

**Akademia Górniczo-Hutnicza  
im. Stanisława Staszica w Krakowie**

---

Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki

KATEDRA AUTOMATYKI



**ZAAWANSOWANE TECHNOLOGIE  
BAZODANOWE**

**WHaToDo. POMOC W ORGANIZACJI CZASU**

**MARTA DRABARCZYK, KRZYSZTOF KUTT,  
MICHAŁ NOWAK**

Kraków 2012

# Spis treści

<b>1. Wprowadzenie</b>	4
1.1. Motywacja	4
1.2. Opis istniejących rozwiązań	4
1.2.1. Google Calendar	4
1.2.2. Tassky	4
1.2.3. EssentialPIM	5
<b>2. Ogólny opis naszego systemu</b>	6
2.1. Co to jest?	6
2.2. Innowacyjność rozwiązania	6
2.3. Przykładowe zastosowanie	6
<b>3. Zarządzanie czasem</b>	8
3.1. Problem?	8
3.2. Metodyki zarządzania czasem	8
3.3. Pomysły wykorzystane w naszym projekcie	9
<b>4. Analiza problemu</b>	11
4.1. Przypadki użycia	11
4.1.1. Ogólny diagram przypadków użycia	11
4.1.2. UC1: Dodanie zadania	12
4.1.3. UC2: Zmiana priorytetu zadania	12
4.1.4. UC3: Sugerowanie terminu dla zadania	12
4.2. Baza danych	13
<b>5. Metodyka pracy w naszej grupie</b>	14
<b>6. Implementacja</b>	15
6.1. Technologia	15
6.2. Moduły	15
6.3. Algorytmy	16

6.3.1. Sugestia terminu dla wybranego zadania .....	16
6.3.2. Sugestia planu na najbliższy tydzień.....	17
6.4. Warstwa Prezentacji.....	17
<b>7. Podsumowanie .....</b>	<b>18</b>
7.1. Co udało się zrealizować? .....	18
7.2. Napotkane problemy .....	18
7.3. Możliwości dalszego rozwoju projektu .....	18
7.4. Promocja i reklama.....	18

# **1. Wprowadzenie**

## **1.1. Motywacja**

Życie codzienne dostarcza nam niezliczonej ilości informacji, które w dużej mierze związane są z jakimiś zadaniami do wykonania. Zadania te mogą być niewielkie, jak konieczność wykonania telefonu, ale mogą być to też wielogodzinne projekty. Ich cechą wspólną jest na pewno to, że muszą zostać zapamiętane. Pamięć ludzka jest jednak zawodna i często zdarza się, że pamiętamy o czymś przez jakiś czas, a później wylatuje z głowy i nie zostaje wykonane. Naszym projektem jest aplikacja webowa, której głównym celem jest właśnie pomoc w organizacji czasu, poprzez udostępnienie prostego w użyciu narzędzia do zarządzania zadaniami w postaci strony internetowej.

## **1.2. Opis istniejących rozwiązań**

Na rynku dostępnych jest wiele produktów wspomagających organizację czasu. Przed rozpoczęciem projektu dokonaliśmy przeglądu istniejących rozwiązań. Poniżej przedstawiamy opis trzech z nich.

### **1.2.1. Google Calendar**

Aplikacja internetowa skierowana do wszystkich osób potrzebujących kalendarza. Produkt pozwala na używanie kilku kalendarzy (własnych, bądź dzielonych z innymi). Jest możliwość definiowania stałych zadań (powtarzających się co określony czas) i usuwania ich jednorazowych wystąpień. Istnieją wersje na urządzenia mobilne (on- i offline). Oferuje różne możliwości przypomnień o zadaniach (pop-up, mail, sms). Produkt jest całkowicie darmowy. Wykorzystywany z powodu łatwości obsługi i dostępności (posiadając konto mailowe google, automatycznie posiada się konto w innych usługach google, m.in. w kalendarzu).

### **1.2.2. Tassky**

Aplikacja internetowa wspomagająca zarządzanie zadaniami indywidualnie i w grupie. Dostępna zarówno dla komputerów stacjonarnych jak i urządzeń mobilnych. Brak widoku kalendarza, ale jest możliwość przeglądania zadań wpisanych na konkretny dzień. Możliwość

ustalenia priorytetu (wg własnych kryteriów). Możliwość zlecania zadań innym użytkownikom. Korzystanie z serwisu jest płatne wg ustalonego cennika (np 1 osoba przez 1 miesiąc - 2,46 zł brutto).

