



Università
Ca' Foscari
Venezia

Ingegneria del software 20/21

Piano di testing

SNASA

Giulio Nicola 875297

Alessandro Piazza 869386

Leonardo Mazzon 868445

Klevis Hasa 869641

Indice

1. Panoramica	3
1.1. Scopo	3
1.2. Ambito	4
2. Riepilogo dei test	4
2.1. Ambito dei test	4
3. Analisi del campo e test del focus	5
3.1. Contenuto rilasciato	5
3.2. Test di regressione	5
3.3. Test delle piattaforme	5
4. Obiettivi del test di progressione	6
5. Altri Test	7
5.1. Sicurezza	7
5.2. Stress test & Test di volume(S&T)	7
5.3. Test di connessione(CT)	7
5.4. Ripristino di emergenza/Backup	8
5.5. Testing di unità	8
5.6. Test di Integrazione	8
6. Strategie dei Test	9
6.1. Responsabilità Livelli di Test	9
6.2. Tipo di test e approccio	9
6.3. Build Strategy	10
6.4. Pianificazione esecuzione dei test	10
6.5. Servizi, dati e pianificazione della fornitura di risorse	11
6.5.1. Ambiente di test	11
6.5.2. Accesso ad altre applicazioni	11
6.5.3. Requisiti dei tester	11
6.5.4. Requisiti dei dati	11
6.5.5. Risorse & Conoscenze	11
6.6. Strumenti di test	12
6.7. Procedura di consegna dei test	12
6.8. Metriche dei test	12

7. Piano dell'ambiente dei test	13
7.1. Gestione dell'ambiente dei test	13
7.2. Dettagli dell'ambiente di test	13
7.2.1. Testers	13
7.2.2. Hardware e Firmware	14
7.2.3. Software	14
7.2.4. Interfacce	14
7.2.5 Altri materiali	14
7.3. Ambiente da stabilire	14
7.4. Controllo ambiente	15
7.5. Ruoli nell'ambiente e responsabilità	15
8. Assunzioni	16
8.1. Assunzioni	16
8.2. Dipendenze	16
9. Criteri di entrata ed uscita	16
10. Piano Amministrativo	17
10.1. Approvazioni	17
10.2. Programma di Test Milestones	18
10.3. Training	19
10.4. Gestione dei difetti	19
11. Definizioni	20
12. Riferimenti	21
13. Contatti	22

Versione	Data
1.0	13/11/2020
1.1	18/11/2020
1.2	13/01/2021

1. Panoramica

1.1. Scopo

Questo documento tratta le varie attività di testing che verranno effettuate con lo scopo di verificare il corretto funzionamento dell'applicazione.

- **Lo scopo dei test:** Trovare possibili malfunzionamenti nell'applicazione e verificarne l'integrità prima che venga rilasciata l'applicazione o un aggiornamento;
- **Responsabilità dei test:** I test verranno suddivisi tra i vari componenti del gruppo e, se necessario, persone terze;
- **Strategie di testing:** Come strategie di testing abbiamo utilizzato *White-Box Testing* e *Back-to-Back Testing*;
- **Criteri di entrata e uscita:** Sono un insieme di condizioni che devono essere soddisfatte per creare una condizione adatta e favorevole per i test;
- **Stime dei test:** Valutando le componenti a rischio, individueremo i test che necessitano di una tempistica maggiore;
- **Rischi, problemi, assunzioni e dipendenze dei test:** Evidenziano eventuali aspetti da considerare per garantire un corretto funzionamento dell'applicazione;
- **Pianificazione dei test :** Comprende l'organizzazione, la distribuzione delle attività di testing, delle risorse e del tempo assegnato ad ogni test;
- **Risultati dei test:** I risultati verranno analizzati dal gruppo per poter avere una maggior consapevolezza delle varie problematiche.

1.2. Ambito

L'ambito di questo documento è quello di descrivere la modalità di redazione del piano di test relativo a un progetto software, e trattare le varie fasi di testing che verranno adottate dal team di sviluppo per fornire anche al cliente una panoramica relativa ai test che verranno effettuati ed ai risultati ottenuti.

La fase iniziale include tutti i requisiti indispensabili, alla fine della fase iniziale un tester deve essere in grado di:

- Creare un test manuale con tutti i passaggi necessari
- Salvarlo
- Recuperarlo ed avere la possibilità di visualizzarlo durante l'esecuzione
- Inserire i risultati e i commenti appropriati
- Visualizzare i risultati

Man mano che il team lavorerà con il prodotto si definiranno le esigenze per la seconda fase.

