

**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI, INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ**

Aplikacja do listy zakupów

Autorzy: Marcin Bereźnicki, Jakub Burczyk, Bartek Gawęda, Konrad Golemo, Filip Żmijewski

Kierunek i rok studiów: Automatyka i Robotyka, V rok studia magisterskie

Przedmiot: Zarządzanie projektami

Spis treści

[1. Opis aplikacji 3](#_Toc119959208)

[2. Założenia projektu 4](#_Toc119959209)

[3. Przebieg pracy 5](#_Toc119959210)

[3.1. Sprint I 5](#_Toc119959211)

[3.2. Sprint II 8](#_Toc119959217)

[3.3. Sprint III 9](#_Toc119959223)

[3.4. Sprint IV 10](#_Toc119959229)

[4. Aplikacja finalna 11](#_Toc119959235)

[5. Podsumowanie 12](#_Toc119959236)

# Opis aplikacji

Aplikacja powinna zapewnić następujące funkcjonalności:

* Tworzenie listy zakupów,
* Harmonogramowanie list,
* Archiwizowanie list,
* Dodawanie i usuwanie produktów,
* Możliwość ustalenia ilości oraz ręcznego dodania ceny,
* Edycja pozycji na liście zakupów,
* Zaznaczanie zakupionych produktów,
* Sortowanie po wybranym kryterium,
* Możliwość oznaczania kolorami wybranych produktów oraz list,
* Możliwość dodania wymaganego terminu zakupu,
* Możliwość sprawdzenia czy produkt jest już na liście,
* Tworzenie kopii listy zakupów,
* Nazywanie list,
* Przechowywanie kart lojalnościowych,
* Udostępnianie listy zakupów.

# Założenia projektu

Projekt jest realizowany zgodnie z metodologią Scrum. Na każdy sprint przewidziana jest zmiana Scrum Mastera, w celu możliwości sprawdzenia się w tej roli.

Realizacja aplikacji jest podzielona na dwie części: backend oraz frontend. Do realizacji części backendowej zostanie użyty język programowania Python, natomiast do frontendu biblioteka języka JavaScript, React.js.

Celem lepszej organizacji pracy utworzono tablicę Kanban w serwisie Trello oraz repozytorium w serwisie GitHub.

Link do tablicy: https://trello.com/b/D0T6aDvO/shopping-list-app

Link do repozytorium: https://github.com/Mar-Ber/Shopping-List-App/tree/restfulsetup

# Przebieg pracy

## Sprint I

### Podział zadań

* Marcin Bereźnicki - Scrum Master, Product Owner
* Jakub Burczyk - Backend
* Bartek Gawęda - Backend
* Konrad Golemo - Backend
* Filip Żmijewski - Frontend

### Cel sprintu

Celem I sprintu jest stworzenie podstawowego backend-u, umożliwiającego przyszłą integrację z frontend-em oraz podstaw frontend-u.

W pierwszej kolejności przeprowadzona zostanie dyskusja na temat wyboru narzędzia API oraz bibliotek. Następnie zostanie stworzony prototyp i przetestowane zostaną jego możliwości. W tym celu napisana zostanie główna klasa API. W kolejnym kroku konieczne będzie stworzenie możliwości zapisu i odczytu list zakupów do/z pliku JSON, zawierającego wszystkie listy oraz przedmioty w nich zapisane. Do tego konieczne będzie także zdefiniowanie struktury pliku JSON. Dzięki temu powstanie możliwość przechowywania potrzebnych elementów w bazie danych.

W ramach frontend-u zostanie stworzona pierwsza wersja wyglądowa interfejsu użytkownika wraz z dodaniem podstawowych funkcjonalności. Dodana zostanie opcja tworzenia kolejnych list zakupów z możliwością przechowywania ich na stałe w jednym widoku oraz funkcjonalność pozwalająca usuwać i edytować te listy.