### **1.2.3. EssentialPIM**

Aplikacja dostępna dla użytkowników systemów z rodziny Windows. Skierowana do osób chcących efektywniej zarządzać swoim kalendarzem i listą zadań do wykonania. Dostępna w dwóch wersjach: podstawowej (darmowej) i profesjonalnej (kosztującej 39,95\$ - podstawowe źródło utrzymania aplikacji), różniących się dostępnymi funkcjonalnościami. Aplikacja pozwala na tworzenie różnokolorowych kalendarzy i list zadań. Pozwala na określenie priorytetu danego zadania (wg zdefiniowanej odgórnie listy) i stopnia ukończenia (procentowo). Można łatwo dokonywać zmiany w kalendarzu, dzięki wsparciu mechanizmu drag and drop. Dodatkowo aplikacja zawiera wsparcie dla notatek i obsługi e-maili (książka adresowa).

## 2. Ogólny opis naszego systemu

### 2.1. Co to jest?

Stworzony system ma być pomocą w organizacji czasu. Użytkownik ma do dyspozycji trzy główne części:

- **listę zadań**, do której w prosty sposób może dodać kolejne zadanie do wykonania,
- **kalendarz**, w którym umieszczane są zadania o określonym terminie realizacji (np zajęcia na uczelni, czy umówiona wizyta u klienta),
- **obszar roboczy**, dzięki któremu można przydzielić odpowiedni priorytet do zadania.

Dostęp do systemu uzyskiwany jest przez przeglądarkę internetową. W przyszłości planowane jest stworzenie odpowiednich arkuszy stylów dostosowanych od urządzeń mobilnych, aby możliwe było również wygodne korzystanie z aplikacji za pomocą telefonów komórkowych.

### 2.2. Innowacyjność rozwiązania

Innowacyjność naszego rozwiązania polega na wykorzystaniu metod wspierających zarządzanie sobą w czasie. Przy ich wykorzystaniu, stworzyliśmy asystenta, który nie tylko przechowuje informacje o naszych zadaniach i przypomina o zbliżających się deadlineach, ale także pozwala na określanie ważności zadań. **Najważniejszą innowacją naszej aplikacji jest proponowanie terminów wykonania obowiązków, które nie mają określonej konkretnej daty realizacji** (na podstawie priorytetów i informacjach o dostępności wolnego czasu).

W tym momencie działa proponowanie terminu dla jednego kolejnego zadania. Planowana jest implementacja proponowania terminów dla zadań na cały najbliższy tydzień: system ocenia, które zadania należy wykonać w jakich terminach, uwzględniając różne priorytety zadań i deadline'y każdego z nich.

### 2.3. Przykładowe zastosowanie

Czwartek rano. Znana i zabiegana dziennikarka, Marzena, upiększa się w łazience. Po godzinnej walce z włosami stwierdza, że nie jest w stanie doprowadzić swojej fryzury do porzą-

danego stanu. Musi iść do fryzjera. Najbliższe kilka dni ma wypełnione różnymi spotkaniami i zadaniami. Nie wie kiedy może sobie na to pozwolić.

Wyciąga telefon, otwiera aplikację, wpisuje nazwę zadania (*wizyta u fryzjera*) i określa jego czas trwania (*szacuje, że zajmie jej to 2 godziny*). Ustala deadline na sobotę na godzinę 19:00 (*na ten czas zaplanowane jest jej najbliższe spotkanie przed kamerą*).

Po kilku sekundach aplikacja proponuje jej zaplanowanie wizyty u fryzjera na piątek na 10:00. Wtedy ma małą dwugodzinną przerwę pomiędzy odprowadzeniem dziecka do przedszkola i spotkaniem ze swoim szefem. Dzwoni do ulubionego zakładu fryzjerskiego i rezerwuje termin na wizytę.

Marzena jest zadowolona, że problem został rozwiązany. Może dalej walczyć o pokój na świecie i rzetelność informacji. Nie musi zaprzątać sobie głowy problemem *kiedy iść do fryzjera?*.

