



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	13
2.3.1	Brain Focus	13
2.3.2	Productivity Challenge Timer	14
2.3.3	Habits	15
2.3.4	Pomodone	16
2.4	Propozycja rozwiązania	17
2.4.1	Projekty	17
2.4.2	Bullet Journal	18
2.4.3	Themes	18
2.5	Kontekst systemu	18
2.6	Cele i odbiorcy systemu	19
2.6.1	Cele systemu	19
2.6.2	Uczestnicy (należy ponownie rozważyć)	20
2.6.3	Grupa docelowa	22
3	Analiza	23
3.1	Wymagania (w trakcie ponownego rozważania)	23
3.1.1	Wymagania ogólne i dziedzinowe	23
3.1.2	Wymagania funkcjonalne	24
3.1.3	Wymagania poza funkcjonalne	30
3.1.4	Wymagania na środowisko docelowe	31

3.2	Wymagania jakościowe i inne	32
3.3	Przypadki użycia oraz diagramy (w trakcie przerabiania i tworzenia nowych)	32
4	Planowanie	35
4.1	Metodyka pracy	35
4.2	Narzędzia	35
4.2.1	Komunikacja	35
4.2.2	Repozytorium oraz CI \CD	36
4.2.3	Cloud	36
4.3	Technologie	36
5	Projektowanie	37
5.1	Architektura projektu	37
5.1.1	Frontend	37
5.1.2	Backend	37
5.1.3	Baza Danych	39
5.1.4	Chmura	39
6	Implementacja rozwiązania	40
7	Historia sprintów	41
7.1	Rozpoznanie dziedzinowe	41
7.1.1	Założenia sprintu	41
7.1.2	Wykonane zadania	41
7.1.3	Napotkane problemy	41
7.2	Wstępna dokumentacja	41
7.2.1	Założenia sprintu	41
7.2.2	Wykonane zadania	42
7.2.3	Napotkane problemy	42
7.3	Use Case + WPP	42
7.3.1	Założenia sprintu	42
7.3.2	Wykonane zadania	42
7.3.3	Napotkane problemy	42
7.4	Dalsza praca przy SWS	42
7.4.1	Założenia sprintu	42
7.4.2	Wykonane zadania	42
7.4.3	Napotkane problemy	43
7.5	Przygotowanie narzędzi do tworzenia systemu	43
7.5.1	Założenia sprintu	43
7.5.2	Wykonane zadania	43
7.5.3	Napotkane problemy	43
7.6	Diagramy funkcjonalności, wymagania bazy danych oraz responsywne rozłożenie strony	43

7.6.1	Założenia sprintu	43
7.6.2	Wykonane zadania	43
7.6.3	Napotkane problemy	43
7.7	Diagramy klas i nawigacja strony	44
7.7.1	Założenia sprintu	44
7.7.2	Wykonane zadania	44
7.7.3	Napotkane problemy	44
7.8	Diagram sekwencji oraz statystyki i energie na stronie	44
7.8.1	Założenia sprintu	44
7.8.2	Wykonane zadania	44
7.8.3	Napotkane problemy	44
7.9	Algorytmy oraz rangi na stronie	44
7.9.1	Założenia sprintu	44
7.9.2	Wykonane zadania	45
7.9.3	Napotkane problemy	45
7.10	Firebase, Heroku, projekty	45
7.10.1	Założenia sprintu	45
7.10.2	Wykonane zadania	45
7.10.3	Napotkane problemy	45
7.11	Firebase autoryzacja, API refactor	45
7.11.1	Założenia sprintu	45
7.11.2	Wykonane zadania	46
7.11.3	Napotkane problemy	46
7.12	Integracja	46
7.12.1	Założenia sprintu	46
7.12.2	Wykonane zadania	46
7.12.3	Napotkane problemy	46
7.13	Prezentacja i demo	46
7.13.1	Założenia sprintu	46
7.13.2	Wykonane zadania	46
7.13.3	Napotkane problemy	47
7.14	Zmiana technologii na backendzie, retrospekcja	47
7.14.1	Założenia sprintu	47
7.14.2	Wykonane zadania	47
7.14.3	Napotkane problemy	47
7.15	Festiwal pomysłów i mockup'y	47
7.15.1	Założenia sprintu	47
7.15.2	Wykonane zadania	47
7.15.3	Napotkane problemy	48
7.16	Festiwal pomysłów ciąg dalszy	48
7.16.1	Założenia sprintu	48
7.16.2	Wykonane zadania	48
7.16.3	Napotkane problemy	48
7.17	Autoryzacja i budowanie bibliografii	48

7.17.1	Założenia sprintu	48
7.17.2	Wykonane zadania	48
7.17.3	Napotkane problemy	48
7.18	Smartwatch'e i MBTI	49
7.18.1	Założenia sprintu	49
7.18.2	Wykonane zadania	49
7.18.3	Napotkane problemy	49
7.19	Odnowienie dokumentacji i bezpieczeństwo	49
7.19.1	Założenia sprintu	49
7.19.2	Wykonane zadania	49
7.19.3	Napotkane problemy	49
7.20	Autoryzacja, dokumentacja	49
7.20.1	Założenia sprintu	49
7.20.2	Wykonane zadania	50
7.20.3	Napotkane problemy	50
7.21	Końcowe prace nad przepisanyimi serwisami	50
7.21.1	Założenia sprintu	50
7.21.2	Wykonane zadania	50
7.21.3	Napotkane problemy	50
7.22	Integracja i LaTeX	50
7.22.1	Założenia sprintu	50
7.22.2	Wykonane zadania	50
7.22.3	Napotkane problemy	51
7.23	Przepisywanie dokumentacji	51
7.23.1	Założenia sprintu	51
7.23.2	Wykonane zadania	51
7.23.3	Napotkane problemy	51
7.24	Podpowiedzi i naprawa błędów	51
7.24.1	Założenia sprintu	51
7.24.2	Wykonane zadania	51
7.24.3	Napotkane problemy	51
7.25	DZW i integracja rang	52
7.25.1	Założenia sprintu	52
7.25.2	Wykonane zadania	52
7.25.3	Napotkane problemy	52
7.26	SWS i nowe minutniki	52
7.26.1	Założenia sprintu	52
7.26.2	Wykonane zadania	52
7.26.3	Napotkane problemy	52
7.27	Strona główna i przerwy	52
7.27.1	Założenia sprintu	52
7.27.2	Wykonane zadania	53
7.27.3	Napotkane problemy	53
7.28	Retrospekcja, ulepszanie parallax'a	53

7.28.1	Założenia sprintu	53
7.28.2	Wykonane zadania	53
7.28.3	Napotkane problemy	53
7.29	Zbieranie informacji	53
7.29.1	Założenia sprintu	53
7.29.2	Wykonane zadania	53
7.29.3	Napotkane problemy	54
7.30	Eksperymenty z Azure, odpowiedzi i końcowe prace nad Gamitude 1.0	54
7.30.1	Założenia sprintu	54
7.30.2	Wykonane zadania	54
7.30.3	Napotkane problemy	54
7.31	Koniec pracy nad Gamitude 1.0, dodanie folderów dla projektów, zmiany w bazie danych	54
7.31.1	Założenia sprintu	54
7.31.2	Wykonane zadania	54
7.31.3	Napotkane problemy	55
7.32	Obsługa folderów	55
7.32.1	Założenia sprintu	55
7.32.2	Wykonane zadania	55
7.32.3	Napotkane problemy	55
7.33	Pełne połączenie frontend'u i backend'u, docker i poprawki strony głównej w wersji 2.0	55
7.33.1	Założenia sprintu	55
7.33.2	Wykonane zadania	55
7.33.3	Napotkane problemy	56
7.34	Minutniki w 2.0, uspołnienie wizji	56
7.34.1	Założenia sprintu	56
7.34.2	Wykonane zadania	56
7.34.3	Napotkane problemy	56
7.35	Pierwsze wersje Bullet Journal'a i Gamitude Themes	56
7.35.1	Założenia sprintu	56
7.35.2	Wykonane zadania	56
7.35.3	Napotkane problemy	57
7.36	Gamitude Themes dla rang, Bullet Journal API	57
7.36.1	Założenia sprintu	57
7.36.2	Wykonane zadania	57
7.36.3	Napotkane problemy	57
7.37	Stoper, kontrola ciągłości sesji, kupowanie rang	57
7.37.1	Założenia sprintu	57
7.37.2	Wykonane zadania	58
7.37.3	Napotkane problemy	58
7.38	Nowa strona główna, ustawienia użytkownika, weryfikacja maili	58
7.38.1	Założenia sprintu	58

7.38.2	Wykonane zadania	58
7.38.3	Napotkane problemy	59
7.39	Edycja i usuwanie folderów oraz czasomierzy, dokumentacja — planowanie, implementacja, wymagania oraz diagramy . .	59
7.39.1	Założenia sprintu	59
7.39.2	Wykonane zadania	59
7.39.3	Napotkane problemy	60
7.40	Integracja, demo, włączanie sesji z poziomu Bullet Journal’a .	60
7.40.1	Założenia sprintu	60
7.40.2	Wykonane zadania	60
7.40.3	Napotkane problemy	60
7.41	61
7.41.1	Założenia sprintu	61
7.41.2	Wykonane zadania	61
7.41.3	Napotkane problemy	61
8	Testy systemu	62
9	Testy w grupach docelowych	63
10	Prezentacja systemu w działaniu	64
11	Nakład pracy	65
12	Wkład własny	66
12.1	Paweł Benkowski	66
12.2	Robert Deyk	66
12.3	Stanisław Lutkiewicz	66
13	Podsumowanie	67
	Bibliografia	68
14	Załączniki	69

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 prace, 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 utrzymanie produktywności bez ubytku w innych sferach życia np. życia rodzinnego.

