



Progetto di Ingegneria del Software 2022/23

Università Ca' Foscari Venezia

Piano di Testing
1.0

Team Angio

13/11/2022



Document Informations

Progetto Standard	MyLocalBooking	MLB
Deliverable	Piano di Testing	
Data di Consegna	13/11/2022	
Team Leader	Pietro Visconti	885448@stud.unive.it
Team members	Nicola Marizza Pietro Donega Mirco Mellara	887004@stud.unive.it 881909@stud.unive.it 882963@stud.unive.it

Document History

Version	Issue Date	Stage	Changes	Contributors
1.0	13/11/2022	Draft	Creazione documento	Mellara Mirco



Indice

Introduzione	4
Obiettivi	4
Glossario	5
Testing	6
Cosa testare	6
Come verranno effettuati i test	6
Risorse richieste	6
Contesto d'uso	7
Piattaforma testing	7
Prerequisiti generali	7
Tipologia testing scelta	8
Tipologie	8
Metodologia	9
Testing per specifica	10
Altri test (Stress dati, privacy, Sicurezza db)	13
Altri strumenti	14
Test sulle interazioni	15
Prerequisiti per funzionalità	16
Criteri di inizio e fine testing	18
Milestones	20
Raccolta dati Testing	20
Riferimenti	20



Introduzione

Questo documento racchiude al suo interno tutte le informazioni relative all'organizzazione del “**Testing**” del Software. Con testing si intende l'esecuzione di particolari pattern di interazione col sistema volti a controllare il funzionamento di esso in risposta a particolari input forniti dagli utenti; inoltre, grazie a questo procedimento, sarà possibile scovare **malfunzionamenti**, dovuti a **difetti** in fase di realizzazione del codice.

Si vuole comunque ricordare che il testing **non** è in alcun modo **esaustivo** e che non esiste un piano in grado di scovare ogni possibile difetto!

Obiettivi

Più che un singolo obiettivo, questo documento si pone più obiettivi: essi sono infatti, il definire correttamente ed esaustivamente...

1. Le tecniche di testing che si vogliono usare
2. Modalità di verifica dei singoli requisiti
3. I casi di test che verranno usati
4. Schedule del testing: tempo e risorse allocate
5. Procedure di registrazione dei test
6. Requisiti hardware e software per questa attività
7. Vincoli che condizionano il testing



Glossario

Test: Esecuzione del programma o di una sua componente, in particolari condizioni, volto a scovare dei malfunzionamenti.

Malfunzionamento: Incapacità del Software di comportarsi secondo le aspettative, in un certo istante e in determinate condizioni, spesso causato da difetti in fase di realizzazione

Difetto: Causa, a livello di codice che compone il Software, che causa la presenza di malfunzionamenti.

Bug: sinonimo di "Difetto"

Stress testing: Tipologia di testing, che si focalizza sul corretto funzionamento del Software in condizioni dove i dati da elaborare dal Software e le componenti a cui si appoggia (tipo il database), si presentano in quantità elevata in lassi di tempi ristretti.

Unità: La più piccola componente atomica che compone il software. Essa è indivisibile perché non risulta utile frammentarla ulteriormente, visto che la sua complessità è semplice.

Modulo: Gruppo di più unità che compongono il software, racchiuse sotto uno specifico criterio

Sistema: Software nella sua interezza + il dispositivo su cui esso viene eseguito, in un particolare ambiente



Testing

Cosa testare

Il testing verrà effettuato per il sistema e tutte le sue **componenti singole**.

Il seguito del documento illustrerà nel dettaglio le modalità per ogni singola unità, per un gruppo di esse (detto **Modulo**, vedi Glossario), o il sistema nella sua totalità ed interezza.

Come verranno effettuati i test

I test verranno effettuati inizialmente sui dispositivi virtuali che fornisce Android Studio.

Questi dispositivi, vengono “emulati” sulla macchina degli sviluppatori e non richiedono risorse particolari, se non un elevato consumo di memoria da parte della macchina stessa.

