



Università
Ca'Foscari
Venezia



Corso:

Ingegneria del Software [CT0090]
2020/2021

Membri:

Melania Gottardo	874240
Mario Coci	876422
Monan Nasir	870374
Simone Checco	869408

Team:

Junipero

PIANO DI TESTING

VERSIONE 1.1

INDICE

1. INTRODUZIONE

1.1. Obiettivo

1.2. Ambito

2. GLOSSARIO

3. RIEPILOGO DEI TEST

3.1. In ambito di test

3.2. Fuori dall'ambito di test

4. ANALISI DELLE AREE DI TEST E DEL CAMPO DI APPLICAZIONE

4.1. Rilascio del prodotto

4.2. Test di regressione

4.3. Piattaforma di testing

5. OBIETTIVI DEL TEST DI PROGRESSIONE

6. ALTRI TEST

6.1. Privacy

6.2. Connettività

6.3. Stress e volume

6.4. Ripristino di emergenza / Backup

6.5. Test sulle unità

6.6. Test di integrazione

7. TEST SCHEDULING

- 7.1. Livelli di responsabilità
- 7.2. Approccio e tipi di test
- 7.3. Costruire una strategia
- 7.4. Pianificazione dell'esecuzione dei test
- 7.5. Struttura, dati e piano di fornitura delle risorse
- 7.6. Strumenti di test
- 7.7. Verifica della procedura di testing
- 7.8. Metriche

8. PIANO DELL'AMBIENTE DI TEST

- 8.1. Test sulle interazioni
- 8.2. Dettagli sull'ambiente di test
- 8.3. Scelta dell'ambiente
- 8.4. Controllo dell'ambiente
- 8.5. Ruoli e responsabilità sull'ambiente

9. PRESUPPOSTI E DIPENDENZE

- 9.1. Presupposti
- 9.2. Dipendenze

10. CRITERI DI INIZIO E FINE TEST

11. PIANO AMMINISTRATIVO

- 11.1. Approvazioni
- 11.2. Test milestones
- 11.3. Gestione dei difetti

12. RIFERIMENTI

1. INTRODUZIONE

1.1. OBIETTIVO

Questo documento servirà per racchiudere tutte le modalità di verifica per testare il funzionamento del sistema software in sviluppo.

Le modalità di verifica serviranno al team per individuare malfunzionamenti al fine di rendere tutto il sistema in grado di soddisfare i punti del documento dei requisiti.

Lo scopo di questo documento è di definire :

- l'ambito del test, le aree di interesse e gli obiettivi;
- la responsabilità del test;
- la strategia dei test per i livelli e i tipi di test effettuati per questa versione;
- i criteri di ingresso e uscita;
- le basi delle stime dei test;
- eventuali rischi, problemi, ipotesi e dipendenze dei test;
- il programma dei test e le milestone;
- i risultati del test.

1.2. AMBITO

Questo documento specifica i test che saranno eseguiti dal team. Il documento definisce i requisiti generali di testing e fornisce una vista integrata delle attività da completare nel processo di testing.

Il suo scopo è quello di definire:

- cosa sarà testato (sezione 5 e 6);
 - come verrà effettuato il test (sezione 7);
 - che risorse sono richieste, e quando (sezione 7).
-

2. GLOSSARIO

- **Activity:** un'activity rappresenta una schermata con l'interfaccia utente di un'applicazione ed il suo scopo è quello di permettere l'interazione con l'utente;
- **Ambiente di testing:** luogo fisico dove i test saranno eseguiti. Essendo l'applicazione del gioco "ruba-bandiera" basata su ampi spazi, l'ambiente di testing deve essere un luogo non al chiuso e molto grande;
- **Android:** sistema operativo basato su kernel Linux per dispositivi mobili, quali smartphone e tablet, nel nostro caso verranno considerate solamente versioni di Android superiori alla 6.0 (Marshmallow);
- **Android Studio:** Android è l'ambiente di sviluppo integrato dove si svilupperà la nostra applicazione;
- **API:** è l'acronimo di Application Programming Interface, serve ad indicare un gruppo di procedure già disponibili al programmatore per la realizzazione di un determinato compito;
- **App:** abbreviazione della parola "applicazione" per indicare un software per dispositivi mobile dotato di particolari procedure per raggiungere un determinato obiettivo;
- **Backup:** con backup si intende la messa in sicurezza dei dati di un sistema informatico attraverso la creazione di ridondanza delle informazioni stesse da utilizzare in caso di guasto per fare il ripristino dei dati;
- **Bug:** il bug è un problema che porta ad un malfunzionamento del software, producendo ad esempio un risultato inatteso o errato, tipicamente dovuto ad un errore di scrittura del codice. Un codice con un numero ragionevolmente alto di bug è detto "buggato";
- **Database:** rappresenta la locazione in cui vengono memorizzati tutti i dati riguardanti le partite nel database, i dati verranno strutturati in modo tale da facilitare l'inserimento e l'aggiornamento degli stessi in modo dinamico;
- **Firebase:** è un database NoSQL sviluppato da Google che permette di avere un database in tempo reale per la gestione dei dati, inoltre è una piattaforma per la creazione di applicativi mobile e web;
- **GitHub:** GitHub è un servizio di hosting di repository Git, ma offre molte più funzionalità. GitHub fornisce un'interfaccia grafica basata sul web, il controllo dell'accesso e diverse funzionalità di collaborazione, come strumenti di base per la gestione delle attività per ogni progetto;
- **GPS:** si intende il sistema di geolocalizzazione all'interno del dispositivo che il giocatore utilizza per giocare;
- **Host:** per host si intende il giocatore che sta ospitando una partita;
- **Java:** linguaggio di programmazione orientato agli oggetti, progettato per essere più indipendente dalla piattaforma di esecuzione;

- **Log:** file che contengono messaggi relativi al sistema;
 - **Peer-to-peer:** per sistema peer-to-peer si intende un modello di architettura logica di rete informatica in cui i nodi non sono gerarchizzati unicamente come client o server, ma ognuno può offrire servizi verso gli altri host della rete;
 - **Smartphone:** dispositivo dotato di tecnologie hardware avanzate rispetto ad un comune cellulare, quali uno schermo touch, una memoria RAM e spazio di archiviazione minimi per installare svariate app, un processore di discreta potenza per far girare il sistema operativo e le app che verranno successivamente installate, un sistema di sensoristica avanzato (gps, giroscopio, accelerometro, ecc.) ed una connessione ad internet per permettere la comunicazione con internet;
 - **Unit testing:** chiamato anche “test sulle unità”, con unit testing si intende l’attività di testing di singole unità software;
-