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 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 nic nie robić”.

Co i dlaczego nie działa?

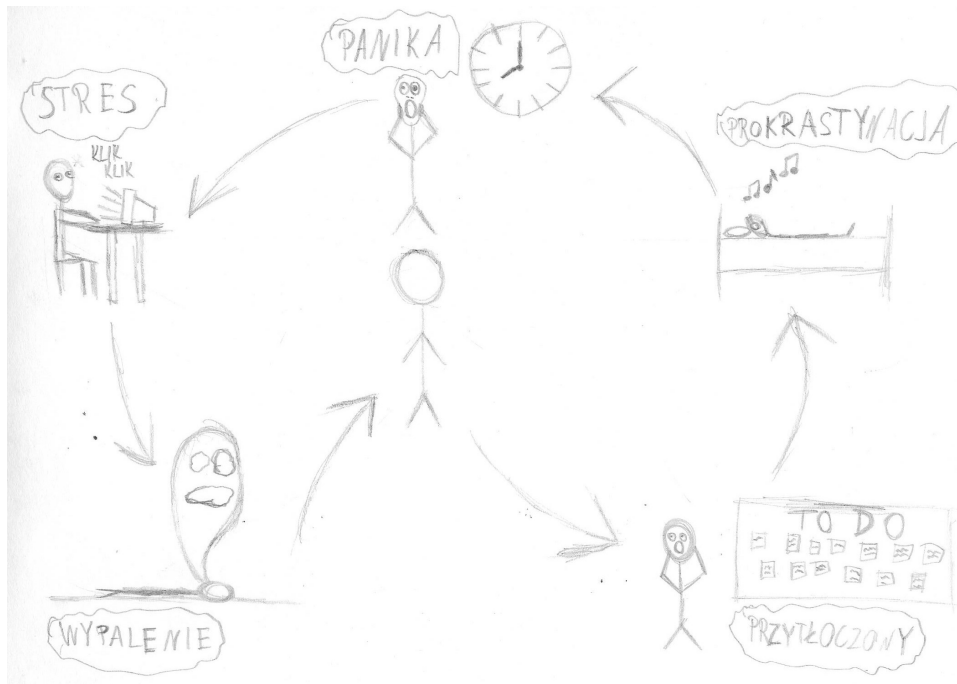
Ludzie operują w kontekście czasu i właśnie go próbują rozplanowywać. Jednakże czas jest surowcem skończonym, którego wszyscy ludzie mają dokładnie tyle samo.

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

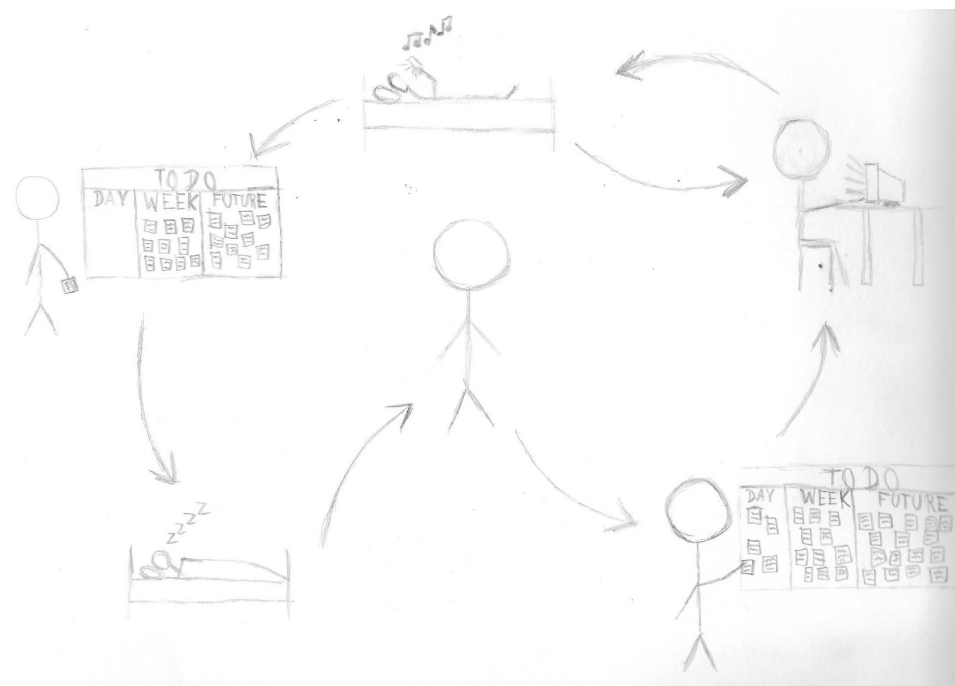
Jednak taki sposób myślenia jest typowy dla ludzi przepracowujących się, ponieważ czują, że czas im ucieka, i muszą za wszelką cenę nie dopuścić do zmarnowania choćby sekundy.

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

2.2 Rich picture



Rysunek 2.1: Użytkownik przed skorzystaniem z systemu.



Rysunek 2.2: Użytkownik korzystający z systemu.

2.3 Konkurencyjne rozwiązania

2.3.1 Brain Focus

O Aplikacji

Brain Focus jest to aplikacja mobilna, pozwalająca na odmierzanie czasu w sesjach z przerwami. Nie jest skomplikowana w użyciu, poprzez dosyć 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 osiągane 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 stworzą, 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 użytkow-

ników 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 miałoby 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

Użytkownik będzie miał do wyboru kilka schematów organizacji pracy, każdy z nich będzie szanował ideę skupienia na energii zamiast na czasie.

Każda czynność będzie dodawała jedną, bądź kilka z 4 statystyk albo odnawiała jedną, bądź kilka z 4 energii.

Statystyki i odpowiadające energie:

Siła dla zadań fizycznych oraz odpowiadająca jej energia ciała.

Twórczość dla zadań artystycznych oraz odpowiadająca jej energia emocji.

Inteligencja dla zadań umysłowych oraz odpowiadająca jej energia umysłu.

Biegłość dla zadań humanistyczno-językowych oraz odpowiadające jej energia duszy.

2.4.1 Projekty

Pierwszym z nich są projekty, czyli organizacja ogólnej tematyki kierunku działań bądź konkretny plan bez wyszczególnionych celów pośrednich.

Użytkownik ma w tym wypadku jakiś cel, na tyle odległy, że wykonanie go nie jest możliwe podczas jednej, nieważne jak długiej, sesji pracy.

Przykładami mogą być tu odpowiednio nauka Python'a oraz kurs Python'a, ponieważ oba zadania wymagają czasu dużo dłuższego od maksymalnej długości ludzkiego skupienie.

Projekty można organizować w foldery np. w folderze programowanie backend'owe mógłbym mieć trzy projekty NodeJs, DenoJs oraz Django.

Projekty mogą również przybrać jedną z 2 form, statystyk bądź energii.

Projekty statystyk to ćwiczenia siłowe, praca umysłowa, praca kreatywna, nauka języka itp.

Projekty energii to rozciąganie, medytacja, techniki oddychania itp.

W przypadku tego typu zadań najlepsze efekty daje systematyczna praca.

Dlatego projekty polegają na:

Stworzeniu projektu i przypisaniu mu nazwy, folderu, statystyk bądź energii boostowanych oraz dominującej, oraz domyślnej metody pracy.

Na przykładzie nauki Python'a mogłoby to być:

nazwa: Nauka Python'a

folder: Programowanie

statystyki boostowane: inteligencja

statystyka dominująca: inteligencja

domyślna metoda pracy: Pomodoro

Nad tak stworzonym projektem możemy teraz pracować w sesjach wybranej metodyki aż do osiągnięcia celu.

