# **Politechnika Koszalińska 2020**

**DOKUMENTACJA PROJEKTU ZESPOŁOWEGO**

Łukasz Mrzygłód - U-14808

Tomasz Patrzałek - U-14819

Mateusz Orelik - U-14814

Dariusz Waltoś - U-15218

Informatyka, sem. VI

Studia Stacjonarne

**SPIS TREŚCI**

[**1. Grupa** 3](#_Toc40713842)

[**2. Temat projektu** 3](#_Toc40713843)

[**3. Protokół założycielski** 3](#_Toc40713844)

[**4. Mapa myśli** 5](#_Toc40713845)

[**5. Mapa koncepcyjna** 5](#_Toc40713846)

[**6. Niefunkcjonalności projektu** 6](#_Toc40713847)

[**7. Funkcjonalności projektu** 6](#_Toc40713848)

[**8. Specyfikacja funkcji** 7](#_Toc40713849)

[**9. Narzędzia** 7](#_Toc40713850)

[**10. Zdefiniowanie użytkowników** 8](#_Toc40713851)

[**11. Diagram przypadków użycia** 8](#_Toc40713852)

[**12. Szczegółowy opis przypadków użycia** 9](#_Toc40713853)

[**13. Słownik** 12](#_Toc40713854)

[**14. Polityka** 13](#_Toc40713855)

[**15. Prototyp interfejsu użytkownika** 13](#_Toc40713856)

[**16. Analiza prototypów interfejsu użytkownika** 16](#_Toc40713857)

[**17. Biblioteki zewnętrzne używane w aplikacji** 16](#_Toc40713858)

[**18. Testy** 16](#_Toc40713859)

[**19. Dziennik kontaktów z klientem oraz osoba odpowiedzialna** 17](#_Toc40713860)

[**20. Raport dla klienta** 18](#_Toc40713861)

[**21. Diagram Gantta (harmonogram prac)** 18](#_Toc40713862)

[**22.Implementacja interfejsów GUI** 18](#_Toc40713863)

[**23. Implementacja wybranych funkcji aplikacji** 20](#_Toc40713864)

[**24.Dodanie dodatkowego żądanie klienta** 22](#_Toc40713865)

[**25. Zmiana sposobu przechowywania danych** 22](#_Toc40713866)

[**26. Aktualna faza implementacji funkcji** 23](#_Toc40713867)

[**27. Założenia dokumentacji użytkownika** 24](#_Toc40713868)

[**28. Instrukcja użytkownika** 26](#_Toc40713869)

[**1. Wprowadzenie** 27](#_Toc40713870)

[**2. Opis aplikacji** 27](#_Toc40713871)

[**3. Wymagania aplikacji** 27](#_Toc40713872)

[**4.1 Rejestracja konta użytkownika** 28](#_Toc40713873)

[**4.2 Logowanie użytkownika** 31](#_Toc40713874)

[**4.3 Operowanie kalendarzem** 33](#_Toc40713875)

[**4.4 Otworzenie listy To-Do** 34](#_Toc40713876)

[**4.5 Wybranie dnia kalendarza, dodanie zadania do listy To-Do** 36](#_Toc40713877)

[**4.6 Oznaczenie zadania w liście To-Do jako zrobione/nie zrobione** 39](#_Toc40713878)

[**5. Podsumowanie** 40](#_Toc40713879)

[**29. Zasady wdrażania aplikacji** 41](#_Toc40713880)

# 

# **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. Szczegółowy opis przypadków użycia**

|  |  |
| --- | --- |
| Nazwa | Obsługa kalendarza |
| Nr funkcji | 1 |
| Autor | Mrzygłód |
| Priorytet | Wysoki |
| Aktorzy | Użytkownik |
| Opis | Funkcja polega na obsłudze kalendarza w aplikacji |
| Warunek początkowy | Wyświetlony został kalendarz, aktor wybrał konkretny dzień z kalendarza |
| Główny przepływ zdarzeń | Scenariusz główny - kalendarz |
| Alternatywny przepływ zdarzeń | - |
| Zakończenie | Otworzenie panelu z wybranym dniem z kalendarza |
| Warunek zakończenia | Jeśli wyświetlony zostanie panel z wybranym dniem |
| Rozejścia | Stwórz wydarzenie w kalendarzu |

|  |  |
| --- | --- |
| Nazwa | Stworzenie listy zadań do wykonania |
| Nr funkcji | 3 |
| Autor | Mrzygłód |
| Priorytet | Wysoki |
| Aktorzy | Użytkownik |
| Opis | Funkcja polega na stworzeniu listy zadań, do której będą dodawane zadania do wykonania przez użytkownika |
| Warunek początkowy | Wybranie konkretnego dnia z kalendarza, dla którego ma zostać stworzona lista zadań |
| Główny przepływ zdarzeń | Scenariusz główny - lista zadań |
| Alternatywny przepływ zdarzeń | - |
| Zakończenie | Stworzenie listy zadań |
| Warunek zakończenia | Jeśli zostanie stworzona lista zadań umożliwiająca dodawanie do niej zadań |
| Rozejścia | Dodaj zadanie do listy zadań |

