



Politechnika Krakowska im.  
Tadeusza Kościuszki  
Wydział Fizyki, Matematyki i  
Informatyki



### Projekt Zespołowy

**Projekt:** HomeBudget – aplikacja mobilna ułatwiająca zarządzanie budżetem domowym

Data oddania:

Studia: Stacjonarne I stopnia  
Kierunek: Informatyka  
Rok akademicki: 3

Wykonali:  
Ślusarz Dominik  
Nr albumu: 104997  
Paradowski Krzysztof  
Nr albumu: 104976

# Spis treści

1. Sformułowanie zadania projektowego. Dziedzina projektowania.....	3
Cel i przeznaczenie projektu.....	3
Określenie wysokopoziomowych wymagań dla systemu.....	3
Cele.....	3
Przewidywane korzyści.....	3
Strategia i przyszłość.....	3
Założenia i ograniczenia.....	3
Bezpieczeństw i kontrola.....	3
Określenie kto będzie korzystał z systemu.....	3
Użytkownik.....	3
2. Harmonogram działań projektowych.....	4
3. Opis stosowanych technologii. Uzasadnienie wyboru technologii.....	4
4. Rozwiązania zapewniające skalowalność i wydajność.....	5
5. Wymagania funkcjonalne i нефункционалне.....	5
6. Architektura systemu. Wybór narzędzi i wymagania sprzętowe.....	6
7. Dokumenty przechowywane w systemie. Opis procesów.....	6
8. Diagramy DFD, ERD, STD. Obraz zachowania systemu w czasie, diagramy ELH.....	6
9. Interfejs graficzny aplikacji.....	6
10. Schemat komunikacji między komponentami architektury aplikacji.....	6
11. Walidacja wprowadzanych danych.....	6
12. Mechanizmy autentykacji / autoryzacji.....	6
13. Wymagania i warunki do uruchomienia aplikacji.....	6
14. Charakterystyka środowiska implementacyjnego. Konfiguracja.....	7
15. Opis etapów uruchamiania aplikacji (programu).....	7
16. Kody źródłowe (reprezentatywne fragmenty).....	7
17. Odniesienie do implementacji: definicje klas, powiązania pomiędzy klasami.....	7
18. Wyniki przeprowadzonych testów. Interpretacja wyników.....	7
19. Informacje o stosowanych pakietach niestandardowych.....	7
20. Informacje o zastosowaniu specyficznego podejścia do rozwiązania problemu.....	7
21. Ograniczenia aplikacji (programu). Możliwe rozszerzenia projektu.....	7
22. Raport z postępu prac.....	7
23. System pomocy. Słownik danych.....	7
24. Podsumowanie i wnioski.....	7
25. Bibliografia.....	7

# 1. Sformułowanie zadania projektowego. Dziedzina projektowania.

## ***Cel i przeznaczenie projektu***

**Określenie wysokopoziomowych wymagań dla systemu**

### ***Cele***

- stworzenie łatwej w obsłudze aplikacji zarządzania budżetem
- możliwość planowania tego na co i kiedy chcemy przeznaczyć daną ilość pieniędzy
- udostępnienie możliwości tworzenia i umieszczania planów na co i kiedy chcemy wydać pieniądze
- prosty i przejrzysty interfejs skierowany do osób, które nie miały wcześniej do czynienia z podobnymi rozwiązaniami informatycznymi
- dostęp mobilny

### ***Przewidywane korzyści***

System pozwoli na łatwiejsze zarządzanie budżetem. Ułatwi dostęp do zapisanych wcześniej decyzji. Pozwoli na zapamiętanie za co i kiedy mamy zapłacić lub kto i kiedy ma nam oddać pieniądze.

### ***Strategia i przyszłość***

W czasach, w których zaistnienie w internecie jest częścią każdego poważnego przedsięwzięcia chcemy zapewnić naszemu klientowi jak najprostsze narzędzie do zarządzania budżetem własnym lub własnej firmy.