2.4.2 Bullet Journal

Drugim z nich jest Bullet Journal, czyli rozplanowywanie zadań do wykonania jako poszczególne zadania w kontekście czasu.

Standardowo dzieli się na zadania na dany dzień, tydzień, miesiąc, nieokreśloną przyszłość oraz po terminie, ale użytkownik może dodać dowolne przedziały czasowe.

Jako że ludzie często lubią oddzielać własne zadania od pracy, dodaliśmy możliwość posiadania wielu dzienników.

2.4.3 Themes

Tutaj możemy wydać zarobione podczas sesji bądź za wykonywanie zadań statystyki, aby nabyć rangi.

Również tutaj zakupimy nowe wyglądy oraz dźwięki dla naszej aplikacji.

Rangi są awatarami użytkownika, początkowo dostaje się jedną z 4 darmowych rang koala, żółwia, węża bądź leniwca.

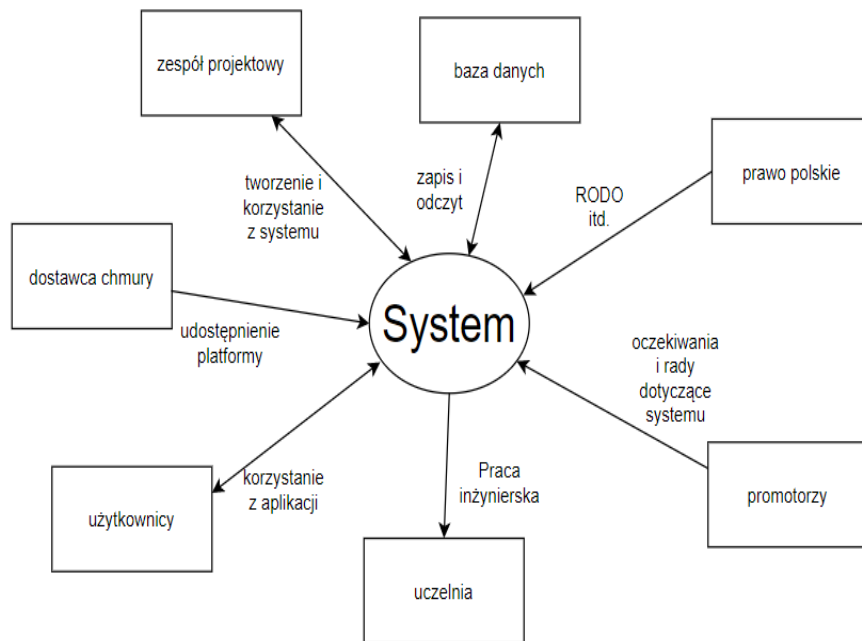
Po poświęceniu pewnego czasu na prace użytkownik zapewne poczuje, że nie jest już takim leniwcem, ale jest już np. wilkiem.

Jest to pewna forma trofeum oraz przyswojenie sobie pewnego obrazu swojego zwierzęcia duchowego.

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 Rhythmn[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.



Rysunek 2.3: Diagram systemu

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 będzie aplikacja internetowa. W przyszłości planujemy rozszerzyć system o aplikację mobilną oraz desktop'ową. Spodziewaną korzyścią będzie wzrost produktywności długoterminowej pośród 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	
Identyfikator	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	
Identyfikator	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	
Identyfikator	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	
Identyfikator	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	
Identyfikator	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	
Identyfikator	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	
Identyfikator	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 przechowywania informacji
Wymagania	
Karta Udziałowca	
Identyfikator	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	
Identyfikator	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	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			
Identyfikator:	W01	Priorytet:	M — must
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 feedbacku w postaci ankiet.		
Udziałowiec	Wydawca, Użytkownik końcowy		
Wymagania powiązane	Brak		

3.1.2 Wymagania funkcjonalne

Karta Wymagania			
Identyfikator:	F01	Priorytet:	M — must
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	e-mail użytkownika, hasło użytkownika, nick		
Warunki początkowe	Użytkownik nie posiada konta		
Warunki końcowe	Użytkownik zarejestrował się i może się zalogować		
Sytuacje wyjątkowe	Błędny e-mail, błędne hasło, użytkownik już posiada konto		
Szczegóły implementacji	Sprint 23.12.2020		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		

Karta Wymagania			
Identyfikator:	F02	Priorytet:	M — must (musi być)
Nazwa	System statystyk		
Opis	Jako użytkownik podczas postępowania w używaniu aplikacji chciałbym widzieć swój postęp w pracy nad projektami w postaci statystyk		
Kryteria akceptacji	Obliczanie statystyk użytkownika na podstawie wykonywanych projektów.		
Dane wejściowe	czas poświęcony na zadanie, statystyki ulepszane przez zadanie, główna statystyka ulepszana przez zadanie		
Warunki początkowe	Użytkownik jest zalogowany, zakończył sesję pracy		
Warunki końcowe	Aktualizacja statystyk		
Sytuacje wyjątkowe	Użytkownik nie wykonał żadnej sesji		
Szczegóły implementacji	Sprint		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	F03, F05		

Karta Wymagania			
Identyfikator:	F03	Priorytet:	M — must (musi być)
Nazwa	System zarządzania energią użytkownika		
Opis	Jako użytkownik chcę, żeby aplikacja śledziła moje zasoby energetyczne, by widzieć swój stan energii.		
Kryteria akceptacji	Zmiana zasobów energii użytkownika przy wykonywaniu konkretnych projektów lub przerw.		
Dane wejściowe	Czas poświęcony na zadanie, statystyki ulepszone przez zadanie, główna statystyka ulepszana przez zadanie		
Warunki początkowe	Użytkownik jest zalogowany, została wykonana sesja pracy.		
Warunki końcowe	Zmniejszenie/zwiększenie statusu energii		
Sytuacje wyjątkowe	Brak energii, nie wykonano żadnej sesji		
Szczegóły implementacji	Sprint		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	F02, F05		

Karta Wymagania			
Identyfikator:	F04	Priorytet:	M — must (musi być)
Nazwa	System zarządzania folderami użytkowników		
Opis	Jako użytkownik chciałbym mieć możliwość dodawania, usuwania i edycji folderów dla projektów.		
Kryteria akceptacji	Użytkownik ma możliwość dodawania, usuwania i edycji folderów.		
Dane wejściowe	Ikona, nazwa folderu		
Warunki początkowe	Użytkownik jest zalogowany.		
Warunki końcowe	Stworzenie folderu, usunięcie folderu, edycja folderu		
Sytuacje wyjątkowe	Nie podano danych		
Szczegóły implementacji	Sprint		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	F05		

Karta Wymagania			
Identyfikator:	F05	Priorytet:	M — must (musi być)
Nazwa	System zarządzania projektami użytkowników		
Opis	Jako użytkownik chciałbym mieć możliwość dodawania, usuwania i śledzenia moich projektów		
Kryteria akceptacji	Użytkownik ma możliwość dodawania, usuwania i śledzenia projektów.		
Dane wejściowe	Nazwa projektu, statystyki ulepszane, dominująca statystyka, folder, metoda pracy		
Warunki początkowe	Użytkownik jest zalogowany.		
Warunki końcowe	Stworzenie projektu, usunięcie projektu, edycja projektu		
Sytuacje wyjątkowe	Nie podano danych		
Szczegóły implementacji	Sprint		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	F02,F04		

Karta Wymagania			
Identyfikator:	F06	Priorytet:	M — must (musi być)
Nazwa	System zarządzania dziennikami		
Opis	Jako użytkownik chcę mieć możliwość zarządzania dziennikami do planowania zadań.		
Kryteria akceptacji	Użytkownik może dodawać, usuwać, edytować swoje dzienniki.		
Dane wejściowe	Nazwa, ikona, połączony projekt		
Warunki początkowe	Użytkownik jest zalogowany		
Warunki końcowe	Dodano, usunięto, edytowano dziennik		
Sytuacje wyjątkowe	Nie podano danych		
Szczegóły implementacji	Sprint		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		

Karta Wymagania			
Identyfikator:	F07	Priorytet:	M — must (musi być)
Nazwa	System zarządzania stronami dziennika		
Opis	Jako użytkownik chcę mieć możliwość zarządzania stronami dziennika.		
Kryteria akceptacji	Użytkownik może dodawać, usuwać, edytować swoje strony w dziennikach.		
Dane wejściowe	Nazwa, ikona, dziennik, początek zakresu dni wyświetlany przez stronę w formie liczby, koniec zakresu		
Warunki początkowe	Użytkownik jest zalogowany, użytkownik ma utworzony dziennik		
Warunki końcowe	Dodano, usunięto, edytowano dziennik		
Sytuacje wyjątkowe	Nie podano danych, podano niepoprawny zakres		
Szczegóły implementacji	Sprint		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		

