



S.A.F.MILKY

Test Plan

1.0



S.A.F. MILKY
Smart Artificial Farming Milky

Data: 10/12/2021



Progetto: S.A.F.MILKY	Versione: 1.0
Documento: Test Plan	Data: 10/12/2021

Coordinatore del progetto:

Nome	Matricola
Alfonso Schettino	0512105445
Antonio Giorgio	0512106036

Partecipanti:

Nome	Matricola
Alfonso Schettino	0512105445
Antonio Giorgio	0512106036

Scritto da:	Alfonso Schettino, Antonio Giorgio
--------------------	------------------------------------

Revision History:

Data	Versione	Descrizione	Autori
10/12/2021	1.0	Introduzione	Alfonso Schettino, Antonio Giorgio
10/12/2021	1.0	Relazione con altri documenti	Alfonso Schettino, Antonio Giorgio
10/12/2021	1.0	Panoramica del sistema	Alfonso Schettino
10/12/2021	1.0	Fetures da testare	Antonio Giorgio
10/12/2021	1.0	Pass/Fail criteria	Alfonso Schettino
10/12/2021	1.0	Approccio	Antonio Giorgio
10/12/2021	1.0	Sospensione e ripristino	Antonio Giorgio



10/12/2021	1.0	Materiale di testing	Alfonso Schettino
10/12/2021	1.0	Test cases	Alfonso Schettino
10/12/2021	1.0	Testing schedule	Antonio Giorgio



Sommario

1. Introduzione	5
2. Relazione con altri documenti	6
3. Panoramica del sistema	7
4. Features da testare	7
5. Pass/Fail criteria	8
6. Approccio	8
7. Sospensione e ripristino	9
8. Materiale di testing	10
9. Test cases	10
10. Testing schedule	14

1. Introduzione

S.A.F.MILKY è stato pensato per facilitare l'interazione tra i clienti e l'azienda casearia, al fine di semplificare le operazioni di acquisto e controllo degli acquisti effettuati.

Il documento di Test Plan ha l'obiettivo di descrivere ed analizzare le attività di testing per l'app mobile S.A.F.MILKY.

Il fine è quello di garantire che ogni aspetto funzioni in modo corretto.

All'interno del documento sono riportate le strategie di testing adottate, quali funzionalità saranno testate e gli strumenti scelti per la rilevazione degli errori, con lo scopo di presentare ai clienti finali un'applicazione mobile priva di malfunzionamenti.

Sono state pianificate attività di testing per le seguenti gestioni:

- Gestione autenticazione;
- Gestione acquisti;
- Gestione informazioni;

2. Relazione con altri documenti

Per la corretta individuazione dei test case, si fa riferimento ad altri documenti prodotti:

Relazioni con il Requirements Analysis Document (RAD)

I test case pianificati nel Test Plan sono elaborati in relazione ai requisiti funzionali e non funzionali presenti nel RAD.

Relazioni con il System Design Document (SDD)

I test case pianificati nel Test Plan devono rispettare la suddivisione in sottosistemi presentata nell'SDD.

Relazioni con l'Object Design Document (ODD)

Per ciò che concerne il test di unità e di integrazione, maggiormente legati all'ODD e alla divisione in package del sistema, essi saranno scritti e documentati unicamente all'interno del codice dell'applicativo.

Per tale motivo, nel presente documento, non vi saranno riferimenti al loro design.

3. Panoramica del sistema

Il sistema proposto basa la sua architettura sul sistema three-tier.
Come linguaggi di programmazione verranno usati:

- Java;
- Xml;
- SQL.

Per la creazione dell'app verrà utilizzato Android Studio, mentre per la creazione del database verrà utilizzato MySQL.

4. Features da testare

Di seguito la lista delle features di cui si effettuerà il testing per le varie gestioni:

Gestione autenticazione:

- Ricerca cliente specifico;
- Autenticazione del cliente;
- Ricerca di tutti i clienti registrati;
- Registrazione nuovo cliente.

Gestione acquisti:

- Completamento acquisti;
- Aggiunta di un ordine;
- Ricerca degli ordini;
- Aggiunta del prodotto al carrello;
- Rimozione del prodotto dal Carrello;
- Ricerca dei prodotti inseriti nel carrello.

Gestione informazioni:

- Ricerca informazioni azienda;
- Ricerca informazioni prodotto;
- Ricerca dati del cliente;
- Modifica dati del cliente.

5. Pass/Fail criteria

Le attività di testing sono mirate ad identificare la presenza di faults (errori) all'interno del sistema, per effettuare un successivo intervento di eliminazione.

L'esito di un test case è valutato mediante un oracolo, inteso come il risultato atteso della sua esecuzione, basandosi sui requisiti.

Un test ha successo (pass) se, dato un input al sistema, l'output ottenuto è diverso dall'output atteso dall'oracolo.

Un test fallisce (fail) se, dato un input al sistema, l'output ottenuto è uguale all'output atteso dall'oracolo.

Tutto il testing sarà considerato valido se tutti i seguenti vincoli saranno rispettati:

- Testare tutti i requisiti funzionali;
- Effettuare test di regressione ogni volta che si introducono nuove caratteristiche al sistema o vengono modificate quelle presenti;
- Raggiungere un branch coverage non inferiore al 75%.

6. Approccio

Il testing dell'intero sistema si compone di tre fasi:

- 1) Testing di sistema;
- 2) Testing di integrazione;
- 3) Testing di unità.

