



POLSKO-JAPOŃSKA
AKADEMIA TECHNIK
KOMPUTEROWYCH

Karta Projektu

Temat projektu: Gamitude manage your Energy, not your Time
Akronim: GTM
Opiekun: Tadeusz Puźniakowski
Konsultanci: Tadeusz Puźniakowski Marek Bednarczyk
Cele projektu: Dostarczenie użytkownikom narzędzia do zarządzania ich energią, opartego o najnowsze odkrycia dotyczące ludzkiej produktywności. Jednocześnie chcemy, aby użytkownicy zaczęli postrzegać pracę bądź naukę bardziej pozytywnie dzięki powiązaniom pracy z elementami gier RPG.
Rezultaty projektu: Kompletny system do zarządzania projektami opierający się na najnowszych badaniach dotyczących ludzkiej produktywności.
Miary sukcesu: Działający system projektów, Bullet Journal, Gamitude Themes.
Ograniczenia: Czasowe, umiejętnościowe, budżetowe

Wykonawca	Numer albumu	Specjalizacja	Tryb studiów
Robert Deyk	s17707	SI	Stacjonarne
Paweł Benkowski	s16569	SI	Stacjonarne
Stanisław Lutkiewicz	s17535	SI	Stacjonarne

Data ukończenia projektu:	to be determined	Recenzent:	to be determined
----------------------------------	------------------	-------------------	------------------

Spis treści

1	Wprowadzenie	8
1.1	Kontekst pracy	8
1.2	Przedstawienie problemu	9
1.3	Cel i zakres projektu	9
1.4	Podjęcie do projektu	9
1.5	Rezultaty	9
1.6	Organizacja dokumentu	9
1.7	Słownik pojęć	10
2	Omówienie problemu	11
2.1	Przedstawienie problemu	11
2.2	Rich picture	12
2.3	Konkurencyjne rozwiązania	12
2.3.1	Brain Focus	12
2.3.2	Productivity Challenge Timer	13
2.3.3	Habitu	14
2.3.4	Pomodoro	15
2.4	Propozycja rozwiązania	15
2.5	Kontekst systemu	16
2.6	Cele i odbiorcy systemu	17
2.6.1	Cele systemu	17
2.6.2	Uczestnicy(należy ponownie rozważyć)	17
2.6.3	Grupa docelowa	19
3	Analiza	20
3.1	Wymagania(w trakcie ponownego rozważania)	20
3.1.1	Wymagania ogólne i dziedzinowe	20
3.1.2	Wymagania funkcjonalne	21
3.1.3	Wymagania pozafunkcjonalne	27
3.1.4	Wymagania na środowisko docelowe	28
3.2	Wymagania jakościowe i inne	28
3.3	Przypadki użycia oraz diagramy (w trakcie przetwarzania i tworzenia nowych)	28

3.3.1	Przypadki użycia	28
3.3.2	Diagramy	28
4	Planowanie	29
4.1	Metodyka pracy	29
4.2	Narzędzia	29
4.3	Technologie	29
5	Projektowanie i finalna implementacja	30
5.1	Architektura projektu	30
5.1.1	Frontend	30
5.1.2	Backend	30
5.1.3	Baza Danych	30
5.1.4	Chmura	30
5.2	Implementacja rozwiązania	30
6	Historia sprintów	31
6.1	Rozpoznanie dziedzinowe	31
6.1.1	Założenia sprintu	31
6.1.2	Wykonane zadania	31
6.1.3	Napotkane problemy	31
6.2	Wstępna dokumentacja	31
6.2.1	Założenia sprintu	31
6.2.2	Wykonane zadania	32
6.2.3	Napotkane problemy	32
6.3	Use Case + WPP	32
6.3.1	Założenia sprintu	32
6.3.2	Wykonane zadania	32
6.3.3	Napotkane problemy	32
6.4	Dalsza praca przy SWS	32
6.4.1	Założenia sprintu	32
6.4.2	Wykonane zadania	32
6.4.3	Napotkane problemy	33
6.5	Przygotowanie narzędzi do tworzenia systemu	33
6.5.1	Założenia sprintu	33
6.5.2	Wykonane zadania	33
6.5.3	Napotkane problemy	33
6.6	Diagramy funkcjonalności, wymagania bazy danych oraz re- sponsywne rozłożenie strony	33
6.6.1	Założenia sprintu	33
6.6.2	Wykonane zadania	33
6.6.3	Napotkane problemy	33
6.7	Diagramy klas i nawigacja strony	34
6.7.1	Założenia sprintu	34

6.7.2	Wykonane zadania	34
6.7.3	Napotkane problemy	34
6.8	Diagram sekwencji oraz statystyki i energie na stronie	34
6.8.1	Założenia sprintu	34
6.8.2	Wykonane zadania	34
6.8.3	Napotkane problemy	34
6.9	Algorytmy oraz rangi na stronie	35
6.9.1	Założenia sprintu	35
6.9.2	Wykonane zadania	35
6.9.3	Napotkane problemy	35
6.10	Firebase, Heroku, projekty	35
6.10.1	Założenia sprintu	35
6.10.2	Wykonane zadania	35
6.10.3	Napotkane problemy	35
6.11	Firebase autoryzacja, API refactor	36
6.11.1	Założenia sprintu	36
6.11.2	Wykonane zadania	36
6.11.3	Napotkane problemy	36
6.12	Integracja	36
6.12.1	Założenia sprintu	36
6.12.2	Wykonane zadania	36
6.12.3	Napotkane problemy	36
6.13	Prezentacja i demo	37
6.13.1	Założenia sprintu	37
6.13.2	Wykonane zadania	37
6.13.3	Napotkane problemy	37
6.14	Zmiana technologii na backendzie, retrospekcja	37
6.14.1	Założenia sprintu	37
6.14.2	Wykonane zadania	37
6.14.3	Napotkane problemy	37
6.15	Festiwal pomysłów i mockup'y	38
6.15.1	Założenia sprintu	38
6.15.2	Wykonane zadania	38
6.15.3	Napotkane problemy	38
6.16	Festiwal pomysłów ciąg dalszy	38
6.16.1	Założenia sprintu	38
6.16.2	Wykonane zadania	38
6.16.3	Napotkane problemy	38
6.17	Autoryzacja i budowanie bibliografii	38
6.17.1	Założenia sprintu	38
6.17.2	Wykonane zadania	39
6.17.3	Napotkane problemy	39
6.18	Smartwatch'e i MBTI	39
6.18.1	Założenia sprintu	39

6.18.2	Wykonane zadania	39
6.18.3	Napotkane problemy	39
6.19	Odnowienie dokumentacji i bezpieczeństwo	39
6.19.1	Założenia sprintu	39
6.19.2	Wykonane zadania	39
6.19.3	Napotkane problemy	39
6.20	Autoryzacja, dokumentacja	40
6.20.1	Założenia sprintu	40
6.20.2	Wykonane zadania	40
6.20.3	Napotkane problemy	40
6.21	Końcowe prace nad przepisanyimi serwisami	40
6.21.1	Założenia sprintu	40
6.21.2	Wykonane zadania	40
6.21.3	Napotkane problemy	40
6.22	Integracja i LaTeX	41
6.22.1	Założenia sprintu	41
6.22.2	Wykonane zadania	41
6.22.3	Napotkane problemy	41
6.23	Przepisywanie dokumentacji	41
6.23.1	Założenia sprintu	41
6.23.2	Wykonane zadania	41
6.23.3	Napotkane problemy	41
6.24	Podpowiedzi i naprawa błędów	41
6.24.1	Założenia sprintu	41
6.24.2	Wykonane zadania	42
6.24.3	Napotkane problemy	42
6.25	DZW i integracja rang	42
6.25.1	Założenia sprintu	42
6.25.2	Wykonane zadania	42
6.25.3	Napotkane problemy	42
6.26	SWS i nowe minutniki	42
6.26.1	Założenia sprintu	42
6.26.2	Wykonane zadania	42
6.26.3	Napotkane problemy	43
6.27	Strona główna i przerwy	43
6.27.1	Założenia sprintu	43
6.27.2	Wykonane zadania	43
6.27.3	Napotkane problemy	43
6.28	Experymenty z Azure, ulepszanie parallax'a	43
6.28.1	Założenia sprintu	43
6.28.2	Wykonane zadania	43
6.28.3	Napotkane problemy	43
6.29	Zbieranie informacji	44
6.29.1	Założenia sprintu	44

