

AndCourier



Kamil Janowski

Mateusz Kowalski

Mobilna aplikacja bazodanowa wspomagająca pracę
kurierską

AndCourier

Dokumentacja

Spis treści

Opis projektu	3
Wymagania serwera AndCourier Server	3
Wymagania funkcjonalne	3
Wymagania нефункционалне	3
Wymagania klienta AndCourier Client	3
Wymagania funkcjonalne	3
Wymagania нефункционалне	4
Baza danych	4
Model bazy danych	4
Opis danych	5
Couriers	5
Logins	5
Parcels	5
ParcelType	5
Customers	5
Opis używanych technologii	5
Android 2.2	5
Pakiety niestandardowe	7
Przypadki użycia	7
Opisowe	7
Diagramy	10
Klient	10
Server	10
Diagramy klas	11
Aktywności klienta	11
Komunikacja po stronie klienta	11
Komunikacja po stronie serwera	12

Diagramy sekwencji.....	13
Logowanie	13
Przetworzenie wiadomości przez serwer	14
Ekrany	14
Opis ekranów.....	14
Mapa.....	17
Plan testów	17
Możliwe rozszerzenia	18
Słownik danych.....	18
Bibliografia.....	18
Załączniki	19

Opis projektu

Projekt składa się z dwóch aplikacji:

- a) AndCourier Server udostępniający dane dotyczące paczek, które należy dostarczyć, oraz umożliwiający weryfikację kuriera. Pozwala też na dodawanie do systemu zarówno nowych paczek jak i zarządzanie kurierami (ich dodawanie i usuwanie z systemu). Ponadto serwer sam jest w stanie utworzyć bazę danych na serwerze MySQL, którą później będzie zarządzał.
- b) AndCourier Client będącego kliencką aplikacją mobilną, która umożliwia logowanie się do serwera, przeglądanie informacji o paczkach, które należy dostarczyć, zgłaszanie dostarczenia paczki oraz przydzielanie zadanie dostarczenia paczki do konkretnego kuriera.

Wymagania serwera AndCourier Server

Wymagania funkcjonalne

1. Możliwość obsługi serwera z konsoli, dzięki czemu możliwa będzie obsługa przy połączeniu za pomocą protokołu SSH.
2. Serwer obsługuje polecenia opisane w załączniku nr 1: „AndCourier Server – instrukcja obsługi”.
3. Serwer zapisuje wszystkie logi dotyczące nawiązywanych z nim połączeń klienckich oraz przebiegu wymiany danych pomiędzy serwerem, a klientem. Stare logi są kasowane przy każdym uruchomieniu serwera.
4. Serwer sam jest w stanie zbudować potrzebną mu bazę danych na udostępnionym mu serwerze MySQL.
5. Serwer przeprowadza autentykację łączących się z nim użytkowników.
6. Serwer udziela odpowiedzi na wszystkie zapytania klienta na podstawie danych zawartych w bazie danych.
7. Login i hasło przy logowaniu powinny być nie dłuższe niż 45 znaków.

Wymagania нефunkcjonalne

1. Dostęp do serwera MySQL.
2. Zainstalowany dowolny system operacyjny, na którym jest możliwe zainstalowanie JRE.
3. Zainstalowana Java Runtime Environment w wersji co najmniej 7.0 .
4. Komputer musi mieć podłączoną klawiaturę, lub udostępniać zdalny pulpit z wystarczającymi uprawnieniami, aby operować na serwerze, lub udostępniać połączenie SSH poprzez które będzie można uzyskać wystarczające uprawnienia do operowania na serwerze.
5. Udostępniony otwarty port 666.
6. Dostęp do sieci.

