

2015

# Dokument detaliczny projektu

Sporządził:

*Pietrzak Andrzej*

# SPIIS TREŚCI

## 1) Wprowadzenie.

❖ Cel .....	4
❖ Zakres .....	4
❖ Definicje, akronimy i skróty .....	4
❖ Odsyłacze .....	5
❖ Omówienie .....	5

## 2) Standardy projektu, konwencje, procedury.

❖ Standardy projektowe .....	6
❖ Standardy dokumentacyjne .....	6
❖ Konwencje nazewnictwa .....	6
❖ Standardy programistyczne .....	7
❖ Narzędzia rozwijania oprogramowania .....	7

## 3) Specyfikacja komponentów.

❖ Komponent 1 - RankingView .....	8
❖ Komponent 2 - GameView .....	8
❖ Komponent 3 – sqlAdapter .....	9
❖ Komponent 4 – CreatePlayers .....	9
❖ Komponent 5 – CreatePlayersAvatar .....	9
❖ Komponent 6 – ChooseProfile .....	10
❖ Komponent 7 - AboutView .....	10
❖ Komponent 8 - QuestionData .....	11
❖ Komponent 9 - SaveResult .....	11

**4) Załączniki**

❖ Harmonogram prac (diagram Gantta) .....	12
❖ Konceptualny diagram klas.....	13
❖ Fizyczny diagram klas .....	14
❖ Diagram przypadków użycia .....	15
❖ Interfejs aplikacji.....	15

# Wprowadzenie

## I. Cel

Niniejszy dokument ma za zadanie sprecyzować sposób realizowanych prac. Określić założenia projektu, standardy, narzędzia i komponenty wchodzące w skład implementacji oraz opis realizacji tych komponentów.

## II. Zakres

Założeniem projektu jest uzyskanie grywalnej wersji aplikacji, której trudność rośnie wraz z kolejnymi progami gwarantowanymi.

Gra będzie odmianą popularnej produkcji telewizyjnej „Milionerzy”, gdyż będzie oparta o zasady wymienionego teleturnieju. Aplikacja ma za zadanie umożliwić osobom zainteresowanym tego typu rozrywką dobrą zabawę poprzez sprawdzenie własnej wiedzy z różnych dziedzin. Gra będzie aplikacją mobilną, działającą pod systemem Android w wersji 4.0.3 Ice Cream Sandwich lub wyższej, jej tytuł to "**Milionerki**".

## III. Definicje, akronimy i skróty.

- |                          |  |
|--------------------------|--|
| <b>Baza pytań</b>        | – Baza danych, z której pobierane będą pytania do aplikacji.   |
| <b>Próg gwarantowany</b> | – Wynik osiągany po poprawnej odpowiedzi odpowiednio na 5. i 10. pytanie, po jego osiągnięciu gracz nie może uzyskać mniejszego wyniku, pomimo pomyłki w którymś z kolejnych pytań i w efekcie przegraniu gry. |
| <b>Prowadzący</b>        | – Wirtualna osoba odpowiedzialna za sprawdzanie poprawności odpowiedzi udzielonych przez gracza.   |
| <b>Gracz</b>             | – Osoba korzystająca z aplikacji.  |

## **IV. Odsyłacze**

1063-2001 – IEEE Standard for Software User Documentation

## **V. Omówienie**

Dokument ten powstał na bazie Specyfikacji wymagań. Zawiera on definicje standardów, strategii i konwencji które będą przestrzegane podczas realizacji projektu. Dalsza część dokumentu zawiera informacje o modułach i komponentach systemu i interfejsie graficznym aplikacji.

# Standardy projektu, konwencje, procedury

## **I. Standardy projektowe**

Podczas wykonywania projektu wykorzystywany jest model kaskadowy tworzenia oprogramowania. Został on wybrany ze względu na:

- 1) Kontrolę kosztów.
- 2) Kontrolę jakości.
- 3) Widoczność procesu.

## **II. Standardy dokumentacyjne**

Wszystkie dokumenty są tworzone na podstawie jednego szablonu, natomiast diagramy są sporządzane wg standardu UML. Podczas pisania kodu programu programista będzie stosować komentarze JavaDoc pozwalające na proste wygenerowanie czytelnej dokumentacji projektu. Specyfikacja wymagań jest zgodna ze standardem IEEE 1063-2001.