## 3. Zarządzanie czasem

### 3.1. Problem?

Czas jest zasobem ograniczonym w formacie ilościowym, który co więcej nie daje się cofnąć, składować, nie można go kupić, zwielokrotnić, nie daje się niczym zastąpić i nieustannie upływa. Największym problemem ludzi jest marnowanie czasu<sup>1</sup>.

### 3.2. Metodyki zarządzania czasem

W związku z rosnącą ilością zadań i zobowiązań oraz nieroszczeniem się czasu, tworzy się różne sposoby zarządzania czasem tak, aby go jak najlepiej wykorzystać i jak najmniej marnować. Podejścia są różne, niektóre proponują tylko jakieś małe usprawnienia, inne natomiast starają się być kompleksowymi systemami, które pomogą na wszystkie problemy.

Przykładowe propozycje:

- **Lista zadań.** Najprostsza rzecz: wypisujemy wszystko na liście i skreślamy to co już zrobione.
- **Kalendarz.** Druga prosta rzecz: w odpowiednim miejscu (data i godzina) wpisujemy to, co mamy zrobić o danej porze.
- **Zasada Pareto**, mówiąca, że *20% pracy przynosi 80% efektów, a pozostałe 80% działań to tylko 20% efektów*. Zasada ta jest wykorzystywana do przydzielania priorytetów i planowania działań.
- **Prawo Parkinsona:** *praca rozszerza się wprost proporcjonalnie do czasu wyznaczonego do jej wykonania*.
- **Getting Things Done**, rozbudowany, kompleksowy system zarządzania wszystkimi zadaniami jakie są do zrobienia: od prostych i szybkich rzeczy takich jak telefon z życzeniami do babci, po duże, długofalowe projekty.

---

<sup>1</sup>za: [http://pl.wikipedia.org/wiki/Zarządzanie\\_czasem](http://pl.wikipedia.org/wiki/Zarządzanie_czasem), dostęp: 2012-09-21



### 3.3. Pomysły wykorzystane w naszym projekcie

W naszym projekcie skorzystaliśmy z dwóch pomysłów na zarządzanie czasem.

1. **Należy dążyć do „oczyszczenia umysłu”**, czyli zwolnić się z obowiązku pamiętania o wszystkich zobowiązaniach i planach. Jest to myśl zaczerpnięta ze wspomnianej wcześniej metodyki *Getting Things Done*. W tym celu absolutnie każde zadanie należy zapisywać, aby nie trzeba było o nim pamiętać.

W naszej aplikacji zostało to zastosowane w postaci najprostszej możliwości dodawania zadania. W przypadku gdy mamy jakąś szybką myśl, że trzeba coś zrobić, wystarczy, że dodamy zadanie które ma tylko ustaloną nazwę. Dzięki temu nie musimy już pamiętać o tym zadaniu, a dokładniejsze dane (opis, deadline, ...) uzupełnimy np wieczorem, podczas przeglądania kalendarza.

2. **Matryca Eisenhowera** (tab. 3.1). Opiera się na dwóch wymiarach:

- Ważność: ma związek z naszą misją i najważniejszymi celami. Jeśli coś wpływa znacząco na osiągnięcie istotnych dla nas celów, to znaczy, że jest ważne.
- Pilność: to kryterium związane z czasem. Zwraca uwagę na termin wykonania jakiegoś zadania. Data ta może być mniej lub bardziej oddalona w czasie, w związku z tym dzieli nasze aktywności na mniej lub bardziej pilne.

WAŻNE	<b>Zarządzanie kryzysami:</b> ważne telefony, sprawy naglące, zadania z przydzielonym terminem	<b>Zarządzanie samym sobą:</b> tworzenie i realizowanie planów, dbanie o zdrowie, nauka
NIEWAŻNE	<b>Zarządzanie cudzymi priorytetami:</b> zadania zlecone przez kogoś innego, sprawy naglące, ale nieistotne dla naszych celów	<b>Zarządzanie pożeraczami czasu:</b> codzienne przyjemności, „złodzieje czasu”, internet, telewizja
	PILNE	NIEPILNE

Tablica 3.1: Matryca Eisenhowera

Ważne jest, aby najpierw odpowiednio rozdzielić zadania, a następnie odpowiednio rozdzielić siły i czas na aktywności z każdej z ćwiartek. Zadania z „zarządzania kryzysami” należy wykonać teraz, zadania z „zarządzania samym sobą” są najważniejsze z punktu widzenia rozwoju i na nie należy poświęcić najwięcej czasu, zadania z „zarządzania cudzymi priorytetami” należy w miarę możliwości oddelegować innym osobom, a zadania

z „zarządzania pożeraczami czasu” należy ograniczać (ale nie rezygnować z nich całkowicie). Więcej o tej koncepcji można przeczytać w internecie, bądź w książce „7 nawyków skutecznego działania” Stephena Coveya.