Wymagania klienta AndCourier Client

Wymagania funkcjonalne

1. Logowanie do serwera.

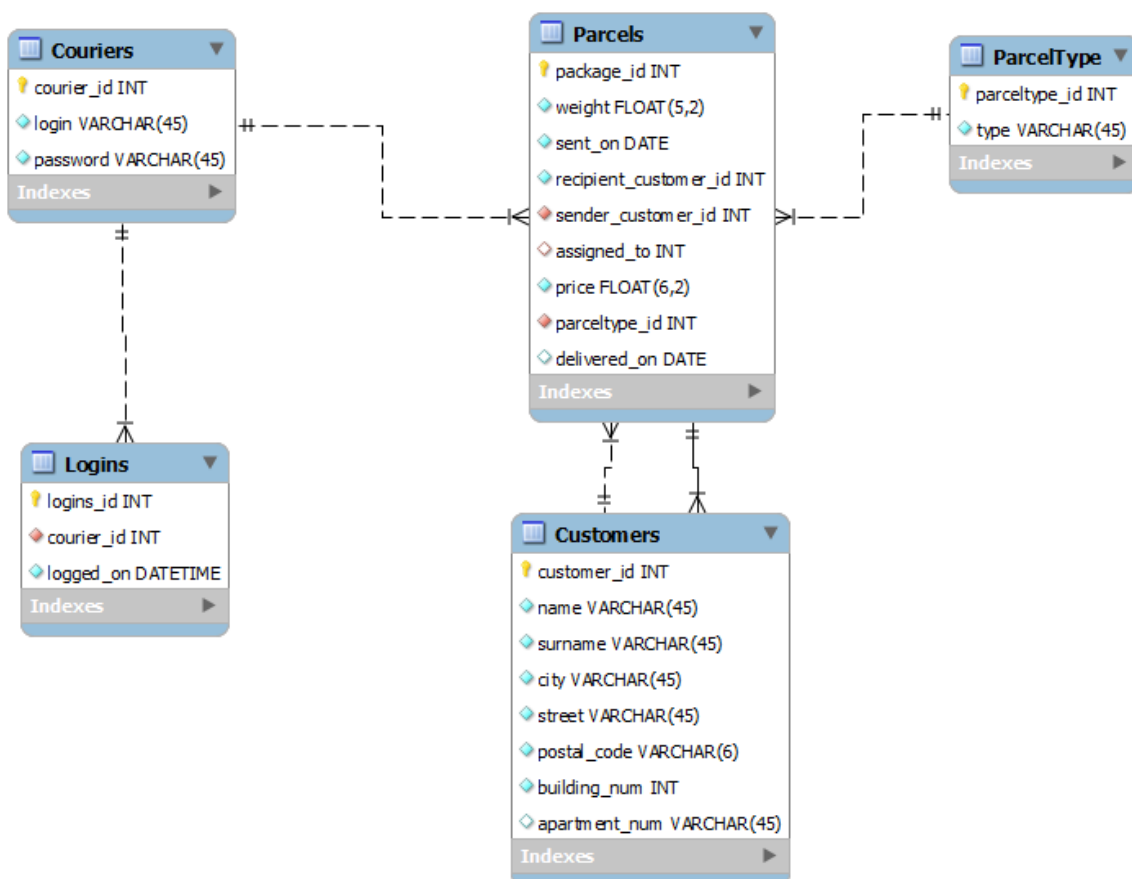
2. Zapamiętywanie podstawowych informacji niezbędnych do zalogowania: adres serwera i login (istnieje konieczność każdorazowego podania hasła).
3. Wylogowywanie.
4. Pobieranie listy paczek, które nie są przydzielone do żadnego kuriera.
5. Pobieranie listy paczek, które są przydzielane do aktualnie zalogowanego kuriera.
6. Przydzielanie nieprzydzielonych paczek do aktualnie zalogowanego kuriera.
7. Oznaczanie paczki jako dostarczonej.
8. Przeglądanie szczegółów dotyczących wyświetlanych paczek (waga, data wysłania, adres nadawcy, adres odbiorcy, cena którą należy uiścić, rodzaj paczki)

Wymagania niefunkcjonalne

1. Urządzenie mobilne z zainstalowanym systemem operacyjnym Android w wersji co najmniej 2.2 .
2. Urządzenie spełniające minimalne wymagania sprzętowe wymienione w dokumentacji Androida 2.2 .

Baza danych

Model bazy danych



Opis danych

Couriers

Tabela przechowująca listę wszystkich kurierów. Zawiera następujące informacje: identyfikator kuriera, login i hasło.

Logins

Zawiera informacje o dacie i godzinie logowania kuriera do systemu.

Parcels

Tabela opisująca paczki, które trafiają do firmy kurierskiej. Zawiera następujące informacje: waga paczki, kiedy była wysłana, odnośnik do klienta, który nadał paczkę oraz odnośnik do klienta, który jest adresatem paczki, cena jaka została uiszczona za paczkę, odnośnik do kuriera, który ma ją dostarczyć (jeśli już jakiś został przypisany), typ paczki (list, paczka zwykła, paczka delikatna itp.) oraz datę, kiedy została dostarczona (domyślnie null, co oznacza, że jeszcze nie została dostarczona).