Karta Wymagania			
Identyfikator:	F08	Priorytet:	M — must (musi być)
Nazwa	System zarządzania zadaniami w dziennikach		
Opis	Jako użytkownik chcę mieć możliwość zarządzania zadaniami w dziennikach.		
Kryteria akceptacji	Użytkownik może dodawać, usuwać, edytować swoje zadania w dziennikach.		
Dane wejściowe	Nazwa, opis, powiązany projekt, termin wykonania, tagi		
Warunki początkowe	Użytkownik jest zalogowany, użytkownik ma utworzony dziennik z przynajmniej jedną stroną		
Warunki końcowe	Dodano, usunięto, edytowano dziennik		
Sytuacje wyjątkowe	Nie podano danych		
Szczegóły implementacji	Sprint		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		

Karta Wymagania			
Identyfikator:	F09	Priorytet:	M — must (musi być)
Nazwa	Zarządzanie czasomierzem		
Opis	Jako użytkownik chcę mieć możliwość zarządzania czasomierzami.		
Kryteria akceptacji	Użytkownik może dodawać, usuwać, edytować swoje czasomierze.		
Dane wejściowe	Nazwa, skrót, czas sesji pracy, czas krótkiej przerwy, czas długiej przerwy, czas nadpracy, interwał przerw krótkich i długich		
Warunki początkowe	Użytkownik jest zalogowany		
Warunki końcowe	Dodano, usunięto, edytowano czasomierz		
Sytuacje wyjątkowe	Nie podano danych		
Szczegóły implementacji	Sprint		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		

Karta Wymagania			
Identyfikator:	F10	Priorytet:	M — must (musi być)
Nazwa	system sesji pracy		
Opis	Jako użytkownik chcę mieć możliwość uruchamiania czasomierza, by rozpocząć sesję pracy bądź przerwę.		
Kryteria akceptacji	Użytkownik może uruchamiać, jak i zatrzymywać czasomierz sesji pracy oraz przerw.		
Dane wejściowe	Wybrany projekt, wybrane zadanie, domyślny czasomierz		
Warunki początkowe	Użytkownik jest zalogowany, został wybrany projekt bądź zadanie.		
Warunki końcowe	Uruchomienie, jak i zatrzymanie sesji pracy bądź przerwy		
Sytuacje wyjątkowe	Nie wybrano projektu, nie wybrano zadania		
Szczegóły implementacji	Sprint		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		

Karta Wymagania			
Identyfikator:	F11	Priorytet:	W – won't
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		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		

Karta Wymagania			
Identyfikator:	F12	Priorytet:	W – won't
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		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		

Karta Wymagania			
Identyfikator:	F13	Priorytet:	W – won't
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		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		

Karta Wymagania			
Identyfikator:	F14	Priorytet:	W – won't
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		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		

Karta Wymagania			
Identyfikator:	F15	Priorytet:	W – won't
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.		
Udziałowiec	Użytkownik końcowy, Zespół projektowy		
Wymagania powiązane	Brak		

3.1.3 Wymagania poza funkcjonalne

Karta Wymagania			
Identyfikator:	NF01	Priorytet:	M – must (musi być)
Nazwa	C#		
Opis	Backend powinien być napisany w C#		
Udziałowiec	Zespół projektowy, Wydawca		
Wymagania powiązane	Brak		

Karta Wymagania			
Identyfikator:	NF02	Priorytet:	M – must (musi być)
Nazwa	React		
Opis	Frontend powinien być napisany, korzystając z biblioteki React.		
Udziałowiec	Zespół projektowy, Wydawca		
Wymagania powiązane	Brak		

Karta Wymagania			
Identyfikator:	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			
Identyfikator:	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		

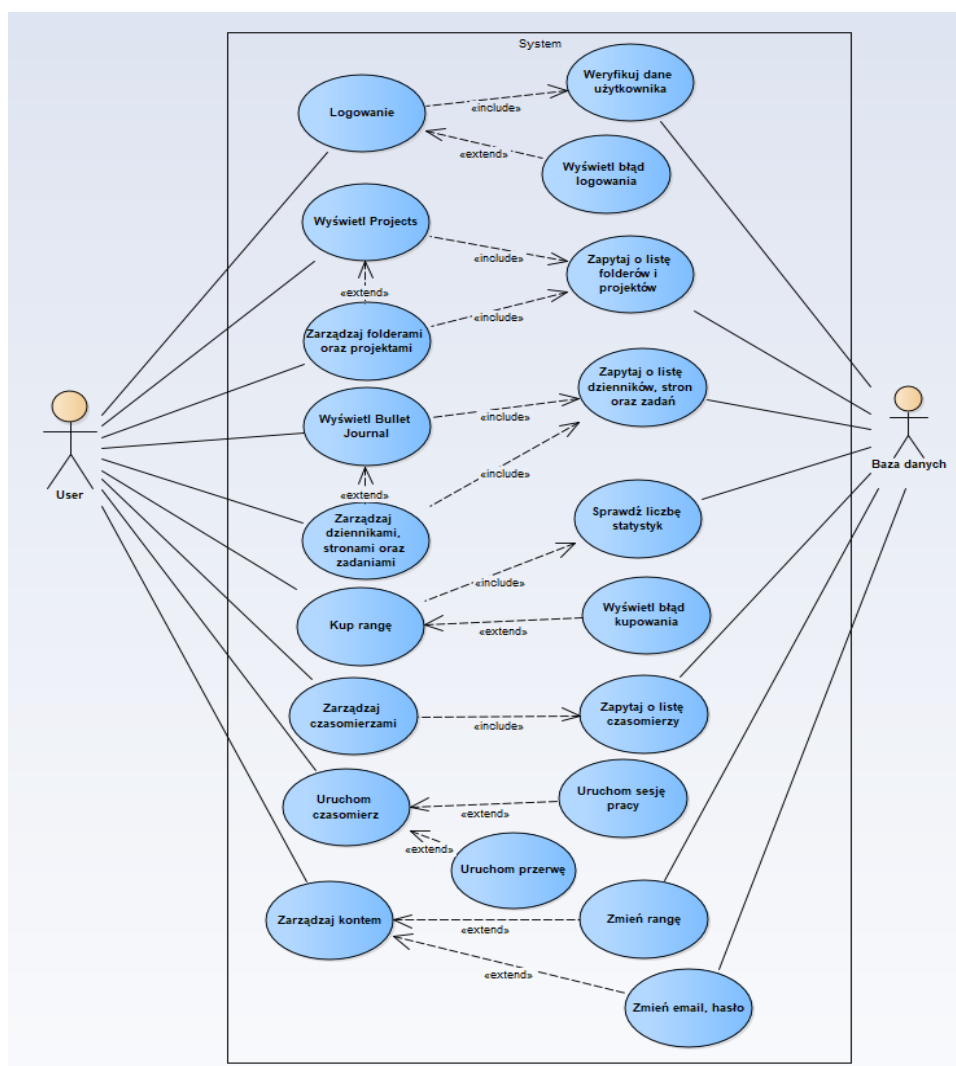
3.1.4 Wymagania na środowisko docelowe

Karta Wymagania			
Identyfikator:	ŚD01	Priorytet:	M – must (musi być)
Nazwa	Kompatybilność z komputerami		
Opis	Produkt końcowy musi być przystosowany do korzystania w przeglądarce na komputerach.		
Kryteria akceptacji	System responsywny testowany na przeglądarkach Google Chrome i Mozilla Firefox		
Udziałowiec	Zespół projektowy, Użytkownik końcowy, Wydawca		
Wymagania powiązane	Brak		