W naszym projekcie wykorzystaliśmy matrycę w dwojaki sposób:

- (a) Jest jednym z dwóch, obok kalendarza, głównych widoków naszej aplikacji. Pozwala to użytkownikom w łatwy sposób zarządzać swoimi priorytetami i mieć szybki dostęp do tej matrycy.
- (b) Planowane jest również wykorzystanie matrycy do proponowania planu na najbliższy tydzień. Przygotowany algorytm (zobacz: sekcja 6.3.2, niestety nie został zaimplementowany), na podstawie przydziału zadań do poszczególnych ćwiartek matrycy, miał decydować które zadania należy wykonać w najbliższym tygodniu.

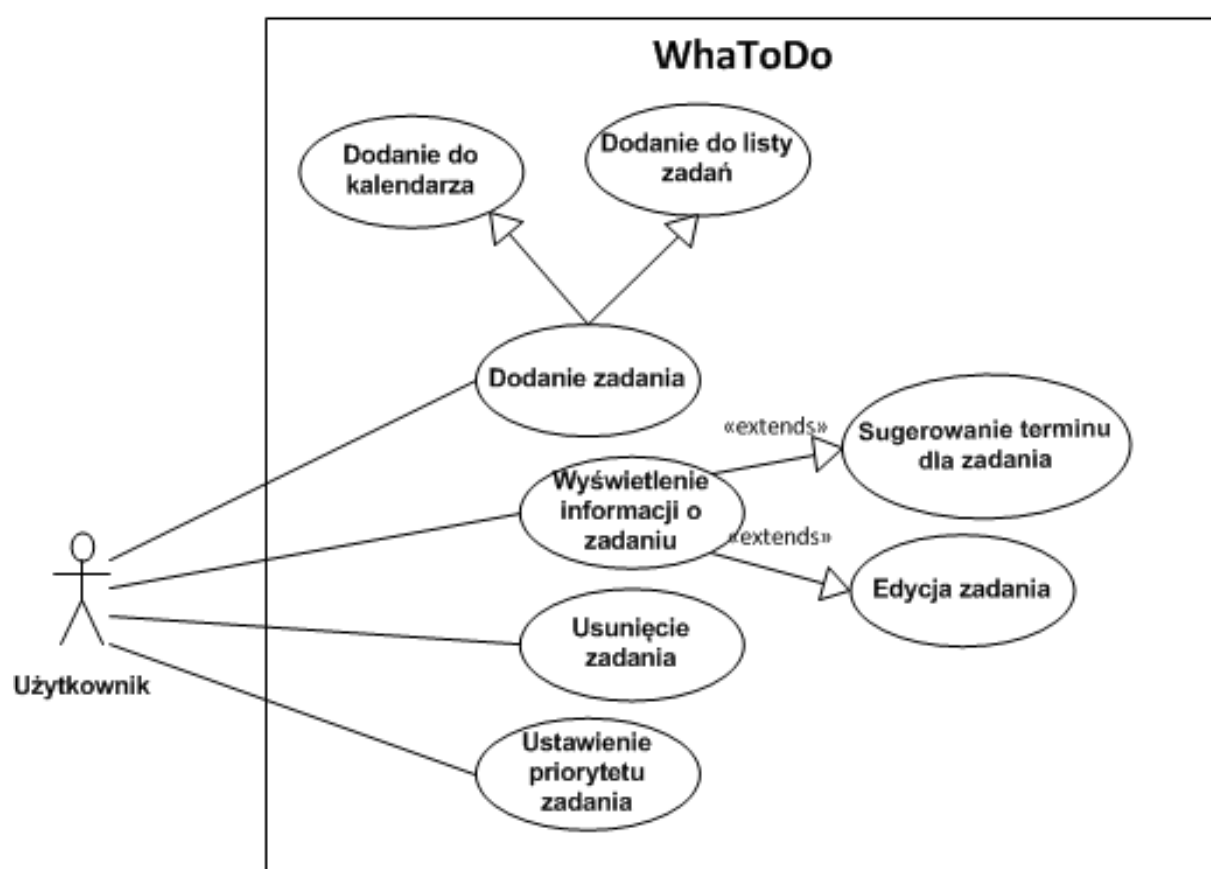
### **Dlaczego zdecydowaliśmy się akurat na te dwa pomysły zarządzania czasem?**

