

Numer zlecenia i nazwa i akronim projektu: 2023/001 Mapa Sportu - MS	Zleceniodawca: PJATK	Zleceniobiorca: PJATK
Zespół projektowy: Kamil Powierza Adrian Stoltmann Antoni Kania Dawid Feister Kacper Tokarzewski Michał Heppner	Kierownik projektu: Kamil Powierza	Opiekun projektu: Cysewski Grzegorz, Łabuda Marta
Nazwa dokumentu: Wstępny Plan Projektu	Odpowiedzialny za dokument: Kamil Powierza	

Historia dokumentu

Wersja	Opis modyfikacji	Rozdział / strona	Autor modyfikacji	Data
1.0	Wstępny plan projektu	całość	Dawid Feister	16.11.2023
1.1	Wersja finalna	całość	Dawid Feister	24.01.2024

Opis etapu: Celem etapu jest określenie założeń projektu (cele, zakres, ograniczenia, priorytety), przedstawienie wizji docelowego rozwiązania – kształtu systemu i strategii procesu wytwórczego oraz opisanie poszczególnych etapów, przedstawienie harmonogramu prac projektowych i konstrukcyjnych.

Oczekiwane produkty: Wstępny Plan Projektu w postaci dokumentu o nazwie GRX(XYZ)-WPP-WER1.doc o strukturze według poniższego szablonu. W czasie jego tworzenia należy wykorzystać dostępne w laboratoriach lub freeware narzędzia CASE.

1 Założenia projektu

{Klient i problemy biznesowe; Krótki opis projektu, jego uwarunkowania i priorytety}

Klientem projektu jest PJATK. Aplikacja została stworzona jako rozwiązanie biznesowe mające na celu zwiększenie dostępności do aktualnej bazy obiektów sportowych. Jej głównym celem jest zintegrowanie informacji o tych obiektach oraz zapewnienie użytkownikom nawigacji, co ma zachęcić ich do regularnej aktywności fizycznej. Aplikacja umożliwia łatwe znalezienie obiektów sportowych, wyznaczenie tras do nich, zapewnia informacje o obiektach oraz uwzględnia różnorodne uwarunkowania i priorytety związane z aktywnością fizyczną.

1.1 Cele i zakres projektu

{Zakres projektu – odróżnić od zakresu systemu}

- Zwiększenie zainteresowania aktywnością fizyczną.
- Ułatwienie lokalizacji obiektów sportowych i tworzenie wydarzeń.
- Zachęcenie do prowadzenia zdrowszego trybu życia.

1.2 Spodziewane produkty

{Produkty projektu}

Aplikacja Mapa Sportu będzie aplikacją webową dostępną na urządzeniach typu komputer stacjonarny oraz smartfon. Mapa Sportu będzie wymagała połączenia z serwisami udostępniającymi mapy w celu umieszczania znaczników w miejscach obiektów sportowych oraz nawigowanie do nich.

1.3 Udziałowcy

{Adresaci, użytkownicy, inni udziałowcy}

Udziałowcami w projekcie "Mapa Sportu" są różne podmioty, zarówno ożywione, jak i nieożywione, które mają wpływ na projekt lub są bezpośrednio związane z jego funkcjonowaniem. Szczegółowe informacje o udziałowcach zawarte są w dokumencie „Specyfikacja Wymagań Systemowych” jako dokument załączony do projektu.

1.4. Uwarunkowania, ograniczenia, założenia strategii

{Czasowe, środowiska, standardy; Inne (dodatkowe zobowiązania)}

Ochrona danych, bezpieczeństwo, niezawodność, wydajność, łatwość obsługi, skalowalność, wsparcie dla różnych przeglądarek, koszt utrzymania aplikacji, bieżąca aktualizacja danych to kluczowe wymagania i standardy.

1.5. Priorytety

{Wśród celów projektu, priorytet projektu}

1. Wysoki priorytet (must-have):

- a) Wyświetlanie mapy z lokalizacją miejsc sportowych: To kluczowa funkcja, ponieważ umożliwia użytkownikom znalezienie i lokalizowanie obiektów sportowych w wygodny sposób.
- b) Zarządzanie bazą danych miejsc sportowych: Konieczne dla administratorów, aby utrzymać dokładne informacje o obiektach sportowych.

2. Średni priorytet (should-have):

- a) Tworzenie wydarzeń: Umożliwia użytkownikom organizowanie i dołączanie do wydarzeń, co może zwiększyć zaangażowanie społeczności.
- b) Logowanie i rejestracja użytkowników: Dostęp do pełnych funkcji aplikacji dla zalogowanych użytkowników.

