



Inżynierski projekt zespołowy

Temat: Projekt i wykonanie aplikacji umożliwiającej połączenie i synchronizację danych pomiędzy dwoma urządzeniami działającymi na systemie android.

DYPLOMANT:

Radosław Patelka

radoslaw.patelka@pollub.edu.pl

PROWADZĄCY:

Dr inż. Tomasz Szymczyk

Spis treści:

- 1. Wstęp**
- 2. Cel i zakres pracy**
- 3. Instalacja środowiska**
- 4. Wymagania funkcjonalne i нефunkcjonalne**
- 5. Scenariusz użycia**
- 6. Oprogramowanie**
- 7. Screeny prototypu aplikacji**
- 8. Wnioski**
- 9. Literatura**

1. Wstęp

Człowiek jako istota stadna, łączy się w mniejsze grupki, w których żyje, a nieodłączną częścią tego życia stanowią zakupy. W dzisiejszych czasach, kiedy technologia gna do przodu, ludzie odkrywają coraz to lepsze sposoby, aby pokonywać wciąż większe odległości i nadal mieć możliwość komunikowania się z osobami bliskimi. W związku z tą sytuacją, kiedy człowiek często pokonuje większe odległości by wykonać sprawunki, wyszedłem z pomocną ręką, aby ułatwić mu komunikację zorientowaną na zakupy i stworzyłem aplikację „Lista zakupów”. Wykorzystałem w niej technologię systemu Android, aparat fotograficzny, synchronizację dwóch urządzeń mobilnych poprzez serwer ftp. Dzięki tej aplikacji, niezależnie od odległości, człowiek będący w zasięgu telefonii komórkowej, będzie mógł połączyć się z drugą osobą, by zweryfikować listę zakupów.

Android, na który owa aplikacja powstała, to system operacyjny stworzony dla urządzeń mobilnych, telefonów komórkowych, smartfonów, netbooków oraz tabletów. Został zrealizowany w wielu wersjach, począwszy od wersji 1.0 Apple Pie (ang. szarlotka) wydanej we wrześniu 2008 roku aż do tej pory najnowszej wersji 7.1 Nougat (ang. nugat) wydanej 4 października 2016 .

2. Cel i zakres pracy

2.1. Cel pracy

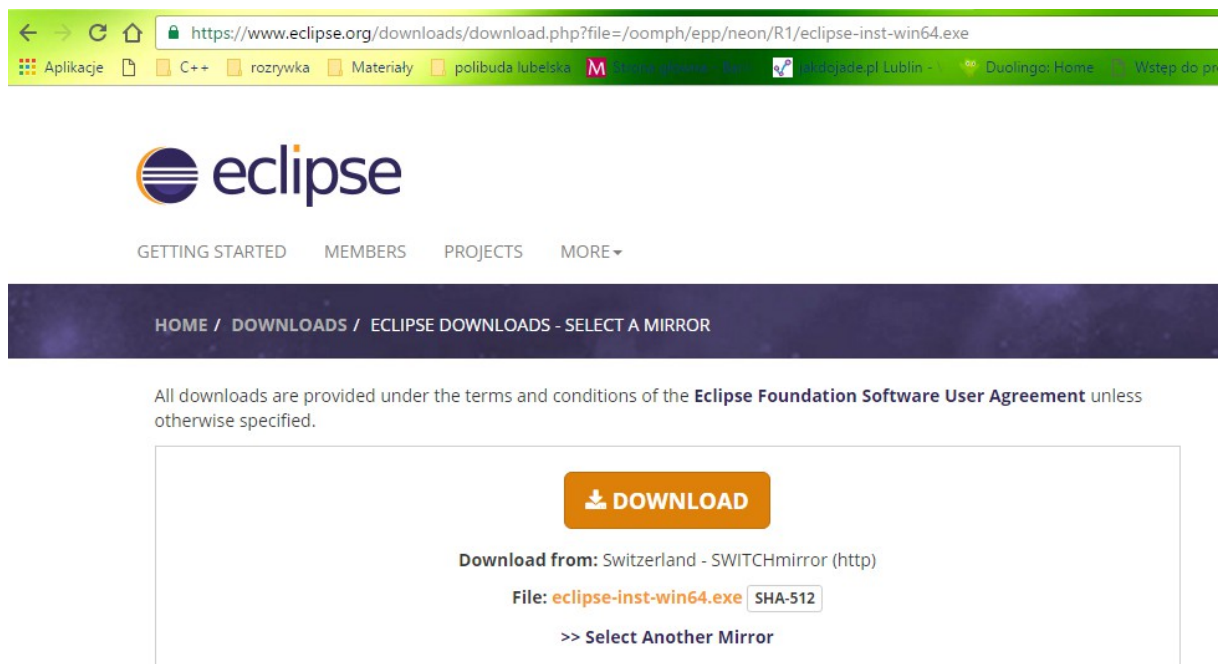
Celem pracy jest stworzenie aplikacji umożliwiającej synchronizację przez serwer ftp, dwóch urządzeń mobilnych, w celu stworzenia/zaktualizowania listy zakupów, oraz, z wykorzystaniem aparatu, przechowywanie zdjęć paragonów w celu zniwelowania ryzyka ich zgubienia/zniszczenia. Wykorzystując komunikację przez serwer dana jest możliwość przekazywania informacji za pośrednictwem internetu.

