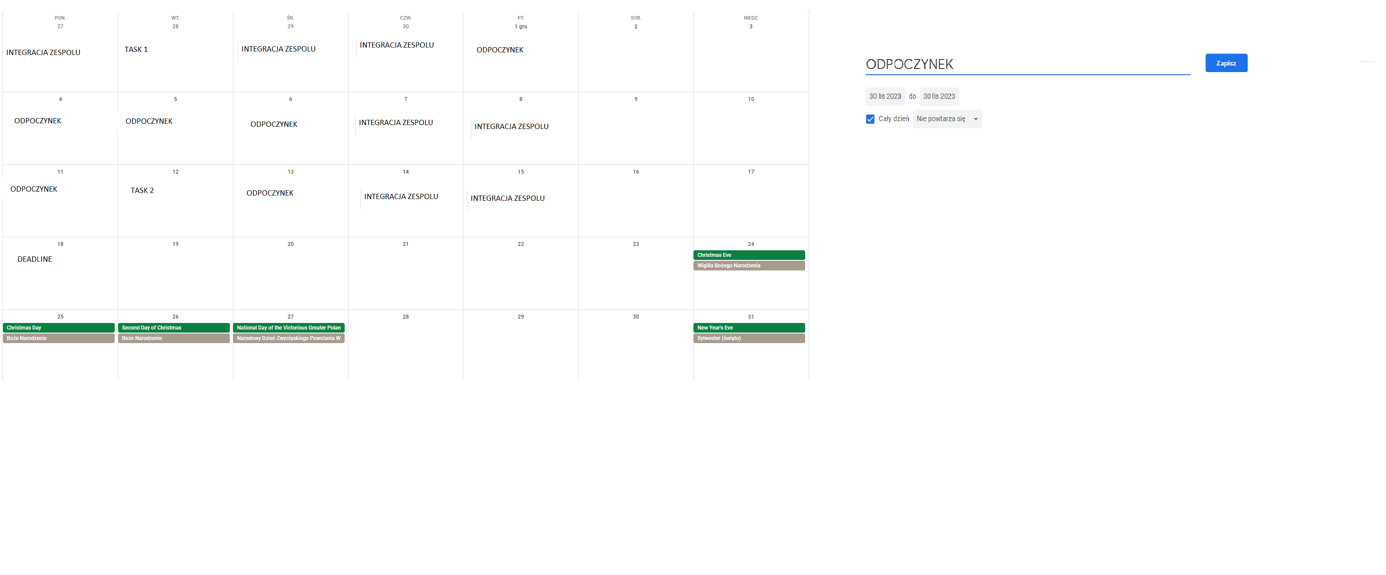
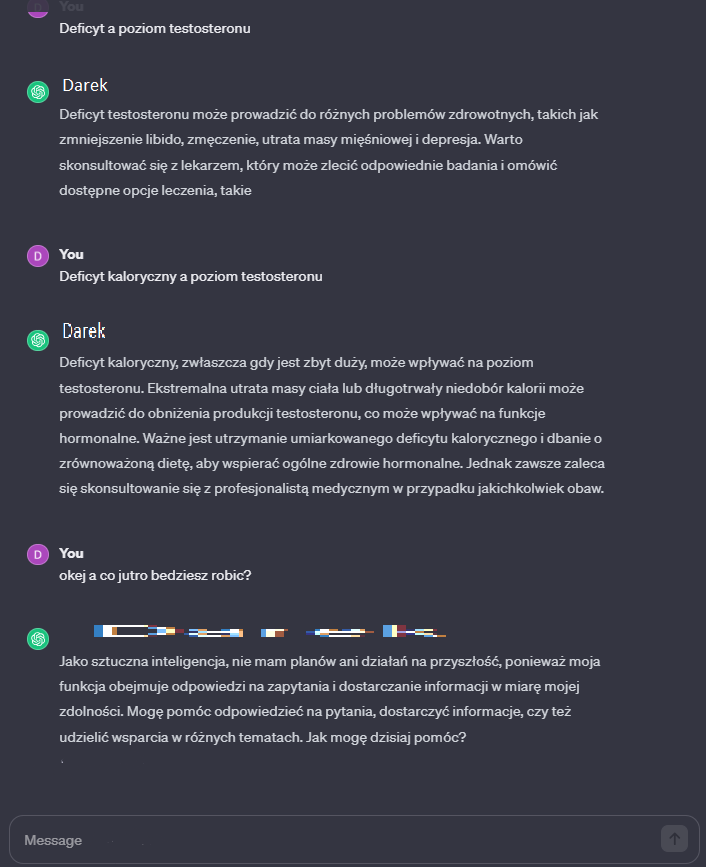
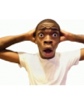
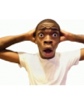
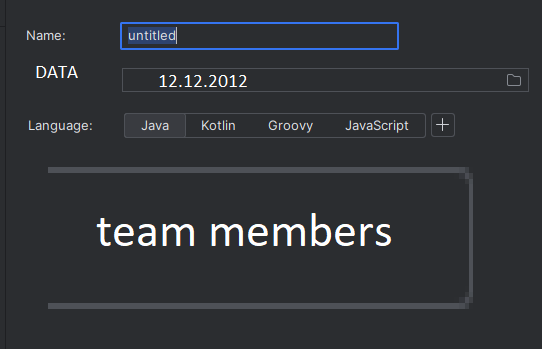
**Koncepcja wykonania systemu**

