# Treasure Hunt



Ingegneria del Software

**Di Cerbo Enrico 399000124 Prof M. Di Penta**

**Merolese Salvatore 399000125**

**Salomone Ennio 399000122**

Sommario

**Analisi e specifica dei requisiti software**

## 1.Introduzione

Treasure hunt è un’applicazione android improntata sulla caccia al tesoro, integrando l’esercizio sportivo podistico con una componente ludica caratterizzata da quiz.

L’applicazione fornisce ai runner, o “hunter”, di mettere alla prova le proprie capacità intellettuali attraverso la risoluzione di un numero prefissato di quiz sequenziali appartenenti ad uno stesso percorso. Tali quiz sono attivabili in specifici punti di una città, permettendo all’hunter di testare cosi anche le proprie doti atletiche spostandosi da un punto all’altro del percorso per attivare i vari checkpoint della “caccia al tesoro”.

Un hunter può creare un caccia al tesoro, decidendo dove inserire i checkpoint e quale domanda collegare ad esso. Tali domande possono essere scelte da un elenco prefissato di quiz così da non fornire vantaggi a chi crea la sfida.

Tutte le funzionalità di Treasure Hunt sono accessibili solo ad utenti registrati che hanno effettuato il login.

### 1.1 Definizioni e termini chiave

* *TreasureHunt*: nome dell’applicazione
* *Sfida*: competizione sportiva tra diversi Treasure Hunter
* *Percorso*: Un insieme di checkpoint che l’hunter deve raggiungere per poter terminare la sfida.
* *Archivio:* termine generico per indicare la persistenza di informazioni su ciclisti, sfide, ecc.
* *Hunter*: Utilizzatore dell’applicazione che dovrà registrarsi per effettuare l’accesso e potrà scegliere se creare una nuova sfida o partecipare ad una sfida già ideata da altri hunter.
* *Indovinello*: Enigma che l’utente deve risolvere per poter continuare il percorso.
* *Checkpoint*: Punto intermedio del percorso in cui l’hunter deve risolvere un indovinello proposto.
* *Credenziali:* dati inseriti nella fase di Login per accedere all’applicazione
* *Form:* interfaccia che consente all’utente di inserire i dati
* *Geoposizione:* posizione geografica del ciclista sulla mappa
* *Marker:* icona che rappresenta la posizione di un *hunter* sulla mappa

## 2.Utenti

Gli utenti a cui è rivolta l’app sono runners e podisti di livello amatoriale che vogliono avvicinarsi a questo sport integrando una componente social e ludica.

Gli hunter possono utilizzare le seguenti funzionalità:

* Creare una caccia al tesoro
* Partecipare ad una caccia al tesoro

## 3.Descrizione Generale

**Treasure Hunt** permette agli hunter di:

* Registrarsi
* Effettuare il Login
* Visualizzare sulla mappa le sfide disponibili
* Partecipare ad una sfida
* Creare una sfida
* Visualizzare i propri risultati
* Effettuare il Logout

## 4.Componenti dell’applicazione

**Treasure Hunt** è costituito da:

* Un dispositivo con sitema opeativo Android
* Un’interfaccia grafica che mette a disposizione degli utenti le varie funzionalità dell’applicazione
* Un server che ospita un database dove sono memorizzate le informazioni relative agli utenti e alle sfide

## 5.Business Rules

* Un utente non registrato non può accedere alle funzionalità dell’applicazione.
* Se un utente in fase di registrazione non compila tutti i campi o inserisce informazione valide come un indirizzo email senza ‘@’ non potrà registrarsi e dovrà ripetere l’operazione.
* Se un utente registrato prova ad accedere omettendo uno dei campi o inserendo credenziali errate non potrà accedere all’app.
* Se un utente registrato inserisce credenziali errate non potrà accedere all’applicazione.
* Un utente che ha effettuato l’accesso può decidere se partecipare ad una caccia al tesoro disponibile o di crearne una nuova.

## 6.Specifiche dei casi d’uso

|  |  |
| --- | --- |
| **Caso d’uso: UC-01** | **Registrazione** |
| **ATTORI** | * Hunter * Archivio Utenti |
| **INPUT** | * Email * Username * Password |
| **PRECONDIZIONI** | Email e username non devono essere già nell’archivio |
| **OUTPUT** | All’hunter viene notificata l’avvenuta registrazione |
| **POSTCONDIZIONI** | L’hunter è memorizzato nel database |
| **SCENARIO PRIMARIO** | 1. L’hunter dall’activity login, cioè l’activity di lancio, accede all’activity di registrazione 2. L’hunter inserisce le proprie credenziali: 3. Email 4. Username 5. Password 6. Il sistema verifica che il form sia riempito in tutti i suoi campi 7. Il sistema verifica che nel campo mail sia presente in carattere ‘@’ 8. Il sistema verifica che l’email non sia presente nell’archivio 9. Il sistema verifica che l’username non sia presenta nell’archivio 10. Il sistema informa il ciclista della corretta registrazione 11. Il caso d’uso termina |
| **SCENARIO ALTERNATIVO A** | **L’utente omette uno dei campi** |
|  | 1. Il sistema informa l’hunter con un messaggio di errore 2. Il caso d’uso riparte dal punto 2 |
| **SCENARIO ALTERNATIVO B** | **L’email o l’username sono già presenti in archivio** |
|  | 1. Il sistema informa il ciclista con un messaggio di errore 2. Il caso d’uso riparte dal punto 2 |
| **SCENARIO ALTERNATIVO C** | **Uno o più campi del form non sono compilati correttamente** |
|  | 1. Il sistema informa il ciclista con un messaggio di errore 2. Il caso d’uso riparte dal punto 2 |