2.2. Zakres pracy

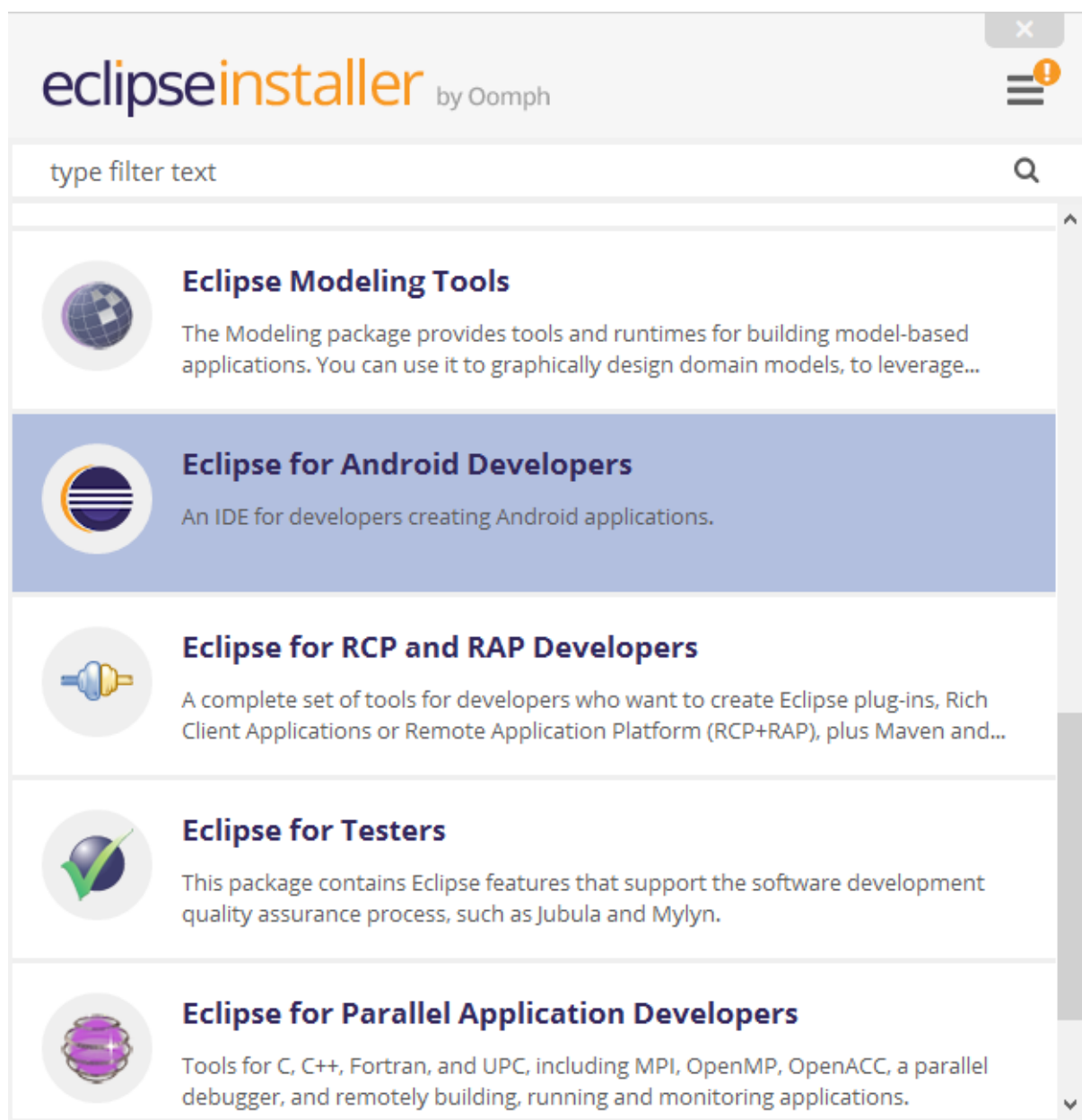
1. Analiza środowiska i oprogramowania , w którym zostanie stworzona aplikacja.
2. Wykonanie aplikacji.
3. Określenie wymagań funkcjonalnych/niefunkcjonalnych.
4. Oprogramowanie aplikacji w wybranym środowisku programistycznym.

