**F2F – Families-To-Families Sharing Care of Children  
MIUR FISR 2020**

**Progetto di Ingegneria del Software 2021/22  
Università Ca’ Foscari Venezia**

**Piano di Testing**

versione 1.0  
 **Black Buffalos**

14 novembre 2021

*Informazioni sul Documento*

| MUR FISR 2020 |  | Acronimo | | | F2F | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Coordinatore Progetto | Prof. Agostino Cortesi | | | | | | | |
| Deliverable | Piano di Testing | | | | | | | |
| Data di Consegna | 14/11/2021 | |  | | |  | |  |
| Team Leader | Giovanni Costa | | |  | | | 880892@stud.unive.it | |
| Team members | Massimo Cailotto 880873@stud.unive.it  Filippo Di Gennaro 882795@stud.unive.it Matteo Minardi 880895@stud.unive.it  Andrea Munarin 879607@stud.unive.it | | | | | | | |

*Document History*

| Version | Issue Date | Stage | Changes | Contributors |
| --- | --- | --- | --- | --- |
| 1.0 | 14/11/2021 | Draft | First version | All members of the team |
| 1.1 |  | Draft |  |  |
| 2.0 |  | Final |  |  |

**Indice**

[***1.***](#bookmark=id.6ga854x14p0y) ***Introduzione*** **4**

[***2.***](#bookmark=id.kjgcjugtlaoq) ***Processo di testing* 4**

[***3.***](#bookmark=id.l7qou7jplg30) ***Modalità di verifica dei requisiti* 4**

[***4.***](#bookmark=id.4wpxvmsse1bh) ***Elementi testati* 7**

[***5.***](#bookmark=id.riz1w2bxswra) ***Schedule del testing* 8**

[5.1 Tempo](#bookmark=id.lioplmgn7d13) 8

[5.2 Risorse allocate](#bookmark=id.ewf65fjl82hl) 8

[***6.***](#bookmark=id.hq63lm96lrfk) ***Procedure di registrazione dei test* 10**

[***7.***](#bookmark=id.l1y2ovkdqvwl) ***Requisiti* 11**

[7.1 Requisiti hardware](#bookmark=id.3yltkrrj73he) 11

[7.2 Requisiti software](#bookmark=id.509c52maf0dc) 11

[**8. Vincoli**](#bookmark=id.8gscpxqhgmwd) **11**

[**9. Riferimenti**](#bookmark=id.4pve6t53el1y) **11**

1. **Introduzione**

All’interno di questo documento si descrivono i vari test che si intende svolgere nello sviluppo di questo progetto. Nello specifico verranno testati diversi moduli, sottosistemi e componenti, per verificarne il loro corretto funzionamento, anche in merito all’aggiunta di funzionalità al progetto esistente sviluppato dal team di Families Share.

Verrà inoltre svolta la verifica dei requisiti e la validazione di conformità di tali requisiti rispetto all’idea originale.

1. **Processo di Testing**

Per lo svolgimento del processo di testing si utilizzerà maggiormente la strategia “Bottom-up”: nello specifico si inizia il testing dalle componenti più in basso e si continua salendo verso l’alto, realizzando dei “drivers” che simulano l’ambiente nel quale le componenti sono valutabili.

Questa metodologia ha i vantaggi di poter individuare rapidamente errori architetturali e il fatto di non dover richiedere di aver già completato la struttura del sistema prima di poter iniziare qualsiasi test. Inoltre, è appropriato per sistemi object-oriented.

Anche se con questa strategia è possibile non riuscire ad individuare errori di progettazione di alto livello se non molto avanti nel processo, il team ritiene che questa sia la più adatta alla realizzazione del processo di testing

I membri del team non impegnati direttamente nello sviluppo software in quella fase di programmazione si dedicheranno al testing approfondito delle funzionalità, anche se comunque durante la produzione del codice sorgente lo sviluppatore si occuperà della verifica di funzionamento basica.

