**1. Grupa**

Data założenia: 26.02.2020

Liczba członków: 4

Członkowie:

- Łukasz Mrzygłód

- Dariusz Waltoś

- Mateusz Orelik

- Tomasz Patrzałek

**2. Temat projektu**

Tematem projektu jest aplikacja mobilna „Organizer”. Głównym założeniem projektu  
jest stworzenie aplikacji, która usprawni zarządzanie czasem oraz planowanie przebiegu dnia.

**3. Protokół założycielski**

Spis treści:

1. Temat projektu i wytyczne projektu

2. Zespół pracujący przy projekcie

3. Wynagrodzenie

4. Kary

5. Metody komunikacji

6. Postanowienia końcowe

1.Temat projektu i wytyczne projektu

Projekt zlecony zespołowi nazywa się „Organizer”. Celem projektu jest stworzenie aplikacji mobilnej na system Android, która pomoże w organizacji czas. Projekt rozpoczyna się z dniem 26.02.2020, a zakończenie projektu planowane jest na 03.06.2020. Projekt jest tworzony na podstawie tematu pracy inżynierskiej członka projektu Łukasza Mrzygłóda, a rezultatem ukończenia projektu jest zaliczenie zajęć projektowych.

2.Zespół pracujący przy projekcie i rozliczanie się

1. Tworzenie zespołu

Do tworzenia aplikacji został dobrany zespół studentów Informatyki na specjalizacji „Technologie Internetowe i Mobilne”.

1. Kierownik projektu i jego wybranie

Kierownik projektu został wybrany przez członków zespołu i został nim Łukasz Mrzygłód.

1. Członkowie zespołu projektowego

-Łukasz Mrzygłód(kierownik projektu, programista, projektant interfejsu)

-Tomasz Patrzałek(programista, tester)

-Dariusz Waltoś( programista, projektant interfejsu)

-Mateusz Orelik (programista, tester)

1. Rozliczanie się z kierownikiem

Zespół rozlicza się z wykonania zadań z kierownikiem, co niedzielę w systemie tygodniowym.

3. Wynagrodzenia i kary i terminy rozliczeń:

1. Nagrody:

Każdy z członków zespołu będzie wynagradzany za swoją pracę w systemie tygodniowym (w każdą środę pracującą) przez otrzymanie wpisu z zaliczeniem etapu projektu z danego tygodnia.

1. Kary:
2. Niepojawienie się na spotkaniu:

Ustna reprymenda od kierownika projektu

1. Niestosowanie się do zaleceń kierownika projektu:

Ustna reprymenda kierownika projektu

1. Niewykonanie zadania na czas:

System kar za niedotrzymanie terminów rozliczenia się z danego etapu jest następujący. Przy nie rozliczeniu się z etapem członek zespołu odpowiedzialny za dany element który spowodował niemożność rozliczenia etapu jest zobowiązany do wykonania zadania fizycznego, wymyślonego przez resztę zespołu, w celu pobudzenia krążenia i zwiększenia poziomu endorfin i adrenaliny we krwi, oraz przemyślenia swojego postępowania.

1. Rozliczenia

Członkowie projektu rozliczani są w systemie tygodniowym co niedzielę przez kierownika projektu i co środę przez prowadzącego zajęcia projektowe.

5.Metody komunikacji

Komunikacja z zespołem będzie odbywała się przez narzędzie

- Trello

- Facebooka

- Githuba

6.Postanowienia końcowe

Kierownik projektu zastrzega sobie prawo do użycia projektu przy tworzeniu własnego projektu inżynierskiego. Tym samym pozbawiając resztę członków grupy projektowej do roszczenia sobie praw do projektu bez względu na ilość włożonej w niego pracy.