ParcelType

Tabela wymieniająca typy paczek np. list, paczka zwykła, paczka delikatna.

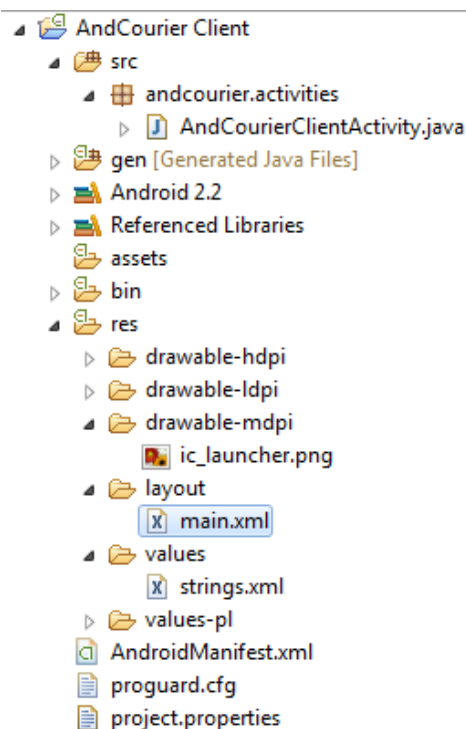
Customers

Tabela opisująca klientów. Zawiera następujące informacje: imię, nazwisko, miasto, ulicę, kod pocztowy, numer budynku i numer mieszkania (jeśli istnieje).

Opis używanych technologii

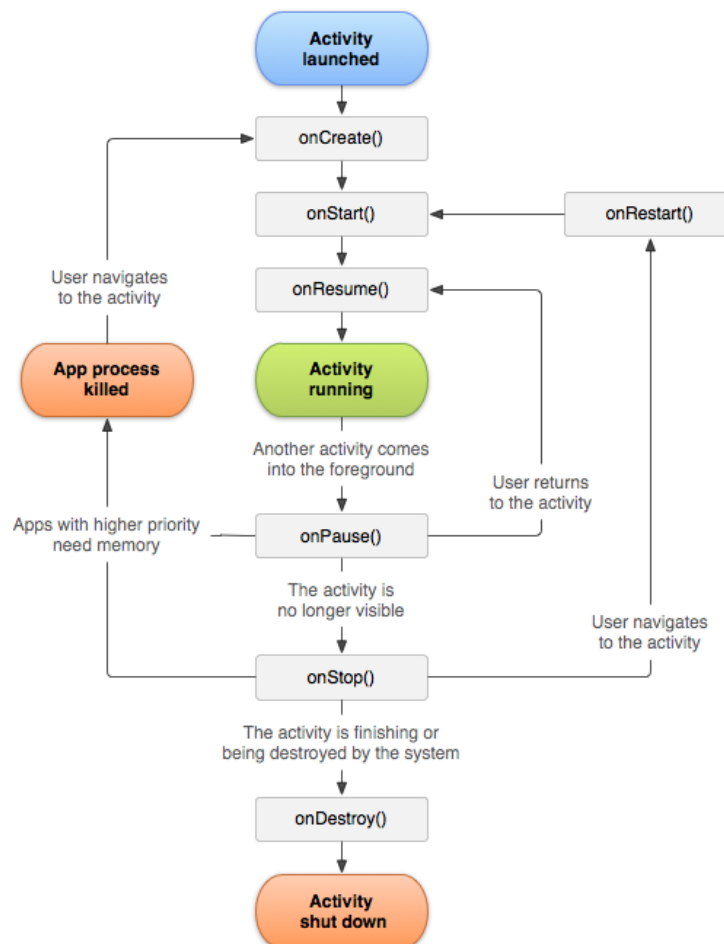
Android 2.2

Android jest systemem operacyjnym przeznaczonym na urządzenia mobilne. Poniżej znajduje się standardowa struktura projektu przeznaczonego na tę platformę.



Pomimo faktu, że system operacyjny Android uruchamia aplikacje napisane w Javie, kładzie on na nie wymóg używania znajdującego się w Android SDK Android Frameworka, który umożliwia zarządzanie całym cyklem życia aplikacji.

