

Plan Projektu – Programowanie obiektowe

1.Opis Aplikacji

Proponowana przez nas aplikacja to tzw. „Bill Reminder”. W obecnych czasach ze względu na mnogość usług z których korzystamy pamiętanie o wszystkich, to nie lada wyzwanie. Aplikacja przypominająca o rachunkach w naszej wizji zawierać powinna bazę danych, w której zapisywane będą dane o rachunkach takie jak:

- Nazwa rachunku
- Opis rachunku
- Cena rachunku
- Termin płatności (Z wyborem okresu płatności, rachunek cotygodniowy, miesięczny, półroczny etc.)
- Informacja o opłaceniu rachunku (BillPaid)

2.Wykorzystywane biblioteki i Frameworki, szczegóły

Do utworzenia bazy danych planujemy wykorzystać bibliotekę taką jak Room(Rekomendowana przez Google), bądź Requery. Ponadto planowane jest użycie Frameworku Dagger2 w celu wstrzykiwania zależności takich jak w/w baza danych. Możliwe jest także wykorzystanie bibliotek do obsługi np. zdjęć typu Glide, bądź Lottie do animacji w celu poprawy wyglądu interfejsu użytkownika. Wybrany do aplikacji Design Pattern to MVP. Możliwe będzie też zapisanie wszystkich rachunków do bazy danych i przywrócenie jej, w razie np. zmiany telefonu.

3.Planowane w aplikacji aktywności i fragmenty.

Działając zgodnie z zaleceniami Google planujemy, by aplikacja posiadała jedną aktywność, zaś pozostałe widoki były jej fragmentami korzystającymi z kontenera aktywności. Dopuszczalna jest druga aktywność będąca ekranem ładowania podczas uruchomienia aplikacji tzw. „Splash Screen” podczas działania tej aktywności będą przyznawane uprawnienia aplikacji do określonych danych. Rachunki zostaną przedstawione na liście (RecyclerView), w ekranie głównym planowana jest tylko nazwa + ewentualne logo usługi.

Planowane Aktywności :

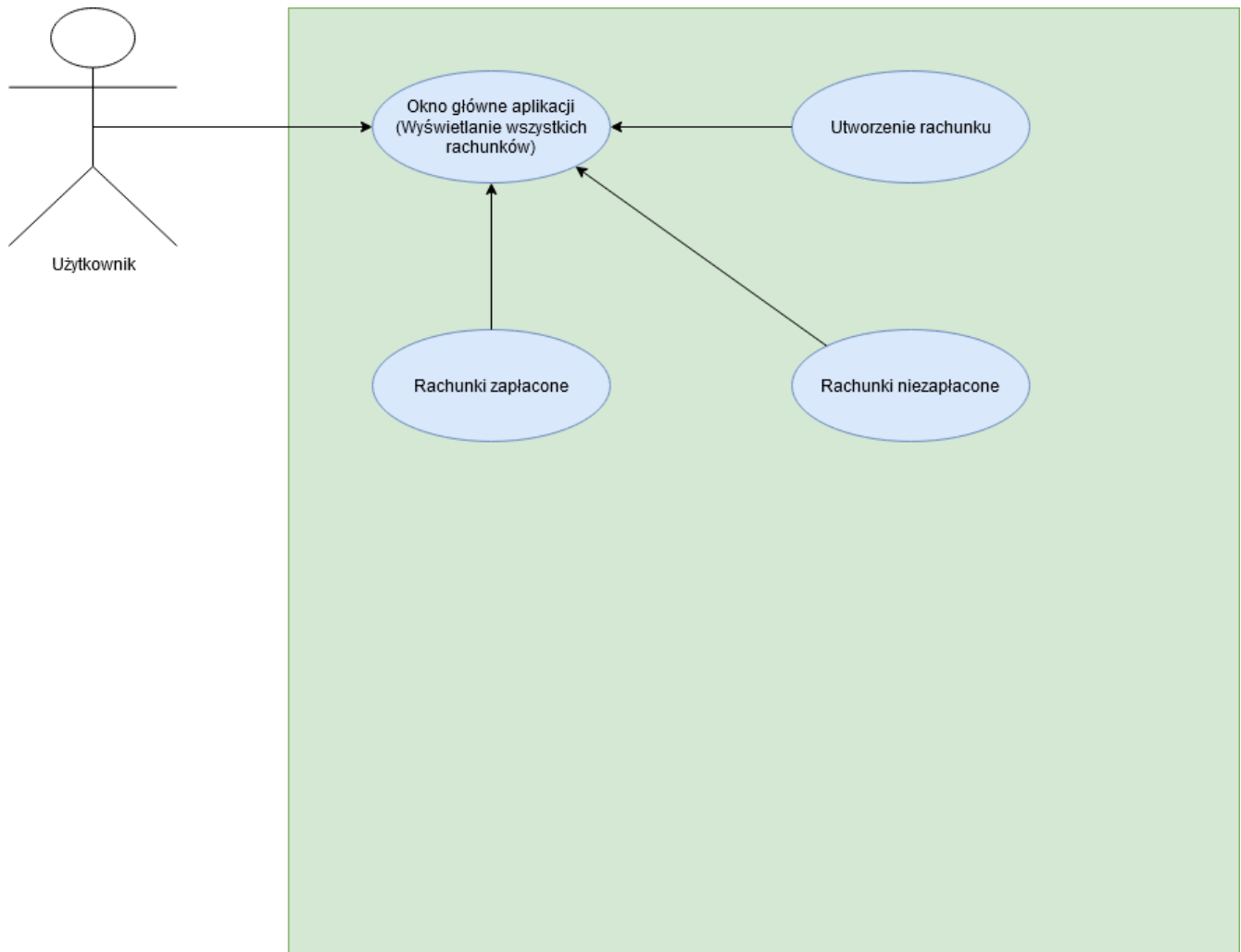
- — Splash Screen
- Aktywność Główna (W tym wyświetlanie wszystkich rachunków wraz z ich informacjami)

Planowane Fragmenty:

- Dodawanie rachunku
- Rachunki opłacone
- Rachunki nieopłacone

4. Diagram UML wykorzystanych klas (założenie):

- MainActivity (Główne okno aplikacji, wyświetla wszystkie rachunki – oraz ich zawartości po naciśnięciu na dany rachunek)
- AddBill (Dodaje rachunek, nazwa rachunku, cena, cykl płacenia i czy jest powtarzalny)
- ~~BillDetails (Zmienia wartości rachunku lub usuwa go)~~
- BillPaid (Wyświetla wszystkie zapłacone rachunki)
- AllBills (Wyświetlanie wszystkich rachunków)
- BillUnPaid (Wyświetla wszystkie niezapłacone rachunki)
- DatabaseHelperImp (Implementacja bazy danych)



Wojciech Gródecki 238459

