

# **System Przypominajek**

## **Dokumentacja**

### **Wykonawcy:**

Bartłomiej Jędrzejczyk

Jakub Brzegowski

Angelika Chwastek

## **Spis treści**

<b>Zagadnienie biznesowe</b>	<b>3</b>
<b>Wymagania systemowe i funkcjonalne</b>	<b>4</b>
1. Logowanie i konto użytkownika	4
2. Tworzenie notatek i przypomnień	4
3. Wyświetlanie przypomnień	4
4. Baza danych i utrwalanie informacji	4
5. Model architektury	5
<b>Harmonogram prac i zespół projektowy</b>	<b>6</b>
<b>Analiza zagadnienia i jego modelowanie</b>	<b>7</b>
<b>Implementacja i testowanie</b>	<b>8</b>
<b>Podsumowanie</b>	<b>9</b>

## **Zagadnienie biznesowe**

Naszym zagadnieniem biznesowym jest System Przypominajek, po angielsku RemiSystem, mający zaspokoić potrzebę organizacji czasu w coraz bardziej zabieganym świecie. Chcemy stworzyć prosty w obsłudze i lekki serwis, który pozwoli użytkownikowi na zapisywanie notatek oraz dodawanie do nich przypomnień, mających dotrzeć do niego w odpowiednim czasie. System powinien być w stanie osiągnąć użytkownika nawet jeżeli ten go aktywnie nie używa, co chcemy osiągnąć przy pomocy opcji powiadomień przez e-mail.

Proponowana aplikacja zaspokaja wspomnianą potrzebę, ale mogłaby też być wykorzystana w większym systemie jako jedna z funkcji, np. w elektronicznym kalendarzu.

## Wymagania systemowe i funkcjonalne

### 1. Logowanie i konto użytkownika

Możliwość zalogowania się na podstawie nazwy użytkownika oraz hasła. Do każdego konta powinien być przypisany adres e-mail. Użytkownik ma możliwość zmiany hasła oraz ustawiania preferencji: czy chce otrzymywać powiadomienia mailowo czy wyłącznie jako pop-up.

### 2. Tworzenie notatek i przypomnień

Użytkownik zalogowany ma możliwość tworzenia notatek. Notatka powinna zawierać: tytuł oraz opcjonalnie treść i powinna być automatycznie przypisywana danemu użytkownikowi. Nie ma możliwości przypisywania notatek innym użytkownikom.

Po stworzeniu notatki użytkownik ma możliwość dodania do niej przypomnień. Przypomnień może być wiele i mogą być jednym z dwóch typów:

- **zwykłe** – zawiera datę oraz godzinę (z dokładnością do minuty) wyświetlenia przypomnienia.
- **cykliczne** – stworzone w oparciu o format CRON. Będą wyświetlane wielokrotnie w ustalonym przez użytkownika czasie, np. cotygodniowo.

Użytkownik zalogowany ma również możliwość edycji oraz usuwania zarówno swoich notatek jak i przypomnień z nimi powiązanych.

### 3. Wyświetlanie przypomnień

System wyświetla przypomnienia w formie pop-up w przeglądarce o odpowiednim czasie z dokładnością do minuty. Jeżeli użytkownik wybrał sobie taką opcję w preferencjach, system dodatkowo wysła powiadomienie na adres e-mail użytkownika.

### 4. Baza danych i utrwalanie informacji

Firebase będzie przechowywać informacje o użytkownikach takie jak: login, hasło, email. Będzie zwracać id, co umożliwi odnoszenie się do użytkownika w bazie danych.

Baza danych będzie przechowywać informacje o notatkach i przypomnieniach.