3. Niski priorytet (nice-to-have):

- a) Dołączanie do wydarzeń: Może zwiększyć interakcję między użytkownikami, ale nie jest niezbędne do podstawowego funkcjonowania aplikacji.
- b) Odzyskiwanie konta: Przydatne, ale w przypadku, gdy użytkownicy zapomną swoich danych logowania.

1.6. Projekty powiązane i partnerzy zewnętrzni

{Jeśli realizowany projekt jest fragmentem jakiegoś większego projektu (jego zamkniętą funkcjonalnością, modulem) lub kiedy do jego prawidłowego / pełnego działania współpraca z innymi projektami jest konieczna / wymagana / zalecana}

Nie dotyczy, „Mapa Sportu” jest odrębnym i w pełni niezależnym projektem.

1.7. Budżet

{Usunąć punkt jeśli nie dotyczy}

50 tys PLN

2 Analiza ryzyka

{Wskazanie głównych zagrożeń poprzez wskazanie:

- kontekstu zagrożenia tj. miejsca, w którym zagrożenie może wystąpić np. zespół (w tym umiejętności, doświadczenie, obciążana innymi zadaniami itp.), proces wytwórczy (w tym budżet, harmonogram, przydzielanie zasobów itp.),
- czynników ryzyka,
- prawdopodobieństwa wystąpienia,
- możliwych skutków zdarzenia oraz ich wpływu na projekt (dotkliwości skutków zdarzenia).

Należy uwzględnić również zdarzenia, które mają pozytywny wpływ na projekt – szanse, które planujemy wykorzystać. (opisać podobnie jak zagrożenie).

Zalecane przedstawienie w formie tabel(i) lub punktów}

1. Zagrożenia:

a) Brak odpowiednich umiejętności w zespole:

- Kontekst zagrożenia: Zespół może nie posiadać wystarczających umiejętności w zakresie konkretnych technologii potrzebnych do stworzenia aplikacji.

- Czynniki ryzyka: Brak doświadczenia, nieodpowiednie przygotowanie, brak wiedzy w niektórych obszarach.
- Prawdopodobieństwo wystąpienia: Średnie.
- Skutki: Opóźnienia, niższa jakość, potrzeba szkoleń lub zatrudnienia dodatkowych osób.

b) Problemy z budżetem:

- Kontekst zagrożenia: Przewidziane koszty mogą okazać się niedostateczne.
- Czynniki ryzyka: Zbyt optymistyczne oszacowania kosztów, nieprzewidziane wydatki, zmienne wymagania.
- Prawdopodobieństwo wystąpienia: Średnie.
- Skutki: Ograniczenie zakresu projektu, opóźnienia, konieczność szukania dodatkowych źródeł finansowania.

c) Problemy z harmonogramem:

- Kontekst zagrożenia: Opóźnienia w realizacji zadań mogą prowadzić do przekroczenia ustalonego harmonogramu.
- Czynniki ryzyka: Zbyt optymistyczne założenia czasowe, problemy w komunikacji w zespole, zmiany w zakresie projektu.
- Prawdopodobieństwo wystąpienia: Wysokie.
- Skutki: Wydłużenie czasu dostarczenia projektu, niewywiązanie się z terminów, możliwe konsekwencje finansowe.

2. Szanse:

a) Nagłe wzrost zainteresowania aplikacją:

- Kontekst szansy: Nagłe zainteresowanie użytkowników aplikacją może przyczynić się do szybkiego wzrostu popularności.
- Czynniki korzyści: Skuteczna kampania marketingowa, dobre opinie użytkowników, trafne trafienie w potrzeby rynku.
- Prawdopodobieństwo wystąpienia: Średnie.
- Skutki: Szybsze osiągnięcie celu biznesowego, zwiększenie liczby użytkowników, większy zasięg.

b) Nowe możliwości technologiczne:

- Kontekst szansy: Pojawienie się nowych technologii lub rozwiązań może znacząco poprawić funkcjonalności aplikacji.
- Czynniki korzyści: Śledzenie trendów technologicznych, elastyczność w architekturze aplikacji.
- Prawdopodobieństwo wystąpienia: Średnie.
- Skutki: Poprawa efektywności, zwiększenie konkurencyjności, lepsza satysfakcja użytkowników.