**4. Mapa myśli**

****

**5. Mapa koncepcyjna**

**6. Niefunkcjonalności projektu**

|  |  |  |
| --- | --- | --- |
| Nr cechy | Cecha niefunkcjonalna | Opis cechy |
| 1 | Zmiana kolorystyki aplikacji | Użytkownik może zmienić kolor aplikacji |
| 2 | Zmiana czcionki w aplikacji | Zmiana czcionki wyświetlanego tekstu |
| 3 | Zmiana kolorystyki kategorii | Użytkownik może zmienić kolor danej kategorii |
| 4 | Zmiana kolorystyki wydarzeń | Użytkownik może zmienić kolor danego wydarzenia |
| 5 | Zmiana języka aplikacji | Zmiana języka wyświetlanego w całej aplikacji z polskiego na angielski i na odwrót |
| 6 | Modyfikacja powiadomień | Modyfikowanie powiadomień (dźwięk, kolor) |
| 7 | Modyfikacja kalendarza | Zmiana kolorystyki kalendarza |
| 8 | Zmiana formatu dat | Zmiana formatu dat z domyślnego (DD-MM-RRRR) na np. RRRR-MM-DD |
| 9 | Zmiana systemu godzinowego | Zmiana pomiędzy systemem godzinowym używanym przy rozpisie wybranego dnia (24-godzinny lub 12-godzinny AM, PM) |
| 10 | Wyświetlenie panelu informacji | Wyświetlenie panelu z informacjami o wersji aplikacji, jej autorze oraz z danymi do ew. kontaktu z autorem |

**7. Funkcjonalności projektu**

|  |  |  |
| --- | --- | --- |
| Nr funkcji | Nazwa funkcji | Opis funkcji |
| 1 | Obsługa kalendarza | Wyświetlanie kalendarza. Możliwość wyboru roku, miesiąca oraz dnia. Po wybraniu konkretnego dnia przejście do panelu odpowiadającemu temu dniu. |
| 2 | Generowanie planu dnia wybranego  w kalendarzu | Po wybraniu dnia z kalendarza wyświetla się panel z listą zadań oraz informacjami na temat priorytetu zadań, ilością czasu wolnego i pracy w ciągu danego dnia oraz notatkami do danych zadań |
| 3 | Stworzenie listy zadań do wykonania | Stworzenie listy, w której zawierać się będą dodane przez użytkownika zadania do wykonania danego dnia. |
| 4 | Dodanie zadań do listy zadań | Dodanie zadań zadeklarowanych przez użytkownika do wcześniej utworzonej listy zadań. |
| 5 | Dodawanie kategorii dla każdego  z zadań | Stworzenia kategorii zadań (np. praca, dom, szkoła) w zależności od środowiska. |
| 6 | Kwalifikacja zadań do kategorii | Podpięcie utworzonych wcześniej zadań do określonych kategorii. |
| 7 | Nadawanie określonego priorytetu poszczególnym zadaniom | Nadanie priorytetu  (od 1 do 5, 1 - najważniejszy, 5 - najmniej ważny) każdemu zadaniu. |
| 8 | Dodawanie notatek do zadań | Możliwość dodatkowego opisu każdego z utworzonych zadań (np. podanie numeru telefonu kogoś do kogo musimy zadzwonić) |
| 9 | Dodanie przypomnienia do ważnych/cyklicznych zadań | Możliwość ustawienia przypomnienia w postaci notyfikacji na określony czas przed wykonaniem zadania (np. przypomnienie o przelewie na rachunki) |
| 10 | Wyświetlenie analizy danego dnia  (lista zadań, czas wolny, czas pracy) | Wyświetlenie analitycznych danych dot. konkretnego dnia tj. czas pracy, czas wolny, lista zadań do wykonania. |