|  |  |
| --- | --- |
| Nazwa | Dodanie zadań do listy zadań |
| Nr funkcji | 4 |
| Autor | Mrzygłód |
| Priorytet | Wysoki |
| Aktorzy | Użytkownik |
| Opis | Funkcja polega na dodaniu zadania do stworzonej wcześniej listy zadań |
| Warunek początkowy | Wybrana została stworzona wcześniej lista zadań |
| Główny przepływ zdarzeń | Scenariusz główny - zadanie do wykonania |
| Alternatywny przepływ zdarzeń | - |
| Zakończenie | Dodanie zadania do listy zadań |
| Warunek zakończenia | Jeśli zadanie zostało poprawnie dodane do listy zadań |
| Rozejścia | Kwalifikacja zadań do kategorii, Nadawanie określonego priorytetu poszczególnym zadaniom,  Dodawanie notatek do zadań,  Dodanie przypomnienia do ważnych/cyklicznych zadań |

|  |  |
| --- | --- |
| Nazwa | Kwalifikacja zadań do kategorii |
| Nr funkcji | 6 |
| Autor | Mrzygłód |
| Priorytet | Średni |
| Aktorzy | Użytkownik |
| Opis | Funkcja polega na zakwalifikowaniu dodanego zadania do konkretnej kategorii |
| Warunek początkowy | Dodane zostało zadanie do listy zadań |
| Główny przepływ zdarzeń | Scenariusz główny - zadania |
| Alternatywny przepływ zdarzeń | - |
| Zakończenie | Zakwalifikowanie zadania do kategorii |
| Warunek zakończenia | Jeśli zadanie zostało poprawnie zakwalifikowane do kategorii |
| Rozejścia | Nadawanie określonego priorytetu poszczególnym zadaniom,  Dodawanie notatek do zadań,  Dodanie przypomnienia do ważnych/cyklicznych zadań |

|  |  |
| --- | --- |
| Nazwa | Nadawanie okreslonego priorytetu zadaniom |
| Nr funkcji | 7 |
| Autor | Mrzygłód |
| Priorytet | Średni |
| Aktorzy | Użytkownik |
| Opis | Funkcja polega na nadaniu zadaniu priorytetu |
| Warunek początkowy | Poprawnie stworzone zadanie |
| Główny przepływ zdarzeń | Główny scenariusz - zadanie |
| Alternatywny przepływ zdarzeń | - |
| Zakończenie | Nadanie priorytetu zadaniu |
| Warunek zakończenia | Jeśli zostanie poprawnie nadany priorytet zadaniu |
| Rozejścia | Kwalifikacja zadań do kategorii,  Dodawanie notatek do zadań,  Dodanie przypomnienia do ważnych/cyklicznych zadań |

|  |  |
| --- | --- |
| Nazwa | Dodawanie notatek do zadań |
| Nr funkcji | 8 |
| Autor | Mrzygłód |
| Priorytet | Niski |
| Aktorzy | Użytkownik |
| Opis | Funkcja polega na dodaniu notatki, opisu do zadania |
| Warunek początkowy | Poprawnie stworzone zadanie |
| Główny przepływ zdarzeń | Główny scenariusz - zadanie |
| Alternatywny przepływ zdarzeń | - |
| Zakończenie | Dodanie notatki do zadania |
| Warunek zakończenia | Jeśli zostanie poprawnie dodana notatka do zadania |
| Rozejścia | Kwalifikacja zadań do kategorii, Nadawanie określonego priorytetu poszczególnym zadaniom,  Dodanie przypomnienia do ważnych/cyklicznych zadań |

|  |  |
| --- | --- |
| Nazwa | Dodanie przypomnienia do ważnych/cyklicznych zadań |
| Nr funkcji | 9 |
| Autor | Mrzygłód |
| Priorytet | Niski |
| Aktorzy | Użytkownik, System |
| Opis | Funkcja polega na ustawieniu systemowego powiadomienia przypominającego o wykonaniu zadania ważnego, bądź powtarzającego się co jakiś okres czasu |
| Warunek początkowy | Poprawnie stworzone zadanie |
| Główny przepływ zdarzeń | Główny scenariusz - zadanie |
| Alternatywny przepływ zdarzeń | - |
| Zakończenie | Dodanie przypomnienia do zadania |
| Warunek zakończenia | Jeśli poprawnie zostało ustawione przypomnienie o ważnym/cyklicznym zadaniu |
| Rozejścia | Kwalifikacja zadań do kategorii, Nadawanie określonego priorytetu poszczególnym zadaniom,  Dodawanie notatek do zadań |

# **13. 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

# **14. 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. |

# **15. 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.





# **16. 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** |

# **17. Biblioteki zewnętrzne używane w aplikacji**

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

Appium – posłuży do zautomatyzowania testów aplikacji mobilnej.

SQLite - lokalna baza danych urządzenia

# **18. 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.

Najważniejszymi częściami aplikacji, którę będą przetesowane będzie jej funkcjonalność. Zakładane jest 80% pokrycie testami funkcjonalności.