3 Wizja rozwiązania

{zakładany kształt systemu i organizacja działania}

Na architekturę systemu składać się będą m.in.:

- aplikacja webowa
- serwer backendowy
- baza danych

Na organizację działania składać się będą m.in.:

- zespół programistów i projektantów
- zarządzanie projektem
- wsparcie techniczne i administracyjne
- wsparcie klienta

3.1. Architektura systemu

{opis architektury, wykorzystywane wzorce (np. MVC), frameworki, itp }

1. Architektura:

a) Aplikacja webowa:

- Frontend: Interfejs użytkownika będzie stworzony w oparciu o HTML, CSS i JavaScript. Użycie nowoczesnych bibliotek/frameworków do budowy interfejsu użytkownika, umożliwia dynamiczne funkcjonalności aplikacji. Wykorzystanie OpenStreetMap i Leaflet, które są potrzebne do tworzenia map.
- Backend: Serwer backendowy będzie zbudowany z użyciem Java, frameworków/bibliotek takich jak: Gradle, Spring Boot, Spring Web, Spring Security, Spring Data, JPA, junit. Narzędzi takich jak: Lombok, Postman, Swagger, postgresSQL, Docker. Wykorzystanie architektury RESTful API dla komunikacji między frontendem a backendem oraz użycie Xamppa.

b) Baza danych:

- Wybór odpowiedniej bazy danych - PostgreSQL

2. Wykorzystywane wzorce i frameworki:

a) Wzorec MVC (Model-View-Controller):

- Podział aplikacji na warstwy modelu danych, warstwy widoku dla interfejsu użytkownika i warstwy kontrolera dla logiki biznesowej.

b) Frameworki i biblioteki:

- Użycie narzędzi zgodnych z założeniami architektury.

3. Wzorce graficzne aplikacji:

a) Wygląd graficzny:

- W projektowaniu interfejsu użytkownika i mockupów zostanie użyte narzędzie Figma, a dla potrzeb graficznych takich jak np.. tworzenie ikon wykorzystane zostanie Illustrator.

3.2. Rozwiązania techniczne

{kształt systemu, koncepcja, standardy, technologia, dekompozycja systemu na podsystemy, opis interakcji pomiędzy podsystemami }

1. Kształt systemu:

a) Warstwa frontendu:

- Interfejs użytkownika oparty na nowoczesnych frameworkach zgodnych w założeniami architektury
- Wykorzystanie komponentów do budowy interaktywnych widoków mapy, filtrów, wyników wyszukiwania itp.

b) Warstwa backendu:

- Zastosowanie serwera backendowego opartego dla obsługi żądań i odpowiedzi.
- Utworzenie RESTful API dla komunikacji między frontendem a backendem, zapewniającego dostęp do danych.

c) Baza danych:

- Użycie bazy danych, PostgreSQL do przechowywania informacji o obiektach sportowych, użytkownikach, recenzjach, itp.

2. Koncepcja i standardy:

a) Bezpieczeństwo:

- Zastosowanie standardów bezpieczeństwa, takich jak uwierzytelnianie użytkowników za pomocą logowania

b) Responsywność:

- Projektowanie aplikacji z myślą o responsywności, aby zapewnić optymalne działanie na różnych urządzeniach - komputerach, tabletach i smartfonach.

3. Technologie i dekompozycja systemu:

a) Podsystemy:

- Moduł mapy: Zawierający interaktywną mapę z miejscami sportowymi, ich lokalizacją i szczegółowymi informacjami.
- Moduł użytkownika: Zarządzanie kontem użytkownika, tworzenie profili, dodawanie recenzji i ocen obiektów sportowych.
- Moduł wyszukiwania: Zapewniający filtrowanie obiektów sportowych według różnych kryteriów (lokalizacja, typ sportu, ocena itp.).
- Moduł administracyjny: Do zarządzania danymi, obsługi zgłoszeń, kontroli jakości treści itp.

b) Interakcje pomiędzy podsystemami:

- Moduł mapy komunikuje się z modułem wyszukiwania i użytkownika w celu wyświetlania informacji, recenzji i ocen.
- Moduł użytkownika może przekazywać dane do modułu administracyjnego (np. zgłoszenia użytkowników).

4. Opis interakcji pomiędzy podsystemami:

a) Przykładowe interakcje:

- Użytkownik loguje się i wyszukuje obiekty sportowe w okolicy.
- Moduł wyszukiwania przekazuje zapytanie do bazy danych i zwraca wyniki pasujące do kryteriów użytkownika.
- Moduł mapy otrzymuje dane i wyświetla obiekty sportowe na interaktywnej mapie.
- Użytkownik może wybrać dany obiekt sportowy, obejrzeć jego szczegóły, oceny i recenzje.

- Dodatkowo, moduł administracyjny może otrzymywać zgłoszenia użytkowników dotyczące nieodpowiednich danych, zgłoszenia obiektów i podejmować działania zaradcze.