6.29.2	Wykonane zadania	44
6.29.3	Napotkane problemy	44
6.30	Podpowiedzi i końcowe prace nad Gamitude 1.0	44
6.30.1	Założenia sprintu	44
6.30.2	Wykonane zadania	44
6.30.3	Napotkane problemy	44
6.31	Koniec pracy nad Gamitude 1.0, dodanie folderów dla projek- tów, zmiany w bazie danych	44
6.31.1	Założenia sprintu	44
6.31.2	Wykonane zadania	45
6.31.3	Napotkane problemy	45
6.32	Koniec pracy nad Gamitude 1.0, dodanie folderów dla projek- tów, zmiany w bazie danych	45
6.32.1	Założenia sprintu	45
6.32.2	Wykonane zadania	45
6.32.3	Napotkane problemy	45
6.33	Pełne połączenie frontend'u i backend'u, docker i poprawki strony głównej w wersji 2.0	46
6.33.1	Założenia sprintu	46
6.33.2	Wykonane zadania	46
6.33.3	Napotkane problemy	46
6.34	Minutniki w 2.0, uspołnienie wizji	46
6.34.1	Założenia sprintu	46
6.34.2	Wykonane zadania	46
6.34.3	Napotkane problemy	46
6.35	Pierwsze wersje Bullet Journal'a i Gamitude Themes	47
6.35.1	Założenia sprintu	47
6.35.2	Wykonane zadania	47
6.35.3	Napotkane problemy	47
6.36	47
6.36.1	Założenia sprintu	47
6.36.2	Wykonane zadania	47
6.36.3	Napotkane problemy	47
7	Testy systemu	48
8	Testy w grupach docelowych	49
9	Prezentacja systemu w działaniu	50
10	Nakład pracy	51

11 Wkład własny	52
11.1 Paweł Benkowski	52
11.2 Robert Deyk	52
11.3 Stanisław Lutkiewicz	52
12 Podsumowanie	53
Bibliografia	54
13 Załączniki	55

Rozdział 1

Wprowadzenie

1.1 Kontekst pracy

Energie[1] dzielą się na 4 rodzaje: duszy, ciała, emocji i umysłu. Istnieją różne techniki próbujące wspomóc ich efektywne wykorzystywanie jak metoda Pomodoro[2] czy Ultradian Rhythm[3].

Pierwsza technika polega na dzieleniu czasu pracy na 25-minutowe sesje, po których następuje 5-minutowa przerwa. Po wykonaniu 5 takich sesji przerwa wydłuża się do 15 minut. Technika ta jest szczególnie przydatna podczas nauki.

Jak wiadomo z badań, po nauce należy dać mózgowi czas na 'przetworzenie' nowych informacji. Tworzy on w tym czasie nowe synapsy, dołączając nowo nabytą wiedzę do istniejącej sieci informacji. Jeżeli napotka terminy podobne do nowej informacji, połączy je, umacniając strukturę.

Druga technika polega na dzieleniu czasu na pracy na 90-minutowe sesje, po których następuje 30-minutowa przerwa. Technika ta jest oparta na naturalnym cyklu ludzkiej aktywności podczas dnia. Ultradian Rhythm jest szczególnie przydatny przy zadaniach rutynowych bądź częściowo rutynowych. Przykładem takiego zadania jest praca.

Są to tylko dwie najpopularniejsze metodyki pracy, dzięki którym zwiększamy swoją wydajność, ale wciąż prowadzimy badania nad tym tematem, szukając nowych technik, tak aby Gamitude oferował, jak najszersze możliwości personalizacji.

Niekiedy jednak projekty trzeba rozplanować jako zadania w kontekście czasu. Na takie okazje planujemy stworzyć Bullet Journal[?]'a. Jest to dziennik ze stronami dedykowanymi na zadania z terminem na dzisiaj, na za tydzień, na za miesiąc bądź w nieokreślonej przyszłości.

Aby użytkownik czuł, że jego praca przynosi efekty dodamy również elementy grywalizacji[4], czyli statystyki oraz rangi[5], gdzie statystyki można rozumieć jako sztuczną walutę podzieloną na 4 rodzaje a rangi jako awatara użytkownika kupowanego za te waluty.

1.2 Przedstawienie problemu

Według pracy harwardzkiej, ludzie organizujący sobie pracę, mają tendencje do siadania i wykonywania jej bez przerwy aż do ukończenia, ignorując holistyczną naturę działania ludzkiego organizmu oraz naturalne cykle zachodzące w ludzkim ciele. W pracy czas jest porównywany do węgla a energia do wiatru. Jest to nawiązanie do przemysłu energetycznego i cech źródeł odnawialnych i nieodnawialnych.

1.3 Cel i zakres projektu

Celem jest stworzenie systemu, który swoje działanie bazuje na energiach omówionych wcześniej, zwracając także uwagę na nie przepracowywanie użytkowników. Zakresem projektu jest stworzenie zestawu narzędzi szanujących

higienę pracy oraz indywidualne preferencje organizacji pracy przez użytkowników.

1.4 Podejście do projektu

Zwinne, Scrumban, aplikacja webowa oparta o JS, backend w .NET, wybrane ze względu na wieloplatformowość.

1.5 Rezultaty

Ludzie potrafią utrzymać produktywność w dłuższym okresie, bez ubytków w żadnej z dziedzin życia.

1.6 Organizacja dokumentu

Omówienie problemu, analiza projektu, projektowanie i implementacja systemu, historia implementacji, testy systemu, testy w grupie docelowej, nakład pracy, wkład własny, podsumowanie

1.7 Słownik pojęć

Bullet Journal – dziennik, w którym można rozplanowywać zadania

Elastic Habits – stopniowanie zadania na 3 poziomy

Projekt – Czynność bądź umiejętność, nad którą użytkownik chce pracować, nie ma podziału na podpunkty, skupiający się na czasie

Statystyki-

Energie-

Rangi-

Motywy-

Tier-

Foldery-

Notatniki-

Strony-

Zadania-

Boosted statistics-

Dominant statistic-

Czasomierz-

Rozdział 2

Omówienie problemu

2.1 Przedstawienie problemu

Problem a ludzie

Problemem jest bycie produktywnym.

Są ludzie, którzy nie robią absolutnie nic, dopóki nie są do tego przymuszeni.

Są tacy, którzy z każdym zadaniem czekają do ostatniej chwili, potem pracują się i powtarzają ten cykl.

Są też tacy, którzy starają się za bardzo i gdy w końcu ich organizm się buntuje, popadają w długie okresy nieróbstwa.

I na koniec są też pracoholicy, którzy tłumią zbuntowany organizm i dopiero choroba taka jak rak odciąga ich od pracy.

Wszystko to są objawy braku higieny pracy i nieefektywność w wykorzystywaniu energii.

Człowiek nierobiący nic zawsze ma jej pełno, chociaż wydaje mu się, że nie ma jej wcale.

Prokrastynator ma wrażenie, że energia pojawia się znikąd dopiero chwile przed terminem oddania.

Ci, którzy się starają, na początku każdego okresu nieróbstwa mają wrażenie, że po raz kolejny przegrali walkę z lenistwem.

Pracoholicy natomiast mają wrażenie, że są pełni energii, chociaż od dawna jadą na pustym baku.

Chociaż nie każdy popada w wymienione ekstrema, dobrze obrazują one ogólny problem z efektywnym wykorzystaniem energii.

Większość ludzkości nie utrzymuje stałego tempa pracy, na bieżąco wykonując wszystkie zaplanowane zadania.

Zamiast tego wahają się między stanami „Jestem do tyłu z robotą, trzeba się

spieszyć” a „Jestem do przodu z robotą, mogę teraz tydzień nie nie robić”.

Co i dlaczego nie działa?

Ludzie operują w kontekście czasu i właśnie go próbują rozplanowywać. Jednakże czas jest surowcem skończonym, którego wszyscy ludzie mają dokładnie tyle samo.

Posiadając już tę wiedzę, ludzie wpadają w następną pułapkę hasłowego „nie liczy się to, ile masz czasu, ale jak go wykorzystasz”.

Jednak taki sposób myślenia jest typowy dla ludzi przepracowujących się, ponieważ czują, że czas im ucieka, i muszą za wszelką cenę nie dopuścić do zmarnowania choćby sekundy.

To pokazuje, jak chorobliwe jest skupienie myśli na czasie i zignorowanie lub błędna interpretacja sygnałów własnego organizmu.

2.2 Rich picture

2.3 Konkurencyjne rozwiązania

2.3.1 Brain Focus

O Aplikacji

Brain Focus jest to aplikacja mobilna, pozwalająca na odmierzenie czasu w sesjach z przerwami. Nie jest skomplikowana w użyciu, poprzez dość prosty wygląd co nie rozprasza użytkownika. Daje ona możliwość pełnej modyfikacji ich minutnika od czasu pracy i długości przerwy do ich ilości i częstotliwości występowania dłuższej przerwy. Posiada ona również funkcję blokowania innych aplikacji, by pomóc użytkownikom pozostać skupionym. Aplikacja również daje nam dostęp do ogólnych statystyk użytkownika tj. średnia długość pracy nad zadaniami itp.

Zalety i wady

Na pewno zaletami aplikacji jest jej prosty wygląd, jak i pełna możliwość modyfikacji minutnika, który może zostać przypisany do konkretnego zadania.

Do jej wad można zaliczyć brak nagradzania użytkownika za jego poświęcony czas z aplikacją.