Harmonogram tworzenia testów funkcjonalnych.

|  |  |  |
| --- | --- | --- |
| Nr funkcji | Nazwa funkcji | Średni czas tworzenia[h] |
| 1 | Obsługa kalendarza | 8 |
| 2 | Generowanie planu dnia wybranego  w kalendarzu | 10 |
| 3 | Stworzenie listy zadań do wykonania | 6 |
| 4 | Dodanie zadań do listy zadań | 4 |
| 5 | Dodawanie kategorii dla każdego  z zadań | 4 |
| 6 | Kwalifikacja zadań do kategorii | 6 |
| 7 | Nadawanie określonego priorytetu poszczególnym zadaniom | 4 |
| 8 | Dodawanie notatek do zadań | 4 |
| 9 | Dodanie przypomnienia do ważnych/cyklicznych zadań | 6 |
| 10 | Wyświetlenie analizy danego dnia  (lista zadań, czas wolny, czas pracy) | 4 |

W przypadku testów niefunkcjonalności będą użyte w tym przypadku unit testy i zakładane jest 40% pokrycie

Harmonogram tworzenia testów niefunkcjonalności:

|  |  |  |
| --- | --- | --- |
| Nr cechy | Cecha niefunkcjonalna | Średni czas tworzenia[h] |
| 1 | Zmiana kolorystyki aplikacji | 2 |
| 2 | Zmiana czcionki w aplikacji | 2 |
| 3 | Zmiana kolorystyki kategorii | 2 |
| 4 | Zmiana kolorystyki wydarzeń | 2 |
| 5 | Zmiana języka aplikacji | 6 |
| 6 | Modyfikacja powiadomień | 4 |
| 7 | Modyfikacja kalendarza | 2 |
| 8 | Zmiana formatu dat | 4 |
| 9 | Zmiana systemu godzinowego | 4 |
| 10 | Wyświetlenie panelu informacji | 2 |

# **19. 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.

# **20. 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),

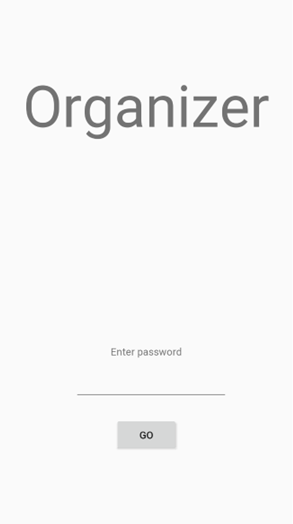
- Na żądanie klienta dodano formularz logowania, oraz dodawanie użytkowników i zabezpieczenie przeciwko botom(01.04.2020r)

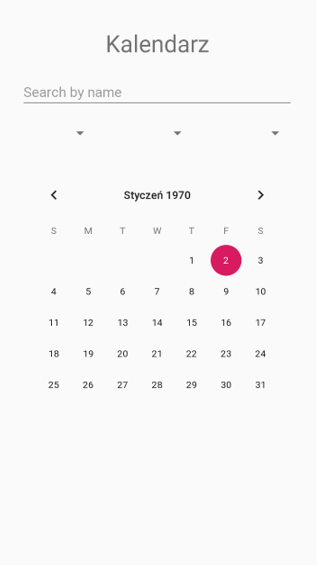
# **21. Diagram Gantta (harmonogram prac)**

Harmonogram prac ujęty jest w pliku akursza Excel dołączonym do dokumentacji - „harmonogram-prac.xlsx”.

# **22.Implementacja interfejsów GUI**

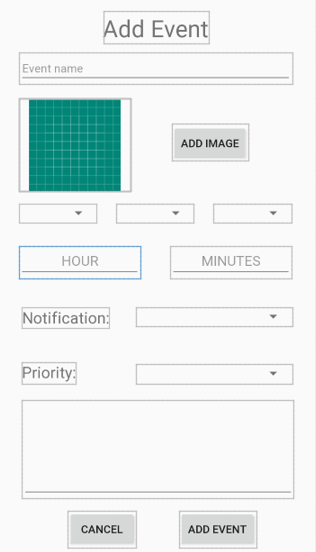
1. główny widok aplikacji oraz widok kalendarza





1. widok kalendarza:

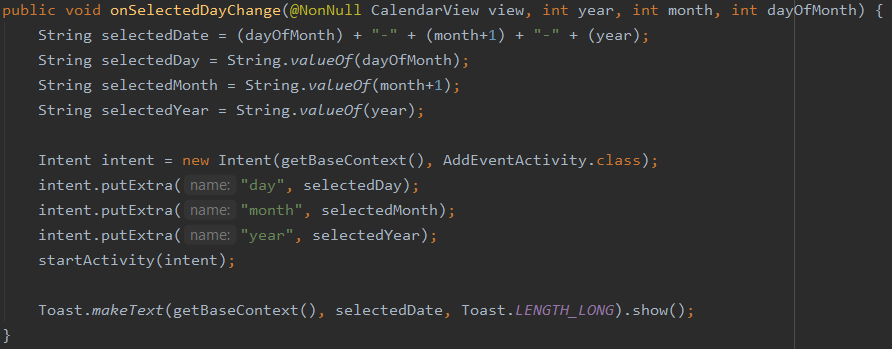
b) widok formularza dodawania zdarzeń:



# **23. 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.



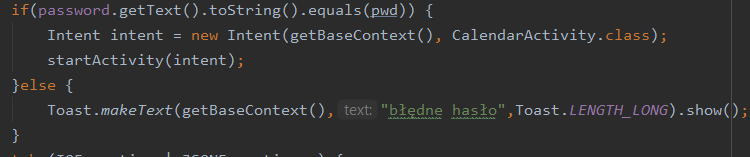
1. Sprawdzanie hasła przy próbie włączenia aplikacji

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.