3.3. *Technologia i zamierzone środowisko*

{docelowe, wytwórcze, języki }

Technologie wykorzystywane w projekcie:

- HTML: vHTML5;
- CSS: vCSS3;
- JavaScript: vECMAScript 2022;
- OpenStreetMap: ostatnia stabilna wersja;
- Leaflet: ostatnia stabilna wersja;
- Java: v19(+ późniejsze aktualizacje);
- Windows 11 (+aktualizacje)
- Google Chrome: v91.0.4472(+ późniejsze aktualizacje);
- Opera: 12.0(+ późniejsze aktualizacje);
- Figma: v2023(+ późniejsze aktualizacje);
- Illustrator: v28.1;
- XAMPP: v8.2.12;
- Spring Boot: v3.2.0;
- Spring Web: v6.0(+ późniejsze aktualizacje);
- Spring Security: v6.0(+ późniejsze aktualizacje);
- Spring Data: v2023.1.2(+ późniejsze aktualizacje);
- JPA: v2.2(+ późniejsze aktualizacje);
- Lombok: v1.20(+ późniejsze aktualizacje);
- JUnit: v5.7.0;
- Postman: v9.16.0;
- Swagger: v3.0(+ późniejsze aktualizacje);
- PostgreSQL: v15.

Środowisko i narzędzia wytwórcze:

- IntelliJ IDEA 2023.3.2 do pracy nad kodem Java i testowania jednostkowego i integracyjnego;
- WebStorm v2023.3 oraz Visual Studio v2022 do frontendu;
- serwer lokalny do postawienia bazy danych i hostowania aplikacji;
- narzędzie Gradle v8.5 do zarządzania zależnościami, budowania i uruchamiania aplikacji;
- Git wraz z platformą GitHub do zarządzania kodem źródłowym i wersjonowania;
- Docker v25.0.1 jako platforma do konteneryzacji.

4 *Proces wytwarzania*

{dobór strategii wytwarzania (patrz poniższe podpunkty)}

W procesie wytwarzania będziemy stosować metodykę zwinną taką jak Scrum. Podzielimy projekt na krótkie sprinty i regularnie będziemy dostarczać kolejne etapy i funkcjonalności z nimi związane.

4.1. *Strategia prowadzenia prac*

{wskazanie wg jakiej strategii będzie prowadzony projekt; pokazanie 1-2 alternatywnych rozwiązań; uwzględnienie ryzyka jakie niesie ze sobą każde z rozwiązań; punkt powinien być zakończony uzasadnionym wyborem strategii prowadzenia prac}

W kontekście projektu "Mapa Sportu" z uwzględnieniem jego specyfiki, częstych zmian w wymaganiach oraz konieczności szybkiego reagowania na potrzeby użytkowników, metodyka Scrum wydaje się być najbardziej odpowiednim rozwiązaniem. Jego dynamiczny, iteracyjny charakter pozwala na ciągle dostarczanie wartości i łatwiejsze radzenie sobie z ewolucją projektu. Praca w metodyce Scrum daje możliwość szybkiego reagowania na zmiany i dostosowywania priorytetów, co w przypadku projektu mapy sportowej, gdzie dynamiczność wymagań jest kluczowa, jest istotnym atutem. Scrum zapewnia transparentność postępu prac dzięki regularnym spotkaniom, co umożliwia wczesne wykrycie problemów oraz skuteczne zarządzanie ryzykiem.

4.2. *Proces wytwórczy*

{szczegółowe opisanie wybranej strategii procesu, etapy procesu, cykl życia, cele, zadania, powiązania, produkty, miary}

1. Etap planowania (Sprint Planning):

- a) Cele: Określenie funkcjonalności do zaimplementowania w danym okresie (sprint) oraz sprecyzowanie backlogu produktu.
- b) Zadania: Ustalenie zadań do wykonania, ich priorytetów oraz szacowanie czasu na realizację.
- c) Powiązania: Zespół wytwarza plan, jak osiągnąć cele sprintu oraz zatwierdza funkcjonalności do wdrożenia.
- d) Produkty: Sprecyzowany backlog produktu, plan sprintu.

2. Cykl życia sprintu:

- a) Cele: Rozwój i dostarczenie funkcjonalności zgodnie z ustalonym planem.
- b) Zadania: Codzienne spotkania w celu synchronizacji działań, realizacja zadań przypisanych do sprintu.
- c) Powiązania: Regularne spotkania podsumowujące postęp prac (Sprint Review), retrospekcje po zakończeniu sprintu (Sprint Retrospective).
- d) Produkty: Inkrementalne dostarczanie funkcjonalności po każdym zakończonym sprincie.

3. Cele Sprint Review:

- a) Cele: Przedstawienie wykonanych prac klientowi, uzyskanie opinii i feedbacku, zaplanowanie kolejnych kroków na podstawie otrzymanego inputu.
- b) Zadania: Prezentacja zrealizowanych zadań, dyskusja z klientem, aktualizacja backlogu.
- c) Powiązania: Wykorzystanie informacji z review do ulepszania produktu, weryfikacja postępów.
- d) Produkty: Aktualizacja backlogu, ewentualnie nowe wytyczne i kierunki rozwoju.