**Aplikacja „Ogar”**

**Autorzy: Dominik Szczepaniak & Olejarz**









1. Model konceptualny rzeczywistości
   * Tabela Projektów:
     + ProjektID (PK)
     + Nazwa projektu
     + Opis
     + Data rozpoczęcia
     + Data zakończenia
     + Status projektu (np. w trakcie, zakończony, zawieszony)
   * Tabela Zespołów:
     + ZespółID (PK)
     + Nazwa zespołu
     + Opis zespołu
   * Tabela Członków Zespołów:
     + CzłonekID (PK)
     + Imię
     + Nazwisko
     + Email
     + Rola w projekcie (np. lider, programista, tester)
   * Tabela Faz Projektu:
     + FazaID (PK)
     + Nazwa fazy
     + Opis fazy
   * Tabela Zadania:
     + ZadanieID (PK)
     + Nazwa zadania
     + Opis zadania
     + Status zadania (np. do zrobienia, w trakcie, zakończone)
     + Termin wykonania
     + ProjektID (FK - klucz obcy powiązany z tabelą Projektów)
     + FazaID (FK - klucz obcy powiązany z tabelą Faz Projektu)
     + PrzypisaneCzłonki (FK - klucz obcy powiązany z tabelą Członków Zespołów)
   * Tabela Zależności Zadań:
     + ZależnośćID (PK)
     + ZadanieID (FK - klucz obcy powiązany z tabelą Zadania)
     + ZadanieZależneID (FK - klucz obcy powiązany z tabelą Zadania)
   * Tabela Zasobów Projektowych:
     + ZasóbID (PK)
     + Nazwa zasobu
     + Opis zasobu
     + Dostępność
     + ProjektID (FK - klucz obcy powiązany z tabelą Projektów)
   * Tabela Komunikacji:
     + KomunikacjaID (PK)
     + Data
     + Nadawca
     + Odbiorca
     + Treść
     + ProjektID (FK - klucz obcy powiązany z tabelą Projektów)
   * Tabela Zdarzeń Projektu:
     + ZdarzenieID (PK)
     + Data zdarzenia
     + Opis zdarzenia
     + ProjektID (FK - klucz obcy powiązany z tabelą Projektów)
   * Użytkownik:
     + Imie
     + Nazwisko
     + Data urodzenia
     + Email
     + NrTelefonu
2. Podstawowe elementy aplikacji i ich powiązania

* Skorzystamy z chmury amazon AWS, na której postawimy dockery odpowiednio dla serwera oraz bazy danych.
* Serwer​​ będzie napisany w ASP.NET.
* Komunikacja audio-wideo​​ oparta zostanie o standard WebRTC.
* Baza danych​​ to postgresql.
* Frontend napisany w ASP.NET przy pomocy React

1. Omówienie interfejsów aplikacji

* W interfejsie stawiamy na prostotę i minimalizm.
* Zakładamy, że naszej aplikacji będą używały osoby w różnym wieku i zaawansowaniu z komputerem. Z tego powodu widoki aplikacji zostaną zaprojektowane w myśl zasady Accessibility.
* Będą przejrzyste, bez zbędnych dystrakcji.
* Zakładamy wypuszczenie aplikacji na rynek międzynarodowy, dlatego zaimplementujemy tłumaczenia widoków.

1. Baza danych

-- Tworzenie tabeli Projektów

CREATE TABLE Projekty (

ProjektID INT PRIMARY KEY,

Nazwa\_projektu VARCHAR(255),

Opis TEXT,

Data\_rozpoczecia DATE,

Data\_zakonczenia DATE,

Status\_projektu VARCHAR(50)

);

-- Tworzenie tabeli Zespołów

CREATE TABLE Zespoly (

ZespolID INT PRIMARY KEY,

Nazwa\_zespolu VARCHAR(255),

Opis\_zespolu TEXT

);

-- Tworzenie tabeli Czlonkow Zespolow

CREATE TABLE Czlonkowie\_Zespolow (

CzłonekID INT PRIMARY KEY,

Imie VARCHAR(50),

Nazwisko VARCHAR(50),

Email VARCHAR(255),

Rola\_w\_projekcie VARCHAR(50),

ZespolID INT,

FOREIGN KEY (ZespolID) REFERENCES Zespoly(ZespolID)

);