## **III. Konwencje nazewnnicze**

Nazewnictwo zastosowane w projekcie jest ukierunkowane na jednoznaczność i prostotę. Komponenty, jak i poszczególne funkcje są nazywane w języku angielskim, w ten sposób by można było jednoznacznie odczytać cel danej funkcji czy komponentu. Wersje dokumentów posiadają historie zmian oraz datę wykonania, nazywane są w następujący sposób: nazwa dokumentu ver X, gdzie X to numer wersji (począwszy od 1, zwiększany zgodnie z panującymi standardami w zależności od dokonanych zmian w dokumencie).

## **IV. Standardy programistyczne**

Programista zastosuje podejście obiektowe do programowania, co ułatwia pisanie kodu, jego zrozumienie oraz zastosowanie ewentualnych zmian.

## **V. Narzędzia rozwijania oprogramowania**

Projekt zostanie napisany w programie Android Studio w wersji 1.1 na system operacyjny Android w wersji 4.0.3 Ice Cream Sandwich.

Podczas tworzenia projektu będą wykorzystywane także inne programy m.in.:

- 1) Visual Paradigm – program do tworzenia diagramów UML,
- 2) Microsoft Office w wersji 2007 – wykorzystany przy tworzeniu dokumentacji,
- 3) GanttProject – program do tworzenia harmonogramów,
- 4) Adobe Photoshop CS6 – program do edycji grafiki, wykorzystywany przez grafika.

# Specyfikacja komponentów

Identyfikator komponentu 1	
Typ	<b>RankingView</b> wyświetlający wyniki graczy
Cel	Wykorzystanie komponentu do prezentacji danych pobranych z bazy
Funkcja	Prezentacja osiągniętych przez graczy wyników
Komponenty podporządkowane	ListView
Zależności	DrawerItemCustomAdapter sqlAdapter
Interfejsy	View
Zasoby	Obrazki znajdujące się w folderze drawable

Identyfikator komponentu 2	
Typ	<b>GameView</b> wyświetlający widok gry
Cel	Wykorzystanie komponentu do prezentacji pytań oraz oprawy graficznej, tworzącej klimat gry
Funkcja	Sprawdzanie poprawności udzielonych przez gracza odpowiedzi oraz wyliczanie wyniku gracza
Komponenty podporządkowane	ListView TextView Dialog Button
Zależności	DrawerItemCustomAdapter QuestionsData SaveResult
Interfejsy	View
Zasoby	Obrazki znajdujące się w folderze drawable



Identyfikator komponentu 3	
Typ	<b>sqlAdapter</b> łączący się z bazą danych
Cel	Wykorzystanie komponentu do połączenia z bazą danych
Funkcja	Pobranie pytania lub wyniku gracza, bądź zapisanie wyniku
Komponenty podporządkowane	--
Zależności	SQLiteDatabase
Interfejsy	--
Zasoby	DatabaseHelper

Identyfikator komponentu 4	
Typ	<b>CreatePlayers</b> tworzący profil gracza
Cel	Wykorzystanie komponentu do stworzenia profilu gracza
Funkcja	Stworzenie nowego profilu gracza
Komponenty podporządkowane	TextView EditText
Zależności	sqlAdapter
Interfejsy	View
Zasoby	Obrazki znajdujące się w folderze drawable

Identyfikator komponentu 5	
Typ	<b>CreatePlayersAvatar</b> pozwalający graczowi wybrać avatar, który będzie skojarzony z jego profilem
Cel	Wykorzystanie komponentu do wyboru avatara gracza
Funkcja	Wybór, spośród dostarczonych przez developera, avatara przez gracza, który będzie wyświetlany wraz z imieniem, nazwiskiem i wynikiem gracza, w zestawieniu najlepszych wyników
Komponenty podporządkowane	TextView
Zależności	--
Interfejsy	View
Zasoby	Obrazki znajdujące się w folderze drawable

