

Imię i nazwisko studenta: Szymon Hejmanowski

Nr albumu: 184487

Poziom kształcenia: Studia pierwszego stopnia

Forma studiów: stacjonarne

Kierunek studiów: Informatyka

Profil: Teleinformatyka

Imię i nazwisko studenta: Patryk Gołembiewski

Nr albumu: 184587

Poziom kształcenia: Studia pierwszego stopnia

Forma studiów: stacjonarne

Kierunek studiów: Informatyka

Profil: Teleinformatyka

PROJEKT DYPLOMOWY INŻYNIERSKI

Tytuł projektu w języku polskim: System wspomagający wzmacnianie odporności psychicznej

Tytuł projektu w języku angielskim: A system for strengthening mental toughness

Opiekun pracy: mgr inż. Krystyna Dziubich

STRESZCZENIE

Niniejszy projekt inżynierski został wykonany w celu stworzenia aplikacji mobilnej wzmacniającej odporność psychiczną. Zmotywowany jest on chęcią pozytywnego wpływu na zdrowie psychiczne użytkowników. Zapoznanie się z tematyką odporności psychicznej oraz przegląd istniejących rozwiązań, pozwoliły zgromadzić podstawową wiedzę potrzebną do rozpoczęcia tworzenia wizji aplikacji. Następnie, zastosowane metody kreatywności, takie jak mapa empatii lub scenariusze „as is / to be”, oraz badania ankietowe pomogły w procesie wymyślenia funkcjonalności systemu. Po stworzeniu wizji aplikacji nastąpił czas opracowania zagadnień technicznych, takich jak architektura systemu oraz struktura bazy danych. Po fazie implementacji przeprowadzone zostało studium przypadku zastosowania systemu z udziałem grupy wybranych ochotników. Potwierdziło ono, w ocenie użytkowników, pozytywny wpływ systemu na ich odporność psychiczną.

Słowa kluczowe: odporność psychiczna, aplikacje mobilne, Flutter, Firebase, metody kreatywności

ABSTRACT

The purpose of the engineering project was to make a mobile application for strengthening mental toughness. The main motivation behind it, is to improve mental health of its users. The first step was to dive into the subject of mental toughness and to make an overview of existing solutions. The next thing was using creativity techniques like empathy map or "as is / to be" scenarios and survey study, to help come up with the best functionalities to implement. When the vision of the application was done, it was time to start working on technical aspects of the implementation, like system architecture and database structure. When the implementation was finished, there was a case study conducted with a group of picked volunteers. The opinions of the users proved, that the software made a positive impact on their mental toughness.

Keywords: mental toughness, mobile applications, Flutter, Firebase, creativity techniques

SPIS TREŚCI

Wykaz ważniejszych oznaczeń i skrótów	6
1. WSTĘP I CEL PRACY (PATRYK GOŁEMBIEWSKI, SZYMON HEJMANOWSKI)	7
2. ANALIZA MERYTORYCZNA	8
2.1. Odporność psychiczna (Szymon Hejmanowski)	8
2.2. Przegląd istniejących rozwiązań (Patryk Gołembiewski)	9
2.3. Zastosowane metody kreatywności (Szymon Hejmanowski)	12
2.3.1. Scenariusze użycia	12
2.3.2. Mapa empatii	16
2.3.3. Negatywna burza mózgów	18
2.4. Badania ankietowe (Patryk Gołembiewski)	18
2.4.1. Warunki początkowe	18
2.4.2. Przebieg badania oraz wnioski	19
2.5. Wizja aplikacji (Szymon Hejmanowski)	20
2.5.1. Kryteria wyboru funkcjonalności	20
2.5.2. Zbiór funkcjonalności	21
2.6. Refleksje dotyczące analizy merytorycznej	22
3. ASPEKT TECHNICZNY SYSTEMU	24
3.1. Technologia implementacji (Szymon Hejmanowski)	24
3.1.1. Frontend	25
3.1.2. Backend	25
3.2. Architektura systemu (Szymon Hejmanowski)	26
3.2.1. Elementy systemu	26
3.2.2. Komunikacja	27
3.3. Przypadki użycia aplikacji (Patryk Gołembiewski)	27
3.4. Schemat bazy danych (Patryk Gołembiewski)	30
3.5. Projektowanie interfejsu użytkownika (Szymon Hejmanowski)	31
3.6. Implementacja i testowanie (Patryk Gołembiewski)	32
3.6.1. Praca w zespole	32
3.6.2. Implementacja	33
3.6.3. Testowanie	37
3.6.4. Problemy w trakcie implementacji	37
3.7. Refleksje dotyczące analizy technicznej	39
4. STUDIUM PRZYPADKU ZASTOSOWANIA SYSTEMU	40
4.1. Założenia i przebieg (Patryk Gołembiewski)	40
4.1.1. Cele ankiety	40
4.1.2. Struktura pytań	40
4.1.3. Kryteria wyboru ankietowanych	41
4.1.4. Sposób przeprowadzenia badania	42
4.2. Analiza wyników (Szymon Hejmanowski)	42

5. PODSUMOWANIE (PATRYK GOŁEMBIEWSKI, SZYMON HEJMANOWSKI).....	44
Wykaz literatury	45
Wykaz rysunków	45
Wykaz tabel	46
Dodatek A: Treść pytań pierwszego badania ankietowego	48
Dodatek B: Treść pytań drugiego badania ankietowego	50

WYKAZ WAŻNIEJSZYCH OZNACZEŃ I SKRÓTÓW

UI - Interfejs Użytkownika (ang. *User Interface*)

UML – Zunifikowany Język Modelowania (ang. *Unified Modeling Language*)

EA – Enterprise Architect

PR – *Pull Request*

JSON - *JavaScript Object Notation*

SQL – Strukturalny Język Zapytań (ang. *Structured Query Language*)

API – Interfejs Programowania Aplikacji (ang. *Application Programming Interface*)

1. WSTĘP I CEL PRACY (PATRYK GOŁEMBIEWSKI, SZYMON HEJMANOWSKI)

W obecnym dynamicznie rozwijającym się świecie, którego charakterystyką jest szybkie tempo życia, wiele codziennych wyzwań oraz presja społeczeństwa, niesamowicie istotnym aspektem naszego życia jest zdrowie psychiczne. Problemy związane z odpornością psychiczną, definiowaną jako zdolność jednostki do radzenia sobie z trudnymi sytuacjami w życiu, wyzwaniami oraz stresem, stają się z każdym dniem większą bolączką dla współczesnego społeczeństwa. Niniejsza praca powstała w celu zaadresowania opisanego problemu, dzięki połączeniu możliwości, jakie daje nam współczesna technologia oraz psychologicznego modelu 4C.

Celem pracy jest stworzenie aplikacji mobilnej, która będzie łączyła w sobie najlepsze cechy istniejących już rozwiązań oraz doda nowe funkcjonalności na podstawie zebranej wiedzy w dziedzinie twardości psychicznej. Aplikacja ta pod względem technicznym powinna zapewniać przejrzysty i intuicyjny interfejs graficzny użytkownika, szybką obsługę zapytań do serwera oraz możliwość tworzenia własnego konta w celu separacji danych. Praca nad aplikacją rozpoczyna się od analizy merytorycznej, która realizowana była pod opieką dr inż. Anny Bobkowskiej. Rozdział ten zakłada zgłębienie wiedzy w dziedzinie odporności psychicznej z dużym naciskiem na wykorzystanie modelu 4C, następnie dokonuje przeglądu istniejących rozwiązań, który pozwala na ekstrakcję najczęściej występujących funkcjonalności podobnych systemów. Zastosowanie takich metod kreatywności jak: scenariusze użycia, mapa empatii czy negatywna burza mózgów oraz przeprowadzenie badania ankietowego, pozwala na wykreowanie wielu potencjalnych funkcjonalności, z których część zostaje ostatecznie wybrana i włączona do wizji systemu.

Następnym etapem jest stworzenie systemu od strony technicznej. Pierwszym krokiem jest wybór odpowiednich technologii implementacyjnych, które zapewnią optymalizację pracy nad aplikacją. Następnie dzięki architekturze systemu, możliwe jest zagłębienie się w poszczególne jego części oraz to, w jaki sposób między sobą się komunikują. Schemat bazy danych wyjaśnia jakie informacje przechowywane są na serwerze oraz jaki jest sposób ich organizacji. Gdy stworzone są już techniczne podstawy, nadchodzi czas na zrealizowanie makiet graficznych interfejsu użytkownika, które będą wzorem przy implementacji kodu źródłowego poszczególnych ekranów. Przypadki użycia ułatwiają prześledzenie procesu korzystania z aplikacji i zadbanie o każdy jej aspekt. Praca opisuje również sam proces implementacji i testowania. Ostatecznie przeprowadzone jest studium przypadku zastosowania systemu. Opisane są jego cele oraz przebieg, a także analiza uzyskanych wyników.

2. ANALIZA MERYTORYCZNA

W celu stworzenia optymalnego narzędzia pierwszym krokiem, przed zgłębieniem aspektów technicznych, jest właściwe przygotowanie od strony merytorycznej. Aby stworzyć aplikację, która będzie zgodna z założeniami, zgłębiliśmy wiedzę w dziedzinie psychologii pod kątem odporności psychicznej oraz prześledziliśmy podobne aplikacje dostępne na rynku. Uzbrojeni w nową wiedzę zastosowaliśmy wybrane metody kreatywności, aby usprawnić proces tworzenia pomysłów oraz wizji aplikacji. Przeprowadziliśmy dodatkowo badanie ankietowe wśród osób, które korzystały z podobnych narzędzi, co dało nam dodatkową perspektywę oraz pomysły na naszą aplikację. Ostatecznie dzięki wszystkim wymienionym procesom, stworzyliśmy wizję realizowanego systemu.

2.1. Odporność psychiczna (Szymon Hejmanowski)

Odporność psychiczna, zwana również twardością, wytrzymałością lub rezyliencją to cecha, która jest bardzo ceniona. Osoba, która posiada tę cechę, ma zdolność wytrwania w trudnych sytuacjach i skupienia się na zadaniu. Wytrzymałość psychiczna nie jest czymś, z czym się człowiek rodzi, raczej jest rozwijana i doskonalona z biegiem czasu poprzez doświadczenia i wyzwania, które testują granice naszej odporności. Utrzymanie pozytywnego nastawienia i umiejętność dostosowania się do zmieniających się okoliczności to kluczowe czynniki w budowaniu odporności psychicznej. Ostatecznie wytrzymałość psychiczna jest tym, co odróżnia tych, którzy odnoszą sukcesy, od tych, którym się to nie udaje.

Stosowanie skutecznych strategii radzenia sobie w różnych sytuacjach może pomóc nam wzmocnić naszą odporność psychiczną, która jest niezbędnym narzędziem w pokonywaniu trudności i przeciwności losu. Nie wszyscy rodzimy się z takim samym poziomem odporności psychicznej, ale ta cecha nie zależy wyłącznie od naszych wrodzonych cech charakteru czy predyspozycji genetycznych. Poprzez praktykę i naukę możemy rozwijać naszą odporność psychiczną i stać się bardziej zdolni do znoszenia życiowych trudności. Potwierdza to swoimi słowami psycholog oraz psychoterapeuta z wieloletnim doświadczeniem – Aleksandra Baca-Marzecka opisała „Badacze uważają odporność psychiczną za cechę osobowości warunkowaną genetycznie, którą jednak można modyfikować przez doświadczenie oraz wykształcanie pewnych nawyków. Siła i odporność psychiczna nie są stałe, można je rozwijać w zasadzie w każdym wieku” [1].

Dbanie o zdrowie psychiczne i budowanie odporności psychicznej stało się najwyższym priorytetem w dzisiejszym zabieganym i stresującym świecie. Ponieważ stres i napięcie są powszechnymi zjawiskami, negatywnie wpływającymi na dobrostan, skupienie się na zdrowiu psychicznym stało się kluczowym tematem. Osoby o silnej odporności psychicznej są lepiej przygotowane do radzenia sobie z problemami, są mniej podatne na stres, lepiej radzą sobie z emocjami i relacjami z innymi oraz odnoszą większe sukcesy w osiąganiu swoich celów. Mówiąc o odporności psychicznej, nie można nie wspomnieć o modelu 4C, który definiuje jej najważniejsze składniki. Model 4C składa się z czterech filarów:

- **Wyzwanie** (ang. *challenge*) – oznacza, że powinniśmy postrzegać trudności i wyzwania jako szanse do rozwoju i nauki, a nie jako problemy i przeszkody. Wyzwania stymulują rozwój i umożliwiają nam wzmocnienie swojej odporności psychicznej, dzięki czemu jesteśmy

w stanie radzić sobie z przeciwnościami losu. Istotne jest, abyśmy traktowali problemy jako okazje do wzmocnienia swojej odporności psychicznej i szansę do rozwoju.

- **Pewność siebie** (ang. *confidence*) – jest ona kluczowa dla budowania odporności psychicznej, ponieważ pomaga nam w radzeniu sobie z trudnymi sytuacjami i wyzwaniami. Posiadanie silnej pewności siebie oznacza, że jesteśmy przekonani o swoich umiejętnościach, zdolnościach i wartości, co pozwala nam na radzenie sobie z przeciwnościami losu bez utraty poczucia własnej wartości.
- **Zaangażowanie** (ang. *commitment*) – oznacza, że jesteśmy gotowi do poświęceń i pracy w celu osiągnięcia swoich celów. W kontekście budowania odporności psychicznej oznacza to, że jesteśmy skłonni do regularnej pracy nad sobą, rozwoju i poszukiwania nowych umiejętności, co przekłada się na naszą zdolność do radzenia sobie z przeciwnościami.
- **Kontrola / poczucie wpływu** (ang. *control*) - oznacza, że mamy wpływ na swoje życie i decyzje. Poczucie kontroli nad własnym życiem wpływa na naszą zdolność do radzenia sobie z trudnościami i wyzwaniami, ponieważ daje nam poczucie pewności i siły. Istotne jest, abyśmy mieli poczucie kontroli nad swoim życiem, abyśmy czuli się w stanie wpływać na swoją sytuację życiową.

„To właśnie percypowanie nowych wyzwań jako potencjalnych szans na odniesienie sukcesu (wyzwanie), wiara we własne siły i możliwości (pewność siebie), wytrwałość w dążeniu do wyznaczonego celu (zaangażowanie) oraz poczucie realnego wpływu na swoje życie (kontrola) stanowią cztery filary (cztery razy C) ogólnej odporności psychicznej.” — Małgorzata Ohme [2]

2.2. Przegląd istniejących rozwiązań (Patryk Gołembiewski)

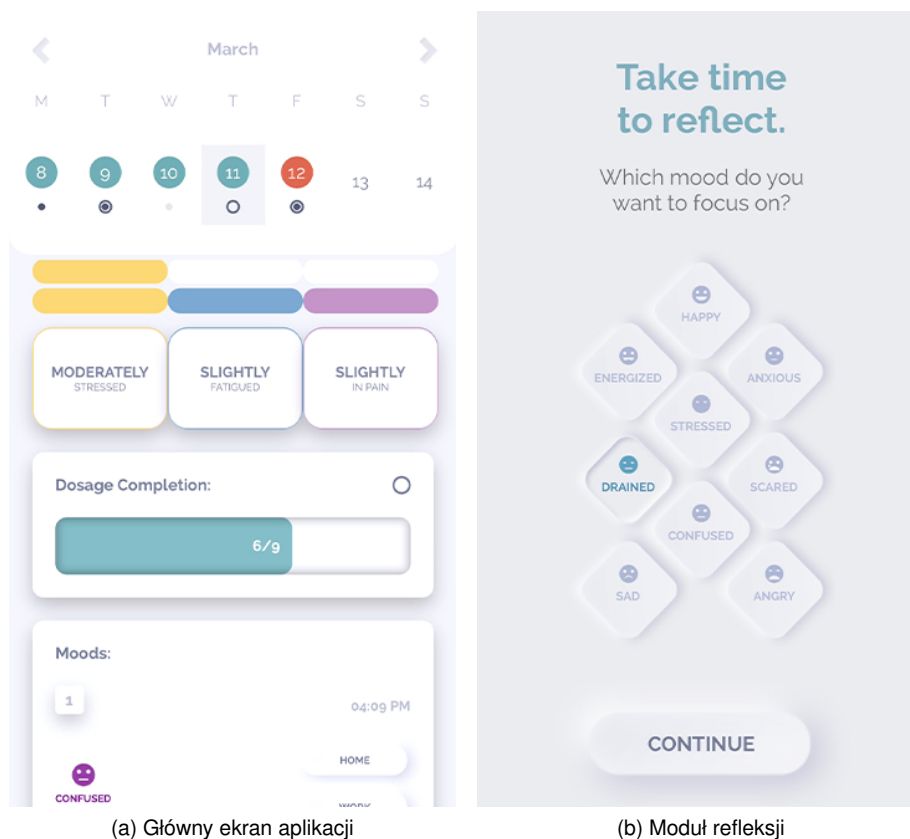
Jednym ze sposobów wzmocniania odporności psychicznej jest korzystanie z dedykowanych do tego aplikacji. Szczególnie wygodne w użytkowaniu, ze względu na dostępność, są systemy dostępne na urządzeniach mobilnych. Dostępne są różne rozwiązania i funkcjonalności, które pozytywnie oddziałują na samopoczucie człowieka i radzenie sobie z trudnościami i wyzwaniami żywotnymi, a także na rozwijanie pewności siebie i umiejętności radzenia sobie ze stresem. W tym akapicie skupimy się na omówieniu systemów i aplikacji, które zostały stworzone w celu wzmocniania rezyliencji psychicznej i pomagają w osiągnięciu lepszej jakości życia. Przeanalizowane zostały następujące aplikacje:

- Personal Zen[6],
- Breath2Relax[7],
- Sanvello[8],
- Serenita[9],
- Woebot[10],
- Fitbit.[11]

Poniżej znajdują się opisy każdej z nich.