Każde okno wyświetlane w systemie jest reprezentowane przez obiekt klasy dziedziczącej klasę Activity. Każda aktywność cechuje się następującym cyklem życia:



Pierwszą uruchamianą metodą w naszej aktywności jest `onCreate()`. To w niej zawsze pobieramy nasz interfejs oraz wszystkie potrzebne uchwytty do widoków.

Każdy jeden interfejs przedstawiający pojedyncze okno jest reprezentowany przez plik XML znajdujący się w katalogu `res/layout`. Katalog ten zawiera zbiór domyślnych layoutów, niemniej jednak istnieje możliwość utworzenia dodatkowych katalogów zawierających layouty `res/layout-land` oraz `res/layout-port`, które będą reprezentować dane okno tylko w konkretnej orientacji urządzenia.

Istnieje możliwość tworzenia aplikacji obsługującej w bardzo prosty sposób wiele wersji językowych. W projekcie katalog „values” jest tworzony automatycznie i zawiera pliki XML reprezentujące domyślne stringi wyświetlane przez aplikację. Istnieje jednakowoż możliwość tworzenia dodatkowych katalogów o nazwach „values-<oznaczenie_narodowe>” (np. „values-pl”) i tworzeniu w nim analogicznych plików do tych znajdujących się w katalogu values, ale zawierających stringi zapisane w innym języku. Dzięki temu, jeśli użytkownik ma ustawiony w systemie urządzenia język, który jest

obsługiwany przez aplikację, zostanie on przez nią automatycznie wybrany. W przeciwnym razie użyte zostaną stringi domyślne, znajdujące się w katalogu „values”.

Każda aplikacja androidowa posiada plik AndroidManifest.xml. Zawiera on podstawowe informacje o aplikacji takie jak zasoby, z których aplikacja będzie korzystać (np. dostęp do internetu, GPS, aparat), nazwa aplikacji, czy jej ikona oraz spis wszystkich aktywności pojawiających się w aplikacji wraz z domyślnymi ich opcjami takimi jak nazwa okna, czy domyślne ułożenie ekranu.

Pakiety niestandardowe

Poza Android Frameworkiem w projekcie użyty jest tylko jeden pakiet wychodzący poza standard Javy – JDBC Connector. Jest to plik JAR wydany przez Oracle na licencji Open Source umożliwiający pisanie aplikacji w Javie, korzystających z bazy danych MySQL.

W procesie pisania aplikacji tworzona jest też biblioteka ServerCommunicator, która jest wykorzystana do komunikacji klienta z serwerem.

Przypadki użycia

Opisowe

Id.	1	
Cel	Wylogowanie	
Warunek początkowy	Użytkownik jest zalogowany i otwarty jest główny ekran aplikacji	
Kroki		
1.1 Użytkownik przyciska przycisk urządzenia odpowiedzialny za otwieranie menu	1.2 Aplikacja wyświetla menu zawierające przycisk „Wyloguj”	
1.3 Użytkownik przyciska przycisk „Wyloguj”	1.4 Aplikacja wyświetla informację o tym, że użytkownik został wylogowany	
Warunek końcowy	Aplikacja wyświetla ekran logowania	

Id.	2	
Cel	Zalogowanie	
Warunek początkowy	Użytkownik dopiero co włączył aplikację i nie jest jeszcze zalogowany	
Kroki		
		2.1 Aplikacja wyświetla ekran powitalny z zapytaniem o login i hasło użytkownika *
2.2 Użytkownik wpisuje login i hasło oraz zatwierdza je, wciskając przycisk „Połącz”		2.3 Aplikacja przechodzi do głównego ekranu *
Wyjątki		
2.1 Jeśli jest to pierwsze uruchomienie aplikacji, aplikacja najpierw wyświetla ekran z zapytaniem o adres serwera (przypadek użycia 3)		
2.3 Jeśli podane przez użytkownika login lub hasło są niepoprawne, krok się nie spełnia, a zamiast tego aplikacja wyświetla informację o błędnych danych do logowania		

Id.	3	
Cel	Zapisanie adresu hosta przy pierwszym uruchomieniu	
Warunek początkowy	Aplikacja uruchamiana jest po raz pierwszy i wyświetla komunikat powitalny	
Kroki		
3.1 Użytkownik podaje w jedyne wolne pole adres serwera i przyciska przycisk „Połącz”		3.2 Aplikacja wyświetla ekran logowania *
Wyjątki		
3.2 Jeśli podany przez użytkownika adres serwera nie jest właściwy, ekran się nie zmienia i aplikacja wyświetla komunikat o błędzie połączenia		
Warunek końcowy		
Aplikacja wyświetla ekran logowania		