### Początek sprintu

|  |  |  |
| --- | --- | --- |
| **Task** | **Osoba/-y odpowiedzialna** | **Status** |
| API tools | Marcin Bereźnicki, Jakub Burczyk, Konrad Golemo, Bartek Gawęda | Odbyła się dyskusja na temat możliwych rozwiązań. Na podstawie analizy i doświadczenia dokonano wyboru frameworku Flask. |
| Flask API prototype | Jakub Burczyk | Jakub stworzy prototyp API w celu jego przetestowania |
| API main Class | Konrad Golemo, Bartek Gawęda | Oczekiwanie na przetestowanie prototypu API |
| Define item JSON and Marshmallow structure | Marcin Bereźnicki,  Konrad Golemo, Bartek Gawęda | Konieczne będzie przeprowadzenie krótkiej dyskusji odnośnie struktury pliku JSON w celu utworzenia jej w odpowiednio dostosowanej formie. |
| Save and load database to JSON files | Jakub Burczyk | Po przetestowaniu API i rozpoczęciu pracy nad główną klasą API powstaną metody do zapisu i odczytu danych. |
| Item database | Marcin Bereźnicki,  Jakub Burczyk, Filip Żmijewski | Konieczne będzie przeprowadzenie dyskusji odnośnie utworzenia bazy danych produktów. |
| Creating interface for shopping lists | Filip Żmijewski | Filip zajmie się utworzeniem pierwszej wersji interfejsu. |
| Adding basic functionality | Filip Żmijewski | Konieczne będzie dodanie możliwości tworzenia kolejnych list zakupów (z możliwością przechowywania ich na stałe w jednym z widoków) oraz usuwanie bądź edytowanie tych list. |
| Making front prettier | Filip Żmijewski | Po utworzeniu bardzo prostego interfejsu, konieczne będzie jego ulepszenie wizualne. |