2. Riepilogo dei test

2.1. Ambito dei test

In campo

Lo scopo del testing sarà quello di testare tutte le classi implementate, le modifiche, le funzionalità e il corretto avvio e chiusura dell'applicazione. Terminata la fase di sviluppo dell'applicazione, la versione alfa verrà provata dai componenti del gruppo; dopo un'attenta analisi e la risoluzione di possibili errori dell'applicazione si passerà ad una versione beta.

Fuori campo

Dopo lo sviluppo della prima versione dell'applicazione verrà effettuato un test su tutto il sistema da parte di utenti selezionati dal team, che poi forniranno dei feedback per possibili future implementazioni o malfunzionamenti. Questi test sono importanti al fine di facilitare l'usabilità dell'applicazione prima del rilascio ufficiale.

3. Analisi del campo e test del focus

3.1. Contenuto rilasciato

Per il contenuto del progetto fare riferimento al documento di specifica dei requisiti.

3.2. Test di regressione

Sarà necessario effettuare un test di regressione quando si aggiungerà una nuova funzione al software, quando si eseguiranno riparazioni di bug ed errori di elaborazione e quando si risolveranno problemi di performance così da verificare che le funzionalità precedentemente testate funzionino correttamente.

3.3. Test delle piattaforme

L'applicazione verrà testata alternando l'emulatore android di Android Studio a dispositivi android fisici tra cui :

- Xiaomi Redmi 4 Pro con Android 10.0;
- Xiaomi Redmi Note 8 Pro con Android 10.0;
- Huawei Honor 6C Pro con Android 7.0;
- Huawei Honor 9 con Android 9.0;
- OnePlus 5T con Android 10.0;
- OnePlus One con Android 10.0
- Samsung Galaxy S2 con Android 10.0

4. Obiettivi del test di progressione

Questa sezione servirà come traccia delle macro funzioni fornite dall'applicazione descritte sul documento dei requisiti.

REF	FUNZIONE	OBIETTIVI TEST	CRITERIO DI VALUTAZIONE	RIFERIMENTO	PRIORITÀ
Test che saranno effettuati					
SRF0	Main Activity e componenti grafiche.	Visualizzazione home con navigazione, default nel client	Nessun errore nella visualizzazione.	Documento dei requisiti	Media
SRF1	Nav Component.	Cambio sezione.	Nessun errore nella navigazione.	Documento dei requisiti	Media
SRF2	Nearby Connections.	Ricerca dispositivi server.	Il client trova il server.	Documento dei requisiti	Alta
SRF3	Nearby Connections.	Disponibilità alla connessione.	Il server riceve richieste di connessione.	Documento dei requisiti	Alta
SRF4	Nearby Connections.	Il dispositivo viene connesso.	Due o più dispositivi connessi dopo aver accettato.	Documento dei requisiti	Alta
SRF5	Camera, storage.	Il dispositivo selezionato inizia a registrare.	Il dispositivo registra un video riproducibile correttamente.	Documento dei requisiti	Alta
SRF6	Camera, NearbyConnections.	Il file video viene inviato al server.	Il file creato è stato ricevuto correttamente senza errori ed è riproducibile.	Documento dei requisiti	Alta
SRF7	Storage.	Viene memorizzato il percorso del file selezionato.	Il percorso deve essere valido.	Documento dei requisiti	Media

SRF8	FFmpeg.	Più file video vengono uniti.	Il video risultante è l'insieme dei video selezionati.	Documento dei requisiti	Media
------	---------	-------------------------------	--	-------------------------	-------

5. Altri Test

5.1. Sicurezza

Il test di sicurezza ha lo scopo di osservare la perdita di informazione possibile nel passaggio di file da un dispositivo ad un altro e la corretta avvenuta di connessione tra i dispositivi.

I test di sicurezza che saranno effettuati sull'applicazione sono i seguenti :

- completezza del video generato;
- impossibilità da parte di altri utenti non previsti e possibili malintenzionati di connettersi;
- privacy dell'utente rispettata.

5.2. Stress test & Test di volume(S&T)

Ha come obiettivo verificare che il sistema sopporti il carico massimo previsto in fase di progettazione. Il sistema viene testato oltre i limiti finché fallisce, questa operazione verrà effettuata da Mazzon Leonardo. Lo stress test verrà effettuato monitorando le risorse hardware utilizzate dall'applicazione, il tempo di esecuzione e aumentando i dispositivi connessi. Non ci si aspetta particolari sovraccarichi.

