Komunikator  
Plan testów  
Krzysztof Gwozdowicz  
Wersja 1.0

**I Wstęp**

1.Tworzenie nazw plików

Pliki zawierające opisy testów powinny zawierać w nazwie swój unikalny ID, nazwe testowanej funkcjonalności (maksymalnie 4 słowa) wszystko oddzielane twardymi spacjami. Przykład: 151\_Instalacja\_Android\_4.4\_offline

2.Wprowadzenie

Wprowadzenie do projektu.

3. Cele

Celem jakościowym projektu jest osiągnięcie poziomu, w którym nie występują poważne i nawracające błędy krytyczne w trakcie użytkowania aplikacji, a użytkownik aplikacji jest w stanie bez problemu połączyć się z serwerem, odczytywać oraz wysłać wiadomości.

Celem kosztowym jest nie przekroczenie 15 złotych.

Celem czasowym jest zakończenie fazy testowania do 12.06.2017

4. Podejście do testów

Strategia testów mająca doprowadzić do osiągnięcia pożądanych celów.

**II Opis testów**

1. Testowany obiekt

1.1. Obiekt: Aplikacja  
1.1.1. Główny plik wykonalny  
1.1.2. Instalacja/deinstalacja  
1.1.3. Funkcje/Narzędzia

1.2. Obiekt: Serwer  
1.1.1. Główny plik wykonalny  
1.1.2. Instalacja/deinstalacja  
1.1.3. Funkcje/Narzędzia

2. Funkcjonalność do przetestowania

2.1 Aplikacja klienta

2.1.1. Instalacja aplikacji

2.1.2. Dezinstalacja aplikacji

2.1.3. Uruchomienie aplikacji

2.1.4. Wyłączanie aplikacji

2.1.5. Wprowadzanie nazwy użytkownika

2.1.6. Łączenie się z serwerem

2.1.7. Zmiana kanału

2.1.8. Wysyłanie wiadomości

2.1.9. Otrzymywanie wiadomości

2.1.10. Elementy wyświetlane na ekranie

2.1.11. Rozłączanie się z serwerem

2.1.12. Działanie elementów gui

3.1 Aplikacja serwera

3.1.1. Instalacja serwera

3.1.2. Uruchomienie serwera

3.1.3. Działanie serwera

3.1.4. Obsługa wiadomości

3.1.5. Obsługa użytkowników

3.1.6. Dezinstalacja serwera

3. Funkcjonalność nietestowana

4. Wymagania systemowe

Urządzenie mobilne z systemem Android w wersji 4.0.3 lub nowszym z dostępem do internetu

Urządzenie stacjonarne z systemem Windows 7 lub nowszym z dostępem do internetu.

5. Wejście/Wyjście dla procesu tworzenia produktu

1. Serwer oraz aplikacja mobilna mogą zostać uruchomione

2. Aplikacja mobilna jest w stanie się połączyć z serwerem

3. Aplikacja mobilna jest w stanie łączyć się i przełączać między kanałami na serwerze.

6. Standardy/Bibliografia  
- IEEE Standard for Software Test Documentation (ANSI/IEEE std 829)  
- Kaner et al. Testing Computer Software, 2nd edition. New York: Wiley, 1993

7. Dostawy testowe

Paczka testów otrzymana w dniu 01.05.2017 od testerów znajdująca się na serwerze projektu

7.1 Plan testów

7.1.1 Wstępny plan testów (zaaprobowany)

7.1.2 Dokumenty dodatkowe do planu wykonanych testów  
Tablice przypadków testowych, matryce oraz inne materiały powiązane z testami z określonymi potwierdzeniami weryfikacji i ukończeniem testów.

7.1.3 Finalny plan testów(zaaprobowany)  
Końcowy plan testów jest pomniejszą wersją planu testów programistycznych. Plan ten jest wyprodukowany i używany w końcowych cyklach testów. Musi zawierać potwierdzenia zgodności z wcześniejszymi planami.

7.1.4. Dokumenty dodatkowe do planu finalnych testów  
Tablice przypadków testowych, matryce oraz inne materiały powiązane z testami z określonymi potwierdzeniami weryfikacji i ukończeniem testów.

7.2. System śledzenia defektów

7.2.1. Raporty defektów  
- Lista podsumowująca znalezione defekty  
- Pełny opis defektów

7.2.2. Baza danych defektów  
Za bazę danych defektów służy dokument o nazwie Bug\_report.xml znajdujący się w folderze Testy na serwerze projektu

7.3 Końcowy raport wydania

----

**III Zarządzanie projektem testowym**

1. Odpowiedzialni za testy  
 -Kacper Tomalik

-Marcin Omelan

-Piotr Frankiewicz

-Krzysztof Gwozdowicz

2. Odpowiedzialni za testy

Tworzenie oraz weryfikowanie testów: Wszyscy w zespole testowym

Wykonywanie testów: Krzysztof Gwozdowicz, Marcin Omelan

Nadzorowanie procesu testowania: Piotr Frankiewicz

3. Zadania testowe  
- Stworzenie planów testów wraz z przypadkami testowymi

- Poddanie planu testów procesowi przeglądu wraz z uzyskaniem odpowiednich zgód  
- Uzyskać wymagania dla sprzętu/oprogramowania/narzędzia  
- Stworzenie bazy danych defektów  
- Przeprowadzenie testów  
- Zaraportowanie defektów  
- Przeprowadzenie spotkania dotyczącego defektów  
- Stworzenie okresowego raportu  
- Stworzenie końcowego raportu

4. Plan testów oraz harmonogram  
Dostawy nowych testów wykonywane będą co tydzień.

Wysyłka efektów testowania wysyłane na repozytorium przeprowadzane będzie na bierząco.