Mentre i test finali, sulla totalità del sistema, verranno anche effettuati su dei dispositivi fisici compatibili con i requisiti richiesti dall'applicazione e illustrati nel “Documento dei requisiti” nel dettaglio.

In particolare per i test relativi alla posizione geografica, sarà possibile utilizzare dei valori fittizi per il proprio dispositivo virtuale, grazie a dei tool forniti da Android Studio stesso.

I test relativi al database invece prevederanno anche casi rivolti alla gestione di molti dati in contemporanea (vedi Stress Testing) e alla sicurezza di esso e verranno effettuati tramite dispositivi virtuali.

Tutti i test saranno preparati ed eseguiti con la complicità di uno o più membri del team di sviluppo

Risorse richieste

La risorsa principale che viene richiesta è una discreta disponibilità di memoria e di potenza di calcolo per le macchine su cui s'intende effettuare il testing, altrimenti potrebbero sorgere complicazioni indesiderate e soprattutto fuori dal nostro controllo, che porterebbero ad un fallimento del piano di testing.



Contesto d'uso

Piattaforma testing

I test verranno effettuati utilizzando sia l'emulatore di Virtual Devices fornito da Android Studio, che dispositivi Smartphone compatibili, appartenenti ai membri del gruppo, su cui viene installata l'applicazione.

Tutte le categorie di testing verranno effettuate su ambo le piattaforme.

Brevemente esse sono:

- Testing sulle singole unità (o testing atomici)
- Testing sui moduli (o testing di interoperabilità)
- Testing sull'applicazione completa (o testing totale)

Ma verranno approfondite nella sezione apposita sottostante.

Prerequisiti generali

I seguenti prerequisiti sono vincolanti alla corretta esecuzione dei Test.

- Dispositivo con Sistema Operativo Android 8.0 (Oreo) o successivo:
L'applicazione è stata pensata per venire utilizzata solo da dispositivi con sistema operativo Android. In particolare la versione deve essere quella 8.0 (Oreo) o una qualsiasi release successiva..
- Il dispositivo deve essere connesso ad Internet:
L'applicazione non può collegarsi ed interagire con il database in assenza di connessione ad Internet.
- Connessione al database:
Senza la connessione al database non è possibile autenticare l'utente o fornire ad esso qualsiasi dato utile alla propria esperienza, motivo per cui è essenziale che esso sia funzionante e costantemente accessibile
- L'applicazione deve avviarsi correttamente ed è essere interagibile per l'utente:
Eventuali errori dell'installazione del pacchetto contenente l'applicazione risulterebbero fatali e renderebbero impossibile per l'utente utilizzare l'applicazione, quindi è di vitale importanza che venga programmato e testato correttamente



Tipologia testing scelta

Abbiamo optato per testare a diversi livelli, con uno stile **"Bottom-up"**.

Questo implica che cominceremo con i test sulle singole unità, per poi procedere a quelli modulari e solo alla fine quelli di sistema, assicurandoci che con la scalata della gerarchia, rimangano sempre funzionanti anche le componenti inferiori.

La scelta di questa tipologia tipicamente non fa risaltare gli errori all'inizio del processo di testing, ma è molto utile per stabilire una base per dei software orientati allo stile di programmazione ad oggetti, come nel nostro caso.

Tipologie

- Singola Unità: Comprendono i testing su funzionalità singole (come ad esempio un cambio password, o la visualizzazione di uno spazio), ossia piccole componenti atomiche che appartengono a moduli più grandi.
Costituiscono il più basso livello di testing.
- Modulo: Comprendono testing su componenti più grandi del Sistema, che si appoggiano e fanno uso di più funzionalità. Un esempio potrebbe essere il profilo dell'utente: Il Modulo profilo è composto dalle singole funzionalità di "Visualizzazione Profilo", "Cambio dati personali", "Cambio password". Per testare il cambio di una password, sarà quindi necessario che nel modulo interagiscano e si coordinino correttamente le singole unità che lo compongono.
- Multi-Modulo: Comprendono i test su più moduli, ma non nella loro totalità.
Questo livello è soprattutto utile per fornire un livello d'isolamento migliore, se dovessero verificarsi complicazioni nei test di sistema, per scovare quali sono i moduli che non cooperano come dovrebbero. Un esempio potrebbe essere l'accesso al profilo dell'utente che si trova in un'altra schermata. Questa funzionalità prevede la cooperazione dei moduli "Navigazione" e "Profilo" per esempio.
- Sistema: Comprendono test sulla totalità dei moduli, che devono cooperare tra loro per fornire a questo punto tutte le funzionalità possibili.
Per testare il cambio password ora è necessario che il modulo del profilo operi insieme al modulo relativo all'autenticazione, quello della navigazione, etc... Per permettere in ogni circostanza possibile che sia possibile effettuare il cambio password (ad esempio per utenti non loggati, deve essere possibile loggarsi, poi accedere alla schermata del profilo, poi cambiare la password)