Personal Zen to aplikacja mobilna zaprojektowana specjalnie dla osób doświadczających stresu i lęku. W ramach głównej praktyki użytkownicy codziennie sprawdzają poziom stresu i nastroju. Na podstawie opinii aplikacja ustala spersonalizowany cel, na przykład zalecaną liczbę minut grania dla danego użytkownika, zazwyczaj od 5 do 10 minut dziennie. W grze pojawiają się dwa duchy – przyjacielski i zły. Użytkownikom zaleca się podążanie ścieżką stworzoną przez dobrego ducha. Użytkownicy powtarzają to ćwiczenie, aż osiągną swoje codzienne cele. Inne

funkcje aplikacji obejmują dziennik nastroju i treści psychoedukacyjne. Przykładowe zrzuty ekranu z aplikacji zostały przedstawione na rysunku 2.1.



(a) Główny ekran aplikacji

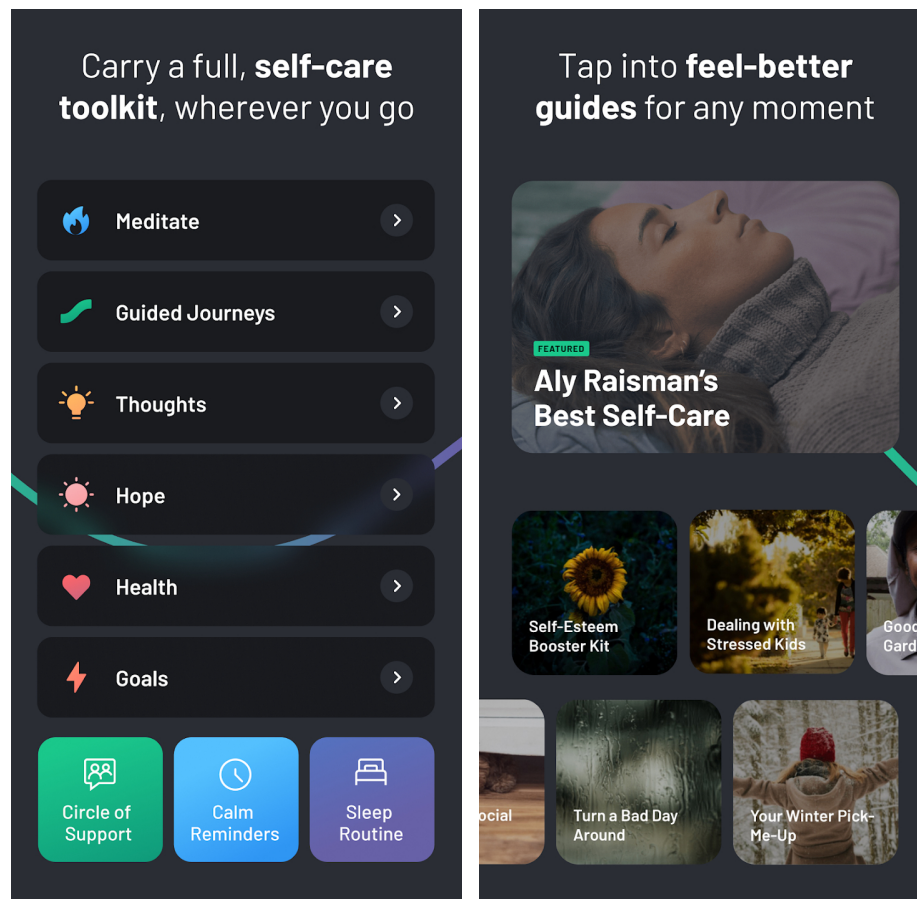
(b) Moduł refleksji

Rys. 2.1. Zrzuty ekranu z aplikacji Personal Zen

Breathe2Relax to darmowa mobilna aplikacja z dziedziny zdrowia i fitnessu. Zawiera ona ćwiczenie prowadzonego oddechu przeponowego oraz przydatne informacje na temat wpływu stresu. Taka forma pracy z oddechem związana jest często z praktyką *mindfulness*.

Sanvello, zarówno aplikacja internetowa, jak i mobilna, oferuje użytkownikom ulgę od negatywnych emocji, takich jak niepokój, depresja i stres. Wykorzystując strategie wywodzące się z terapii poznawczo-behawioralnej, medytacji uwagi, relaksacji i śledzenia nastroju/zdrowia, Sanvello oferuje podejście oparte na dowodach. Ten program zaspokaja potrzeby użytkownika, dostosowując swoje rekomendacje na podstawie generowanych raportów i preferencji. Po zarejestrowaniu się Sanvello żąda aktualizacji obecnego stanu użytkownika, w tym nastroju, stanu zdrowia i celów, dostarczając precyzyjnych zaleceń dotyczących określonych działań. Aplikacja oferuje różnorodne opcje, od wielodniowych ścieżek samopomocy z przewodnikiem, po techniki relaksacyjne, a nawet czaty grupowe. Przykładowe zrzuty ekranu z aplikacji zostały przedstawione na rysunku 2.2

Dzięki **Serenicie** użytkownicy mogą bez wysiłku zarządzać i śledzić poziom stresu. Aby ocenić poziom stresu, aplikacja wykorzystuje aparat telefonu w funkcji „Stress Check”. Aby zredukować stres lub zwiększyć koncentrację, Serenita zapewnia różnorodne ćwiczenia. Aktywność interaktywnej relaksacji rozpoczyna się od pomiaru poziomu stresu i wzorców oddychania za pomocą 50-sekundowego testu stresu. Podczas ćwiczenia oddechowego obecny jest wizualny przewodnik, który instruuje użytkownika, kiedy należy wdychać, wstrzymywać i wydychać powietrze. Po pomyślnym ukończeniu ćwiczenia oddechowego użytkownik otrzymuje punkty. Korzystając



(a) Ekran wyboru modułu

(b) Ekran z poradnikami

Rys. 2.2. Zrzuty ekranu z aplikacji Senvello

z innego wzorca oddychania, ćwiczenie Interaktywne Skupienie zostało zaprojektowane w celu zwiększenia koncentracji. Długość ćwiczeń można dostosować do własnych potrzeb, do 5 minut. Po wykonaniu ćwiczenia aplikacja dostarcza informacji o wzroście lub spadku poziomu stresu w całym ćwiczeniu. Funkcja „trend” dostarcza wykres postępu użytkownika na każdym ćwiczeniu w czasie. Dostępnych jest pięć sesji każdego ćwiczenia, które zawarte zostały w bezpłatnej wersji aplikacji.

Woebot jest w pełni zautomatyzowanym agentem konwersacyjnym. Woebot jest dostępny za pośrednictwem urządzeń mobilnych poprzez aplikację komunikatora. Aplikacja zawiera lekcje, interaktywne ćwiczenia i filmy wideo oparte na terapii poznawczo-behawioralnej. Interakcje z Woebotem są dosyć krótkie (1-2 minuty). Istnieje jednak możliwość wydłużenia sesji.

Poniższy fragment przedstawia zestawienie aplikacji oraz funkcjonalności, które najczęściej w nich występowały. Opis funkcjonalności przytaczanych w tabeli 2.1:

- Dziennik – zapisuje się w nim codzienny nastrój i poziom stresu.
- Gra – prosta w założeniu, pomaga uspokoić myśli i na chwilę się na czymś skupić.
- Oddech – instruktor oddechu, ułatwia uspokojenie się i wejście w stan medytacji
- Ćwiczenia – lista ćwiczeń dobranych tak, aby pomagały w relaksacji.
- Chatbot – czat AI z którym możemy pogadać.
- Sen – pomaga wyregulować godziny snu i poprawić jego jakość.

Po przeanalizowaniu powyższych oraz innych aplikacji, autorom nasunęły się następujące refleksje. Użytkownik ma dostęp do wielu różnych rozwiązań, jednak wiele z nich posiada funk-

Tabela 2.1. Porównanie funkcji podobnych systemów

	Dziennik	Gra	Oddech	Ćwiczenia	Chatbot	Sen
Personal Zen	+	+				
Breath2Relax			+			
Sanvello	+		+	+		
Serenita	+	+	+			
Woebot	+				+	
Fitbit	+	+				+

cjonalności niewpływające bezpośrednio na zwiększenie odporności psychicznej. Przykładem są gry, które odciągają uwagę odbiorcy od złych myśli. Pozwalają one jedynie na chwile zapomnieć o problemach, jednak gdy użytkownik przestanie używać tego modułu, problemy wracają. Ponownie musi wrócić do świata rzeczywistego i zderzyć się ze swoimi problemami. Kolejnym spostrzeżeniem jest możliwa wadliwość sztucznej inteligencji w sprawach emocjonalnych i szeroko rozumianej psychiki człowieka. Czat bot może mieć trudności w zrozumieniu kontekstu rozmowy, co może prowadzić do błędów interpretacyjnych i nieodpowiednich odpowiedzi, a w skutku, negatywnie odbić się na użytkowniku. Inną wadą jest brak różnorodności w ćwiczeniach. Choć aplikacja oferuje interesujące i skuteczne ćwiczenia, to po pewnym czasie użytkownicy mogą się nudzić i potrzebować czegoś nowego, aby utrzymać swoją motywację do korzystania z aplikacji. Przykładowo Personal Zen oferuje tylko jedno główne ćwiczenie, co może być niewystarczające dla niektórych użytkowników. Brak nowych treści lub różnorodnych ćwiczeń może skutkować tym, że użytkownicy przestaną korzystać z aplikacji lub szukają innych aplikacji, które oferują więcej różnorodności.

2.3. Zastosowane metody kreatywności (Szymon Hejmanowski)

Tworzenie aplikacji mającej na celu poprawę odporności psychicznej to proces wymagający wykorzystania różnych narzędzi i technik. Jednym z kluczowych aspektów jest kreatywność, która pozwala na generowanie nowych pomysłów i podejście do projektu z różnych perspektyw. W celu zwiększenia kreatywności, w ramach pracy inżynierskiej zastosowaliśmy trzy różne metody, a mianowicie: scenariusze użycia, mapę empatii oraz negatywnej burzy mózgów. Ich wykorzystanie miało na celu uzyskanie bardziej wartościowych i innowacyjnych rozwiązań, które są dostosowane do potrzeb i oczekiwań użytkowników. W dalszej części rozdziału przedstawione zostaną szczegółowe informacje na temat każdej z tych metod, wraz z opisem ich zastosowania w projektowaniu aplikacji.

2.3.1. Scenariusze użycia

Metoda scenariuszy użycia to podejście projektowe polegające na stworzeniu szczegółowych opisów procesów, jakie użytkownicy będą wykonywać przy korzystaniu z danego produktu lub usługi. Polega ona na identyfikacji problemów, jakie mogą napotkać użytkownicy w trakcie interakcji z systemem, a następnie na opracowaniu sposobów, w jaki te problemy mogą zostać rozwiązane. Scenariusze użycia mogą być tworzone na różnych etapach projektowania, a ich celem jest poprawa użyteczności i łatwości korzystania z danego systemu.

W przypadku naszego projektu scenariusze użycia stworzono w formie tabeli z kolumnami *AS IS* oraz *TO BE*. Do jednej kolumny, oznaczonej jako *AS IS*, wpisane zostały opisy sytuacji, w jakich może znajdować się osoba z problemami na podłożu odporności psychicznej. Kolumna *TO BE* zawiera z kolei scenariusz, w którym z wykorzystaniem naszej aplikacji osoba radzi sobie z danym problemem. Za pomocą tej metody stworzyliśmy opisy sytuacji, zawierające zbiór pomy-

słów na funkcjonalności, jakie moglibyśmy zaimplementować w naszej aplikacji. Pomogło to nam w procesie generowania funkcji systemu oraz zobrazowało które z pomysłów mają praktyczne zastosowanie. Poniżej znajduje się zbiór zestawień *AS IS* oraz *TO BE* scenariuszy użycia.

Scenariusz 1

- *AS IS*: Osoba szybko zniechęca się do wykonywanych zadań, ponieważ nie czuje namacalnego progresu. Przez to, że myśli jedynie o tym, co ma jeszcze do wykonania (a nie odnotowuje, ile już zrobiła) ma ciągłe wrażenie braku postępu, co ją wysoce demotywuje. Brak kontroli oraz zaangażowania.
- *TO BE*: Osoba dzięki aplikacji i zapisywaniu każdego zadania może w każdej chwili sprawdzić, ile już zrobiła, dzięki temu czuje progres, co motywuje ją do dalszego działania. „Odhażanie” kolejnego zadania z listy, powoduje małe zastrzyki dopaminy, co w długim terminie tworzy w mózgu połączenia łączące wykonywanie kolejnych zadań z przyjemnością.

Scenariusz 2

- *AS IS*: Osoba nie ma jasno wyznaczonych i sprecyzowanych celów, do jakich dąży. Chce schudnąć, jednak nigdzie nie odnotowała ile i w jakim czasie. Przez to, że cel jest bardzo ogólny i bez wyznaczonego terminu, jest on tak naprawdę jedynie marzeniem. Osobie tej, wydaje się, że od czasu do czasu zje mniej lub pójdzie pobiegać, więc dąży do realizacji celu, jednak nie zdaje sobie sprawy, że wykonywana praca jest wyraźnie zbyt mała, aby odnieść rezultat. Tkwi w przekonaniu, że dąży do celu, jednak w rzeczywistości jedynie stoi w miejscu. Taki status quo w dziedzinie jej celów obniża jej pewność siebie, ma wrażenie, że jest gorsza od innych, bo też pracuje, jednak bez rezultatu. Dodatkowo traci poczucie kontroli, ma wrażenie, że nie ma wpływu, na to co się dzieje w jej życiu. Nie jest w stanie realnie się zaangażować. Również taki stan negatywnie wpływa na obszar wyzwań z modelu 4C – nigdy nie ponosi jednoznacznej porażki (ponieważ jej cel nie ma terminu ani konkretnych założeń), z której mogłaby wyciągnąć lekcje, a jednocześnie nigdy nie osiąga celu.
- *TO BE*: Osoba dodaje nowy cel w aplikacji zgodnie z formularzem. Podaje tam obiektywne i mierzalne wartości, które chce osiągnąć (konkretna waga, ilość pieniędzy, jakie chce zarobić, ilość materiału do nauczania) oraz termin realizacji tych zadań. W obszarze celu może również ustawić tak zwane kamienie milowe, czyli cele pośrednie, które są kluczowe do osiągnięcia ostatecznego celu. Widok bliższego celu pośredniego na horyzoncie daje jej dużą motywację na początku i podtrzymuje ją w momentach zwątpienia. Osoba ta ma realne cele i w pełni świadomie dąży do nich każdego dnia. Osoba widząc postęp w kolejnych celach, zyskuje poczucie kontroli nad sobą oraz własnym życiem. Pozwala jej się to również zaangażować i trwać w swoich postanowieniach. Jeśli któregoś z celów nie wypełni, zdaje sobie sprawę ze swojej porażki, co pozwala jej wyciągnąć lekcje, wpływa to dobrze na obszar Wyzwania. Większa kontrola oraz widoczny rozwój zwiększa jej pewność siebie i przekłada się pozytywnie na inne obszary.

Scenariusz 3

- *AS IS*: Osoba ma skłonności do pesymizmu, lubi się zamartwiać i znajdować we wszystkim jedynie negatywy. Jest to dla niej demotyujące i zabija w niej chęć rozwoju. Osoba twierdzi, że rozwój i tak nic nie zmieni i że w ogóle jest bez sensu.

- *TO BE*: Aplikacja zapewnia osobie codzienne cytaty motywujące oraz inspirujące myśli tych, którzy wiele osiągnęli. O ile nie wpływa to z dnia na dzień na osobę, to jednak w dłuższym okresie pozwala jej unikać pułapki ciągłego pesymizmu i odżywia jej mózg pozytywnymi myślami.

Scenariusz 4

- *AS IS*: Osoba przez lata złych nawyków kształtowanych przez system szkolny oraz własne zaniedbania ma duże problemy ze snem. Nie zdaje sobie sprawy z tego, na jak wiele obszarów zdrowia oraz pracy mózgu on wpływa. Zaniedbuje sen, śpiając w różnych godzinach oraz często niewystarczająco długo. Ma problemy z przyswajaniem i zapamiętywaniem, również pamięć mięśniowa na tym traci, przez co np. nauka gry na fortepianie staje się dla niej dużo trudniejsza. Odczuwa również negatywne skutki słabej higieny snu poprzez częste przeziębienia. Nierzadko musi odpuszczać przez to treningi lub wyjścia ze znajomymi. Ma poczucie braku kontroli nad własnym zdrowiem. Dodatkowo trudności w przyswajaniu informacji oraz w nauce nowych czynności fizycznych zmniejszają jej pewność siebie.
- *TO BE*: Dzięki modułowi poprawy snu osoba może dowiedzieć się o tym, jak ogromne skutki ma jego zaniedbywanie. Dzięki przejrzeniu panelu poświęconemu krótkim informacjom z badań o negatywnych skutkach niepoprawnego snu znajduje potencjalne rozwiązanie swoich wielu problemów. Motywuje ją to do działania w tym obszarze życia. Ustala w aplikacji harmonogram snu, co skutkuje codziennymi przypomnieniami lub blokowaniem innych aplikacji w odpowiednich godzinach. Sprawdza również panel zapewniający krótkie porady o tym, jak zadbać o dobrą jakość snu (zmniejszenie temperatury pokoju, przygotowanie do snu poprzez zmniejszenie tętna). Poprawa ilości i jakości snu wpływa pozytywnie na każdy aspekt życia osoby. Lepiej działający układ immunologiczny redukuje często liczbę przeziębień, dzięki czemu osoba nie musi opuszczać treningów i innych aktywności. Osoba przyswaja więcej materiału i pamięta go lepiej, co również redukuje czas, jaki musi poświęcać na naukę. Dodatkowo nauka gry na fortepianie idzie nagle zauważalnie lepiej. Osoba zyskuje pewność siebie oraz poczucie kontroli nad własnym życiem i zdrowiem.