3. Instalacja technologii

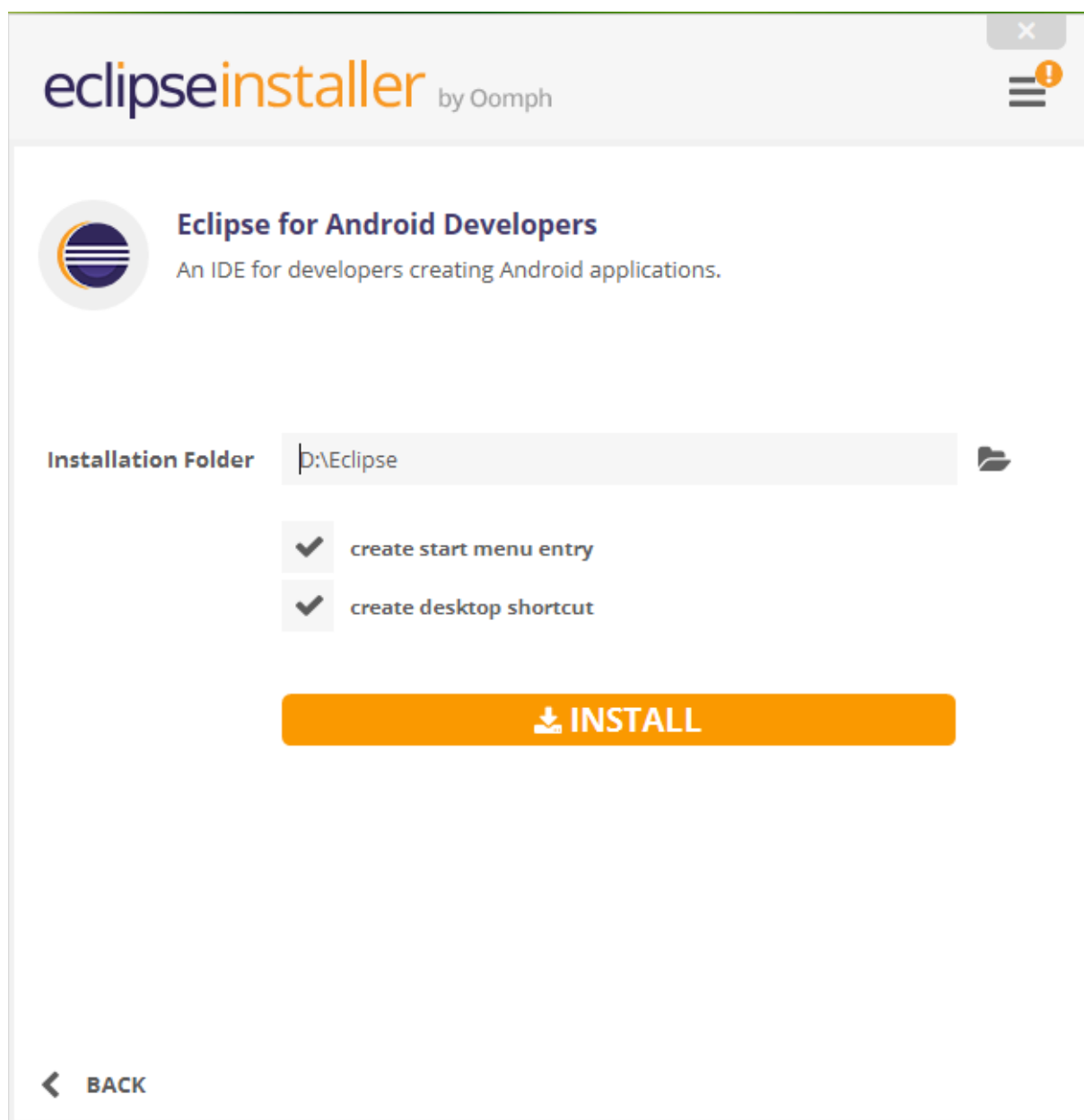
3.1.Instalacja Eclipse



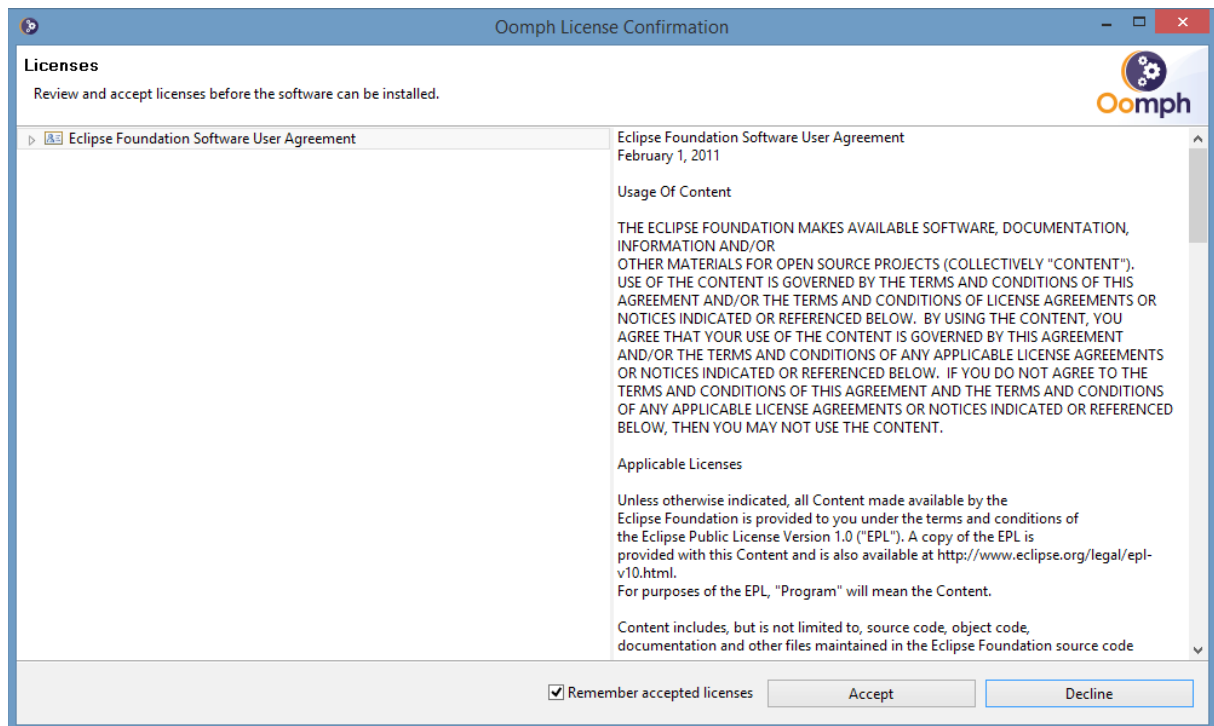