5. Kryteria wejścia i wyjścia dla kamieni milowych projektu  
Definicje, opisy oraz mierzalne kryteria dla kamieni milowych.

6. Harmonogram planu testów

6.1. Harmonogram  
Wszelkie zadania testowe przydzielane będą przez menadżera projektu do konkretnych osób na początku każdego dnia pracy.

6.2. Ocena liczebności zespołów  
Do wykonania projektu testowego potrzebne będą cztery osoby.

7. Potrzeby treningowe  
BRAK

8.Potrzeby środowiskowe

8.1. Komponenty testowe  
- Komputer z dostępem do internetu  
- Urządzenie mobilne z dostępem do internetu

8.2. Narzędzia testowe  
- Komputer z dostępem do internetu  
- Urządzenie mobilne z dostępem do internetu  
- Dokumenty dotyczące testów

8.3. Budynki, sprzęt, serwisy  
Testowanie odbędzie się w domach pracowników biorących udział w projekcie.

9. Plan integracji  
BRAK

10. Zawieszenie i ponowne rozpoczęcie testów  
Testy zawieszane będą w momencie wyczerpania przypadków testowych w danym tygodniu.

Testy wznawiane będą w momencie otrzymania nowych paczek z testami lub w momencie rozpoczęcia nowego okresu testowego

11. Kryterium zakończenia testów  
Okres zakończenia procesu testowania aplikacji określony zostanie przez właściciela projektu.

12. Proces śledzenia defektów

12.1. Proces  
Wszystkie defekty oprogramowania opisywane są w odpowiednich dokumentach.

12.2. Narzędzie śledzenia defektów (baza danych)  
Jest to dokument będący arkuszem kalkulacyjnym.

12.3. Definiowanie ważności defektów  
Ocena ważności defektu jest subiektywną metodą używaną do zaraportowania stopnia ważności każdego zaraportowanego defektu. Defekty oceniane będą na podstawie intuicji osób testujących oprogramowanie.

12.3.1.1. Krytyczny  
Ważność 1— Defekt krytyczny (ang. *show-stopper* - blokujący wydanie) występuje, gdy brakuje podstawowej funkcjonalności, użyteczności lub wydajności produktu podczas normalnych operacji; nie ma możliwości stworzenia "obejścia" (ang. *workaround*).

Zaliczamy do tej kategorii również defekty niezwiązane z funkcjonalnością, ale będące jasnymi pomyłkami w np. nazwie firmy, błędna instrukcja obsługi etc.

Kilka dodatkowych kategorii:  
- Wyłączenie lub uszkodzenie aplikacji/sprzętu  
- Brak sukcesu przy wykonywaniu podstawowych funkcji

12.3.2.2. Poważny  
Ważność 2— Poważny defekt dla najważniejszych funkcjonalności, które nie działają prawidłowo w określonych warunkach lub funkcjonalności drugiego rzędu, które w ogóle nie działają lub słaba jakość funkcjonalności lub wydajności podczas normalnych operacji, trudności z użyciem podstawowych funkcjonalności etc.

Zasadniczo, tego rodzaju defekty powinny zostać naprawione w cyklu tworzenia oprogramowania. Jednak w ostatecznej fazie testowej mogą zostać zaakceptowane (znaczy nienaprawione).

12.3.3.3. Niekrytyczny  
Ważność 3 — Niekrytyczne defekty to te z nich, które wprowadzają pewne problemy w obsłudze, niezdarzające się zbyt często. Zaliczamy tutaj: pomniejsze problemy w wyświetlaniu, błędy językowe, pomniejsze problemy z działaniem oraz podobne. Zazwyczaj błędy tego typu powinny być naprawione kiedy czas pozwoli z minimalnym nakładem sił przez programistów.  
Wiele małych defektów może się przełożyć na ogólną niską ocenę jakości produktu.

13. Śledzenie statusu testów i raportowanie  
Raporty testów tworzona oraz raportowane do właściciela projektu co dwa tygodnie.

14. Ryzyko i incydenty  
Ryzyko głównie wiąże się z niedotrzymaniem terminów.

15. Proces potwierdzania zgodności

15.1. Potwierdzanie zgodności planu testów  
Za tworzenie oraz zgodność planu testów odpowiedzialny jest Krzysztof Gwozdowicz

15.2 Potwierdzanie zgodności końcowej wydanego produktu  
Potwierdzeniem zgodności końcowej wydanego produktu zajmuje się Piotr Frankiewicz