Identyfikator komponentu		6
Typ		<b>ChooseProfile</b> pozwalający graczowi wybrać profil, który będzie przez niego używany
Cel		Wykorzystanie komponentu do wyboru profilu gracza
Funkcja		Wybór, spośród stworzonych przez gracza, profilu, z którym gracz będzie utożsamiany podczas trwania rozgrywki. Wyniki przez niego osiągnane, będą zapisywane poprzez użycie danych profilu
Komponenty podporządkowane		TextView ListView
Zależności		DrawerItemCustomAdapter sqlAdapter ObjectDrawerItem
Interfejsy		View
Zasoby		Obrazki znajdujące się w folderze drawable

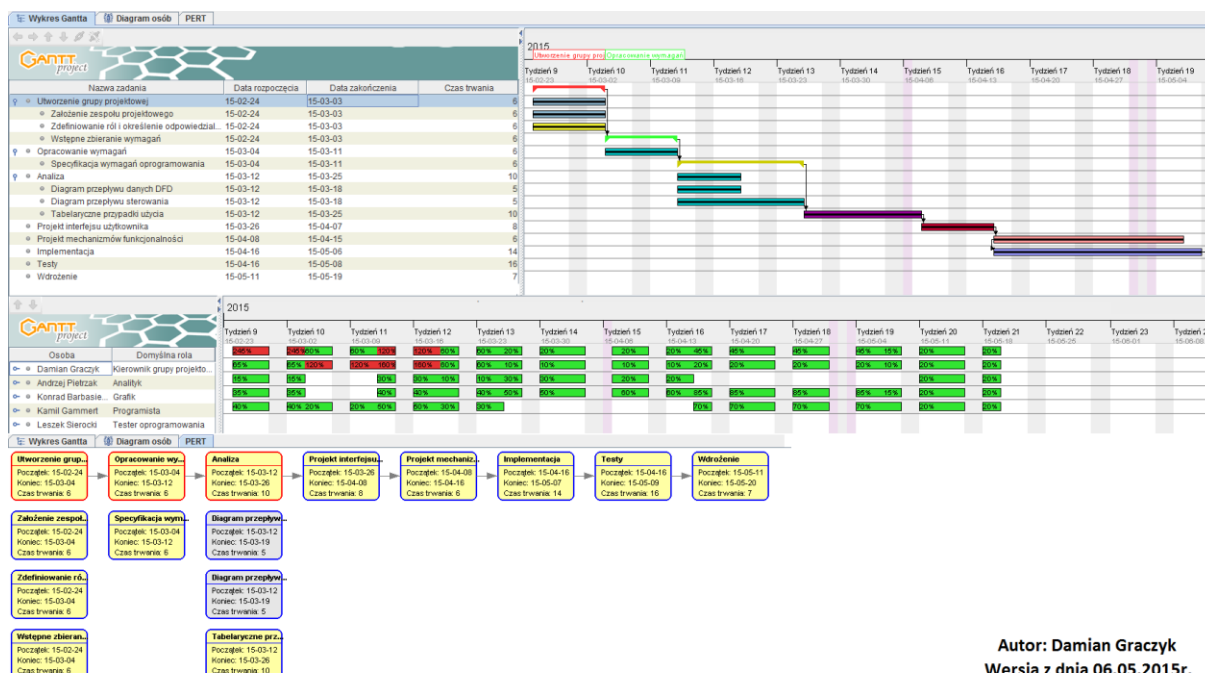
Identyfikator komponentu		7
Typ		<b>AboutView</b> wyświetlający informacje dotyczące gry
Cel		Wykorzystanie komponentu do prezentacji informacji
Funkcja		Przedstawienie informacji dotyczących systemu rozgrywki, wersji oprogramowania oraz danych autorów
Komponenty podporządkowane		--
Zależności		--
Interfejsy		View
Zasoby		Obrazki znajdujące się w folderze drawable

Identyfikator komponentu 8	
Typ	<b>QuestionData</b> odpowiadający za losowanie pytania
Cel	Wykorzystanie komponentu do losowania pytania i sprawdzenia poprawności odpowiedzi
Funkcja	Losowanie pytania dla gracza oraz sprawdzenie poprawności odpowiedzi, której udzielił
Komponenty podporządkowane	--
Zależności	sqlAdapter ObjectDrawerItem GameView
Interfejsy	--
Zasoby	Pytania pobrane z sqlAdapter