Metodologia

Il test sulle singole unità viene effettuato fino a quando esse non superano il test di funzionamento unitario.

A questo punto l'unità entra in uno stato dove rimane "congelata" in attesa di poter eseguire testing di essa nel contesto modulare, che avverrà una volta che tutte le unità del modulo avranno superato il proprio test unitario.

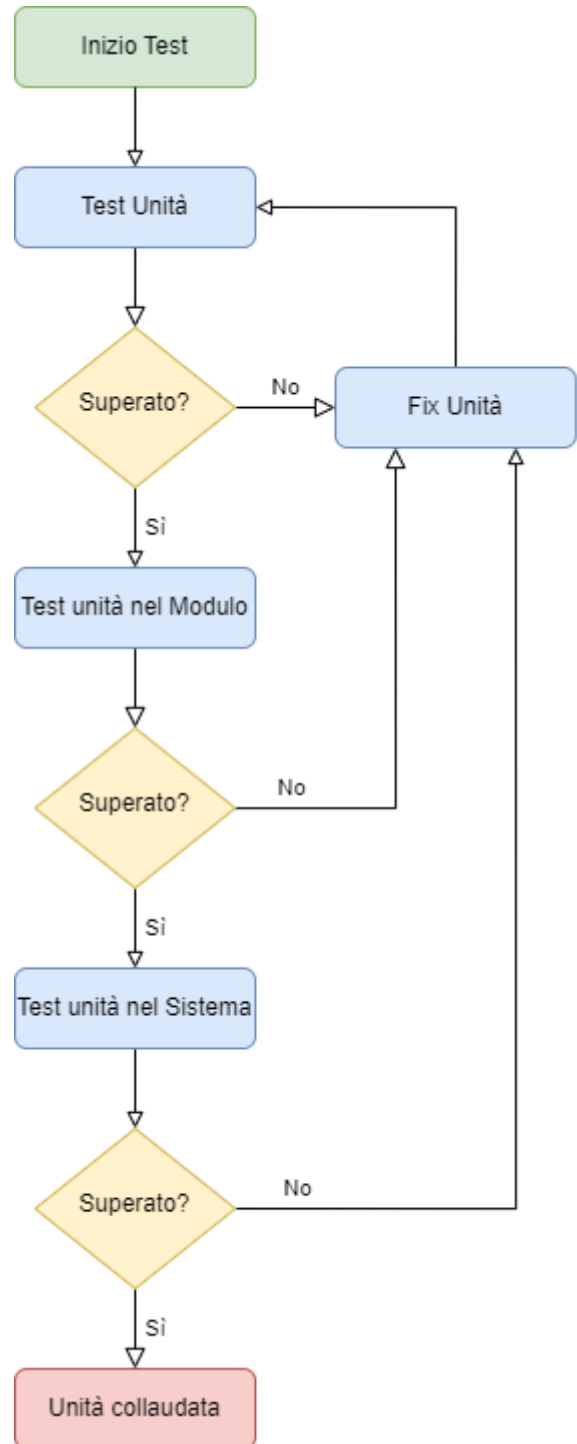
I test multi-modulo sono da considerarsi una variante dei test di modulo, ma siccome sono a testare l'interoperabilità tra vari moduli più il corretto funzionamento delle unità in esse, sono stati riportati in questo grafico.