5.3. Test di connessione(CT)

Il test di connettività verrà eseguito da tutti i membri per avere maggior compatibilità tra diversi dispositivi. Si controllerà la corretta connessione di più dispositivi tramite nearby. L'affidabilità delle API Nearby sarà determinante per un risultato soddisfacente.

5.4. Ripristino di emergenza/Backup

Per ripristino di emergenza si intende l'insieme delle misure tecnologiche e logistico/organizzative atte a ripristinare sistemi, dati a fronte di gravi emergenze che ne intacchino la regolare attività.

La gestione dei dati è affidata agli operatori dei servizi di hosting/cloud come GitHub e Google. Non si prevedono perdite di dati significative.

5.5. Testing di unità

Il test di unità ha lo scopo di isolare una parte specifica del codice del programma e determinare il corretto funzionamento. Durante lo sviluppo, un modulo del programma può essere una singola funzione o procedura.

Il test di unità verrà effettuato da Nicola Giulio.

5.6. Test di Integrazione

Il test di integrazione determina effettivamente se il software può essere testato dal cliente, poiché la maggior parte degli errori è già stata rimossa dal software prima che un cliente avvii il test di accettazione.

Con il test di integrazione utilizzeremo un approccio Bottom-Up.

Il test di integrazione verrà effettuato da Piazza Alessandro.

6. Strategie dei Test

6.1. Responsabilità Livelli di Test

Livelli di test e priorità delle componenti: P(prioritaria), S(secondaria).

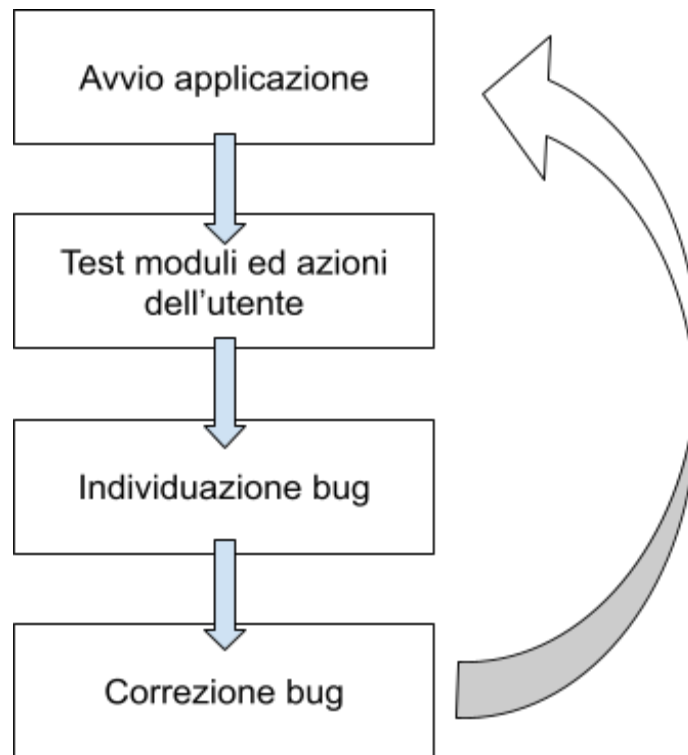
Livello di Test	Parti esterne	Team di sviluppo	Azienda
Unit Test		P	
Integrazione		P	
Sicurezza	P	S	
Connettività		P	
Collaudo		S	P
Produzione		S	P

6.2. Tipo di test e approccio

Tipo di test	Obiettivi
White-Box	Testare il comportamento dell'applicazione dato un certo input.
Back-to-Back	Analizzare i diversi output a seconda dell'implementazione.
Bottom-Up	Verificare il corretto funzionamento delle componenti.
Requisiti di progressione	<ul style="list-style-type: none">rispetto dei requisiti definiti nel documento dei requisiti;esecuzione delle funzioni in modo accurato;gestione corretta delle condizioni di errore;corretta funzione delle interfacce;corretto caricamento dei dati.
Requisiti di regressione	<ul style="list-style-type: none">le nuove funzionalità devono essere indipendenti dalle vecchie al fine di fornire maggiore semplicità in caso di correzione degli errori.