3. RIEPILOGO DEI TEST

3.1. IN AMBITO DI TEST

L’ambiente di testing, per quanto riguarda la nostra applicazione, può estendersi a tutto il campus scientifico o ad altre aree molto estese. In questo ambiente la nostra applicazione dovrà essere in grado di poter creare una partita, creare le squadre ed assegnare i ruoli, controllare la posizione dei partecipanti, capire quando una partita termina con un vincitore o termina per mancanza di requisiti (esempio: manca di un host), capire chi ha vinto ed aggiornare le statistiche per ogni partecipante.

3.2. FUORI DALL’AMBITO DI TEST

Le caratteristiche fuori ambito dell’ applicazione riguardano soprattutto la velocità con cui verrà aggiornata la posizione dei giocatori. Il sistema dovrà aggiornarsi in modo sempre più rapido man mano che il cacciatore si avvicina alla bandiera, in modo tale da prendere decisioni sempre più in tempo reale. Un altro requisito fuori ambito riguarda il controllo della partita, il sistema, deve essere in grado di capire se i requisiti persistono per tutta la partita, oppure in mancanza di uno di questi deve terminare la partita in corso.

4. ANALISI DELLE AREE DI TEST E DEL CAMPO DI APPLICAZIONE

4.1. RILASCIO DEL PRODOTTO

L'applicazione verrà sviluppata nell'ambiente di sviluppo Android Studio utilizzando le librerie principali di Java (JDK 8) e di Android (SDK) compatibili con le versioni di Android 6.0 e successive, inoltre per garantire una comunicazione affidabile con un database esterno si sfruttano le API di tale database (Firebase).

Tale applicazione pertanto sarà un software in grado di operare all'interno di uno smartphone Android dotato di connessione a internet e opportuna sensoristica (GPS, WI-FI, Bluetooth, sensore radio) per permettere la lettura e scrittura di dati in un database, creando così un sistema peer-to-peer durante l'utilizzo nelle funzioni di principali dell'applicazione (Vedi Documento di specifica dei requisiti, sezione 3).

A tal proposito è necessario che l'applicazione sia in grado di:

- **connettersi al database** esterno e poterne leggere i contenuti che essa sono necessari al fine di risolvere determinate attività, di poter aggiornare i dati nel database, in particolare i dati che faranno riferimento ad uno specifico utente (il possessore dello smartphone, in quanto ogni smartphone avrà un ID univoco nel database);
- **ottenere i dati sulla propria posizione** attraverso il GPS in coordinate (latitudine e longitudine) e di poter usufruire di tali dati per eseguire delle operazioni su di essi, inoltre sarà richiesto che lo smartphone riesca ad aggiornare i dati sulla sua posizione ogni tot di tempo;
- **giocare ad una partita**, simulata grazie all'uso di più smartphone che cooperano all'interno di un database.

4.2. TEST DI REGRESSIONE

L'applicazione sarà composta da piccoli componenti modulari, utilizzati solamente quando necessario per assolvere determinati compiti. Durante lo sviluppo di tali componenti è necessario avvenga uno unit testing ogni qualvolta ne si termini lo sviluppo onde evitare possibili future anomalie, specialmente durante l'implementazione degli stessi in moduli di componenti più complesse.

Quindi, non tutte le possibili funzionalità presenti nell'applicazione useranno tutte le componenti del progetto, così da limitare la formazione bug e permettere una migliore manutenzione del sorgente.

In ogni caso, ogni qualvolta verrà realizzata una versione dell'applicazione o verranno aggiunte delle unità, si verificherà il corretto funzionamento delle unità e delle funzioni precedentemente inserite e correttamente funzionanti nel sistema.

4.3. PIATTAFORMA DI TESTING

I test delle singole componentistiche, verranno effettuati sulle macchine da lavoro che useremo per sviluppare l'applicazione in quanto Android Studio offre un simulatore di smartphone Android integrato con la possibilità di impostare valori arbitrari sulla sensoristica "fittizia" di tale dispositivo (es. simulare di essere ad una latitudine X e longitudine Y). I test sui moduli contenenti più componenti verranno anch'esse effettuate sui computer da lavoro, in quanto si vuole comprendere se due o più componenti riescono a garantire una perfetta interoperabilità al fine di non produrre errori che potrebbero danneggiare l'esito dell'implementazione finale del modulo nella prova in esterna. Infine il test dell'applicazione finale, quindi si simulerà una vera partita, questo test finale sarà svolto in due fasi, la prima affiancata dai computer di lavoro per valutare gli esiti delle varie operazioni che vengono effettuate durante l'esecuzione dell'applicazione, la seconda fase invece, prevede la simulazione di una partita a cielo aperto utilizzando solo gli smartphone e controllando successivamente le azioni effettuate nel database esterno.

5. OBIETTIVI DEL TEST DI PROGRESSIONE

Questa sezione descrive in dettaglio la progressione degli obiettivi dei test che saranno coperti dal team per questo progetto.

Questa documentazione avverrà in una tabella per semplificare la leggibilità e la tracciabilità dei test.