Ponieważ te dwie myśli z powodzeniem stosujemy we własnym życiu. Odciążenie umysłu pozwala cieszyć się tym co jest tu i teraz i nie myśleć o tym co mamy zrobić: to wszystko mamy zapisane. Natomiast podział zadań na priorytety pozwala skupić się na tym co naprawdę ważne, zamiast tylko „gonić za kolejnymi deadline’ami”. Pomaga to na przykład skuteczniej zarządzać swoim czasem w sesji egzaminacyjnej, co potwierdza nasze doświadczenie.

## 4. Analiza problemu

### 4.1. Przypadki użycia

#### 4.1.1. Ogólny diagram przypadków użycia



Rysunek 4.1: Ogólny diagram przypadków użycia

Poniżej definicje trzech wybranych przypadków użycia.

#### **4.1.2. UC1: Dodanie zadania**

**Cel:** użytkownik chce dodać nowe zadanie

**Zdarzenie wyzwalające (trigger):** użytkownik wybiera opcję Dodaj nowe zadanie

**Scenariusz główny:**

1. System wyświetla formularz wprowadzania informacji o zadaniu
2. Użytkownik wprowadza dane (nazwa, opis, czas trwania)
3. System zapisuje dane w bazie danych i odświeża widok (pojawia się nowe zadanie na liście zadań)

#### **4.1.3. UC2: Zmiana priorytetu zadania**

**Cel:** użytkownik chce zmienić priorytet zadania

**Scenariusz główny:**

1. Użytkownik klika na zadanie i przeciąga je w odpowiednie miejsce (na listę zadań, co powoduje usunięcie priorytetu; na ćwiartkę obszaru roboczego - ustawienie ustalenie priorytetu)
2. System zapisuje zmiany w bazie danych