6.3. Build Strategy

La strategia adottata per ogni test è la seguente:



6.4. Pianificazione esecuzione dei test

Tutti i test effettuati (primari e complementari) saranno allegati in un altro documento in formato xlsx. Il template dei test è il seguente:

Codice requisito	Descrizione	Codice testato es. funzioni	Input	Risultato atteso	Risultato ottenuto	Numero di prove	Nome del tester	Firma del tester	Priorità

6.5. Servizi, dati e pianificazione della fornitura di risorse

6.5.1. Ambiente di test

I dettagli dei test effettuati e l'ambiente di test saranno allegati nel documento descritto nel punto precedente.

6.5.2. Accesso ad altre applicazioni

Accesso richiesto all'applicazione del file manager.

6.5.3. Requisiti dei tester

Ogni persona che funge da tester deve possedere i seguenti requisiti:

- applicazione installata sul device di test;
- possibile chiave di accesso all'applicazione;
- accesso ai documenti di test forniti dal team per raccogliere informazioni.

6.5.4. Requisiti dei dati

Prima si effettueranno dei test singoli sulle funzioni e parti di codice di interesse immettendo come input dati provvisori con lo scopo di garantire la correttezza del codice scritto, successivamente si eseguiranno dei test simulando dati reali al fine di scovare possibili errori generati.

6.5.5. Risorse & Conoscenze

Per poter effettuare i test da parte del team di sviluppo sarà necessario possedere :

- Conoscenza del linguaggio di programmazione Java in ambiente Android;
- Conoscenza base del file-system del dispositivo;
- Conoscenza del funzionamento e sviluppo dell'applicazione.

Per il personale di test si raccomanda possedere:

- nozioni base d'uso di un dispositivo smartphone;
- abilità visiva nella norma;
- abilità motoria delle mani e braccia funzionante.

6.6. Strumenti di test

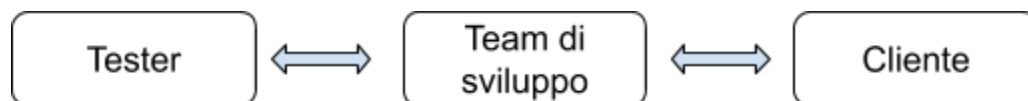
I seguenti strumenti saranno usati in supporto agli sviluppatori per gestire e amministrare il test.

Processo	Strumento
Creazione dei casi di test	Google Sheets, Google Docs
Monitoraggio dei test	Google Sheets, Google Docs
Esecuzione dei casi di test	Emulatore Android Studio, smartphone
Gestione dei casi di test	Google Sheets
Gestione dei difetti	Google Sheets, email per comunicare con i tester

6.7. Procedura di consegna dei test

Il team di sviluppo e il cliente si accorderanno, se necessario, per trovare persone disposte a collaborare come tester dell'applicazione per favorire maggiore trasparenza e soddisfare i bisogni tra cliente ed azienda.

La comunicazione avverrà tramite email per l'invio di documenti e risultati seguendo questo processo di comunicazione:



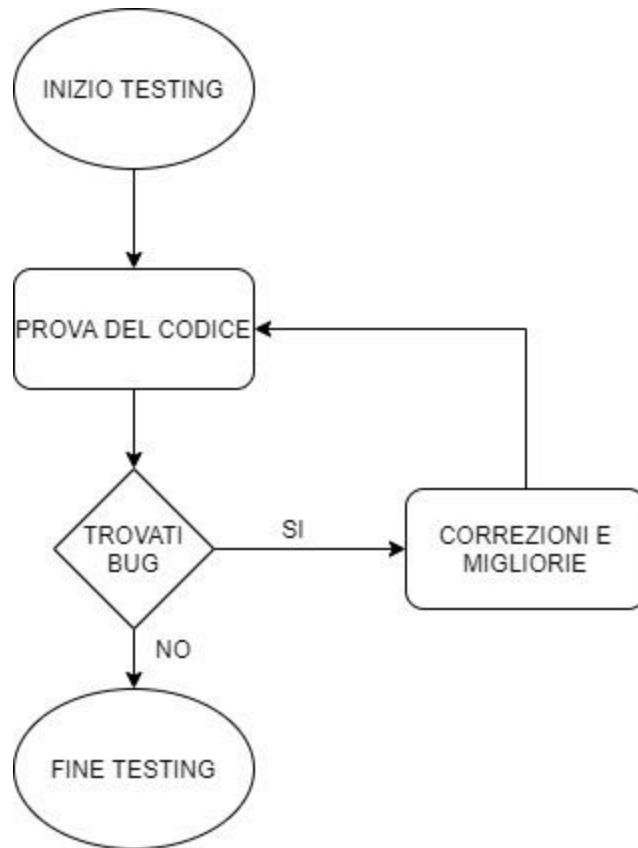
6.8. Metriche dei test

Dei test verrà monitorato l'utilizzo delle risorse, come specificato nel punto 5.2, in particolare CPU, memoria RAM, velocità della rete. L'applicazione non deve pesare sull'hardware o sistema al punto di rallentare lo smartphone. Utilizzando il "Profiler" di Android Studio sarà possibile monitorare tali risorse già nell'emulatore, altrimenti esistono applicazioni di terzi per monitorare il dispositivo fisico in fase di test.