Superato anche questo test, tornerà di nuovo essere "congelata" in attesa dei test sul sistema, che cominceranno una volta che tutti i moduli saranno stati implementati.

Se in qualsiasi test, dovessero presentarsi degli errori, l'unità dovrà venir sistemata e dovrà ricominciare con il testing al livello più basso, ossia quello unitario.

Un'unità che supera tutti i test viene definita come collaudata.

Idealmente, un'unità che supera il test a livello unità, non dovrebbe poi mai dover più modificata, ma è comunque necessario testarla contesti di livello superiore.



il

volti
che
non

ad

di

nei



Testing per specifica

In questa sezione, per ogni specifica riportata nel Documento dei requisiti, viene fornito un obiettivo del test, con relativo criterio di correttezza e priorità. Inoltre viene specificato di che tipologia di test si tratta.

Funzionalità	Specifica	Obiettivo	Criterio correttezza	Tipologia	Priorità
Login	S-01	Verificare che l'utente possa effettuare il login	Se i dati inseriti corrispondono a quelli di un utente esistente, effettua l'autenticazione, altrimenti solleva l'errore opportuno.	Unità	Alta
Logout	S-02	Verificare che l'utente possa effettuare il logout	L'utente termina correttamente la propria sessione	Unità	Media
Registrazione	S-03	Verificare che l'utente possa creare un account	La creazione del profilo va a buon fine e viene salvato correttamente tutto all'interno del database. Altrimenti vengono sollevati gli errori corrispondenti corretti.	Unità	Alta
Visualizzazione profilo	S-04	Verificare che l'utente possa visualizzare il proprio profilo	L'utente deve poter vedere tutti i propri dati, che devono essere corretti.	Unità	Alta
Segnalazione	S-05	Verificare che l'utente invii un feedback	La procedura, se rispetta i vincoli relativi al contenuto del form di feedback, deve salvare correttamente nel database il feedback	Unità	Bassa
Cambio password	S-06	Verificare che l'utente possa cambiare la propria password	Se la nuova password rispetta i vincoli definiti a livello database, allora deve aggiornarla correttamente a livello database	Unità	Bassa
Cambio dati	S-07	Verificare che l'utente possa cambiare i propri dati anagrafici	Se i nuovi dati rispettano i vincoli definiti a livello database, allora deve aggiornarli correttamente a livello database, inoltre deve poter accedere al cambio	Unità	Bassa



			password		
Lasciare una recensione	S-08	Verifica che l'utente possa lasciare una recensione	La procedura di recensione deve funzionare ed essa deve contare nella media delle recensioni presso tale spazio.	Unità	Bassa
Visualizzazione spazi	S-09	Verificare che l'utente veda gli spazi effettivi entro 2 km della posizione che intende considerare	L'app deve essere in grado di fornire la corretta visualizzazione totale degli spazi intorno alla posizione richiesta, che può essere quella predefinita o quella attuale dell'utente.	Unità	Alta
Modifica posizione predefinita	S-10	Verificare che la posizione predefinita possa venir modificata	L'utente modifica correttamente la propria posizione predefinita.	Unità	Alta
Effettuare prenotazione	S-11	Verificare che l'utente possa prenotarsi ad uno spazio	La prenotazione va a buon fine e viene salvata, abbassando di uno anche la disponibilità presso tale slot. Oppure se dovessero esserci particolari eccezioni, le deve trattare nel modo corretto	Modulo / Multi - Modulo	Alta
Visualizzazione prenotazioni	S-12	Verificare che l'utente possa visualizzare le proprie prenotazioni	L'utente vede correttamente le proprie prenotazioni future e quelle passate entro 30 giorni.	Unità	Alta
Eliminare prenotazione	S-13	Verificare che la cancellazione di una prenotazione vada a buon fine, se può essere eliminata	La prenotazione viene cancellata correttamente e viene alzato di uno la disponibilità dello slot relativo	Modulo	Media
Eliminazione automatica prenotazione	S-13	Verificare che le prenotazioni passate, di almeno 30 giorni, vengano eliminate da sole	Se tali prenotazioni vengono rimosse dal database e l'utente non le visualizza più, allora è corretto	Modulo	Media
Informazioni spazio	S-14	Verificare che le informazioni vengano mostrate all'utente	Le informazioni mostrate devono essere corrette e aggiornate, vi devono inoltre essere degli elementi che permettano la registrazione	Unità	Alta
Aggiungere uno spazio	S-16	Verificare l'aggiunta di uno spazio	Il provider deve essere in grado di aggiungere uno spazio, con le relativi	Unità	Alta