**8. Specyfikacja funkcji**

|  |  |  |
| --- | --- | --- |
| Nr funkcji | Nazwa funkcji | Specyfikacja funkcji |
| 1 | Obsługa kalendarza | - kalendarz gregoriański  - podział na lata, miesiące i dni |
| 2 | Generowanie planu dnia wybranego  w kalendarzu | - każdy dzień miesiąca podzielony na przedziały czasowe co godzinę (00:00-01:00, 01:00-02:00 itd.) |
| 3 | Stworzenie listy zadań do wykonania | - utworzenie listy na zadania według wyżej wymienionego podziału godzinowego każdego dnia |
| 4 | Dodanie zadań do listy zadań | - wybranie przedziału czasowego, nadanie krótkiego opisu zadania do wykonania (max 150 znaków) |
| 5 | Dodawanie kategorii dla każdego  z zadań | - stworzenie grupy zadań o określonej kategorii środowiska np. dom, praca, szkoła  - nadanie nazwy kategorii (max 30 znaków) |
| 6 | Kwalifikacja zadań do kategorii | - dodanie stworzonych zadań do określonej kategorii w zależności od środowiska wykonywanego zadania |
| 7 | Nadawanie określonego priorytetu poszczególnym zadaniom | - nadanie każdemu zadaniu priorytetu przypisując daną wartość numeryczną (od 1 do 5) |
| 8 | Dodawanie notatek do zadań | - dołączenie krótkiego opisu do każdego zadania (max 50 znaków) |
| 9 | Dodanie przypomnienia do ważnych/cyklicznych zadań | - stworzenie przypomnienia w formie notyfikacji przypominające o zbliżającym się terminie wykonania zadania |
| 10 | Wyświetlenie analizy danego dnia  (lista zadań, czas wolny, czas pracy) | - wyświetlenie panelu z danymi analitycznymi takimi jak lista wszystkich zadań, czas pracy w godzinach, czas wolny w godzinach, liczba zadań, liczba zadań w danej kategorii |

**9. Narzędzia**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nazwa narzędzia | Przeznaczenie | Podmiot | Wersja | Repozytorium |
| Android Studio | Zaprogramowanie aplikacji | JetBrains | 3.6 | GitHub |
| GitHub | Kontrola wersji | GitHub, Inc. |  | GitHub |
| Pakiet MS Office | Sporządzenie specyfikacji oraz sprawozdania projektu | Microsoft | Office 2019 | GitHub |
| Draw.io | Sporządzenie potrzebnych diagramów do wykonania projektu | SEIBERT MEDIA Corp. | 12.6.5 | GitHub |
| Axure RP9 | Projektowanie interfejsów | Axure Software Solutions, Inc. | 9.0.0 | GitHub |

**10. Zdefiniowanie użytkowników**

System – system na urządzeniu użytkownika

Użytkownik – osoba korzystająca z aplikacji

**11. Diagram przypadków użycia**

****

**12. Słownik**

Ogranizer – narzędzie do usprawnienia zarządzania planem dnia

Kalendarz – rozpis dni danego miesiąca

Plan konkretnego dnia – podział godzinowy danego dnia

Przypomnienia – notyfikacje przypominające o wykonaniu zadania

Lista zadań – zbiór czynności do wykonania danego dnia

Czas pracy/czas wolny – godziny pracy, godziny czasu wolnego

Hierarcha zadań – podział zadań według określonego priorytetu

Kategorie zadan – podział zadań według danego środowiska np. dom, praca, szkoła, sport

Notatki – informacje dodatkowe o każdym zadaniu

**13. Polityka**

|  |  |
| --- | --- |
| Na rzecz | Opis |
| bezpieczeństwa | Aplikacja chroniona będzie poprzez hasło, które użytkownik będzie wprowadzał przy pierwszym uruchomieniu aplikacji. Hasło to, będzie można w każdym momencie zmienić, poprzez odpowiednią autoryzację (mail, numer telefonu). |
| jakości | Aplikacja umożliwiać będzie wykonanie każdej czynności i każdego zagadnienia zadeklarowanego podczas opracowywania projektu oprogramowania. Aplikacja będzie na bieżąco aktualizowana w zależności od potrzeb. Każdy element aplikacji będzie opracowany z myślą o prostocie w używaniu oraz maksymalizacji możliwości dla użytkownika. |
| niezawodności | Aplikacja stworzona zostanie w środowisku umożliwiającym dostosowanie aplikacji dla szerszego grona odbiorców. Aplikacja testowana będzie dla różnych rodzajów urządzeń (oprogramowanie, wielkość ekranu, ograniczenia sprzętowe). Wszystkie funkcje aplikacji zostaną dopracowane i wszelkie możliwe do wykrycia błędy zostaną poprawione jeszcze przed stworzeniem finalnego produktu. Wszelkie błędy wykryte podczas użytkowania po wypuszczeniu finalnego produktu, będą naprawiane w aktualizacjach aplikacji. |