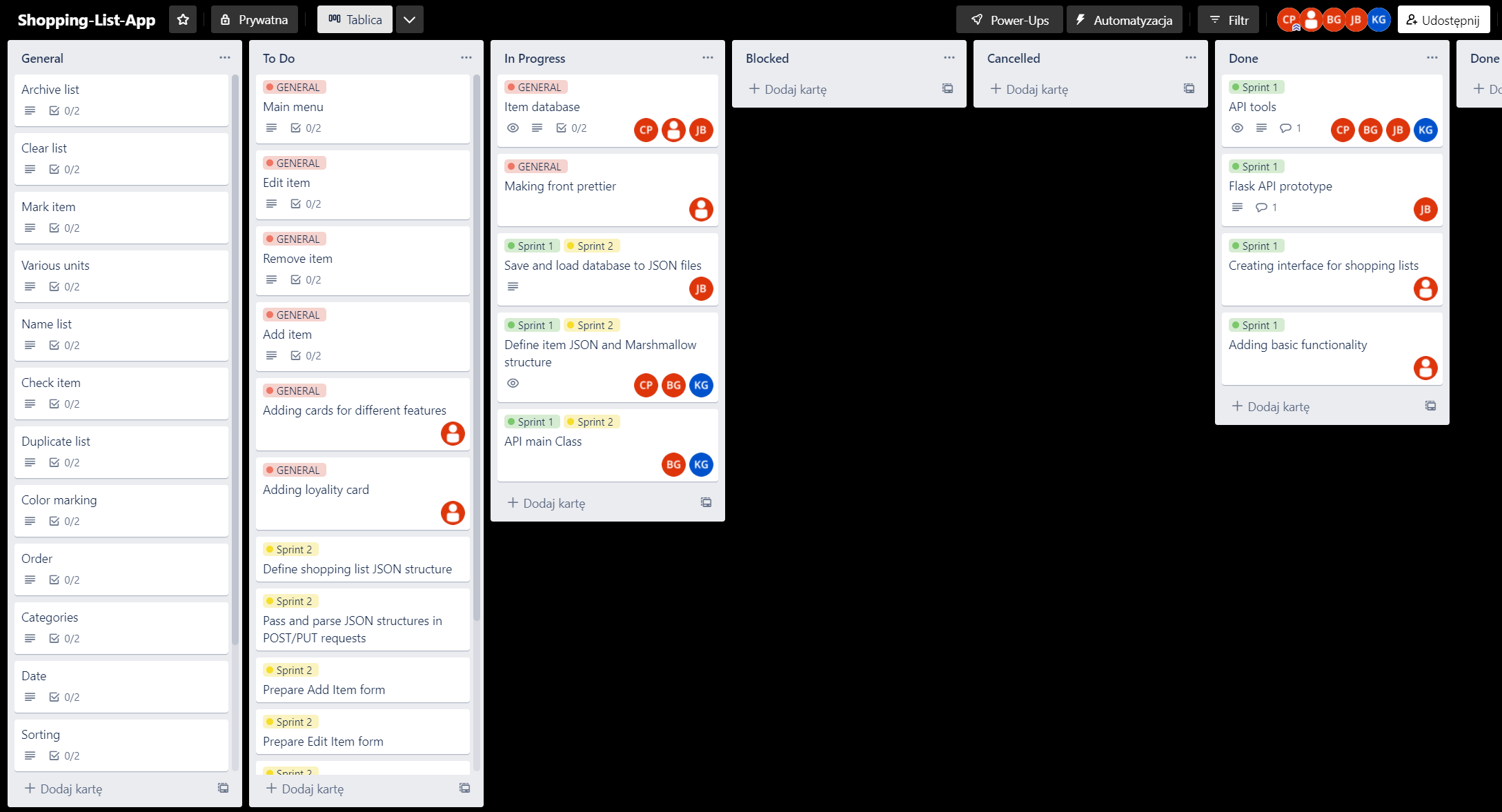
### Przebieg sprintu

|  |  |  |
| --- | --- | --- |
| **Task** | **Osoba/-y odpowiedzialna** | **Status** |
| API tools | Marcin Bereźnicki, Jakub Burczyk, Konrad Golemo, Bartek Gawęda | Done |
| Flask API prototype | Jakub Burczyk | Jakub przetestował API, wybór pozostaje bez zmian. |
| API main Class | Konrad Golemo, Bartek Gawęda | Rozpocznie się etap tworzenia głównej klasy API. |
| Define item JSON and Marshmallow structure | Marcin Bereźnicki,  Konrad Golemo, Bartek Gawęda | Odbyła się dyskusja odnośnie struktury pliku JSON. W razie konieczności zostanie dostosowana w późniejszym czasie. |
| Save and load database to JSON files | Jakub Burczyk | Po przetestowaniu API i rozpoczęciu pracy nad główną klasą API powstaną metody do zapisu i odczytu danych. |
| Item database | Marcin Bereźnicki,  Jakub Burczyk, Filip Żmijewski | Odbyła się dyskusja odnośnie bazy danych. Najwygodniejszym rozwiązaniem na ten moment jest przechowywanie wszystkich danych w formie pliku. |
| Creating interface for shopping lists | Filip Żmijewski | Udało się utworzyć pierwszą wersję interfejsu. |
| Adding basic functionality | Filip Żmijewski | Dodawanie możliwości tworzenia kolejnych list zakupów oraz ich usuwanie bądź edytowanie w trakcie realizacji. |
| Making front prettier | Filip Żmijewski | Filip zacznie dostosowywać wygląd aplikacji. |

### Koniec sprintu

|  |  |  |
| --- | --- | --- |
| **Task** | **Osoba/-y odpowiedzialna** | **Status** |
| API tools | Marcin Bereźnicki, Jakub Burczyk, Konrad Golemo, Bartek Gawęda | Done |
| Flask API prototype | Jakub Burczyk | Done |
| API main Class | Konrad Golemo, Bartek Gawęda | Główna klasa w trakcie powstawania. Ze względu na opóźnienia wynikające z natłoku zadań nie udało się w pełni stworzyć prototypu głównej klasy API. |
| Define item JSON and Marshmallow structure | Marcin Bereźnicki,  Konrad Golemo, Bartek Gawęda | Done |
| Save and load database to JSON files | Jakub Burczyk | Metody do zapisu i odczytu danych z pliku w trakcie powstawania. |
| Item database | Marcin Bereźnicki,  Jakub Burczyk, Filip Żmijewski | Oczekiwanie na utworzenie metod do zapisu i odczytu danych z pliku. |
| Creating interface for shopping lists | Filip Żmijewski | Done |
| Adding basic functionality | Filip Żmijewski | Done |
| Making front prettier | Filip Żmijewski | Dostosowywanie wyglądu w trakcie realizacji |

### Widok tablicy Kanban na zakończenie sprintu



## Sprint II

### Podział zadań

* Marcin Bereźnicki - Backend
* Jakub Burczyk - Backend
* Bartek Gawęda - Scrum Master, Product Owner
* Konrad Golemo - Backend
* Filip Żmijewski - Frontend

### Cel sprintu

Celem sprintu jest dokończenie dwóch zadań z poprzedniego, oraz rozbudowa i dodanie kolejnych funkcjonalności. Rozdzielając zadania i definiując cele na ten sprint podjęta została decyzja o rezygnacji z zadania na obecny sprint, jakim było stworzenie aplikacji do testowania API, ze względu na to iż zaproponowane i zaaprobowane zostało wykorzystanie aplikacji Postman