Funzionalità	Rif.	Obiettivo del test	Criteri di valutazione	Priorità
Creazione utente	S-01	Verificare se il file locale per la memorizzazione dei dati viene trovato o meno e, in questo secondo caso, verificare di poterlo creare e scriverci correttamente.	Ricerca va a buon fine, creazione va a buon fine, scrittura va a buon fine.	Alta
Visualizzazione profilo	S-02	Verificare la corretta lettura del file.	Lettura va a buon fine e legge i dati corretti.	Alta

Creazione partita	S-04	Verificare che nel database venga creata l'entry per la partita (prendendo i dati dal file locale dell'host) con le relative entry per i giocatori ed aver aggiunto una entry per l'host della partita.	Creazione entry nel database per partita e giocatori andata a buon fine.	Alta
Generazione id partita	S-09	Verificare che la creazione dell'id di una partita ne produca uno univoco tra le partite in corso in quel momento.	Unicità dell'id per una partita.	Alta
Comunicazione id partita	S-05	Verificare che, una volta generato l'id, questo venga comunicato all'host.	Comunicazione corretta dell'id della partita.	Alta
Numero partecipanti	S-06	Verificare la lettura delle entry del database per i giocatori.	Lettura delle entry del database andata a buon fine.	Media
Accesso ad una partita	S-08	Verificare che, inserendo l'id di una partita esistente non ancora avviata, gli utenti (nei limiti del regolamento) potranno creare delle nuove entry tra i giocatori di quella partita nel database.	Aggiunta di entry nel database andata a buon fine, non superamento dei limiti imposti.	Alta
Avvio partita	S-07	Verificare il mutamento delle entry del database da "giocatori" ad una entry per ciascuna squadra.	Manipolazione del database andata a buon fine.	Alta
Smistamento squadre	S-10	Verificare lo smistamento delle squadre in modo che si rispetti il regolamento del gioco.	Smistamento squadre corretto.	Alto
Assegnazione ruoli	S-11	Verificare l'assegnazione dei ruoli in modo che si rispetti il regolamento del gioco.	Assegnazione ruoli corretta.	Alta
Posizione bandiere	S-12	Verificare la modifica delle entry dei giocatori "Flag Keeper" con la posizione corrente.	Aggiornamento delle entry andato a buon fine e corretto.	Alta
Rilevazione posizione bandiere	S-13	Verificare che durante lo svolgimento della partita, le posizioni delle bandiere vengano aggiornate e lette correttamente da tutti i giocatori "Flag Stealer".	Lettura ripetuta ravvicinatamente delle entry andata a buon fine.	Alta
Comunicazione con la squadra	S-14	Verificare che la comunicazione con la propria squadra avvenga istantaneamente.	Passaggio di comunicazioni istantaneo.	Media

Uscire dalla partita	S-15	Verificare che, quando un giocatore decide di lasciare di proposito la partita, l'entry corrispondente a tale giocatore venga cancellata dal database e aggiunta di un punto tra le partite perse nel file locale dell'utente.	Cancellazione della entry dal database andata a buon fine, scrittura nel file locale andata a buon fine.	Alta
Lasciare la partita	—	Verificare che, quando un giocatore lascia non di proposito la partita, l'entry corrispondente a tale giocatore venga cancellata dal database e aggiunta di un punto tra le partite perse nel file locale dell'utente.	Cancellazione della entry dal database andata a buon fine, scrittura nel file locale andata a buon fine.	Alta
Annullamento per scarsità di giocatori	S-16	Verificare che, quando il numero di giocatori è inferiore a 4, la partita venga annullata per tutti i giocatori ancora in gioco (modificando l'entry della partita) e verificare che dal database venga eliminata l'entry della partita in questione.	Annullamento partita andato a buon fine per tutti i giocatori, modifica della entry andata a buon fine, cancellazione della entry andata a buon fine.	Alta
Annullamento per scarsità di bandiere	S-17	Verificare che, quando il numero di giocatori "Flag Keeper" è inferiore a 2, la partita venga annullata per tutti i giocatori ancora in gioco (modificando l'entry della partita) e verificare che dal database venga eliminata l'entry della partita in questione.	Annullamento partita andato a buon fine per tutti i giocatori, modifica della entry andata a buon fine, cancellazione della entry andata a buon fine.	Alta
Tempo di allontanamento dai giocatori avversari	S-18	Permettere agli utenti di potersi muovere nel campo fisico prima dell'effettivo inizio della partita così da poter essere distanti dai giocatori della squadra avversaria.	Il timer è funzionante e al suo termine fa iniziare la partita a tutti i giocatori.	Bassa
Aggiornamento calcolo obiettivo	S-19	Verificare che il calcolo tra la direzione e la distanza tra il proprio dispositivo e quello delle bandiere sia aggiornato quando richiesto.	Aggiornamento direzione e distanza dall'obiettivo.	Alta
Andamento partita	S-20	Verificare che tutti i dispositivi sappiano se la partita è ancora in corso o meno andandolo a leggere nel database e verifica di cancellazione della entry della partita dal database.	Lettura periodica entry della partita andata a buon fine, cancellazione entry partita andata a buon fine.	Alta
Risultato finale	S-21	Verificare che, alla	Lettura della entry della	Media

		terminazione corretta di una partita, tutti i giocatori ricevano il verdetto e verificare che in tutti i dispositivi venga aumentato di uno il punteggio delle partite vinte o perse.	squadra dal database andata a buon fine e scrittura nel file locale andata a buon fine.	
Fine partita	S-22	Verificare che, al termine di ogni partita, gli utenti tornino direttamente alla home dell'app.	Percorso eseguito correttamente.	Media

6. ALTRI TEST

6.1. PRIVACY

Per assicurare che l'applicazione rispetti la privacy degli utenti, verranno effettuati dei test sul database per controllare che non vengano salvati i dati dei giocatori per un tempo più lungo della durata della partita a cui partecipano, come la loro posizione associata al loro id personale.

I dati che verranno immessi nel database verranno automaticamente eliminati alla fine di ogni partita giocata, sarà possibile accedere in lettura e scrittura al database solamente in riferimento ad una determinata partita identificata da un Id univoco presente al momento nel database, pertanto i giocatori non potranno accedere a dati fuori dal loro contesto di gioco, inoltre, solamente chi possiede l'app potrà accedere al database, infatti è possibile accedere al database solo agli sviluppatori per fare i test o della manutenzione.

Funzionalità	Rif.	Obiettivo del test	Criteri di valutazione	Priorità
Privacy	S-15/ 16/ 17/ 20	Verificare la cancellazione dal database dei dati di una partita con tutte le relative entry con i dati sui giocatori e verificare che quando un giocatore esce o lascia una partita la sua entry venga cancellata.	I dati riguardanti la partita cancellata e i suoi giocatori non devono più essere presenti sul database.	Alta

6.2. CONNETTIVITÀ

Essendo la connessione ad internet e al GPS fondamentale per il funzionamento dell'app durante le partite, sarà necessario assicurarsi che la connessione rimanga sempre attiva, senza anomalie e che la precisione del GPS sia accurata e sempre aggiornata.