**14. Prototyp interfejsu użytkownika**

Podczas projektowania interfejsu użytkownika powstały trzy propozycje interfejsu graficznego użytkownika, dla potrzeb dalszej analizy numerowane kolejno 1, 2 i 3.

**Prototyp 1**



**Prototyp 2**

****

**Prototyp 3**

****

Dodatkowo zaprojektowany został jeden wariant listy rzeczy do zrobienia, oraz formularza dodania listy rzeczy do zrobienia do konkretnego, wybranego dnia.

****

****

**15. Analiza prototypów interfejsu użytkownika**

Analiza poszczególnych prototypów interfejsu opracowana została poprzez ocenę cech zawartych w poniższej tabeli. Ocena danego prototypu to zakres od 1 do 5 (1 - najgorszy,

5 - najlepszy).

|  |  |  |  |
| --- | --- | --- | --- |
| **Cecha/Prototyp** | **Prototyp 1** | **Prototyp 2** | **Prototyp 3** |
| Czytelność | **5** | **1** | **3** |
| Ergonomia | **3** | **1** | **5** |
| Łatwość nawigacji | **5** | **1** | **3** |
| Prostota obsługi | **5** | **3** | **3** |

**16. Biblioteki zewnętrzne używane w aplikacji**

Junit – będzie wykorzystane do napisania unit testów dla funkcji w aplikacji.

Appium – posłuży do zautomatyzowania testów aplikacji mobilnej.

**17. Testy**

Każda funkcja będzie posiadała Unit testy.

Oprócz nich po zaimplementowaniu większej części funkcjonalności aplikacji będą wykonane poniższe testy:

- End-to-end testy, które sprawdzają jak użytkownik korzystałby z aplikacji,

- Testy wydajności, które sprawdzają jak wydajna jest aplikacja,

- Smoke testy, sprawdzają podstawowe funkcjonalności aplikacji.

**18. Dziennik kontaktów z klientem oraz osoba odpowiedzialna**

Kontakt z klientem odbywa się drogą mailową. Osobą odpowiedzialną za kontakt z klientem jest Dariusz Waltoś.

Raz na tydzień przesyłamy klientowi informację dotyczącą postępu projektu. Po maksymalnie dwóch dniach klient odpisuje wiadomość zawierającą jego uwagi lub ich brak.

Jeżeli po 2 dniach nic nie odpisze, następnego dnia wykonywany jest telefon do niego i następuje rozmowa telefoniczna dotycząca postępu projektu.

Jeżeli uwagi klienta są zrozumiałe i zespół nie ma z nimi problemu, są one uwzględniane w projekcie, w innym przypadku wykonywany jest do niego telefon w którym Dariusz Waltoś przedstawia wątpliwości zespołu i jest prowadzona dyskusja, które zmiany powinny zostać wprowadzone, a które nie.

Po ich rozważeniu przedstawiana jest ostateczna wersja rozwinięcia projektu zespołowi.