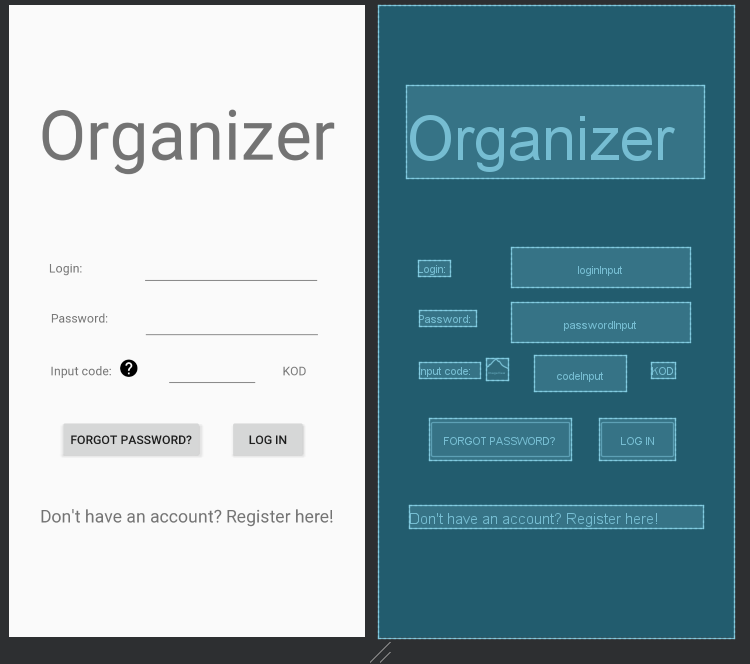


1. Przejście z ekranu startowego do widoku kalendarza

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



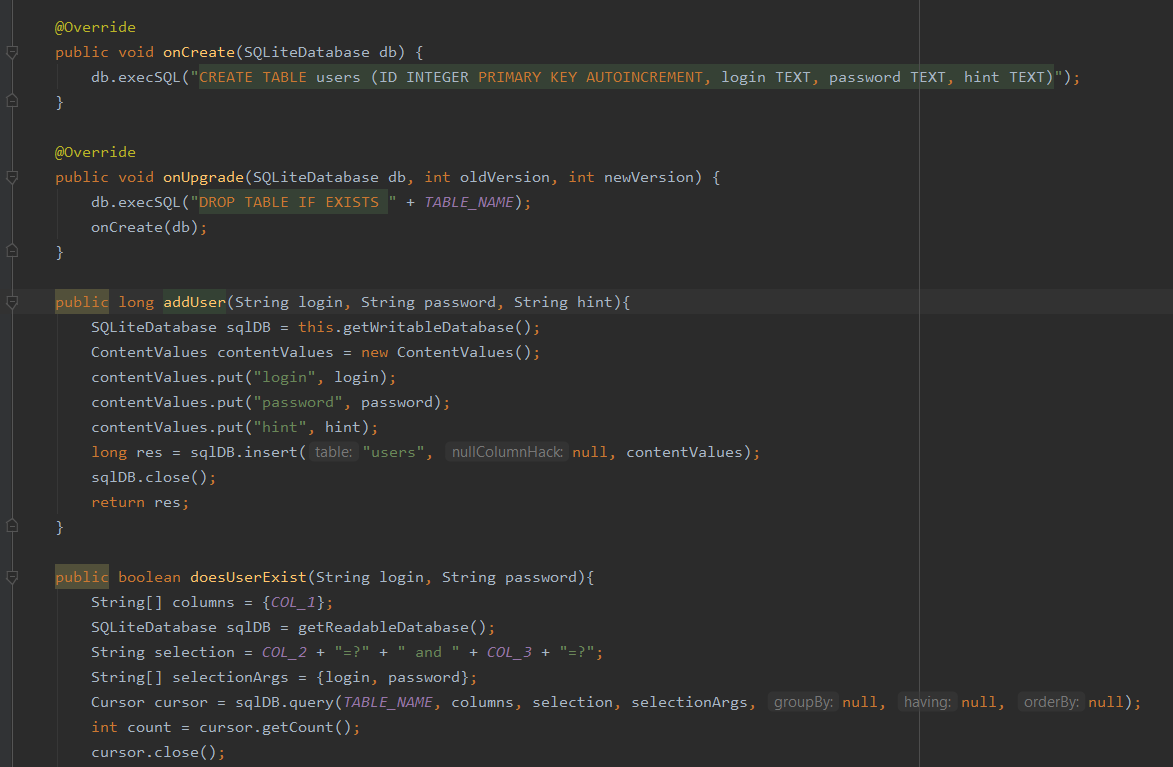
# **24.Dodanie dodatkowego żądanie klienta**

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.

# **25. Zmiana sposobu przechowywania danych**

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

.



Skutkowało to dodaniem klasy DatabaseHelper.

# 

# **26. Aktualna faza implementacji funkcji**

a)tabela

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nr funkcji | Nazwa funkcji | Data rozpoczęcia | Data zakończenia | Wykonawcy | Stopień realizacji 0-100[%] | Status |
| 1 | Rejestracja | 05.04.2020 | 05.04.2020 | Tomasz Patrzałek | 100 | poprawny |
| 2 | Logowanie | 05.04.2020 | 05.04.2020 | Tomasz Patrzałek | 100 | poprawny |
| 3 | Obsługa kalendarza | 06.04.2020 | 07.04.2020 | Tomasz Patrzałek | 70 | nieskończony |
| 4 | Nawigacja pomiędzy oknami | 07.04.2020 | 08.04.2020 | Łukasz Mrzygłód | 50 | nieskończony |
| 5 | Dodanie zadań do listy zadań | 09.04.2020 | - | Mateusz Orelik | 5 | nieskończony |

b)wykres słupkowy



# 

# **27. Założenia dokumentacji użytkownika**

**1. Dla kogo jest dokumentacja użytkownika (dla dziecka, młodzieży, gracza, a może seniora)?**

Dokumentacja jest dla każdego użytkownika, który potrzebuje aplikacji wspomagającej zarządzanie czasem wolnym, niezależnie od jego wieku.