Funzionalità	Rif.	Obiettivo del test	Criteri di valutazione	Priorità
Refresh della posizione	S-12/13/19	Verificare di ottenere dati in real-time sulle posizioni.	I dati riguardanti la posizione dovranno essere aggiornati entro un massimo di 3 secondi.	Alta
Connessione ad internet	S-04/09/06/12/13/14/18/19/20	Verificare la costante comunicazione con il database esterno.	Lettura periodica dei dati aggiornati attraverso il log di sistema.	Alta

6.3. STRESS E VOLUME

Per garantire un'adeguata robustezza del sistema verranno effettuati dei test sulla capacità di dati che il database può sopportare e con quanta rapidità possono variare ed essere letti per un'esperienza di gioco fluida e senza intoppi o annullamenti. Si farà riferimento anche ai test per il caricamento.

Funzionalità	Rif.	Obiettivo del test	Criteri di valutazione	Priorità
Traffico dati in scrittura nel database	S-04/09/08/07/11/12/14/18/19	Verificare la corretta scrittura in un contesto dove più dispositivi scrivono nel database simultaneamente.	Verrà testata la scrittura simultanea da parte di più simulatori, i quali aggiornano le proprie entry nel database ripetutamente per comprendere se possono generare conflitti.	Media
Lettura dal database	S-09/08/07/10/11/13/14/18/20/21	Verificare la lettura simultanea da parte di più dispositivi dei dati all'interno del database esterno.	Lettura simultanea da più simulatori di entry diverse, se il test avrà un esito positivo, si testerà la lettura simultanea di una entry soltanto da parte degli stessi simulatori.	Alta

6.4. RIPRISTINO DI EMERGENZA / BACKUP

Al fine di non perdere del lavoro sviluppato, verranno effettuati dei backup su un sistema di versionamento (Github) in quanto il rischio di perdere dati è imprevedibile e nel caso nel quale si produca una versione instabile o corrotta del codice sarà possibile ritornare ad una versione precedente del programma.

6.5. TEST SULLE UNITÀ

Ogni singola unità implementata contenente nuove funzionalità verrà testata singolarmente dai membri del team affinché corrisponda al proprio obiettivo e funzioni correttamente producendo il risultato sperato. Per unità si identificano sia componenti, sia singole porzioni di codice.

6.6. TEST DI INTEGRAZIONE

Successivamente ai test sulle unità, queste verranno testate sulla loro cooperazione perché non vadano in conflitto l'una con l'altra, pertanto, la loro comunicazione dovrà essere stabile e priva di errori. I test di integrità vedranno più step, con una inglobazione di unità sempre maggiore, quindi, in maniera incrementale, per cercare di minimizzare gli errori ed individuarli man mano.

7. TEST SCHEDULING

7.1. LIVELLI DI RESPONSABILITÀ

In questa sezione si esplicheranno i livelli di responsabilità dei test che devono essere applicati.

Livello di test	Team di progetto	Professore	Esterno
Privacy	P	—	—
Connettività	P	S	S
Stress e volume	P	—	—
Ripristino di emergenza / backup	P	—	S
Test sulle unità	P	—	—
Test di integrazione	P	—	—

Legenda:

- P: Responsabilità primaria
- S: Responsabilità secondaria

7.2. APPROCCIO E TIPI DI TEST

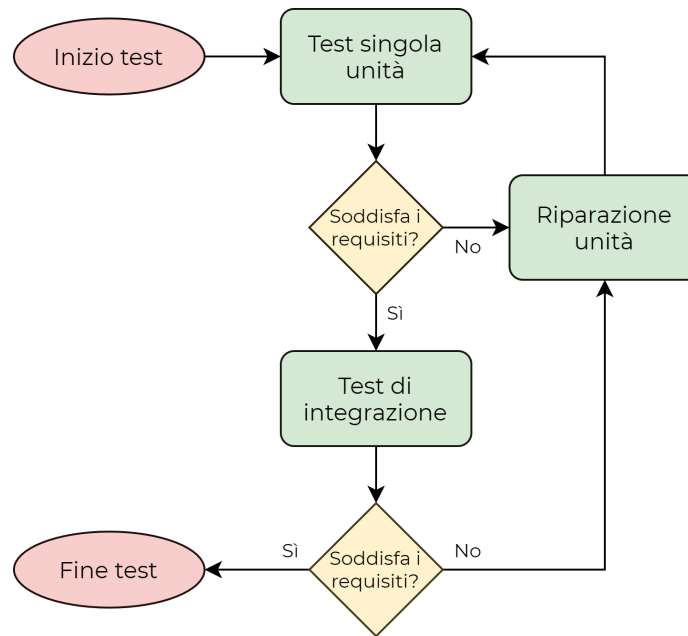
L'approccio che si è deciso di adottare è un approccio misto. I test verranno eseguiti in maniera **back-to-back** per assicurarsi che le diverse versioni del programma rispettino sempre le versioni precedenti senza riscontrare risultati differenti in merito ai test precedentemente eseguiti. I test sulle versioni verranno, però, eseguiti come test **bottom-up** per assicurarsi che prima funzionino le diverse unità e poi che esse funzionino insieme.

- La scelta di effettuare dei test back-to-back è stata indirizzata dal fatto di testare diverse versioni del programma con lo stesso input, e confrontare gli output ottenuti in modo che non si vengano mai a creare problemi nuovi su questioni e problemi già affrontate. Essendo l'approccio del team di tipo Agile, il progetto prevede più versioni dell'applicazione per avere diversi risultati e prototipi, questa tecnica di testing è ottimale per l'individuazione di errori tra versioni diverse.
- La scelta di effettuare dei test bottom-up è stata indirizzata dal fatto di iniziare dalle componenti più a basso livello, per verificare che tutte le componenti funzionino correttamente, prima singolarmente e poi insieme. Questo rende più difficoltoso individuare errori di alto livello all'inizio, ma permette di avere una base solida su un sistema orientato ad oggetti.

I test che si andranno a realizzare saranno di due diverse tipologie, riportate nella tabella qui sotto.