### ***Założenia i ograniczenia***

- konieczność wykupienia serwera
- koszt wdrożenia systemu musi ograniczyć się do określonych środków finansowych
- dostosowanie projektu do ogólnie przyjętych i porównanych wymagań klientów już na etapie wdrażania
- konieczność aktualizacji aplikacji w miarę wykrywania luk zabezpieczeń (jeżeli takie zaistnieją)
- konieczność wdrażania aktualizacji usuwających błędy; poprawiających funkcjonalność oraz stabilność aplikacji
- konieczność pisania aplikacji na nowe wersje systemu w razie braku kompatybilności "w przód"

### ***Bezpieczeństwo i kontrola***

- konieczność zabezpieczenia danych użytkowników systemu
- poinformowanie klienta systemu, że za przechowywane w nim dane odpowiada on jako użytkownik systemu

**Określenie kto będzie korzystał z systemu**

### ***Użytkownik***

- zarządzanie opcjami systemu
- zarządzanie środkami
- generowanie wykresu wydatków
- wyświetlenie sumy rachunków dla poszczególnych kategorii oraz całości

## 2. Harmonogram działań projektowych.

Data	Opis
10 III 2015	Wybór tematu projektu
14 III 2015	Ref. 1, zarys projektu
28 IV 2015	Ref 2

## 3. Opis stosowanych technologii. Uzasadnienie wyboru technologii.

**Java**- obiektowy język programowania. Jest językiem tworzenia programów źródłowych kompilowanych do kodu bajtowego, który jest wykonywany przez wirtualną maszynę. Głównymi koncepcjami Javy są: obiektowość, dziedziczenie, prostota, niezależność od architektury(co w przypadku Androida nie koniecznie jest prawdą ponieważ system posiada lekko zmodyfikowaną maszynę wirtualną), niezawodność i bezpieczeństwo.

Wybór padł na Javę ponieważ jest domyślnym językiem oprogramowania na platformę Android. Posiada ogromną bazę bibliotek standardowych, więc zaoszczędza to czas podczas pisania własnych klas, które są w dodatku świetnie zoptymalizowane, API oraz rozbudowaną społeczność programistyczną.

**PHP**- obiektowy język programowania zaprojektowany do generowania stron internetowych i budowania aplikacji webowych w czasie rzeczywistym. Skrypty napisane w tym języku są odpalane po stronie servera.

Komunikacja z bazą danych na oddzielnym serverze może być przeprowadzona na wiele sposobów. Jednym z nich jest użycie skryptów PHP do akcji na bazie danych. Dzięki temu odciążamy samą aplikację przenosząc część zadań na server.

**MySQL**- wolnodostępny system zarządzania relacyjnymi bazami danych.

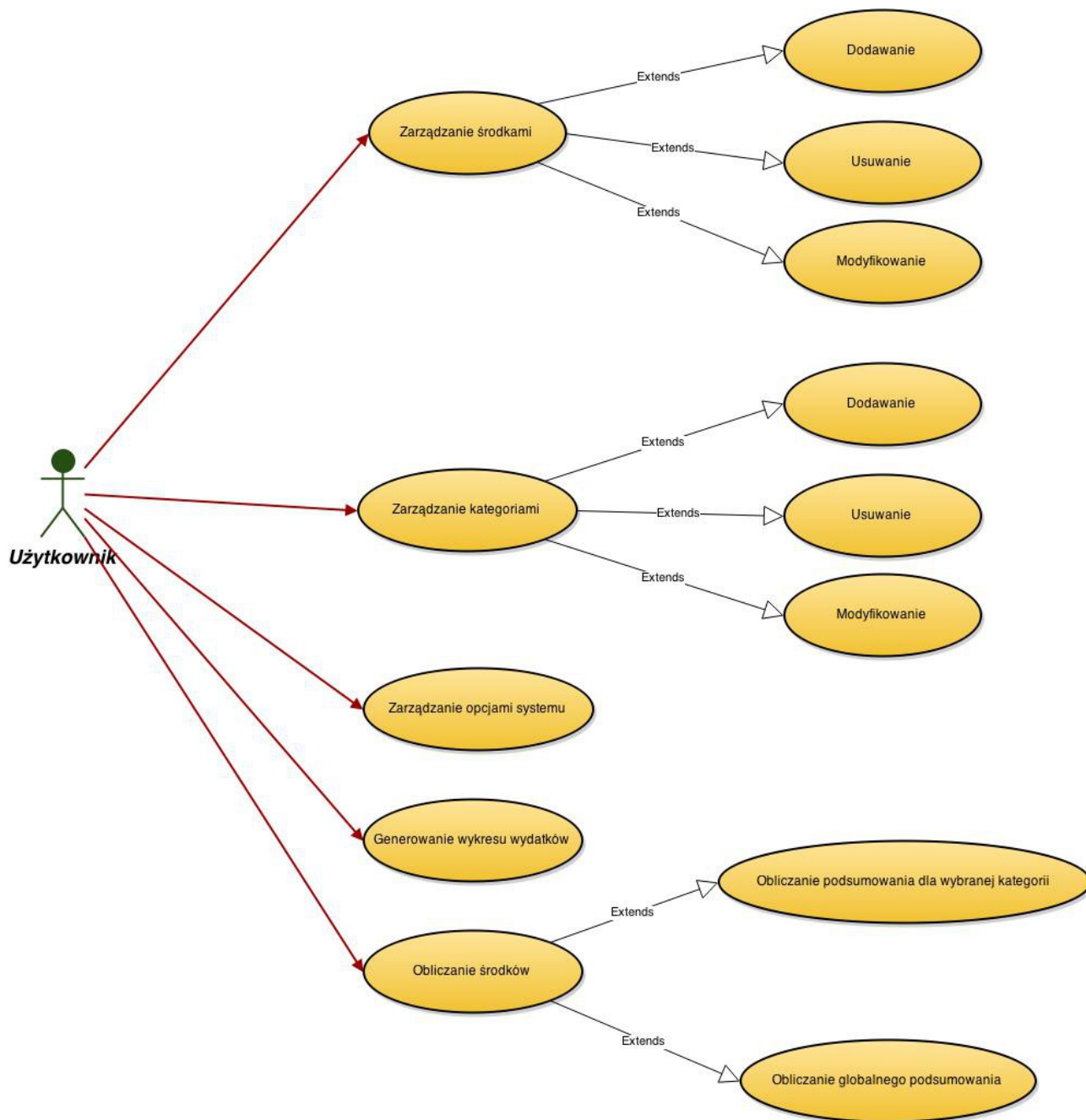
Jako jeden z najpowszechniejszych systemów, był oczywistym wyborem do przetrzymywania danych użytkownika.