**19. Raport dla klienta**

Raport będzie zawierać:

- Ilość nowych użytkowników w czasie(wzrost lub spadek),

- Średnia częstotliwość używania aplikacji przez użytkownika(wzrost lub spadek),

- Ilość ocen na Google Play w czasie( średnia oraz ilość),

- Demografia użytkowników aplikacji( wiek, płeć),

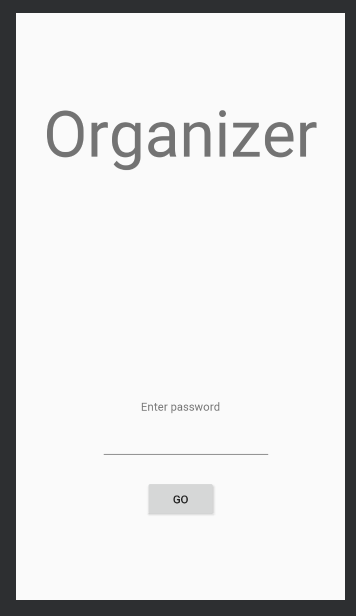
- Na żądanie klienta dodano formularz logowania, oraz dodawanie użytkowników i zabezpieczenie przeciwko botom(01.04.2020r)

**20. Diagram Gantta (harmonogram prac)**

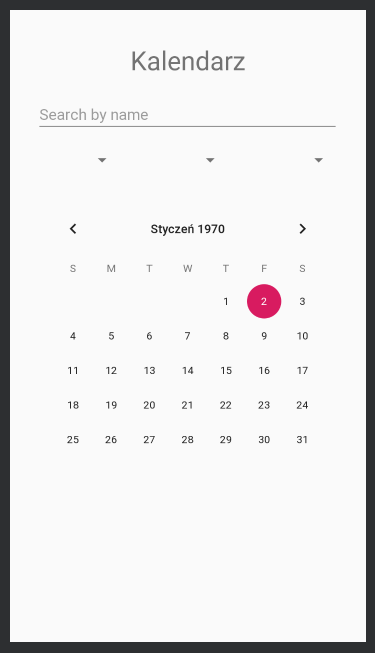
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Członek Data | 26.02.2020 | 04.03.2020 | 11.03.2020 | 18.03.2020 | 25.03.2020 | 01.04.2020 |
| Łukasz Mrzygłód (lider) | - założenie grupy projektowej  - sformułowanie tematu projektu  - sporządzenie protokołu założycielskiego | - poprawa protokołu założycielskiego | - zaprojektowanie Interfejsu  -poprawienie protokołu założycielskiego |  | Dodanie wybranych funkcji | Dodanie do dokumentacja nowego rozdziału dokumentującego nowe żądanie klienta |
| Dariusz Waltoś | - stworzenie mapy myśli  - stworzenie mapy konceptualnej | - zdefiniowanie użytkowników  - stworzenie diagramu przypadków użycia | - dziennik kontaktów z klientem  - raport dla klienta |  | Zastanowienie się na przypadkami testowymi, oraz znalezienie potrzebnych bibliotek zewnętrznych | Wybranie sposobu przechowywania danych |
| Mateusz Orelik | - stworzenie listy narzędzi  - stworzenie słownika | - opis niefunkcjonalności projektu | - polityka na rzecz: bezpieczeństwa, jakości, niezawodności |  | Dodanie wybranych funkcji | Wybranie sposobu przechowywania danych |
| Tomasz Patrzałek | - opis funkcjonalności  - specyfikacja funkcjonalności | - poprawa opisu  i specyfikacji funkcjonalności  - dokończenie niefunckjonalności  - aktualizacja dokumentacji | - polityka na rzecz: bezpieczeństwa, jakości, niezawodności | - analiza propozycji interfejsu graficznego użytkownika (GUI) | Implementacja interfejsu GUI | Zrealizowanie formularza logowania |

**21.Implementacja interfejsów GUI**

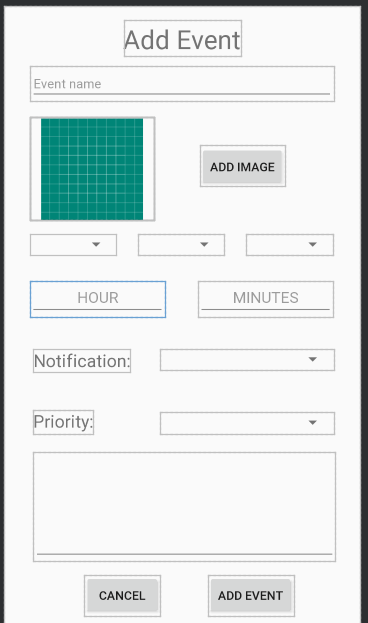
1. główny widok aplikacji:



1. widok kalendarza:



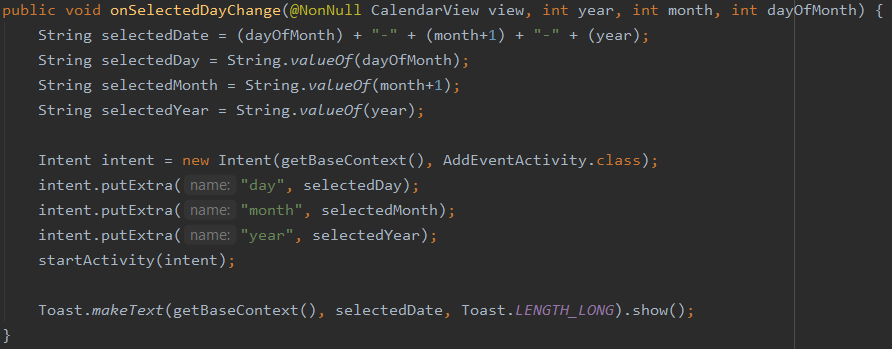
1. widok formularza dodawania zdarzeń:



**22. Implementacja wybranych funkcji aplikacji**

1. Przechodzenie do panelu dodawania wydarzenia

Po wybraniu w widoku kalendarza konkretnego dnia zostaje włączony formularz dodawania zdarzenia na dany dizeń.



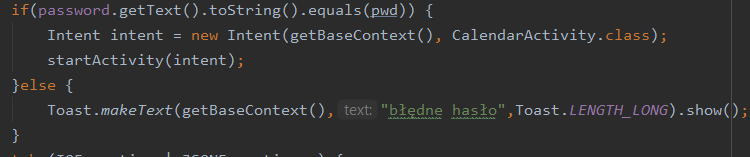
1. Sprawdzanie hasła przy próbie włączenia aplikacji

Po włączeniu się aplikacji naszym oczom ukazuje się ekran startowy jak w punkcie 21. Należy wtedy wprowadzić hasło. Funkcja przewiduje 3 stany wyjściowe wszystko zostało wykonane poprawnie i przechodzi do widoku kalendarza, zostało wprowadzone błędne hasło co skutkuje wyświetleniem komunikatu o błędnym haśle, lub wyświetlenie komunikatu o problemie z plikiem gdy wystąpo problem podczas wczytywania hasła z pliku.

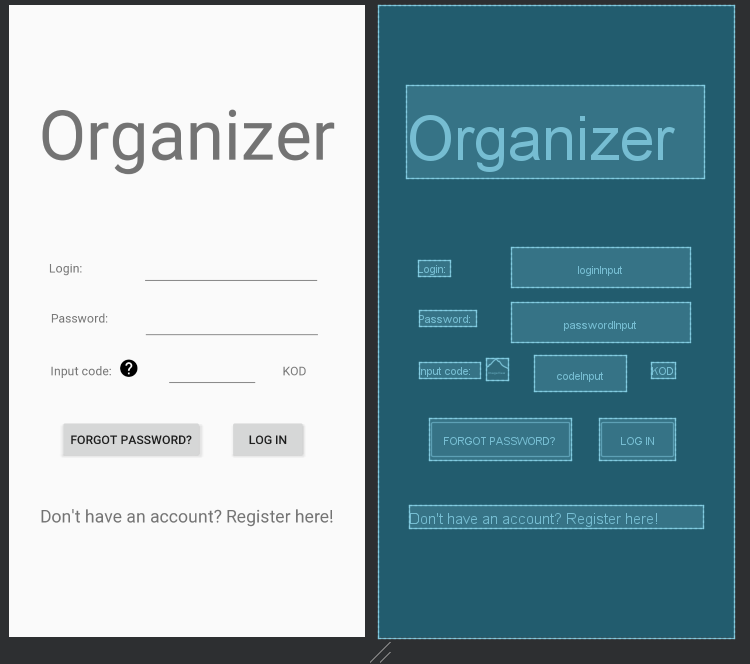


1. Przejście z ekranu startowego do widoku kalendarza

Po udanym sprawdzeniu hasła program przechodzi z widoku głównego do widoku kalendarza