Tipo di test	Obiettivi
Requisiti di progressione	<p>L'app deve:</p> <ul style="list-style-type: none"> • rispettare i requisiti definiti; • funzionare e performare accuratamente; • approcciare correttamente le condizioni di errore; • far funzionare le interfacce correttamente; • comunicare correttamente con il database; • caricare e scrivere i dati correttamente. <p>I test occorreranno periodicamente e in maniera controllata, affinché la soluzione coincida con i requisiti definiti.</p>
Test di regressione	<p>Ad ogni versione dell'app e ad ogni aggiunta di componenti, è necessario che le vecchie componenti e le vecchie funzionalità continuino a funzionare correttamente.</p>

Per ogni test, la procedura sarà la seguente.



7.3. COSTRUIRE UNA STRATEGIA

Come accennato precedentemente, verrà adottata una strategia di tipo bottom-up, ovvero, si svilupperanno tutti i piccoli componenti in delle classi, le quali verranno testate singolarmente in activity di test analizzando il log presente nella console, successivamente, se tali componenti soddisfano i requisiti, si procede ad inglobare gli stessi in dei moduli, ovvero delle classi contenenti più componenti per analizzare la loro interoperabilità. Come per le singole componenti, i test avverranno in delle activities di prova analizzando il log durante l'esecuzione dell'applicazione. Infine verrà svolto un test di integrazione dove si analizzeranno i possibili errori generati durante l'esecuzione dell'applicazione stessa. L'approccio che abbiamo deciso di scegliere è di avere questi test bottom-up per comporre dei test back-to-back da effettuare ad ogni versione del programma per verificarne la correttezza in merito agli aspetti già consolidati nella versione precedente e come questi si incastrano con le nuove funzionalità. Questa strategia prevede una manutenzione/riparazione costante dei componenti "difettosi" ogni qualvolta vengono prodotti e testati.

7.4. PIANIFICAZIONE DELL'ESECUZIONE DEI TEST

I test verranno effettuati secondo la valutazione dello schema del test di progressione del presente documento (vedi sezione 5 e 6), l'ordine dei test avverrà seguendo tali tabelle. È necessario pianificare correttamente quali test verranno prima di altri, in quanto alcuni moduli sfrutteranno le operazioni di diverse componenti, tale attività permette inoltre di favorire l'incapsulamento e la modularità dei sorgenti all'interno dell'applicazione.

7.5. STRUTTURA, DATI E PIANO DI FORNITURA DELLE RISORSE

Non viene definita una struttura precisa per effettuare i test, come anticipato precedentemente i test dovranno essere svolti attraverso l'utilizzo delle proprie macchine da lavoro durante lo sviluppo del progetto stesso, pertanto sono test che si possono svolgere direttamente lavorando in remoto, questa azione è necessaria in quanto vista la pandemia che affligge l'Italia e il resto del mondo, non è possibile spostarsi liberamente.

Dopo aver implementato correttamente tutti i componenti e i moduli nell'applicazione dimostrativa, sarà necessario trovarsi per testare l'applicazione a cielo aperto, inoltre, come accennato nei documenti precedenti, il test all'aperto deve essere svolto con le condizioni a seguire:

- dovranno esserci almeno 4 dispositivi Android con l'applicazione installata correttamente ed avviata, è inoltre necessario che i dispositivi siano dotati di connessione ad Internet;
- il campo di gioco scelto deve permettere il libero movimento dei giocatori della partita, e possibilmente, senza intrusioni da parte di persone esterne al test, sempre per la pandemia sarà doveroso garantire un distanziamento sociale di almeno 1 metro da ogni persona;
- dovranno essere svolte più partite per testare i vari casi: partita vinta da un team o da un altro e partita pareggiata. Inoltre è necessario analizzare successivamente i cambiamenti avvenuti nel punteggio di ogni giocatore e i movimenti effettuati nel database esterno.

7.6. STRUMENTI DI TEST

In questa sezione verranno dichiarati gli strumenti che saranno utilizzati dal team per i test.

Processo	Strumento
Creazione casi di test	Documenti Google - Piano di testing
Tracciamento casi di test	Fogli Google
Esecuzione casi di test	Manualmente
Gestione casi di test	Fogli Google
Gestione difetti	Fogli Google

7.7. VERIFICA DELLA PROCEDURA DI TESTING

Ogni risultato prodotto sui singoli componenti verrà comunicato ai membri del team per procedere alla sua effettiva implementazione all'interno di altri moduli o nell'applicazione stessa, inoltre verrà modificato il presente documento per mantenere la tracciabilità dei casi di test. È necessario operare con più membri per garantire che l'applicazione non presenti anomalie successivamente. Solamente l'applicazione finale, quindi la versione dotata di interfaccia grafica verrà consegnata al docente assieme a tutta la documentazione prodotta in fase di sviluppo.