4. Cele Sprint Retrospective:

- a) Cele: Analiza pracy zespołu, identyfikacja problemów, omówienie co poszło dobrze, co należy poprawić.

- b) Zadania: Dyskusja w zespole na temat procesu, identyfikacja i dokumentacja punktów do poprawy, wyznaczenie działań korygujących.
- c) Powiązania: Wdrażanie poprawek i ulepszeń w kolejnych sprintach.
- d) Produkty: Plan działania dla kolejnych sprintów w celu eliminacji wykrytych problemów.

4.2.1. {Nazwa etapu/zadania}

Jeśli projekt realizowany w kilku etapach należy dla każdego etapu określić: cele, produkty etapu, główne zadania, miary oceny i kryteria akceptacji; wskazanie potencjalnych zagrożeń realizacji etapu (skutków, czynników ryzyka, prawdopodobieństwa wystąpienia, działań); zasady zapewnienia jakości i zarządzania.

Jeśli projekt realizowany bez podziału na kilka etapów powyższe musi być wyspecyfikowane tylko jeden raz dla całego projektu.

1. Analiza wymagań

- a) Cele: Zrozumienie oczekiwań użytkowników odnośnie do funkcjonalności i interfejsu platformy Mapa sportu.
- b) Produkty etapu: Dokumentacja wymagań użytkownika, produktu.
- c) Główne zadania: Zebranie i analiza danych dotyczących preferencji użytkowników.
- d) Miary oceny i kryteria akceptacji: Uzyskanie kompletnego zestawu wymagań od użytkowników.
- e) Zagrożenia: Niedokładne zrozumienie potrzeb użytkowników oraz brak reprezentatywności zebranych danych.
- f) Zasady zapewnienia jakości i zarządzania: Regularna weryfikacja zebranych danych.

2. Projektowanie interfejsu

- a) Cele: Stworzenie intuicyjnego i atrakcyjnego interfejsu użytkownika (UI) oraz zaprojektowanie funkcjonalności aplikacji.
- b) Produkty etapu: Prototypy interfejsu użytkownika.
- c) Główne zadania: Tworzenie prototypów interfejsu użytkownika oraz definiowanie funkcjonalności aplikacji.
- d) Miary oceny i kryteria akceptacji: Uzyskanie aprobaty od użytkowników dla prototypów interfejsu.
- e) Zagrożenia: Nieodpowiedni design UI/UX oraz brak spójności między funkcjonalnościami.
- f) Zasady zapewnienia jakości i zarządzania: Testy użyteczności interfejsu z udziałem użytkowników oraz regularne przeglądy i iteracje prototypów.

3. Implementacja i testowanie

- a) Cele: Stworzenie działającej wersji platformy Mapa sportu.
- b) Produkty etapu: Gotowa aplikacja internetowa.
- c) Główne zadania: Programowanie i implementacja funkcjonalności oraz testowanie aplikacji.
- d) Miary oceny i kryteria akceptacji: Zgodność z założeniami funkcjonalnymi oraz testy sprawdzające wydajność i stabilność aplikacji.
- e) Zagrożenia: Błędy w implementacji funkcji oraz problemy wydajnościowe.
- f) Zasady zapewnienia jakości i zarządzania: Testy jednostkowe i testy integracyjne oraz debugowanie i naprawa błędów.

5 Infrastruktura

{technologia, narzędzia, środowisko i ich dostępność; dopasowanie infrastruktury do przyjętych założeń i procesu wytwórczego}

1. Technologia:

- a) Serwery: Dostęp do serwerów w chmurze lub lokalnych serwerowniach.
- b) Bazy danych: Wykorzystanie bazy danych PostgreSQL.
- c) Języki programowania i frameworki:
 - Java + Spring do tworzenia backendu.
 - JavaScript, HTML, CSS do frontendu.
 - Wykorzystanie odpowiednich frameworków i narzędzi zgodnie z pkt. 3.3 tego dokumentu.

2. Narzędzia:

- a) IDE:
 - IntelliJ IDEA dla Javy
 - WebStorm oraz Visual Studio dla JavaScriptu, HTML, CSS.
- b) Narzędzia do zarządzania projektem:
 - Platforma do zarządzania projektem, np. Jira, Google Drive
- c) System kontroli wersji:
 - Wykorzystanie Git (np. GitHub, GitLab).

3. Środowisko:

- a) Wytwórcze środowisko pracy:
 - Zapewnienie środowiska deweloperskiego dla programistów do tworzenia i testowania kodu.
- b) Testowanie i wdrożenie:
 - Środowiska testowe do weryfikacji funkcjonalności.
 - Automatyzacja wdrożeń.