**2. Jaka będzie forma dokumentacji ogólna, fragmentaryczna, szczegółowa, kompleksowa? Proszę opisać i uzasadnić.**

Forma dokumentacji to bedzie dokument PDF zawierający spis treści, oraz poszczególne rozdziały dotyczące funkcjonalności aplikacji. Każdy rozdział bedzie poświęcony szczegółowo określonej funkcjonalności, zawierając opis w formie tekstu, oraz zrzuty ekranu pokazujące obsługę danej funkcji. Taka forma jest według nas najbardziej przyjazna każdemu użytkownikowi.

**3. Jaki zastosujecie język? Młodzieżowy, specjalny, ogólny?**

Zastosujemy język ogólny, bez żadnych specjalistycznych zwrotów i określeń, tak aby każdy użytkownik, niezależnie od poziomu wykształcenia był w stanie zrozumieć opisywane zagadnienia.

**4. Jaka forma dokumentacji? (komiks, audiobook, tutorial video, historyjka),**

Tak jak zostało to opisane w punkcie drugim, będzie to forma dokumentu PDF z rozdziałami (na podobieństwo instrukcji papierowych).

**5. Sposoby wdrażania dokumentacji (załącznik do aplikacji, dodatkowa ulotka, adres www, …)**

Dokumentacja będzie dołączana do aplikacji instalowanej na urządzeniu mobilnym, ewentualnie w aplikacji będzie znajdować się odnośnik do miejsca, z której ową dokumentację można pobrać.

**6. Kto z zespołu przygotuje dokumentacje?**

Lider zespołu zadecyduje na etapie tworzenia dokumentacji

**7. Jakie założenia czasowe – np. 50h roboczych, max.**

Około 40 godzin roboczych.

**8. Na jakim poziomie będzie szkolony użytkownik? Czy należy takie szkolenie przeprowadzić?**

Użytkownik zostanie zapoznany z każdą funkcjonalnością w szczegółowej formie, tak, aby nie miał wątpliwości jakie możliwości posiada aplikacja.

# **28. Instrukcja użytkownika**

# **1. Wprowadzenie**

Poniższa dokumentacja jest instrukcją użytkownika, przedstawiającą wybrane funkcjonalności aplkacji. Ma ona na celu przestawić szczegółowy opis oraz przedstawienie krok po kroku, jak wykorzystać wybrane funkjonalności.

# **2. Opis aplikacji**

Aplikacja „Organizer App” to kompaktowa aplikacja mobilna oferująca wsparcie w rozsądnym gospodarowaniu czasem użytkownika. Funkcje, które zostaną przedstawione w instrukcji to:

- rejestracja konta użytkownika do lokalnej bazy danych urzączenia

- logowanie na poprzednio utworzone konto w lokalnej bazie danych urządzenia

- operowanie kalendarzem (zmiana miesiąca, roku, wybranie dnia)

- możliwość otworzenia listy „To-Do” - rzeczy do zrobienia

- możliwość wybrania dnia z kalendarza, po czym ukazuje się panel dodania pozycji do listy „To-Do”

- możliwość oznaczenia pozycji w liście „To-Do” jako wykonane, bądź niewykonane

# **3. Wymagania aplikacji**

**Wymagania minimalne:**

**-** urządzenie mobilne z systemem Android minimum 4.0 (Ice Cream Sandwich)

- 50 MB wolnej przestrzeni w pamięci urządzenia

**Wymagania optymalne:**

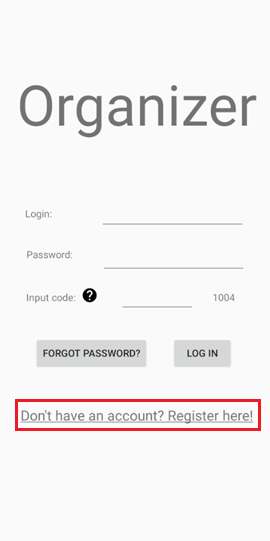
- urządzenie mobilne z systemem Android 8.0 (Oreo) bądź wyższym

- 200 MB wolnej przestrzeni w pamięci urządzenia

**4. Funkcjonalności aplikacji**

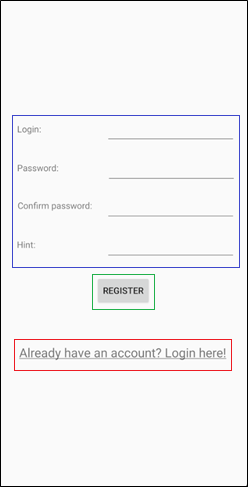
## **4.1 Rejestracja konta użytkownika**

Po włączeniu aplikacji użytkownikowi wyświetla się poniższy ekran startowy umożliwiający zalogowanie do aplikacji.



Aby przejść do ekranu umożliwiającego rejestrację, należy stuknąć pole zaznaczone na czerwono na powyższym zrzucie ekranu.

Po stuknięciu w przedstawioną opcję, użytkownikowi wyświetla się poniższy ekran umożliwiający rejestrację konta użytkownika w lokalnej bazie danych urządzenia mobilnego.