Rys. 3.1. Pobranie ze strony



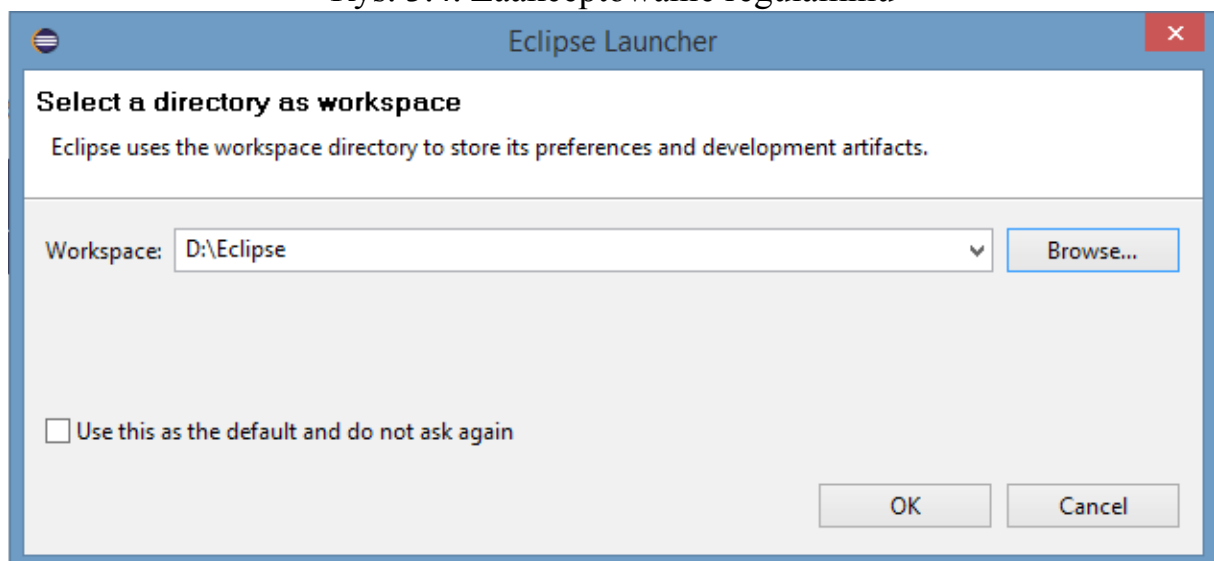
Rys. 3.2. Wybór odpowiedniego środowiska programistycznego