#### **4.1.4. UC3: Sugerowanie terminu dla zadania**

**Cel:** użytkownik chce otrzymać od systemu propozycję terminu dla zadania z listy zadań

**Zdarzenie wyzwalające (trigger):** użytkownik wybiera opcję *Zasugeruj termin zadania*

**Scenariusz główny:**

1. System wyświetla propozycję terminu
2. Użytkownik akceptuje ją
3. System zapisuje zmiany w bazie danych

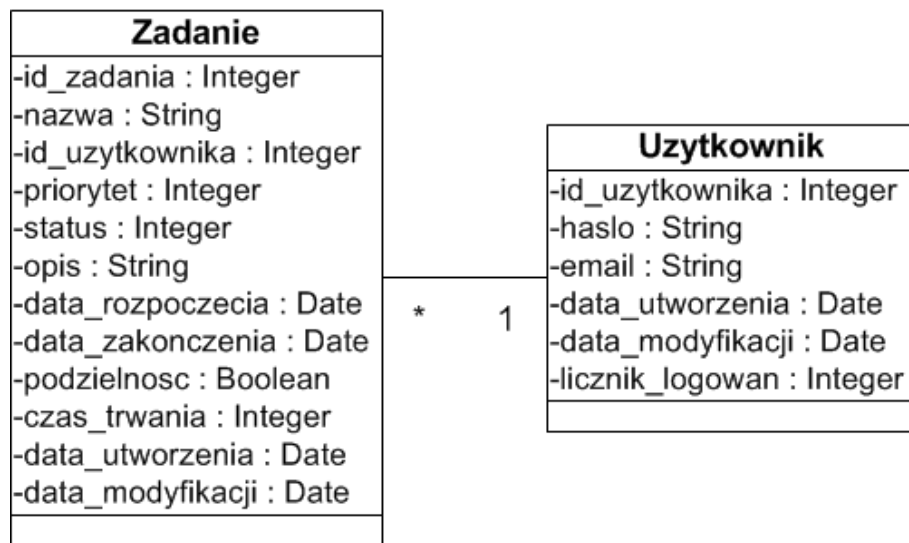
**Scenariusz alternatywny:**

**2A** Użytkownik nieakceptuje propozycji

**2A1** System oblicza inny dostępny termin

**2A2** Scenariusz główny jest kontynuowany od punktu 1.

## 4.2. Baza danych



Rysunek 4.2: Schemat bazy danych

## 5. Metodyka pracy w naszej grupie

### 1. *Jaką metodykę pracy przyjęliśmy?*

Wzorowaliśmy się na metodykach zwinnych. Na nieregularnych spotkaniach, zarówno „przy piwie”, jak i za pośrednictwem internetu, omawialiśmy postępy, problemy i dalsze plany. Każdy podejmował się zadania, które uważał, że jest w stanie wykonać. W przypadku problemów zgłaszaliśmy to w zespole i próbowaliśmy rozwiązać je wspólnie.

### 2. *W jaki sposób się komunikowaliśmy?*

Główne ustalenia dotyczące projektu omawialiśmy na wspólnym spotkaniu przy stole i kartce papieru. Mniejsze uwagi odnośnie poprawy/zmiany/dodania funkcjonalności odnotowywaliśmy w specjalnie do tego celu stworzonym dokumencie na Google Docs. Pilne rozmowy przeprowadzane były za pośrednictwem czatu na Gmailu.

### 3. *Jakie narzędzia wykorzystywaliśmy do synchronizacji?*

W celu synchronizacji tworzonego kodu, wykorzystaliśmy repozytorium założone w serwisie `github.com`

## 6. Implementacja

### 6.1. Technologia

- Cały nasz projekt został stworzony w środowisku **Netbeans**.
- Aplikacja została oparta na frameworku **Ruby on Rails**. Wybór ten pociągnął za sobą dwie główne konsekwencje. Z jednej strony ułatwienie w postaci wymuszenia przez RoR korzystania z wzorca Model-View-Controller oraz w postaci dużej liczby dostępnych modułów i małej liczby potrzebnego kodu. Z drugiej strony było to dla nas dużym utrudnieniem, ponieważ nie mieliśmy wcześniej styczności zarówno z frameworkiem Ruby on Rails, jak i z samym językiem programowania Ruby. Należy tutaj zwrócić uwagę, że nie jest to framework prosty - dużo czasu zajęły nam próby zrozumienia jak on działa.
- Wszystkie informacje o użytkownikach oraz zadaniach przechowywane są w bazie danych **PostgreSQL**.
- Wykorzystano również fragment możliwości oferowanych przez **HTML5**. Konkretnie: tag `<details>` oraz natywny *drag&drop*.

### 6.2. Moduły

System składa się z czterech modułów:

- **Użytkownicy**. Odpowiedzialnego za zarządzanie użytkownikami serwisu: tworzenie i kasowanie kont.
- **Autoryzacja**. Odpowiedzialnego za logowanie i wylogowywanie z aplikacji oraz utrzymywanie sesji.
- **Zadania**. Odpowiedzialnego za tworzenie, edycję, kasowanie i przechowywanie zadań. Również za przydzielanie im priorytetów i proponowanie terminów realizacji (dla zadań, które nie mają ściśle określonego terminu).
- **Kalendarz**. Odpowiedzialnego za wyświetlanie kalendarza, zawierającego zadania przesłane z modułu zadań.

## 6.3. Algorytmy

### 6.3.1. Sugestia terminu dla wybranego zadania

Algorytm otrzymuje na wejściu trzy parametry:

- zadanie (w implementacji jest to konkretnie id zadania)
- godzinę rozpoczęcia i zakończenia danego dnia (*godziny w których można przydzielać zadania; aktualnie są one narzucone z góry i zapisane bezpośrednio w kodzie; w przyszłości użytkownik miałby opcję określić te godziny dla każdego dnia tygodnia*)
- czas rozpoczęcia wyszukiwania (*opcjonalne; domyślnie algorytm rozpoczyna wyszukiwanie terminu dla zadania od „teraz”; tym parametrem można określić od jakiego terminu ma rozpocząć wyszukiwanie*)