Chcielibyśmy wykorzystać ich zalety w formie pełnej modyfikacji minutnika razem z możliwością przypisywania go do zadania, ponieważ w naszym rozwiązaniu oczekujemy, że użytkownicy będą mieli bardzo zróżnicowane projekty, co spowoduje potrzebę modyfikacji minutnika pod projekty.

Model biznesowy i popularność

Aplikacja ta używa modelu biznesowego ‘freemium’ gdzie w podstawowej formie aplikacja nie wysyła powiadomień oraz ma limitowane tworzenie kategorii zadań natomiast w wersji Pro po zakupieniu mamy nielimitowane tworzenie kategorii zadań, dostajemy dostęp do powiadomień w aplikacji, uzyskujemy dostęp do jej widget’u, jak i dostęp do wszystkich przyszłych funkcji aplikacji.

Obecnie popularność aplikacji na platformie Google Play w liczbie pobrań przekracza 1000000 pobrań.

2.3.2 Productivity Challenge Timer

O Aplikacji

PCT jest to aplikacja mobilna, pozwalająca na dodawanie projektów, nad którymi chcemy pracować i umożliwia nam pracę nad nimi z odliczaniem czasu pracy w formie minutnika. Wraz z systematycznym pracowaniem nad naszymi projektami, aplikacja nagradza nas nowymi rangami w zależności od poświęconego czasu i systematyczności. Aplikacja prowadzi podstawowe statystyki tj. ilość sesji wykonana przy danym projekcie w danym dniu czy średnia ilość sesji w tygodniu. Jest także system osiągnięć, które zdobywamy wraz z używaniem aplikacji.

Zalety i wady

Zaletą aplikacji jest to, że aplikacja wynagradza użytkownika za spędzany z nią czas w formie rang, które ulepszają się wraz z systematyczną pracą.

Do jej wad możemy zaliczyć ograniczenie możliwości tworzenia projektów do 4 w darmowej wersji aplikacji, ograniczona edycja minutnika, jak i duże kary za brak systematyczności w formie dużych spadków w rangach użytkownika. System może również powodować przyśpieszone wypalenie, bądź nieefektywną pracę poprzez zachęcanie do ciągłej, nieprzerwanej pracy.

Chcielibyśmy wykorzystać formę gratyfikacji użytkownika w formie rang, tak by nasza aplikacja jednocześnie motywowała naszych użytkowników do dalej pracy z systemem, jak i wykazywała postępy osiągnięte przez użytkownika.

Model biznesowy i popularność

Aplikacja używa modelu ‘freemium’ gdzie w darmowej wersji jesteśmy ograniczeni do stworzenia 4 projektów, otrzymujemy agresywne reklamy w aplikacji, jak i mamy ograniczoną liczbę osiągnięć do zdobycia. Natomiast w

wersji PRO mamy nieograniczoną liczbę projektów do tworzenia, nie dostajemy reklam, odblokowujemy dostęp do obecnie niemożliwych do zdobycia osiągnięć, jak i w przyszłości nowych.

Obecnie popularność aplikacji na platformie Google Play w liczbie pobrań przekracza 500000 pobrań.

2.3.3 Habitica

O Aplikacji

Habitica to aplikacja mobilna, jak i webowa, w której tworzymy swoją postać i ulepszamy ją poprzez wykonywanie zadań. Użytkownik na początku tworzy postać, którą rozwija wraz z używaniem aplikacji. Postać posiada statystyki tj. siła, inteligencja, kondycja i percepcja. Wraz z wykonywaniem zadań, postać zadaje określone punkty obrażeń stworą, które jeśli zostaną pokonane, dają postaci punkty doświadczenia. Punkty doświadczenia po osiągnięciu pewnego progu, zwiększają poziom postaci. Statystyki możemy zwiększać poprzez dodawanie ich za dostępne punkty po osiągnięciu poziomu bądź z posiadanych przedmiotów.

Zalety i wady

Zaletą aplikacji jest widoczna gamifikacja, co przyciąga dużą grupę użytkowników, która kojarzy owe systemy doświadczenia i statystyk z gier komputerowych. Zaletą również jest element aplikacji, gdzie tworzymy drużynę z innymi graczami, by pokonać silniejsze stwory.

Wadą aplikacji jest brak możliwości śledzenia czasu na wykonanie zadania bądź brak jakiegokolwiek minutnika, zachęcającego do systematycznej pracy. Wadą także jest duże skomplikowanie aplikacji, co może odstraszyć nowych użytkowników.

Chcielibyśmy wykorzystać system statystyk konta w naszym systemie, ponieważ idzie to w parze z osiąganiem rang jako forma nagradzania użytkownika za spędzany czas.

Model biznesowy i popularność

Model biznesowy aplikacji opiera się o mikro transakcje, gdzie możemy wykupywać walutę premium w grze, by osiągnąć rzeczy niedostępne dla użytkowników bez owej waluty tj. wyglądy postaci, misje na otrzymanie towarzyszy w formie chowańców czy możliwość zresetowania postaci.

Obecnie popularność aplikacji na platformie Google Play w liczbie pobrań

przekracza 1000000 pobrań oraz szacowana ilość całkowita wszystkich użytkowników oscyluje w granicach 4000000 użytkowników.

2.3.4 Pomodone

O Aplikacji

Pomodone to aplikacja webowa, jak i mobilna, gdzie mamy możliwość dodawania zadań do wykonania oraz mamy możliwość włączenia sesji pracy nad konkretnym zadaniem. Aplikacja pozwala nam na zablokowanie konkretnych stron bądź aplikacji, by zapewnić użytkownikom pełne skupienie nad wykonanym zadaniem.

Zalety i wady

Zaletą aplikacji jest jej wieloplatformowość, ponieważ jest dostępna na każdym systemie operacyjnym, na komputerach stacjonarnych, systemach mobilnych, jak i przeglądarkach w formie widget'u. Kolejną zaletą jest też duża możliwość modyfikacji minutnika.

Wadą jest pozornie prosty wygląd aplikacji, w której bardzo łatwo można się zgubić, będąc nowym użytkownikiem. Wadą także można by nazwać brak podpowiedzi dla użytkownika ,jeśli zgubiłby się w aplikacji.

Chcielibyśmy nie popełnić błędu, gdzie użytkownik bardzo łatwo może się zgubić w aplikacji, dlatego chcemy wytworzyć system, gdzie wszystkie potrzebne informacje zawsze będą na widoku oraz użytkownicy również będą mieli do dyspozycji podpowiedzi ,jeśli miałby trudności z korzystania z systemu.

Model biznesowy i popularność

Model biznesowy aplikacji używa płatnej subskrypcji do korzystania z aplikacji. Dla studentów jest rok darmowej subskrypcji dzięki programowi Github Education.

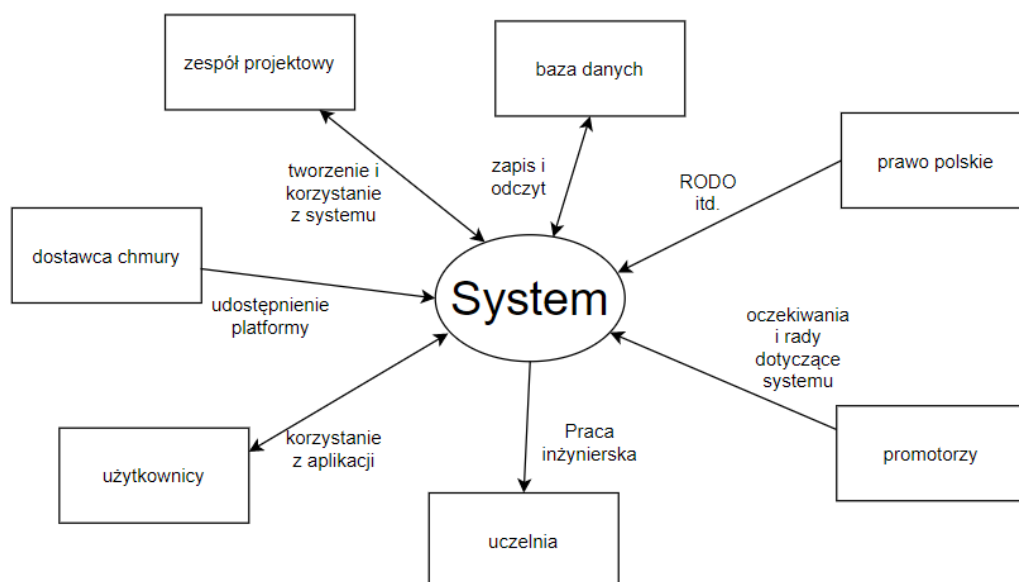
Obecnie popularność aplikacji na platformie Google Play w liczbie pobrań przekracza 10000 pobrań. Brak informacji o użytkownikach z innych platform.

2.4 Propozycja rozwiązania

Opis ogólny działania naszej aplikacji pozwalający na zrozumieniu podstawowych działań, na jakie nasz system będzie pozwalał.

