

# Wprowadzenie do zarządzania projektami deweloperskimi

laboratorium

## *Zadanie G*



Jakub Engielski  
Jan Kwiatkowski  
Szymon Michno  
Michał Myśków  
Anna Sokołowska

# Spis treści:

<b>1. Temat oraz opis projektu</b>	<b>3</b>
a)Temat projektu	3
b)Opis projektu	3
<b>2. Grupa Projektowa:</b>	<b>4</b>
<b>3. Założenia Funkcjonalne:</b>	<b>4</b>
a)Użytkownik powinien móc:	4
b)Administrator serwisu powinien móc:	5
c)System powinien:	5
<b>4. Technologie oraz narzędzia:</b>	<b>5</b>
<b>5. Opis działania aplikacji / systemu</b>	<b>7</b>
<b>6. Event Storming</b>	<b>7</b>
Opis Event Stormingu:	7
Kolejność modelowania:	8
Zrzuty ekranu z modelowania:	9
<b>7. Backlog</b>	<b>11</b>
<b>8. Estymata</b>	<b>13</b>
<b>9. Podział na sprinty</b>	<b>16</b>
Sprint 1:	16
Sprint 2:	16
Sprint 3:	16
Sprint 4:	16
<b>10. Zrzuty Ekranu z podziału sprintów:</b>	<b>17</b>
<b>11. Interfejs Aplikacji</b>	<b>19</b>
Zrzuty Ekranu	19
Opis	22
<b>12. Informacje uruchomieniowe w środowisku deweloperskim</b>	<b>23</b>
<b>13. Opis wdrożenia aplikacji i sposobu jej hostowania</b>	<b>25</b>
Opis wdrożenia:	25
Opis sposobu hostowania:	25
<b>14. Podsumowanie oraz Wnioski</b>	<b>26</b>

# **1. Temat oraz opis projektu**

## **a)Temat projektu**

Wedding-planner

## **b)Opis projektu**

Celem naszego projektu jest:

Stworzenie aplikacji do planowania ślubów która będzie miała w sobie takie elementy jak rejestr gości z potwierdzeniem przybycia oraz możliwością ręcznej modyfikacji tej funkcji przez państwo młodych, planer miejsc siedzących oraz umożliwi stworzenie własnej listy rzeczy do wykonania.

Państwo młodzi otrzymają także unikalny kod, wysyłany wraz z zaproszeniem, który pozwoli gościom na podgląd jak wyglądają miejsca siedzące.

## 2. Grupa Projektowa:

- Jakub Engielski - Leader, Project Manager, Frontend
- Jan Kwiatkowski - Frontend, Backend
- Szymon Michno - Tester
- Michał Myśków - Dev-ops, Backend
- Anna Sokołowska - Frontend, Backend

## 3. Założenia Funkcjonalne:

### a)Użytkownik powinien móc:

1. Zarejestrować się – także przy użyciu social media
2. Zalogować się – także przy użyciu social media
3. Zresetować hasło w wypadku zapomnienia
4. Wybrać rodzaj Sali, z wcześniej przygotowanej listy (mała, średnia, duża)
5. Wprowadzić adres odbywania się wesela.
6. Wprowadzić datę wesela
7. Wprowadzić dane gości, oraz ew. konflikty między nimi
8. Usuwanie gości
9. Edycja danych gości
10. Wybrać opcję manualnego bądź automatycznego przydzielenia miejsc
11. Mieć możliwość manualnej edycji przydzielonych miejsc
12. Mieć możliwość wysłania zaproszeń mailem.
13. Gość powinien mieć możliwość użycia link w zaproszeniu by potwierdzić swoją obecność.
14. Klient powinien móc mieć możliwość ręcznej edycji potwierdzenia przyjsia.
15. Wygenerować gotowy tekst zaproszenia do wydruku 16. Móc przygotować checklistę co musi jeszcze przygotować na swoje wesele

17. Móc skorzystać z kalkulatora kosztów wesela (wszystkie dane wprowadzane przez użytkownika)
18. Gość powinien móc obejrzeć przydzielone miejsca

### **b) Administrator serwisu powinien móc:**

1. Przejrzeć logi administracyjne
2. Mieć dostęp do danych wszystkich użytkowników i ich danych.
3. Móc przeprowadzać akcje administracyjne dotyczące użytkowników.

### **c) System powinien:**

1. Sprawdzać i wyświetlać ostrzeżenie gdy dwie nie lubiące się osoby są przydzielone do miejsc obok siebie
2. Jeśli nie da się rozłożyć bezpiecznie nie lubiących się osób, zignorować zasadę z wyświetleniem ostrzeżenia
3. Zapisywać wszystkie wprowadzane dane do bazy danych w odpowiednich tabelach

## **4. Technologie oraz narzędzia:**

- PHP - Ze względu na szeroką dostępność urządzeń obsługujących przeglądarki internetowe stwierdziliśmy że zrobimy aplikację internetową. Więc wykorzystamy jeden z najsłynniejszych języków backendowych z którym nasza grupa miała do tej pory największą styczność.
- Symfony 5.3.9 - Framework webowy pozwalający na zautomatyzowanie typowych czynności wykonywanych w czasie tworzenia stron internetowych. Pozwala on wykorzystać nam gotowe biblioteki do większości bazowych funkcji aplikacji internetowych co oszczędza nam dużo czasu. Nie musimy pisać od nowa funkcji, które od lat były optymalizowane oraz udoskonalane do niemal perfekcji. Wybraliśmy Symfony zamiast na przykład Laravela gdyż chcieliśmy spróbować coś czego nie

używaliśmy wcześniej przez co poszerzyć nasze możliwości.