Come descritto nel Piano di Progetto, il Backup Engineer si dovrà occupare della validazione del codice, quindi anche del suo corretto funzionamento

1. **Modalità di verifica dei requisiti**

Seguendo il documento dei requisiti, in questo capitolo viene spiegato come saranno verificati tutti quelli elencati (possono essere esaminati sia singolarmente che in gruppo - se legati allo stesso scopo)

Per requisiti funzionali vedi Documento dei requisiti, Sezione 5.

| **Modalità di verifica** | **RF inclusi** |
| --- | --- |
| Una volta entrati nell’applicazione si ricerca la sezione apposita per la creazione dell’utente, si inseriscono tutti i dati personali e si conferma la creazione.  Si controlla che la pagina relativa all’area personale sia presente e quindi che l’utente sia stato creato correttamente.  Durante l’inserimento dei dati è possibile inserire un figlio, effettuando implicitamente la creazione di un bambino all’interno dell’app. | RF01, RF02, RF03 |
| Si entra nella pagina dedicata al profilo utente per visualizzare le sue informazioni.  In particolare si possono osservare i suoi dati personali, tutte le attività a cui è iscritto ed i gruppi di cui fa parte. | RF04 |
| Nella sezione legata alla creazione di un gruppo, ne si aggiunge il nome per poi decidere se inserire dei partecipanti iniziali.  Successivamente si può creare un link di invito al gruppo e aggiungere o rimuovere componenti.  Si effettua un controllo sulla corretta visualizzazione del gruppo (con tutti i suoi partecipanti) e sul giusto funzionamento delle opzioni di aggiunta e rimozione. | RF05, RF06, RF13 |
| Nella sezione apposita, si crea un’attività e si inseriscono i partecipanti iniziali (facoltativo).  Successivamente, osservando l’attività dalla pagina sarà possibile:   * gestire i partecipanti e le altre informazioni * rimuovere l’attività * annullare l’attività   Si controlla che l’attività sia correttamente visualizzata e che funzionino le operazioni di rimozione/annullamento. | RF07, RF08, RF10, RF11, RF12, RF14 |
| Dalla sezione di visualizzazione di una qualsiasi attività, si seleziona la funzionalità di relativo contact tracing.  Si inseriscono i dati relativi ad una persona (contagiato/possibile contagiato) e si controlla la giusta rappresentazione delle persone che hanno avuto dei contatti con essa (i dati riportati risalgono fino ai 4 giorni precedenti). | RF09 |
| Dalla sezione di visualizzazione di una qualsiasi attività si imposta la “Richiesta green pass” per tutti i partecipanti.  L’applicazione ricorderà tramite un messaggio all’utente di presentare il green pass in presenza all’attività.  L’utente avrà inoltre la possibilità di caricare il proprio certificato nell’area personale: in questo caso viene controllato che, una volta caricato, il certificato venga visualizzato correttamente. | RF15 |
| Si accede nella pagina di gestione della bacheca e si carica un contenuto multimediale tramite l’apposita sezione. In seguito si potranno visualizzare tutti i contenuti pubblicati da tutti gli utenti partecipanti al gruppo relativo.  É possibile eliminare dei contenuti caricati precedentemente, controllando prima che non siano più presenti all’interno della bacheca (e quindi non visualizzabili da altri utenti). | RF16, RF17 |
| Si entra nella pagina relativa alla gestione del mercatino digitale.  Si condivide un oggetto, specificando il nome e una breve descrizione, e si controlla che l’oggetto sia stato caricato correttamente.  Bisogna testare anche la giusta rappresentazione dello stato del bene condiviso (se è già stato prenotato per uno scambio oppure se è disponibile).  Si prova inoltre ad eliminare un prodotto già caricato in precedenza e si controlla che tutte le informazioni correlate ad esso siano state eliminate (ad esempio se l’oggetto era stato prenotato per uno scambio). | RF18, RF19 |

Inoltre, si vuole specificare che si utilizzerà ampiamente la verifica statica, cioè il processo di verificare la corrispondenza tra un sistema software e la sua specifica senza eseguire il codice. Ha i vantaggi di scoprire anticipatamente gli errori durante il processo di codifica, è usualmente più cost-effective del “detect testing” a livello di moduli e permette di combinare la ricerca di errori con altri tipi di valutazioni di qualità.