2.5 Kontekst systemu

Aplikacja webowa kompatybilna z większością przeglądarek i komputerów z dostępem do Internetu. Można z niej korzystać o dowolnej porze dnia, używając urządzeń mobilnych. Chcemy podzielić projekt na mikro serwisy, pozwoli nam to na łatwą skalowalność i późniejsze pielęgnowanie projektu. Nasz system będzie się składał z wielu funkcjonalności, z których użytkownik będzie mógł zarządzać swoją pracą tj. zarządzanie projektami, Bullet Journal czy Elastic Habits[6] wspierane przez system rang, system energii czy system osiągnięć. System zarządzający projektami pomaga nam dzielić sobie nasze zadania na sesję o określonym wymiarze czasowym wybranym przez użytkownika. Okresy te bazowano na technikach tj. Pomodoro[2] 25/5, Ultradian Rhythm[3]90/30, flow state[7] oraz Just 5[8]. Wykonywanie projektów jest nagradzane statystykami, które zwiększają się w zależności, jakiego typu był projekt. Po spełnieniu danego warunku użytkownik może zostać nagrodzony osiągnięciem za przekroczenie pewnego kamienia milowego w swojej pracy nad projektem. Zadaniem Bullet Journal'a jest rozplanowanie pomniejszych zadań z projektu w czasie i wizualizacja ich na tablicy, by użytkownik mógł śledzić jakie zadania musi wykonać danego dnia, by wyrobić się w terminie. Elastic Habits miałby za zadanie pomóc użytkownikowi wyrobić sobie nawyk, poprzez poziomowanie sobie zaplanowanego zadania. W zależności od ogólnego samopoczucia użytkownika może on wybrać łatwiejszą bądź trudniejszą wersję zadania, dalej utrzymując nawyk wykonywania go.



2.6 Cele i odbiorcy systemu

2.6.1 Cele systemu

Chcemy stworzyć nowe narzędzie do organizacji pracy z oryginalnym podejściem zaczerpniętym z systemów RPG, opartym na odkryciach w dziedzinie zarządzania energią[1]. Chcemy aby użytkownicy korzystający z naszego narzędzia nie zarządzali wyłącznie swoim czasem, ale także i energią, która jest równie ważna. Efektem końcowym będzie aplikacja internetowa. W przyszłości planujemy rozszerzyć system o aplikację mobilną oraz desktopową. Spodziewaną korzyścią będzie wzrost produktywności długoterminowej wśród użytkowników. Produktywność użytkowników moglibyśmy sprawdzać poprzez ankiety, porównujące czas poświęcony na zadania bez korzystania z systemu oraz z nim np. użytkownik skończył zaplanowane zadanie tydzień szybciej lub pracował przez 2 godziny dłużej niż bez korzystania z systemu.

2.6.2 Udziałowcy(należy ponownie rozważyć)

Karta Udziałowca	
Indentyfikator	UOB01
Nazwa	Zespół projektowy
Opis	Zespół projektowy tworzy oraz opiekuje się systemem
Typ udziałowca	Ożywiony, bezpośredni
Punkt widzenia	Perspektywa twórców systemu
Ograniczenia	Brak
Wymagania	
Karta Udziałowca	
Indentyfikator	UOB02
Nazwa	Użytkownik końcowy
Opis	Przeciętny, finalny użytkownik korzystający z aplikacji
Typ udziałowca	Ożywiony, bezpośredni
Punkt widzenia	Perspektywa użytkownika
Ograniczenia	Nie ma dostępu do warstwy technicznej - bazy danych, kodu itp.
Wymagania	
Karta Udziałowca	
Indentyfikator	UOB03
Nazwa	Sponsorzy
Opis	Osoba, która finansuje projekt i egzekwuje wymagania
Typ udziałowca	Ożywiony, bezpośredni
Punkt widzenia	Perspektywa ekonomiczna
Ograniczenia	Nie powinien narzucać technologii przy tworzeniu projektu
Wymagania	

Karta Udziałowca	
Indentyfikator	UOB04
Nazwa	Wydawca
Opis	Osoba, która jest odpowiedzialna za sfinalizowanie projektu i wypuszczenie go na rynek
Typ udziałowca	Ożywiony, bezpośredni
Punkt widzenia	Perspektywa ekonomiczna
Ograniczenia	Nie powinien narzucać technologii przy tworzeniu projektu
Wymagania	
Karta Udziałowca	
Indentyfikator	UOB05
Nazwa	Promotorzy
Opis	Doradcy w sprawach dotyczących projektu
Typ udziałowca	Ożywiony, bezpośredni
Punkt widzenia	Perspektywa twórców projektu
Ograniczenia	Nie zawsze dostępni
Wymagania	
Karta Udziałowca	
Indentyfikator	UNB01
Nazwa	Media
Opis	Strony internetowe, reklamy, artykuły, audycje itp.
Typ udziałowca	Nieożywiony, bezpośredni
Punkt widzenia	Perspektywa ekonomiczna
Ograniczenia	Zero wpływu na budowę projektu
Wymagania	
Karta Udziałowca	
Indentyfikator	UNB02
Nazwa	Baza danych
Opis	Jedna, wspólna baza danych na cały system
Typ udziałowca	Nieożywiony, bezpośredni
Punkt widzenia	Perspektywa techniczna
Ograniczenia	Skończona ilość pamięci do przechowywanie informacji
Wymagania	
Karta Udziałowca	
Indentyfikator	UNB03
Nazwa	Prawo polskie
Opis	Zgodnie z RODO mamy obowiązek dbać o bezpieczeństwo danych osobowych wszystkich użytkowników
Typ udziałowca	Nieożywiony, bezpośredni
Punkt widzenia	Perspektywa prawna
Ograniczenia	Brak
Wymagania	

Karta Udziałowca	
Indentyfikator	UNB04
Nazwa	Dostawca usług chmurowych
Opis	Serwer w chmurze odpowiedzialny za przetwarzanie wszystkich żądań pomiędzy serwisami i użytkownikami końcowymi
Typ udziałowca	Nieożywiony, bezpośredni
Punkt widzenia	Perspektywa techniczna
Ograniczenia	Ograniczenia sprzętowe, specyfikacja sprzętu (np. moc procesora serwerowego, przepustowość Internetu)
Wymagania	

2.6.3 Grupa docelowa

Grupa docelowa to osoby, które chcą poprawić swoją higienę pracy, chcą stać się systematyczni, uniknąć przepracowania. (ładniej opisać)

Rozdział 3

Analiza

3.1 Wymagania(w trakcie ponownego rozważania)

3.1.1 Wymagania ogólne i dziedzinowe

Karta Wymagania			
Indentyfikator:	W01	Priorytet:	M - must(musi być)
Nazwa	Zwiększenie efektywności pracy użytkowników systemu		
Opis	Końcowy produkt systemu ma za zadanie zwiększać produktywność jego użytkowników, weryfikowane jest to na podstawie prac z Harvardu i user feedback'u w postaci ankiet.		
Udziałowiec	Wydawca, Użytkownik końcowy		
Wymagania powiązane	Brak		

Karta Wymagania			
Indentyfikator:	W01	Priorytet:	M - must(musi być)
Nazwa	System skórek		
Opis	Produkt będzie zarabiał na sprzedaży skórek(alternatywna oprawa graficzna i personalizacja systemu pracy)		
Udziałowiec	Wydawca, Sponsorzy		
Wymagania powiązane	Brak		

3.1.2 Wymagania funkcjonalne

- Nazwa funkcji/usługi

Karta Wymagania			
Indentyfikator:	F01	Priorytet:	M - must(musi być)
Nazwa	System Autoryzacji użytkownika		
Opis	Jako użytkownik muszę mieć możliwość zarejestrowania się w serwisie i późniejszego logowania się		
Kryteria akceptacji	Bezpieczny system autoryzacji zabezpieczony przed atakami na bazę danych, potwierdzenie maila po rejestracji, jedno konta na 1 mail		
Dane wejściowe	Brak		
Warunki początkowe	Brak		
Warunki końcowe	Brak		
Sytuacje wyjątkowe	Brak		
Szczegóły implementacji	Brak		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		
Karta Wymagania			
Indentyfikator:	F02	Priorytet:	M - must(musi być)
Nazwa	System rang użytkowników		
Opis	Jako użytkownik podczas progresowania w trakcie używania aplikacji chciałbym być przypisywany do różnych rang		
Kryteria akceptacji	Przypisywanie rangi do danego użytkownika oraz obliczanie jego statystyk na podstawie wykonywanych projektów		
Dane wejściowe	Brak		
Warunki początkowe	Brak		
Warunki końcowe	Brak		
Sytuacje wyjątkowe	Brak		
Szczegóły implementacji	Brak		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		