4. Dostępność:

- a) Zapewnienie stabilności serwerów:
 - Używanie serwerów o wysokiej dostępności i skalowalności.
- b) Bezpieczeństwo:
 - Odpowiednie zabezpieczenia serwerów i danych.

5.1. Zakładane zasoby

{Czym dysponujemy? Co jest nieodzowne? Co musimy pozyskać?; praca ludzka; środki materialne (trwałe, nietrwałe tj.: zasoby sprzętowe, specyficzny sprzęt, zasoby programowe, narzędzia, infrastruktura; środki niematerialne (wiedza, doświadczenie); usługi; pieniądze; czas}

1. Zasoby ludzkie:

- a) Zespół programistów i projektantów odpowiedzialny za tworzenie aplikacji, interfejsu użytkownika, backendu, interakcji z bazą danych, projektowanie interfejsów, zarządzanie projektem.
- b) Zespół administracyjny zapewniający obsługę, wsparcie techniczne, zarządzanie serwerami, utrzymanie aplikacji.

2. Środki materialne:

- a) Sprzęt komputerowy dla członków zespołu programistów, projektantów i zespołu administracyjnego.
 - b) Dostęp do serwerów lub lokalne serwery dla hostowania aplikacji i bazy danych.
3. Środki programowe:
- a) Oprogramowanie/Licencje na środowiska programistyczne (np. IntelliJ IDEA, WebStorm), narzędzia do zarządzania projektem (np. Jira), system kontroli wersji (Git/GitHub), narzędzia do testowania i debugowania aplikacji.
4. Infrastruktura:
- a) Środowiska deweloperskie dla programistów do tworzenia i testowania kodu.
 - b) Środowiska testowe do weryfikacji funkcjonalności aplikacji przed wdrożeniem.
 - c) Narzędzia i procedury do automatyzacji procesu wdrażania aplikacji.
5. Zasoby niematerialne:
- a) Umiejętności programistyczne, doświadczenie w tworzeniu aplikacji webowych, znajomość technologii używanych w projekcie.
6. Pieniądze:
- a) Środki finansowe na pokrycie kosztów związanych z tworzeniem aplikacji, wynagrodzenia dla zespołu, infrastrukturę IT, licencje oprogramowania.
7. Czas:
- a) Czas pracy zespołu, w którym planowane jest tworzenie i rozwijanie aplikacji, nie później niż do 27.01.2024 roku.
 - b) Wytyczone terminy na poszczególne etapy projektu, sprinty w metodyce Scrum, terminy dostarczania kolejnych funkcjonalności.

5.2. Organizacja zespołu, odpowiedzialność

{ struktura organizacyjną projektu: kierownik, zespół, komitet sterujący, odpowiedzialności; zależność od specyfiki projektu (np. wdrożeniowego) }

Struktura organizacyjna projektu oraz :

- a) Kierownikiem projektu jest Kamil Powierza jako osoba odpowiedzialna za ogólne kierowanie projektem, koordynację działań zespołu, nadzór nad postępami, zarządzanie zasobami i harmonogramem.
- b) Zespół programistów i projektantów to studenci odpowiedzialni za tworzenie aplikacji, interfejsu użytkownika, backendu, interakcji z bazą danych, testowanie aplikacji, oraz regularne raportowanie postępów.

5.3. Infrastruktura techniczna

1. Serwery:
 - wykorzystanie serwerów do hostowania aplikacji webowej i serwera backendowego. - uruchomienie serwerów baz danych PostgreSQL do przechowywania danych dotyczących obiektów sportowych, użytkowników, recenzji itp.

2. Języki programowania i frameworki:

- dla backend, użycie języka programowania Java, w połączeniu z odpowiednim frameworkiem Spring, do tworzenia logiki biznesowej i obsługi żądań aplikacji.
- dla frontend, użycie języków takich jak JavaScript, HTML, CSS oraz popularnych bibliotek/frameworków do tworzenia interfejsu użytkownika.

3. Baza danych:

- baza danych PostgreSQL, która jest skalowalna i zapewniająca niezbędną wydajność do przechowywania danych aplikacji.

4. Narzędzia deweloperskie:

- IDE, wykorzystanie narzędzi programistycznych, takich jak IntelliJ IDEA do pracy nad kodem Java, WebStorm oraz Visual Studio do frontendu, które ułatwiają rozwój aplikacji.
- system kontroli wersji: Git w połączeniu z platformą taką jak GitHub, GitLab do zarządzania kodem źródłowym, wersjonowania i współpracy zespołu.
- narzędzia testowe do testowania jednostkowego, integracyjnego oraz funkcjonalnego aplikacji w celu zapewnienia jakości kodu i aplikacji.