1. **Elementi Testati**

Prima di iniziare con il testing dell’applicazione vera e propria, si verificherà che il dispositivo sul quale verrà installata soddisfi tutti i requisiti necessari per garantire il completo funzionamento in maniera corretta.  
 Alcuni elementi sui quali si porrà particolare attenzione saranno:

* Stato della batteria
* Quantità di memoria disponibile
* Funzionamento del sistema di localizzazione GPS
* Funzionamento del sistema di connessione ad Internet

Solo una volta terminate le verifiche di questi pre-requisiti fondamentali, si passerà all’analisi dell’applicazione.  
Si tiene presente che prima di aggiungere una nuova funzionalità, o iniziarne il testing, gli elementi già introdotti in versioni precedenti debbano funzionare correttamente.

Verrà eseguito il test per ogni singola caratteristica aggiunta.

I test delle macrocomponenti avverranno seguendo l’approccio bottom-up (vedi paragrafo 2), iniziando dunque con l’analisi delle loro proprietà specifiche e finendo con quelle più generali.   
  
L’ordine in cui saranno eseguite le operazioni non sarà casuale, infatti verrà data priorità alle funzionalità principali, qui elencate:

* Gestione e tracciamento emergenza covid
* Bacheca condivisione
* Mercatino digitale

Una volta assicurato il funzionamento efficace ed efficiente delle feature fondamentali, si passerà al controllo di quelle secondarie, ovvero:

* Chat dedicata
* Migliore integrazione luoghi
* Orari di disponibilità
* Dark mode

1. **Schedule del Testing**

* 1. **Tempo**

Il Testing delle funzionalità principali dovrà essere svolto più volte, in particolare è necessario svolgere tutte le varie verifiche entro le date in cui è stato definito il rilascio delle milestone, come indicato nel piano di progetto.

Le date in cui si stima di eseguire i test principali, in relazione alla milestone, sono:

* **14/12/2021** → Milestone 1: Rilascio beta (versione 1.0 codice sorgente): 15/12/2021;
* **28/11/2021** → Milestone 2: Versione completa: 29/11/2021;
* **13/01/2022** → Milestone 3: App finale (versione 1.1. del codice) e allineamento documentazione: 15/01/2022

In questa tabella inoltre, vengono riportati i tempi medi stimati per il testing delle varie componenti:

| **Tipo di test** | **Tempo** |
| --- | --- |
| Server | 2H |
| Operazioni CRUD | 2H |
| Backend | 1.5H |
| Frontend | 1H |

* 1. **Risorse allocate**

In questa sezione vengono elencati gli strumenti e risorse che sono state utilizzate per il testing dal team. Ogni risorsa è considerata indispensabile per il completo e corretto testing dell’app.

| **Risorsa** | **Scopo della risorsa** | **Specifiche** |
| --- | --- | --- |
| Smartphone | Utilizzato per testare come l'applicazione viene eseguita e per monitorare lo sviluppo direttamente sulla piattaforma | Numero di telefoni: 3  (con versioni di android differenti) |
| Emulatore Android | Utilizzato per testare e monitorare l'applicazione da una piattaforma che non sia un dispositivo smartphone | Android Studio Emulator |
| Computer | Utilizzato per lo sviluppo delle funzionalità dell'applicazione | Numero di computer: 5 |
| Connessione internet | Utilizzata per la condivisione del codice e comunicazione con il team | Di qualsiasi capacità |
| Team | Gli elementi delegati allo sviluppo | Membri del team: 5 |