- MariaDB (MySQL) - System zarządzania relacyjną bazą danych. MariaDB ma znacznie większą liczbę nowych funkcji w odróżnieniu od MySQL'a, co czyni ją lepszą pod względem wydajności i orientacji na użytkownika.
- Twig - pozwala nam on na używanie szablonów, które nie tylko ułatwiają i przyspieszają pisanie kodu PHP ale czynią go też o wiele bardziej czytelnym.
- Docker - Umożliwi nam łatwe pakowanie, dostarczanie i uruchomienie aplikacji jako lekkiego, przenośnego, samowystarczającego kontenera, który może działać praktycznie wszędzie.
- PhpStorm - To lekkie i inteligentne środowisko programistyczne z którego będziemy korzystać podczas pisania kodu w języku programowania PHP. To środowisko zawiera intuicyjny, szybki i wygodny edytor kodu oraz pozwala na debugowanie.
- Trello - jest to wygodna tablica pozwalająca nam na organizację pracy poprzez przypisywanie zadań do konkretnych osób. Możemy też nadać konkretnym zadaniom wagę w zależności od ich ważności, oraz dzięki temu mamy czytelny backlog zadań.
- Discord - komunikator głosowo-tekstowy pozwalający nam na łatwe utrzymanie kontaktu w grupie. Możemy też udostępniać tam zrzuty ekranu oraz inne ważne informacje.  
Oraz pozwala nam on na udostępnianie ekranu na żywo dzięki czemu zwiększa to nam wygodę pracy.
- Git - system kontroli wersji, pozwalający wielu użytkownikom pracować na różnych komputerach bez zaburzania przepływu pracy. Dzięki temu jesteśmy w jakiś logiczny sposób prowadzić pracę zdalną w zespole. Pozwala on też na automatyczne przeprowadzanie testów przy commitach.

## 5. Opis działania aplikacji / systemu

Aplikacja po rejestracji lub zalogowaniu umożliwia dostęp do większości funkcjonalności. Niezalogowany użytkownik posiada możliwość zalogowania (także poprzez SocialMedia), zarejestrowania się lub zresetowania hasła.

Po uzyskaniu dostępu do aplikacji możemy wykonać takie akcje jak:

- Edycja danych zalogowanego użytkownika
- Tworzenie oraz edycja wesela (wybór sali i daty wraz z weryfikacją dostępności)
- Dodawanie, edycja oraz zapraszanie gości do wcześniej utworzonego wesela
- Potwierdzanie obecności gości na weselu poprzez link w wiadomości email lub manualne przełączenie przez organizatora
- Generowanie plików zaproszeń RSVP
- Automatyczne oraz manualne przydzielanie miejsc dla gości
- Skorzystanie z systemu checklisty (listy zadań do wykonania przy organizacji wesela). Umożliwia on dodawanie, usuwanie, edycję oraz oznaczenie wykonania zadania.
- Skorzystanie z kalkulatora kosztów wesela. Umożliwia on dodawanie, usuwanie, edycję oraz zsumowanie wszystkich kosztów wesela.

## 6. Event Storming

### a) Opis Event Stormingu:

- 1) Określiłmy nasz główny cel
- 2) Określiłmy legendę
- 3) Określiłmy aktorów (rodzaje użytkowników)
- 4) Podzieliliśmy proces na etapy oraz zadania
- 5) Zaczęliśmy rozpisywać wykonane akcje, kategoryzując je jako użytkownika bądź serwera

- 6) Przeprowadziliśmy ścieżki konkretnych użytkowników przez etapy tego potrzebujące

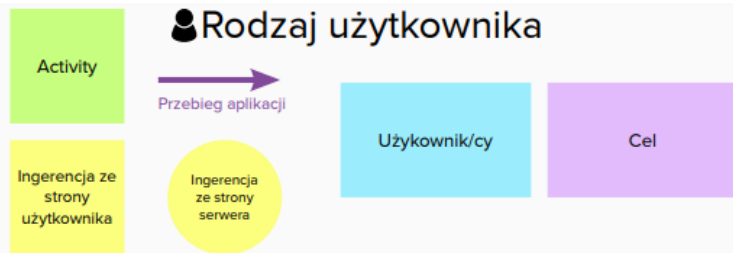
**b) Kolejność modelowania:**

- 1) Proces logowania/rejestracji
- 2) Tworzenie wesela
- 3) Zarządzanie Weselem
- 4) RSVP
- 5) Użytkowanie strony
- 6) Podgląd miejsc