### Początek sprintu

|  |  |  |
| --- | --- | --- |
| **Task** | **Osoba/-y odpowiedzialna** | **Status** |
| API main Class | Konrad Golemo, Bartek Gawęda | Główna klasa w trakcie powstawania. Ze względu na opóźnienia wynikające z natłoku zadań nie udało się w pełni stworzyć prototypu głównej klasy API. |
| Save and load database to JSON files | Jakub Burczyk | Metody do zapisu i odczytu danych z pliku w trakcie powstawania. |
| Item database | Marcin Bereźnicki,  Jakub Burczyk, Filip Żmijewski | Oczekiwanie na utworzenie metod do zapisu i odczytu danych z pliku. |
| Making front prettier | Filip Żmijewski | Dostosowywanie wyglądu w trakcie realizacji |
| Define shopping list JSON structure | Marcin Bereźnicki, | Ustalono zarys rozwiązania |
| Communication with API | Filip Żmijewski | Przedyskutowano wątpliwości co do sposobu rozwiązania |
| Pass and parse JSON structures in POST/PUT requests | Konrad Golemo,  Jakub Burczyk | Ustalono zarys rozwiązania |
| Prepare Add Item form | Konrad Golemo | Omówiono w jaki sposób najlepiej zaimplementować tą funkcjonalność |
| Prepare Edit Item form | Bartek Gawęda | Omówiono w jaki sposób najlepiej zaimplementować tą funkcjonalność |
| Prepare Add List form | Marcin Bereźnicki, | Omówiono w jaki sposób najlepiej zaimplementować tą funkcjonalność |

### Przebieg sprintu

|  |  |  |
| --- | --- | --- |
| **Task** | **Osoba/-y odpowiedzialna** | **Status** |
| API main Class | Konrad Golemo, Bartek Gawęda | Prace przebiegają bez problemu |
| Save and load database to JSON files | Jakub Burczyk | W trakcie sprintu zdecydowano, że funkcjonalność ta będzie mocno powiązana z funkcjami do dodawania i edycji list zakupów, przez co zadanie oczekuje na ukończenie wspomnianych elementów. |
| Item database | Marcin Bereźnicki,  Jakub Burczyk, Filip Żmijewski | Oczekiwanie na utworzenie metod do zapisu i odczytu danych z pliku. |
| Making front prettier | Filip Żmijewski | Dostosowywanie wyglądu w trakcie realizacji |
| Define shopping list JSON structure | Marcin Bereźnicki, | Przedyskutowano wątpliwości które pojawiły się w trakcie prac, ale nie wpłynęło to na realizację zadania |
| Communication with API | Filip Żmijewski | Prace przebiegają bez problemu |
| Pass and parse JSON structures in POST/PUT requests | Konrad Golemo,  Jakub Burczyk | Prace przebiegają bez problemu |
| Prepare Add Item form | Konrad Golemo | Prace przebiegają bez problemu |
| Prepare Edit Item form | Bartek Gawęda | Prace przebiegają bez problemu |
| Prepare Add List form | Marcin Bereźnicki, | Prace przebiegają bez problemu |

### Koniec sprintu

|  |  |  |
| --- | --- | --- |
| **Task** | **Osoba/-y odpowiedzialna** | **Status** |
| API main Class | Konrad Golemo, Bartek Gawęda | Done |
| Save and load database to JSON files | Jakub Burczyk | Zadanie oczekuje na implementację innych funkcji celem integracji całości. |
| Item database | Marcin Bereźnicki,  Jakub Burczyk, Filip Żmijewski | Oczekiwanie na utworzenie metod do zapisu i odczytu danych z pliku. |
| Making front prettier | Filip Żmijewski | Dostosowywanie wyglądu w trakcie realizacji |
| Define shopping list JSON structure | Marcin Bereźnicki, | Done |
| Communication with API | Filip Żmijewski | Done |
| Pass and parse JSON structures in POST/PUT requests | Konrad Golemo,  Jakub Burczyk | Done |
| Prepare Add Item form | Konrad Golemo | Done |
| Prepare Edit Item form | Bartek Gawęda | Nie udało się rozpocząć zadania |
| Prepare Add List form | Marcin Bereźnicki, | Zadania nie udało się ukończyć w terminie |

### Widok tablicy Kanban na zakończenie sprintu

Obraz zawierający tekst, zrzut ekranu, wewnątrz

Opis wygenerowany automatycznie

## Sprint III

### Podział zadań

* Marcin Bereźnicki -
* Jakub Burczyk -
* Bartek Gawęda -
* Konrad Golemo -
* Filip Żmijewski -

### Cel sprintu

### Początek sprintu

### Przebieg sprintu

### Koniec sprintu

### Widok tablicy Kanban na zakończenie sprintu

## Sprint IV

### Podział zadań

* Marcin Bereźnicki -
* Jakub Burczyk -
* Bartek Gawęda -
* Konrad Golemo -
* Filip Żmijewski -

### Cel sprintu

### Początek sprintu

### Przebieg sprintu

### Koniec sprintu

### Widok tablicy Kanban na zakończenie sprintu

# Aplikacja finalna

# Podsumowanie