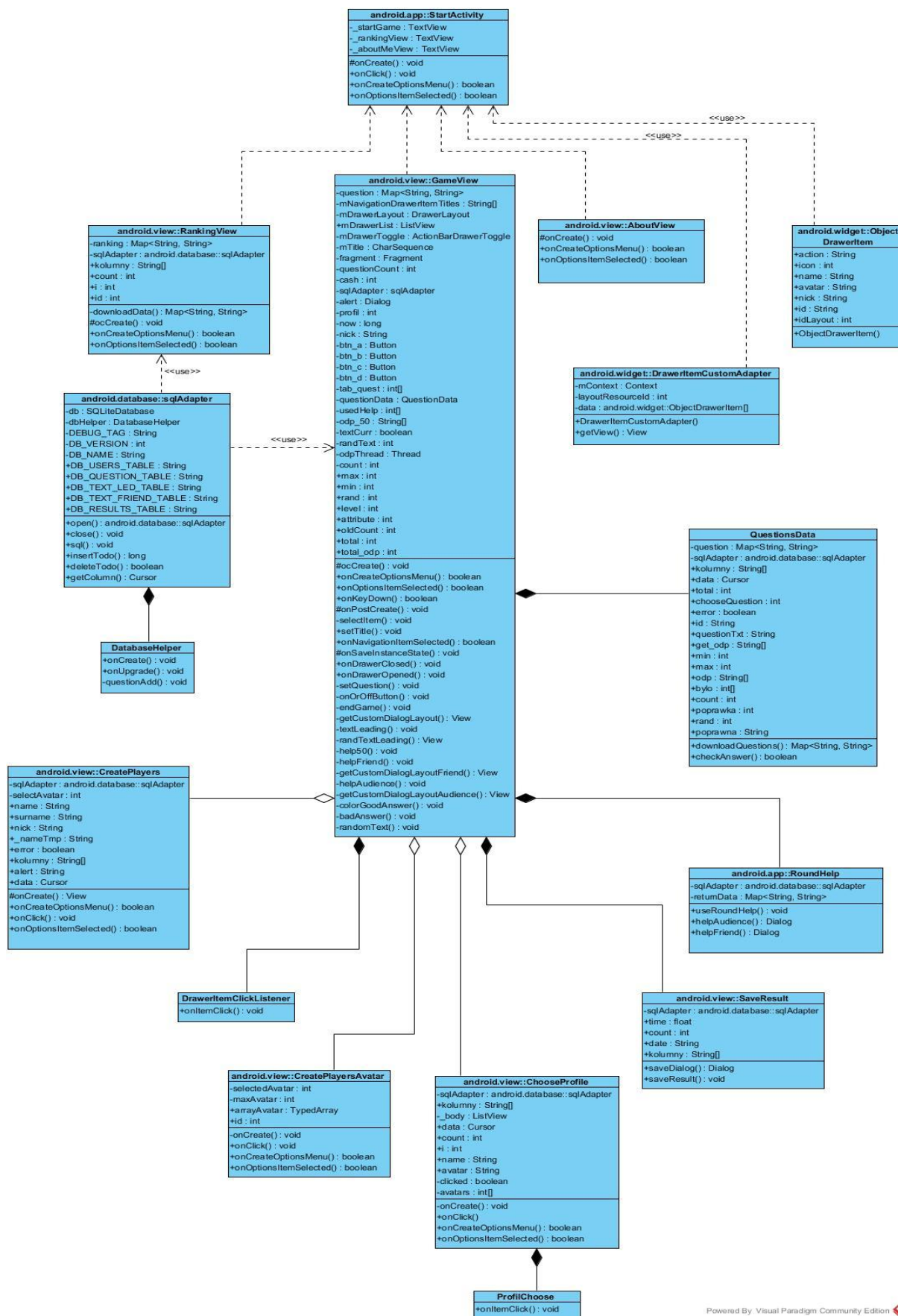
Identyfikator komponentu 9	
Typ	<b>SaveResult</b> pozwalający graczowi zapisać wynik
Cel	Wykorzystanie komponentu do zapisania wyniku gracza po zakończeniu gry
Funkcja	Wyświetlenie okna dialogowego pozwalającego graczowi zapisać uzyskany wynik
Komponenty podporządkowane	Dialog
Zależności	sqlAdapter GameView
Interfejsy	--
Zasoby	Uzyskany wynik pobrany z GameView