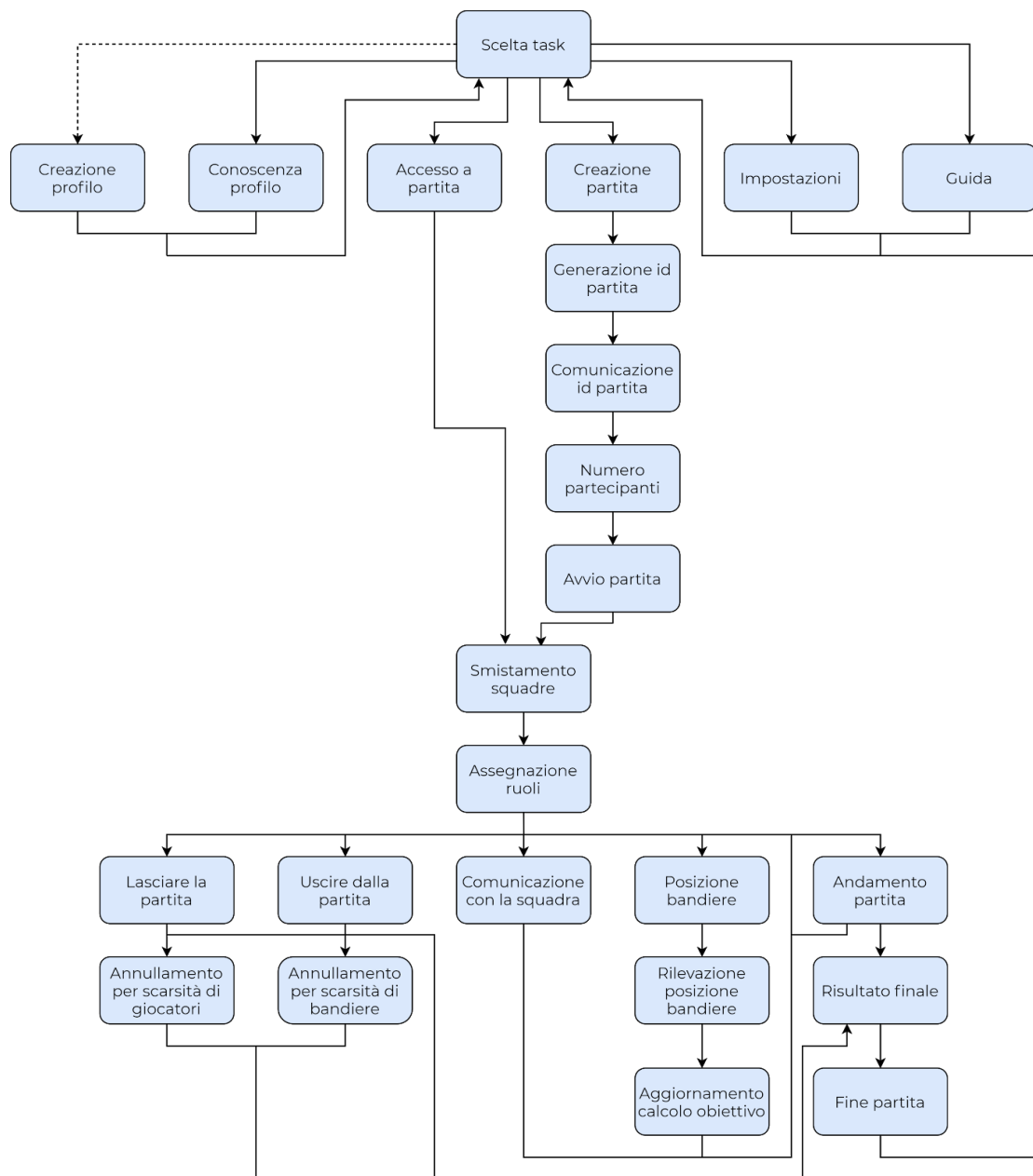
7.8. METRICHE

Seguendo le linee guida dettate dalla tabella del test di progressione (Sezione 5 del presente documento) si procederà ai test dei componenti sia per i singoli, che per componenti più complesse e non si potrà procedere allo sviluppo di altri componenti se non sono stati risolti i conflitti generati da quelli appena sviluppati, appunto per evitare di imbattersi in numerosi errori successivamente. Possiamo permettere che avvengano alcune anomalie dettate dall'ambiente in cui testeremo l'applicazione, ad esempio la localizzazione non potrà essere accurata in ambienti chiusi, pertanto non si aggiornerà costantemente, tuttavia dovrà assolutamente operare secondo la specifica nel suo ambiente reale d'uso, dunque, all'aperto dovrà aggiornarsi molto frequentemente.

8. PIANO DELL'AMBIENTE DI TEST

8.1. TEST SULLE INTERAZIONI

Qui verrà mostrato il diagramma di interazioni che si potranno avere con l'ambiente di test.



8.2. DETTAGLI SULL'AMBIENTE DI TEST

Essendo che l'applicazione deve avere un minimo di giocatori, è necessario che tutti i componenti del gruppo testino l'applicazione contemporaneamente, possibilmente su dispositivi diversi in modo tale da scoprire eventuali malfunzionamenti in ambienti diversi. L'ambiente dove si svolgeranno i vari test, è il campus scientifico dell'Università Ca' Foscari situato in Via Torino 155 (Mestre). L'ambiente di gioco simulato sarà diviso in due modalità, un test che si svolgerà completamente all'aperto, per vedere come si comporta l'applicazione nell'ambiente ideale e una prova dove ci sarà la possibilità di entrare all'interno di alcuni edifici visto che al loro interno, la ricezione del segnale non è sempre ottima. Il secondo test servirà anche a capire come si comporta l'applicazione nel controllo dei requisiti per continuare la partita (per esempio, se l'host perde il segnale la partita deve essere annullata) e vedere, se in mancanza di essi, effettivamente la partita viene annullata, in che modalità ed in quanto tempo.

8.3. SCELTA DELL'AMBIENTE

Qui vengono definiti i requisiti per l'ambiente di test per le varie funzionalità.

Funzionalità	Requisiti
Creazione utente	L'applicazione deve essere appena stata aperta.
Visualizzazione profilo	Nel dispositivo deve essere presente il file in locale con i dati dell'utente.
Creazione partita	Nel dispositivo deve essere presente il file in locale con i dati dell'utente e il dispositivo deve essere connesso ad internet e avere il GPS attivo.
Generazione id partita	La partita deve essere stata creata e il dispositivo deve mantenere i requisiti stabiliti per quella funzionalità.
Comunicazione id partita	La partita deve essere stata creata e il dispositivo deve mantenere i requisiti stabiliti per quella funzionalità.
Numero partecipanti	La partita deve essere stata creata e il dispositivo deve mantenere i requisiti stabiliti per quella funzionalità.
Accesso ad una partita	Nel dispositivo deve essere presente il file in locale con i dati dell'utente e il dispositivo deve essere connesso ad internet e avere il GPS attivo.
Avvio partita	La partita deve essere stata creata, il dispositivo deve mantenere i requisiti stabiliti per la funzionalità "Creazione partita" o "Accesso ad una partita" e il numero di entry di giocatori nel database deve essere tra le 4 e le 10..
Smistamento squadre	La partita deve essere stata avviata e il dispositivo deve mantenere i requisiti stabiliti per quella funzionalità.
Assegnazione ruoli	La partita deve essere stata avviata e il dispositivo deve mantenere i requisiti stabiliti per quella funzionalità.
Posizione bandiere	La partita deve essere stata avviata e il dispositivo deve mantenere i requisiti stabiliti per quella funzionalità.
Rilevazione posizione bandiere	La partita deve essere stata avviata e il dispositivo deve mantenere i requisiti stabiliti per quella funzionalità.
Comunicazione con la squadra	La partita deve essere stata avviata e il dispositivo deve mantenere i requisiti stabiliti per quella funzionalità.
Uscire dalla partita	La partita deve essere stata avviata e il dispositivo deve mantenere i requisiti stabiliti per quella funzionalità.
Lasciare la partita	La partita deve essere stata avviata.
Annullamento per scarsità di giocatori	La partita deve essere stata avviata e il dispositivo deve mantenere i requisiti stabiliti per quella funzionalità.
Annullamento per scarsità di bandiere	La partita deve essere stata avviata e il dispositivo deve mantenere i requisiti stabiliti per quella funzionalità.