			informazioni al riguardo. Tale spazio verrà mostrato ai clienti		
Informazioni sullo spazio proprio	S-17	Verificare le informazioni relative ad uno spazio	Il provider visualizza correttamente tutte le informazioni relative al proprio spazio, comprese le prenotazioni degli utenti	Unità	Alta
Lista spazi messi a disposizione	S-18	Verificare quali spazi sono stati messi a disposizione	Il provider visualizza correttamente tutti gli spazi che ha messo a disposizione	Modulo	Alta
Eliminazione di uno spazio	S-19	Verificare la cancellazione di uno spazio	Il provider elimina dal database uno spazio e tutte le informazioni ad esso relativo. Inoltre tutte le prenotazioni relative, effettuate dai clienti, dovranno venir cancellate	Multi-Modulo	Media
Blacklisting	S-20	Verificare che un utente non abbia più accesso agli spazi di un certo provider	Il provider aggiunge alla propria blacklist un utente. Tale utente non potrà più effettuare prenotazioni presso spazi del provider e tutte le sue prenotazioni già esistenti verranno cancellate	Sistema	Bassa
Un-Blacklisting	S-21	Verificare che un utente abbia di nuovo accesso agli spazi di un certo provider	Il provider rimuove dalla propria blacklist un utente, che potrà di nuovo effettuare prenotazioni, anche se quelle esistenti al momento del blacklisting non verranno ripristinate	Sistema	Bassa
Strike	S-22	Verificare che un utente abbia ricevuto uno strike	Il provider aggiunge uno strike all'utente selezionato, per i propri spazi. Al raggiungimento di una quota fissata da esso, il cliente dovrà venir automaticamente black-listato.	Sistema	Bassa

NOTA: I test di unità, a meno che non venga specificato ulteriormente, in realtà prevedono la collaborazione del modulo “Database” + una singola funzionalità. Questi test verranno trattati come test di unità, ma dipendono dal corretto funzionamento e testing del modulo “Database”.



Altri test (Stress dati, privacy, Sicurezza db)

Privacy

Funzionalità	Specifica	Obiettivo	Criterio correttezza	Tipologia	Priorità
Privacy	S-13, S-19	Garantire che certi dati vengano cancellati dal database entro un certo ammontare di tempo o su richiesta degli utenti	I dati vengono eliminati correttamente dal database e non sono in alcun modo reperibili	Multi - Modulo	Alta

Stress test

Funzionalità	Specifica	Obiettivo	Criterio correttezza	Tipologia	Priorità
Stress Database	///	Il database deve riuscire ad effettuare le operazioni CRUD basilari anche in caso di più richieste simultanee, gestendo correttamente le procedure	Il database riesce a gestire un numero di richieste da reputarsi soddisfacente, in base alle esigenze.	Sistema	Alta
Stress Dispositivo	///	Il dispositivo è in grado di elaborare diverse azioni correttamente	Il dispositivo, ricevute in input diverse e molte azioni (quali tap dello schermo, swipe, attivazione/disattivazione della posizione, ecc...), deve riuscire ad elaborarle correttamente senza far crashare l'applicazione	Sistema	Bassa

Sicurezza DB

Funzionalità	Specifica	Obiettivo	Criterio correttezza	Tipologia	Priorità
SQL-Injection	///	Il database resiste ad attacchi di tipo SQL-Injection	Tutti i dati ricevuti in input sono sanificati/controllati e non causano SQL-Injection in alcun modo	Sistema	Alta
Vincoli	///	Tutti i dati nel database rispettano vincoli predefiniti	I trigger e vincoli relativi al database funzionano correttamente nella loro totalità	Sistema	Alta