## 4. Rozwiązania zapewniające skalowalność i wydajność.

System będzie dostosowany do możliwości dodania użytkownika na tym samym urządzeniu poprzez podanie odpowiedniego tokena lub wygenerowanie nowego.

Aplikacja będzie posiadała różne szablony dostosowujące się do rozmiaru urządzenia na którym będzie uruchamiana.

## 5. Wymagania funkcjonalne i нефункционалне.



Identyfikator	Nazwa	Opis	Realizacja
W_1	Zarządzanie środkami	Dodawanie, usuwanie, modyfikacja wydatków/przychodów	
W_2	Zarządzanie kategoriami	Dodawanie, usuwanie, modyfikacja kategorii wydatków/przychodów	
W_3	Generowanie wykresu wydatków	Wygenerowanie wykresu przedstawiającego procentowy udział każdego segmentu wydatków	
W_4	Zarządzanie opcjami systemu	Ustawianie opcji systemu	
W_5	Obliczanie środków	Obliczanie sumy	

		wydatków/przychodów, podsumowania z poszczególnych kategorii jak i globalnego	
--	--	---	--

## 6. Architektura systemu. Wybór narzędzi i wymagania sprzętowe.

Narzędzia:

- Android Studio (Android IDE)
- NetBeans (wielojęzykowe IDE wykorzystane do PHP)