Tempo di allontanamento dai giocatori avversari	La partita deve essere stata avviata e il dispositivo deve mantenere i requisiti stabiliti per quella funzionalità.
Aggiornamento calcolo obiettivo	La partita deve essere stata avviata e il dispositivo deve mantenere i requisiti stabiliti per quella funzionalità.
Andamento partita	La partita deve essere stata avviata e il dispositivo deve mantenere i requisiti stabiliti per quella funzionalità.
Risultato finale	La partita deve essere terminata in maniera corretta.
Fine partita	La partita deve essere terminata in maniera corretta.
Privacy	La partita deve essere stata conclusa per poter effettuare la cancellazione dei dati di tale partita.
Refresh della posizione	È necessario avere il GPS attivo, possibilmente con la precisione aumentata.
Connessione ad Internet	Bisogna disporre di una connessione ad internet per permettere lo scambio di dati con il database.
Traffico dati in scrittura nel database	Deve essere stata avviata una connessione con il database e tutti i dispositivi associati ad una determinata partita devono essere dotati di connessione ad Internet.
Lettura dal database	Deve essere stata avviata una connessione con il database e tutti i dispositivi associati ad una determinata partita devono essere dotati di connessione ad Internet.

8.4. CONTROLLO DELL'AMBIENTE

Prima di iniziare con i vari test, è necessario controllare i requisiti d'ambiente::

- **Dispositivi Android:** Essendo l'applicazione sviluppata interamente per dispositivi Android, è necessario che i tester abbiano tutti un telefono con questo sistema operativo. possibilmente i dispositivi dovranno avere la batteria carica per completare almeno una partita;
- **Migliora Posizione:** I dispositivi utilizzati dovranno avere la possibilità di migliorare la precisione di localizzazione, senza questa opzione è possibile giocare, ma è probabile che la geolocalizzazione sia meno precisa e di conseguenza la partita potrebbe riscontrare dei problemi.
- **Ambiente di test:** L'area di test dovrà essere ben coperta dal segnale telefonico in modo da avere condizioni ottimali per quanto riguarda il segnale. Il primo test si svolgerà interamente in un'area all'aperto, mentre nel secondo, sarà possibile anche entrare negli edifici, in modo da vedere come si comporta l'applicazione in momenti non ottimali;
- **Meteo:** La giornata di test sarà possibile effettuarla solo con un clima soleggiato in modo da non avere impedenze sul segnale ed inoltre, non è possibile organizzare una partita con un clima temporalesco.
- **Numero minimo di partecipanti:** Non è possibile eseguire un test se sono presenti meno di quattro persone.

8.5. RUOLI E RESPONSABILITÀ SULL'AMBIENTE

In questa sezione sono definiti dei ruoli, le responsabilità che comporta ciascun ruolo e chi sarà associato a quei ruoli.

Ruolo	Responsabilità	Responsabile
Condizioni climatiche	Controllare le condizioni climatiche affinché esse siano favorevoli e miti, per permettere di verificare un uso corretto e accurato dell'app.	Mario Coci
Stato del sistema	Verificare che tutti i dispositivi usati siano aggiornati all'ultima versione e a piena carica.	Simone Checco
Connessione e GPS	Garantire la presenza e la stabilità della connessione ad internet (Wi-fi o dati mobili, preferibilmente la seconda) e del GPS acceso ed impostato su "precisione migliorata".	Melania Gottardo
Collegamento database	Assicurarsi che il sistema comunichi in maniera stabile con il database in maniera rapida.	Monan Nasir

9. PRESUPPOSTI E DIPENDENZE

9.1. PRESUPPOSTI

Lo sviluppo è stato diviso in moduli ed assegnati ai rispettivi membri del team. I moduli e le funzionalità sono specificati nel documento di progetto per ritenere l'applicazione completa e pronta all'uso. Lo sviluppo è stato diviso in moduli che andranno testati singolarmente prima dell'inserimento nell'applicazione. Una volta inseriti si andrà anche a testare l'applicazione stessa in modo da rilevare eventuali malfunzionamenti tra moduli. Una volta terminati i moduli, si andrà a testare l'applicazione intera, in modo tale che, se si dovesse riscontrare qualche malfunzionamento basterà controllare il modulo interessato e correggerlo.

9.2. DIPENDENZE

Prima della prova finale, sarà necessario attivare il miglioramento delle posizioni in modo da avere una precisione migliore, caratteristica abbastanza fondamentale per quanto riguarda il funzionamento del gioco. Inoltre, per verificare se l'applicazione funziona in ogni sua parte, sarà necessario simulare una vera e propria partita di conseguenza, per questo, ogni test deve essere fatto in un ambiente molto spazioso e con una buona copertura di segnale telefonico. Sarà inoltre necessario verificare come si comporta l'applicazione quando non si verificano più i presupposti per continuare la partita, quindi verranno simulati degli scenari dove l'applicazione dovrà annullare la partita. Visto che la maggior parte dei test dovrà svolgersi ad una elevata distanza il team dovrà rimanere sempre in contatto telefonico in modo tale da capire se lo scenario in cui sono capitati è voluto o meno.

10. CRITERI DI INIZIO E FINE TEST

Qui sono definiti in dettaglio i criteri per i quali un test può cominciare o può essere definito concluso. Questo aiuta a capire a quale fase di test il team è arrivato.