# Załączniki

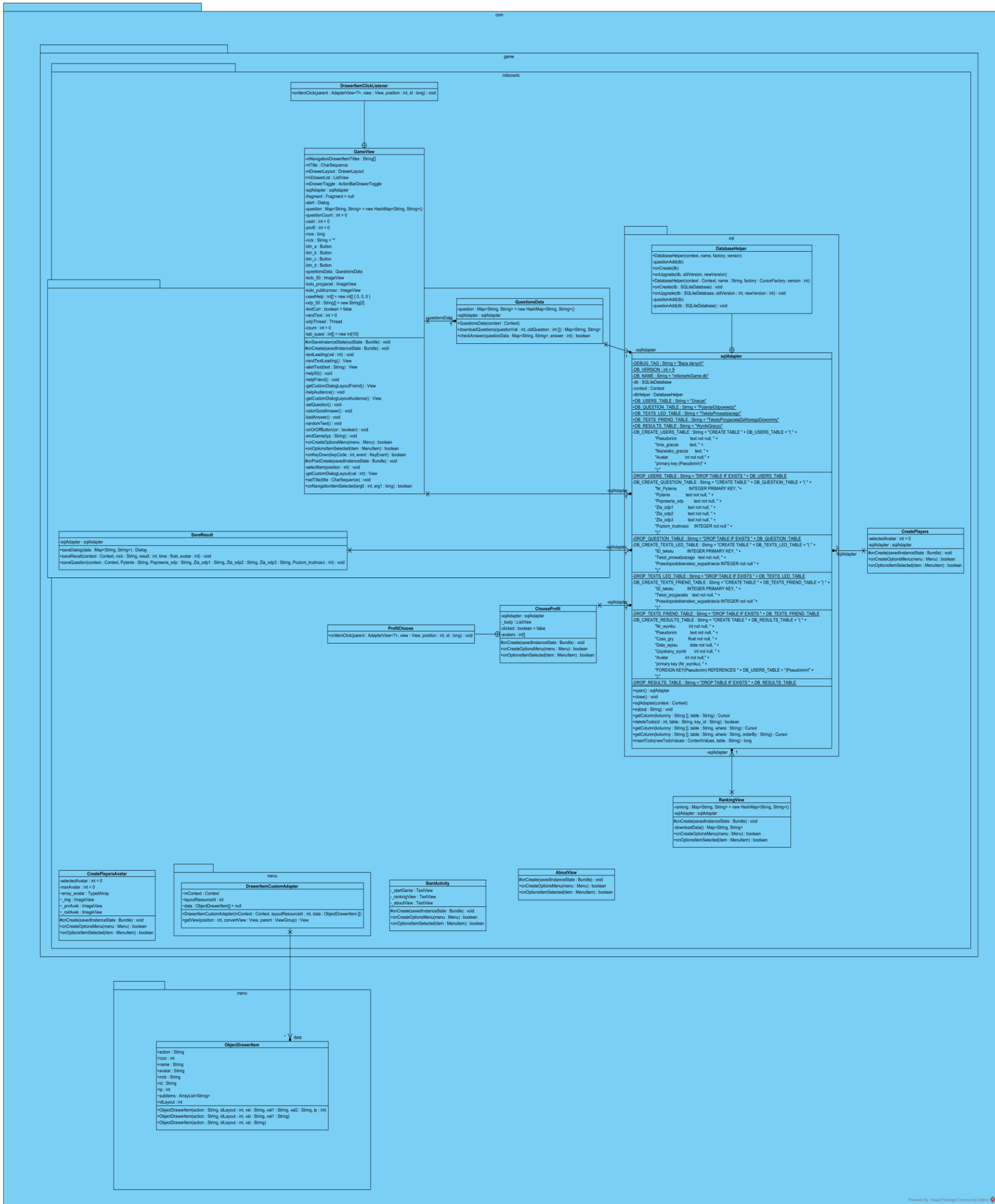
## I. Harmonogram prac (diagram Gantt)