## c) Zrzuty ekranu z modelowania:

### 👤 Rodzaj użytkownika

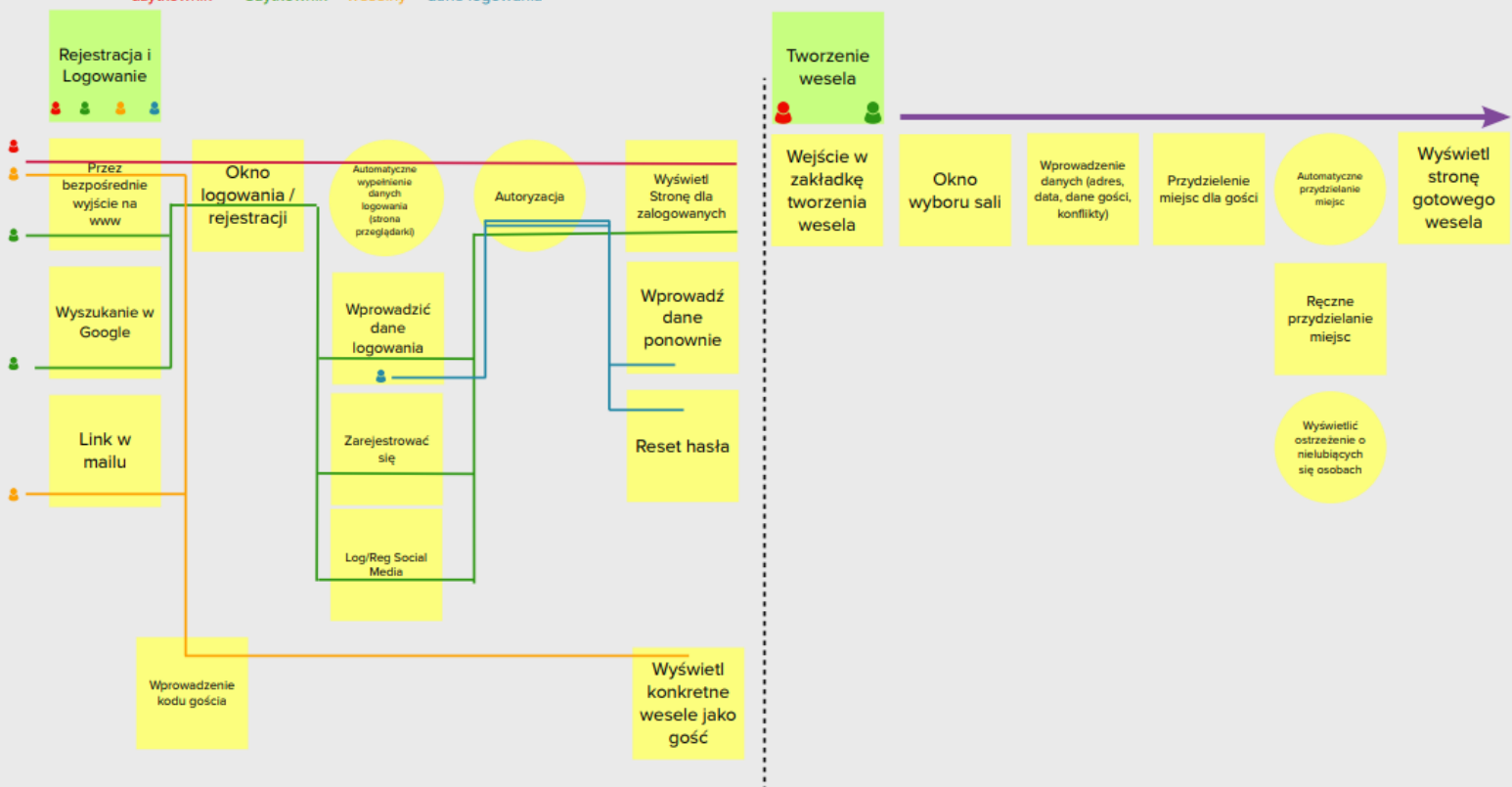


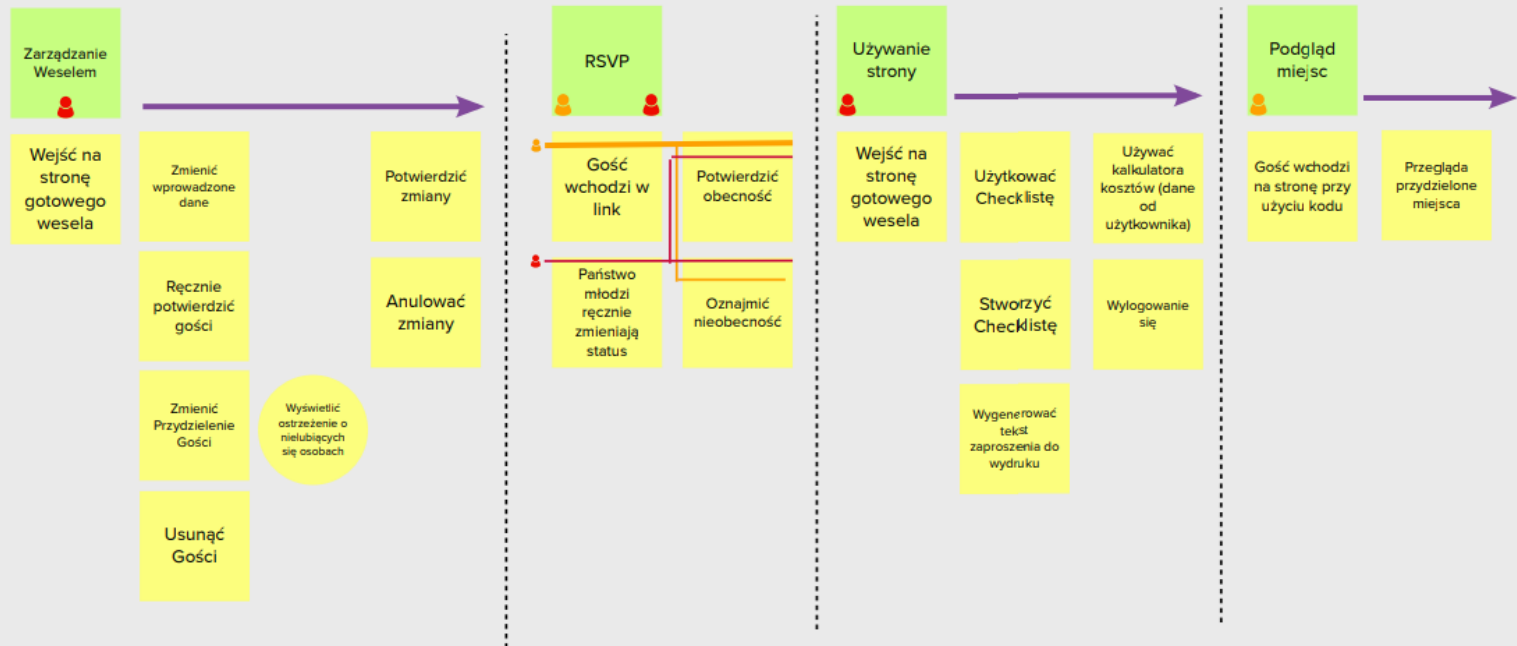
## WeddingPlanner

Janusz i Grażyna -  
Państwo Młodzi

Organizacja Wesela

👤 Powracający użytkownik  
👤 Nowy Użytkownik  
👤 Gość weselny  
👤 Żle wprowadzone dane logowania





## 7. Backlog

1. Dockeryzacja środowiska (utworzenie pliku docker-compose)
2. Dokumentacja uruchamiania aplikacji w środowisku deweloperskim
3. Utworzenie szkieletu aplikacji Symfony
4. Utworzenie modelu bazy danych
5. Utworzenie podstawowego interfejsu testowego
6. Zaimplementowanie systemu logowania/rejestracji
7. Dodanie funkcjonalności logowania/rejestrowania przy pomocy social media
8. Zaimplementować opcję resetowania hasła
9. Zdefiniowanie 3 rodzajów sal (mała, średnia, duża)
10. Zaimplementowanie systemu tworzenia wesela:
  - a. Wprowadzanie adresu
  - b. Wprowadzanie danych gości
  - c. Wprowadzanie daty wesela
  - d. Wprowadzenie danych dotyczących konfliktów między gośćmi
  - e. Wybór sposobu przydziału miejsc (automatyczny/manualny)
  - f. Utworzenie algorytmu przydzielającego miejsca
  - g. Utworzenie opcji wyświetlającej konflikt między gośćmi w postaci ostrzeżenia
  - h. Utworzenie prostego interfejsu użytkownika dla głównej strony wesela
11. Zaimplementowanie systemu edycji wcześniej utworzonego wesela:
  - a. Usuwanie gości
  - b. Edycja danych gości