Scenariusz 5

- *AS IS*: Osoba ma problemy ze skupieniem. W każdej wolnej chwili przegląda social media lub ogląda seriale. Nie pozostawia jej to praktycznie żadnego czasu na uspokojenie myśli. Jedyne, kiedy jest sama ze swoimi myślami to moment, kiedy kładzie się spać. Nie pozwala jej to wyłączyć się i spokojnie zasnąć. Skupienie się przez dłuższy moment na pracy głębokiej lub na czynności niedostarczającej do mózgu dużych ilości bodźców jest dla niej praktycznie nierealne. Przy nudnych aktywnościach odruchowo co chwilę sięga po telefon. Podczas czytania książki, osoba się irytuje, musi wielokrotnie czytać każde zdanie, ponieważ co chwilę jej myśli pędzą w inną stronę. Ciągła praca mózgu na wysokich obrotach powoduje u osoby niepokój. Nie jest nawet do końca świadoma, z czego on wynika, jednak wyraźnie go odczuwa z różną intensywnością na przestrzeni dnia. Osoba taka nie ma poczucia kontroli nawet nad własnymi myślami. Nie jest w stanie zaangażować się w pracę, którą wykonuje.
- *TO BE*: Dzięki użytkowaniu aplikacji z zawartym modulem medytacji / pracy nad oddechem, osoba zaznaje każdego dnia parunastu minut uspokojenia myśli. Jest w stanie codziennie

pracować nad swoją uwagą. W jej głowie zaczynają pojawiać się nowe myśli. Po pewnym okresie codziennej medytacji niepokój osoby stopniowo zanika, jest ona w stanie o wiele łatwiej i lepiej skupić się na wykonywanych czynnościach. Często zdarza jej się wchodzić w tryb „flow”. Czynności, które wcześniej wydawały się nudne (np. czytanie książki), teraz sprawiają jej przyjemność. Na co dzień nie ma wrażenia ciągłej gonitwy myśli, jest w stanie być obecna w danej chwili. Spokojny i uważny umysł pozytywnie przekłada się również na zasypianie. Przed snem osoba korzystając z aplikacji, uspokaja myśli i kładzie się do snu w idealnym stanie. Osoba zyskuje dużo lepsze poczucie kontroli oraz jest w stanie angażować się w codzienne wyzywania.

Scenariusz 6

- *AS IS*: Osoba przechodzi przez każdy kolejny dzień bez refleksji, nie zwraca uwagi na to, jakie wydarzenia z dnia wpływają na jej samopoczucie. Często popełnia te same błędy, ponieważ nie poświęca czasu na przemyślenie ich i znalezienie rozwiązania. Nie jest do końca świadoma swoich własnych emocji. Wraca w głowie do przeszłości jednak nie w celu jej analizy i wyciągnięcia wniosków, lecz w celu rozmyślań nad tym, co by było, gdyby postąpiła inaczej. Tworzy w ten sposób negatywne nastawienie do swojej własnej osoby, jednocześnie nic nie zyskując. Traci pewność siebie.
- *TO BE*: Osoba z użyciem aplikacji poświęca codziennie chwilę, by przeanalizować miniony dzień. Zapisuje problemy, jakie wystąpiły danego dnia oraz znajduje ich rozwiązania / sposoby na uniknięcie w przyszłości. Jeśli dany problem pojawiał się kolejny raz, myśli nad tym, dlaczego poprzednie rozwiązanie było nieskuteczne i jak je poprawić / zmienić. Ocenia również swoje samopoczucie na koniec dnia, dzięki czemu zyskuje świadomość, co tak naprawdę wpływa na jej odczucia. Dzięki widocznej eliminacji błędów oraz świadomości emocji czuje, że zyskuje pewność siebie oraz kontrolę.

Scenariusz 7

- *AS IS*: Osoba każdego dnia budzi się bez konkretnych założeń, co do tego, co chciałaby zrobić. Ma w głowie zarys planów na dzień, jednak są one nieskonkretyzowane. W dni, w które ma wiele zadań, nie czuje potrzebnej presji, przez co dzień nie jest optymalny, a co za tym idzie, nie udaje jej się zrobić wszystkiego, co powoduje stres i niezadowolenie. W dni, kiedy ma mało zadań i dużo czasu, zamiast skorzystać z możliwości i zaplanować ciekawą aktywność na ten dzień lub wykonać jakąś pracę na zapas, traci pół dnia na bezproduktywne czynności. Nie ma poczucia kontroli nad własnym dniem, nie jest w stanie optymalnie wykonywać swoich wyzwań oraz nie jest tak zaangażowana, jakby mogła być.
- *TO BE*: Osoba codziennie wieczorem po wypełnieniu dziennika samopoczucia przechodzi do sekcji planowania następnego dnia. Na spokojnie zbiera wszystkie rzeczy, które powinna jutro zrobić i tworzy z nich listę. Nadaje im priorytety, szacunkowy czas wykonania oraz godziny. Widzi czy jest na niedoczasie i musi jutro wyjątkowo się wysilić, czy może ma trochę wolnego czasu, aby wyjść na kawę ze znajomymi. Następnego dnia od rana skoncentrowana jest na wypełnianiu zadań. Daje jej to duże poczucie satysfakcji, co ułatwia przystępowanie do kolejnych zadań. Dzięki przydziałowi na każde zadanie swojego czasu jest w stanie skupić się na obecnie wykonywanej czynności. Dodatkowo świadomość, na jakim etapie pracy jest w ciągu dnia, redukuje w dużym stopniu jej stres i niepokój. Odha-

czenie ostatniego zadania w trakcie danego dnia powoduje wyrzut dopaminy i zachęca do powtórzenia cyklu dnia kolejnego.

2.3.2. Mapa empatii

Drugą metodą kreatywności, którą zastosowaliśmy, jest mapa empatii[12]. Jest to narzędzie, które pozwala na lepsze zrozumienie potrzeb i oczekiwań użytkowników. Polega ona na stworzeniu mapy, na której umieszczamy informacje na temat użytkowników, ich myśli, emocji, działań oraz otoczenia. Dzięki temu możemy uzyskać bardziej szczegółowe informacje na temat użytkowników i ich potrzeb, a także wykorzystać te informacje do tworzenia bardziej wartościowych rozwiązań. Użyty przez nas szablon mapy empatii został przedstawiony na rysunku 2.3.

- **Kogo chcemy zrozumieć?** Osoba, która chce poprawić swoją odporność psychiczną. Osoba taka mierzy się na co dzień z wieloma wyzwaniami i problemami chce być w stanie zwalczać je z większą łatwością oraz mieć większy komfort psychiczny.
- **Co ta osoba musi zrobić?** Zapoznać się z aplikacją. Nauczyć się jej obsługi. Uzupełnić cele długoterminowe. Codziennie zaglądać do aplikacji, aby uzupełnić dziennik oraz wyznaczyć cele na kolejny dzień.
- **Co osoba widzi?** Na co dzień widzi świat pełen trudności i problemów. Niezliczoną ilość przytłaczających wyzwań. W Internecie widzi innych, którzy cieszą się życiem i wydają się nie posiadać żadnej troski. Widzi ciągle zmiany, do których nie jest w stanie przywyknąć oraz ludzi, z których każdy wydaje się obcy.
- **Co to osoba mówi?** Mówi, że nie jest w stanie sobie poradzić z codziennością. Narzeka na nieustające trudności. Mówi, że nie czuje kontroli nad własnym życiem.
- **Co ta osoba robi?** Stoi w miejscu, pragnąc zmiany. Nie jest w stanie jednak niczego wskórać. Codziennie obiecuje sobie, że zrealizuje postawione cele, jednak jest to tylko złudne poczucie. Każdego dnia idzie spać o innej porze, przez co nie jest w stanie się wyspać. Nie pielęgnuje relacji z bliskimi. Jest reaktywna, zarówno w życiu zawodowym, jak i osobistym. Nie pomaga innym.
- **Co ta osoba słyszy?** Od innych słyszy, że musi wziąć się w garść i zrobić coś ze swoim życiem. Docierają do niej stale słowa krytyki od innych. Koledzy namawiają ją na wyjścia, jednak ona nie ma siły ani ochoty wychodzić z domu.
- **Co ta osoba myśli i czuje?**
 - **Bóle:** Ma lęk przed tym, że nie będzie w stanie zrealizować wszystkich zadań, oraz że nie radzi sobie z codziennymi problemami. Jest sfrustrowana swoją postawą i ciągłym brakiem poprawy. Boi się, że przez swoje zaniedbywanie kontaktów oraz złe nastawienie odwrócą się od niej znajomi. Obawia się o swoje zdrowie fizyczne z powodu zaniedbywania sfery somatycznej egzystencji.
 - **Nadzieje:** Ma nadzieję, że walka z codziennymi trudnościami będzie dla niej łatwiejsza. Chciałaby z uśmiechem stawiać czoła kolejnemu dniu. Zależy jej również na poprawie relacji z ludźmi. Liczy, że jej zdrowie fizyczne również się poprawi. Chciałaby mieć pełną kontrolę nad swoim życiem oraz większą pewność siebie. Obecnie czuje złość, że nie jest w stanie realizować swoich postanowień.

Empathy Map Canvas

Designed for:

Designed by:

Date:

Version:

1 WHO are we empathizing with?
 Who is the person we want to understand?
 What is the situation they are in?
 What is their role in the situation?

2 What do they need to DO?
 What do they need to do differently?
 What job(s) do they want or need to get done?
 What decision(s) do they need to make?
 How will we know they were successful?

3 What do they SEE?
 What do they see in the marketplace?
 What do they see in their immediate environment?
 What do they see others saying and doing?
 What are they watching and reading?

4 What do they SAY?
 What have we heard them say?
 What can we imagine them saying?

5 What do they DO?
 What do they do today?
 What behavior have we observed?
 What can we imagine them doing?

6 What do they HEAR?
 What are they hearing others say?
 What are they hearing from friends?
 What are they hearing from colleagues?
 What are they hearing second-hand?

7 What do they THINK and FEEL?
PAINS
 What are their fears, frustrations, and anxieties?
GAINS
 What are their wants, needs, hopes and dreams?

What other thoughts and feelings might motivate their behavior?

Last updated on 16 July 2017. Download a copy of this canvas at <http://gamestorming.com/empathy-map/>
 © 2017 Dave Gray, xplane.com

Rys. 2.3. Wykorzystany schemat mapy empatii

2.3.3. Negatywna burza mózgów

Trzecią metodą kreatywności, którą wykorzystaliśmy, jest negatywna burza mózgów. Jest to technika generowania pomysłów, która polega na znajdowaniu problemów, aby móc im potem przeciwdziałać. Metoda ta pozwala na stworzenie listy potencjalnych złych doświadczeń podczas korzystania z aplikacji. Dzięki temu pobudzamy wyobraźnię i wyzwalamy myślenie o niestandardowych rozwiązaniach. Była to ostatnia z metod kreatywności, jaką zastosowaliśmy, więc w tym przypadku poprzednie metody kreatywności w dużym stopniu wypełniły analizowany obszar systemu, jednak metoda negatywnej buzy mózgów miała na celu pomóc nam spostrzec niektóre przeoczone aspekty tworzenia systemu. Celem było zwrócenie uwagi między innymi na to, że aplikacja powinna być stworzona w taki sposób, aby korzystanie z niej nie stało się kolejnym czasochłonnym obowiązkiem w życiu danej osoby. Prócz tego, aby dana osoba korzystała z pełnym zaufaniem z metod zawartych w aplikacji, powinniśmy przedstawiać dowody naukowe popierające słuszność danej funkcjonalności. Rezultaty zostały przedstawione w tabeli 2.2.

Tabela 2.2. Wnioski płynące z negatywnej burzy mózgów

POTENCJALNE PROBLEMY	ROZWIĄZANIE
Korzystanie z aplikacji zajmuje zbyt dużo czasu w ciągu dnia. Staje się to dodatkowym obowiązkiem.	Optymalizacja czasu, który potrzebny jest na korzystanie z aplikacji. Stworzenie intuicyjnego UI (ang. <i>User Interface</i>).
Osoba przestaje używać aplikacji po krótkim czasie, ponieważ mija jej chwilowa motywacja.	Powiadomienia przypominające o postępie w celach oraz hasła motywujące.
Bariera językowa. Osoba nie ma dostępnego swojego języka.	Aplikacja dostępna w języku polskim oraz angielskim.
Osoba nie jest przekonana co do metod wykorzystywanych w aplikacji, nie chce z nich korzystać, ponieważ nie wie, czy działają.	W każdym module zawiera się panel z krótkim opisem, na jakiej zasadzie działa dana funkcjonalność. Dodatkowo poparte są badaniami.
Osoba nie widzi postępu w swoich działaniach.	Każdy panel ma swój pasek postępu i cele do wypełnienia. Np. medytuj 1000 razy

2.4. Badania ankietowe (Patryk Gołembiewski)

Ankiety mają coraz większe znaczenie w dzisiejszych badaniach nad zdrowiem psychicznym i zachowaniem. W kontekście aplikacji mobilnych, ankiety dotyczące potrzeb użytkowników, mogą dostarczyć cennych informacji na temat oczekiwań i preferencji dotyczących narzędzi wspierających zdrowie psychiczne. Wyniki takich badań mogą pomóc w projektowaniu i udoskonalaniu aplikacji, tak aby lepiej odpowiadały na potrzeby użytkowników. Istnieje wiele różnych aplikacji mobilnych, które mają na celu wspieranie zdrowia psychicznego, dlatego ankiety mogą pomóc w unikaniu błędów oraz mankamentów występujących w istniejących rozwiązaniach.

2.4.1. Warunki początkowe

Aby grupa osób badanych była reprezentatywna, wybraliśmy spośród ochotników osoby, które miały wcześniej styczność z aplikacjami, o celu zbliżonym do naszego. Miało to zapewnić dobre odzwierciedlenie grupy docelowej tworzonej aplikacji w osobach ankietowanych. Osoby różniły się pod względem miejsca zamieszkania w obrębie Polski, wieku oraz płci, jednak wszystkie miały element wspólny, czyli zainteresowanie kwestią dbania o swoją odporność psychiczną. Ponieważ zależało nam na obszernej opinii użytkowników, pytania w badaniu były otwarte, a gru-

pa osób badanych składała się z 10 uczestników. Pozwoliło to poznać nam dogłębne przemyślenia osób badanych oraz skrupulatnie je przeanalizować. Charakter badania z założenia miał stawiać na pozyskanie szerokiego spojrzenia mniejszej, lecz reprezentatywnej grupy osób. Treść wszystkich pytań została zawarta w załączniku.

2.4.2. Przebieg badania oraz wnioski

W początkowej fazie kwestionariusza zaczęliśmy od pytań odnośnie do wzmacniania odporności psychicznej. Większość ankietowanych jest zainteresowana wzmacnianiem odporności oraz stara się aktywnie nad nią pracować. Jest to dobry znak, gdyż nasza aplikacja mogłaby zostać jedną z możliwości dalszej pracy nad sobą dla takich osób. Istnieje zatem potrzeba utworzenia takiego systemu. Następnie ankietowani zostali zapytani o znajomość modelu 4C – „Czy jesteś zaznajomiony z modelem 4C?”. Niestety okazuje się, że nie do końca wiedzą czym on jest. Po zapoznaniu się z filarami tego modelu mieli oni wybrać najważniejszy z nich według ich uznania. Faworytami okazały się pewność siebie oraz zaangażowanie. W kolejnym pytaniu odpowiedzi użytkowników nawiązały do drugiego z wymienionych filarów. Uznali oni brak zaangażowania oraz trzymanie się wyznaczonych celów za najtrudniejsze elementy życia codziennego – „[Najtrudniejsze jest] trzymanie się wyznaczonych przez siebie celów i kultywowanie zdrowych nawyków [...] systematyczność.” – stwierdził jeden z ankietowanych.

Następnym zagadnieniem poruszonym w ankiecie był pozytywny wpływ różnych rzeczy na stan psychiczny. Analizując odpowiedzi, doszliśmy do wniosku, że różne metody poprawy odporności psychicznej działają na różne osoby w różny sposób. Wśród najczęściej wymienianych sposobów na poprawę odporności psychicznej są poprawa snu i diety, regularne ćwiczenia fizyczne, medytacja w celu wyciszenia i zrozumienia własnych emocji oraz śledzenie postępu, aby widzieć progres i nie tracić zaangażowania. Pełna treść pytań zawartych w badaniu została załączona w dodatku A.

Ostatnia część dotyczyła aplikacji, które wzmacniają odporność psychiczną. Ankietowani jednogłośnie uznali, że aplikacje mogą mieć duży wpływ na zwiększanie odporności. W kolejnym pytaniu wymienili sposoby wpływania systemów wspierających odporność psychiczną:

- „Motywowanie do realizacji celów, uspokajanie.”
- „Poprzez dawanie porad, jak i pomocnych zadań.”
- „Mogą pokazać, że jesteś w czymś stabilny lub zrobiłeś jakiś postęp względem przeszłości.”
- „[...] aplikacja, która pozwoli się na chwilę zatrzymać, przypomnieć o postawionych sobie celach i wyzwaniach, sprawdzić, jakie robimy postępy.”

Użytkownicy napisali również powody, dla których wcześniej rezygnowali z korzystania z podobnych aplikacji. Pojawiającą się odpowiedzią było rozbieżność funkcjonalności po różnych aplikacjach, a co za tym idzie szybkie zniechęcanie się do nich. Pytaniem zwieńczającym ankietę była prośba o wypisanie funkcjonalności, które wg ankietowanych będą użyteczne w aplikacji. Oto przykładowe, najczęstsze odpowiedzi:

- „Możliwość monitorowania progresu, system nagród”
- „Dziennik codziennego stanu emocjonalnego, możliwość połączenia zegarka z funkcją monitorowania stresu i monitorowanie wyników, motywowanie w osiąganiu celów – system nagród”
- „Śledzenie postępu w postanowionych celach, instrukcje, poradniki mające na celu poprawić jakość snu, diety, zarządzanie czasem.”