-- Tworzenie tabeli Faz Projektu

CREATE TABLE Fazy\_Projektu (

FazaID INT PRIMARY KEY,

Nazwa\_fazy VARCHAR(255),

Opis\_fazy TEXT

);

-- Tworzenie tabeli Zadania

CREATE TABLE Zadania (

ZadanieID INT PRIMARY KEY,

Nazwa\_zadania VARCHAR(255),

Opis\_zadania TEXT,

Status\_zadania VARCHAR(50),

Termin\_wykonania DATE,

ProjektID INT,

FazaID INT,

PrzypisaneCzlonki INT,

FOREIGN KEY (ProjektID) REFERENCES Projekty(ProjektID),

FOREIGN KEY (FazaID) REFERENCES Fazy\_Projektu(FazaID),

FOREIGN KEY (PrzypisaneCzlonki) REFERENCES Czlonkowie\_Zespolow(CzłonekID)

);

-- Tworzenie tabeli Zależności Zadań

CREATE TABLE Zaleznosci\_Zadan (

ZaleznoscID INT PRIMARY KEY,

ZadanieID INT,

ZadanieZależneID INT,

FOREIGN KEY (ZadanieID) REFERENCES Zadania(ZadanieID),

FOREIGN KEY (ZadanieZależneID) REFERENCES Zadania(ZadanieID)

);

-- Tworzenie tabeli Zasobów Projektowych

CREATE TABLE Zasoby\_Projektowe (

ZasobID INT PRIMARY KEY,

Nazwa\_zasobu VARCHAR(255),

Opis\_zasobu TEXT,

Dostepnosc VARCHAR(50),

ProjektID INT,

UzytkownikID INT,

FOREIGN KEY (ProjektID) REFERENCES Projekty(ProjektID),

FOREIGN KEY (UzytkownikID) REFERENCES Uzytkownicy(UzytkownikID)

);

-- Tworzenie tabeli Komunikacji

CREATE TABLE Komunikacja (

KomunikacjaID INT PRIMARY KEY,

Data\_komunikacji DATETIME,

Nadawca VARCHAR(255),

Odbiorca VARCHAR(255),

Tresc\_komunikatu TEXT,

ProjektID INT,

FOREIGN KEY (ProjektID) REFERENCES Projekty(ProjektID)

);

-- Tworzenie tabeli Zdarzeń Projektu

CREATE TABLE Zdarzenia\_Projektowe (

ZdarzenieID INT PRIMARY KEY,

Data\_zdarzenia DATETIME,

Opis\_zdarzenia TEXT,

ProjektID INT,

FOREIGN KEY (ProjektID) REFERENCES Projekty(ProjektID)

);

-- Tworzenie tabeli Użytkowników

CREATE TABLE Uzytkownicy (

UzytkownikID INT PRIMARY KEY,

Imie VARCHAR(50),

Nazwisko VARCHAR(50),

Email VARCHAR(255),

Haslo VARCHAR(255),

Rola VARCHAR(50),

Data\_utworzenia\_konta DATETIME

);

1. Przedstawienie głównych zasad kodowania

* Aplikacja ASP.NET zostaną napisane w Visual Studio
* JavaScript zostanie napisany zgodnie ze standardem ECMAScript7
* Frontend​ - w celu efektywnego developmentu skorzystamy z Visual Studio Code.

1. Bardziej dokładna identyfikacja ryzyka i opracowanie zasad zarządzania ryzykiem

Przykładowe ryzyka:

1. Biznesowe

a. Brak zainteresowania ze strony firm

b. Powstanie konkurencyjnej aplikacji z lepszą, bogatszą ofertą

c. Brak zainteresowania użytkowników aplikacji

d. Zły model biznesowy

i. Błędne założenie obciążania firm zamiast użytkowników końcowych

ii. Źle dobrane pakiety transferu

iii. Za mała darmowa pula transferu - rezygnacja firm

2. Techniczne

a. Wysoka awaryjność systemu

b. Za mała wydajność serwerów

c. Niezrozumiały interfejs użytkownika

W przypadku małego zainteresowania firm, wprowadzimy całkowicie darmowy okres próbny. W przypadku małego zainteresowania użytkowników, zainwestujemy w dodatkowe reklamy w mediach społecznościowych i kampanie na popularnych stronach internetowych. Model biznesowy można zmienić na podobny do Google. Udostępnimy naszą aplikację całkowicie za darmo, jednak po przyjęciu się na rynku zaczniemy pobierać stopniowo opłaty. Nasza aplikacja będzie w pełni skonteneryzowana oraz oparta na chmurze. W przypadku zbyt małej wydajności serwera, można szybko i w łatwy sposób przenieść się na wydajniejszy model subskrypcji. Zabezpieczy nas to przed przeciążeniem i w rezultacie długim oczekiwaniem na informację zwrotną z serwera.

1. Zmiany względem tablicy koncepcyjnej

Brak większych zmian.