Karta Wymagania			
Indentyfikator:	F03	Priorytet:	M - must(musi być)
Nazwa	System zarządzania energią użytkownika		
Opis	Jako użytkownik chcę żeby aplikacja śledziła moje zasoby energetyczne i podpowiadała jak mogę nimi lepiej zarządzać		
Kryteria akceptacji	Zmiana zasobów energii użytkownika przy wykonywaniu konkretnych projektów lub przerw		
Dane wejściowe	Brak		
Warunki początkowe	Brak		
Warunki końcowe	Brak		
Sytuacje wyjątkowe	Brak		
Szczegóły implementacji	Brak		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		

Karta Wymagania			
Indentyfikator:	F04	Priorytet:	M - must(musi być)
Nazwa	System zarządzania projektami użytkowników		
Opis	Jako użytkownik chciałbym mieć możliwość dodawania, usuwania i śledzenia projektów lub zadań		
Kryteria akceptacji	Użytkownik ma możliwość dodawania, usuwania i śledzenia projektów przez siebie stworzonych		
Dane wejściowe	Brak		
Warunki początkowe	Brak		
Warunki końcowe	Brak		
Sytuacje wyjątkowe	Brak		
Szczegóły implementacji	Brak		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		

Karta Wymagania			
Indentyfikator:	F05	Priorytet:	M - must(musi być)
Nazwa	Bullet Journal		
Opis	Jako użytkownik chcę mieć możliwość rozplanowania zadań na dni oraz zobaczenia ich rozłożonych w czasie na tablicy lub w postaci kalendarza		
Kryteria akceptacji	Użytkownik może dodawać swoje zadania wraz z datami ich wykonania, które zostają zwizualizowane w postaci tablicy lub kalendarza		
Dane wejściowe	Brak		
Warunki początkowe	Brak		
Warunki końcowe	Brak		
Sytuacje wyjątkowe	Brak		
Szczegóły implementacji	Brak		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		
Karta Wymagania			
Indentyfikator:	F06	Priorytet:	C – could
Nazwa	System osiągnięć użytkownika		
Opis	Jako użytkownik chciałbym co jakiś czas być nagradzany za osiągnięcia przy dochodzeniu do kamieni milowych podczas korzystania z aplikacji		
Kryteria akceptacji	Użytkownik otrzymuje osiągnięcia za przekroczenie pewnych kamieni milowych		
Dane wejściowe	Brak		
Warunki początkowe	Brak		
Warunki końcowe	Brak		
Sytuacje wyjątkowe	Brak		
Szczegóły implementacji	Brak		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		

Karta Wymagania			
Indentyfikator:	F07	Priorytet:	C – could
Nazwa	System rankingowy użytkowników		
Opis	Jako użytkownik chciałbym mieć dostęp do tablic rankingowych gdzie mógłbym porównywać swoje osiągnięcia z innymi użytkownikami		
Kryteria akceptacji	Użytkownik jest w stanie sprawdzić swoją pozycję w rankingu dotyczącą danego projektu		
Dane wejściowe	Brak		
Warunki początkowe	Brak		
Warunki końcowe	Brak		
Sytuacje wyjątkowe	Brak		
Szczegóły implementacji	Brak		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		
Karta Wymagania			
Indentyfikator:	F08	Priorytet:	C – could
Nazwa	System znajomych użytkowników		
Opis	Jako użytkownik chciałbym móc dodawać innych użytkowników do swojej listy znajomych żeby sprawdzać ich postępy		
Kryteria akceptacji	Użytkownik może dodawać znajomych, wyświetlanych w formie listy, u których może sprawdzać ich postępy		
Dane wejściowe	Brak		
Warunki początkowe	Brak		
Warunki końcowe	Brak		
Sytuacje wyjątkowe	Brak		
Szczegóły implementacji	Brak		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		

Karta Wymagania			
Indentyfikator:	F09	Priorytet:	C – could
Nazwa	Energy Assistant		
Opis	Jako użytkownik chciałbym żeby moje rzeczywiste poziomy energii były lepiej rozpoznawane		
Kryteria akceptacji	Użytkownik zależnie jaki prowadzi tryb życia, bądź w zależności od jego warunków fizycznych jak i psychicznych miałby dostosowywaną ilość energii na dany dzień		
Dane wejściowe	Brak		
Warunki początkowe	Brak		
Warunki końcowe	Brak		
Sytuacje wyjątkowe	Brak		
Szczegóły implementacji	Brak		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		
Karta Wymagania			
Indentyfikator:	F10	Priorytet:	C – could
Nazwa	Elastic Habits		
Opis	Jako użytkownik chciałbym móc podzielić swoje zadanie na różne poziomy trudności		
Kryteria akceptacji	Użytkownik może wybrać z dany poziom trudności zadania by wyrobić sobie nawyk wykonywania go.		
Dane wejściowe	Brak		
Warunki początkowe	Brak		
Warunki końcowe	Brak		
Sytuacje wyjątkowe	Brak		
Szczegóły implementacji	Brak		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		

- Interfejs z otoczeniem

Karta Wymagania			
Indentyfikator:	I01	Priorytet:	M – must (musi być)
Nazwa	Integracja mikro serwisów		
Opis	Nasz projekt strukturalnie będzie zbudowany z wielu mikro serwisów i wymagana jest integracja między nimi(komunikatywność)		
Kryteria akceptacji	Funkcje w każdym serwisie umożliwiające komunikowanie się z innymi serwisami		
Dane wejściowe	Brak		
Warunki początkowe	Brak		
Warunki końcowe	Brak		
Sytuacje wyjątkowe	Brak		
Szczegóły implementacji	Brak		
Udziałowiec	Zespół projektowy		
Wymagania powiązane	Brak		
Karta Wymagania			
Indentyfikator:	I02	Priorytet:	M – must (musi być)
Nazwa	Baza danych		
Opis	Jedna zintegrowana baza danych dla wszystkich serwisów		
Kryteria akceptacji	Baza danych w MongoDB która będzie obsługiwać wszystkie mikro serwisy projektu, będzie posiadała dane użytkowników i wszystkiego co jest związane z aplikacją		
Dane wejściowe	Brak		
Warunki początkowe	Brak		
Warunki końcowe	Brak		
Sytuacje wyjątkowe	Brak		
Szczegóły implementacji	Brak		
Udziałowiec	Zespół projektowy		
Wymagania powiązane	Brak		

3.1.3 Wymagania pozafunkcjonalne

Karta Wymagania			
Indentyfikator:	NF01	Priorytet:	M – must (musi być)
Nazwa	C#		
Opis	Serwisy backendowe powinny być napisane w C#		
Udziałowiec	Zespół projektowy, Wydawca		
Wymagania powiązane	Brak		
Karta Wymagania			
Indentyfikator:	NF02	Priorytet:	M – must (musi być)
Nazwa	React.js		
Opis	Serwis frontendowy powinien być napisany korzystając z biblioteki React.js		
Udziałowiec	Zespół projektowy, Wydawca		
Wymagania powiązane	Brak		
Karta Wymagania			
Indentyfikator:	NF03	Priorytet:	M – must (musi być)
Nazwa	Czas wdrożenia		
Opis	System należy wdrożyć do końca semestru zimowego 2020/2021		
Udziałowiec	Zespół projektowy, Promotorzy, Uczelnia, Wydawca		
Wymagania powiązane	Brak		
Karta Wymagania			
Indentyfikator:	NF04	Priorytet:	M – must (musi być)
Nazwa	System powinien być dostępny 24 godziny na dobę, 7 dni w tygodniu		
Opis	Dostęp do systemu powinien być umożliwiony w dowolnej chwili danego dnia		
Udziałowiec	Zespół projektowy, Użytkownik końcowy, Wydawca		
Wymagania powiązane	Brak		

3.1.4 Wymagania na środowisko docelowe

Karta Wymagania			
Indentyfikator:	ŚD01	Priorytet:	M – must (musi być)
Nazwa	Kompatybilność przeglądarek		
Opis	Produkt końcowy musi być kompatybilny z 3 najnowszymi wersjami popularnych przeglądarek		
Kryteria akceptacji	System kompatybilny z 3 najnowszymi wersjami przeglądarek Google Chrome i Mozilla Firefox		
Udziałowiec	Zespół projektowy, Użytkownik końcowy, Wydawca		
Wymagania powiązane	Brak		

3.2 Wymagania jakościowe i inne

- System szyfrowania danych użytkowników: szyfrowanie transmisji, hasel i ciasteczek
- Wsparcie dla ostatnich trzech wersji Google Chrome i Mozilla Firefox
- System dostępny 24 godziny na dobę z wyłączeniem prac technicznych
- System ma być łatwy w obsłudze – średnio 5 kliknięć na wykonanie dowolnej

3.3 Przypadki użycia oraz diagramy (w trakcie przetwarzania i tworzenia nowych)

3.3.1 Przypadki użycia

3.3.2 Diagramy

Rozdział 4

Planowanie

4.1 Metodyka pracy

Po dłuższym namyśle zdecydowaliśmy, że dobrym dla nas podejściem byłoby podążanie sprintami z metodologii Scrum oraz posiadanie tablicy zadań z metodologii Kanban[9]. Sprinty wymuszają na nas ciągłą, stałą pracę by co tydzień wypuszczać nowe wersje naszego systemu. Zapewnia to stałą motywację do pracy by nie osiągnąć punktu długotrwałej stagnacji w projekcie. Tablica zadań z metodologii Kanban pozwala nam na jasne podzielenie zadań w zespole projektowym oraz ułatwia określenie w jakim stopniu dane zadanie jest wykonane. Wybraliśmy tą metodykę ze względu na fakt, że nasz system jest dość nowatorski w swojej kategorii, przez co wymaga stałej weryfikacji przez użytkowników.