2.5. *Wizja aplikacji (Szymon Hejmanowski)*

Głównym założeniem naszej aplikacji jest poprawa odporności psychicznej jej użytkowników. Poprzez działanie oparte na modelu 4C ma ona ułatwiać wprowadzanie metod, pozytywnie wpływających na rezyliencję w codziennym życiu. Chcemy, aby aplikacja była bodźcem pozwalającym na systematyczny oraz przemyślany rozwój psychiczny użytkownika. Dzięki dobraniu odpowiednich funkcjonalności oraz ich przemyślanej implementacji otrzymujemy produkt, który pozwala na optymalne wprowadzenie nowych nawyków w życie każdego z nas. Aplikacja będzie się nazywać GoodMentality.

Celem systemu jest pozytywna zmiana w życiu użytkownika. Aplikacja ma uczyć nowych nawyków oraz pomagać wprowadzić je w życie. Dodatkowo jej celem jest zmiana myślenia użytkownika oraz zwiększenie jego świadomości na tematy związane z odpornością psychiczną.

Nasz system będzie przeznaczony dla szerokiego spektrum użytkowników, w tym dla studentów, pracowników biurowych oraz rodziców. System będzie prosty w obsłudze i intuicyjny, dzięki czemu będzie łatwo dostępny dla użytkowników o różnym poziomie zaawansowania technicznego.

2.5.1. *Kryteria wyboru funkcjonalności*

W niniejszym punkcie zostaną przedstawione kryteria, które zostały wykorzystane do oceny poszczególnych funkcjonalności systemu. Każde kryterium zostało starannie przemyślane i dopracowane, aby zapewnić obiektywną ocenę funkcjonalności i porównanie ich między sobą. Poszczególne funkcjonalności były oceniane liczbą z przedziału od 1 do 5. Sumując oceny z każdego z kryteriów, otrzymywaliśmy wynik dla danej funkcji systemu o maksymalnej wartości 20 pkt. W ramach kryteriów oceny zostały uwzględnione następujące czynniki:

- **Wpływ na odporność psychiczną** – ocenia, jak silny jest wpływ danej funkcjonalności na odporność psychiczną na podstawie wiedzy zebranej w trakcie analizy dziedziny. Punktacja obliczana jest na podstawie tego, jak często dany pomysł pojawiał się na poszczególnych etapach analizy.
- **Spójność z wizją** – subiektywna ocena pokazująca jak bardzo pomysł wpasowuje się w zamysł aplikacji oraz jej cele.
- **Oryginalność** – ocenia jak bardzo pomysł oraz jego konkretna implementacja jest innowacyjna względem podobnych systemów.
- **Łatwość implementacji** – ocenia poziom trudności implementacji danego pomysłu oraz zasoby, jakie potrzebne są na jego realizację, gdzie 1 oznacza trudne a 5 łatwe.

Każde z tych kryteriów zostało opisane i zdefiniowane, aby umożliwić jednoznaczną interpretację i ocenę funkcjonalności. Na podstawie ocen dokonano wyboru najlepszych funkcjonalności (opisane w następnej sekcji), które zostaną włączone do finalnej wersji systemu. Ilość wybranych modułów do zaimplementowania podyktowana została estymatą ich złożoności. Jest to ilość, jaka została uznana za możliwą do implementacji w określonym czasie. Pozostałe funkcjonalności mogą zostać rozpatrzone w kolejnych iteracjach tworzenia systemu. Tabela 2.3 przedstawia punktację, wszystkich rozpatrywanych pomysłów na funkcjonalności.

Tabela 2.3. Ustalane oceny poszczególnych funkcjonalności

Pomysł	Wyływ	Spójność	Oryg.	Impl.	Suma
Planner	5	5	2	3	15
Regulator snu	3.5	4.5	4	4	16
Hasła motywujące	2	4	5	5	16
Dziennik	3.5	4.5	1	4.5	13.5
Baza ćwiczeń fizycznych	2.5	3	2	4	11.5
Dźwięki uspokajające	1	2	4	4	11
Gra relaksacyjna	1	2.5	2	1	6.5

2.5.2. Zbiór funkcjonalności

Planner

Moduł planner'a dotyczy organizacji czasu oraz zadań i składa się z trzech konkretnych funkcjonalności:

- lista celów długoterminowych,
- lista bieżących zadań,
- planowanie następnego dnia / tygodnia.

Moduł ten zgodnie z zasadą *od ogółu do szczegółu* tworzy optymalną przestrzeń ułatwiającą tworzenie nowych celów, rozbijanie ich na mniejsze „podcele”, generowanie na ich podstawie zadań, następnie ich realizację oraz śledzenie postępu.

Praca z modulem zaczyna się od zdefiniowania swojego celu długoterminowego. Użytkownik wypełnia informacje dotyczące nazwy celu. Następnie ma możliwość rozbicia celu na kolejne pomniejsze osiągnięcia i zadania, które to może od razu dodać do listy bieżących zadań. Jest to oddzielny komponent, składający się z tasków zaczerpniętych z planu długoterminowego oraz tych ogólnych, które dodaje na co dzień. Z poziomu zakładki wszystkich zadań, użytkownik ma możliwość dodania wybranych pozycji do listy zadań przeznaczonych na dzisiejszy dzień, oraz na cały tydzień.

Regulator snu

Moduł odpowiedzialny za naprawę lub utrzymanie jakości oraz ilości snu. Składa się on z:

- ogólnych statystyk i informacji obrazujących destrukcyjne skutki zaniedbywania snu,
- możliwości ustalenia harmonogramu i codziennych przypomnień o porze snu,
- informacji o dobrych i prostych praktykach, które znacząco poprawiają jakość snu.

Moduł ten ma przede wszystkim za zadanie uświadomić użytkownikowi, jak destrukcyjne w skutkach są powszechne zaniedbania w codziennym śnie, zarówno pod kątem fizycznym, jak i psychicznym. Realizuje to za pomocą krótkich, lecz treściwych informacji i statystyk. Dodatkowo istnieje możliwość regulacji godzin snu lub pomocy w ich utrzymywaniu, zarówno za pomocą przypomnień o zbliżających się godzinach snu, jak i poradach dotyczących poprawy jego jakości.

Gdy użytkownik otworzy moduł regulatora snu, wyświetli mu się jedna ze statystyk negatywnego oddziaływania zbyt krótkiego spania. Użytkownik może przewijać okienko, czytając kolejne informacje. Użytkownik ma również możliwość podania godziny, o której chodzi spać oraz godziny, o której chciałby chodzić regularnie spać. Jeśli godziny te są rozbieżne, system wyświetli sugerowane godziny snu w kolejnych dniach, które stopniowo będą zbliżały się do tych docelowych. Zgodnie z harmonogramem, aplikacja będzie codziennie przypominać z odpowiednim wyprzedzeniem o nadchodzącej porze snu. Dodatkowo użytkownik może przejrzeć spis prostych porad,

które po wprowadzeniu w życie poprawią jego jakości snu (np. zmniejszenie temperatury pokoju, unikanie niebieskiego światła) wraz z wyjaśnieniem ich działania.

Hasła motywujące

Moduł haseł motywacyjnych stopniowo ma poprawiać nastawienie i samopoczucie użytkownika poprzez codzienne wyświetlanie nowego cytatu motywującego lub inspirującej myśli tych, którzy wiele osiągnęli.

Praca z modułem zaczyna się od wybrania go spośród wszystkich dostępnych. Po wejściu powinien wyświetlić się jeden cytat z bazy wszystkich haseł. Aktualizowanie hasła dostępne będzie raz dziennie. Dziać się to będzie automatycznie każdego dnia. Możliwe jest również wybranie opcji powiadomień. Codziennie rano użytkownik otrzymywać będzie hasło w postaci powiadomienia (nie będzie trzeba wchodzić w aplikację, żeby to zobaczyć). W tym module widoczny będzie również przycisk, dzięki któremu użytkownik będzie mieć możliwość dodania własnego hasła do istniejącej bazy. Zrobione to zostanie na zasadzie prostego formularza z dwoma polami. Jedno odpowiedzialne będzie za wprowadzenie hasła, a drugie, opcjonalne za autora.

Dziennik

Moduł ten składać się będzie z dwóch funkcjonalności:

- wprowadzanie wpisów,
- historia nastrojów.

Użytkownik po wybraniu odpowiedniego modułu w głównym oknie aplikacji, dostępne będzie miał dwie opcje: wprowadzenie wpisu oraz podgląd historii. W przypadku pierwszej z nich, wyświetlony zostanie formularz, który będzie zawierać takie pytania, jak: np. „jak się dzisiaj czujeś?”, „jakie miłe rzeczy cię spotkały?”, oraz wybór dzisiejszego nastroju (zły, średni, dobry). Po wypełnieniu formularza zostanie on wysłany do bazy. W przypadku drugiej opcji wyświetlone zostaną wpisy w postaci rekordów. Będą one w kolorach, odpowiadających wybranemu nastrojowi oraz zawarta będzie na nich data wprowadzenia wpisu. Można będzie również wybrać interesujący użytkownika dzień, aby odczytać szczegóły.

2.6. Refleksje dotyczące analizy merytorycznej

Każdy etap powyższego rozdziału miał duży wpływ na ostateczny rezultat. Dzięki zagłębieniu się w dziedzinę jasnym stało się, co jest najważniejsze w kwestii rozwoju odporności psychicznej, które z powszechnie znanych metod mają uzasadnienie w badaniach, a które z nich są jedynie powszechnie powtarzanymi mitami. Model 4C i spojrzenie przez pryzmat jego działania, pozwoliło kontynuować pracę kreatywną, aby stworzyć odpowiednie funkcjonalności. Zdobyta wiedza pozwoliła przeanalizować podobne systemy, które są dostępne na rynku i z których wiele osób korzysta. Dzięki tej analizie wyklarowało się, jakie funkcje podobnych aplikacji są rzadkie i warte rozwinięcia w tworzonej systemie. Przegląd aplikacji o zbliżonym celu pozwolił również dostrzec ich wady, dzięki czemu wiadome było, co można zrobić lepiej, aby stworzyć optymalny system. Następnie rozpoczął się proces kreatywny, który miał skutkować ustaleniem funkcjonalności opisywanego systemu. Aby ułatwić ten etap, niezwykle przydatne okazały się użyte metody kreatywności. Pozwoliły one z rozmytej, ogólnej wizji systemu przejść do konkretnego, bardziej szczegółowego planu jego realizacji. Za ich pomocą zdefiniowana została lista funkcjonalności,

bazująca na realnych scenariuszach potencjalnych użytkowników oraz ich potrzebach i odczuciach. Metody kreatywności pomogły również wykryć błędne założenia w niektórych aspektach systemu. Przeprowadzenie ankiet wśród osób zainteresowanych tematem rozwoju odporności psychicznej, okazało się dla niezwykle uczącym doświadczeniem. Poznanie opinii i spostrzeżeń innych osób wniosło powiew świeżych pomysłów do pracy. Niezwykle cenne było również zauważenie powtarzalności w odpowiedziach na niektóre pytania. Możliwe było, dzięki temu stwierdzenie co w praktyce sprawia ludziom największy problem, a co za tym idzie, jakie funkcjonalności są najbardziej potrzebne. Co więcej, dzięki pytaniom dotyczącym podobnych systemów, zebrano opinie, na temat wad konkurencyjnych aplikacji i tego, na co warto zwrócić uwagę przy projektowaniu systemu.

3. ASPEKT TECHNICZNY SYSTEMU

Poniższa część pracy skupia się na aspekcie technicznym tworzonego systemu. Dzięki przygotowanej w poprzednim rozdziale wizji aplikacji możliwe jest skupienie się teraz na konkretnej implementacji. Przedstawione zostaną aspekty takie jak wybór technologii i architektura systemu. Opisany zostanie również proces implementacji i testowania.

3.1. *Technologia implementacji (Szymon Hejmanowski)*

W dzisiejszym dynamicznym środowisku technologicznym, tworzenie aplikacji mobilnych stanowi zadanie wymagające przemyślanego podejścia do wyboru odpowiednich technologii. Wraz z różnorodnością dostępnych platform, języków programowania oraz narzędzi deweloperskich, napotykamy szeroki zakres możliwości i wyzwań podczas decydowania, jak zrealizować wizję aplikacji mobilnej. W miarę jak branża ewoluuje, zauważalne są rozbieżności pomiędzy różnymi technologiami, a także ich wpływ na wydajność, dostępność oraz doświadczenia użytkownika. Od natywnych aplikacji po rozwiązania hybrydowe i progresywne, każda opcja posiada swoje unikalne cechy, a dokonany wybór wpływa zarówno na proces deweloperski, jak i finalny produkt.

Główne założenia technologiczne obejmowały:

- **Przenośność** – istotną właściwością tworzonej aplikacji była możliwość budowania stworzonego systemu zarówno na telefony posiadające system Android, jak i te, bazujące na iOS. Obecne technologie pozwalają na budowanie plików wykonywalnych pod różne systemy z tego samego kodu źródłowego. Było to istotne wymaganie wybieranej technologii, ponieważ pozwalało stworzyć aplikację na dwa przewidziane systemy operacyjne, bez potrzeby zgłębiania różnic w ich budowie. W znaczącym stopniu wpływało to na czas realizacji, dzięki czemu możliwe było skupienie się na implementacji konkretnych funkcjonalności.
- **Aktualność** – kolejnym ważnym czynnikiem było to, czy dana technologia używana jest obecnie w branży. Założeniem było zdobywanie wiedzy oraz poznanie mechanizmów wysokopoziomowych technologii, które obecnie najczęściej są wykorzystywane.
- **Dokumentacja** – z powodu zagłębiania się w nowe obszary, warta uwagi była dokumentacja i stopień jej rozwoju. Poznając nowe języki programowania, biblioteki lub narzędzia, często ważniejsze od nich samych, jest sposób, w jaki zostały opisane. Porządna dokumentacja znacząco zmniejsza próg wejścia oraz skraca czas potrzebny do nauki. Dodatkowym aspektem z nią związanym jest społeczność danej technologii. W przypadkach, kiedy społeczność taka jest duża i aktywna, znacznie łatwiej rozwijać aplikacje. Można dzięki niej uzyskać obszerne materiały do nauki, pomoc w konfiguracji środowiska lub znaleźć rozwiązanie pewnego problemu implementacyjnego.
- **Skalowalność** – w przypadku chęci późniejszego rozwoju aplikacji ważnym czynnikiem jest również łatwość, z jaką może się ona skalować. W opisywanej implementacji nie był to kluczowy aspekt, jednak było to coś wartego uwagi. Nie ponosi się dużych kosztów, tworząc od początku oprogramowanie, które łatwo będzie skalować. W przeciwnym jednak razie można narazić się na wiele dodatkowej pracy oraz inne problemy w przyszłości.
- **Kompatybilność** – działanie każdej aplikacji internetowej możemy podzielić na dwie strony. Jedną z nich jest strona prezentacji zawartości użytkownikowi – tzw. *frontend*. Druga zajmu-

je się przetwarzaniem informacji – *backend*. Najczęściej do realizacji części prezentacji po stronie klienta oraz do części logiki po stronie serwera wykorzystywane są dwie różne technologie, które muszą się ze sobą sprawnie komunikować. W celu płynnego procesu pisania kodu źródłowego konieczne było dobranie odpowiednich technologii w taki sposób, aby ich kompatybilność była jak najlepsza.

3.1.1. *Frontend*

Narzędzie, które zostało wybrane do tworzenia interfejsu użytkownika (UI) to **Flutter**, stworzony przez Google. Bazuje on na obiektowym języku Dart, tej samej firmy. Jest to nowa technologia na rynku – obecna od 2018 roku. Flutter jest narzędziem wieloplatformowym, co zapewnia, dzięki możliwości tłumaczenia języka Dart na natywny kod maszynowy dla systemów ARM oraz x86, a zatem w pełni spełnia wymaganie przenośności tworzonej aplikacji. Jest on już dziś używany przez wiele dużych firm, a z każdym rokiem ilość ta się zwiększa. Dokumentacja Fluttera jest niezwykle obszerna i stale rozwijana. Korzystanie z takich środowisk programistycznych jak Visual Studio Code, umożliwia nam dostęp do jego dokumentacji po wybraniu dowolnego implementowanego widżetu, bez konieczności otwierania przeglądarki. Chociaż kryterium skalowalności odnosi się głównie do strony serwerowej, ponieważ to ona musi przetwarzać dane każdego użytkownika, to również omawiany *frontend* można ocenić pod tym kątem. Z racji, że duży napływ użytkowników, jest mocno związany z rozwojem aplikacji, istotny wpływ na skalowalność ma łatwość, z jaką możemy dodawać nowe funkcjonalności do działającej już aplikacji. W tej kwestii Flutter spisuje się bardzo dobrze. Dzięki możliwości tworzenia elementów interfejsu takich jak przyciski, pola tekstowe czy okna kontekstowe w osobnych plikach, możemy je raz przygotować, by następnie sięgać po nie, kiedy to potrzebne. Możliwe jest stworzenie szablonu ekranu, z odpowiednimi marginesami, kolorami tła oraz innymi właściwościami. Tworząc wtedy każdy nowy ekran, załączamy stworzony szablon, a następnie dokładamy na nim potrzebne elementy. Opisana możliwość ponownego wykorzystywania stworzonych widżetów pozytywnie wpływa na skalowalność aplikacji. Kolejnym pozytywem tego *frameworku* jest to, że wszystko, co znajduje się we Flutterze to widżet. Elementy ekranu to kolejne widżety zagnieżdżone w sobie. Wpływa to na łatwość tworzenia skomplikowanych ekranów oraz ich reorganizacji. We Flutterze dostępne są dwie klasy, po których dziedziczy każdy ekran. *StatelessWidget* oraz *StatefulWidget*. W przypadku, gdy dany ekran ma elementy, które zmieniają jego stan podczas korzystania z niego (np. licznik, który wyświetla ile razy kliknęło się w ekran) konieczne jest zastosowanie *StatefulWidget*, do każdego innego przypadku, kiedy ekran wyświetla jedynie statyczne informacje, przeznaczony jest *StatelessWidget*. Inne zalety Fluttera to:

- Możliwość szybkiej weryfikacji zmian wprowadzonych w kodzie, dzięki funkcji HotReload, która umożliwia wprowadzanie zmian w kodzie przy otwartej aplikacji.
- Wysoka wydajność, zbliżona do wydajności oferowanych przez natywne aplikacje [13].