Na przedstawionym powyżej ekranie rejestracji znajdują się następujące elementy interfejsu umożliwiające interakcję użytkownika:

**Formularz rejestracji** - zawiera pola, w których użytkownik wpisuje kolejno:

Login - nazwa użytkownika służąca do logowania np. jarek123

Password - hasło użytkownika służace do logowania np. jarekhaslo123

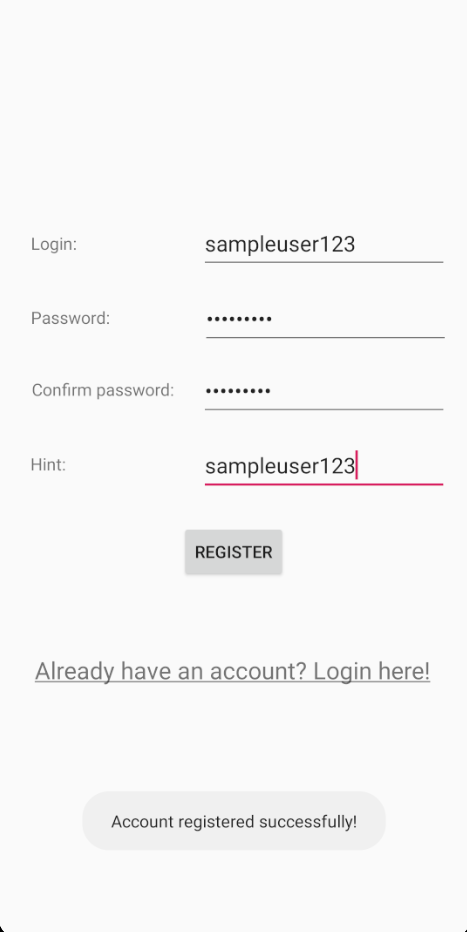
Confirm password - pole na potwórne wpisanie hasła wpisanego w polu Password w celu weryfikacji podanego hasła, oba hasła muszą być identyczne

Hint - jest to podpowiedź wybierana przez użytkownika, która ma pomóc w przypomnieniu hasła w razie jego zapomnienia

**Przycisk „REGISTER”** - przycisk, który należy nacisnąć po wpisaniu wszystkich niezbędnych danych w formularzu rejestracji, aby zakończyć proces rejestracji i stworzyć konto użytkownika.

**Pole powrotu do logowania** - pole, które należy nacisnąć jeśli chce się wrócić do panelu logowania

Poniżej przykładowo wykonany proces rejestracji użytkownika



Z powyżej wpisanymi danymi konto zostało pomyślnie utworzone.

## **4.2 Logowanie użytkownika**

Ekran logowania wygląda następująco.

**Formularz logowania** - zawiera pola, w których użytkownik wpisuje następująco:

Login - nazwa użytkownika wybrana podczas procesu rejestracji

Password - hasło użytkownika wybrane podczas procesu rejestracji

Input code - pole, w które należy przepisać cztero-cyfrowy kod ukazany na prawo od pola wpisywania (w tym przypadku „1004”)

- ikona, po kliknięciu której wyświetli się podpowiedź, o przepisaniu

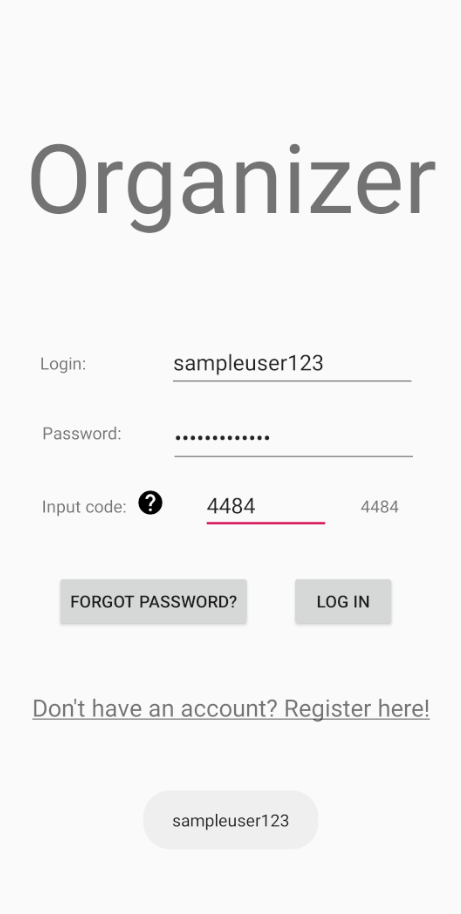
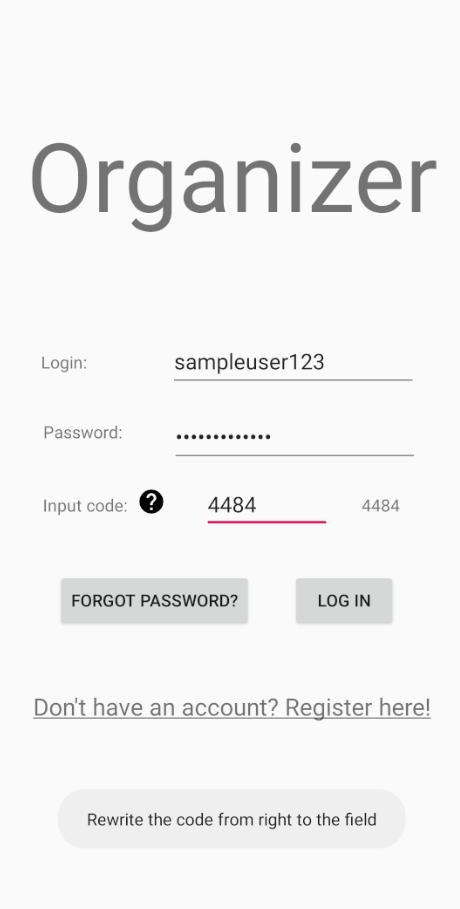
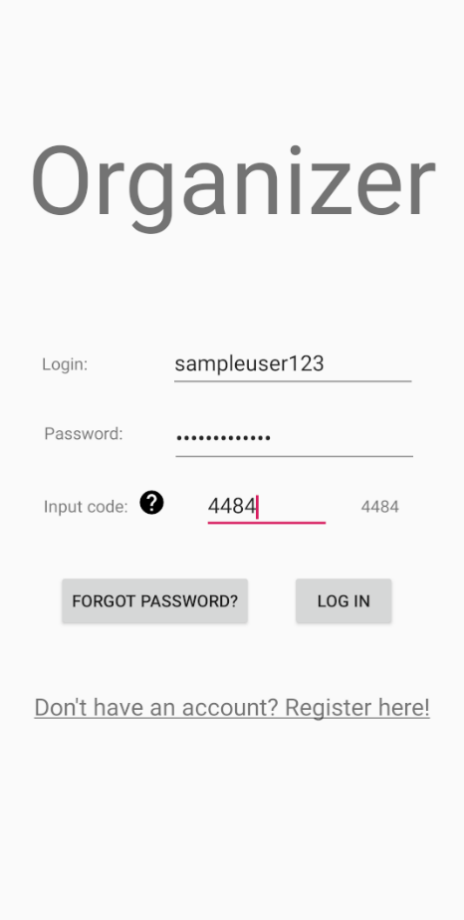
cztero-cyfrowego kodu z prawej strony (w tym przypadku „1004”)