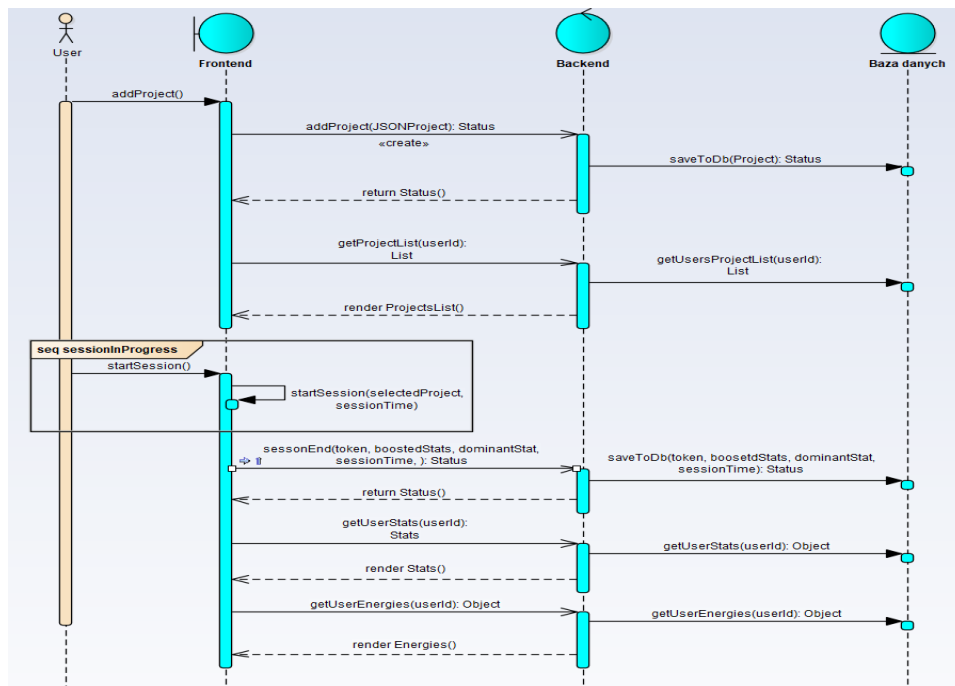
3.2 Wymagania jakościowe i inne

- System szyfrowania danych użytkowników: szyfrowanie haseł użytkowników
- Responsywny system przystosowany dla przeglądarek 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

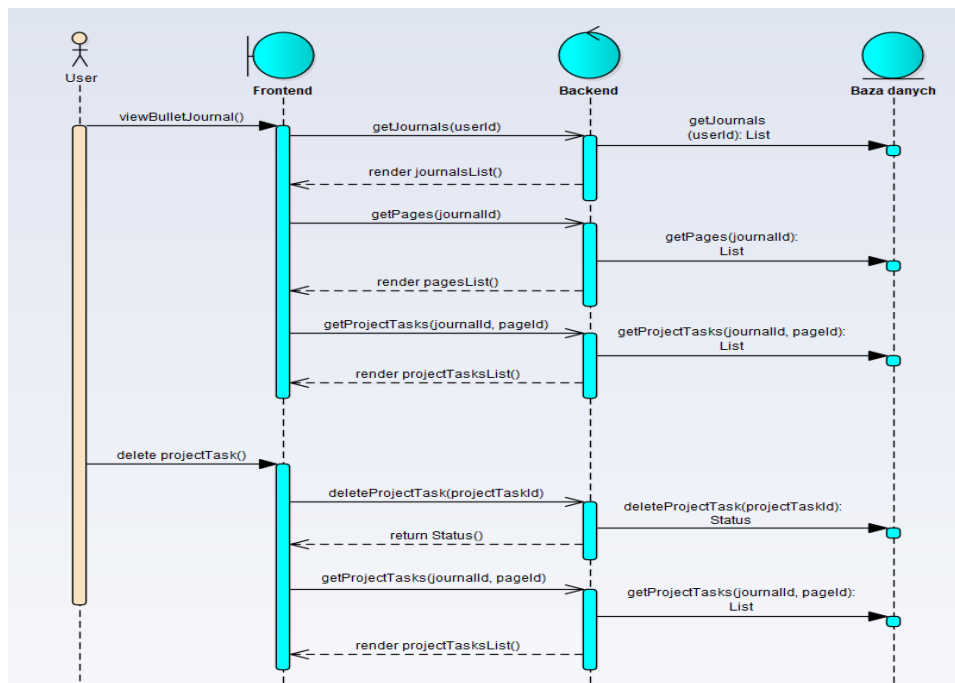
3.3 Przypadki użycia oraz diagramy (w trakcie przetwarzania i tworzenia nowych)



Rysunek 3.1: Use case dla całego systemu



Rysunek 3.2: Diagram sekwencji dla dodawania projektu oraz włączania sesji



Rysunek 3.3: Diagram sekwencji dla załadowania strony Bullet Journal'a oraz usuwania zadania

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]. Sprints 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 tę metodykę ze względu na fakt, że nasz system jest dość nowatorski w swojej kategorii, przez co wymaga stałej weryfikacji przez użytkowników.

4.2 Narzędzia

4.2.1 Komunikacja

Do komunikacji w zespole użyliśmy darmowej wersji Slack'a. Jediną wadą był brak rozmów wideo.

Aby zapełnić tę lukę, stworzyliśmy sobie darmowy pokój na stronie Whereby, gdzie w zespole nieprzekraczającym 4 osób mogliśmy się swobodnie porozumiewać.

Wybór padł na Slack'a ze względu na możliwość zintegrowania go z narzędziami do CI/CD or repozytoriów.

Pozwala to na posiadanie informacji o wszystkich aktualnych działaniach całego zespołu w jednym miejscu.

Tablice z zadaniami posiadamy na platformie Azure DevOps, która również pozwala na tworzenie sprintów.

4.2.2 Repozytorium oraz CI \CD

Repozytorium naszego projektu powstało na Github’ie. Zintegrowaliśmy go z platformą Azure DevOps, która pozwala nam na zbudowanie każdego komponentu projektu od razu po aktualizacji naszego kodu w repozytorium. Jeżeli projekt udało się zbudować na głównej gałęzi, nasze świeżo zbudowane kontenery wysyłane są do chmury.

4.2.3 Cloud

W chmurze postawiliśmy na konteneryzację naszych projektów, która pozwala na łatwe skalowanie oraz pozostanie niezależnym od maszyny hostującej. Google Cloud został wybrany na dostawcę rozwiązań chmurowych. Postanowiliśmy wykorzystać technologie Kubernetes do skalowania oraz zarządzania naszymi kontenerami. Pozwala ona pozostać w miarę możliwości niezależnym od dostawcy chmurowego (decyzja ta też została podjęta ze względu na wcześniejszą architekturę mikro serwisową).

4.3 Technologie

- React/Redux/MUI/TypeScript — frontend
- C#/.NET Core — backend
- MongoDB — baza danych
- Docker/Kubernetes — deployment

Rozdział 5

Projektowanie

5.1 Architektura projektu

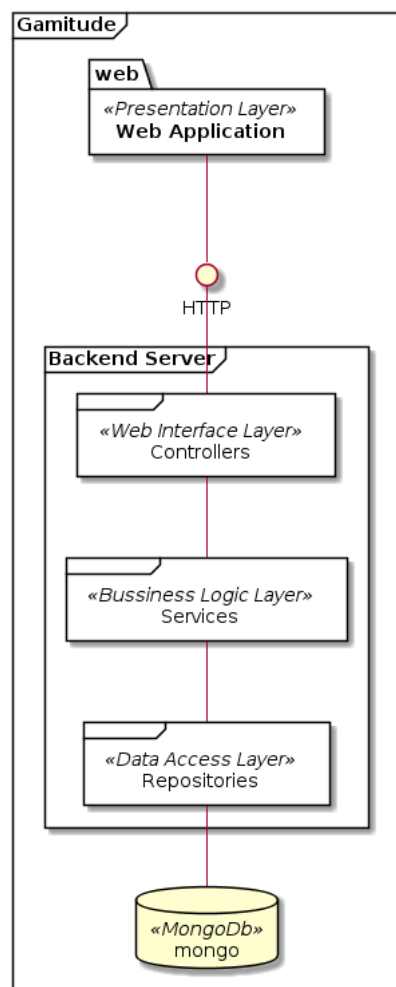
Aplikacja webowa Gamitude została wykonana w architekturze wielowarstwowej typu klient-serwer. Interfejs użytkownika został wykonany w technologii SPA (single page application). Naturalnym więc było podzielenie całej aplikacji na 2 repozytoria Frontentu i Backendu. Komunikacja pomiędzy nimi opiera się na protokole HTTP/HTTPS.

5.1.1 Frontend

Opis Pawła podział na komponenty? Co daje SPA wczytanie całej aplikacji na początku brak przeładowania całej strony pobieranie informacji podczas interakcji z użytkownikiem.

5.1.2 Backend

Aplikacja serwerowa została podzielona na warstwy TODO referencja do rys. W warstwie dostępu do bazy danych został wykorzystany wzorzec projektowy repozytorium lub repozytorium serwis. Pozwala on na zamknięcie logiki potrzebnej do pozyskania danych z bazy danych. Dzięki temu nasza logika biznesowa jest oddzielona od infrastruktury. Każdy obiekt w bazie danych posiada swoje repozytorium. Każdy serwis może korzystać z wielu repozytoriów co pozwala na tworzenie bardziej złożonych struktur. Wykorzystany został także wzorzec projektowy wstrzykiwania zależności. Dzięki temu w każdym miejscu projektu możemy łatwo uzyskać dostęp do potrzebnych nam elementów np. Konfiguracji. Także pozwala nam on na łatwą podmianę danego komponentu aplikacji poprzez zarejestrowanie innej implementacji danego interfejsu.