  - c. Zmienić wcześniej wprowadzone dane wesela (tj. data, miejsce)
  - d. Zmienić ręcznie przydzielenie miejsc gości
12. Utworzenie systemu RSVP:
  - a. Ręczna edycja statusu potwierdzenia obecności
  - b. Możliwość wysłania zaproszeń mailem

- c. Możliwość wydrukowania gotowego tekstu zaproszenia
  - d. Potwierdzenie bądź zaprzeczenie obecności poprzez wejście w link dostarczony w mailu
13. Zaimplementowanie systemu checklisty, utworzonej przez użytkownika dla śledzenia postępu z weselem
  14. Utworzenie kalkulatora kosztu weselów i danie możliwości używania go przez użytkowników
  15. Utworzyć CRUD dla administratora, który da mu wygodny podgląd do danych wszystkich użytkowników
  16. Dodać opcję przeprowadzania akcji administracyjnych w CRUD-zie
  17. W wypadku zbyt dużej ilości konfliktów przy przydzielaniu automatycznym miejsc, zaimplementować wyjątek, który ustawi gości losowo z wyświetleniem ostrzeżenia
  18. Dać możliwość podglądu przydzielonych miejsc dla gości poprzez użycie unikalnego kodu wesela
  19. Implementacja i przeprowadzenie testów
  20. Utworzenie ostatecznej wersji interfejsu użytkownika
  21. Przygotowanie serwera, na którym postawiony zostanie projekt
  22. Zautomatyzowanie testów (najpewniej poprzez funkcje wbudowane w githubie)
  23. Deploy aplikacji na wcześniej przygotowany serwer.
  24. Konfiguracja serwera pod nasz projekt
  25. Przetestowanie aplikacji w środowisku produkcyjnym
  26. Publikacja aplikacji