5. Zabezpieczenia:

- zabezpieczenia aplikacji jako uwierzytelnianie użytkowników, szyfrowanie danych.

5.4. Infrastruktura komunikacyjna

W zależności od konieczności wymiany informacji, dla efektywnej komunikacji stosowane będą:

- a) do komunikacji w zespole projektowym stosowana będzie aplikacja Microsoft Teams lub Facebook oraz regularne spotkania,
- b) do zarządzania projektem wykorzystywana będzie Jira lub Excel w celu śledzenia postępu prac, harmonogramu,
- c) do komunikacji z klientem wykorzystywana będzie platforma mailowa lub spotkania na uczelni,
- d) do przechowywania i udostępniania informacji, plików wykorzystywane będzie Google Drive,

5.5. Infrastruktura dokumentacyjna

{zasady raportowania i dokumentacji}

Na zakończenie każdego sprintu, nastąpi aktualizacja dokumentacji do projektu. Dokumentacja techniczna będzie zawierała zarówno opis struktury, komponentów, interfejsów i wzorców użytych w aplikacji oraz w kodzie źródłowym pojawiają się komentarze. Obowiązywać będzie wersjonowanie dokumentów, czyli utrzymywanie aktualnych wersji dokumentów oraz ich archiwizacja. Dodatkowo obowiązkowo dostęp do dokumentacji będzie

posiadał cały zespół projektowy. Dokumentacja podlegać będzie regularnej ocenie jakościowej, w celu zapewnienia dokładności i zgodności z rzeczywistym stanem projektu. Dla projektu zastosowane zostaną odpowiednie zabezpieczenia i backup dla wszystkich dokumentów.

6 Ramowy harmonogram

Szczegółowe informacje dotyczące ramowego harmonogramu oraz elementów z nim związanych są zawarte w wykresie Gantta, który jest załącznikiem do dokumentu.

6.1. Rejestr produktu

a) Zarządzanie zespołem:

- ustalenie harmonogramu spotkań zespołu;
- przypisanie odpowiedzialności za poszczególne zadania;
- utrzymywanie komunikacji w zespole za pomocą narzędzi Microsoft Teams, Facebook

b) Zarządzanie błędami:

- zgłaszanie błędów w bieżącej komunikacji;
- monitorowanie błędów;
- priorytetyzacja napraw w oparciu o ich wpływ na użytkowników;

c) Zgłaszanie wniosków i sugestii:

- zgłaszanie wniosków na forum grupowym lub podczas spotkań;
- regularne przeglądanie i decydowanie o ich implementacji na spotkaniach;

d) Planowanie przyszłych aktualizacji:

- ustalenie kryteriów i priorytetów dla kolejnych funkcji do dodania zgodnie z dokumentacją;

e) Historia projektu:

- regularne aktualizowanie dokumentacji wraz z postępem projektu;

f) Marketing i promocja:

- w przyszłości celem rozpowszechnienia aplikacji, należy opracować strategię marketingową w celu zwiększenia świadomości o aplikacji;

g) Badania użytkowników:

- przeprowadzenie badań ankietowych lub wywiadów z użytkownikami w celu zrozumienia ich potrzeb i oczekiwań

h) Optymalizacja wydajności:

- monitorowanie wydajności aplikacji i identyfikacja obszarów wymagających optymalizacji;
- implementacja rozwiązań mających na celu poprawę szybkości działania i responsywności aplikacji;

6.2. Rejestr sprintu

6.2.1. Rejestr sprintu 1

{Sprint backlog – tabelaryczne przedstawienie zadań realizowanych w ramach sprintu wraz z przypisaniem osoby realizującej, nakładu godzinowego, itp.

Podpunkt powtórzony tyle razy ile jest planowanych sprintów. }

A. Cel sprintu:

- w pierwszym sprincie głównym celem było przygotowanie podstawowej infrastruktury projektu oraz stworzenie interfejsu graficznego strony głównej i obiektu sportowego.

B. Co zostało osiągnięte:

- zakończono projekt graficzny strony głównej, obejmujący logo, stronę główną i stronę obiektu,
- ukończono projekt graficzny strony obiektu sportowego, zawierający mapę z wykorzystaniem OpenMap API,
- zaimplementowano frontend dla strony głównej i strony obiektu sportowego,
- przygotowano bazę danych i zintegrowano ją z interfejsem użytkownika,
- zaimplementowano podstawową funkcjonalność mapy, umożliwiającą wyświetlanie obiektów sportowych na mapie.

C. Co zostanie przesunięte do kolejnych sprintów:

- brakujące funkcje interakcji użytkownika na stronie głównej,
- dopracowanie wyglądu i responsywności interfejsu,
- testowanie integracyjne i jednostkowe.

