOWISKA` (

`ID\_STANOWISKA` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,

`NAZWA\_STANOWISKA` TEXT NOT NULL UNIQUE,

`OPIS\_STANOWISKA` TEXT NOT NULL

);

CREATE TABLE IF NOT EXISTS `SPRINTY` (

`ID\_SPRINTU` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,

`NAZWA\_SPRINTU` TEXT NOT NULL,

`DATA\_START\_SPRINTU` TEXT NOT NULL,

`DATA\_END\_SPRINTU` TEXT NOT NULL,

`OPIS\_SPRINTU` TEXT NOT NULL,

`ID\_PROJEKTU` INTEGER,

FOREIGN KEY(`ID\_PROJEKTU`) REFERENCES `PROJEKTY`(`ID\_PROJEKTU`)

);

CREATE TABLE IF NOT EXISTS `PROJEKTY` (

`ID\_PROJEKTU` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,

`NAZWA\_PROJEKTU` TEXT NOT NULL UNIQUE,

`ID\_FIRMY\_ZLECAJACEJ` INTEGER NOT NULL,

`DATA\_START` TEXT NOT NULL,

`DATA\_END` TEXT NOT NULL,

`KOSZT\_PROJEKTU` INTEGER NOT NULL,

`OPIS\_PROJEKTU` TEXT NOT NULL,

`STATUS\_PROJEKTU` TEXT,

`KOMENTARZ\_ZLECAJACEGO` TEXT,

`ID\_KIEROWNIKA` INTEGER NOT NULL UNIQUE,

FOREIGN KEY(`ID\_FIRMY\_ZLECAJACEJ`) REFERENCES `UZYTKOWNICY`(`ID\_UZYTKOWNIKA`),

FOREIGN KEY(`ID\_KIEROWNIKA`) REFERENCES `UZYTKOWNICY`(`ID\_UZYTKOWNIKA`)

);

CREATE TABLE IF NOT EXISTS `PRIORYTETY` (

`ID\_PRIORYTETU` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,

`NAZWA\_PRIORYTETU` TEXT NOT NULL UNIQUE,

`OPIS` TEXT

);

CREATE TABLE IF NOT EXISTS `GRUPY` (

`ID\_GRUPY` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,

`NAZWA\_GRUPY` TEXT NOT NULL UNIQUE,

`ID\_LIDERA` INTEGER NOT NULL,

`ID\_PROJEKTU` INTEGER NOT NULL,

`OPIS\_GRUPY` TEXT NOT NULL,

FOREIGN KEY(`ID\_PROJEKTU`) REFERENCES `PROJEKTY`(`ID\_PROJEKTU`),

FOREIGN KEY(`ID\_LIDERA`) REFERENCES `UZYTKOWNICY`(`ID\_UZYTKOWNIKA`)

);

CREATE TABLE IF NOT EXISTS `FIRMY` (

`ID\_FIRMY` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,

`NAZWA\_FIRMY` TEXT NOT NULL UNIQUE,

`ADRES\_FIRMY` TEXT NOT NULL,

`OPIS\_FIRMY` TEXT NOT NULL,

`EMAIL\_FIRMY` TEXT UNIQUE

);