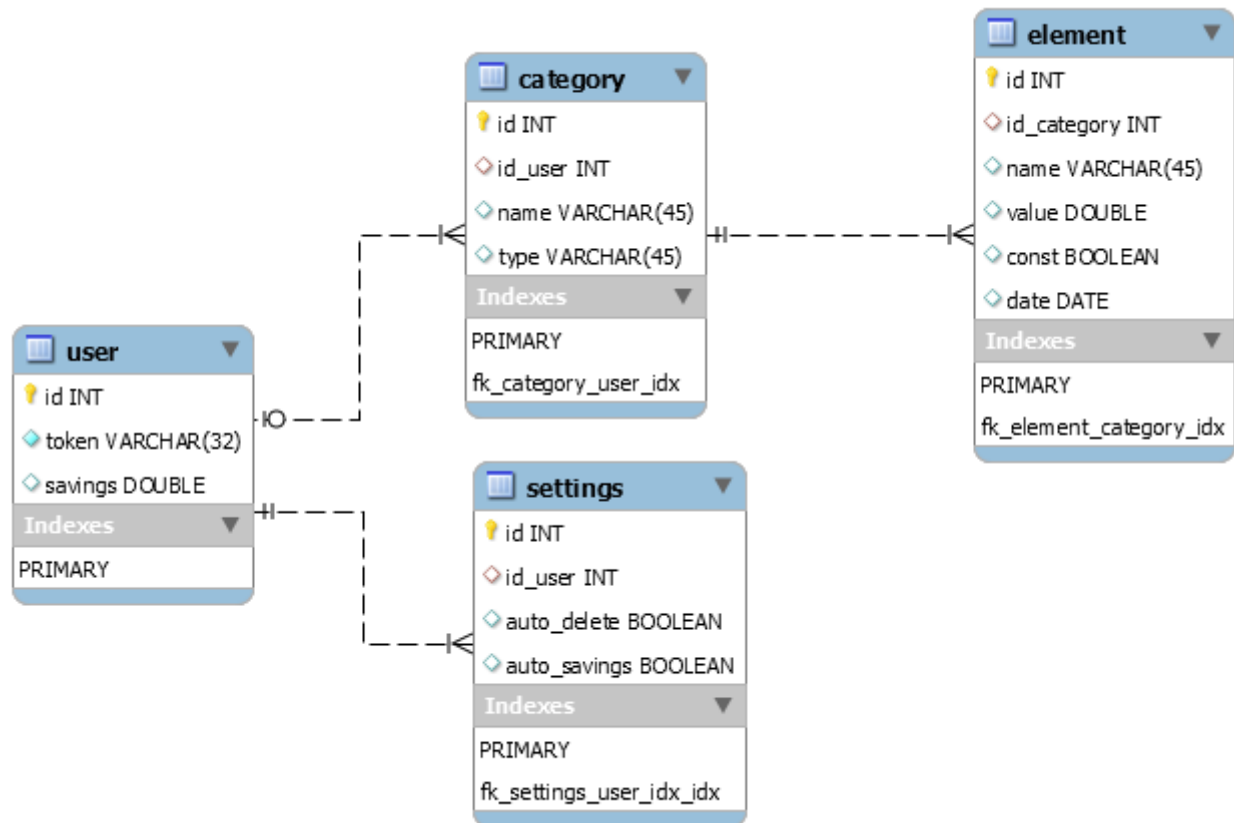
Do poprawnego działania aplikacji niezbędne jest urządzenie z systemem Android w wersji co najmniej 2.3.3 Gingerbread.

## 7. Dokumenty przechowywane w systemie. Opis procesów.

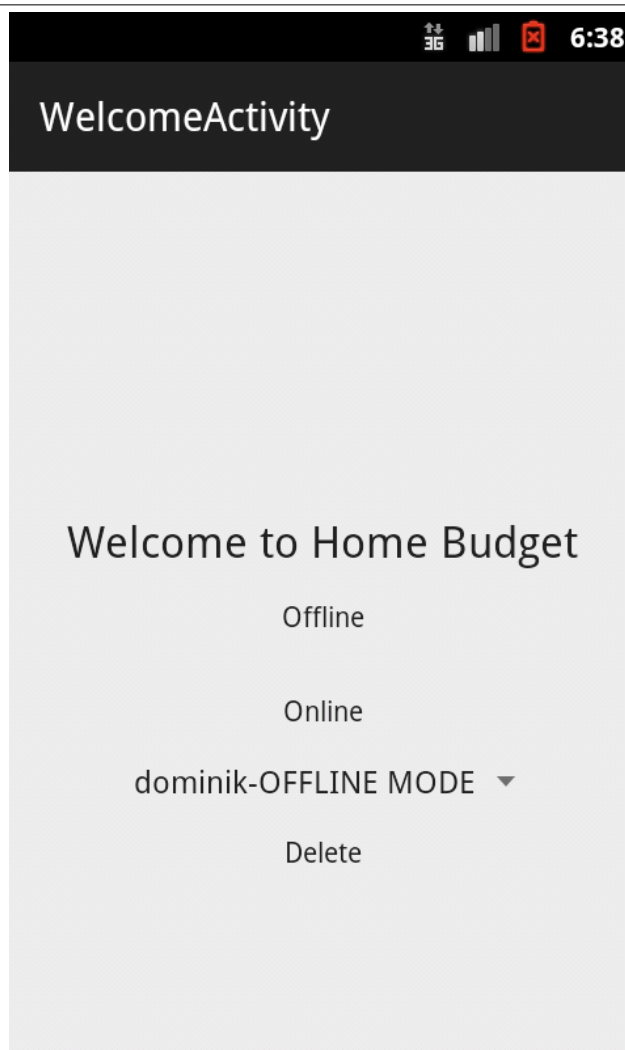
W systemie mobilnym, opartym na systemie Android przechowywane są dane samej aplikacji(kod bajtowy) oraz dane użytkownika przechowywane w pliku w przypadku dostępu offline.