Id	4	
Cel	Przeglądanie paczek przypisanych, ale nie dostarczonych	
Warunek początkowy	Użytkownik jest zalogowany i otwarty jest główny ekran aplikacji	
Kroki		
4.1 Użytkownik przyciska przycisk "Paczki przypisane"		4.2 Aplikacja wyświetla ekran z paczkami, które są przypisane użytkownikowi, ale niedostarczone
Warunek końcowy	Aplikacja wyświetla ekran paczek przypisanych, niedostarczonych	

Id	5	
Cel	Przeglądanie paczek dostarczonych	
Warunek początkowy	Użytkownik jest zalogowany i otwarty jest główny ekran aplikacji	
Kroki		
5.1 Użytkownik przyciska przycisk "Paczki dostarczone"		5.2 Aplikacja wyświetla ekran z paczkami, które są użytkownik dostarczył.
Warunek końcowy	Aplikacja wyświetla ekran paczek dostarczonych	

Id	6	
Cel	Przeglądanie szczegółowych informacji dotyczących paczki	
Warunek początkowy	Użytkownik jest zalogowany i otwarty jest ekran 4	
Kroki		
6.1 Użytkownik wybiera paczkę z listy, klikając na jej typ lub adres		6.2 Aplikacja wyświetla ekran z informacjami na temat paczki: datą nadania, adresem.
Warunek końcowy	Aplikacja wyświetla ekran z informacjami o paczce	

Id	7	
Cel	Przypisanie paczki	
Warunek początkowy	Użytkownik jest zalogowany i wyświetlony jest ekran z informacjami o paczce,	

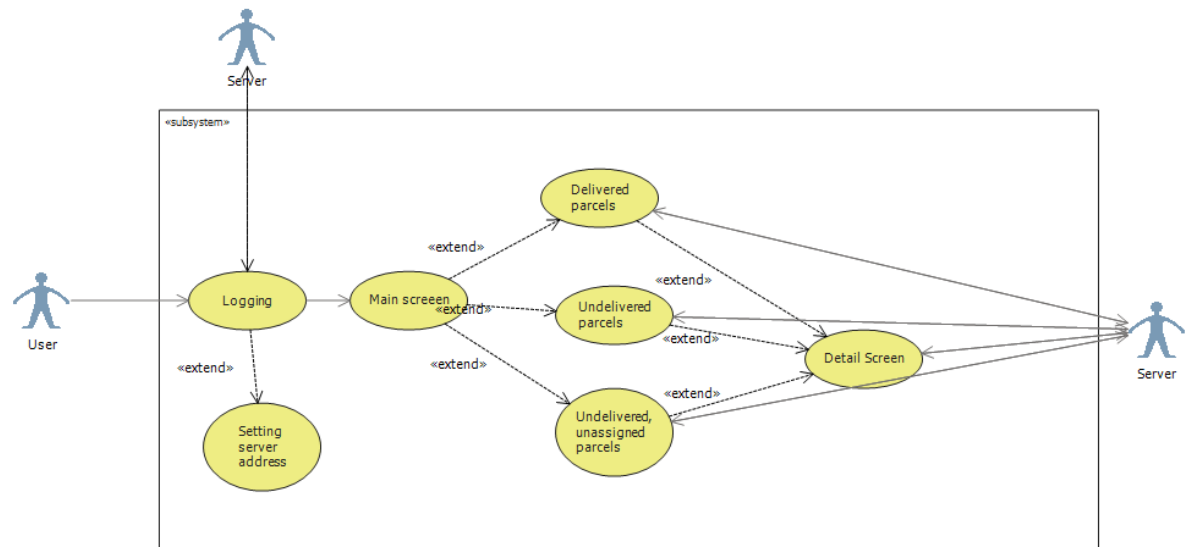
	do którego przeszło się z paczek nieprzypisanych.
Kroki	
7.1 Użytkownik przyciska przycisk "Przypisz"	7.2 Paczka zostaje przeniesiona do listy "Paczek przypisanych, niedostarczonych"
Warunek końcowy	Aplikacja wyświetla ekran listy paczek przypisanych