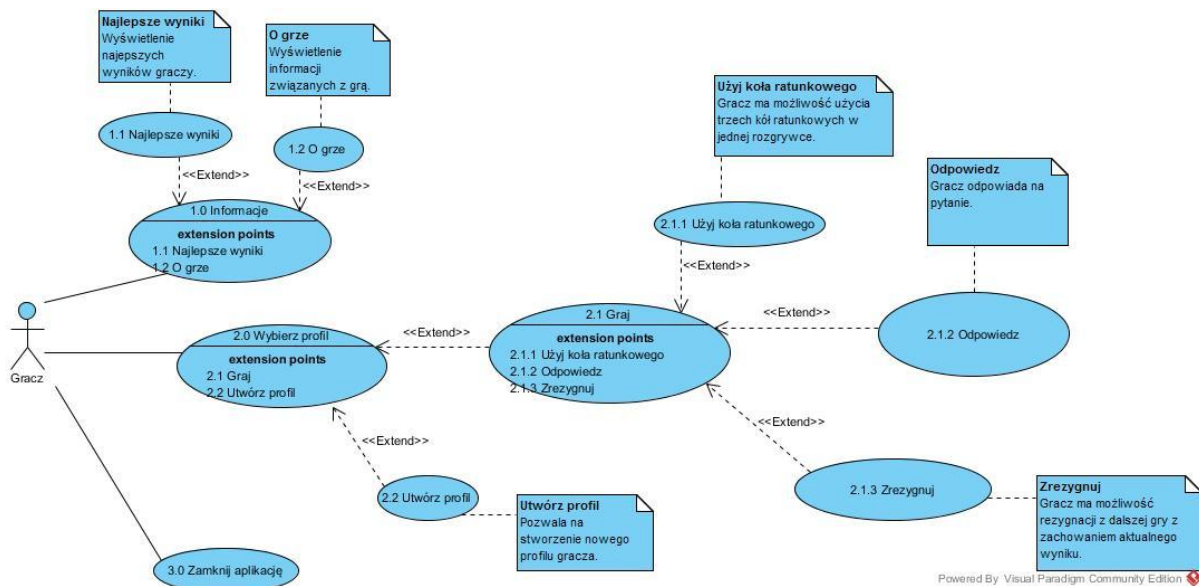
## II. Konceptualny diagram klas



### III. Fizyczny diagram klas



## IV. Diagram przypadków użycia



## V. Interfejs aplikacji

### 1) Widok po uruchomieniu gry