4.2 Narzędzia

Opisanie narzędzi wykorzystanych do tworzenia projektu (slack, azure dev, google cloud itd.)

4.3 Technologie

- React/Redux/MUI - frontend
- .NET - Backend
- MongoDB/Postgres - baza danych

Rozdział 5

Projektowanie i finalna implementacja

5.1 Architektura projektu

5.1.1 Frontend

5.1.2 Backend

5.1.3 Baza Danych

5.1.4 Chmura

5.2 Implementacja rozwiązania

Rozdział 6

Historia sprintów

6.1 Rozpoznanie dziedzinowe

6.1.1 Założenia sprintu

Każdy członek zespołu wybrał sobie pewien sposób na zarządzanie swoimi zadaniami w ciągu dnia w celu zbadania dziedziny problemu.

6.1.2 Wykonane zadania

Paweł: Używanie metodologii Pomodoro i Ultradian rytm do codziennych zadań. Przerabianie kursu React.

Robert: Rozplanowywanie zadań za pomocą GTD (Getting Things Done)

Stanisław: Zapoznanie się i zastosowanie Bullet Journal'a oraz szukanie alternatyw dla narzędzi do dokumentowania projektu (Enterprise Architect, Github).

6.1.3 Napotkane problemy

Brak zastępcy dla programu Enterprise Architect. Spór dotyczący wyboru Azure Repos a Github'em.

6.2 Wstępna dokumentacja

6.2.1 Założenia sprintu

Podczas sprintu, zespół miał za zadanie wytworzyć wstępne dokumenty DZW i SWS oraz znalezienie sposobu na wersjonowanie w EA.

6.2.2 Wykonane zadania

Paweł: Wytworzenie wstępnej wersji dokumentu DZW. Przerabianie kursu React.

Robert: Wytworzenie wstępnej wersji dokumentu SWS.

Stanisław: Rozpoznanie dotyczące wersjonowania w EA.

6.2.3 Napotkane problemy

Wersjonowanie w EA dostępne tylko po wykupieniu licencji.

6.3 Use Case + WPP

6.3.1 Założenia sprintu

Przygotowanie pierwszych diagramów Use Case. Wytworzenie dokumentu WPP.

6.3.2 Wykonane zadania

Paweł: Wytworzenie paru Use Case'ów w programie EA. Przerabianie kursu React.

Robert: Stworzenie dokumentu WPP oraz jednego z Use Case'ów.

Stanisław: Wytworzenie paru Use Case'ów w programie EA.

6.3.3 Napotkane problemy

Brak napotkanych problemów.

6.4 Dalsza praca przy SWS

6.4.1 Założenia sprintu

Doprecyzowanie wymagań systemowych oraz нефункциональных w dokumencie SWS. Podjąć decyzję odnośnie architektury systemu.

6.4.2 Wykonane zadania

Zespół: Wspólna praca nad SWS oraz wybranie architektury mikroserwisowej dla systemu.

Paweł: Przerabianie kursu React.

6.4.3 Napotkane problemy

Brak napotkanych problemów.

6.5 Przygotowanie narzędzi do tworzenia systemu

6.5.1 Założenia sprintu

Znalezienie i skonfigurowanie platformy do komunikacji zespołu, stworzenie repozytoriów dla każdego mikroserwisu oraz przygotowanie wstępnego mockup'u UI dla systemu.

6.5.2 Wykonane zadania

Paweł: Stworzenie pierwszego mockup'u UI dla systemu. Przerabianie kursu React.

Stanisław: Stworzenie repozytoriów dla wszystkich mikroservisów oraz przygotowanie platformy Slack do komunikacji zespołu.

6.5.3 Napotkane problemy

Rozmowy grupowe na platformie Slack były płatne, więc musieliśmy znaleźć alternatywę dla domyślnych rozmów.

6.6 Diagramy funkcjonalności, wymagania bazy danych oraz responsywne rozłożenie strony

6.6.1 Założenia sprintu

Każdy z członków zespołu stworzy diagram funkcjonalności dla swojego mikroservis, który pomoże w określeniu wymagań funkcjonalnych. Utworzony zostanie diagram wymagań dla bazy danych oraz wykres Gantt'a i wstępne ułożenie poszczególnych komponentów na stronie.

6.6.2 Wykonane zadania

Zespół: Tworzenie wykresu Gantt'a.

Paweł: Utworzenie responsywnego rozłożenia strony.

Robert: Stworzenie diagramu funkcjonalności.

Stanisław: Stworzenie diagramu funkcjonalności oraz bazy danych.

6.6.3 Napotkane problemy

Znalezienie darmowego narzędzia dla wykresu Gantt'a.

6.7 Diagramy klas i nawigacja strony

6.7.1 Założenia sprintu

Członkowie zespołu odpowiedzialni za backend projektu wykonają diagramy klas dla swoich serwisów, utworzą struktury plików dla poszczególnych mikroserwisów, przygotowują routing dla nich. Zostanie stworzona nawigacja strony.

6.7.2 Wykonane zadania

Robert i Stanisław: Utworzenie diagramu klas dla mikroserwisów oraz struktury plików dla nich.

Paweł: Implementacja nawigacji strony.

Stanisław: Stworzenie routingu dla mikroserwisu projektów.

6.7.3 Napotkane problemy

Robert nie przygotował routingu.

6.8 Diagram sekwencji oraz statystyki i energie na stronie

6.8.1 Założenia sprintu

Zostanie utworzony diagram sekwencji pozwalający zobrazować przepływ danych w projekcie. Utworzone zostaną widoki statystyk i energii na stronie.

6.8.2 Wykonane zadania

Robert i Stanisław: Utworzenie diagramu sekwencji.

Paweł: Implementacja widoków statystyk i energii.

Robert: Utworzono routing w mikroserwisie rang.

Stanisław: Modyfikacja modelu projektów w bazie danych.

6.8.3 Napotkane problemy

Brak napotkanych problemów.

6.9 Algorytmy oraz rangi na stronie

6.9.1 Założenia sprintu

Stworzenie algorytmów odpowiedzialnych za przyznawanie rangi oraz za wzrost lub spadek energii. Zaimplementowanie struktury bazy danych zgodnej z dokumentacją oraz utworzenie rang na frontendzie.

6.9.2 Wykonane zadania

Paweł: Implementacja widoków rang.

Robert: Algorytm przyznawania rang i statystyk.

Stanisław: Algorytm wzrostu i spadku energii. Implementacja struktury bazy danych.

6.9.3 Napotkane problemy

Brak napotkanych problemów.

6.10 Firebase, Heroku, projekty

6.10.1 Założenia sprintu

Utworzenie konta firebase dla projektu, schematów modeli bazy danych dla mikroserwisów, przeniesienie bazy danych na platformę Heroku, utworzenie wyglądu projektów na frontendzie.

6.10.2 Wykonane zadania

Robert i Stanisław: Stworzenie schematów modeli bazy danych dla mikroserwisów.

Paweł: Implementacja widoków projektów. Refaktoryzacja dotychczasowego kodu.

Stanisław: Przeniesienie bazy danych na platformę Heroku. Utworzenie konta firebase dla projektu.

6.10.3 Napotkane problemy

Modele rang i workflow nie zgodne z założeniami, do poprawy w następnym sprint'ie.

6.11 Firebase autoryzacja, API refactor

6.11.1 Założenia sprintu

Dodanie rang do bazy danych, serwisu firebase do projektu, funkcji tworzenia użytkownika przez firebase, refaktoryzacja serwisów workflow i projektów, utworzenie strony logowania i rejestracji na frontendzie oraz poprawienie wyglądu projektów.

6.11.2 Wykonane zadania

Robert i Stanisław: Poprawa modeli rang i workflow.

Paweł: Utworzenie strony logowania i rejestracji na frontendzie oraz poprawienie wyglądu projektów, dodanie serwisu firebase do projektu oraz funkcji tworzenia użytkownika przez firebase.

Robert: Dodanie rang do bazy danych.

Stanisław: Refaktoryzacja serwisów workflow i projektów.

6.11.3 Napotkane problemy

Algorytmy niedostosowane do działania w czasie. Poprawa na następny tydzień.

6.12 Integracja

6.12.1 Założenia sprintu

Ukończenie prace nad algorytmami związanymi z przyznawaniem rang oraz zmianą energii, dodanie wszystkich działających endpoint'ów do serwisu frontendowego.