## 8. Estymata

Poszczególne zadania:	Jakub Engielski	Michał Myśków	Jan Kwiatkowski	Anna Sokołowska	Szymon Michno	Średnia
Dockeryzacja środowiska (utworzenie pliku docker-compose)	2:00	2:00	4:00	1:30	4:00	2:42
Dokumentacja uruchamiania aplikacji w środowisku deweloperskim	1:00	0:30	1:00	1:00	1:00	0:54
Utworzenie szkieletu aplikacji Symfony	5:00	1:00	5:00	1:00	5:00	3:24
Utworzenie modelu bazy danych	1:30	2:00	1:00	1:30	2:00	1:36
Utworzenie podstawowego interfejsu testowego	0:30	1:00	4:00	1:00	2:00	1:42
Zaimplementowanie systemu logowania/rejestracji	4:30	5:00	4:00	3:30	5:00	4:24
Dodanie funkcjonalności logowania/rejestrowania przy pomocy social media	2:00	3:00	3:00	3:00	3:00	2:48
Zaimplementować opcję resetowania hasła	1:00	3:00	3:00	2:00	3:00	2:24
Zdefiniowanie 3 rodzajów sal (mała, średnia, duża)	0:30	1:30	2:00	1:30	2:00	1:30
Zaimplementowanie systemu tworzenia wesela:	36:00	26:30	18:30	21:10	32:30	26:56
Wprowadzanie adresu	1:00	0:30	0:30	0:20	1:00	0:40
Wprowadzanie danych gości	3:00	1:30	0:30	0:40	2:00	1:32
Wprowadzanie daty wesela	0:30	0:30	0:30	0:20	0:30	0:28
Wprowadzenie danych dotyczących konfliktów między gośćmi	6:00	3:00	1:00	1:30	4:00	3:06
Wybór sposobu przydziału miejsc (automatyczny/manualny)	8:00	04:00	2:00	0:20	4:00	3:40
Utworzenie algorytmu przydzielającego	4:00	5:00	5:00	6:00	7:00	5:24

miejsca						
Utworzenie opcji wyświetlającej konflikty między gośćmi w postaci ostrzeżenia	8:00	6:00	4:00	4:00	6:00	5:36
Utworzenie prostego interfejsu użytkownika dla głównej strony wesela	5:30	6:00	5:00	8:00	8:00	6:30
Zaimplementowanie systemu edycji wcześniej utworzonego wesela:	14:00	12:00	13:00	9:50	15:00	12:46
Usuwanie gości	01:30	2:00	3:00	00:30	2:00	1:48
Edycja danych gości	1:30	1:00	2:00	1:40	2:00	1:38
Zmienić wcześniej wprowadzone dane wesela (tj. data, miejsce)	3:00	3:00	2:00	1:40	3:00	2:32
Zmienić ręcznie przydzielenie miejsc gości	8:00	6:00	6:00	6:00	8:00	6:48
Utworzenie systemu RSVP:	16:00	13:00	13:40	12:00	14:00	13:44
Ręczna edycja statusu potwierdzenia obecności	2:00	2:00	2:00	2:00	2:00	2:00
Możliwość wysłania zaproszeń mailem	2:00	3:00	2:40	3:00	3:00	2:44
Możliwość wydrukowania gotowego tekstu zaproszenia	6:00	3:00	3:00	3:00	3:00	3:36
Potwierdzenie bądź zaprzeczenie obecności poprzez wejście w link dostarczony w mailu	6:00	5:00	6:00	4:00	6:00	5:24
Zaimplementowanie systemu checklisty, utworzonej przez użytkownika dla śledzenia postępu z weselem	8:00	7:00	8:00	5:00	8:00	7:12
Utworzenie kalkulatora kosztu weselów i danie możliwości używania go przez użytkowników	12:00	08:00	10:00	8:00	10:00	9:36
Utworzyć CRUD dla administratora, który da mu wygodny podgląd do danych wszystkich użytkowników	10:00	12:00	14:00	10:00	12:00	11:36

Dodać opcję przeprowadzania akcji administracyjnych w CRUD-zie	4:00	05:00	5:00	6:00	6:00	5:12
W wypadku zbyt dużej ilości konfliktów przy przydzielaniu automatycznym miejsc, zaimplementować wyjątek, który ustawi gości losowo z wyświetleniem ostrzeżenia	2:00	03:00	4:00	3:30	4:00	3:18
Dać możliwość podglądu przydzielonych miejsc dla gości poprzez użycie unikalnego kodu wesela	8:00	09:00	7:00	10:00	10:00	8:48
Implementacja i przeprowadzenie testów	8:00	06:00	8:00	8:00	10:00	8:00
Utworzenie ostatecznej wersji interfejsu użytkownika	72:00	50:00	58:00	60:00	70:00	62:00
Przygotowanie serwera, na którym postawiony zostanie projekt	2:00	03:00	4:00	3:00	2:00	2:48
Zautomatyzowanie testów (najpewniej poprzez funkcje wbudowane w githubie)	6:00	05:00	8:00	5:00	8:00	6:24
Deploy aplikacji na wcześniej przygotowany serwer.	2:00	03:00	2:00	3:00	3:00	2:36
Konfiguracja serwera pod nasz projekt	10:00	06:00	04:00	4:00	5:00	5:48
Przetestowanie aplikacji w środowisku produkcyjnym	4:00	5:00	4:00	6:00	6:00	5:00
Publikacja aplikacji	1:00	1:00	0:30	2:00	1:00	1:06
<b>Suma Potrzebnego Czasu:</b>	<b>233:00</b>	<b>193:30</b>	<b>208:40</b>	<b>192:30</b>	<b>243:30</b>	<b>214:14</b>

## 9. Podział na sprinty

Czas trwania poszczególnych sprintów: 2 tygodnie

### a) Sprint 1:

- Dokeryzacja środowiska (docker-compose)
- Utworzenie modelu bazy danych
- Utworzenie szkieletu aplikacji Symfony
- Utworzenie podstawowego interfejsu testowego
- Zaimplementowanie systemu logowania/rejestracji
- Dokumentacja uruchamiania aplikacji w środowisku deweloperskim