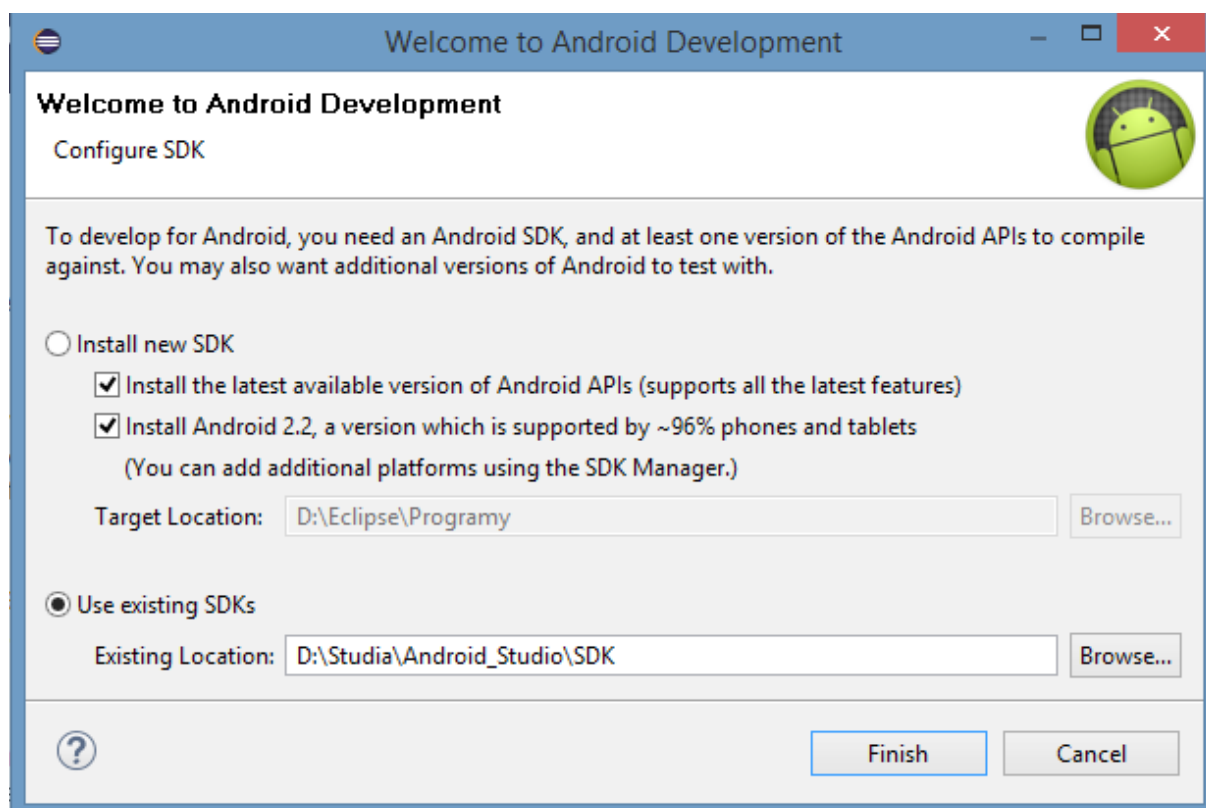
Rys. 3.3. Wybór miejsca instalacji



Rys. 3.4. Zaakceptowanie regulaminu



Rys. 3.5. Wybór miejsca instalacji workspace



Rys. 3.6. Instalacja SDK, bądź skorzystanie z już istniejącego

New Android Project

Create an Android project
Create a new Android project resource.

Project name:

Contents

☒ Create new project
☐ Create new project using sample
☐ Create project from existing source
☒ Use default location

Location:

Target

| SDK Target | Vendor | API Level | Plat... |
|---|-----------------------------|-----------|---------|
| <input checked="" type="checkbox"/> Android 6.0 | Android Open Source Project | 23 | 6.0 |
| <input type="checkbox"/> Android 7.1.1 | Android Open Source Project | 25 | 7.1.1 |
| <input type="checkbox"/> Google APIs | Google Inc. | 23 | 6.0 |

Standard Android platform 6.0

Application

Application name:
Package name:
Activity name:
Min SDK version:

Rys. 3.7. Stworzenie projektu

4. Wymagania funkcjonalne i нефункционалне

4.1. Wymagania funkcjonalne

1. Aplikacja musi działać na wszystkich urządzeniach mobilnych wykorzystujących system Android (od wersji 4.4).
2. Aplikacja musi mieć dostęp do wbudowanego w telefon aparatu.
3. Aplikacja musi mieć dostęp do Internetu w celu przekazywania za pośrednictwem serwera ftp, danych na drugie urządzenie.
4. Na stronie startowej użytkownik posiada: na górze strony informacje o tym jaki użytkownik jest w tym momencie zalogowany, oraz w prawym górnym rogu przycisk „Ustawienia”, poniżej zaś przyciski: „Lista zakupów”, „Twoi użytkownicy”, „Paragony”, „Zakończ”.
5. Przycisk „Ustawienia” umożliwia użytkownikowi zmianę hasła, oraz wylogowanie się.
6. „Lista zakupów” – po wejściu w tę opcję, wyświetla pole służące wpisaniu produktów. Po wpisaniu artykułu i zatwierdzeniu go przyciskiem, pojawia się on wraz z innymi, poniżej w formie listy. Dodatkowo wyświetlany jest komunikat informujący o tym jaki produkt i w jakiej ilości, został dodany. Domyślnym kolorem czcionki dla produktów jest czarny. Kiedy użytkownik dokona zakupu, za pomocą jednego kliknięcia w zakupiony produkt zmienia jego kolor na szary i przenosi go na koniec listy. Aby nieumyślnie nie odznaczyć niewłaściwego artykułu, dodatkowo wyświetla się stosowny komunikat. W przypadku zaznaczenia niewłaściwego produktu, użytkownik stosując ponowne kliknięcie na danym produkcie powoduje zmianę jego koloru z powrotem na czarny, oraz przeniesienie go na szczyt listy, czemu towarzyszy wyświetlenie informacji. Ponadto dłuższe przytrzymanie jednego produktu powoduje wyświetlenie się menu kontekstowego z możliwością wyboru dwóch opcji:
 - „Edytuj” - pozwala na poprawienie potencjalnych błędów w pisowni, oraz zmianę ilości w jakiej dany produkt ma zostać zakupiony
 - „Usuń” - umożliwia usunięcie wybranego produktu z listy.
7. „Twoi użytkownicy” – po wybraniu tego przycisku, użytkownik przenoszony jest do nowej aktywności, w której podaje on nazwę osoby z którą chce się połączyć, lub wybiera go z listy powstałej z wybieranych wcześniej przez niego partnerów. Po wpisaniu nazwy użytkownika, aplikacja łączy się z bazą danych i sprawdza, czy dany użytkownik istnieje. Jeżeli znajduje się on w bazie zostanie

automatycznie dodany do listy. W przypadku błędnego wpisana nazwy, bądź próby połączenia się z nieistniejącą osobą, wyświetla się stosowny komunikat.