6.12.2 Wykonane zadania

Robert i Stanisław: Dostosowanie algorytmów do wymagań.

Paweł: Integracja z API.

6.12.3 Napotkane problemy

Problem z integracją frontendu z API.

6.13 Prezentacja i demo

6.13.1 Założenia sprintu

Utworzenie prezentacji projektu oraz pokazowego demo działania aplikacji.

6.13.2 Wykonane zadania

Robert: Wykonanie pokazowego demo.

Paweł: Stworzenie prezentacji.

6.13.3 Napotkane problemy

Przez problemy z integracją frontendu z API, część projektu musiała zostać postawiona na mockup'ach.

6.14 Zmiana technologii na backendzie, retrospekcja

6.14.1 Założenia sprintu

Przyjrzenie się powstałemu już systemowi i wyznaczenie kierunku dalszych prac.

6.14.2 Wykonane zadania

Zespół: Ustalenie zmiany technologii backendowej na C#, zaakceptowanie pomysłu na utworzenie sklepu z motywami dla aplikacji, wspólna decyzja o zrezygnowaniu z firebase'a.

Paweł: Utworzenie nowego formularza logowania i rejestracji bez firebase'a.

Stanisław: Rozpoczęcie prac nad przerabianiem serwisów z node.js na C#, praca nad serwisem autoryzacji użytkownika, przystosowanie chmury do pracy z technologią .NET, wyeksportowanie obecnej bazy danych do pliku lokalnego w razie zmiany bazy.

6.14.3 Napotkane problemy

Brak napotkanych problemów

6.15 Festiwal pomysłów i mockup'y

6.15.1 Założenia sprintu

Szukanie nowych pomysłów na funkcjonalności. Stworzenie mockupów dla strony głównej oraz bullet journala.

6.15.2 Wykonane zadania

Paweł: Elastic habits, Just 5, flow state, Energy asistant, system osiągnięć czy system rankingu między graczami.

Robert: Stworzenie mockup'u strony głównej oraz Bullet Journal'a.

Stanisław: Pomysł na zbieranie informacji o użytkownikach dla późniejszej sprzedaży.

6.15.3 Napotkane problemy

Brak napotkanych problemów

6.16 Festiwal pomysłów ciąg dalszy

6.16.1 Założenia sprintu

Zakończenie prac nad serwisem projektów oraz zaczęcie prac nad serwisem rang. Dalsze szukanie pomysłów.

6.16.2 Wykonane zadania

Paweł: skills, wybieranie innego zadania jako przerwy, tworzenie statystyk użytkownika z danych zadań.

Stanisław: Zakończono pracę nad serwisem projektów oraz zaczęto nad serwisem rang.

6.16.3 Napotkane problemy

Brak napotkanych problemów

6.17 Autoryzacja i budowanie bibliografii

6.17.1 Założenia sprintu

Połączenie frontendu z serwisem logowania i rejestracji. Zbudowanie podstawowej bibliografii.

6.17.2 Wykonane zadania

Paweł i Stanisław: Połączenie frontendu z przepisany serwisem.

Robert: Zebranie danych dotyczących źródeł informacji do bibliografii.

6.17.3 Napotkane problemy

Brak napotkanych problemów

6.18 Smartwatch'e i MBTI

6.18.1 Założenia sprintu

Szukanie informacji na temat wykorzystania smartwatchy w aplikacji. Rozpoznanie w zakresie różnych typów użytkowników systemu.

6.18.2 Wykonane zadania

Paweł i Robert: Wykonanie badania MBTI.

Stanisław: Pomysł na integrację aplikacji z Google Calendar.

6.18.3 Napotkane problemy

Brak napotkanych problemów

6.19 Odnowienie dokumentacji i bezpieczeństwo

6.19.1 Założenia sprintu

Zebranie całej starej dokumentacji w jedno miejsce. Zapewnienie bezpieczeństwa aplikacji.

6.19.2 Wykonane zadania

Zespół: Modyfikacja dokumentu DZW.

Robert: Zebranie całej dokumentacji i wstawienie jej na platformę Github, napisanie streszczenia projektu.

Stanisław: Szukanie informacji na temat certyfikatu ssl.

6.19.3 Napotkane problemy

Brak napotkanych problemów

6.20 Autoryzacja, dokumentacja

6.20.1 Założenia sprintu

Poprawa bibliografii, zebranie historii sprintów, poprawa wyglądu systemu autoryzacji użytkownika po stronie frontendu.

6.20.2 Wykonane zadania

Robert: Stworzenie bibliografii zgodnie z zaleceniami dziekana, skończenie spisywanie historii sprintów do obecnego stanu, wstępne poprawki do dokumentu SWS.

Paweł: Implementacja poprawek systemu autoryzacji użytkownika po stronie frontendu.

Stanisław: Research na temat architektury systemu.

6.20.3 Napotkane problemy

Brak napotkanych problemów

6.21 Końcowe prace nad przepisanyimi serwisami

6.21.1 Założenia sprintu

Zintegrowanie serwisu projektów, uaktualnienie dokumentów o uwagi promotora, kończenie prac nad serwisem statystyk i energii.

6.21.2 Wykonane zadania

Paweł: Integracja serwisu projektów, walidacja wprowadzonych danych na frontend.

Robert: Uzupełnienie DZW o nowe funkcjonalności, stworzenie dokumentu SWS oraz ulepszenie o uwagi promotora.

Stanisław: Końcowe prace nad serwisem statystyk i energii, naprawianie błędów w stworzonych serwisach.

6.21.3 Napotkane problemy

Brak napotkanych problemów

6.22 Integracja i LaTeX

6.22.1 Założenia sprintu

Integracja zakończonych mikroserwiów, przygotowania do przepisania dokumentacji na latex.

6.22.2 Wykonane zadania

Robert: Wstępna wersja widoku Gamitude Themes, research na temat języka do tworzenia dokumentacji LaTeX.

Paweł: Obsługa błędów dla projektów, integracja serwisu statystyk i energii.

Stanisław: skończenie serwisu statystyk, czytanie o mikroserwisach, prace nad cronem.

6.22.3 Napotkane problemy

Brak napotkanych problemów

6.23 Przepisywanie dokumentacji

6.23.1 Założenia sprintu

Stworzenie domyślnego dokumentu dla pracy oraz rozpoczęcie prac nad przepisywaniem pozostałych dokumentów.

6.23.2 Wykonane zadania

Robert: przerabianie dokumentów na latex.

Paweł: research na temat połączenia kalendarza i Bullet Journal'a.

Stanisław: przerabianie kursu o identity server 4.

6.23.3 Napotkane problemy

Brak napotkanych problemów

6.24 Podpowiedzi i naprawa błędów

6.24.1 Założenia sprintu

Dodanie podpowiedzi dla użytkownika by ułatwić zapoznavanie się z systemem. Naprawa błędów w przeliczaniu statystyk

6.24.2 Wykonane zadania

Robert: dodanie karty projektów w latex.

Paweł: Stworzenie podpowiedzi dla użytkowników na stronie.

Stanisław: Poprawa błędów w obliczaniu statystyk.

6.24.3 Napotkane problemy

Brak napotkanych problemów.

6.25 DZW i integracja rang

6.25.1 Założenia sprintu

Zintegrowanie przepisanego systemu rang, przepisanie dokumentu DZW na LaTeX.

6.25.2 Wykonane zadania

Robert: Dodanie DZW do latex dokumentu, przerobienie bibliografii.

Paweł: Dodanie więcej podpowiedzi do strony, dodawanie projektu przeniesione do modala, połączenie API rang z frontendem.

Stanisław: Naprawiony system rang, dodanie load balancera do projektu by zoptymalizować go w Kubernetes'ie.

6.25.3 Napotkane problemy

Brak napotkanych problemów.

6.26 SWS i nowe minutniki

6.26.1 Założenia sprintu

Wstępna implementacja nowych minutników, zastosowanie certyfikatu ssl na stronie, przepisanie SWS do LaTeX.

6.26.2 Wykonane zadania

Robert: SWS przepisany do latex.

Paweł: Implementacja minutników flow state, just 5 i custom time.

Stanisław: Dodanie certyfikatu SSL.

6.26.3 Napotkane problemy

Brak napotkanych problemów.

6.27 Strona główna i przerwy

6.27.1 Założenia sprintu

Wstępna implementacja przerw, stworzenie pierwszej wersji strony głównej z użyciem parallax.

6.27.2 Wykonane zadania

Robert: Pierwsza wersja strony głównej w parallax .

Paweł: Rozpoczęcie implementacji przerw.

Stanisław: Dodanie opcji spadku dziennych statystyk.

6.27.3 Napotkane problemy

Brak napotkanych problemów.

6.28 Experymenty z Azure, ulepszanie parallax'a

6.28.1 Założenia sprintu

Stworzenie repozytorium dla serwisu użytkownika, sprawdzenie działania bazy MS SQL na Azure.

6.28.2 Wykonane zadania

Robert: Ulepszanie strony głównej z parallax'em.