CREATE TABLE IF NOT EXISTS `CHAT` (

`ID\_WIADOMOSCI` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,

`LOGIN` TEXT NOT NULL,

`TRESC` TEXT NOT NULL,

`DATA\_WYSLANIA` TEXT NOT NULL,

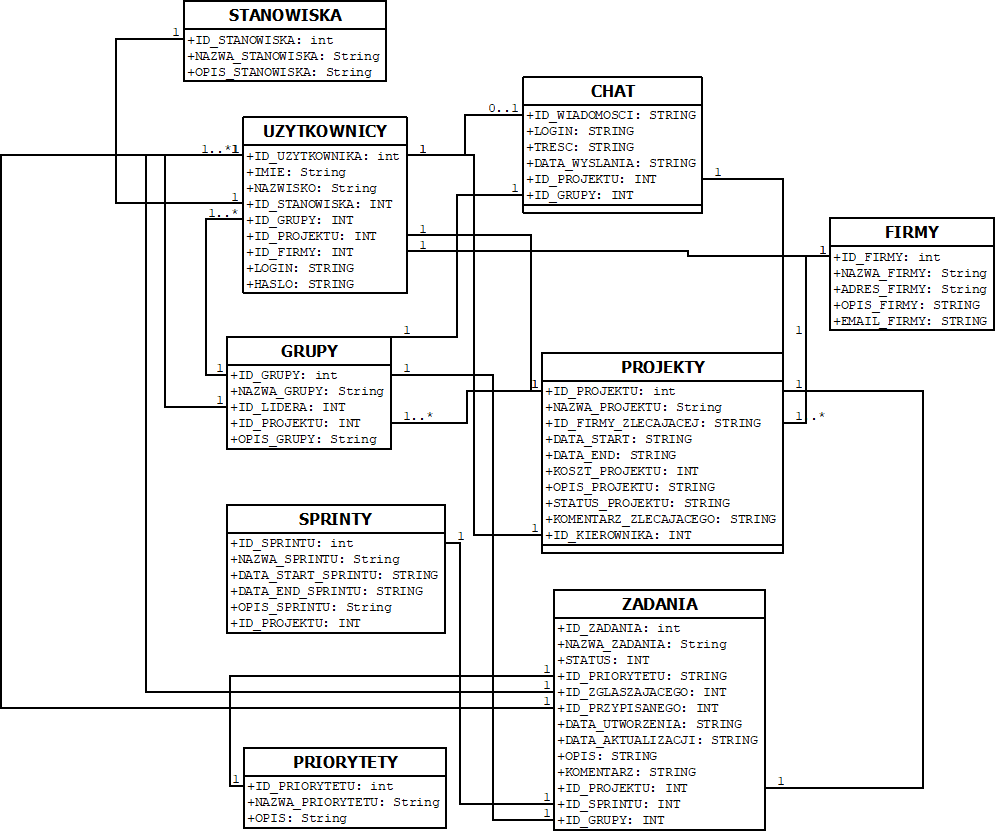
`ID\_PROJEKTU` INTEGER NOT NULL,

`ID\_GRUPY` INTEGER NOT NULL

);

COMMIT;

## Diagram ERD



# Rodzaje dostępów i funkcjonalności

## Konta w Aplikacji:

***Konto Zlecającego Projekt ->*** czyli przedstawiciela klienta, który zleca naszej firmie wykonanie projektu. Dzięki posiadanemu dostępowi będzie mógł kontrolować i śledzić ogólny proces wykonania projektu + ewentualnie dodawać informacje o poprawkach, jakie by chciał, aby zostały wprowadzone w projekcie.

***Konto Szefa*** (właściciel firmy wykonującej projekt) -> podobnie jak wyżej z tym, że to on będzie tworzył nowe zadania/projekty w aplikacji i przypisywał je do odpowiednich kierowników i liderów.

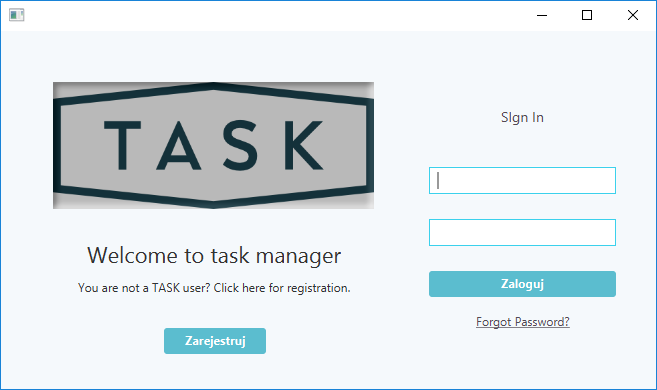
***Konto Kierownika Projektu*** -> osoba, która zarządza grupą lub grupami osób pracujących nad zadaniem/projektem.