**Przycisk „FORGOT PASSWORD”** - przycisk, po wciśnięciu którego wyświetla się podpowiedź do hasła, dla nazwy użytkownika wpisanego w polu Login

**Przycisk „LOGIN”** - przycisk, po wciśnięciu którego następuje weryfikacja wpisanych powyżej danych i logowanie użytkownika

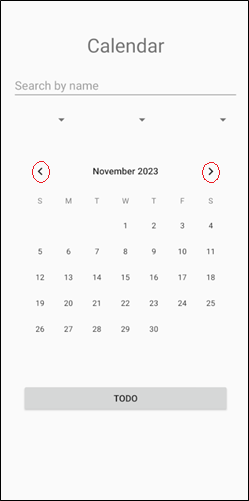
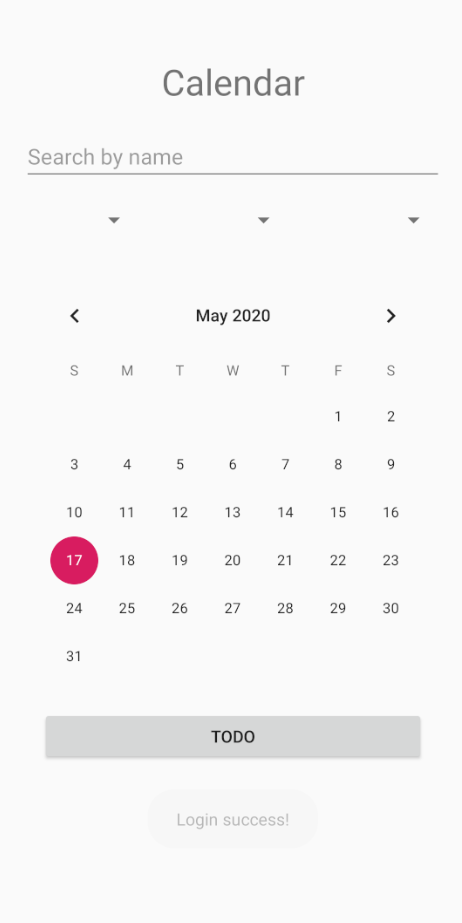
**Pole przejścia do panelu rejestracji** - pole, po wciśnięciu którego następuje przekierowanie użytkownika do wcześniej wspomnianego pola rejestracji

Poniżej przykładowo wykonany proces logowania, oraz interakcja z elementami pomocniczymi interfejsu logowania, kolejno: wypełniony formularz, wciśnięta ikona „?”, oraz wciśnięty przycisk „forgot password”.



## **4.3 Operowanie kalendarzem**

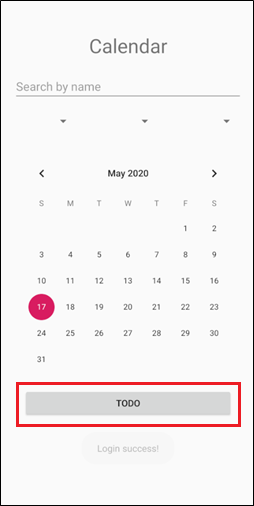
Po poprawnym zalogowaniu użytkownikowi wyświetla się ekran z kalendarzem, którym może operować. Może wybrać dowolny dzień, miesiąc rok. Poniżej prezentacja ekranu z kalendarzem, wyświetlony zaraz po zalogowaniu, oraz ekran ze zmienionym prezentowanym miesiącem oraz rokiem.