Id	8
Cel	Zatwierdzenie dostawy
Warunek początkowy	Użytkownik jest zalogowany i wyświetlony jest ekran z informacjami o paczce, do którego przeszło się z paczek przypisanych, niedostarczonych.
Kroki	
8.1 Użytkownik przyciska przycisk "dostarczono"	8.2 Paczka zostaje przeniesiona do listy "paczek dostarczonych"
Warunek końcowy	Aplikacja wyświetla ekran listy paczek przypisanych, niedostarczonych

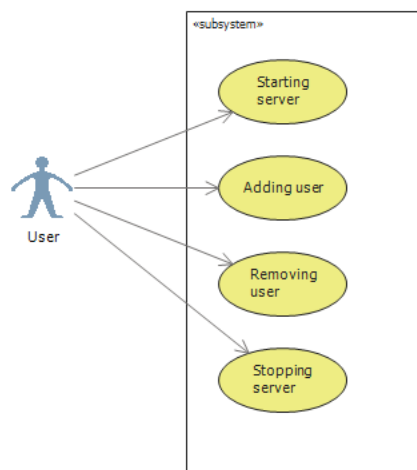
Id	9
Cel	Powrót do poprzedniej strony
Warunek początkowy	Użytkownik jest zalogowany i przeszedł z menu głównego do innego menu
Kroki	
9.1 Użytkownik przyciska przycisk standardowy przycisk wycofania (zgodny ze specyfikacją Androida 2.2)	9.2 Wyświetlony zostaje ekran ze poprzednią stroną
Warunek końcowy	Wyświetlony zostaje ekran ze stroną poprzednią