***Konto Lidera Grupy*** -> zarządza grupą, tworzy zadania oraz przydziela je pracownikom

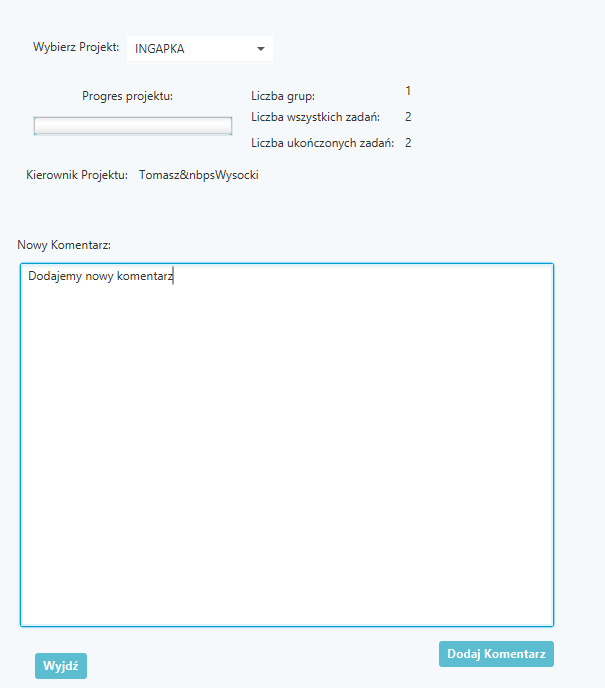
***Konto Pracownika*** -> ma dostęp do wglądu, do otrzymanych zadań i będzie mógł „informować” o postępach w zadaniach do niego przypisanych

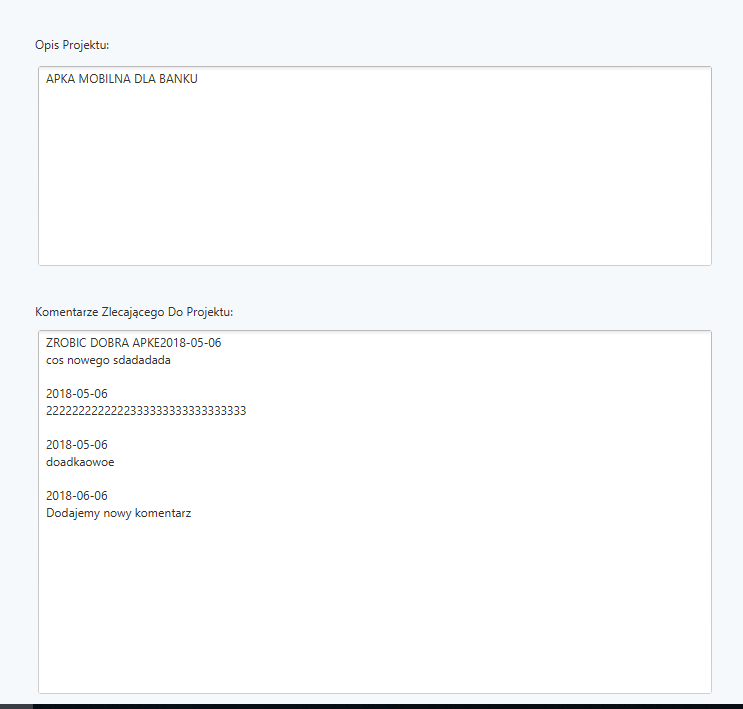
## Panele w Aplikacji: (GUI W WERSJI ALPHA!)