1. **Procedure di Registrazione dei Test**

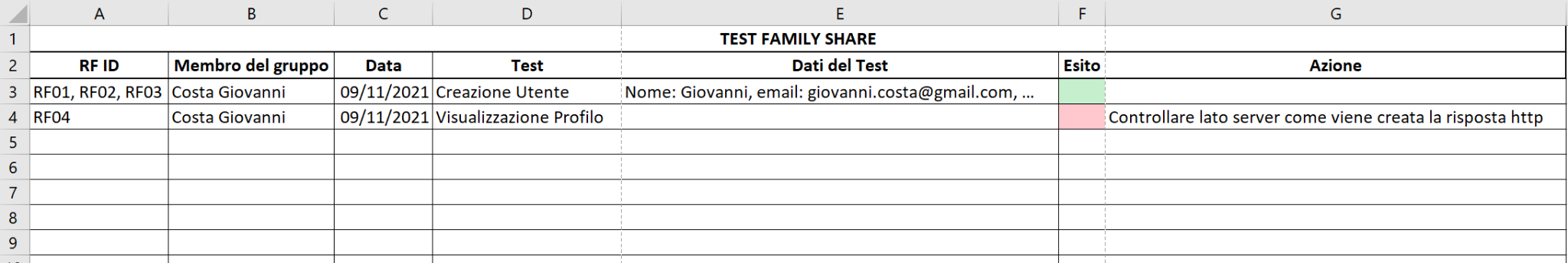
Di seguito è riportata la strategia adottata per la registrazione dei test.   
Ogni membro che effettua un test deve documentare l’esito ottenuto, in questo modo i risultati sono condivisi a tutti i membri del gruppo ed agli utenti che decidono di usufruire dell’applicazione.

Ogni dato e risultato sarà memorizzato all’interno di un foglio di calcolo elettronico (Excel), messo a disposizione all’interno della cartella Drive condivisa (vedi Piano di Progetto) per essere visualizzabile da tutti i membri.

Il formato con cui vengono salvati i test è costituito da una tabella nella quale in ogni riga è presente un test e nelle colonne sono descritti i seguenti campi:

* Id del requisito da verificare
* Membro del gruppo che ha effettuato il test
* Data dell’operazione di verifica
* Nome identificativo e univoco del test
* Dati del test (insieme di input inseriti per testare il sistema)
* Esito del test
* Eventuale azione da eseguire come conseguenza dell’esito del test ottenuto

È fornito un esempio della struttura per la registrazione dei dati:



1. **Requisiti**

* 1. **Requisiti hardware**

Di seguito sono elencati i dispositivi hardware e strumenti necessari utilizzati dal team per il testing dell’applicazione.

* Computer
* Smartphone
* Connessione ad Internet

* 1. **Requisiti software**

La lista delle componenti software e tecnologie necessarie per un completo e corretto testing dell’applicazione è la seguente:

* Sistema operativo PC > Windows 10
* Sistema operativo Android > 6.0
* Android Studio, per eseguire il debug dell’applicazione client
* NodeJS, per eseguire il back end server e ricevere feedback sulle connessioni
* MongoDB e MongoDB Compass, per visualizzare lo stato del database
* GitHub, per avere una traccia delle modifiche
* Visual Studio Code, per modificare la parte server scritta in javascript
* Postman, per testare le richieste e le risposte HTTP

1. **Vincoli**

Il progetto presenta alcuni vincoli principali, sia per il suo sviluppo che per il testing.

Nello specifico questo documento deve essere consegnato entro il 14 novembre per poi svolgere dei test sul codice sorgente e sulle varie features principali; queste verifiche devono essere effettuate entro 15 dicembre, data in cui va consegnato la prima versione del software funzionante.  
Entro il 15 gennaio invece va consegnata la versione del software finale

Viene fatto notare che è possibile testare le varie componenti solo dopo che le altre sono completate, come descritto nel Piano di Progetto.

Prima di introdurre nuovi test è necessario verificare che il codice e le funzionalità già presenti rimangano funzionanti.

1. **Riferimenti**

Per creare questo documento sono stati utilizzati come riferimento:

* Il Documento dei requisiti stilato precedentemente
* Il Piano di Progetto stilato precedentemente
* Materiale messo a disposizione dal professore