A seguito dei risultati il codice verrà ottimizzato nelle parti critiche in soggetto.

7. Piano dell'ambiente dei test

7.1. Gestione dell'ambiente dei test



7.2. Dettagli dell'ambiente di test

7.2.1. Testers

Attualmente oltre al team di sviluppo formato da quattro persone, saranno selezionati soggetti esterni (senza competenze tecniche in possesso) al fine di testare l'applicazione in modo "User-Friendly" con lo scopo di rendere l'applicazione il più intuitiva possibile.

Requisiti di accesso al sistema: i tester devono aver accesso alle funzionalità richieste dall'app.

Requisiti hardware: smartphone dotato di fotocamera, modulo di rete e GPS.

7.2.2. Hardware e Firmware

I requisiti per il testing sono i seguenti:

- Dispositivi smartphone con Android 7.0+ dotato di fotocamera;
- Spazio di archiviazione sufficiente per il video registrato.

7.2.3. Software

- Android 7.0+, fornito dal produttore;
- Google Play Services, forniti dal produttore;
- Permessi per la rete(WiFi, bluetooth...), archiviazione, camera e microfono, forniti dall'utente durante la prima esecuzione.
- In caso i file trasferiti vengano corrotti o la velocità di rete sia bassa: Nearby Share. Software di file sharing disponibile da Android 6 tramite un aggiornamento fornito da Google.

7.2.4. Interfacce

Le interfacce alle applicazioni esterne sono :

- API Nearby-Connections(Google);
- fotocamera proprietaria del dispositivo in uso;
- file manager del dispositivo in uso.

7.2.5 Altri materiali

Non sono richiesti ulteriori materiali.

7.3. Ambiente da stabilire

Task	Requisiti	Responsabile	Start Date	End Date
Installazione dell'applicazione.	Due o più smartphone.	Mazzon Leonardo, Piazza Alessandro	26/11/2020	14/01/2021
Creazione template su fogli di calcolo.	Accesso ad internet.	Nicola Giulio	26/11/2020	13/01/2021

7.4. Controllo ambiente

- Git branches per separare codice di test dal codice principale;
- regole di protezione dei branch.

7.5. Ruoli nell'ambiente e responsabilità

Ruolo	Staff Member	Responsibilities
Release Manager	Leonardo Mazzon	Responsabile per la creazione, il coordinamento e il supporto dell'ambiente di test
Test Manager	Alessandro Piazza	Responsabile in consulenza con il Release Manager per la pianificazione, la stabilità e il funzionamento
Project Manager	Giulio Nicola	Coordina il team di sviluppo gestendo le varie fasi di lavoro e la comunicazione con il cliente
Assistant	Klevis Hasa	Assistente del team di sviluppo

8. Assunzioni

8.1. Assunzioni

Il team di sviluppo offrirà supporto al team di testing nel caso sia necessario chiarire dubbi o spiegare funzionalità.

8.2. Dipendenze

L'ambiente di test verrà configurato a priori per garantire un corretto funzionamento dell'applicazione.

9. Criteri di entrata ed uscita

I criteri d'entrata che utilizzeremo sono:

- L'ambiente di test deve supportare l'hardware e il software, le impostazioni e gli strumenti necessari per l'esecuzione del test;
- disponibilità delle risorse necessarie;
- definizione e approvazione dei requisiti in maniera chiara;
- necessità di avere un codice completo o parzialmente completo per poterlo testare.

I criteri di uscita che utilizzeremo sono:

- Nessun bug critico deve essere presente;
- correzione di tutti i difetti identificati;
- esecuzione di tutti i casi di test;
- rispetto delle scadenze.

10. Piano Amministrativo

10.1. Approvazioni

Indicazione delle persone che partecipano ai test e dei responsabili del team di sviluppo che coordinano le azioni.