3.1.2. *Backend*

Wybrany narzędziem odpowiedzialnym za obsługę oraz przetwarzanie danych po stronie serwera jest **Firestore**. Jest to narzędzie zapewniające zestaw usług, których działanie jest typowe dla wielu aplikacji. Usługi te to między innymi: mechanizm uwierzytelniania użytkowników, baza danych czy monitorowanie aplikacji. Firestore jest produktem również stworzonym przez Google, co zapewnia doskonałą kompatybilność z wybraną wcześniej technologią *frontendową*. Obszernie udokumentowane jest nie tylko samo narzędzie, ale również to, w jaki sposób współ-

pracuje ono z Flutterem. Dzięki wykorzystywaniu serwerów firmy Google skalowanie aplikacji nie stanowi problemu pod względem technicznym – sprowadza się do wykupienia dodatkowych zasobów zgodnie z potrzebami. Podczas współpracy z Flutterem korzystamy z tych samych funkcji do komunikacji z serwerem niezależnie od tego, na jakie urządzenie trafi pisany kod, dzięki czemu rozwiązanie to zapewnia pełną przenośność po obu stronach.

3.2. Architektura systemu (Szymon Hejmanowski)

Na obecnym etapie, stworzone są wszystkie założenia teoretyczne oraz zebrane są funkcjonalności, jakie implementować ma tworzony system. W celu praktycznej realizacji potrzebne jest stworzenie architektury systemu, obejmującej schemat poszczególnych elementów systemu, ich wzajemnych zależności oraz komunikacji.

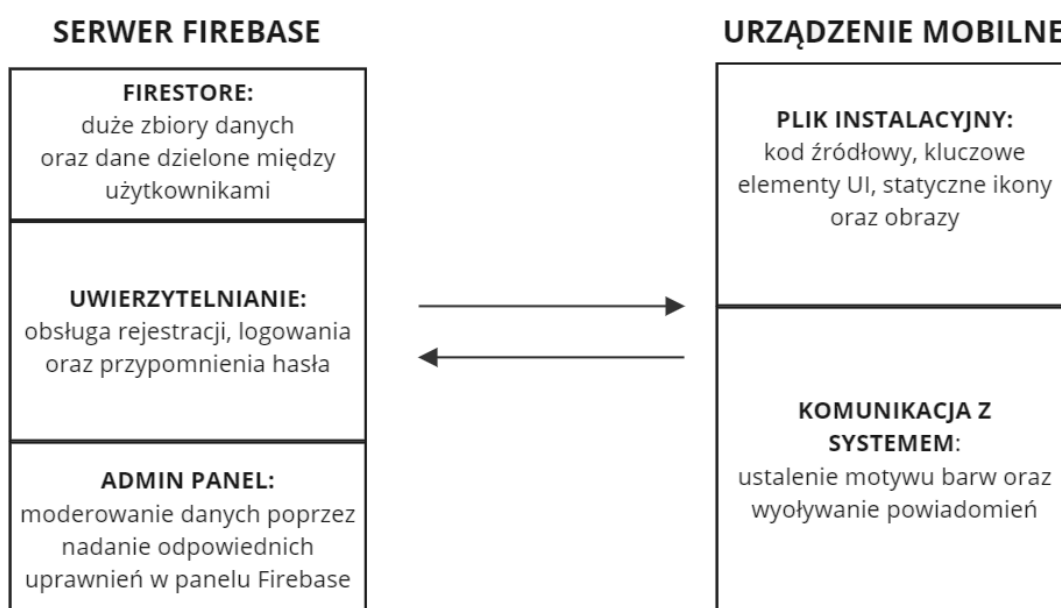
3.2.1. Elementy systemu

W przypadku tworzonej aplikacji system można podzielić na dwa główne elementy: (zawarte również na rysunku 3.1):

- **Urządzenie mobilne użytkownika** – na nie pobierany jest plik ze skompilowanym kodem źródłowym aplikacji (.apk lub .ipa, odpowiednio dla systemu Android oraz iOS). Zawiera on wszystkie statyczne elementy aplikacji, takie jak ikony modułów, tekst dotyczący nawigacji w aplikacji oraz wspomniany kod źródłowy. Dodatkowo niektóre dane użytkownika mogą być zapisywane bezpośrednio w lokalnej pamięci, jeśli chcemy ich używać w kontekście tego konkretnego urządzenia, w tworzonej aplikacji nie było jednak potrzeby korzystania z takiego magazynu lokalnego. Aplikacja komunikuje się również z systemem urządzenia użytkownika w celu dopasowania motywu barw interfejsu – jasny lub ciemny w zależności od ustawień systemowych – oraz w celu wywoływania powiadomień związanych z modułem sennika.
- **Serwer Firebase** – ten element systemu odpowiedzialny jest za wiele zadań. Usługa „Uwierzytelnianie” pozwala na wprowadzanie do aplikacji mechanizmu tworzenia konta użytkownika. W opisywanej aplikacji tworzenie konta możliwe jest za pomocą adresu e-mail. Osoba korzystająca z systemu musi się zarejestrować przy pierwszym użyciu, a następnie wszystkie wprowadzane dane do poszczególnych modułów, powiązane są z jej kontem i dostępne jedynie po zalogowaniu. Możliwe jest również przypomnienie hasła. W tym przypadku użytkownik otrzymuje na swoją skrzynkę pocztową link do formularza, gdzie może wprowadzić nowe hasło. Kolejną usługą serwera Firebase, która jest używana to „Firestore Database”. Jest to nierzelacyjna baza danych, która w przypadku małego skomplikowania powiązań między konkretnymi zbiorami danych jest optymalnym rozwiązaniem. Jest ona najlepszym sposobem, aby przechowywać duże ilości danych, które w przeciwnym wypadku musiałyby obciążać pamięć urządzenia mobilnego użytkownika. Dodatkowo, zewnętrzna baza danych na serwerze jest obligatoryjna w przypadku zbiorów danych dzielonych między użytkownikami. W tworzonym systemie taki zbiór danych to cytaty, które może wprowadzić dowolny użytkownik, a inni mogą ów cytat wyświetlić. Kolejnym elementem systemu możliwym do implementacji jest panel admina, dzięki któremu możliwe byłoby moderowanie wprowadzanych cytatów. W opisywanym przypadku nie ma jednak potrzeby jego implementacji, ponieważ moderowanie danych możliwe jest poprzez panel Firebase, wykorzystując opcję przypisywania uprawnień.

3.2.2. Komunikacja

Istotnym elementem architektury jest komunikacja między serwerem a użytkownikiem. W przypadku korzystania z Firebase + Flutter, dzięki bardzo dobrej integracji tych technologii, implementacja jest stosunkowo prosta i polega na wykorzystywaniu funkcji dostępnych w bibliotekach importowanych do projektu we Flutterze. Większość funkcji, przez które były przesyłane dane, zostały zaimplementowane w trybie asynchronicznym. Pozwala to na brak opóźnień w działaniu interfejsu graficznego użytkownika, ponieważ program nie czeka na otrzymanie danych od serwera, tylko kontynuuje swoje działanie, a pobieranie następuje w tle. Co więcej, dzięki obiektom typu Future, możemy wyświetlać przykładową zawartość strony, dopóki nie otrzymamy danych z serwera. Przykładowy fragment kodu ukazujący asynchroniczną funkcję odpytującą serwer o dane znajduje się na rysunku 3.5 w dalszej części pracy.



Rys. 3.1. Architektura systemu wraz z funkcjami oraz zawartością poszczególnych elementów

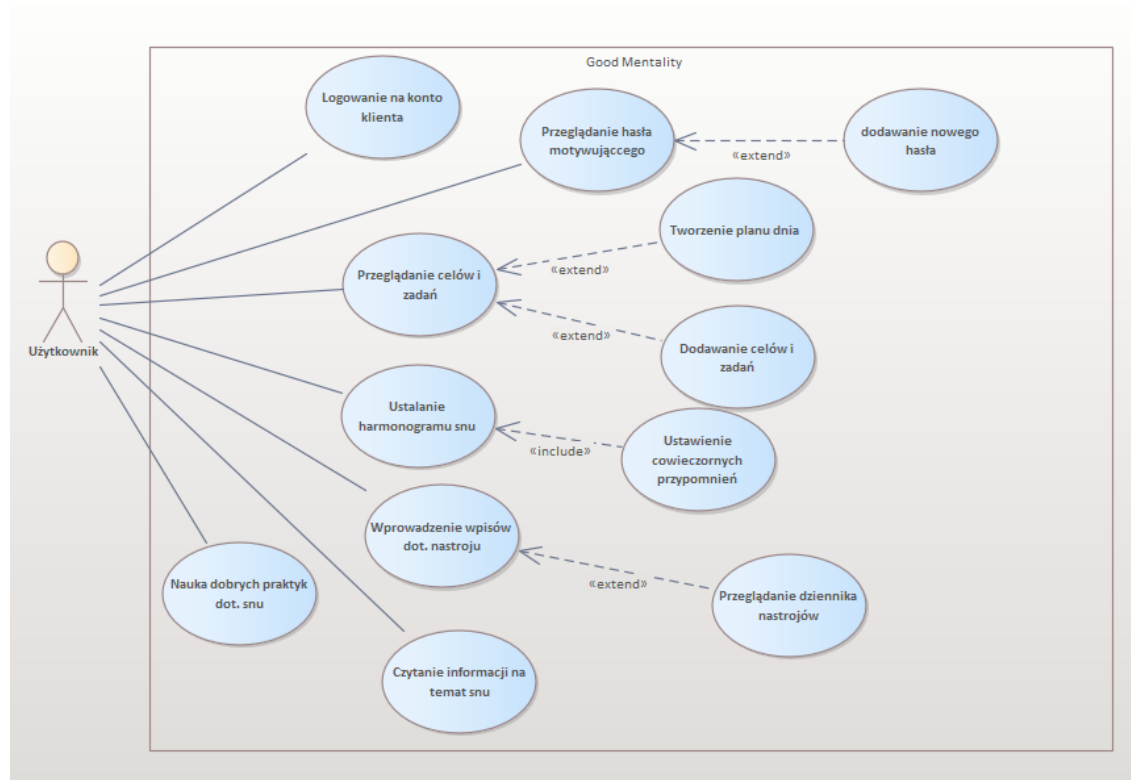
3.3. Przypadki użycia aplikacji (Patryk Gołembiewski)

Podczas tworzenia systemu, ważną częścią jest stworzenie schematu przypadków użycia. Pozwala on modelować zachowanie systemu. Celem takiego diagramu nie jest pokazanie jak wewnątrz działa aplikacja, jak wygląda architektura, lecz opisanie funkcji wysokiego poziomu, interakcji użytkownika. Pomaga to wychwycić również wymagania danego systemu [14]. Niniejsza część skupia się na prezentacji przypadków użycia aplikacji Good Mentality. Rozważane są różne scenariusze, które ukazują proste zadania oraz bardziej rozbudowane sytuacje. Ten rozdział stanowi przewodnik po możliwościach oferowanych przez aplikację.

Do tworzenia diagramów wykorzystano produkt Sparx Systems o nazwie Enterprise Architect. Jest to kompleksowe narzędzie do analizy i projektowania modelu systemu za pomocą UML (*Unified Modeling Language*). Umożliwia on rozwój oprogramowania już od gromadzenia wymagań, poprzez analizę i tworzenie modeli projektowych, aż do testowania i utrzymywania. EA to graficzne narzędzie wspomagające projektowanie dobrego i łatwego w utrzymaniu opro-

gramowania. Dokumentacja jest szczegółowo opisana, co ułatwia użytkowanie początkującym projektantom [15].

W przypadku pracy nad aplikacją Good Mentality projektowanie UML zawężono do skupienia się szczególnie na diagramach przypadków użycia, które są doskonałym narzędziem do opisu funkcji systemu z perspektywy użytkownika. Na rysunkach 3.2 i 3.3 ukazano schemat interakcji użycia systemu.

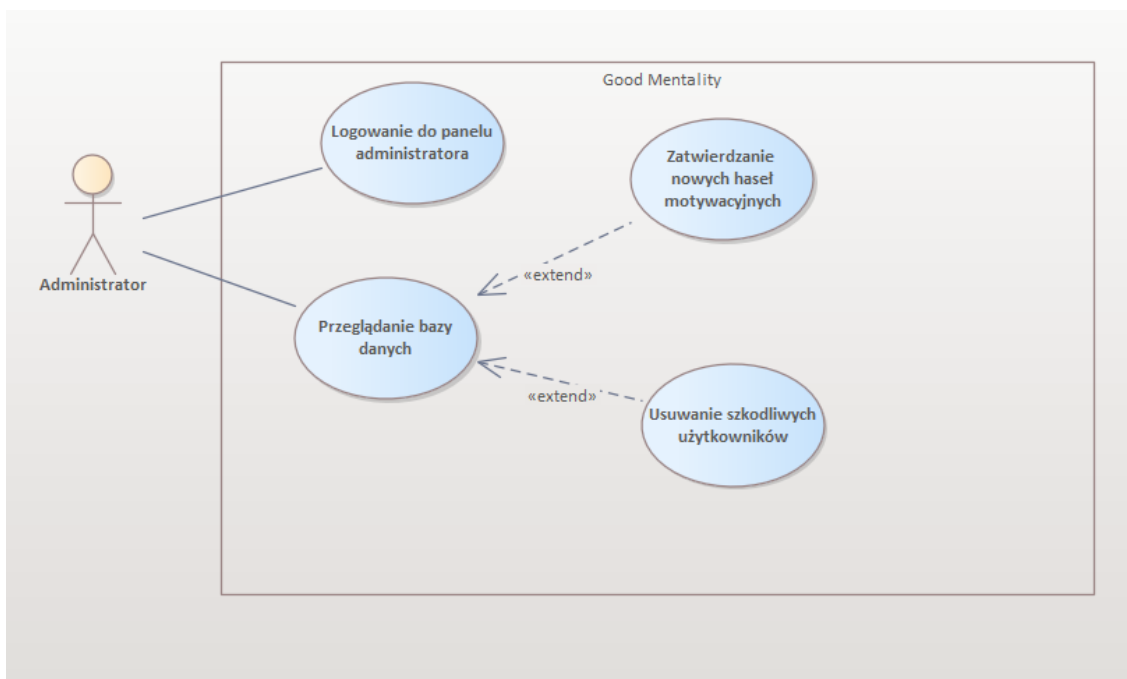


Rys. 3.2. Diagram przypadków użycia użytkownika

Patrząc na diagram, można zauważyć, w jaki sposób użytkownik może wejść w interakcję z projektowanym systemem. Pierwszym kluczowym elementem jest logowanie się na konto. Ma on również możliwość korzystania z wcześniej opisanych funkcjonalności, które mają pomóc mu wzmocnić odporność psychiczną. Jednak w przypadku kilku, podczas przeglądania, istnieje opcja rozszerzenia interakcji o kolejny przypadek użycia, na przykład, kiedy korzysta z planera, może dodać kolejny wpis.

Pomimo tego, że diagram przypadków użycia uznaje się za użyteczny sposób wizualizacji relacji pomiędzy aktorami a systemem, to z racji swojej graficznej natury nie jest w stanie ukazać wszystkich szczegółów komunikacji. Klarowność i zwięzłość takiego schematu jest istotna, jednak nie uwzględnia ona wszystkich detali, jakie mogą być potrzebne w kontekście danego przypadku użycia. W celu uzyskania dokładniejszych informacji potrzebne jest utworzenie opisów. Jest to tekstowe przedstawienie poszczególnych interakcji między użytkownikiem a systemem. W tabelach 3.1, 3.2 oraz 3.3 przedstawiono wybrane opisy przypadków użycia, które wzbogacają zrozumienie funkcji systemu, prezentując szczegółowe narracje oraz kluczowe etapy interakcji.