**Panel Logowania**

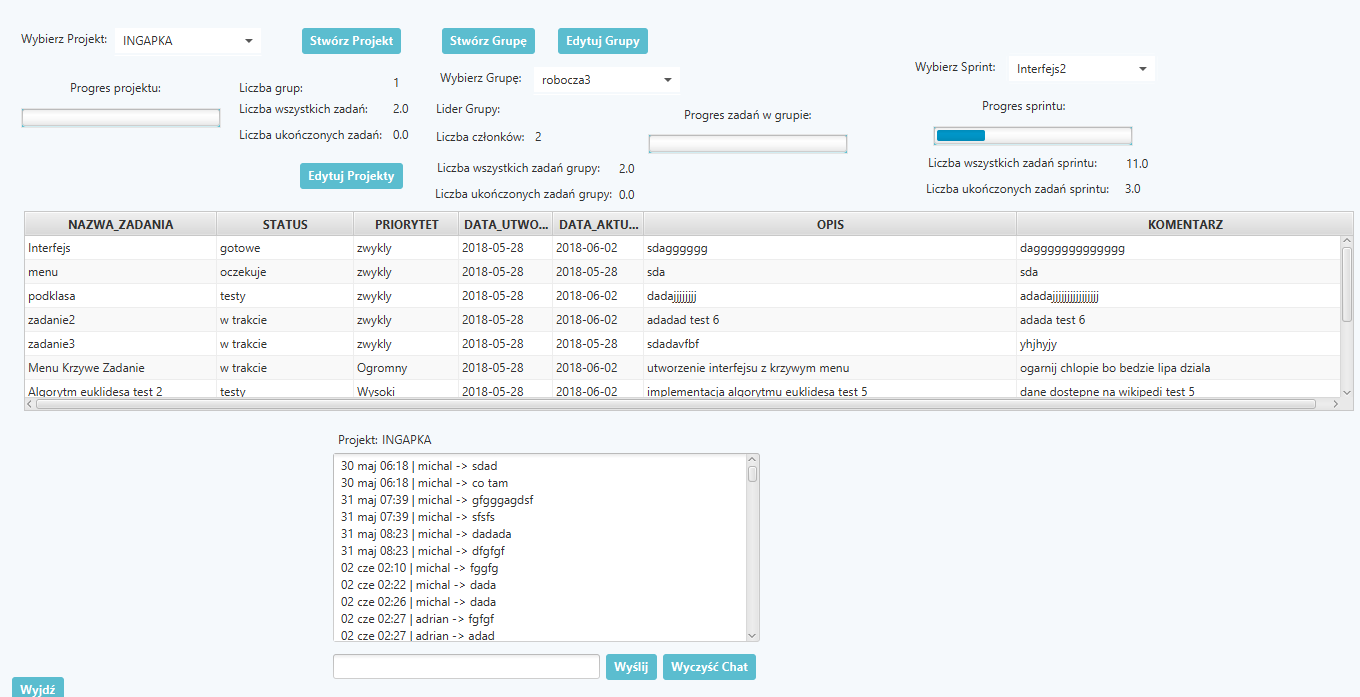


**Panel Zlecającego Projekt**

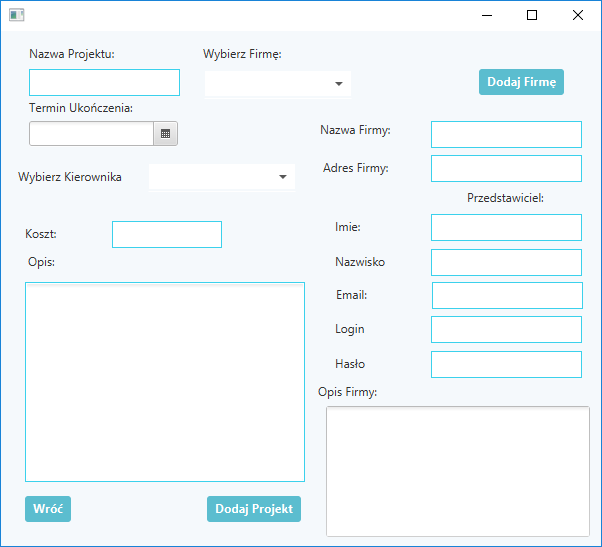




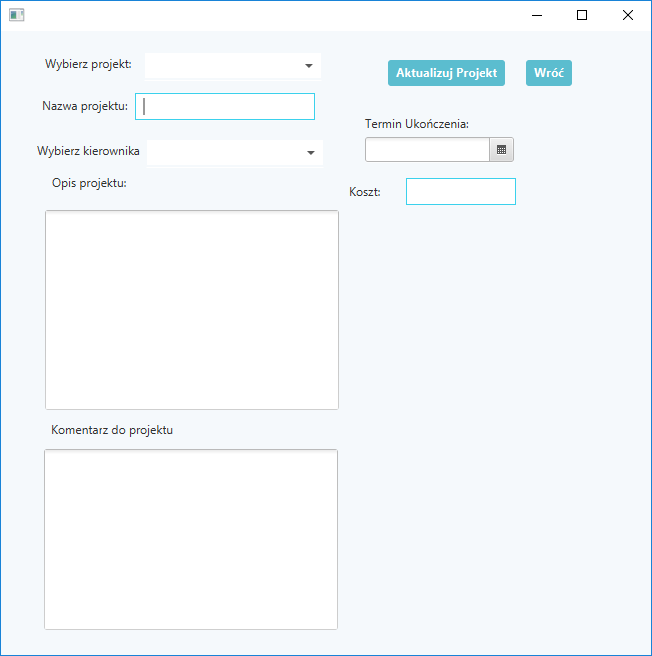
**Panel Szefa (Właściciela firmy)**



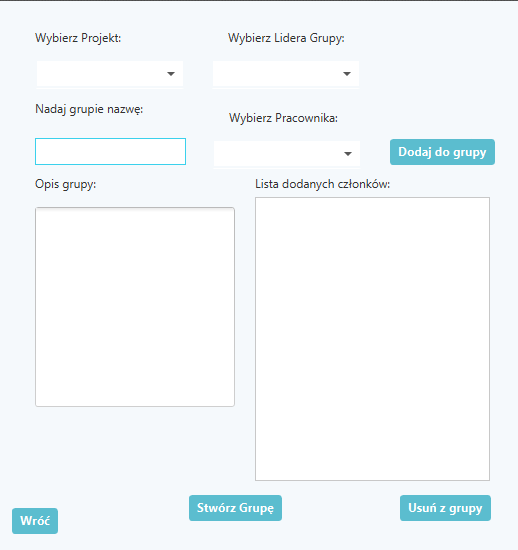
**Panel tworzenia nowych projektów**



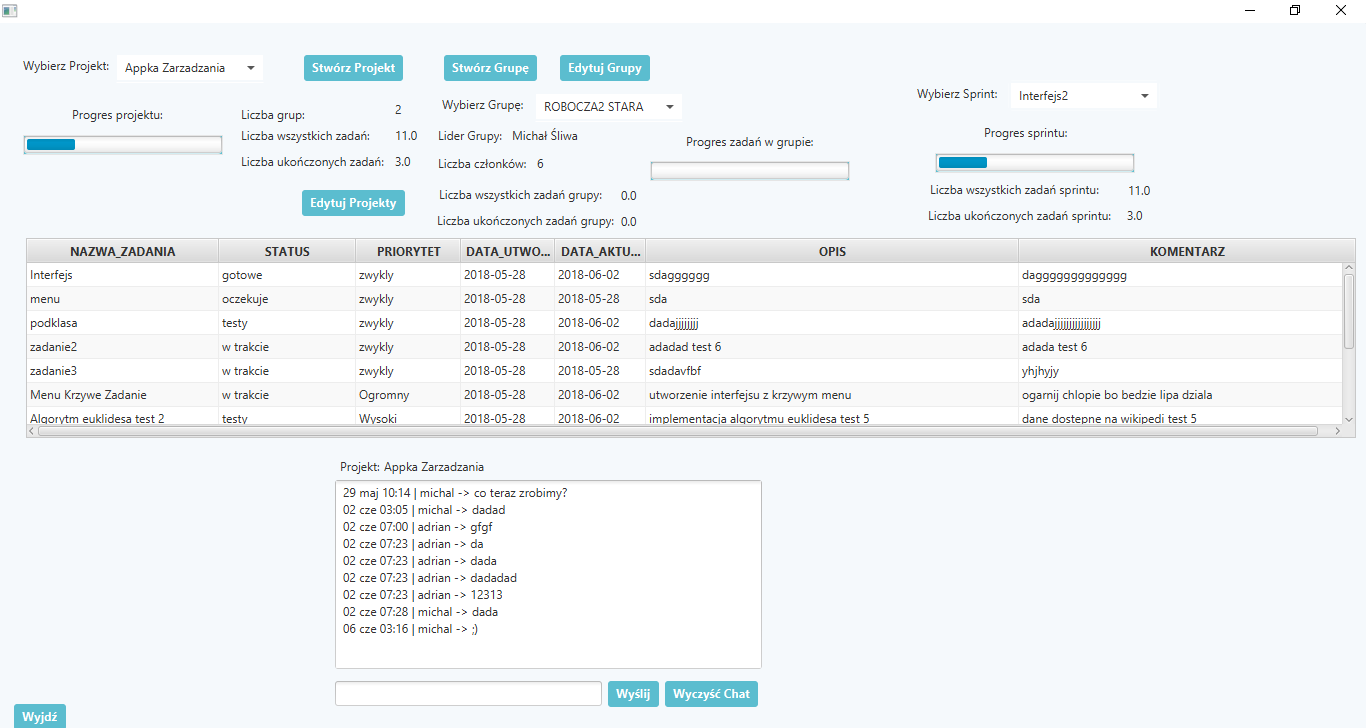
**Panel Edycji Projektów**



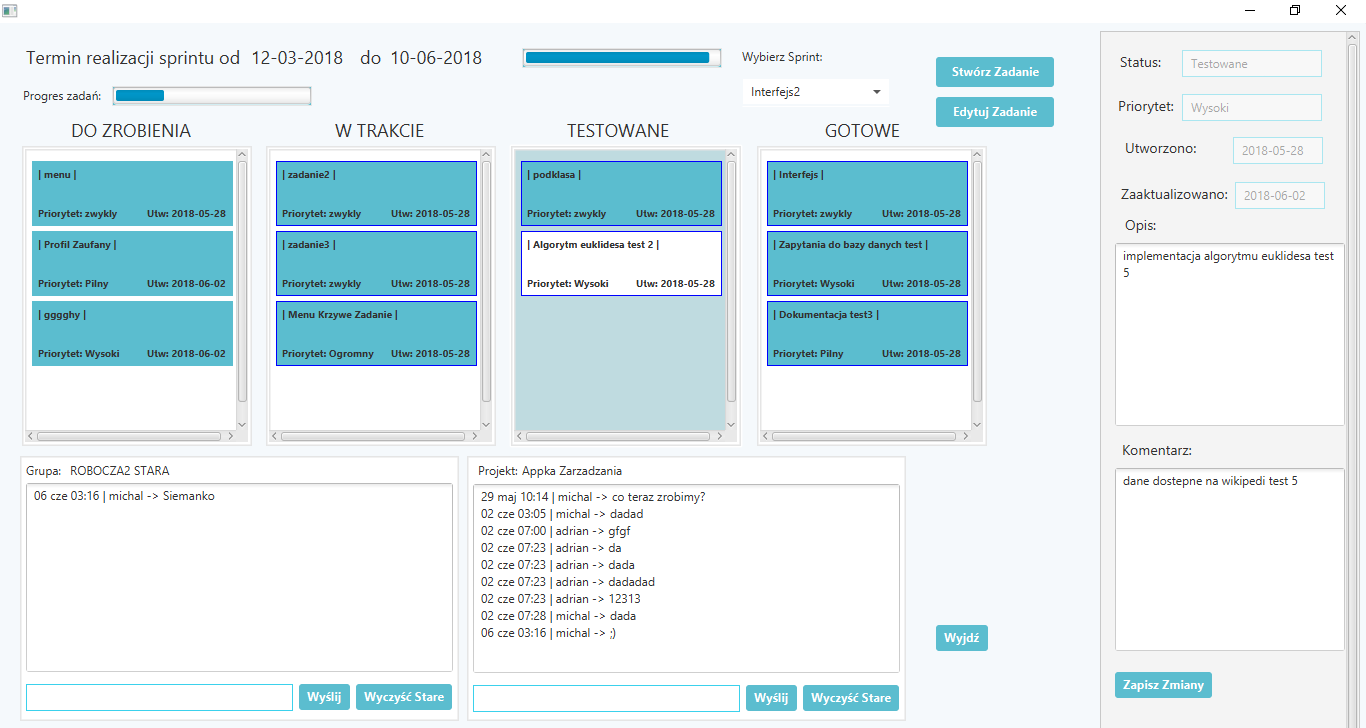