Diagramy

Klient



Server



Diagramy klas

Aktywności klienta

cm

Komunikacja po stronie klienta

gfe

Komunikacja po stronie serwera



Diagramy sekwencji

Logowanie



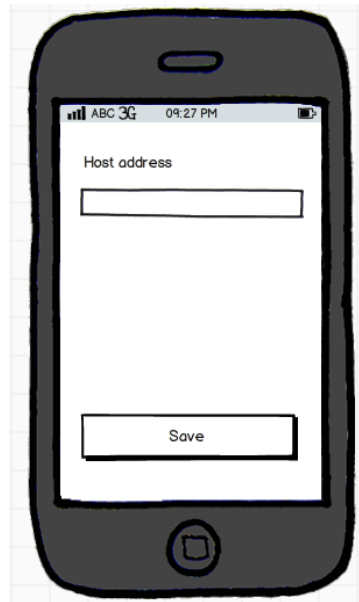
Przetworzenie wiadomości przez serwer



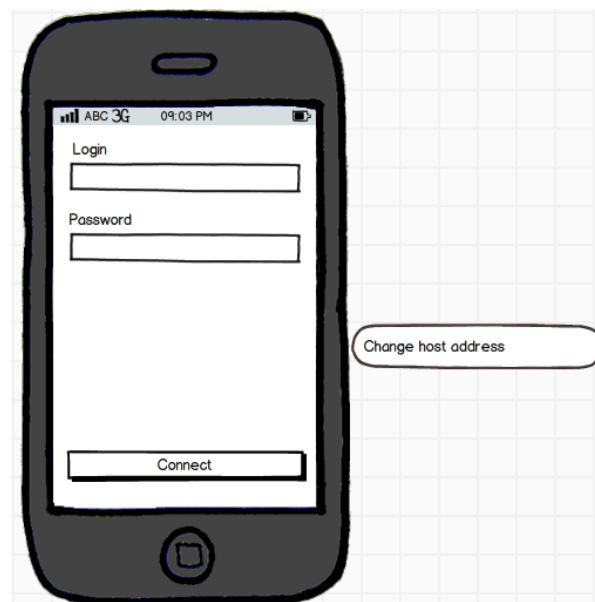
Ekrany

Opis ekranów

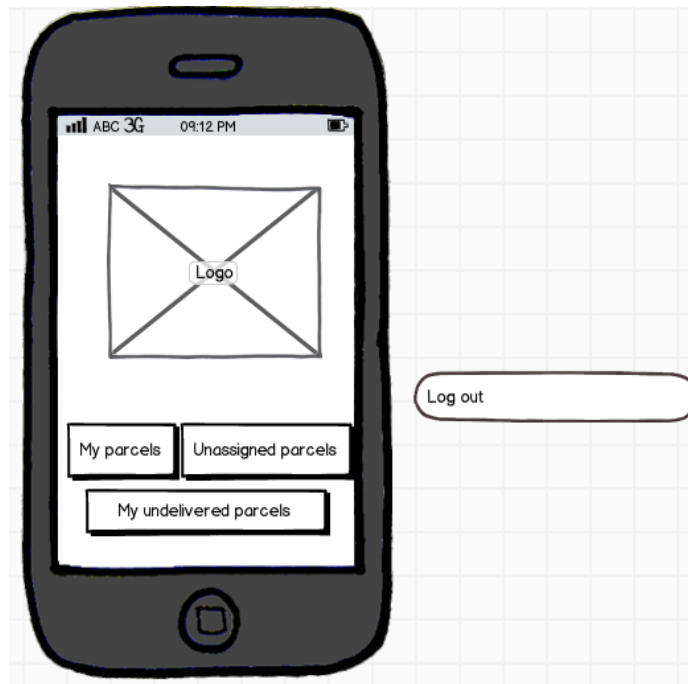
1. Okno wyświetlane przy pierwszym uruchomieniu, z zapytaniem o adres serwera. Można do niego wrócić poprzez przycisk znajdujący się w menu w ekranie logowania



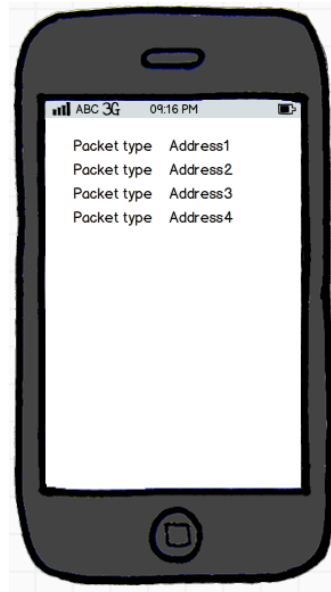
2. Okno logowania



3. Główny ekran aplikacji



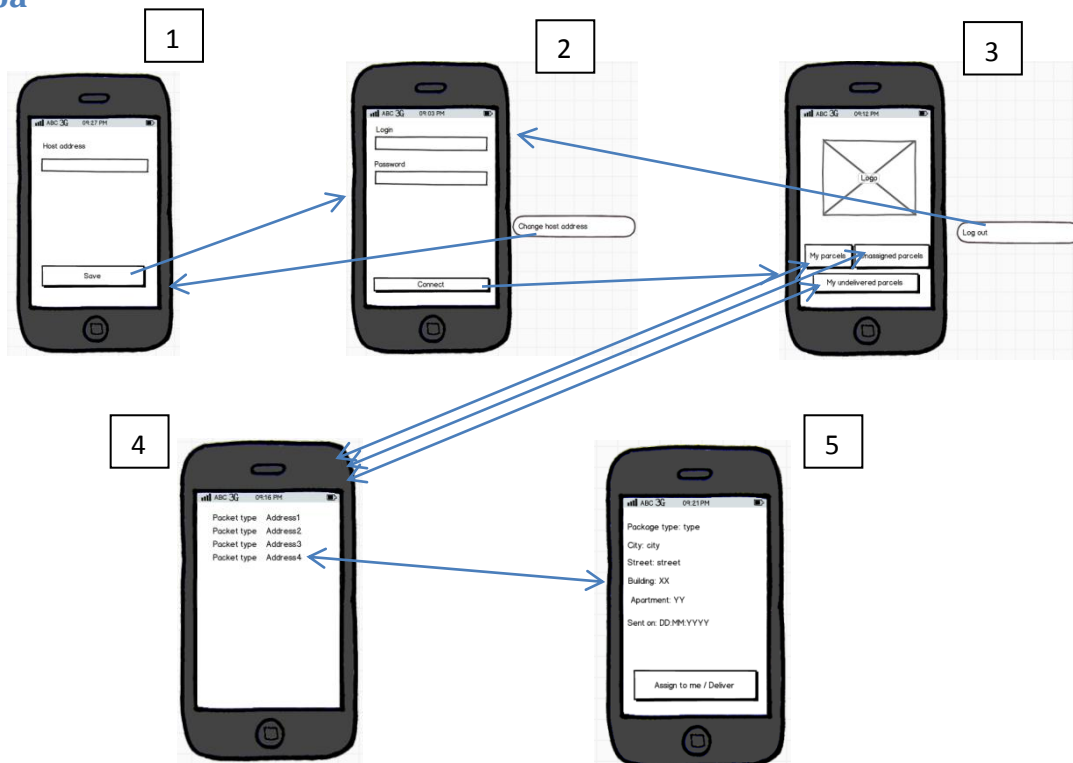
4. Ekran paczek (wyświetlany niezależnie od wybranej opcji w głównym ekranie, ale w zależności od przyciśniętego przycisku, posiadający inne wypełnienie)