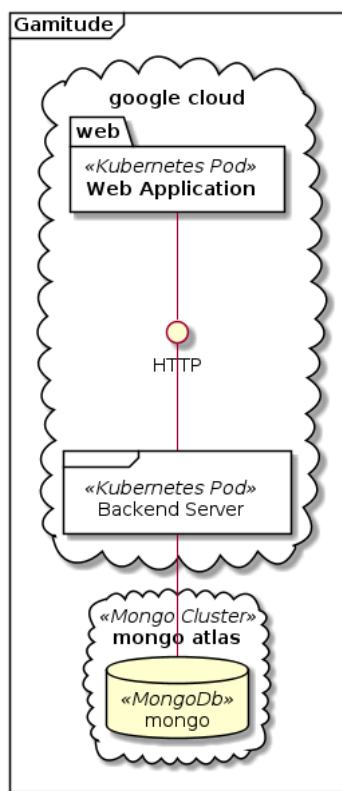
Rysunek 5.1: Uproszczona architektura

5.1.3 Baza Danych

Wybraliśmy baze danych NoSQL do naszego projektu. Ze względu na koszty finansowe oraz łatwiejszą skalowalność kosztem braku transakcji.

5.1.4 Chmura

Wdrożyliśmy naszą aplikację do chmury za pośrednictwem aplikacji Kubernetes pozwoliło to nam pozostać w miarę możliwości niezależnym od dostawcy chmurowego. W dniu dzisiejszym korzystamy z chmury google. Wdrożyliśmy technikę CI/CD która pozwala na zbudowanie oraz aktualizacje rozwiązania przy każdej aktualizacji głównej gałęzi naszych repozytoriów.

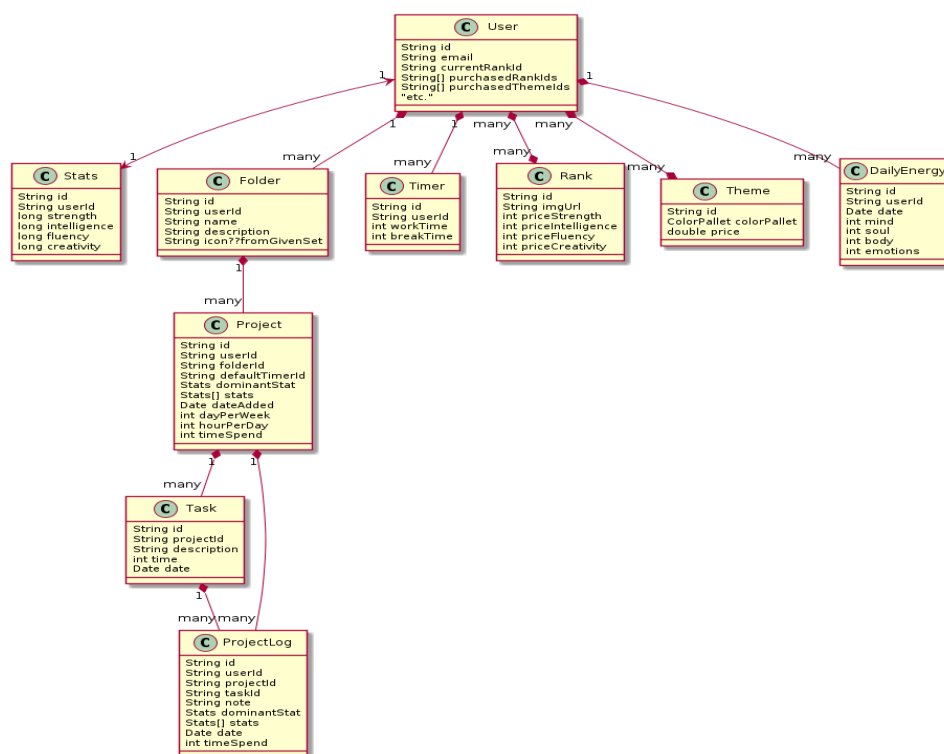


Rysunek 5.2: Architektura chmury

Rozdział 6

Implementacja rozwiązania

Szczegółowy opis końcowego rozwiązania, do którego będziemy się odnosić w następnym rozdziale.



Rysunek 6.1: Schemat bazy danych

Rozdział 7

Historia sprintów

7.1 Rozpoznanie dziedzinowe

7.1.1 Założenia sprintu

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

7.1.2 Wykonane zadania

Paweł: Używanie metodologii Pomodoro i Ultradian rhythm 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).

7.1.3 Napotkane problemy

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

7.2 Wstępna dokumentacja

7.2.1 Założenia sprintu

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

7.2.2 Wykonane zadania

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

Robert: Wytworzenie wstępnej wersji dokumentu SWS.

Stanisław: Rozpoznanie dotyczące kontroli wersji w EA. Rozpoznanie architektury mikro serwisów oraz projektowania jej.

7.2.3 Napotkane problemy

Kontrola wersji w EA dostępne tylko po wykupieniu licencji.

7.3 Use Case + WPP

7.3.1 Założenia sprintu

Przygotowanie pierwszych diagramów Use Case. Wytworzenie dokumentu WPP.

7.3.2 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.

7.3.3 Napotkane problemy

Brak napotkanych problemów.

7.4 Dalsza praca przy SWS

7.4.1 Założenia sprintu

Doprecyzowanie wymagań systemowych oraz niefunkcjonalnych w dokumencie SWS. Podjąć decyzję odnośnie do wyboru architektury systemu.

7.4.2 Wykonane zadania

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

Paweł: Przerabianie kursu React.

7.4.3 Napotkane problemy

Brak napotkanych problemów.

7.5 Przygotowanie narzędzi do tworzenia systemu

7.5.1 Założenia sprintu

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

7.5.2 Wykonane zadania

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

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

7.5.3 Napotkane problemy

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

7.6 Diagramy funkcjonalności, wymagania bazy danych oraz responsywne rozłożenie strony

7.6.1 Założenia sprintu

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

7.6.2 Wykonane zadania

Zespół: Tworzenie wykresu Gantt'a.

Paweł: Utworzenie responsywnego rozłożenia strony.

Robert: Stworzenie diagramu funkcjonalności.

Stanisław: Stworzenie diagramu funkcjonalności oraz bazy danych.

7.6.3 Napotkane problemy

Znalezienie darmowego narzędzia dla wykresu Gantt'a.

7.7 Diagramy klas i nawigacja strony

7.7.1 Założenia sprintu

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

7.7.2 Wykonane zadania

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

Paweł: Implementacja nawigacji strony.

Stanisław: Stworzenie routingu dla mikro serwisu projektów.

7.7.3 Napotkane problemy

Robert nie przygotował routingu.

7.8 Diagram sekwencji oraz statystyki i energie na stronie

7.8.1 Założenia sprintu

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

7.8.2 Wykonane zadania

Robert i Stanisław: Utworzenie diagramu sekwencji.

Paweł: Implementacja widoków statystyk i energii.

Robert: Utworzono routing w mikro serwisie rang.

Stanisław: Modyfikacja modelu projektów w bazie danych.

7.8.3 Napotkane problemy

Brak napotkanych problemów.

7.9 Algorytmy oraz rangi na stronie

7.9.1 Założenia sprintu

Stworzenie algorytmów odpowiedzialnych za przyznawanie rangi oraz za wzrost lub spadek energii. Zaimplementowanie struktury bazy danych zgodnej z do-

kumentacją oraz utworzenie rang na frontend'ie.

7.9.2 Wykonane zadania

Paweł: Implementacja widoków rang.

Robert: Algorytm przyznawania rang i statystyk.

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

7.9.3 Napotkane problemy

Brak napotkanych problemów.

7.10 Firebase, Heroku, projekty

7.10.1 Założenia sprintu

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

7.10.2 Wykonane zadania

Robert i Stanisław: Stworzenie schematów modeli bazy danych dla mikro serwisów.

Paweł: Implementacja widoków projektów. Refaktoryzacja dotychczasowego kodu.

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

7.10.3 Napotkane problemy

Modele rang i workflow niezgodne z założeniami, do poprawy w następnym sprincie.

7.11 Firebase autoryzacja, API refactor

7.11.1 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.

7.11.2 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. Dodanie API do przyjmowania poświęconego czasu.