**Panel Tworzenia oraz Edycji Grup**



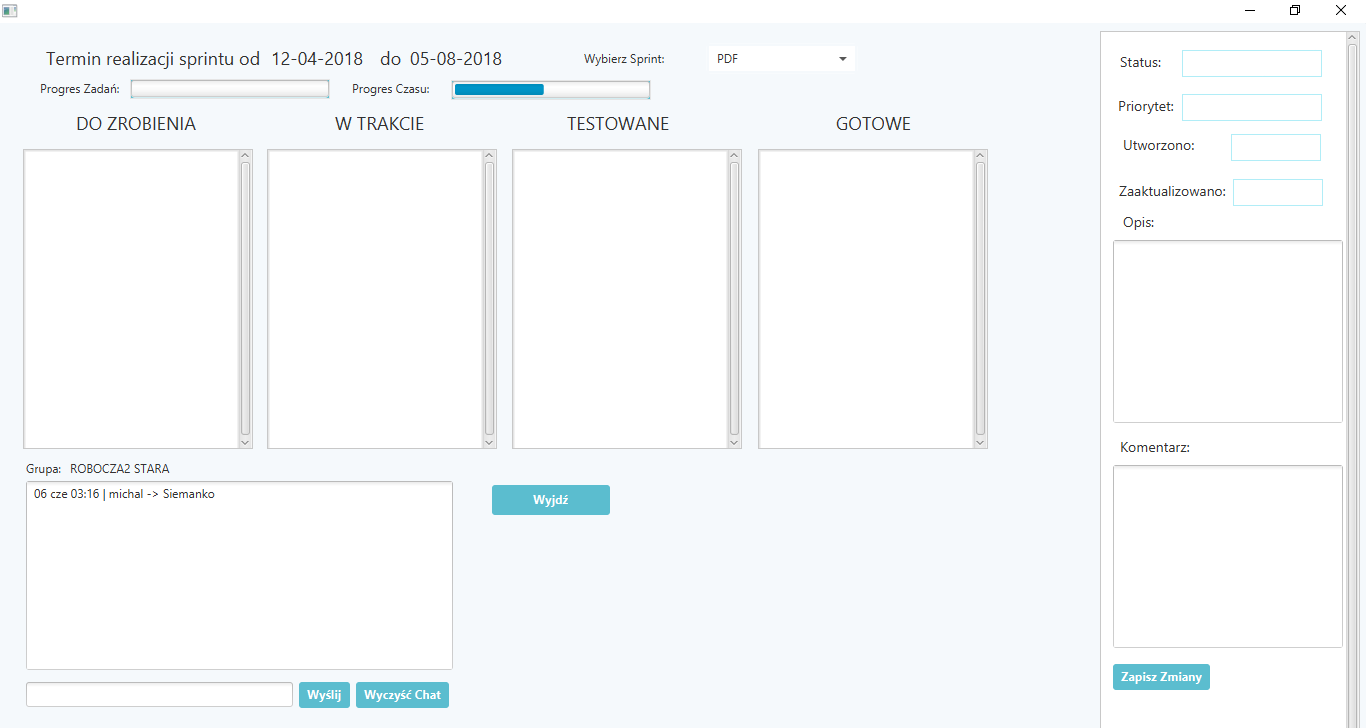
**Panel Kierownika Projektu**



**Panel Lidera Grupy**



**Panel Pracownika**



## Możliwości w aplikacji:

- Tworzenie Zadań / Projektów, nadawanie im nazwy roboczej, informacje, dla kogo jest wykonywany dany projekt, określony budżet i termin zakończenia projektu, przypisywanie do nich kierowników

- Tworzenie grup i przypisywanie do nich pracowników przez Liderów oraz określanie zakresu zadań, jakie mają wykonać + terminy

- Tworzenie konkretnych zadań i przypisywanie do nich pracowników przez Liderów

- Prowadzenie rozmów na chatach w obrębie grupy lub projektu

# Generowanie raportów

Aplikacja pozwala na generowanie raportów z wykonanych prac przy projekcie. Tworzenie takowego raportu pozwala biblioteka IText zaimportowana w projekcie wraz z odpowiednim kodem źródłowym napisanym przez nas, autorów systemu zarządzania zadaniami. W wygenerowanym dokumencie PDF znajdują się wszystkie najważniejsze informacje takie, jak: kto zlecił projekt, pod jaką nazwą, jaki został przeznaczony termin na wykonanie projektu, jaki został określony budżet, opis projektu + dodane w czasie tworzenia projektu komentarze do projektu dodawane przez zleceniodawcę. Zawarte są również informacje o kierowniku projektu, liderach grup, przydzielonych pracownikach do grup oraz wszystkich zadaniom im zleconym i wykonanym.