Prima della fase di implementazione del sistema, avverrà la progettazione dei casi di test di sistema, perfezionati in seguito nella loro fase di esecuzione.

Durante la fase di implementazione avverrà la progettazione dei casi di test di unità.

Durante lo sviluppo saranno eseguite periodiche attività di revisione sul codice prodotto.

Testing di sistema

Per il testing di sistema sarà utilizzato TestRecorder di Android Studio che permette di registrare le azioni che un utente può intraprendere sull'applicazione, in modo da poter implementare ed eseguire i test case di sistema.

Testing di integrità

Verrà utilizzato un approccio bottom-up, metodo ritenuto più adatto per un software basato sul paradigma Object Oriented.

Testing di unità

Per il testing di unità la strategia prevista consiste nel testare ogni metodo delle classi.

I casi di test saranno definiti attraverso un approccio black-box e saranno documentati direttamente nel codice.

7. Sospensione e ripristino

In questa sezione verranno specificati i criteri di sospensione del test e le attività di test che dovranno essere ripetute quando si riprende il test.

Criteri di sospensione

Il testing non verrà sospeso fino alla sua terminazione, anche in caso di rilevazione di una failure.

Il testing potrà essere momentaneamente sospeso nel caso venga restituito, al momento dell'esecuzione, un errore nella definizione di uno dei test stessi.

Criteri di ripristino

Il testing verrà ripreso dopo aver risolto i failure.

8. Materiale di testing

L'hardware necessario per l'attività di test è un semplice cellulare android.

9. Test cases

Gestione autenticazione

TC_GA_1.1 Login

Parametro: Email Formato: [A-Z a-z 0-9 @.]	
Categorie	Scelte
Match mu	1: Mismatch con username [errore] 2: Match con username [property muOk]

Parametro: Password Formato: [A-Z a-z 0-9 ._%+~]	
Categorie	Scelte
Match mp	1: Mismatch con password [errore] 2: Match con password [property mpOk]

Category partition

Codice	Combinazione	Esito
TC_GA_1.1_1	mu1	Errore: Username errato
TC_GA_1.1_2	mu2.mp1	Errore: Password errata
TC_GA_1.1_3	mu2.mp2	Corretto

TC_GA_1.2 Registrazione cliente

Parametro: Nome Formato: [A-Z a-z]	
Categorie	Scelte
Lunghezza In	1: lunghezza <4 [errore] 2: Lunghezza >=4 [property InOk]

Parametro: Cognome Formato: [A-Z a-z]	
Categorie	Scelte
Lunghezza Ic	1: lunghezza <4 [errore] 2: Lunghezza >=4 [property IcOk]

Parametro: Cf Formato: [A-Z a-z 0-9]	
Categorie	Scelte
Lunghezza lcf	1: lunghezza <16 [errore] 2: lunghezza >16 [errore] 2: lunghezza = 16 [property lcfOk]

Parametro: Email Formato: [A-Z a-z 0-9 @.]	
Categorie	Scelte
Match mu	1: Mismatch con email [errore] 2: Match con email [property muOk]

Parametro: Password Formato: [A-Z a-z 0-9 ._%+-]	
Categorie	Scelte
Match mp	1: Mismatch con password [errore] 2: Match con password [property mpOk]

Category partition

Codice	Combinazione	Esito
TC_GA_1.2_1	ln1	Errore: lunghezza <4
TC_GA_1.2_2	ln2.lc1	Errore: lunghezza <4
TC_GA_1.2_3	ln2.lc2.lcf1	Errore: lunghezza <16
TC_GA_1.2_4	ln2.lc2.lcf1.mu1	Errore: mismatch
TC_GA_1.2_5	ln2.lc2.lcf1.mu2.mp1	Errore: mismatch
TC_GA_1.2_	ln2.lc2.lcf1.mu2.mp2	Corretto

Gestione acquisti

TC_GA_1.3 Completamento acquisto

Parametro: Titolare carta Formato: [A-Z a-z]	
Categorie	Scelte
Lunghezza ltc	1: lunghezza <4 [errore] 2: Lunghezza >=4 [property ltcOk]

Parametro: Numero carta Formato: [0-9]	
Categorie	Scelte
Lunghezza Inc	1: lunghezza <9 [errore] 2: Lunghezza >=9 [property IncOk]

Parametro: Cvv Formato: [0-9]	
Categorie	Scelte
Lunghezza lcv	1: lunghezza <3 [errore] 2: Lunghezza =3 [property lcvOk]

Category partition

Codice	Combinazione	Esito
TC_GA_1.3_1	ltc1	Errore: lunghezza <4
TC_GA_1.3_2	ltc2.Inc1	Errore: lunghezza <9
TC_GA_1.3_3	ltc2.Inc2.lcv1	Errore: lunghezza <3
TC_GA_1.3_4	ltc2.Inc2.lcv2	Corretto

10. Testing schedule

Le attività di pianificazione del testing avverranno come definito nei capitoli precedenti, cioè subito dopo la fase di design necessaria per la pianificazione.

La scrittura dei casi di test avverrà in contemporanea con lo sviluppo del codice.

L'esecuzione dei test avverrà sia durante che dopo l'implementazione del sistema.

Una volta concluso lo sviluppo, tutti i test saranno rieseguiti per garantire il corretto funzionamento e produrre i report finali.