Accessibilità	///	Il database deve risultare accessibile in ogni momento	Il database risulta accessibile in qualsiasi momento	Unità	Alta
---------------	-----	--	--	-------	------

Funzionalità del dispositivo

Funzionalità	Specifica	Obiettivo	Criterio correttezza	Tipologia	Priorità
Connessione ad Internet	///	Verifica che il dispositivo sia dotato di una connessione ad Internet	Il dispositivo è connesso ad una rete Internet ed è in grado di comunicare con il server e il database.	Unità	Alta
Geolocalizzazione	///	Verificare che il dispositivo possieda un modo per venir geolocalizzato	La posizione del dispositivo è nota all'applicazione, che può opportunamente comunicarla al server	Unità	Media

Altri strumenti

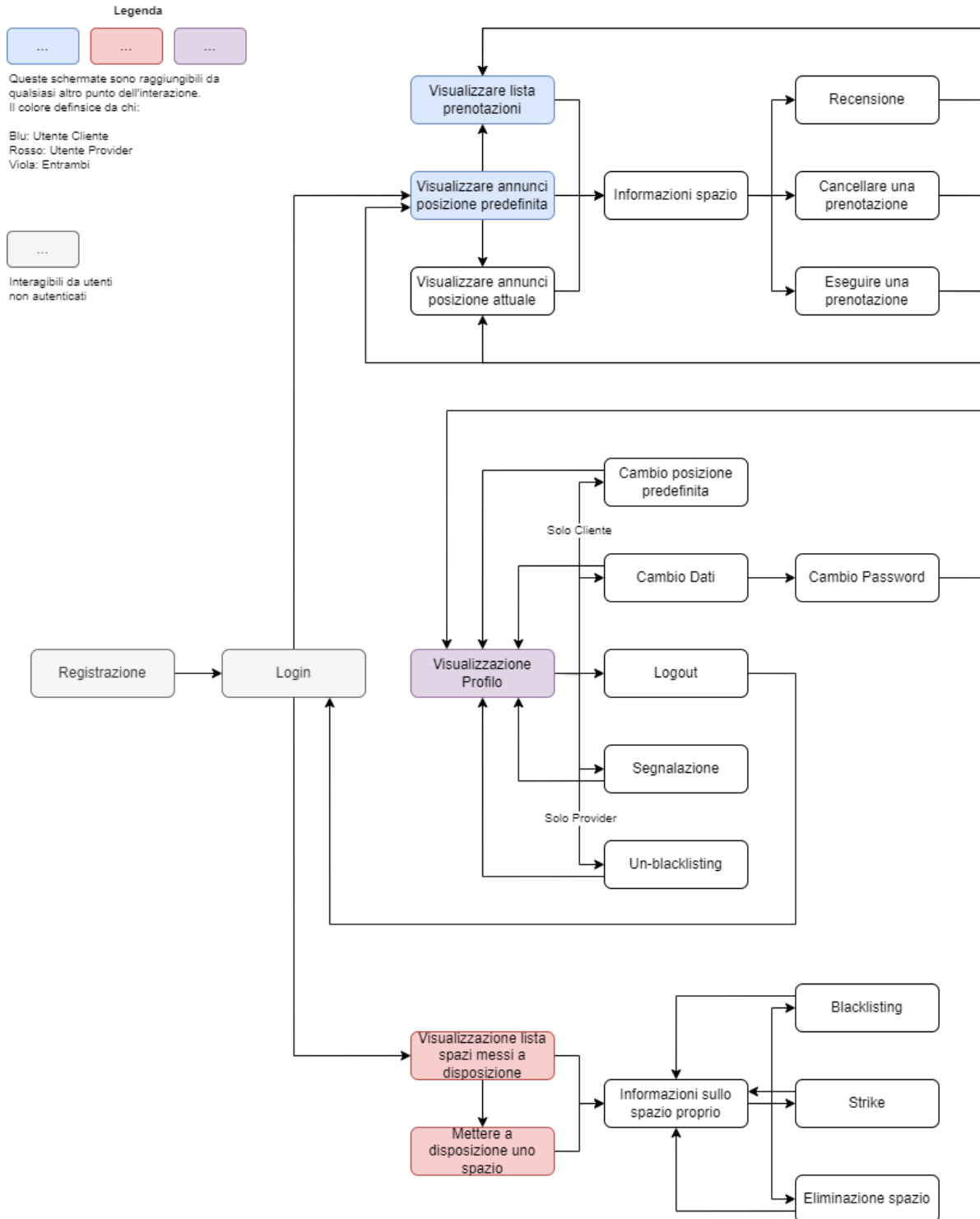
Raccolta risultati test: **Google Sheets**