8. „Paragony” – przełącza do okna, w którym użytkownik ma możliwość wyboru dwóch opcji: „Zrób zdjęcie”, oraz „Pokaż paragony”. Opcja pierwsza przełącza na aparat w celu wykonania zdjęcia paragonu i zapisania go w folderze, druga natomiast przekierowuje do folderu, w którym przechowywane są wykonane wcześniej zdjęcia.
9. „Zakończ” – zamyka program i kończy jego działanie

4.2 Wymagania нефunkcjonalne

Aplikacja napisana w języku Java.

Aplikacja wykorzystuje serwer ftp.

Aplikacja korzysta z aparatu.

Aplikacja korzysta z SQLite.

5. Scenariusz użycia

Osoba 1 znajduje się na zakupach, osoba 2 przebywająca w domu, spostrzega, że poza produktami zawartymi w liście, brakuje jeszcze 2l mleka. W związku z powyższym, bierze telefon, uruchamia aplikację, wchodzi w listę zakupów i dopisuje brakujący produkt. Na telefonie osoby 1, dopisane 2l mleka pojawiają się na szczycie listy zakupów, co czyni je niemożliwym do przeoczenia.

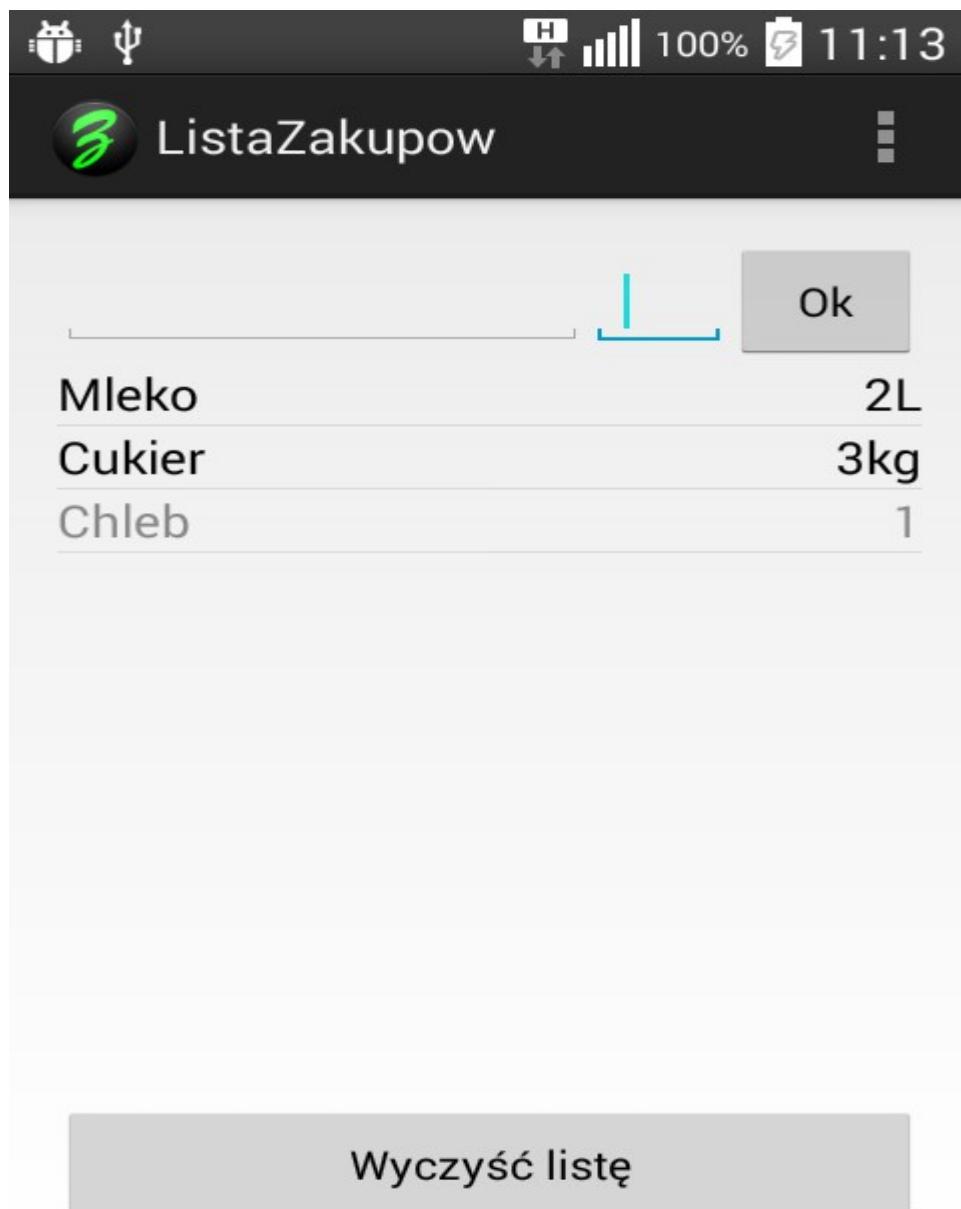
6. Oprogramowanie

```
// Wrzucić kod layout'u strony głównej  
// Wrzucić kod łączenia z użytkownikiem i bazy danych  
// Wrzucić Diagram ERD
```