### b) Sprint 2:

- Zdefiniowanie 3 rodzajów sal (mała, średnia, duża)
- Zaimplementowanie systemu tworzenia wesela:
- Dodanie funkcjonalności logowania/rejestrowania przy pomocy social media
- Utworzenie systemu RSVP
- Utworzyć CRUD dla administratora, który da mu wygodny podgląd do danych wszystkich użytkowników
- Dodać opcję przeprowadzania akcji administracyjnych w CRUD-zie
- Zaimplementować opcję resetowania hasła

### c) Sprint 3:

- W wypadku zbyt dużej ilości konfliktów przy przydzielaniu automatycznym miejsc, zaimplementować wyjątek, który ustawi gości losowo z wyświetleniem ostrzeżenia
- Zaimplementowanie systemu edycji wcześniej utworzonego wesela
- Przygotowanie serwera, na którym postawiony zostanie projekt
- Konfiguracja serwera pod nasz projekt
- Dać możliwość podglądu przydzielonych miejsc dla gości poprzez użycie unikalnego kodu wesela
- Zaimplementowanie systemu checklisty, utworzonej przez użytkownika dla śledzenia postępu z weselem
- Utworzenie kalkulatora kosztu weselów i danie możliwości używania go przez użytkowników

### d) Sprint 4:

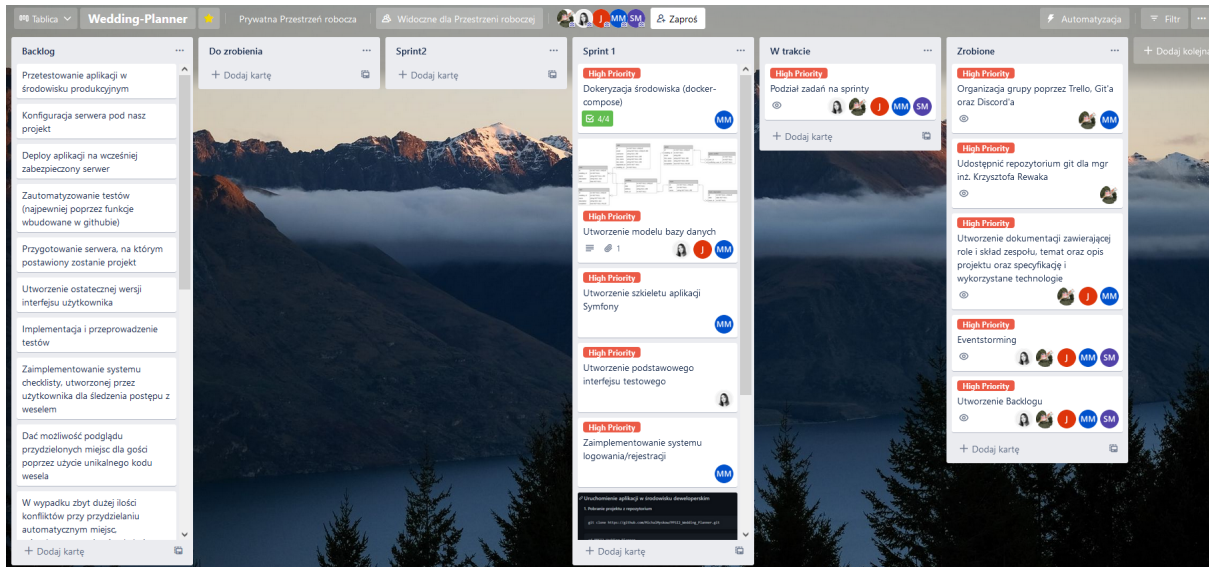
- Utworzenie ostatecznej wersji interfejsu użytkownika
- Implementacja i przeprowadzenie testów
- Zautomatyzowanie testów (najpewniej poprzez funkcje wbudowane w githubie)



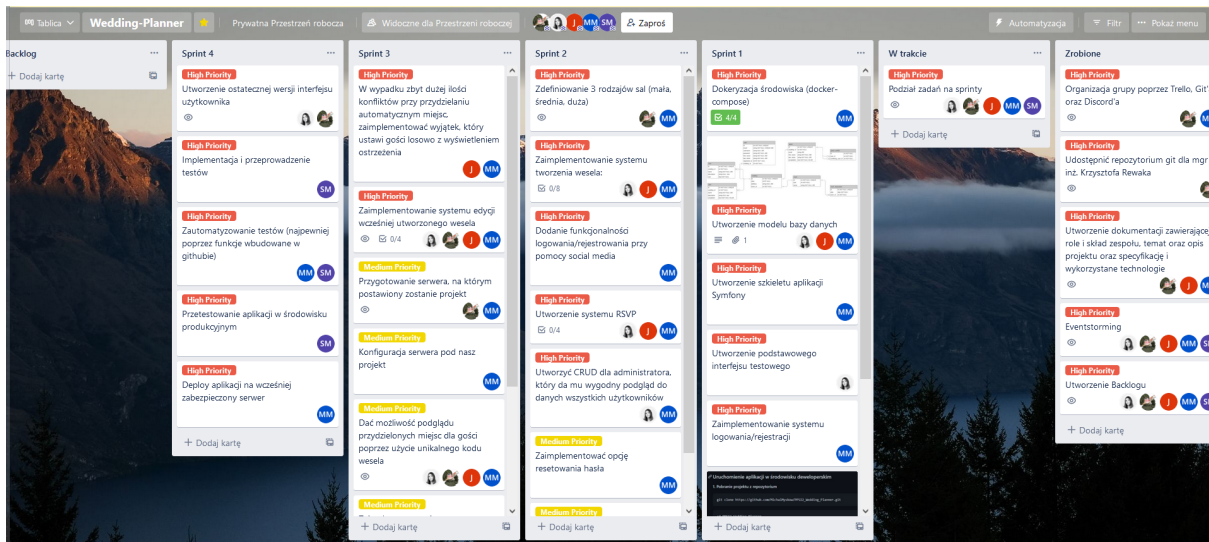
- Przetestowanie aplikacji w środowisku produkcyjnym
- Deploy aplikacji na wcześniej zabezpieczony serwer