Analiza diagramu przypadków użycia i ich opisów przyniosła szereg wniosków kluczowych do zrozumienia i późniejszego tworzenia funkcjonalności. Dzięki temu udało się uchwycić role i obowiązki poszczególnych aktorów w systemie. W sprawie obsługi błędów, a także sytuacji wyjątkowych, zdecydowanie lepiej i dokładniej udało się zidentyfikować potencjalne problemy, jakie



Rys. 3.3. Diagram przypadków użycia administratora

Logowanie na konto klienta	
Warunki początkowe:	Brak
Przebieg	<ol style="list-style-type: none"> 1. System udostępnia możliwość wprowadzenia loginu oraz hasła. 2. Użytkownik wprowadza login oraz hasło. 3. System informuje klienta o poprawnym zalogowaniu (wyjątek: login lub hasło niepoprawne).
Przebiegi alternatywne:	<p>Wyjątek – login lub hasło niepoprawne:</p> <ol style="list-style-type: none"> 3a. System informuje klienta o niepoprawności danych logowania oraz daje możliwość ponownego wprowadzenia danych 3b. Kontynuacja pkt 3
Warunki końcowe:	Klient zalogowany

Tabela 3.1. Przypadek użycia – Logowanie

Przeglądanie hasła motywującego	
Warunki początkowe:	Użytkownik zalogowany
Przebieg	<ol style="list-style-type: none"> 1. System udostępnia możliwość przeglądania haseł motywujących. 2. Użytkownik klika przycisk wylosowania nowego hasła. 3. System wyświetla nowo wylosowane hasło.
Przebiegi alternatywne:	brak
Warunki końcowe:	brak

Tabela 3.2. Przypadek użycia – Przeglądanie hasła motywacyjnego

Dodanie nowego hasła motywacyjnego	
Warunki początkowe:	Użytkownik zalogowany
Przebieg	<ol style="list-style-type: none"> 1. Użytkownik wybiera przycisk dodania hasła. 2. System wyświetla formularz zgłoszenia nowego hasła. 3. Użytkownik wpisuje wymagane dane i klika przycisk wyślij.
Przebiegi alternatywne:	brak
Warunki końcowe:	Hasło zapisane w bazie

Tabela 3.3. Przypadek użycia – Dodanie nowego hasła

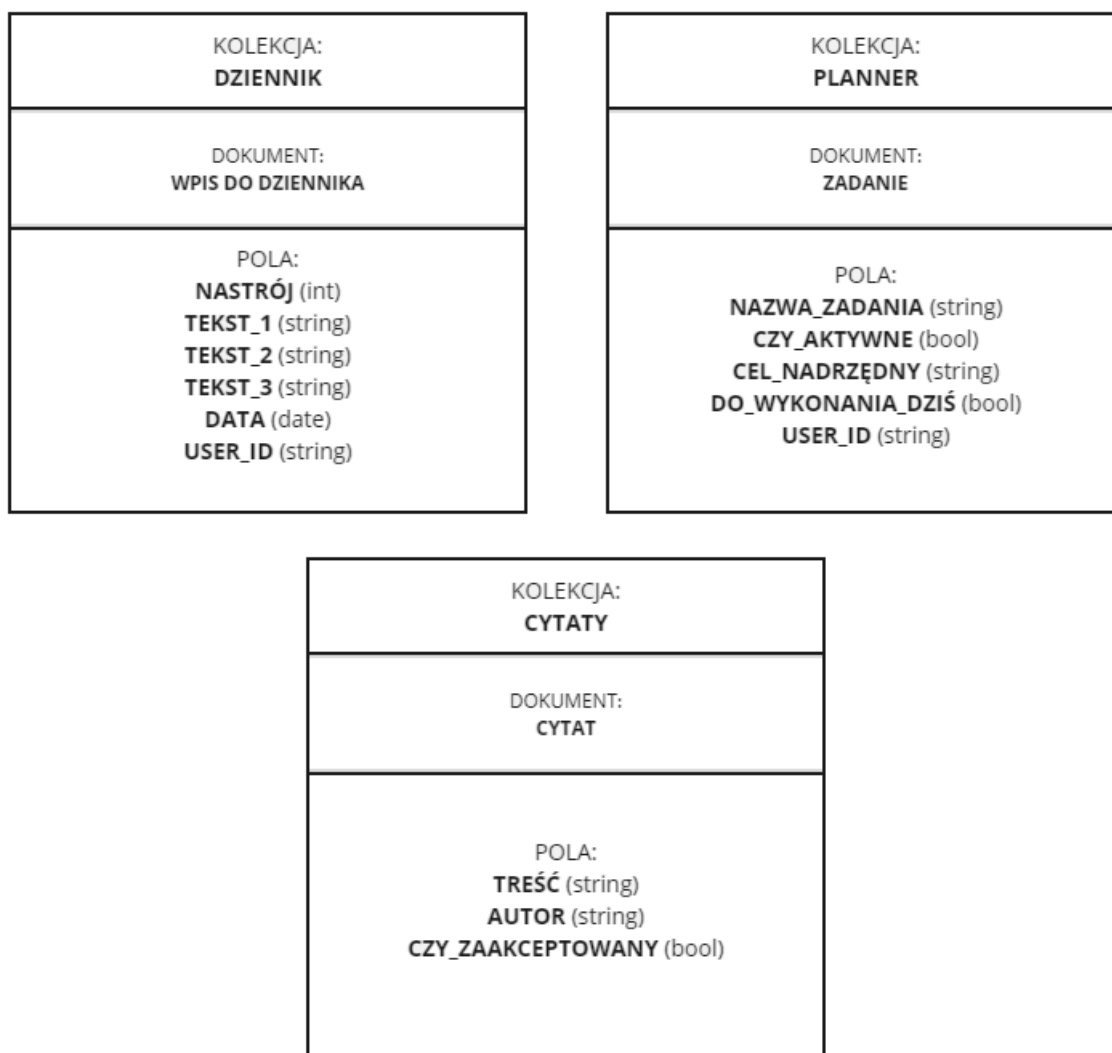
mogą czekać na użytkowników podczas korzystania z aplikacji. Pozwala to wymyślić i później wdrożyć specjalne mechanizmy, które obsługują niewygodne sytuacje. Wnioski te stanowią cenne wytyczne dla kolejnych etapów projektowania systemu.

3.4. Schemat bazy danych (Patryk Gołembiewski)

Baza danych oraz jej architektura jest kluczowa dla prędkości działania aplikacji, łatwości w rozwijaniu oraz skalowaniu. Istnieją różne rodzaje baz danych, jednak możemy podzielić je na dwa główne typy: relacyjne oraz nierelacyjne. To który typ wybrać, zależy od konkretnego systemu, jego architektury oraz celu.

Wykorzystywany Firebase oferuje jednak jedynie dwa typy nierelacyjnych baz danych. Jedną z nich o nazwie *Firebase Realtime Database*, przechowuje dane w formacie JSON (*JavaScript Object Notation*), natomiast druga - *Firestore Database* – jest bazą dokumentową. Pierwsza z nich, sugerowana jest do obsługi prostych danych, łatwych do przechowywania, wtedy gdy zależy nam na jak najmniejszych opóźnieniach. *Firestore Database* jest z kolei bardziej uniwersalnym rozwiązaniem, w którym niewymagane jest spłaszczenie ani denormalizacja danych. Pozwala ona również na lepszą organizację przechowywanych informacji. Ze względu na te czynniki, baza typu dokumentowego *Firestore Database* została uznana za optymalną w przypadku tworzonej aplikacji.

Wybrany typ bazy składa się z dokumentów, które posiadają pola. Wiele dokumentów, które połączone są logiczną zależnością, tworzy kolekcję. Przykładem takich zależności może być kolekcja, zawierająca użytkowników. Każdy użytkownik będzie stanowił jeden dokument, którego pola, będą danymi związanymi z tym użytkownikiem (np. jego nazwa oraz wiek). Ze względu na specyfikę takiej bazy danych oraz różnice względem bazy relacyjnej okazała się ona optymalnym wyborem. Brak stałego schematu danych i idąca za tym elastyczność pozwala na większą wolność w przypadku tworzenia systemu. Dodawanie kolejnych funkcjonalności, do których działania konieczne jest wprowadzenie nowych danych do bazy, nie tworzy konfliktów z już istniejącymi. W takim przypadku wystarczy jedynie stworzyć nową kolekcję danych. Szybkie obsługiwanie zapytań również, nie jest bez znaczenia w przypadku aplikacji mobilnej. Użytkownik oczekuje, że wszystko będzie działać płynnie, a opóźnienie powinno być niezauważalne. Fakt, że taka baza nie działa na podstawie języku zapytań SQL (*Structured Query Language*), nie stanowi problemu, ponieważ API (*Application Programming Interface*) Firebase'a udostępnia funkcje, dzięki którym zapis i odczyt konkretnych danych nie stanowi problemu. Rysunek 3.5 przedstawia przykładową składnię funkcji odpytującą serwer o dane.



Rys. 3.4. Schemat bazy danych (nazwy pól przetłumaczone na język polski)

```
Future<List<String>> getRandomQuote() async {
  QuerySnapshot querySnapshot = await FirebaseFirestore.instance
    .collection('Quotes')
    .where('quoteAccepted', isEqualTo: '1')
    .get();
}
```

Rys. 3.5. Fragment kodu aplikacji przedstawiający zapytanie serwera o dane

Rysunek 3.4 przedstawia zbiór kolekcji oraz opis pól wykorzystywanej dokumentowej bazy danych. Należy zauważyć, że nie zawiera on kolekcji z danymi użytkowników. Wynika to z faktu, że używając wspomnianej usługi uwierzytelniania Firebase, dane o użytkownikach zapisywane są niezależnie od bazy danych w innym miejscu. Ze względu na to, nie ma konieczności tworzenia specjalnej kolekcji w bazie danych.

3.5. Projektowanie interfejsu użytkownika (Szymon Hejmanowski)

W przypadku złożonych aplikacji, składających się z dużej ilości ekranów, niezwykle trudnym zadaniem jest ustalanie tego, jak ma wyglądać dany ekran w trakcie jego implementacji. Gdy tworzona jest mała aplikacja, która zawiera parę ekranów, jest to możliwe, ale nie jest to do-

brą praktyką. Aby odseparować część tworzenia wizualnej koncepcji, od części implementacyjnej stosuje się makiety graficzne (ang. *mockup*). Makiety, są to zwykłe pliki graficzne zawierające jedynie wizualizacje poszczególnych ekranów. Rysunek 3.6 zawiera dwie grafiki stworzone przed przystąpieniem do jakiegokolwiek implementacji. Zostały one przygotowane za pomocą specjalnego narzędzia do tworzenia tego typu wizualizacji o nazwie Uizard.



(a) Ekran główny modułu

(b) Przegląd ostatnich wpisów

Rys. 3.6. Makiety graficzne modułu „Dziennik”

3.6. Implementacja i testowanie (Patryk Gołembiewski)

Kolejnym etapem tworzenia systemu była faktyczna implementacja, na podstawie wizji aplikacji oraz założeń technicznych. Istotnym jej elementem był podział zadań między deweloperami oraz stworzenie środowiska pozwalającego na współpracę. Jak w każdym projekcie, podczas implementacji pojawiły się problemy oraz wyzwania, które trzeba było rozwiązać.

3.6.1. Praca w zespole

Praca implementacyjna została podzielona między dwóch członków zespołu w taki sposób, aby każdy uzyskał podobną estymatę złożoności swoich zadań. Patryk Gołembiewski odpowiedzialny był za moduły: planner oraz dzienniki, oba o wyestymowanej wysokiej złożoności. Szymon Hejmanowski implementował: mechanizm logowania i rejestracji, o złożoności wysokiej, oraz dwa moduły o niskiej złożoności, czyli sennik i cytaty.

Aby możliwa była równoległa implementacja kolejnych funkcji, używany był system kontroli wersji Git. Dzięki stworzonemu repozytorium na serwisie Github możliwa była praca obu członków zespołu na wspólnej bazie kodu źródłowego. Przyjęta została konwencja jednej głównej gałęzi

o nazwie *main*. W przypadku dodawania nowej funkcjonalności przez któregoś z członków zespołu tworzona była specjalna gałąź o odpowiedniej nazwie. Gdy praca nad tym fragmentem kodu była skończona, osoba implementująca tworzyła PR (*Pull Request*). Dzięki temu drugi członek zespołu miał możliwość weryfikacji zmian wprowadzonych w bazie kodu źródłowego. W przypadku gdy jakaś część była według niego niepoprawna, wystawiał on komentarz pod liniijką, której uwaga dotyczyła. Ostatecznie, gdy wszystkie poprawki zostały wprowadzone, mógł on dać akceptację, co pozwalało na wcielenie nowej funkcjonalności do gałęzi głównej.

3.6.2. Implementacja

W niniejszym podrozdziale opisane zostały ważniejsze aspekty pisanej aplikacji, podzielone na części związane z danym zagadnieniem.

Uwierzytelnianie

Cały proces implementacji rozpoczęto od pierwszego ekranu, jaki spotyka użytkownika – panelu logowania. W każdej aplikacji, która udostępnia taką funkcjonalność, konieczne jest dodanie możliwości utworzenia nowego konta dla tych, którzy pierwszy raz mają styczność z systemem oraz opcji przywracania hasła. W tym celu stworzono jeden główny komponent, który domyślnie ustawiony jest na wywołanie widżetu zarządzającego logowaniem i rejestracją lub strony domowej w zależności od tego, czy dany użytkownik jest zalogowany. Stworzony został obserwator, który wraz ze zmianą stanu uwierzytelnienia, wywołuje odpowiednie komponenty. Metoda *authStateChanges()* zwraca strumień, który emituje odpowiednie zdarzenie, pozwalając na reakcję na zmiany w czasie rzeczywistym (kod przedstawiono na rys. 3.7). Jeżeli użytkownik pomyślnie się zaloguje, przekierowany zostaje do właściwej treści aplikacji.

```
class AuthPage extends StatelessWidget {
  const AuthPage({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: StreamBuilder(
        stream: FirebaseAuth.instance.authStateChanges(),
        builder: (context, snapshot) {
          if (snapshot.hasData) {
            return const Home();
          }
          return const LoginOrRegister();
        },
      ),
    );
  }
}
```

Rys. 3.7. Fragment kodu komponentu wywołującego widżet logowania i rejestracji lub widoku domowego

Widżet, który decyduje czy aktualnie wyświetlić stronę logowania, czy stronę rejestracji działa w nieco inny sposób. Domyślnie ukazana zostaje możliwość podania e-mailu oraz hasła w celu przejścia do aplikacji. Tutaj w przypadku zmian, kluczowa jest flaga *showLoginPage*. Od niej zale-

ży, co zostanie pokazane użytkownikowi. Jeżeli osoba kliknie opcję tworzenia konta, flaga zmienia się i zostanie przekazana do części wybierającej. W ten sposób sprawdzając wartość, możliwa jest zmiana widżetu na taki, który zawiera formularz udostępniający tworzenie konta. Proces ten ukazany został na rysunku 3.8 oraz 3.9. W przypadku, gdy użytkownik naciśnie przycisk przejścia do rejestracji, wykonywana jest funkcja *togglePages()*, udostępniana przez komponent zarządzający, która zmienia stan widżetu, dzięki czemu odświeżana jest zawartość. W tym przypadku jest to panel służący tworzeniu nowego konta. Analogiczne wykonano zostało to w przypadku widoku rejestracji.

```
class _LoginOrRegisterState extends State<LoginOrRegister> {
  bool showLoginPage = true;

  void togglePages() {
    setState(() {
      showLoginPage = !showLoginPage;
    });
  }

  @override
  Widget build(BuildContext context) {
    if (showLoginPage) {
      return LoginPage(onTap: togglePages);
    }
    return RegisterPage(onTap: togglePages);
  }
}
```

Rys. 3.8. Fragment kodu komponentu zarządzającego wyświetlaną treścią

```
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    const Text("Don't have an account?"),
    GestureDetector(
      onTap: widget.onTap,
      child: Text(
        " Register here",
        style: TextStyle(
          fontWeight: FontWeight.bold,
          color: Theme.of(context).colorScheme.secondary,
        ),
      ),
    ],
  ),
)
```

Rys. 3.9. Fragment kodu widżetu logowanie umożliwiający zmianę widoku

Nawigacja

Strona domowa aplikacji zawiera kilka przycisków umożliwiających dostęp do właściwych treści. Kiedy użytkownik przejdzie do którejś z nich, pojawia się pewien schemat występujący w każdym innym module. Osiągnięte to zostało za pomocą stworzenia głównego widżetu, który zawiera szkielet wyglądu. Oznacza to, że górny panel oraz pasek nawigacyjny zostały określone w tym komponencie (kod pokazano na rys. 3.10 i 3.11). Aby ukazać treść poszczególnego modułu, w sekcji ciała wywoływany jest odpowiedni widżet (rys. 3.12). Za pomocą tego uzyskano ujednolicenie wyglądu oraz płynniejsze przechodzenie pomiędzy modułami. Nie jest konieczne załadowanie całej strony, jedynie zmieniany jest stan widżetu głównego.

```
bottomNavigationBar: Container(  
  color: Theme.of(context).colorScheme.primary,  
  child: Padding(  
    padding: const EdgeInsets.symmetric(  
      vertical: 3.0,  
      horizontal: 20,  
    ),  
    child: GNav(  
      selectedIndex: _selectedIndex,  
      iconSize: 20,  
      gap: 5,  
      onTabChange: _navigateBottomBar,  
      color: Theme.of(context).colorScheme.inversePrimary,  
      activeColor: Theme.of(context).colorScheme.secondary,  
      backgroundColor: Theme.of(context).colorScheme.primary,  
      padding: const EdgeInsets.all(18),  
      tabs: const [  
        GButton(  
          icon: CupertinoIcons.home,  
        ),  
        GButton(  
          icon: CupertinoIcons.moon,  
        ),  
        GButton(  
          icon: CupertinoIcons.quote_bubble,  
        ),  
        GButton(  
          icon: CupertinoIcons.list_bullet,  
        ),  
        GButton(  
          icon: CupertinoIcons.doc_plaintext,  
        ),  
      ],  
    ),  
  ),  
)
```

Rys. 3.10. Fragment kodu paska nawigacyjnego.

```

int _selectedIndex;

_HomePageState(this._selectedIndex);
void _navigateBottomBar(int index) {
  setState(() {
    _selectedIndex = index;
    if (index == 0) {
      Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => Home(),
        ),
      );
    }
  });
}

```

Rys. 3.11. Funkcja odpowiedzialna za zmianę indeksu służącego do wyświetlenia odpowiedniego modułu.

```

appBar: AppBar(
  iconTheme: IconThemeData(
    color: Theme.of(context).colorScheme.inversePrimary,
  ),
  title: Text(
    _pagesNames[_selectedIndex],
    style: TextStyle(color: Theme.of(context).colorScheme.inversePrimary),
  ),
  centerTitle: true,
  elevation: 0,
  actions: [
    IconButton(
      onPressed: logout,
      icon: Icon(
        Icons.logout_rounded,
        color: Theme.of(context).colorScheme.inversePrimary,
      ))
  ],
),
body: _pages[_selectedIndex],

```

Rys. 3.12. Fragment kodu widżetu, w który poprzez zmianę zawartości ciała pokazuje wybrany moduł.