Informazioni sui test: **Questo documento**

Altre informazioni utili: **Documento per la raccolta dei requisiti**



Test sulle interazioni





Prerequisiti per funzionalità

Prerequisiti generali:

Tutte le funzionalità elencate dipendono da un corretto funzionamento del database, che sia costantemente accessibile.

Con “L'utente deve essere loggato” si intende l'utente di tipo “Cliente”, mentre per quello “Provider” viene specificato.

Funzionalità	Prerequisiti
Login	L'applicazione deve essere stata installata correttamente e deve aprirsi, l'utente deve essersi registrato
Logout	L'utente deve essere loggato, deve essere nel proprio profilo
Registrazione	L'applicazione deve essere stata installata correttamente e deve aprirsi
Visualizzazione profilo	L'utente deve essere loggato
Segnalazione	L'utente deve essere loggato, deve essere nel proprio profilo
Cambio password	L'utente deve essere loggato, deve essere nel proprio profilo
Cambio dati	L'utente deve essere loggato, deve essere nel proprio profilo
Lasciare una recensione	L'utente deve essere loggato, deve poter visualizzare le proprie prenotazioni, deve aver effettuato almeno una prenotazione, la cui data dell'evento è passata a quella odierna
Visualizzazione spazi	L'utente deve essere loggato
Modifica posizione predefinita	L'utente deve essere loggato, deve essere nel proprio profilo
Effettuare prenotazione	L'utente deve essere loggato, deve poter visualizzare le informazioni di uno spazio (quindi deve poter visualizzare gli spazi)
Visualizzazione prenotazioni	L'utente deve essere loggato
Eliminare prenotazione	L'utente deve essere loggato, deve poter visualizzare le proprie prenotazioni



Eliminazione automatica prenotazione	Il sistema deve riconoscere quando da una prenotazione sono trascorsi più di 30 giorni.
Informazioni spazio	L'utente deve essere loggato, deve poter visualizzare gli spazi
Aggiungere uno spazio	L'utente deve essere loggato e di tipo provider
Informazioni sullo spazio proprio	L'utente deve essere loggato e di tipo provider, deve aver aggiunto uno spazio, deve poter vedere la lista di spazi messi a disposizione
Lista spazi messi a disposizione	L'utente deve essere loggato e di tipo provider
Eliminazione di uno spazio	L'utente deve essere loggato e di tipo provider, deve poter vedere la lista di spazi messi a disposizione
Blacklisting	L'utente deve essere loggato e di tipo provider, deve aver aggiunto uno spazio, deve poter vedere le informazioni sullo spazio proprio (quindi anche deve poter vedere la lista di spazi messi a disposizione)
Un-blacklisting	L'utente deve essere loggato e di tipo provider, deve aver "Blacklistato" un utente e deve essere nel proprio profilo
Strike	L'utente deve essere loggato e di tipo provider, deve aver aggiunto uno spazio, deve poter vedere le informazioni sullo spazio proprio (quindi anche deve poter vedere la lista di spazi messi a disposizione)



Criteri di inizio e fine testing

In questa tabella vengono invece definiti quando un test può iniziare e quando risulta essere terminato.

Per un maggior livello di dettaglio relativo alla fine del testing, in particolare ai criteri di correttezza, consultare la parte relativa al ["Testing di specifica"](#).