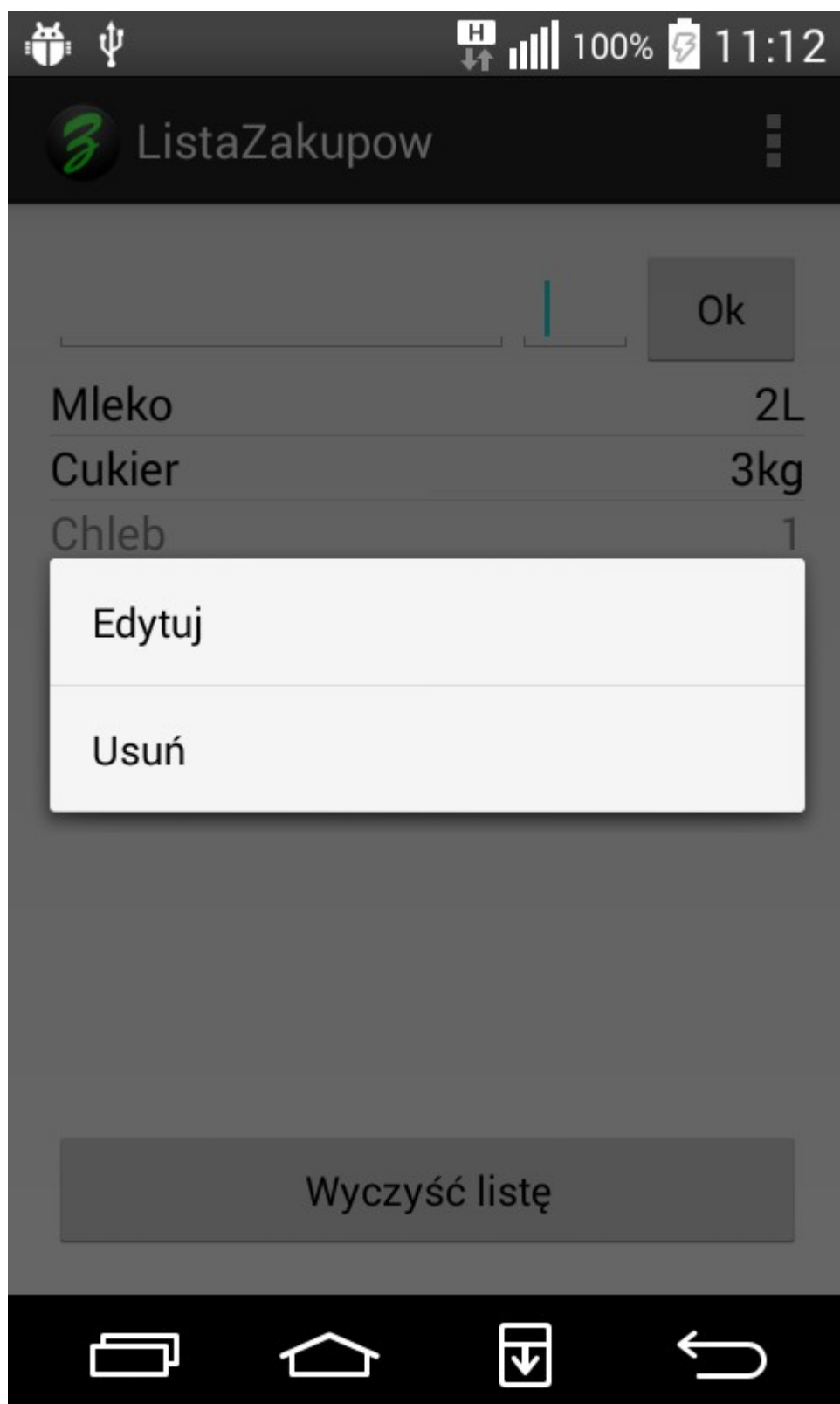
7. Screeny ukazujące działanie aplikacji