D. Ocena sprintu:

- sprint 1 został ukończony zgodnie z założeniami, a większość celów została osiągnięta. Istnieją pewne obszary, które wymagają dalszego dopracowania, ale ogólnie rzecz biorąc, prace przebiegły zgodnie z planem.

E. Zrealizowane taski wraz z osobami za nie odpowiedzialnymi i czasem pracy (historia tasków prowadzona na Github):

- a) Projekty graficzne (Kamil Powierza) 20h,
- b) Frontend (Kacper Tokarzewski, Dawid Feister) 20h,
- c) Backend (Antoni Kania, Adrian Stoltmann) 45h,
- d) Baza danych (Michał Heppner) brak danych o czasie pracy.

6.2.2. Rejestr sprintu 2

{Sprint backlog – tabelaryczne przedstawienie zadań realizowanych w ramach sprintu wraz z przypisaniem osoby realizującej, nakładu godzinowego, itp.

Podpunkt powtórzony tyle razy ile jest planowanych sprintów. }

A. Cel sprintu:

- celem drugiego sprintu było skoncentrowanie się na tworzeniu interfejsu użytkownika dla różnych funkcjonalności aplikacji oraz stworzenie panelu administratora. Z uwagi na najdłuższy czasowo sprint, wprowadzono najwięcej funkcjonalności.

B. Co zostało osiągnięte:

- ukończono projekt graficzny dla stron wydarzenia, rejestracji, logowania oraz panelu użytkownika i administratora,
- zaimplementowano frontend dla wszystkich wymienionych stron,
- przygotowano funkcje autoryzacji i rejestracji użytkowników,
- stworzono panel administracyjny umożliwiający dodawanie, usuwanie i edycję obiektów oraz wydarzeń.

C. Co zostanie przesunięte do kolejnych sprintów:

- testy integracyjne i jednostkowe dla nowych funkcjonalności,
- ostateczne dopracowanie interfejsu użytkownika na wszystkich stronach i poprawa błędów,
- implementacja funkcji zapominania hasła i innych szczegółów dotyczących użytkowników.

D. Ocena sprintu:

- sprint 2 przebiegł z sukcesem, a wszystkie główne cele zostały osiągnięte. Interfejs użytkownika został znacznie rozbudowany, a panel administracyjny jest gotowy do dalszych testów.

E. Zrealizowane taski wraz z osobami za nie odpowiedzialnymi i czasem pracy (historia tasków prowadzona na Github):

- a) Projekty graficzne (Kamil Powierza) 10h,
- b) Frontend (Kacper Tokarzewski, Dawid Feister, Kamil Powierza) 120h,
- c) Backend (Antoni Kania, Adrian Stoltmann) 65h,
- d) Baza danych (Michał Heppner) brak danych o czasie pracy.

6.2.3. Rejestr sprintu 3

{Sprint backlog – tabelaryczne przedstawienie zadań realizowanych w ramach sprintu wraz z przypisaniem osoby realizującej, nakładu godzinowego, itp.

Podpunkt powtórzony tyle razy ile jest planowanych sprintów. }

A. Cel sprintu:

- trzeci sprint miał na celu zintegrowanie wszystkich części projektu, poprawę funkcjonalności które tego wymagały oraz przeprowadzenie testów integracyjnych i jednostkowych.

B. Co zostało osiągnięte:

- zintegrowano frontend z backendem, umożliwiając komunikację między nimi,
- przeprowadzono testy jednostkowe i integracyjne dla wszystkich krytycznych funkcji aplikacji,
- wykryte błędy zostały naprawione, a aplikacja została zoptymalizowana pod kątem wydajności.

C. Co zostanie przesunięte do kolejnych sprintów:

- brak.

D. Ocena sprintu:

- sprint 3 zakończył fazę aktywnego rozwoju projektu, a aplikacja jest gotowa do wdrożenia. Testy przebiegły pomyślnie, a wszystkie znaczące błędy zostały naprawione. Projekt osiągnął swój główny cel i jest gotowy do dalszego rozwoju.

E. Zrealizowane taski wraz z osobami za nie odpowiedzialnymi i czasem pracy (historia tasków prowadzona na Github):

- e) Projekty graficzne (Kamil Powierza) 0h,
- f) Frontend (Kacper Tokarzewski, Dawid Feister, Kamil Powierza) 15h,
- g) Backend (Antoni Kania, Adrian Stoltmann) 10h,
- h) Baza danych (Michał Heppner) brak danych o czasie pracy.

7 Załączniki

Wykres_Gantt vfinal.pdf

Diagram Klas vfinal.pdf

erd-byt vfinal.pdf

Podsystemy_i_komponenty vfinal.pdf