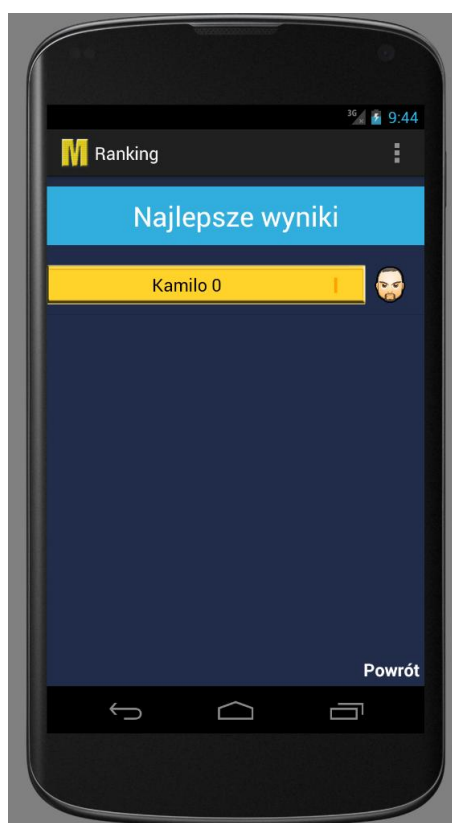




## 2) O grze

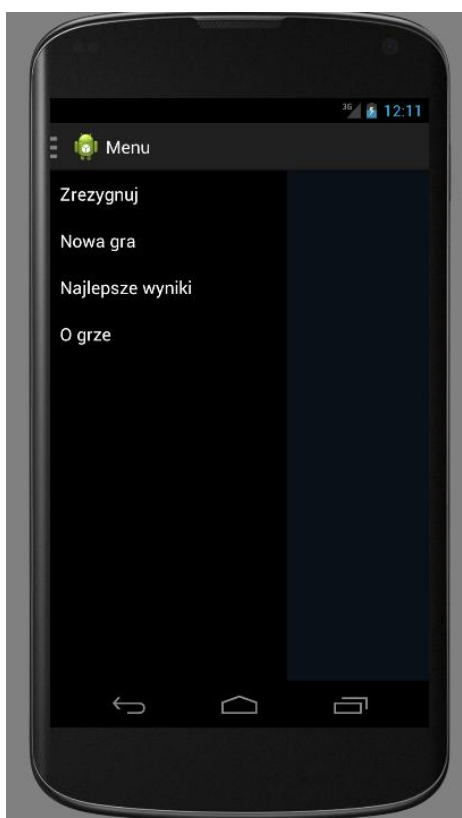


## 3) Najlepsze wyniki





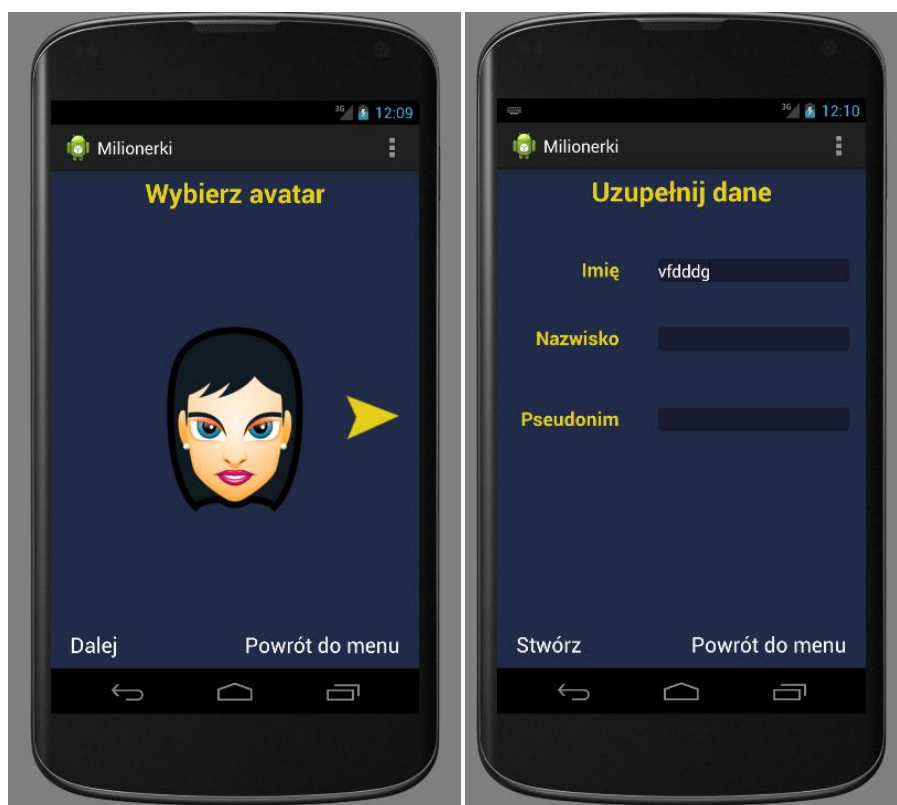
#### 4) Wysuwane menu



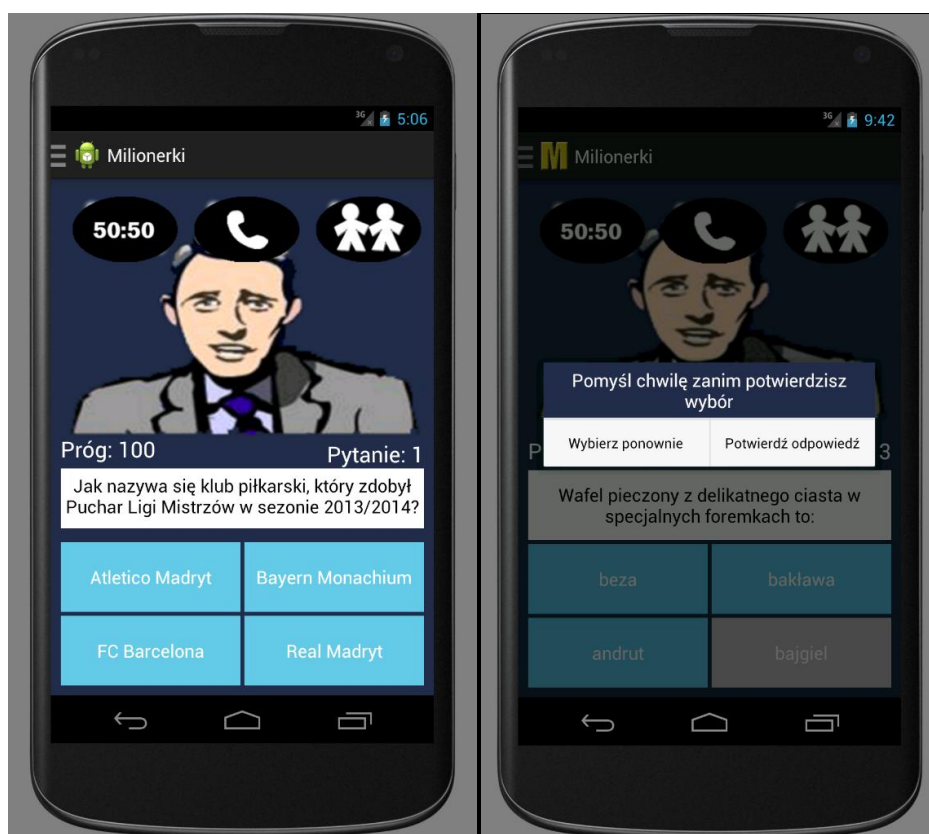