Rys 7.1 Wygląd głównego menu



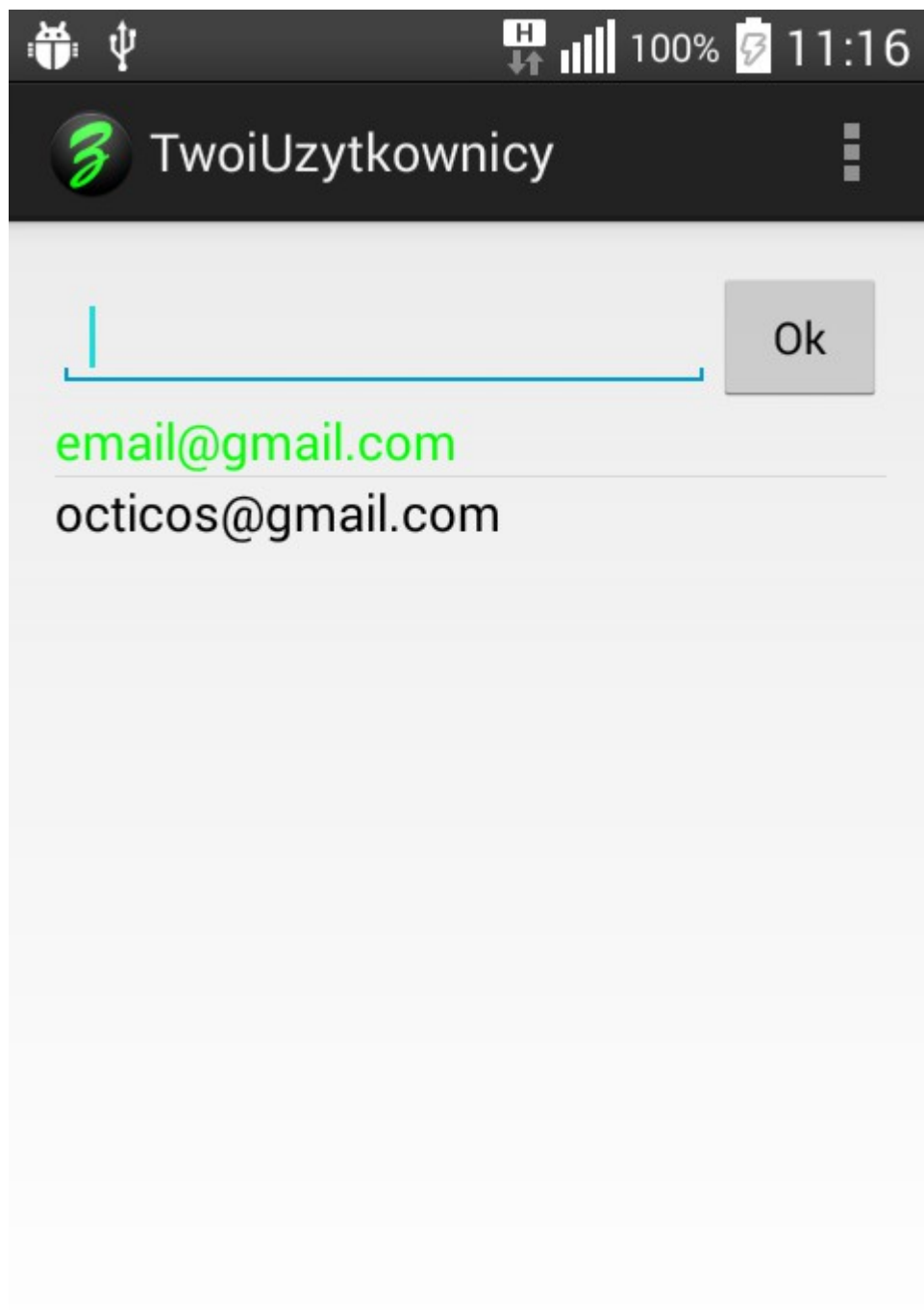
Rys 7.2 Wygląd menu tworzenia listy z jednym już kupionym elementem



Rys 7.3 Widok po przytrzymaniu jednego z artykułów



Rys 7.4 Widok z aparatu



Rys 7.5 Widok połączenia z użytkownikiem

8. Wnioski

Przedstawiona aplikacja umożliwia komunikację pomiędzy dwoma urządzeniami mobilnymi, działającymi na systemie android, za pomocą serwera ftp, dzięki czemu odległość nie jest problemem. Dzięki niej użytkownicy mogą

wzajemnie przekazywać sobie dane (w tym przypadku produkty dostępne w sklepach). Ponadto dzięki wykorzystaniu aparatu wbudowanego w urządzenie, użytkownik ma możliwość zrobić zdjęcie paragonu, czym zaoszczędzi miejsce w portfelu, uodporni się na nietrwałość materiału z jakiego wykonane są paragony, oraz zapewni sobie ułatwienie w przypadku konieczności składania reklamacji. Prosty wygląd, przyjrzyistość i logiczne rozmieszczenie elementów, umożliwiają szybkie zapoznanie się z aplikacją, niezależnie od stopnia posiadanej wiedzy technicznej.

9. Literatura

- [1] <http://stackoverflow.com/questions/9853351/building-my-own-camera-application-automatically-capture-the-image> (dostęp: 05.2016)
- [2] <https://developer.android.com/guide/topics/media/camera.html> (dostęp: 05.2016)
- [3] <http://www.windowsnetworking.com/articles-tutorials/network-protocols/Understanding-FTP-Protocol.html> (dostęp: 11.2016)
- [4] <https://developer.android.com/index.html> (dostęp: 12.2016)