Na serwerze przechowywane są dane użytkownika w przypadku dostępu online.

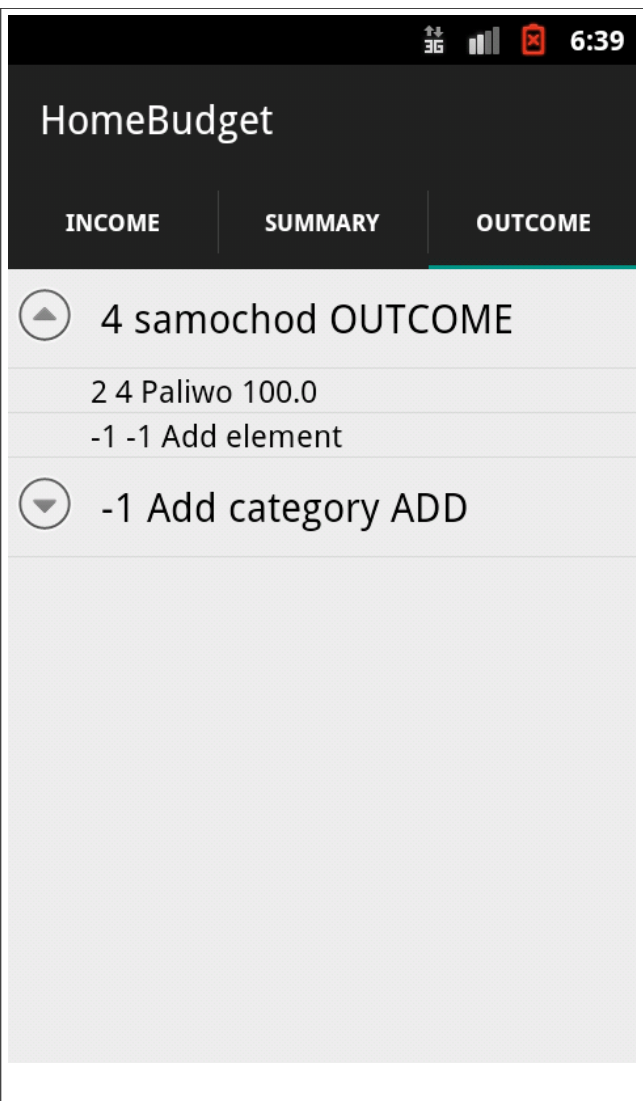
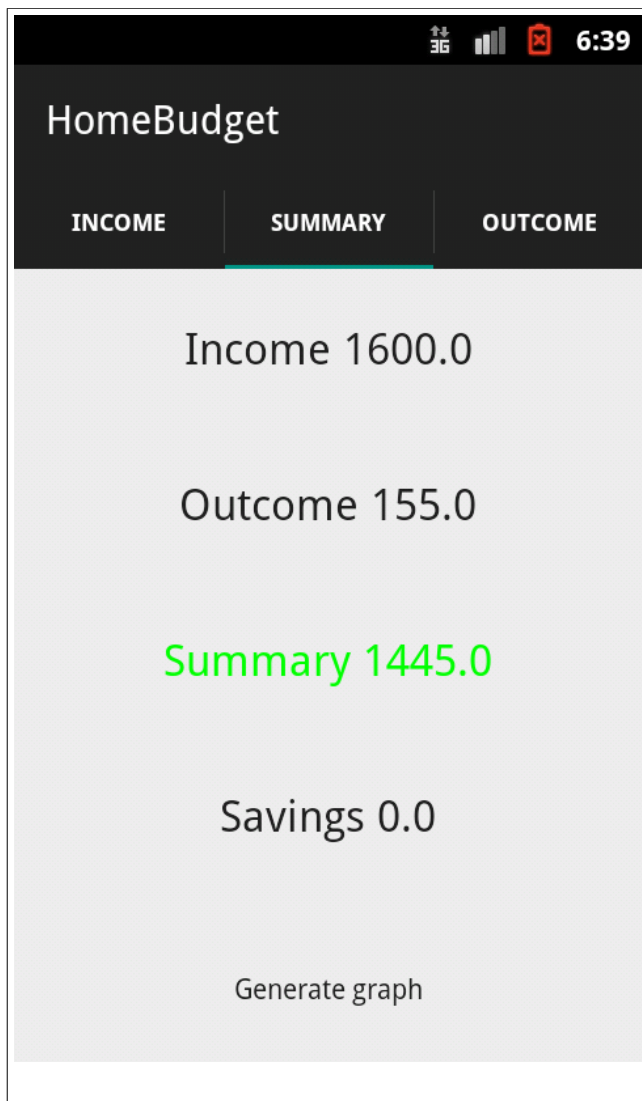
## 8. Diagramy DFD, ERD, STD. Obraz zachowania systemu w czasie, diagramy ELH.