Strzałkami zaznaczonymi na zrzucie ekranu powyżej, po prawej stronie, możemy manipulować wyświetlanym miesiącem wyświetlonego roku.

## **4.4 Otworzenie listy To-Do**

Ekran kalendarza pozwala użytkownikowi otworzyć ekran z listą zadań do wykonania (To-Do).



**Przycisk TODO** - przy pomocy tego przycisku użytkownik może przejść do ekranu zawierającego listę zadań do wykonania, które uprzednio użytkownik dodał, korzystając z funkcji przedstawionej w następnym podrozdziale.



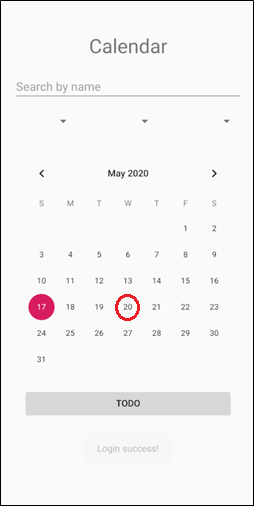
Powyższy ekran zawiera listę zadań do wykonania dodanych przez użytkownika.

**Przycisk LOG OUT** - po wciśnięciu tego przycisku, użytkownik przechodzi do ekranu logowania, wylogowując się ze swojego konta

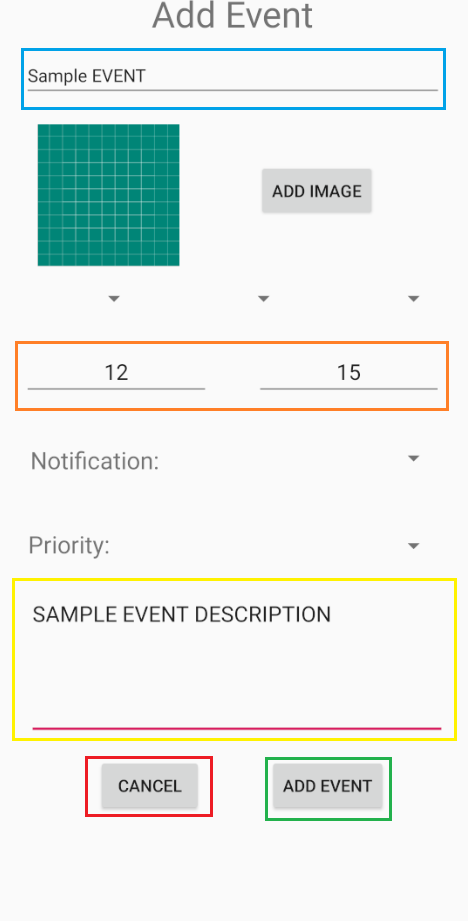
**Przycisk CALENDAR** - po wciśnięciu tego przycisku użytkownik wraca do ekranu z kalendarzem

## **4.5 Wybranie dnia kalendarza, dodanie zadania do listy To-Do**

Na ekranie kalendarza, użytkownik może wybrać i kliknąć konkretny dzień tygodnia, aby przejść do ekranu dodawania zadani do listy To-Do.



Po wciśnięciu np. dnia zaznaczonego na powyższym ekranie, pokaże się ekran dodawania zadania do listy ToDo

Ekran umożliwiający dodanie zadania do listy To-Do na ten moment umożliwia użytkownikowi wpisanie nazwy, godziny, minut oraz opisu zadania.

**EVENT NAME** - nazwa, tytuł dodawanego zadania

**HOUR/MINUTES** - godzina oraz minuta zadania

**SAMPLE EVENT DESCRIPTION** - opis dodawanego zadania

**PRZYCISK CANCEL** - cofa użytkownika do ekranu z kalendarzem

**PRZYCISK ADD EVENT** - dodaje zadanie do listy To-Do

Po dodaniu zadania, możemy wejść do ekranu listy To-Do, wytłumaczonego w poprzednim podrozdziale, oraz sprawdzić dodanie zadanie.



## **4.6 Oznaczenie zadania w liście To-Do jako zrobione/nie zrobione**

Ekran listy To-Do umożliwia użytkownikowi zaznaczenie zadania jako wykonane, po prostu poprzez kliknięcie w dane zadanie. Poniżej wyświetlony przykład zadania wykonaneg i niewykonanego



# **5. Podsumowanie**

Powyższa instrukcja obsługi ukazuje szczegółowy opis wybranych funkcji aplikacji. Elementy przedstawianych funkcjonalności, które nie są opisane, są niedostępne dla użytkownika w aktualnym stadium rozwoju aplikacji.

# **29. Zasady wdrażania aplikacji**

**Wymagania minimalne:**

- urządzenie mobilne z systemem Android minimum 4.0 (Ice Cream Sandwich)

- 50 MB wolnej przestrzeni w pamięci urządzenia

**Wymagania optymalne:**

- urządzenie mobilne z systemem Android 8.0 (Oreo) bądź wyższym

- 200 MB wolnej przestrzeni w pamięci urządzenia

**Instalacja aplikacji:**

W aktualnej fazie implementacji wymagane jest posiadanie oprogramowania Android Studio, by móc zainstalować aplikacje na urządzeniu mobilnym, ale przewidywane jest udostępnienie aplikacji z postaci pliku .apk do pobrania i zainstalowania na urządzeniu.

**Wymagane szkolenie:**

Do użytku aplikacji wystarczy szkolenie w postaci dogłębnego zaznajomienia się z dokumentacją użytkownika