Paweł: Przygotowanie teoretyczne do dodania TypeScript'u do projektu.

Stanisław: Wstawienie bazy MS SQL na Azure, stworzenie repozytorium dla serwisu użytkownika.

6.28.3 Napotkane problemy

Brak napotkanych problemów.

6.29 Zbieranie informacji

6.29.1 Założenia sprintu

Poprawa strony głównej, znaleźć informacje dotyczące mikroserwisów do potencjalnej decyzji o zmianie architektury.

6.29.2 Wykonane zadania

Robert: Przystosowanie strony głównej do szukania więcej informacji, zmiana grafik na stronie głównej.

Paweł: Przygotowanie teoretyczne do dodania TypeScript'u do projektu.

Stanisław: Czytanie artykułów na temat mikroserwisów.

6.29.3 Napotkane problemy

Brak napotkanych problemów.

6.30 Podpowiedzi i końcowe prace nad Gamitude 1.0

6.30.1 Założenia sprintu

Zakończenie pracy nad Gamitude 1.0, możliwość wyłączenia podpowiedzi.

6.30.2 Wykonane zadania

Paweł: Dodanie możliwości wyłączania podpowiedzi dla użytkownika.

Robert: Wstawienie strony głównej i przerobienie minutników na wersję 1.0.

6.30.3 Napotkane problemy

Błąd z opóźniającym się minutnikiem oraz dźwiękami nie grającymi będąc na innej karcie. Naprawienie na przyszły sprint.

6.31 Koniec pracy nad Gamitude 1.0, dodanie folderów dla projektów, zmiany w bazie danych

6.31.1 Założenia sprintu

Dodanie folderów dla projektów, Zakończenie pracy nad Gamitude 1.0, utworzenie API dla folderów, przygotowania do połączenia wszystkich serwisów.

6.31.2 Wykonane zadania

Paweł: projekty wybieralne, dodawanie folderów, refactor componentów.

Robert: Zakończenie pracy nad 1.0, problem z dźwiękami i timerem rozwiązany.

Stanisław: Utworzono API dla folderów, zaimplementowano bazę przy użyciu nowego modelu, refactor kodu dla połączenia repozytoriów, statystyki dostosowane do nowych wymagań, przygotowania dla Gamitude Themes, dostęp do bazy dla nowych obiektów.

6.31.3 Napotkane problemy

Brak napotkanych problemów.

6.32 Koniec pracy nad Gamitude 1.0, dodanie folderów dla projektów, zmiany w bazie danych

6.32.1 Założenia sprintu

Dostosowanie projektów do pracy w folderach, strona główna do wersji 2.0, dostosowanie API folderów do potrzeb frontendu.

6.32.2 Wykonane zadania

Paweł: Obsługa dodawania folderów, łączenie pierwszych nowych api, dostosowanie projektów do pracy w folderach.

Robert: Dostosowanie strony głównej do wersji 2.0, napisanie sortowania importów na potrzeby tworzenia przejrzystego kodu.

Stanisław: Ujednolicono API do folderów z potrzebami frontendu, utworzono API dla minutników, stworzono swagger'a, wykonano testy endpointów, przygotowanie wyciągania danych o projektach do statystyk konta użytkownika.

6.32.3 Napotkane problemy

Uwagi do działania strony głównej. Poprawki do następnego sprintu.

6.33 Pełne połączenie frontend'u i backend'u, docker i poprawki strony głównej w wersji 2.0

6.33.1 Założenia sprintu

Połączenie wszystkich API z frontend'em, poprawa routingu w systemie, gromadzenie nowych pomysłów.

6.33.2 Wykonane zadania

Paweł: Połączenie wszystkich API z frontend'em.

Robert: Dostosowanie sortowania importów do potrzeb zespołu, naniesiono poprawki na stronę główną, naprawiono routing w projekcie.

Stanisław: Gromadzenie pomysłów i informacji potrzebnych do stworzenia Bullet Journal'a, stworzenie docker compose, znaleziono pomysły by zapobiec oszustwom, zainicjalizowana baza danych.

6.33.3 Napotkane problemy

Nie obsługiwane projekty energii, nie poprawnie zwracane ikony folderów, brak etykiet przy minutnikach. Naprawa do następnego sprintu.

6.34 Minutniki w 2.0, uspoźnienie wizji

6.34.1 Założenia sprintu

Dostosować minutniki do wersji 2.0, ustalić wspólną wizję działania Bullet Journal'a, dostosować spis treści i jego zawartość do uwag promotora i konsultanta.

6.34.2 Wykonane zadania

Paweł + Robert: dostosowanie spisu treści i jego zawartości do uwag promotora i konsultanta.

Paweł: Dostosowanie minutników do wersji 2.0, dodanie obsługi błędów na stronie, podłączanie poprawionych API.

Robert: Wrzucenie dostosowanej wersji stroni głównej na repozitorium.

Stanisław: Naprawione API dla projektów energii, folderów oraz minutników.

6.34.3 Napotkane problemy

Duplikowanie się importów.

6.35 Pierwsze wersje Bullet Journal'a i Gamitude Themes

6.35.1 Założenia sprintu

Stworzenie widoku Bullet Journal'a na nowym repozytorium, skończenie integracji z API na produkcji, stworzenie modeli i przykładowych kontrolerów dla Bullet Journal'a, wstępna wersja Gamitude Themes. Omówienie konkurencji do dokumentacji.

6.35.2 Wykonane zadania

Paweł + Robert: Dostosowanie spisu treści w kontekście projektu do zaleceń konsultanta.

Paweł: Dostosowanie wyglądu ikon oraz kolorystyk na stronie, dodanie nowej strony na Gamitude Themes, zaprezentowanie drużynie pomysłu na Gamitude Themes, ujednolicenie fontów by były widoczne.

Robert: Mockup na froncie Bullet Journal'a

Stanisław: szkic kontrolerów i modeli dla Bullet Journal'a, skończona integracja API na produkcji.

6.35.3 Napotkane problemy

Przesunięto implementację wstępnej wersji Gamitude Themes na następny sprint.

6.36

6.36.1 Założenia sprintu

Przygotowanie wstępnej wersji Gamitude Themes, dodanie możliwości tworzenia tasków, stron oraz dzienników do Bullet Journal, dodanie repozytoriów i serwisów do Bullet Journal'a, rozmowa na temat zgodności szkiców modeli i kontrolerów.

6.36.2 Wykonane zadania

Paweł:

Robert:

Stanisław:

6.36.3 Napotkane problemy

Rozdział 7

Testy systemu

testy w formie senariuszy

Rozdział 8

Testy w grupach docelowych

Miejsce na wpisanie raportów z testów na grupach użytkowników

Rozdział 9

Prezentacja systemu w działaniu

Rozdział 10

Nakład pracy

Rozpoznanie problemu, Analiza, projektowanie rozwiązania, implementacja, testy.

Rozdział 11

Wkład własny

11.1 Paweł Benkowski

11.2 Robert Deyk

11.3 Stanisław Lutkiewicz

Rozdział 12

Podsumowanie

Poza podsumowaniem projektu i jaki wpływ ma na społeczność, należałoby wykonać raport pod względem komercjalizacji, który należy umieścić w podsumowaniu.

Bibliografia

- [1] C. M. Tony Schwartz, “Manage your energy, not your time,” *Harvard business review*, oct 2007. artykuł na temat energii na podstawie prac z Harvardu - <https://hbr.org/2007/10/manage-your-energy-not-your-time>.
- [2] Wikipedia, “metodologia pomodoro,” https://en.wikipedia.org/wiki/Pomodoro_Technique.
- [3] T. Oppong, “The 90/30 challenge to superhuman productivity,” *Medium*, jun 2017. metodologia 90/30 - <https://medium.com/thrive-global/the-90-30-challenge-to-superhuman-productivity-df6e0f2f39e9>.
- [4] Wikipedia, “grywalizacja,” <https://pl.wikipedia.org/wiki/Grywalizacja>.
- [5] TierZoo, “How humans broke the game,” apr 2019. system rang - <https://www.youtube.com/watch?v=BJzJtm7OfdQ>.
- [6] B. T. Yesterday, “Make your habits stick forever (elastic habits),” *Curiosity*, feb 2020. <https://www.youtube.com/watch?v=L8SehiX7Bjc>.
- [7] J. Kwik, “Hacking flow for faster learning,” oct 2018. metodologia flow state - <https://www.youtube.com/watch?v=wznxlpf0YQ&t=18s>.
- [8] J. Faletto, “The 5-minute hack for beating procrastination used by instagram’s co-founder,” *Curiosity*, jan 2018. metodologia just 5 - <https://curiosity.com/topics/the-5-minute-hack-for-beating-procrastination-used-by-instagrams-co-founder-curiosity/>.
- [9] J. G. Andrew Stelman, *Agile. Przewodnik po zwinnych metodykach programowania*. Helion, sep 2015. Scrum, sprinty i kanban.

Rozdział 13

Załączniki