## 9. Interfejs graficzny aplikacji.







6:40

SettingsActivity

☐

Auto-savings in first day every month

☐

Auto-delete element in first day every month

Save

Cancel

6:39

VersionActivity

Token:

Version

OFFLINE MODE

1.6

6:40

HomeBudget

INCOME

SUMMARY

OUTCOME

Income 1600.0

Outcome 100.0

Summary 1500.0

6:40

HomeBudget

INCOME

SUMMARY

OUTCOME

3 praca INCOME

1 3 Premia 300.0

3 3 Wyndagrodzenie 1300.0 2015-04-02

-1 -1 Add element

-1 Add category ADD

3G 3:09

ElementActivity

Element

Name

Value

☐ Constant element

☒ Add date

+

Apr

-

+

26

-

+

2015

-

Save

Cancel

## 10. Schemat komunikacji między komponentami architektury aplikacji.

Użytkownik będzie mógł komunikować się z biblioteką za pomocą publicznego API udostępnianego przez bibliotekę. Uzyska w ten sposób dostęp do właściwości, metod i obiektów na których będzie mógł wykonywać operacje.

Wewnętrzne komponenty komunikują się ze sobą wywołując funkcje należące do poszczególnych obiektów.

## 11. Walidacja wprowadzanych danych.

Walidacja danych następuje podczas wprowadzania danych przez użytkownika, Jeżeli podał on nieprawidłowe dane otrzyma informację zwrotną. Dodatkowo zabezpieczenie obejmą także sprawdzanie SQL injection.

## **12. Mechanizmy autentykacji / autoryzacji.**

Aplikacja posiada dwa tryby. Offline który nie posiada autoryzacji, oraz online. W przypadku trybu online użytkownik jest rozpoznawany przez token. Jest on tworzony podczas pierwszego uruchomienia aplikacji w trybie online. Użytkownikowi nadawany jest token który jest przetrzymywany w bazie danych jak i na urządzeniu.

## **13. Wymagania i warunki do uruchomienia aplikacji.**

Jedynym wymaganiem jest posiadanie urządzenia z systemem Android w wersji nie mniejszej niż 2.3.3 Gingerbread.

## **14. Charakterystyka środowiska implementacyjnego. Konfiguracja.**

Android Studio – oficjalne IDE dla aplikacji na systemy Android oparte na IntelliJ IDEA. Oprócz wszystkich funkcjonalności którą zapewnia IntelliJ Android studio zapewnia:

- Elastyczny system Gradle
- Wbudowane generatory plików .apk
- Przykłady kodu, aby pomóc w zbudowaniu powszechnych aplikacji
- Edytor układów ze wsparciem drag and drop
- Narzędzie lint zapewniające wydajność i kompatybilność
- ProGuard oraz możliwość podpisywania aplikacji