Task	Responsabile	Approvazione
Usabilità interfaccia	<i>Firma: Leonardo Mazzon, Alessandro Piazza</i>	<i>Firma: Giulio Nicola</i> <i>Firma.....</i> <i>Firma.....</i>
Registrazione video	<i>Firma: Leonardo Mazzon</i>	<i>Firma: Alessandro Piazza</i> <i>Firma: Giulio Nicola</i> <i>Firma.....</i>
Ricezione file video	<i>Firma: Leonardo Mazzon</i>	<i>Firma: Alessandro Piazza</i> <i>Firma: Giulio Nicola</i> <i>Firma.....</i>
Unione file video	<i>Firma: Alessandro Piazza</i>	<i>Firma: Giulio Nicola</i> <i>Firma: Leonardo Mazzon</i> <i>Firma.....</i>

10.2. Programma di Test Milestones

Abbiamo individuato le seguenti milestones:

	Primo Prototipo		Secondo Prototipo		Release Finale	
Milestone	Data fine pianificata	Data fine reale	Data fine pianificata	Data fine reale	Data fine pianificata	Data fine reale
Creazione interfaccia utente	13/11/2020	13/11/2020	09/12/2020	09/12/2020	04/01/2021	10/01/2021
Connessione tra dispositivi: <ul style="list-style-type: none"> • server • client 	24/11/2020	23/11/2020	18/12/2020	18/12/2020	12/01/2021	12/01/2021
Registrazione individuale del singolo dispositivo client	24/11/2020	24/11/2020	18/12/2020	18/12/2020	12/01/2021	12/01/2021
Invio video al dispositivo server	non previsto	non previsto	18/12/2020	sospeso	12/01/2021	06/01/2021
Unione video	non previsto	non previsto	18/12/2020	21/12/2020	12/01/2021	12/01/2021
Test	25/11/2020	25/11/2020	21/12/2020	21/12/2020	13/01/2021	13/01/2021
Test finale	-	-	-	-	13/01/2021	13/01/2021
Consegna cliente	-	-	-	-	15/01/2021	15/01/2021

10.3. Training

Per garantire l'inizio dei test sono stati identificati i seguenti requisiti di formazione:

Requisito di formazione	Soggetto	Date
Sistema operativo Android	Gruppo	19/10/2020
Suite office	Gruppo	13/10/2020
Android Studio	Gruppo	10/11/2020

10.4. Gestione dei difetti

Ogni bug identificato verrà studiato per capirne la causa e successivamente risolverlo. Se necessario ci si confronterà con i tester.

11. Definizioni

Termini e acronimi che vengono usati all'interno di questo documento.

Termine/Acronimo	Definizione
White box testing	Il sistema è conosciuto dal tester così da permettere l'analisi del processo di elaborazione dell'output dato un certo input.
Black box testing	Il sistema è visto come una "scatola nera" dove è possibile analizzare solo input e output.
Incremental testing	Azione di test su tutto il sistema per ogni funzionalità/modulo aggiunti, con lo scopo di trovare errori imprevisti causati dalle dipendenze di funzioni presenti nel codice
Bottom-up testing	Testing sulle funzioni più interne e specifiche del codice creando delle interfacce opportune per agganciarsi alla parte di codice più ad alto livello(per alto livello si intende codice generale che descrive una funzionalità senza conoscerne i dettagli)
Milestones	Traguardi chiave nello sviluppo del progetto
User-Friendly	"Di facile utilizzo"
Bug	Guasto che porta al malfunzionamento del software.
Android	Sistema operativo sviluppato da Google per dispositivi mobili.
Server	Dispositivo in ascolto che fornisce dati o servizi verso altri dispositivi detti client
Client	Dispositivo in grado di scambiare dati con un server
Unit test	Test di una singola funzione/modulo con controllo dell'input e output

12. Riferimenti

I seguenti documenti sono stati consultati per la creazione di questo documento.

#	Nome Documento	Versione	Commenti
	Documento di piano di progetto	1.0	
	Documento di specifica dei requisiti del progetto	1.0	
	Template del piano di testing	-	Template fornito dal committente

13. Contatti

Le seguenti persone possono essere contattate in riferimento a questo documento

Contatto primario	
Nome	Nicola Giulio
Titolo/Organizzazione	Team Leader
Email	875297@stud.unive.it
Contatti secondari	
Nome	Mazzon Leonardo
Titolo/Organizzazione	Developer/Tester
Email	868445@stud.unive.it
Nome	Piazza Alessandro
Titolo/Organizzazione	Developer/Tester
Email	869386@stud.unive.it
Nome	Klevis Hasa
Titolo/Organizzazione	Developer/Tester
Email	869641@stud.unive.it