5. Okno szczegółowych danych paczki



Mapa



Plan testów

1. Równoczesne napisanie serwera, oraz biblioteki ServerCommunicator, służącej do prostej komunikacji klienta z serwerem, a także testów jednostkowych dla każdego z dostępnych zapytań do serwera:
 - a) Zapytanie o logowanie
 - b) Zapytanie o paczki dostępne w magazynie
 - c) Zapytanie o paczki przypisane do aktualnie zalogowanego kuriera

- d) Zapytanie o paczki przypisane do aktualnie zalogowanego kuriera, ale które nie zostały jeszcze dostarczone
2. Napisanie ekranów klienta oraz testów jednostkowych dla samych ekranów.
3. Wprowadzenie użycia biblioteki ServerCommunicator do klienta oraz przeprowadzenie testów regresji.

Możliwe rozszerzenia

1. Możliwość znajdowania optymalnej trasy na podstawie niedostarczonych paczek.

Słownik danych

1. Wystarczające uprawnienia – uprawnienia do wykonywania aplikacji, do zapisu i odczytu w katalogu domowym aplikacji oraz wszystkich jego podkatalogach oraz do zapisu, odczytu i wykonania wszystkich plików znajdujących się w katalogu domowym aplikacji oraz wszystkich jego podkatalogach.
2. Log – krótka informacja dotycząca zapisana w pliku z rozszerzeniem .log dotycząca konkretnego zdarzenia zaistniałego pomiędzy klientem, a serwerem. Powinna składać się co najmniej z czasu zaistniałego zdarzenia (podanego w formacie timestamp, lub „dzień-miesiąc-rok godzina:minuta:sekunda, przy czym godzina jest podana w formacie 24-godzinny, a każda ze zmiennych poza rokiem, składa się z dwóch cyfr), oddzielonego od daty tabulatorem typu zdarzenia (EVENT lub ERROR) oraz krótkiego opisu zdarzenia zapisanego w języku angielskim, np.
 - a) 02-01-2012 13:06:59 EVENT: User kjanow logged in
 - b) 02-01-2012 13:07:20 EVENT: User kjanow requested some data
 - c) 02-01-2012 13:07:21 EVENT: Server provided kjanow with requested data
 - d) 02-01-2012 13:07:21 ERROR: Server failed in processing the request
3. Klient – osoba będąca nadawcą, lub odbiorcą paczki, lub w odniesieniu do aplikacji - AndCourier Client.
4. Logowanie – nawiązanie połączenia pomiędzy klientem i serwerem, podanie przez klienta loginu i hasła takich, że istnieje rekord w tabeli couriers w bazie danych, który już je zawiera oraz dodanie rekordu do tabeli logins, który zawiera dodatkowo informację o dokładnym czasie podania informacji do logowania.
5. Wylogowanie – zerwanie połączenia pomiędzy klientem i serwerem, a co za tym idzie zakończenie sesji przez serwer.
6. Widok – każdy pojedynczy element znajdujący się w interfejsie użytkownika (zarówno każda kontrolka jak i każdy layout(sposób ułożenia widoków względem siebie)).
7. Aktywność – klasa dziedzicząca klasę Activity, oraz zadeklarowana w manifeście. Jest zbiorem metod obsługujących określony interfejs użytkownika.

Bibliografia

1. Oficjalna wydana przez Google Corp. dokumentacja Androida – Android Developers:
<http://developer.android.com>
2. Oficjalna wydana przez Oracle dokumentacja Javy:
<http://docs.oracle.com/>

3. Oficjalna wydana przez Oracle dokumentacja JDBC Connector:
<http://dev.mysql.com/doc/refman/5.5/en/connector-j.html>

Załączniki

1. AndCourier Server – instrukcja obsługi
2. AndCourier Client – instrukcja obsługi