7.11.3 Napotkane problemy

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

7.12 Integracja

7.12.1 Założenia sprintu

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

7.12.2 Wykonane zadania

Robert i Stanisław: Dostosowanie algorytmów do wymagań.

Paweł: Integracja z API.

Stanisław: Dodanie obsługi błędów związanych z bazą danych.

7.12.3 Napotkane problemy

Problem z integracją frontend'u z API.

7.13 Prezentacja i demo

7.13.1 Założenia sprintu

Utworzenie prezentacji projektu oraz pokazowego dema działania aplikacji.

7.13.2 Wykonane zadania

Robert: Wykonanie pokazowego dema.

Paweł: Stworzenie prezentacji.

7.13.3 Napotkane problemy

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

7.14 Zmiana technologii na backendzie, retrospekcja

7.14.1 Założenia sprintu

Przyjrzenie się powstałemu już systemowi i wyznaczenie kierunku dalszych prac.

7.14.2 Wykonane zadania

Zespół: Ustalenie zmiany technologii backend'owej 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 NodeJs na C#, praca nad serwisem autoryzacji użytkownika, przystosowanie chmury do pracy z technologią .NET, wyeksportowanie obecną bazę danych do pliku lokalnego w razie zmiany bazy.

7.14.3 Napotkane problemy

Brak napotkanych problemów.

7.15 Festiwal pomysłów i mockup'y

7.15.1 Założenia sprintu

Szukanie nowych pomysłów na funkcjonalności. Stworzenie mockup'ów dla strony głównej oraz Bullet Journal'a.

7.15.2 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: Przeniesienie bazy danych z Heroku do chmury mongo atlas. Pomysł na zbieranie informacji o użytkownikach dla późniejszej sprzedaży.

7.15.3 Napotkane problemy

Brak napotkanych problemów.

7.16 Festiwal pomysłów ciąg dalszy

7.16.1 Założenia sprintu

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

7.16.2 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.

7.16.3 Napotkane problemy

Brak napotkanych problemów.

7.17 Autoryzacja i budowanie bibliografii

7.17.1 Założenia sprintu

Połączenie frontend'u z serwisem logowania i rejestracji. Zbudowanie podstawowej bibliografii.

7.17.2 Wykonane zadania

Paweł i Stanisław: Połączenie frontend'u z przepisany serwis.

Robert: Zebranie danych dotyczących źródeł informacji do bibliografii.

Stanisław: Początek przenoszenia projektów z Heroku do chmury Google.

7.17.3 Napotkane problemy

Brak napotkanych problemów.

7.18 Smartwatch'e i MBTI

7.18.1 Założenia sprintu

Szukanie informacji na temat wykorzystania smartwatch'y w aplikacji. Rozpoznanie w zakresie różnych typów użytkowników systemu.

7.18.2 Wykonane zadania

Paweł i Robert: Wykonanie badania MBTI.

Stanisław: Przenoszenie kolejnych mikro serwisów do chmury Google'a. Stworzenie mikro serwisu od autoryzacji. Pomysł na integrację aplikacji z Google Calendar.

7.18.3 Napotkane problemy

Brak napotkanych problemów.

7.19 Odnowienie dokumentacji i bezpieczeństwo

7.19.1 Założenia sprintu

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

7.19.2 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 oraz równoważenia obciążenia

7.19.3 Napotkane problemy

Koszt load balancer'a w chmurze Google'a przewyższa koszt jednej maszyny wirtualnej.

7.20 Autoryzacja, dokumentacja

7.20.1 Założenia sprintu

Poprawa bibliografii, zebranie historii sprintów, poprawa wyglądu systemu autoryzacji użytkownika po stronie frontend'u.

7.20.2 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 frontend'u.

Stanisław: Research na temat architektury systemu.

7.20.3 Napotkane problemy

Brak napotkanych problemów.

7.21 Końcowe prace nad przepisanyimi serwisami

7.21.1 Założenia sprintu

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

7.21.2 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.

7.21.3 Napotkane problemy

Brak napotkanych problemów.

7.22 Integracja i LaTeX

7.22.1 Założenia sprintu

Integracja zakończonych mikro serwisów, przygotowania do przepisania dokumentacji na latex.

7.22.2 Wykonane zadania

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

Paweł: Obsługa błędów dla projektów, integracja serwisu statystyk i energii.
Stanisław: skończenie serwisu statystyk, czytanie o mikro serwisach, prace nad cron'em.

7.22.3 Napotkane problemy

Brak napotkanych problemów.

7.23 Przepisywanie dokumentacji

7.23.1 Założenia sprintu

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

7.23.2 Wykonane zadania

Robert: przerabianie dokumentów na latex.

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

Stanisław: przerabianie kursu o identity server 4.

7.23.3 Napotkane problemy

Brak napotkanych problemów.

7.24 Podpowiedzi i naprawa błędów

7.24.1 Założenia sprintu

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

7.24.2 Wykonane zadania

Robert: dodanie karty projektów w latex.

Paweł: Stworzenie podpowiedzi dla użytkowników na stronie.

Stanisław: Poprawa błędów w obliczaniu statystyk.

7.24.3 Napotkane problemy

Brak napotkanych problemów.

7.25 DZW i integracja rang

7.25.1 Założenia sprintu

Zintegrowanie przepisanego systemu rang, przepisanie dokumentu DZW na LaTeX.

7.25.2 Wykonane zadania

Robert: Dodanie DZW do latex dokumentu, przerobienie bibliografii.

Paweł: Dodanie więcej podpowiedzi do strony, dodawanie projektu przeniesione do modal'a, połączenie API rang z frontend'em.

Stanisław: Naprawiony system rang, dodanie load balancer'a do projektu jako swój własny serwis w Kubernetes'ie. Brak idealnego rozwiązania.

7.25.3 Napotkane problemy

Brak napotkanych problemów.

7.26 SWS i nowe minutniki

7.26.1 Założenia sprintu

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

7.26.2 Wykonane zadania

Robert: SWS przepisany do latex.

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

Stanisław: Dodanie certyfikatu SSL.

7.26.3 Napotkane problemy

Brak napotkanych problemów.

7.27 Strona główna i przerwy

7.27.1 Założenia sprintu

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

7.27.2 Wykonane zadania

Robert: Pierwsza wersja strony głównej w parallax.

Paweł: Rozpoczęcie implementacji przerw.

Stanisław: Dodanie opcji spadku dziennych statystyk.

7.27.3 Napotkane problemy

Brak napotkanych problemów.

7.28 Retrospekcja, ulepszanie parallax'a

7.28.1 Założenia sprintu

Stworzenie repopozytoryum dla serwisu użytkownika.

7.28.2 Wykonane zadania

Robert: Ulepszanie strony głównej z parallax'em.

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

Stanisław: Rozważania na temat zmiany bazy danych, stworzenie repopozytoryum dla serwisu użytkownika.

7.28.3 Napotkane problemy

Brak napotkanych problemów.

7.29 Zbieranie informacji

7.29.1 Założenia sprintu

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

7.29.2 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 mikro serwisów.

7.29.3 Napotkane problemy

Brak napotkanych problemów.

7.30 Eksperymenty z Azure, podpowiedzi i końcowe prace nad Gamitude 1.0

7.30.1 Założenia sprintu

Zakończenie pracy nad Gamitude 1.0, możliwość wyłączenia podpowiedzi, sprawdzenie działania bazy MS SQL na Azure.

7.30.2 Wykonane zadania

Paweł: Dodanie możliwości wyłączenia podpowiedzi dla użytkownika.

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

Stanisław: Stworzenie bazy MS SQL na Azure. Zmiana architektury na Monolit

7.30.3 Napotkane problemy

Błąd z opóźniającym się minutnikiem, dźwiękami niegrającymi w przypadku gdy użytkownik jest na innej karcie przeglądarki. Naprawienie na przyszły sprint.

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

7.31.1 Założenia sprintu

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

7.31.2 Wykonane zadania

Paweł: projekty wybieralne, dodawanie folderów, refactor component'ów.

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.

7.31.3 Napotkane problemy

Brak napotkanych problemów.

7.32 Obsługa folderów

7.32.1 Założenia sprintu

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

7.32.2 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 frontend'u, utworzono API dla minutników, stworzono swagger'a, wykonano testy endpointów, przygotowanie wyciągania danych o projektach do statystyk konta użytkownika.

7.32.3 Napotkane problemy

Uwagi do działania strony głównej. Poprawki do następnego sprintu.

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

7.33.1 Założenia sprintu

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

7.33.2 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.

7.33.3 Napotkane problemy

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

7.34 Minutniki w 2.0, uspoźnienie wizji

7.34.1 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.

7.34.2 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 repozytorium.