Tabela NOTE:

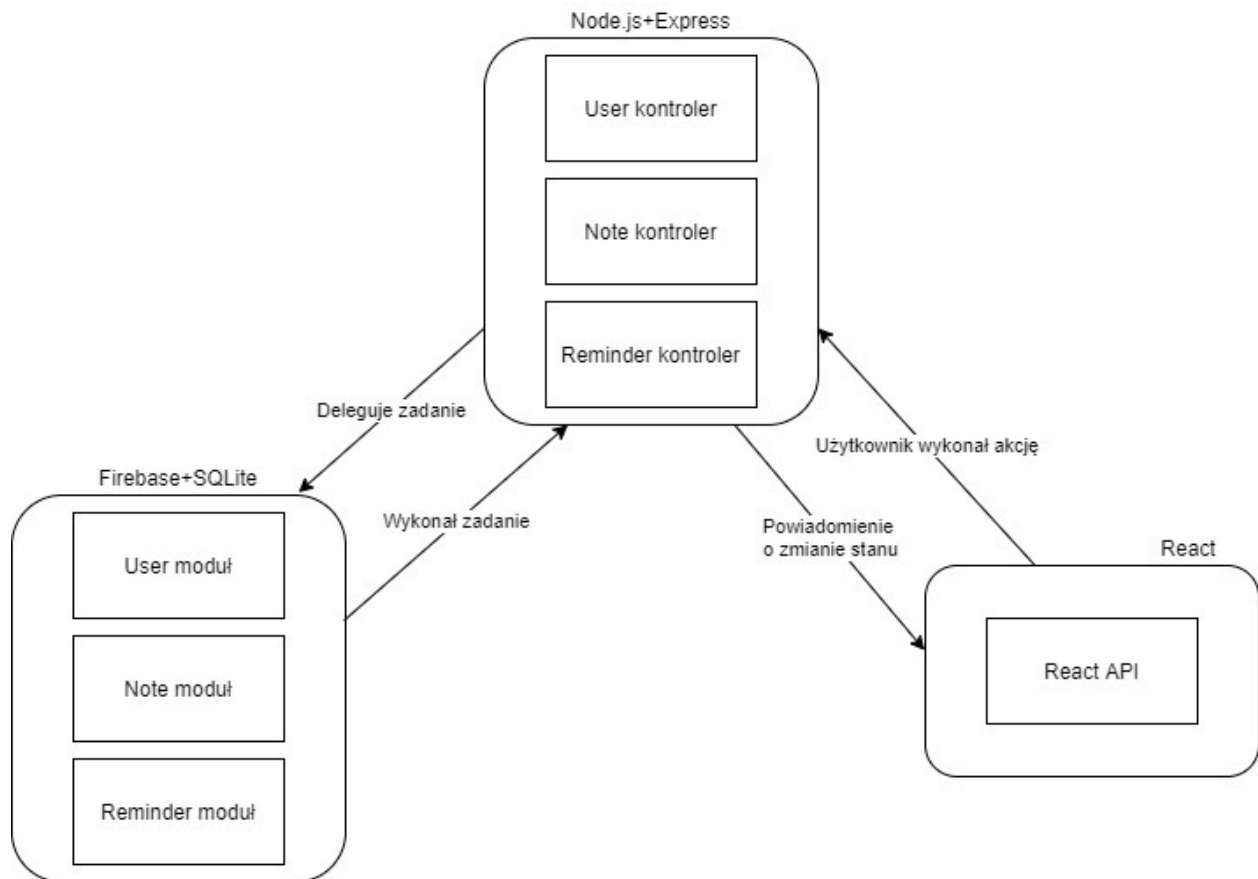
- id – varchar, primary key
- title – varchar, tytuł notatki

- content – varchar, zawartość notatki. Pole nieobowiązkowe.
- user\_id – int, wskazuje na użytkownika, który stworzył notatkę

Tabela REMINDER:

- id – varchar, primary key
- time – varchar (cron w formie stringa), zawiera albo datę przypomnienia albo harmonogram przypominania
- note\_id – int, foreign key; wskazuje na notatkę, której dotyczy przypomnienie

## 5. Model architektury



## Harmonogram prac i zespół projektowy

Data	Wykonawca/wykonawcy	Opis

## **Analiza zagadnienia i jego modelowanie**

## **Implementacja i testowanie**



## **Podsumowanie**