

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

Planowane Fragmenty:

- Dodawanie rachunku
- Detale rachunku

- Rachunki opłacone
- Rachunki nieopłacone
- Wszystkie Rachunki

4. Diagram UML wykorzystanych klas (założenie):

- MainActivity (Główne okno aplikacji, wyświetla rachunki)
- BillAdd (Dodaje rachunek)
- BillDetails (Zmienia wartości rachunku lub usuwa go)
- AllBills (Sortowanie rachunków, pokaż wszystkie rachunki, pokaż tylko te które są opłacone (BillPaid = 1) lub pokaż nieopłacone (BillPaid = 0))
- AppDatabase (Zapis do bazy danych lub przywrócenie z bazy danych)

UML Klas