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

7.34.3 Napotkane problemy

Duplikowanie się importów.

7.35 Pierwsze wersje Bullet Journal'a i Gamitude Themes

7.35.1 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.

7.35.2 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.

7.35.3 Napotkane problemy

Przesunięto implementację wstępnej wersji Gamitude Themes na następny sprint.

7.36 Gamitude Themes dla rang, Bullet Journal API

7.36.1 Założenia sprintu

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

7.36.2 Wykonane zadania

Paweł: Utworzenie wstępnej wersji Gamitude Themes z zakupem rang.

Robert: Możliwość dodawania Journal'i oraz page'y.

Stanisław: Utworzenie repozytoriów i serwisów dla Bullet Journal'a, uaktualnienie historii sprintów o brakujący DevOps, skończenie podstawowego API dla Bullet Journal'a.

7.36.3 Napotkane problemy

Problemy z Redux'em, dokończenie tworzenia task'ów w następnym sprincie. Źle ustalony priorytet zadania (Gamitude Themes).

7.37 Stoper, kontrola ciągłości sesji, kupowanie rang

7.37.1 Założenia sprintu

Dokończenie pracy nad tworzeniem zadań w Bullet Journal, możliwość skończenia zadań oraz łączenie zadań, oraz dzienników z projektem. Dostosowanie czasomierza, możliwość zakupu rang oraz ich filtrowania. Podpięcie API do Gamitude Themes, ulepszenie czasomierza, mniejsza wersja panelu kontrolnego.

7.37.2 Wykonane zadania

Paweł: Dodanie stopera, powiadomienia przy przedwczesnym przerwaniu sesji, dodatkowe informacje dotyczące czasomierza w postaci długości krótkiej przerwy, długiej przerwy, automatyczne blokowanie interakcji z aplikacją podczas przerwy, automatyczna zmiana czasomierza na przerwę. Drobne poprawki do projektów, czasomierza oraz wyglądu aplikacji. Zmieniony formularz tworzenia czasomierza.

Robert: Utworzenie Rich Picture, skończone prace nad tworzeniem zadań do Bullet Journal'a.

Stanisław: dodanie możliwości kupowania rang wraz z transakcjami, dostosowanie czasomierza, dostosowanie Bullet Journal'a.

7.37.3 Napotkane problemy

Łączenie zadań oraz dzienników z projektem wraz z ujednolicaniem dodawania i edycji zadań przeniesiona na następny sprint. Złe oszacowanie ilości czasu poświęconego na mniejszą wersję panelu kontrolnego.

7.38 Nowa strona główna, ustawienia użytkownika, weryfikacja maili

7.38.1 Założenia sprintu

Dodanie do dokumentacji architektury systemu, dodanie wysyłanie e-mail'i przy rejestracji konta, zakończenie pracy nad Bullet Journal'em, aktualizacja wymagań w dokumentacji, odnowienie strony głównej, podstawowe ustawienia użytkownika.

7.38.2 Wykonane zadania

Paweł: Nowa wersja strony głównej, edycja hasła oraz szczegółów konta, naprawa stopera, zapytania przy próbie opuszczenia strony podczas sesji, optymalizacje dostępnościowe oraz wydajnościowe, przełączanie widoczności hasła.

Robert: Refaktoryzacja całego Bullet Journal'a, możliwość przypisywania zadań oraz dzienników do projektów

Stanisław: Weryfikacja maili w stanie zaawansowanym — konfiguracja Send Grid, stworzenie template'u maila, naprawienie stronicowania dla rang. logowanie dostosowane do systemu potwierdzania, poprawki do Bullet Journal'a.

7.38.3 Napotkane problemy

Problem z kodem z GitHub Education w Send Grid.

Niewykonanie 4.1, 4.2, 4.3 - planowanie oraz niedodanie architektury do prezentacji.

Problem z przełączaniem się po Bullet Journal'u oraz niewykonanie możliwości wybierania zadań dla czasomierza.

Niewykonanie 3.1, 3.2, 3.3 - uaktualnienie wymagań, diagramów oraz przypadków użycia.

Niewykonanie Dialog z tutorial'em — nagranie dema działania aplikacji.

Niewykonanie Rich Picture — nie wstawiono nowych wersji Rich Picture do dokumentacji.

7.39 Edycja i usuwanie folderów oraz czasomierzy, dokumentacja — planowanie, implementacja, wymagania oraz diagramy

7.39.1 Założenia sprintu

Dokończenie weryfikacji maili wraz z integracją z frontend'em.

Wykonanie dokumentacji 4.1, 4.2, 4.3 - planowanie oraz dodanie architektury do prezentacji.

Dodanie domyślnych folderów oraz projektów dla zapoznania użytkownika z systemem.

Pełna integracja czasomierzy, folderów oraz rang — możliwość edycji, ustawiania oraz usuwania.

Mini wariant dla panelu kontrolnego.

Wykonanie dokumentacji 3.1, 3.2, 3.3 - uaktualnienie wymagań, diagramów oraz przypadków użycia.

Nagranie dema działania aplikacji.

Wstawienie nowych wersji Rich Picture do dokumentacji.

Rozwiązanie problemu z przełączaniem się po Bullet Journal'u oraz dodanie możliwości wybierania zadań dla czasomierza.

7.39.2 Wykonane zadania

Paweł + Stanisław: Weryfikacji maili, bez możliwości zmiany maila. Dodano na stronę Google Analytics.

Robert: Wykonanie dokumentacji 3.1, 3.2, 3.3 - uaktualnienie wymagań, diagramów oraz przypadków użycia. Wstawienie nowych wersji Rich Picture do dokumentacji.

Stanisław: Wykonanie dokumentacji 4.1, 4.2, 4.3 - planowanie oraz dodanie

architektury do prezentacji.

7.39.3 Napotkane problemy

Niewykonanie Pełna integracja czasomierzy, folderów oraz rang — możliwość edycji, ustawiania oraz usuwania.

Niewykonanie Nagranie demo działania aplikacji.

Niewykonanie Mini wariant dla panelu kontrolnego.

Niewykonanie Rozwiązanie problemu z przełączaniem się po Bullet Journal'u oraz dodanie możliwości wybierania zadań dla czasomierza.

7.40 Integracja, demo, włączanie sesji z poziomu Bullet Journal'a

7.40.1 Założenia sprintu

Pełna integracja czasomierzy, folderów oraz rang — możliwość edycji, ustawiania oraz usuwania.

Nagranie demo działania aplikacji.

Dorobić więcej diagramów sekwencji.

Rozwiązanie problemu z przełączaniem się po Bullet Journal'u oraz dodanie możliwości wybierania zadań dla czasomierza.

Dokumentacja 5.1.2, 5.1.3, 5.1.4 - Projektowanie i architektura systemu.

Wzór dla maila wysyłanego przy rejestracji.

Zmiana ProjectLogs, by zapewnić spójność.

7.40.2 Wykonane zadania

Paweł: Pełna integracja czasomierzy, folderów oraz rang — możliwość edycji, ustawiania oraz usuwania, integracja weryfikacji maili.

Robert: Dorobiono więcej diagramów sekwencji, Rozwiązanie problemu z przełączaniem się po Bullet Journal'u.

Stanisław: Wzór dla maila wysyłanego przy rejestracji, zmiana ProjectLogs, by zapewnić spójność, naprawa błędów w Gamitude Themes, API do sprawdzania integralności danych użytkownika.

7.40.3 Napotkane problemy

Niewykonanie Nagranie demo działania aplikacji.

Problem z wybieraniem zadania/projektu dla czasomierza.

Niewykonanie Dokumentacja 5.1.2, 5.1.3, 5.1.4 - Projektowanie i architektura systemu.

7.41

7.41.1 Założenia sprintu

Dodanie możliwości wybierania zadań dla czasomierza.

Integracja Bullet Journal'a z pozostałą częścią aplikacji.

Nagranie dema działania aplikacji.

Dokumentacja 5.1.2, 5.1.3, 5.1.4 - Projektowanie i architektura systemu.

Tworzenie testów API.

Dostosowanie aplikacji pod nowe ProjectLogs API.

7.41.2 Wykonane zadania

Paweł:

Robert:

Stanisław:

7.41.3 Napotkane problemy

Rozdział 8

Testy systemu

Testy w formie scenariuszy.

Rozdział 9

Testy w grupach docelowych

Miejsce na wpisanie raportów z testów na grupach użytkowników

Rozdział 10

Prezentacja systemu w działaniu

Rozdział 11

Nakład pracy

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

Rozdział 12

Wkład własny

12.1 Paweł Benkowski

12.2 Robert Deyk

12.3 Stanisław Lutkiewicz

Rozdział 13

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 14

Załączniki