## 10. Zrzuty Ekranu z podziału sprintów:

W trakcie rozpisywania zadań:



Zadania już przydzielone:



The image displays a Jira project board for the 'Wedding-Planner' project. The board is organized into four sprints, each with a list of tasks categorized by priority (High, Medium, Low). The tasks are represented as cards with icons indicating their status and assignees.

**Sprint 4:**

- High Priority:** Utworzenie ostatecznej wersji interfejsu użytkownika
- High Priority:** Implementacja i przeprowadzenie testów
- High Priority:** Zautomatyzowanie testów (najpewniej poprzez funkcję wbudowane w githubie)
- High Priority:** Przetestowanie aplikacji w środowisku produkcyjnym
- High Priority:** Deploy aplikacji na wcześniej zabezpieczony serwer

**Sprint 3:**

- High Priority:** W wypadku zbyt dużej ilości konfliktów przy przydzielaniu automatycznym miejsc, zaimplementować wyjątek, który ustawi gości losowo z wyświetleniem ostrzeżenia
- High Priority:** Zaimplementowanie systemu edycji wcześniej utworzonego wesela
- Medium Priority:** Przygotowanie serwera, na którym postawiony zostanie projekt
- Medium Priority:** Konfiguracja serwera pod nasz projekt

**Sprint 2:**

- High Priority:** Zdefiniowanie 3 rodzajów sal (mała, średnia, duża)
- High Priority:** Zaimplementowanie systemu tworzenia wesela:
- High Priority:** Dodanie funkcjonalności logowania/rejestrowania przy pomocy social media
- High Priority:** Utworzenie systemu RSVP
- High Priority:** Utworzyć CRUD dla administratora, który da mu wygodny podgląd do danych wszystkich użytkowników

**Sprint 1:**

- High Priority:** Dokonywanie środowiska (docker-compose)
- High Priority:** Utworzenie modelu bazy danych
- High Priority:** Utworzenie szkieletu aplikacji Symfony
- High Priority:** Utworzenie podstawowego interfejsu testowego

The board also includes a 'Backlog' section on the left and a 'Zakończony Sprint' section on the right. The background of the board features a scenic image of a mountain range.

# 11. Interfejs Aplikacji

## a) Zrzuty Ekranu

The screenshot shows the login interface of the 'Wedding planner' application. The title 'Wedding planner' is displayed in a large, elegant, gold-colored script font at the top center. Below it, the word 'Logowanie' (Login) is written in a smaller, gold-colored script font and underlined. There are two input fields: 'Adres E-mail' and 'Hasło' (Password), both with gold-colored borders. Below the password field is a 'Zaloguj się' (Login) button in gold script. At the bottom, there are links for 'Nie masz konta? Zarejestruj się lub Zaloguj się przez Facebook' and 'Nie pamiętasz hasła?'.

The screenshot shows the registration interface of the 'Wedding planner' application. The title 'Wedding planner' is displayed in a large, elegant, gold-colored script font at the top center. Below it, the word 'Rejestracja' (Registration) is written in a smaller, gold-colored script font and underlined. There are three input fields: 'Adres E-mail', 'Hasło', and 'Nazwa użytkownika', all with gold-colored borders. Below the 'Nazwa użytkownika' field is a checkbox labeled 'Akceptuję politykę prywatności'. Below the checkbox is a 'Zarejestruj się' (Register) button in gold script. At the bottom, there are links for 'Masz już konto? Zaloguj się' and 'Nie pamiętasz hasła?'.

Do wesela zostało: **185 dni!**

👤 Goście

Liczba zaproszonych gości: **56**  
Przyjęte zaproszenia: **0**

## Tworzenie wesela

Imię panny młodej

Nazwisko panny młodej

Imię pana młodego

Nazwisko pana młodego

Planowana data wesela

1 ▼ sty ▼ 2022 ▼  
00 ▼ 00 ▼

Sala weselna

mała (50 miejsc) - Legnica ▼

Stwórz wesele

## Dane wesela

Data wesela: 04.06.2022  
Panna młoda: Marzanna Kowalska  
Pan młody: Grzegorz Bręczyszczykiewicz  
Sala weselna: średnia (100 miejsc) - Wrocław  
[Zmień dane wesela](#)



## Goście

### Wyślij zaproszenia

 Dodaj gościa

Myśków Michał (miniocolin@gmail.com)

Edytuj



Myśków Michał (miniocolin@wp.pl)

Edytuj



## Dodawanie gościa

Email gościa

---

Imię gościa

---

Nazwisko gościa

---

☐ Obecność potwierdzona

Dodaj gościa

---

## Lista

 Dodaj

☐ Przykładowe zadanie 2



☒ Przykładowe zadanie 1



☐ Przykładowe zadanie 3



## **b)Opis**

Dla naszego interfejsu użyliśmy neutralnych, przyjemnych dla oka kolorów. Wybrane zostały one też tak aby jak najbardziej kojarzyły się z weselną atmosferą.