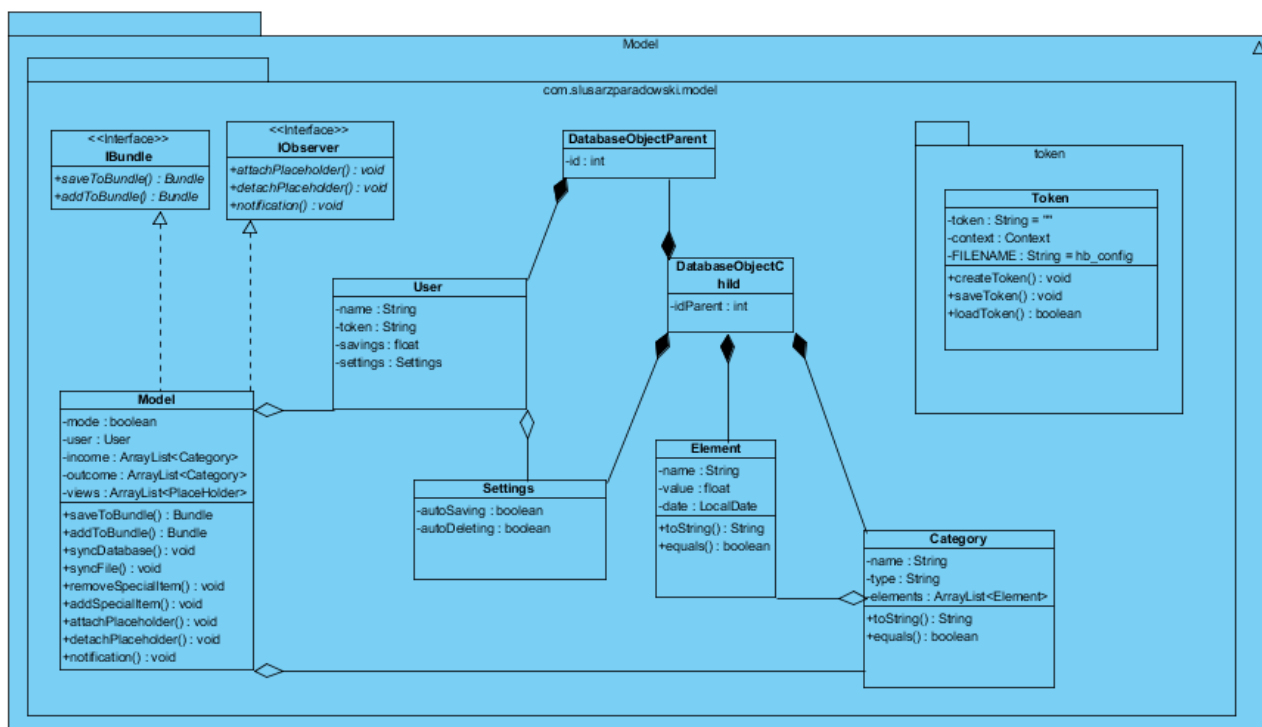
## **15. Opis etapów uruchamiania aplikacji (programu).**

Aplikacja jest uruchamiana na systemie Android(min 2.3.3). Użytkownik wybiera tryb w jakim będzie pracował(online/offline). Po tym wyborze użytkownik będzie mógł zarządzać kategoriami i elementami, generować wykres oraz tworzyć podliczenia.

## **16. Kody źródłowe (reprezentatywne fragmenty).**

Cały kod programu dostępny jest na <https://github.com/Dominik93/Home-budget>

## 17. Odniesienie do implementacji: definicje klas, powiązania pomiędzy klasami.



## 18. Wyniki przeprowadzonych testów. Interpretacja wyników.

## 19. Informacje o stosowanych pakietach niestandardowych.

**Gson** is a Java library that can be used to convert Java Objects into their JSON representation. It can also be used to convert a JSON string to an equivalent Java object. Gson can work with arbitrary Java objects including pre-existing objects that you do not have source-code of.

**Joda Time** - The standard date and time classes prior to Java SE 8 are poor. By tackling this problem head-on, Joda-Time has become the de facto standard date and time library for Java. Note that from Java SE 8 onwards, users are asked to migrate to `java.time` (JSR-310).

Gson-jodatime-serialisers

## 20. Informacje o zastosowaniu specyficznego podejścia do rozwiązania problemu.

## 21. Ograniczenia aplikacji (programu). Możliwe rozszerzenia projektu.

## 22. Raport z postępu prac.

## 23. System pomocy. Słownik danych.

<b>Skrót</b>	<b>Opis</b>
UC	Use Case – przypadek użycia
W	Wymaganie
App	Aplikacja

## **24. Podsumowanie i wnioski.**

## **25. Bibliografia.**