Fase di testing	Quando può iniziare	Quando è finito
Creazione utente	—	Ricerca va a buon fine, creazione va a buon fine, scrittura va a buon fine.
Visualizzazione profilo	Dopo "Creazione utente".	Lettura va a buon fine e legge i dati corretti.
Creazione partita	Dopo "Creazione utente".	Creazione entry nel database per partita e giocatori andata a buon fine.
Generazione id partita	Dopo "Creazione partita".	Unicità dell'id per una partita.
Comunicazione id partita	Dopo "Creazione partita".	Comunicazione corretta dell'id della partita.
Numero partecipanti	Dopo "Creazione partita".	Lettura delle entry del database andata a buon fine.

Accesso ad una partita	Dopo "Creazione utente".	Aggiunta di entry nel database andata a buon fine, non superamento dei limiti imposti.
Avvio partita	Dopo "Creazione partita".	Manipolazione del database andata a buon fine.
Smistamento squadre	Dopo "Avvio partita".	Smistamento squadre corretto.
Assegnazione ruoli	Dopo "Avvio partita".	Assegnazione ruoli corretta.
Posizione bandiere	Dopo "Avvio partita".	Aggiornamento delle entry andato a buon fine e corretto.
Rilevazione posizione bandiere	Dopo "Avvio partita" e "Posizione bandiere".	Lettura ripetuta ravvicinatamente delle entry andata a buon fine.
Comunicazione con la squadra	Dopo "Avvio partita".	Passaggio di comunicazioni istantaneo.
Uscire dalla partita	Dopo "Avvio partita".	Cancellazione della entry dal database andata a buon fine, scrittura nel file locale andata a buon fine.
Lasciare la partita	Dopo "Avvio partita".	Cancellazione della entry dal database andata a buon fine, scrittura nel file locale andata a buon fine.
Annullamento per scarsità di giocatori	Dopo "Avvio partita" e "Uscire dalla partita" o "Lasciare la partita".	Annullamento partita andato a buon fine per tutti i giocatori, modifica della entry andata a buon fine, cancellazione della entry andata a buon fine.
Annullamento per scarsità di bandiere	Dopo "Avvio partita" e "Uscire dalla partita" o "Lasciare la partita".	Annullamento partita andato a buon fine per tutti i giocatori, modifica della entry andata a buon fine, cancellazione della entry andata a buon fine.
Tempo di allontanamento dai giocatori avversari	Dopo "Avvio partita"	Il timer è funzionante e al suo termine fa iniziare la partita a tutti i giocatori.
Aggiornamento calcolo obiettivo	Dopo "Avvio partita" e "Posizione bandiere" e "Rilevazione posizione bandiere".	Aggiornamento direzione e distanza dall'obiettivo.
Andamento partita	Dopo "Avvio partita".	Lettura periodica entry della partita andata a buon fine, cancellazione entry partita andata a buon fine.

Risultato finale	Dopo "Andamento partita".	Lettura della entry della squadra dal database andata a buon fine e scrittura nel file locale andata a buon fine.
Fine partita	Dopo "Risultato finale".	Percorso eseguito correttamente.
Privacy	Dopo "Creazione partita" o "Accesso ad una partita".	I dati riguardanti la partita cancellata non devono più essere presenti sul database.
Refresh della posizione	Dopo "Creazione partita" o "Accesso ad una partita".	I dati riguardanti la posizione dovranno essere aggiornati entro un massimo di 3 secondi.
Connessione ad Internet	Dopo "Creazione partita" o "Accesso ad una partita".	Lettura periodica dei dati aggiornati attraverso il log di sistema.
Traffico dati in scrittura nel database	Dopo "Creazione partita" o "Accesso ad una partita".	Verrà testata la scrittura simultanea da parte di più simulatori, i quali aggiornano le proprie entry nel database ripetutamente per comprendere se possono generare conflitti.
Lettura dal database	Dopo "Creazione partita" o "Accesso ad una partita".	Lettura simultanea da più simulatori di entry diverse, se il test avrà un esito positivo, si testerà la lettura simultanea di una entry soltanto da parte degli stessi simulatori.

11. PIANO AMMINISTRATIVO

11.1. APPROVAZIONI

Le seguenti task, riportate nella tabella, sono state assegnate ad un responsabile e ad una persona che dovrà approvare la task al suo termine.

Task	Responsabile	Approvatore
Configurazione database	Monan Nasir	Checco Simone
Costruzione UI ed activities di base	Coci Mario	Gottardo Melania
Progettazione meccanismo di tracking	Gottardo Melania	Coci Mario
Sviluppo delle classi	Checco Simone	Monan Nasir
Collegamento UI alla business logic	Checco Simone	Monan Nasir
Sviluppo delle activities	Gottardo Melania	Coci Mario
Ampliamento e miglioramento classi activities	Coci Mario	Gottardo Melania
Progettazione chat di gruppo	Monan Nasir	Checco Simone
Gestione molteplici partite simultanee	Monan Nasir	Checco Simone

11.2. TEST MILESTONES

Qui vediamo una struttura che identifica le milestones, per quanto riguarda i test (alcuni dei test back-to-back), a cui viene associata un'ipotetica data di raggiungimento pianificata alla quale dovrà essere affiancata la data di raggiungimento veritiera.

Milestone	Data di fine pianificata	Data di fine effettiva
Rilascio demo	15 Dicembre 2020	30 Dicembre 2020
Versione completa	23 Dicembre 2020	5 Gennaio 2021
Applicazione finale	31 Dicembre 2020	8 Gennaio 2021

11.3. GESTIONE DEI DIFETTI

Nel momento in cui si dovessero riscontrare dei malfunzionamenti o delle anomalie, si cercherà di capire da dove e da cosa derivano e si elaborerà una nuova soluzione. Se invece si dovesse scoprire che non ci sono soluzioni possibili, allora si dovrà elaborare una nuova strategia per aggirare il problema, oppure, risulterebbe molto dispendioso in termini di tempo.

12. RIFERIMENTI

Per creare questo documento sono stati utilizzati come riferimento:

- “piano di testing” di alcuni gruppi degli anni passati;
 - documenti già stilati dal team (piano di progetto e documento di analisi e specifica);
 - template di un “piano di testing” messo a disposizione dal professore.
-