Powiadomienia

W przypadku modułu sennik, jedną z dostępnych funkcji jest regulacja pory snu. Użytkownik może wybrać godzinę, o której aktualnie chodzi do łóżka oraz godzinę docelową. W ten sposób aplikacja za pomocą obliczeń jest w stanie ułożyć plan na kilka kolejnych dni, skracając porę o 10 minut. Każdorazowo na 30 minut przed zaplanowanym czasem spania, aplikacja wysyła powiadomienie, dzięki czemu osoba przypomni sobie i będzie w stanie przygotować się do snu. W kodzie wykorzystywane są funkcje systemowe dostarczane przez pakiet *flutter_local_notifications*. Aby można było ustawić powiadomienia, konieczna jest inicjalizacja ustawień dla systemu Android i iOS oraz całej wtyczki *flutterLocalNotificationsPlugin* [16]. Następnie definiuje się szczegóły takie jak nazwę kanału i priorytet. Na końcu następuje faktyczne planowanie powiadomień.

3.6.3. Testowanie

Ocena sprawności funkcjonowania aplikacji jest niezwykle ważna przed udostępnieniem jej użytkownikom. Dzięki temu można odnaleźć niedociągnięcia, problemy, czy przypadki brzegowe aktualnie nieobsługiwane i powodujące błędy. Istnieje wiele podejść do przeprowadzania testów, a wybór pomiędzy manualnym sprawdzeniem a automatycznym często stanowi podmiot rozważań w trakcie tworzenia oprogramowania.

W kontekście aplikacji pisanej w szkieletcie Flutter, gdzie tempo iteracji i responsywność interfejsu użytkownika jest duża, postawiono na testowanie manualne. Jednym z kluczowych czynników, który zaważył na wyborze, jest elastyczność weryfikacji ręcznej. Można dostosowywać swoje podejście, podczas przeprowadzania prób w zależności co chce się uzyskać. Raz przyjmuje się podejście osoby, która pierwszy raz korzysta z aplikacji tego typu, a raz osoby zorientowanej w tej dziedzinie. To szczególnie ważne w przypadku projektów, gdzie trudno przewidzieć wszystkie możliwe ścieżki, którymi będzie podążać użytkownik.

Na podstawie testów ustalono również stopień, w jakim zostały zaimplementowane zakładane funkcjonalności systemu. Przedstawia to tabela 3.4. Pominięcie niektórych funkcji modułów wynikało z ograniczeń czasowych.

Tabela 3.4. Stopień realizacji poszczególnych modułów

Moduł	Stopień realizacji względem wizji systemu	Pominięte funkcje
Planner	85%	Brak możliwości planowania całego tygodnia
Regulator snu	100%	-
Hasła motywujące	90%	Brak możliwości ustawienia powiadomień
Dziennik	100%	-

3.6.4. Problemy w trakcie implementacji

Pisanie oprogramowania wiąże się z wieloma trudnymi sytuacjami, nad którymi trzeba dłużej pomyśleć lub spojrzeć z szerszej perspektywy, aby znaleźć rozwiązanie. W trakcie tworzenia aplikacji natknięto się na dwa szczególnie istotne problemy. W niniejszej części zostaną one przedstawione wraz z ich rozwiązaniem.

Problem z opóźnionym wyświetlaniem się cytatów

Pierwszym momentem zacięcia było wyświetlanie się haseł motywacyjnych. Kiedy użytkownik wszedł do modułu związanego z tym zagadnieniem, na początku nie pojawiał się żaden cytat.

Zauważono jednak, że kiedy kliknięto przycisk, który odpowiedzialny był za losowanie nowego hasła, to ukazywało się ono. Zagłębiając się w ten problem, odkryto komplikacje związane z przekazywaniem danych z bazy. Nie wykorzystując typu *Future* do przechowywania danych, strona budowała się z pustą zmienną typu *String*. Po wykonaniu tej zmiany i dostosowaniu kodu od razu zauważono poprawę (kod pokazano na rys. 3.13 i rys. 3.14). W celu poprawnego działania programu, z użyciem zmiennej typu *Future*, konieczne było zastosowanie *FutureBuilder*, czyli widżetu, który buduje się na podstawie najnowszej migawki interakcji z *Future* [17] (kod pokazano na rys. 3.14). Można ustawić, co ma się wydarzyć do momentu przyjscia danych, w tym przypadku jest to okrągły wskaźnik postępu. Kiedy informacje z bazy zostaną dostarczone, wyświetli się wynik zapytania. W ten sposób udało się uniknąć problemu z wyświetlaniem się widżetu bez tekstu.

```
late Future<List<String>> futureQuote;

@override
void initState() {
  super.initState();
  futureQuote = getRandomQuote();
}
```

Rys. 3.13. Fragment kodu przedstawiający inicjalizację stanu widżetu z wywołaniem funkcji pobierającej dane z bazy.

```
Future<List<String>> getRandomQuote() async {
  QuerySnapshot querySnapshot = await FirebaseFirestore.instance
    .collection('Quotes')
    .where('quoteAccepted', isEqualTo: '1')
    .get();

  var random = Random();
  var randomDocument =
    querySnapshot.docs.elementAt(random.nextInt(querySnapshot.docs.length));

  return [randomDocument['quote'], randomDocument['quoteAuthor']];
}
```

Rys. 3.14. Fragment kodu przedstawiający funkcję pobierającą dane z bazy.

Problem z ustawieniem powiadomień

Kolejnym problemem, z którym przyszło się mierzyć, było planowanie i wywoływanie powiadomień. Korzystając z wielu internetowych instruktaży, za każdym razem nie udawało się uzyskać zamierzonego celu. Podczas testowania jedynie możliwe było wyświetlenie natychmiastowego powiadomienia (pojawiało się zaraz po naciśnięciu przycisku). Dużym błędem było korzystanie w pierwszej kolejności z instrukcji zamieszczonych w internecie oraz rozwiązań tego problemu na forach. Dopiero dokładna analiza dokumentacji technicznej Fluttera przyniosła oczekiwany rezultat. Poza samym kodem pisanym w języku Dart, konieczne było dodanie kilku linii w generowanych plikach konfiguracyjnych aplikacji na dany system.

```

FutureBuilder<List<String>>(
  future: futureQuote,
  builder: (context, snapshot) {
    if (snapshot.connectionState == ConnectionState.waiting) {
      return const CircularProgressIndicator();
    } else if (snapshot.hasError) {
      return Text('Error: ${snapshot.error}');
    } else {
      return Column(
        children: [
          Container(
            padding: const EdgeInsets.all(20),
            decoration: BoxDecoration(
              color: Theme.of(context).colorScheme.primary,
              borderRadius: BorderRadius.circular(20),
            ),
            child: Text(
              snapshot.data![0],
              style: const TextStyle(
                fontSize: 35,
              ),
            ),
          ),
          const SizedBox(height: 20),
          Container(
            padding: const EdgeInsets.all(20),
            decoration: BoxDecoration(
              color: Theme.of(context).colorScheme.primary,
              borderRadius: BorderRadius.circular(20),
            ),
            child: Text(
              "Autor: ${snapshot.data![1]}",
              style: const TextStyle(
                fontSize: 15,
                fontStyle: FontStyle.italic, // Dodaj kursywę
              ),
            ),
          ),
        ],
      ),
    );
  },
);

```

Rys. 3.15. Fragment kodu ukazujący zastosowanie FutureBuildera.

3.7. Refleksje dotyczące analizy technicznej

Techniczna analiza była drugim kamieniem milowym podczas dążenia do celu projektu. Etap ten zaczął się od przygotowania technicznego. Poznanie technologii oraz wybór odpowiedniej z nich był ważnym czynnikiem – rzutował on, na przebieg całej implementacji. Gdy znane były już narzędzia, jakie miały pomagać w procesie tworzenia aplikacji, ważnym krokiem było stworzenie odpowiedniej architektury systemu. Chociaż w tym przypadku nie była ona nadzwyczaj skomplikowana, to poświęcony czas na jej zaprojektowanie był istotny, ponieważ jednoznacznie określiła ona elementy systemu oraz ich wspólną komunikację. Stworzone przypadki użycia pozytywnie wpłynęły na proces implementacji, ponieważ wyklarowały pożądane zachowanie systemu oraz nawigację między poszczególnymi jego elementami. Schemat bazy danych określił jasno, jakie informacje będą przechowywane na serwerze oraz jaki będzie ich format. Ostatecznym elementem wymaganym przed przystąpieniem do pisania kodu źródłowego, było stworzenie projektu interfejsu graficznego użytkownika. Ułatwiło to w znaczącym stopniu późniejszą jego implementację, ponieważ istniał wzór, z którego można było korzystać podczas implementacji. Wszystkie te etapy były niezbędne do otrzymania rezultatu, jakim jest stworzona aplikacja.

4. STUDIUM PRZYPADKU ZASTOSOWANIA SYSTEMU

Tak jak zostało opisane w jednym z wcześniejszych rozdziałów, w obliczu rosnącej potrzeby dbania o zdrowie psychiczne, aplikacje wspierające odporność stają się niezastąpionym narzędziem w codziennym funkcjonowaniu. Niniejsza część skupia się na kluczowych aspektach dotyczących systemu, przeprowadzając szczegółową ocenę jej użyteczności poprzez odpowiedzi użytkowników w ankiecie.

Na wstępie przedstawione zostaną cele projektowanego formularza, jak dostosowano pytania, aby dowiedzieć się, w jaki sposób aplikacja wpływa na odporność psychiczną użytkowników oraz czy korzystanie z niej jest intuicyjne. Opisana będzie struktura pytań, kryteria doboru uczestników i sposób przeprowadzenia badania.

W ostatniej części rozdziału omówione zostaną wyniki ankiety i ich analiza. Ważne jest, aby na podstawie tak przeprowadzonego kwestionariusza wyciągnąć dalsze wnioski, wyszczególnić co przekonuje użytkowników do korzystania z niej oraz wydobyć mocne strony, które posiada aplikacja. Z drugiej strony istotne jest uświadomienie, jakie elementy wymagają poprawy lub co należy dodać, aby zyskać nowych, potencjalnych interesariuszy. Te wnioski będą stanowiły fundament dla dalszych działań nad udoskonaleniem aplikacji oraz dostosowaniem jej do potrzeb użytkowników.

4.1. Założenia i przebieg (*Patryk Gołembiewski*)

4.1.1. Cele ankiety

Formularz został dostosowany pod kątem konkretnych doświadczeń związanych z odpornością psychiczną. Zadaniem jest zrozumienie, w jaki sposób użytkownicy postrzegają wpływ aplikacji na ich odporność psychiczną i zdolność radzenia sobie ze stresem, presją, czy trudnościami życiowymi.

Drugim kluczowym obszarem są pytania dotyczące intuicyjności korzystania z aplikacji. Kwestionariusz sprawdza, czy interfejs użytkownika jest postrzegany jako intuicyjny, czy menu i funkcje są zrozumiałe, a proces korzystania z aplikacji jest pozbawiony zbędnych trudności. Oceniane będą zdolności użytkowników do swobodnego poruszania się po aplikacji, korzystania z funkcji oraz szybkiego odnajdywania poszczególnych elementów. Ma to na celu identyfikację obszarów, które mogą wymagać poprawek w dziedzinie interakcji użytkownika z systemem.

Zatem ogólnym zadaniem przeprowadzanej ankiety jest uzyskanie kompleksowego zrozumienia doświadczeń użytkowników korzystających z aplikacji Good Mentality. Poprzez połączenie tych dwóch aspektów, możliwe będzie dążenie do stworzenia wizji, umożliwiającej dostosowanie aplikacji tak, aby nie tylko efektywnie wpływała na odporność psychiczną, lecz także była łatwa w użyciu i dostępna dla szerokiej grupy użytkowników.

4.1.2. Struktura pytań

Pytania zadawane w ankiecie można podzielić na trzy rodzaje, które zostały opisane w niniejszej części:

- **Ogólne działanie systemu:** Pytania dotyczące ogólnego działania systemu są skoncentrowane na doświadczeniach i odczuciach użytkowników. Próba zobrazowania całokształtu

działania aplikacji, czy moduły są zintegrowane ze sobą i czy korzystanie przyniosło pozytywne rezultaty, jest niezwykle istotna w kontekście analizy. Przykładowe pytania: „Czy korzystanie z aplikacji wpłynęło na Twoją zdolność radzenia sobie ze stresem? Jeśli tak, opisz pozytywny wpływ.”, „Czy działanie aplikacji było spójne (czy występowały pojedyncze niespójności w interfejsie lub tematyce)?”.

- Intuicyjność: Drugi obszar, który jest poruszany podczas ankiety, dotyczy doświadczeń użytkowników z wytworzonym interfejsem oraz mechaniką aplikacji. Pytania pozwalają uzyskać informacje na temat przemyśleń interesariuszy o łatwości odnajdywania potrzebnych funkcji, trudnościach w nawigacji oraz czy to, co widzą, jest dla nich przyjazne i zrozumiałe. Przykładowe pytanie: "Czy uważasz, że nawigacja między modułami jest łatwa i zrozumiała?".
- Skuteczność poszczególnych modułów: Ważne jest również stwierdzenie, czy poszczególne elementy aplikacji pozytywnie wpływają na budowanie odporności psychicznej oraz, czy treści dydaktyczne umieszczone na stronie okazały się pomocne. Dzięki temu można określić które moduły są szczególnie istotne dla użytkowników, czy pomagają wpłynąć na pozytywny efekt płynący z użytkowania aplikacji. Przykładowe pytanie: „Czy moduł „Sennik” miał istotny wpływ na wspieranie Twojej odporności psychicznej?”.

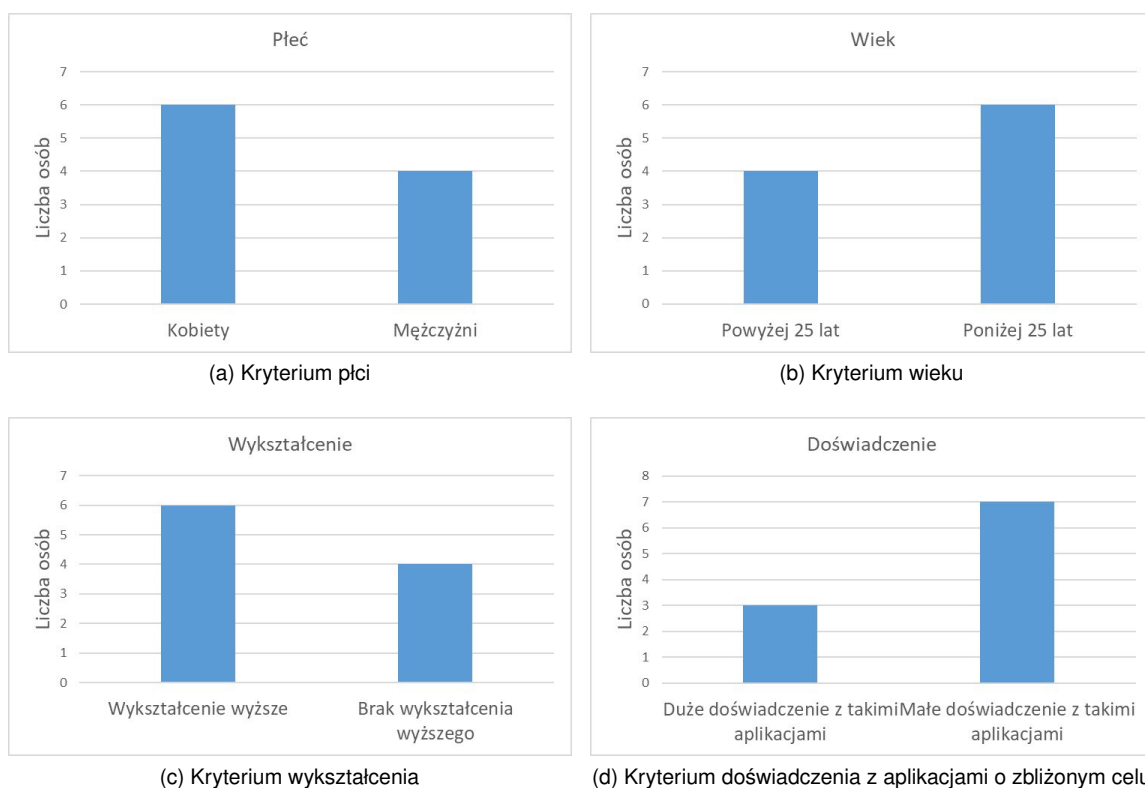
4.1.3. Kryteria wyboru ankietowanych

Aby móc dowiedzieć się potrzebnych rzeczy na temat funkcjonowania aplikacji i jej wpływu na korzystających z niej ludzi, konieczny jest odpowiedni wybór grupy ankietowanych. Umożliwi to uzyskanie kompleksowego obrazu doświadczeń użytkowników związanych z systemem wspierającym wzmacnianie odporności psychicznej. Podczas wyboru takowej grupy wzorowaliśmy się na wymienionych głównych kryteriach:

- Różnorodność demograficzna: Aby ankieta miała odpowiednie znaczenie dla badania skuteczności działania, kluczowe jest zapewnienie reprezentatywności pod względem demograficznym. Ankieta obejmuje osoby o zróżnicowanym wieku, płci i poziomie wykształcenia. Różnorodność ta pozwala na uzyskanie globalnej perspektywy na wpływ aplikacji oraz łatwość użytkowania.
- Doświadczenie z podobnymi narzędziami: Ważne jest również uzyskanie informacji na temat działania systemu od użytkowników, którzy pierwszy raz spotkali się z aplikacją tego typu oraz od wcześniej korzystających z podobnych systemów. Różne poziomy doświadczenia ankietowanych z podobnymi aplikacjami wspierającymi zdrowie psychiczne pozwala na zrozumienie różnic w oczekiwaniach między różnymi grupami użytkowników.
- Zróżnicowane potrzeby zdrowotne: Celem ankietyzacji jest między innymi poznanie wpływu aplikacji na odporność psychiczną użytkowników. Wiele cennych wiadomości pozwoli uzyskać przepytanie osób, które posiadają zróżnicowane potrzeby zdrowotne. Badanie uwzględnia osoby, które są narażone na zwiększony stres, lęki, ale także te nieposiadające problemów, a jedynie chcących działać profilaktycznie. Pozwoli to na zrozumienie, jak aplikacja odpowiada na różne potrzeby użytkowników.