Test	Inizio	Fine
Login	Dopo "Registrazione"	Utente viene autenticato
Logout	Dopo "Login"	Utente termina la propria sessione correttamente
Registrazione	///	Viene registrato nel db l'utente
Visualizzazione profilo	Dopo "Login"	Lettura dati dal db e visualizzazione di essi nella schermata utente vanno a buon fine
Segnalazione	Dopo "Visualizzazione profilo"	Viene salvato nel db il messaggio
Cambio password	Dopo "Cambio dati"	Password aggiornata correttamente nel db
Cambio dati	Dopo "Visualizzazione profilo"	Dati aggiornati correttamente nel db
Lasciare una recensione	Dopo "Visualizzazione profilo". Preferibilmente anche dopo "Eseguire una prenotazione"	Recensione salvata correttamente nel db
Visualizzazione spazi	Dopo "Login"	Spazi mostrati correttamente
Modifica posizione predefinita	Dopo "Visualizzazione profilo"	Posizione aggiornata correttamente nel db.
Effettuare prenotazione	Dopo "Visualizzazione informazioni spazio"	Viene creata la prenotazione nel database.
Visualizzazione prenotazioni	Dopo "Login"	Le prenotazioni vengono lette dal db e mostrate all'utente
Eliminare	Dopo "Visualizzazione spazi"	La prenotazione viene eliminata e la



prenotazione		disponibilità viene aggiornata.
Eliminazione automatica prenotazione	Dopo “Eseguire una prenotazione” e “Visualizzazione prenotazioni”	La prenotazione viene eliminata automaticamente dal database
Informazioni spazio	Dopo “Visualizzazione spazi”	Le informazioni dello spazio vengono lette dal db e mostrate correttamente
Aggiungere uno spazio	Dopo “Login”	Lo spazio proprio, con relative informazioni, viene aggiunto al db
Informazioni sullo spazio proprio	Dopo “Lista spazi messi a disposizione”. Preferibilmente dopo “Aggiungere uno spazio”	Le informazioni dello spazio vengono lette dal db e mostrate correttamente
Lista spazi messi a disposizione	Dopo “Login”	La lista degli spazi messi a disposizione viene letta dal db e mostrata correttamente
Eliminazione di uno spazio	Dopo “Informazioni sullo spazio proprio”. Preferibilmente dopo “Eseguire una prenotazione”	Lo spazio viene rimosso dal db e tutte le prenotazioni ad esso associate
Blacklisting	Dopo “Informazioni sullo spazio proprio” ed “Eseguire una prenotazione”	L’utente blacklistato non può più prenotarsi per spazi del Provider e tutte le sue prenotazioni esistenti, sempre spazi del Provider, vengono cancellate
Un-blacklisting	Dopo “Blacklisting” e “Visualizzazione profilo”	L’utente può di nuovo prenotarsi per spazi del Provider.
Strike	Dopo “Informazioni sullo spazio proprio” ed “Eseguire una prenotazione”	L’utente riceve uno strike per quel Provider, che viene salvato nel database. Se l’aggiunta comportasse il raggiungimento del tetto massimo, l’utente dovrà venir “blacklistato”



Milestones

Qui vengono riportate le date in cui il testing si ritiene debba essere concluso, affiancato alla data effettiva in cui ciò avverrà, che verrà riempita nelle versioni successive di questo documento

Versione	Data pianificata	Data effettiva
Versione 1.0	14/12/2022	///
Versione finale	27/12/2022	///

Raccolta dati Testing

In questa sezione verranno mostrati alcuni risultati o particolari test che avremo effettuato, con un'apposita discussione ove lo ritenessimo necessario. Inoltre verrà mostrato un breve grafico o tabella contenente alcuni dati particolari relativi ai test.

Ovviamente verrà completata con lo svolgersi dei test e al momento risulta essere vuota.

La raccolta dei dati avverrà manualmente in un foglio "Google Sheets", dove ogni membro del gruppo che effettua un test, riporterà i dati rilevanti seguendo il template della tabella sottostante.

TC	Pre-condizione	Input	Output attesti	Output riscontrato	Post-condi zione	Esito	Priorità

Riferimenti

- ❖ Slide fornite dal docente a lezione
- ❖ Piano di Testing d'esempio sviluppato da colleghi degli anni precedenti