1. Jeżeli zadanie ma już określony termin: nic nie rób.
2. Jeżeli zadanie jest dłuższe od jednego dnia roboczego: nic nie rób (*algorytm w podstawowej wersji nie ma możliwości podziału zadań na mniejsze części - planowaliśmy to wprowadzić w kolejnej wersji*).
3. Ustal czas startu na czas rozpoczęcia wyszukiwania, jeżeli ustawiony, w przeciwnym przypadku na aktualny czas. Ustal czas zakończenia na czas startu+czas trwania.
4. Jeżeli czas zakończenia jest późniejszy od deadline'u, zwróć informację o niepowodzeniu i zakończ algorytm.
5. Pobierz wszystkie zadania z ustalonym czasem rozpoczęcia i zakończenia
6. Weź pierwsze zadanie z pobranej listy i sprawdź czy jego termin zachodzi na zakres czas startu-czas zakończenia.
7. Jeżeli terminy na siebie zachodzą, ustal czas rozpoczęcia wyszukiwania na czas zakończenia zadania znalezione w bazie i rozpocznij algorytm od początku.
8. Jeżeli terminy na siebie nie zachodzą i są jeszcze jakieś zadania na liście, wróć do punktu 6.
9. Przedstaw ustalony termin do akceptacji użytkownikowi.
10. W przypadku braku akceptacji, powtórz algorytm. Ustaw czas rozpoczęcia wyszukiwania na czas startu+1 godzina (*1 godzina została wybrana przez nas i nie ma żadnego uzasadnienia. Powoduje tylko i wyłącznie zaproponowanie późniejszego terminu rozpoczęcia zadania*).
11. W przypadku akceptacji, zapisz informacje w bazie.



### 6.3.2. Sugestia planu na najbliższy tydzień

*Algorytm ten nie został zaimplementowany z powodu braku czasu (i początkowo braku umiejętności programowania w Ruby). Tutaj przedstawiamy wersję, którą planowaliśmy zaimplementować w pierwszej wersji aplikacji.*

1. Pobierz listę wszystkich zadań, które nie mają ustalonego terminu.
2. Posortuj listę najpierw wg priorytetów: 1. Ważne-pilne 2. Nieważne-pilne 3. Ważne-niepilne 4. Nieważne-niepilne 5. Bez priorytetu, a następnie wg deadline (od najbliższego do najpóźniejszego).
3. Weź pierwsze zadanie z listy i szukaj dla niego propozycji terminu (wg algorytmu z sekcji 6.3.1).
4. Zapisz w lokalnej tablicy propozycję terminu dla tego zadania (tablica indeksowana po id tasków).
5. Powtórz kroki 3 i 4 do wyczerpania listy zadań, bądź do wyczerpania czasu w tygodniu.
6. Zaprezentuj znalezione terminy użytkownikowi.
7. W przypadku braku akceptacji, powtórz algorytm. Tym razem dla pierwszego zadania wywołaj algorytm sugestii terminu z ustawionym czasem rozpoczęcia wyszukiwania na teraz + random(1:24) godzin (dzięki temu zostaną przedstawione inne propozycje).
8. W przypadku akceptacji, zapisz informacje w bazie.

## 6.4. Warstwa Prezentacji

TODO: Tutaj krótki opis i kilka zrzutów ekranu

## 7. Podsumowanie

### 7.1. Co udało się zrealizować?

TODO: Czy jesteśmy zadowoleni i takie tam.

### 7.2. Napotkane problemy

TODO: Chociażby to, że Ruby jest dla nas nieznaną technologią.

### 7.3. Możliwości dalszego rozwoju projektu

TODO: Trzeba zaglądnąć w listę rzeczy, które mamy w specyfikacji i jakoś opisać + może mamy jakieś inne pomysły jeszcze ;)

### 7.4. Promocja i reklama

Portal nazwaliśmy „whaTodo”, co jest skrótem od angielskiego „what to do”. Nazwa ta wskazuje na główną innowację naszej aplikacji czyli sugerowanie terminów wykonywania zadań, które nie mają ściśle określonego czasu wykonania.

Po dopracowaniu serwisu i zakupieniu domeny, chcieliśmy go udostępnić dla wszystkich zainteresowanych. W ramach promocji naszego serwisu przez pierwszy miesiąc dostęp do wszystkich funkcjonalności byłby darmowy. W kolejnych miesiącach zakładaliśmy użytkowanie aplikacji za symboliczne 2 zł / miesiąc.

Hasło reklamowe: „**Nie wiesz za które zadanie się teraz zabrać? Wejdź na [www.whatodo.pl](http://www.whatodo.pl) i pozwól sobie pomóc!**”