W przypadku badania aplikacji, która wpływa na ludzką psychikę, wzmacniając jej odporność, zakres tematyczny oraz możliwe różne odbieranie funkcjonalności skłania do zaangażowania odpowiedniej liczby ankietowanych. Ostateczna liczba uczestników nie może być zbyt mała, gdyż nie dałoby to adekwatnych i nieprzekłamanych wyników. W przypadku tego kwestionariusza uzyskanie 10 osób chętnych do podzielenia się swoim zdaniem pozwoliło na zebranie odpowiedzi

zróznicowanych pod względem nie tylko demograficznym, ale także potrzeb zdrowotnych i zaawansowania w korzystaniu z podobnych systemów. Na rysunku 4.1 podano liczbę osób dla poszczególnego kryterium.



Rys. 4.1. Rozkład osób biorących udział w badaniu ze względu na poszczególne kryterium

4.1.4. Sposób przeprowadzenia badania

Po zapoznaniu się wybranymi osobami z działaniem systemu i jego funkcjonalnościami, zostali oni poproszeni o wypełnienie wcześniej przygotowanego formularza. Ze względu na to, że użytkownicy mieszkają w różnych częściach Polski, ankieta musiała zostać udostępniona online przez pięć dni. Została ona stworzona za pomocą Google Forms i za pomocą tego narzędzia udostępniono ją pierwszym osobom korzystającym z systemu. Zasada anonimowości została starannie zachowana, co miało zwiększyć uczciwość i otwartość w udzielanych odpowiedziach.

4.2. Analiza wyników (Szymon Hejmanowski)

Analiza obejmuje wybrane kategorie, które zostały wcześniej opisane. Jako efekt rozważań uzyskano głębokie zrozumienie, w jaki sposób aplikacja wpływa na odporność psychiczną oraz jakie kroki można podjąć w przyszłości w celu udoskonalenia systemu. Przejście przez wyniki ankiety umożliwiło ukierunkowanie działań, jakie zostaną podjęte, dostosowując system do potrzeb interesariuszy. Znalezienie optymalnego balansu między identyfikacją mocnych stron a jednocześnie rozpoznaniem obszarów do poprawy, jest kluczowe dla dążenia do dostarczania narzędzia skutecznego i zaspokajającego różnorodne potrzeby użytkowników w zakresie zdrowia psychicznego.

Na podstawie analizy odpowiedzi, wynikających z przeprowadzonego badania, wyróżniono kilka pozytywnych aspektów:

- Intuicyjność interfejsu i łatwość obsługi: Odpowiedzi potwierdzają, że stworzony interfejs użytkownika można uznać za intuicyjny (potwierdziło 10/10 osób). Dodatkowo menu oraz poszczególne funkcjonalności są postrzegane za łatwo dostępne (potwierdziło 9/10 osób). To kluczowy pozytywny, sugerujący poprawną implementację części *frontendowej*. Oznacza to, że użytkownicy doświadczają płynnego korzystania z systemu.
- Efektywność w kształtowaniu odporności psychicznej: Jednoznacznie stwierdzono pozytywny wpływ działania aplikacji na stan psychiczny użytkowników (potwierdziło 10/10 osób). Oznacza to, że wyszukane treści oraz moduły wydają się skutecznym narzędziem we wspieraniu odporności psychicznej. Modułem, który najczęściej był wybierany jako wywierający największy wpływ na wzmacnianie odporności psychicznej, został dziennik nastroju.
- Pozytywne doświadczenia użytkownika: Szereg pozytywnych komentarzy od użytkowników podkreśla ogólne zadowolenie z korzystania z aplikacji. Odpowiedzi sugerują, że narzędzie spełnia oczekiwania oraz dostarcza pozytywnych doświadczeń. Przykład: "Aplikacja posiada ciekawy interfejs, sprawnie działa oraz zdecydowanie pomaga we wzmacnianiu odporności psychicznej."

Analizując odpowiedzi ankietowanych, zaobserwowano kilka obszarów, które wymagają poprawy lub dodania do istniejącego systemu:

- Personalizacja doświadczenia użytkownika: Niektórzy użytkownicy wyrazili chęć większej możliwości personalizacji aplikacji. Jako jedno z częściej pojawiających się usprawnień określano opcję zmiany języka. Aktualnie dostępny jest jeden – polski (dla każdego z ankietowanych ojczysty). Analizując to, wywnioskowano, że może to prowadzić do lepszego zrozumienia treści oraz umożliwić aplikacji dotarcie do szerszego grona odbiorców.
- Feedback i interakcja: W ankiecie pojawiło się kilka sugestii dotyczących dodatkowych mechanizmów otrzymywania informacji zwrotnej i interakcji w aplikacji. Niektórzy użytkownicy zauważyli, że takie dodatkowe elementy mogą zwiększyć zaangażowanie i motywację do korzystania z systemu. Jako przykładowy podano system gratyfikacji za codzienne wpisy do dziennika nastroju.

Podsumowując wyniki analizy zebranych odpowiedzi w ramach przeprowadzanej ankiety, zostało zidentyfikowanych kilka pozytywnych aspektów związanych z działaniem aplikacji. Menu oraz funkcjonalności są postrzegane jako dostępne i łatwo przyswajalne nawet dla użytkowników z małym doświadczeniem z tego typu aplikacjami. Przekłada się to na płynne korzystanie z systemu i rozumienie treści. Dobrze dobrane funkcjonalności razem z odpowiednim interfejsem pozwalają użytkownikom na odpowiednie działanie, które ma na celu wspieranie odporności psychicznej. Jednocześnie analiza odpowiedzi ujawniła obszary wymagające poprawy. Wyniki badania dostarczają cennych informacji na temat efektywności i użyteczności aplikacji. Pozytywne doświadczenia użytkowników stanowią fundament, jednak identyfikacja obszarów do poprawy jest równie istotna, mając na celu dalszy rozwój systemu. Dzięki tej ankiecie można jasno stwierdzić, że w pełni zostały zrealizowane założenia projektu.

5. PODSUMOWANIE (PATRYK GOŁEMBIEWSKI, SZYMON HEJMANOWSKI)

Każda z kolejnych metod projektowania systemu, a następnie jego implementacji była krokiem w kierunku celu – stworzenie optymalnej aplikacji pod kątem wspomagania odporności psychicznej. Ostateczny efekt został uzyskany dzięki wypełnieniu wszystkich zadań zdefiniowanych na samym początku. Zgłębiono temat odporności psychicznej oraz wykonano przegląd podobnych systemów. Wypracowano wizję aplikacji w oparciu o metody kreatywności. Proces analizy, jak i projektu systemu został w pełni udokumentowany w pracy. Etap implementacji i testowania zaowocował stworzeniem modułów zgodnych z wizją systemu. Ostatecznym etapem, było przeprowadzone studium przypadku zastosowania systemu, które to pozwoliło dowieść wartości wykonanej pracy.

Podział pracy praktycznej między członków zespołu wyglądał następująco – Patryk Gołembiewski odpowiedzialny był za dwa moduły o estymowanej dużej złożoności, czyli planner oraz dziennik. Szymon Hejmanowski natomiast implementował mechanizm logowania i rejestracji o wysokiej złożoności oraz dwa moduły o niskiej złożoności, czyli senniki oraz cytaty.

Uczącym aspektem projektu był fakt, jak dużą część tworzenia aplikacji stanowi jej projektowanie. Jest to niezwykle istotny element, lecz często lekceważony. Te detale systemu, które zostały dokładnie rozpatrzone, przemyślane oraz opisane przed faktyczną implementacją, nie stanowiły problemu w trakcie pisania aplikacji. Inaczej wyglądała praca z kodem w przypadku części, które zostały pominięte lub zaniedbane w procesie przygotowawczym. Mimo tego, że często pojawia się chęć jak najszybszego przystąpienia do implementacji, warto zatrzymać się i porządnie przedyskutować ważne kwestie, aby oszczędzić czasu w przyszłości.

Kolejne iteracje tworzenia systemu powinny zakładać dodawanie nowych funkcjonalności, które zostały odrzucone w wizji systemu ze względu na ograniczony czas. Pozytywny wpływ na odbiór aplikacji miałyby również ujednolicenie interfejsu graficznego oraz jego poprawa. Mimo stwierdzonej dużej intuicyjności aspekt estetyczny mógłby zostać ulepszony. W przypadku znacznego rozwoju aplikacji kluczowe byłoby wprowadzanie kolejnych języków. W pierwszej kolejności byłby to język angielski. Co więcej, istotnym usprawnieniem od strony administracyjnej, byłoby wprowadzenie dedykowanego panelu webowego. Mógłby on znaczenie usprawnić moderowanie danych oraz użytkowników w momencie, gdy powstałyby kolejne funkcjonalności tego wymagające.

WYKAZ LITERATURY

- [1] Olga Rymkiewicz, <https://olgarymkiewicz.pl/model-4c-czyli-skad-sie-bierze-odpornosc-psychiczna/>, (data dostępu 03.12.2023 r.).
- [2] Opracowanie zbiorowe: *Coaching eXtra 3/2021*, Burda Publishing Polska, 09.2021.
- [3] Pressman R. S.: *Software Engineering: A Practitioner's Approach*, McGrawHill, 2000
- [4] Nęcka E.: *Trening Twórczości*, Gdańskie Wydawnictwo Psychologiczne, 2013
- [5] Strycharczyk D.: *Odporność Psychiczna*, GWP, 2021
- [6] Personal Zen, <https://personalzen.com/>, (data dostępu 23.05.2023 r.).
- [7] Breath2Relax, <https://apps.apple.com/us/app/breathe2relax/id425720246>, (data dostępu 23.05.2023 r.).
- [8] Sanvello, <https://www.sanvello.com/>, (data dostępu 23.05.2023 r.).
- [9] Serenita, <https://apps.apple.com/us/app/serenita-stress-anxiety/id954746715>, (data dostępu 23.05.2023 r.).
- [10] Woebot, <https://woebothealth.com/>, (data dostępu 23.05.2023 r.).
- [11] Fitbit, <https://apps.apple.com/us/app/fitbit-health-fitness/id462638897>, (data dostępu 23.05.2023 r.).
- [12] Medium, <https://medium.com/@davegray/updated-empathy-map-canvas-46df22df3c8a>, (data dostępu 04.06.2023 r.).
- [13] Olsson M.: *A Comparison of Performance and Looks Between Flutter and Native Applications*, Blekinge Institute of Technology, 2020.
- [14] IBM Documentation, <https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case>, (data dostępu 05.12.2023 r.).
- [15] Enterprise Architect, <https://www.sparxsystems.eu/>, (data dostępu 05.12.2023 r.).
- [16] Dokumentacja Flutter, https://pub.dev/packages/flutter_local_notifications (data dostępu 07.12.2023 r.).
- [17] Dokumentacja Flutter, <https://api.flutter.dev/flutter/widgets/FutureBuilder-class.html> (data dostępu 07.12.2023 r.).

WYKAZ RYSUNKÓW

2.1	Zrzuty ekranu z aplikacji Personal Zen	10
2.2	Zrzuty ekranu z aplikacji Senvello	11
2.3	Wykorzystany schemat mapy empatii	17
3.1	Architektura systemu wraz z funkcjami oraz zawartością poszczególnych elementów	27
3.2	Diagram przypadków użycia użytkownika	28
3.3	Diagram przypadków użycia administratora	29
3.4	Schemat bazy danych (nazwy pól przetłumaczone na język polski)	31
3.5	Fragment kodu aplikacji przedstawiający zapytanie serwera o dane	31
3.6	Makiety graficzne modułu „Dziennik”	32
3.7	Fragment kodu komponentu wywołującego widżet logowania i rejestracji lub widoku domowego	33
3.8	Fragment kodu komponentu zarządzającego wyświetlaną treścią	34
3.9	Fragment kodu widżetu logowanie umożliwiające zmianę widoku	34
3.10	Fragment kodu paska nawigacyjnego.	35
3.11	Funkcja odpowiedzialna za zmianę indeksu służącego do wyświetlenia odpowiedniego modułu.	36
3.12	Fragment kodu widżetu, w który poprzez zmianę zawartości ciała pokazuje wybrany moduł.	36
3.13	Fragment kodu przedstawiający inicjalizację stanu widżetu z wywołaniem funkcji pobierającej dane z bazy.	38
3.14	Fragment kodu przedstawiający funkcję pobierającą dane z bazy.	38
3.15	Fragment kodu ukazujący zastosowanie FutureBuildera.	39
4.1	Rozkład osób biorących udział w badaniu ze względu na poszczególne kryterium .	42

WYKAZ TABEL

2.1	Porównanie funkcji podobnych systemów	12
2.2	Wnioski płynące z negatywnej burzy mózgów	18
2.3	Ustalone oceny poszczególnych funkcjonalności	21
3.1	Przypadek użycia – Logowanie	29
3.2	Przypadek użycia – Przeglądanie hasła motywacyjnego	29
3.3	Przypadek użycia – Dodanie nowego hasła	30
3.4	Stopień realizacji poszczególnych modułów	37

DODATEK A: TREŚĆ PYTAŃ PIERWSZEGO BADANIA ANKIETOWEGO

1. Czy jesteś zainteresowany poprawą odporności psychicznej?
Tak / Nie
2. Czy aktywnie pracujesz nad poprawą swojej odporności psychicznej?
Tak / Nie
3. Czy jesteś zaznajomiony z modelem 4C?
Tak / Nie
4. Który z czterech obszarów modelu 4C jest Twoim zdaniem najbardziej istotny?
 - Wyzwanie
 - Pewność siebie
 - Zaangażowanie
 - Poczucie wpływu
5. Opisz co najbardziej wpływa na Twój stan psychiczny?
Odpowiedź opisowa
6. Jakie elementy z życia codziennego są dla Ciebie najtrudniejsze?
Odpowiedź opisowa
7. Jakie metody poprawy odporności psychicznej działają na Ciebie najlepiej?
Odpowiedź opisowa
8. Czy korzystałeś z aplikacji wspierających odporność psychiczną?
Tak / Nie
9. Jeśli tak, to jakich?
Odpowiedź opisowa
10. Czy uważasz, że aplikacje mogą wspierać odporność psychiczną człowieka?
Tak / Nie
11. Jeśli tak, to w jaki sposób?
Odpowiedź opisowa
12. Z jakich funkcjonalności wpływających na twój stan psychiczny korzystałeś?
 - Dziennik nastroju / ocena poziomu stresu
 - Treści psychoedukacyjne
 - Ćwiczenia przyciągające uwagę / odciągające uwagę od stresujących sytuacji
 - Instruktor oddechu
 - Rady przyszłościowe
 - Inne (pole tekstowe)
13. Czy wyżej wymienione funkcjonalności pomogły ci?
Tak / Nie
14. Jeśli nie, dlaczego uważasz, że nie wpłynęły pozytywnie?
Odpowiedź opisowa
15. Co najbardziej odrzuca cię od takich aplikacji?
 - Wymagają zbyt dużej ilości czasu
 - Brak określonego języka
 - Zniechęcasz się po krótkim okresie użytkowania
 - Brak wiary w ich skuteczność

- Inne: (pole tekstowe)

16. Jakie funkcjonalności byłyby pomocne w takiej aplikacji?

Odpowiedź opisowa

W przypadkach, gdy pytanie mogło być nie jednoznaczne, zawarty był dodatkowy opis lub przykładowa odpowiedź.

DODATEK B: TREŚĆ PYTAŃ DRUGIEGO BADANIA ANKIETOWEGO

1. Jakiek są Twoje ogólne doświadczenia z korzystania z tej aplikacji?
Odpowiedź opisowa.
2. Czy korzystanie z aplikacji wpłynęło na Twoją zdolność radzenia sobie ze stresem?
Tak / Nie
3. Jeśli tak, opisz pozytywny wpływ.
Odpowiedź opisowa.
4. Czy działanie aplikacji było spójne (czy nie występowały pojedyncze niespójności w interfejsie lub tematyce)?
Tak / Nie
5. Jeśli nie, podaj przykłady.
Odpowiedź opisowa.
6. Czy oceniasz interfejs aplikacji jako intuicyjny, łatwy w obsłudze?
Tak / Nie
7. Czy uważasz, że nawigacja między modułami aplikacji jest łatwa i zrozumiała?
Tak / Nie
8. Który z modułów aplikacji uważasz za najbardziej skuteczny w kształtowaniu twojej odporności psychicznej?
 - Sennik
 - Hasła motywacyjne
 - Planner
 - Dziennik nastroju
9. Opisz dlaczego.
Odpowiedź opisowa.
10. Czy moduł Sennik miał istotny wpływ na wspieranie Twojej odporności psychicznej?
Tak / Nie
11. Czy moduł Hasła motywacyjne miał istotny wpływ na wspieranie Twojej odporności psychicznej?
Tak / Nie
12. Czy moduł Planner miał istotny wpływ na wspieranie Twojej odporności psychicznej?
Tak / Nie
13. Czy moduł Dziennik nastroju miał istotny wpływ na wspieranie Twojej odporności psychicznej?
Tak / Nie
14. Czy uważasz, że aplikacja jest odpowiednio spersonalizowana do Twoich potrzeb?
Tak / Nie
15. Jeśli widzisz potrzebę zmiany/dodania czegoś w aplikacji, proszę to opisać.
Odpowiedź opisowa.