#### 5) Wybór profilu po rozpoczęciu gry



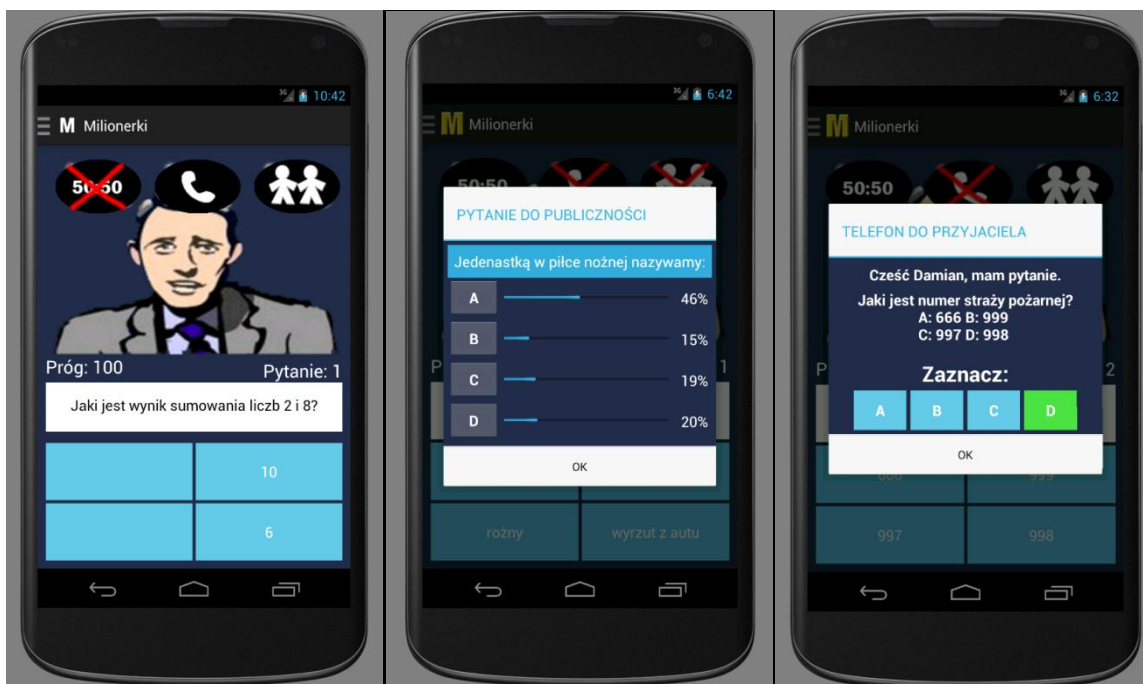
## 6) Stworzenie nowego profilu



## 7) Widok gry



## 8) Koła ratunkowe



## 9) Zakończenie gry