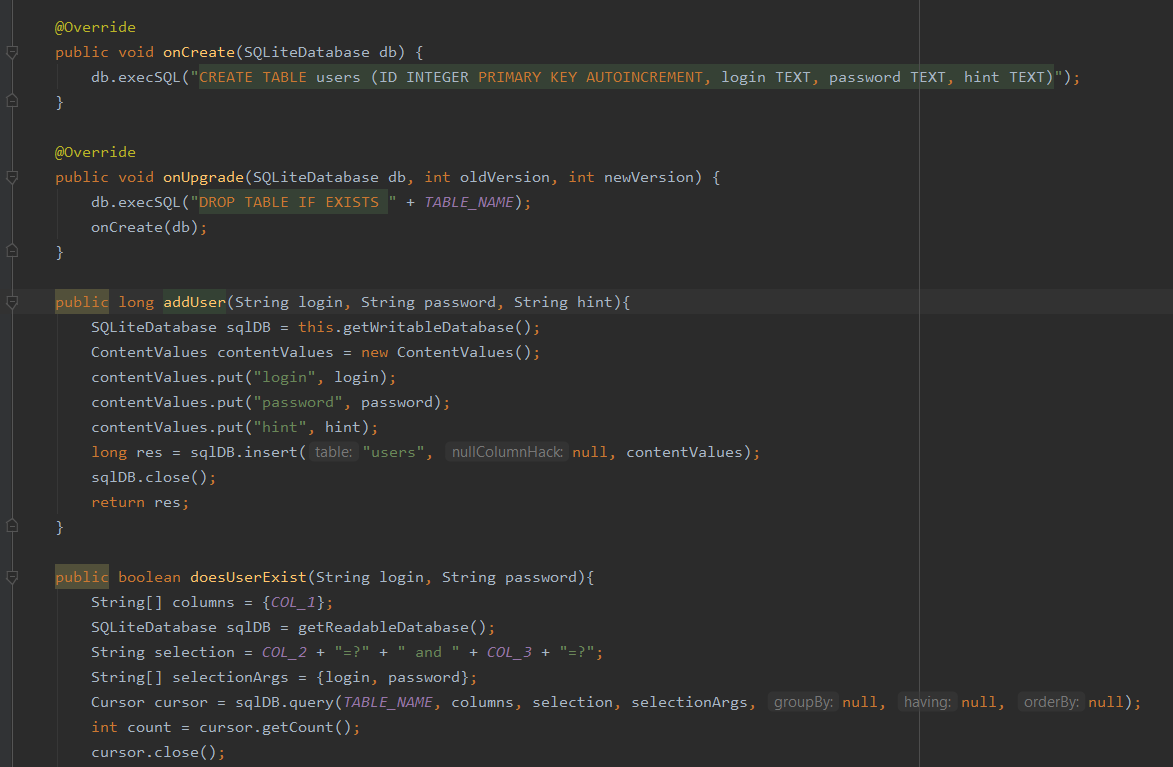


23.Dodanie dodatkowego żądanie klienta

1. Zmiana ekranu głównego z pola na hasło i przycisku na formularz logowania z zabezpieczeniem przeciw botom i opcją dodania nowego użytkownika i przypomnienia hasła.

24. Zmiana sposobu przechowywania danych:

Podczas prac zespół uznał że lepszym sposobem do przechowywania danych użytkownika będzie wprowadzenie bazy danych w technologi SQLite. W skutek czego system plików JSON został zastąpiony przez struktury lokalnej bazy danych.



Skutkowało to dodaniem klasy DatabaseHelper.