Czcionka przypomina oficjalnie wyglądającą kursywę.

Staraliśmy się aby był on jak najbardziej intuicyjny nawet dla osób, które nie mają codziennej styczności z komputerami.

Używamy także RWD, dzięki czemu nasza aplikacja jest w pełni czytelna zarówno na urządzeniach desktopowych jak i mobilnych.

Co jest bardzo ważne, biorąc pod uwagę ilość użytkowników różnego rodzaju smartfonów.

## 12. Informacje uruchomieniowe w środowisku deweloperskim

Uruchomienie aplikacji w środowisku deweloperskim

### 1. Pobranie projektu z repozytorium

```
git clone https://github.com/MichalMyskow/PPSI2_Wedding_Planner.git  
cd PPSI2_Wedding_Planner
```

### 2. Skopiowanie konfiguracji pliku .env do .env.local

```
cp .env .env.local
```

### 3. Uruchomienie kontenerów dockerowych

```
docker-compose up -d --build
```

### 4. Pobranie zależności Composer

```
docker-compose exec php composer install
```

### 5. Pobranie zależności npm

```
docker-compose exec php npm install
```

### 6. Budowanie Assetów

```
docker-compose exec php npm run dev
```

### **Migracje:**

#### Utworzenie migracji

```
php bin/console doctrine:migrations:diff  
php bin/console doctrine:migrations:migrate
```

### **Bez migracji:**

#### Utworzenie schematu bazy danych

```
php bin/console doctrine:schema:update --force
```

**Ładowanie zdefiniowanych wcześniej danych do bazy danych:**

*php bin/console doctrine:fixtures:load*

**Udostępnione porty:**

**Aplikacja powinna być dostępna pod:**

<http://localhost:8080>

**phpMyAdmin powinien być dostępny pod:**

<http://localhost:8081>

**Dane logowania do phpMyAdmin:**

**Serwer:** *database*

**Użytkownik:** *user*

**Hasło:** *123qwe*

**Nazwa bazy danych:** *weddingplannerdb*

**Przydatne komendy:**

**Docker - uruchomienie kontenerów:**

*docker-compose up -d*

**Docker - zatrzymanie kontenerów:**

*docker-compose stop*

**Docker - zatrzymanie i usunięcie kontenerów:**

*docker-compose down*



## **13. Opis wdrożenia aplikacji i sposobu jej hostowania**

### **a) Opis wdrożenia:**

Aplikacja wdrożona została przy pomocy aplikacji Deployer. Napisana jest ona w języku PHP, obsługuje najpopularniejsze frameworki (w tym Symfony) oraz jest natychmiast gotowa do użycia po instalacji w formie paczki. Aplikacja ta wymaga podania repozytorium git, brancha, z którego będziemy korzystać oraz danych dotyczących serwera (adres, dane użytkownika, port, ścieżka wdrożenia). Pozwala również na określenie konkretnych zadań wykonywanych przy Deployu nowej wersji aplikacji np. automatyczne pobieranie zależności. Aplikacja weryfikuje każdą nową wersję aplikacji i w razie wykrycia problemów pozwala na cofnięcie wersji do ostatniej działającej. Pozwala to na wdrażanie bez przestojów, ponieważ Deployer nie pozwala na wdrożenie niedziałającej wersji co pozwala na nieprzerwaną pracę aplikacji.

### **b) Opis sposobu hostowania:**

Aplikacja hostowa jest na zdalnym serwerze z zainstalowanym systemem operacyjnym Debian GNU/Linux 11. Domena wykupiona została w serwisie nazwa.pl.

Aplikacja dostępna jest pod adresem: <https://weddingplannerproject.pl/>

## 14. Podsumowanie oraz Wnioski

Projekt ten nie był aż tak prostym przedsięwzięciem jak zakładaliśmy przy wybraniu tematu. Jednak ucząc się po drodze udało się nam go ukończyć. Nigdy wcześniej nie zajmowaliśmy się niczym związanym z tematyką weselną. Musieliśmy więc przeprowadzić trochę researchu, aby poprawnie utworzyć naszą aplikację.

Poza techniczną stronę projektu musieliśmy się też zmierzyć po raz pierwszy w ramach uczelni z podziałem zadań na sprinty oraz estymatą. Niektórzy członkowie naszego zespołu mieli już z tym styczność w życiu zawodowym, natomiast niektórzy musieli zainteresować się tym tematem po raz pierwszy.

Nie mieliśmy też osoby, która personalnie byłaby zainteresowana frontendem. Musiała więc to być praca wspólna, aby osiągnąć nasz cel końcowy.

Możemy więc wynieść następujące wnioski:

- Przygotowanie aplikacji wymaga wiedzy poza czystą wiedzą techniczną, jak w naszym wypadku wiedzę o weselach
- Warto posiadać członków zespołu, którzy posiadają głęboką wiedzę w danym zakresie, np. Dev-ops
- Wykupienie domeny jest bardzo proste
- Organizacja grupy może być wymagająca, gdy niektórzy członkowie są osobami pracującymi
- Flow pracy może łatwo być przerwany w wypadku chorób i różnych prywatnych spraw w zespole
- Nie należy być zbyt optymistycznym w przygotowywaniu estymaty