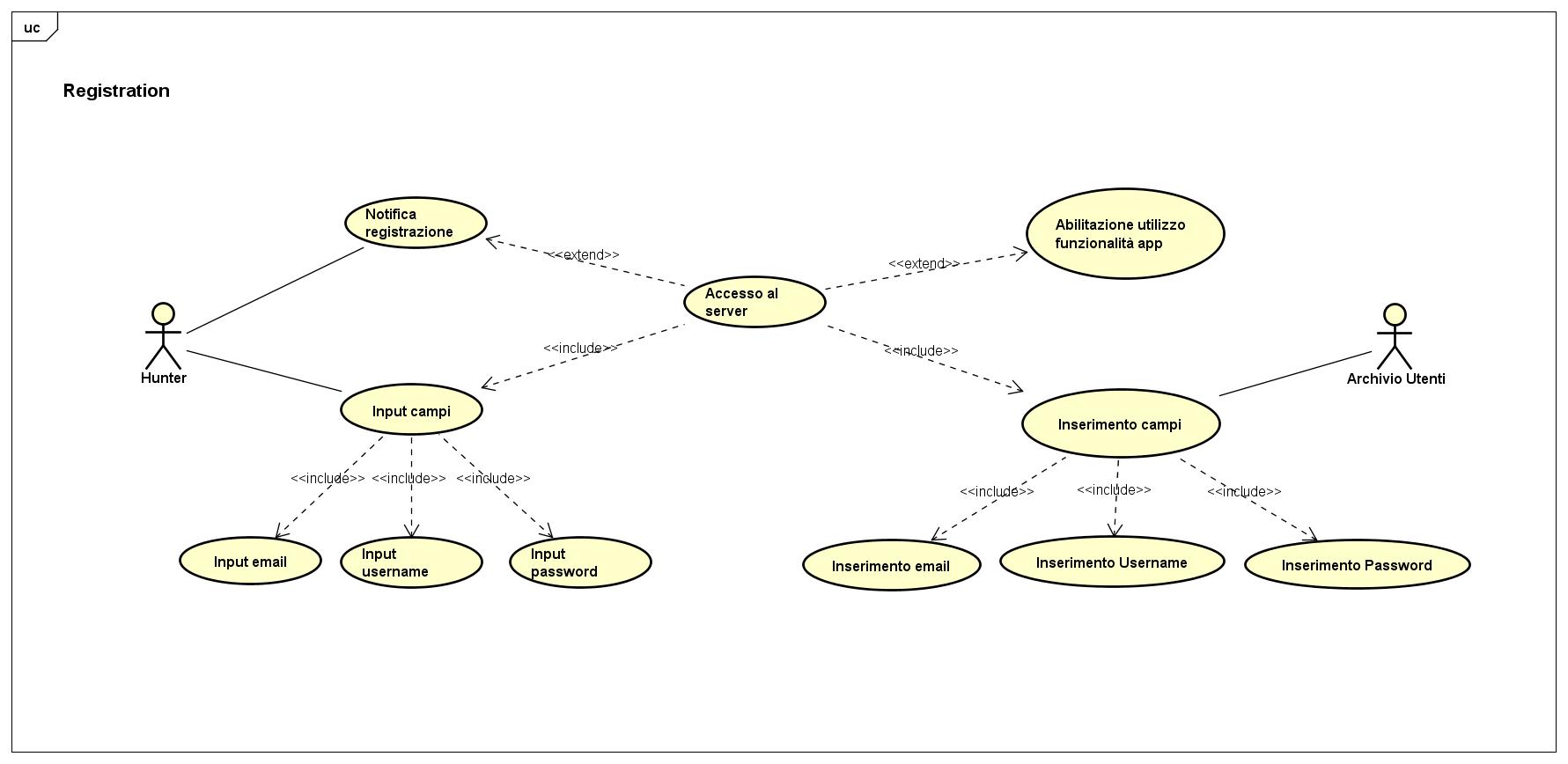
|  |  |
| --- | --- |
| **Caso d’uso: UC-02** | **Login** |
| **ATTORI** | * Hunter * Archivio Utenti |
| **INPUT** | * Username * Password |
| **PRECONDIZIONI** | * L’username sia presente nell’archivio * La password sia associata all’username |
| **OUTPUT** | * All’Hunter viene notificato del corretto accesso * L’hunter è abilitato all’utilizzo dell’applicazione |
| **POSTCONDIZIONI** | * L’hunter è abilitato all’utilizzo del sistema |
| **SCENARIO PRIMARIO** | 1. L’hunter inserisce le proprie credenziali:    1. Username    2. Password 2. Il sistema verifica che il form sia riempito in tutti i suoi campi 3. Il sistema verifica che l’username sia presente nell’archivio 4. Il sistema verifica che la password corrisponda all’username inserita 5. Il sistema informa l’hunter del corretto accesso 6. L’hunter accede alle funzionalità dell’applicazione 7. Il caso d’uso termina |
| **SCENARIO ALTERNATIVO A** | **Il form non viene compilato con tutte le informazioni richieste** |
|  | 1. Il sistema informa l’hunter con un messaggio di errore 2. Il caso d’uso riparte dal punto 1. |
| **SCENARIO ALTERNATIVO B** | **La password non corrisponde all’username inserita** |
|  | 1. Il sistema informa l’hunter con un messaggio di errore 2. Il caso d’uso riparte dal punto 1. |

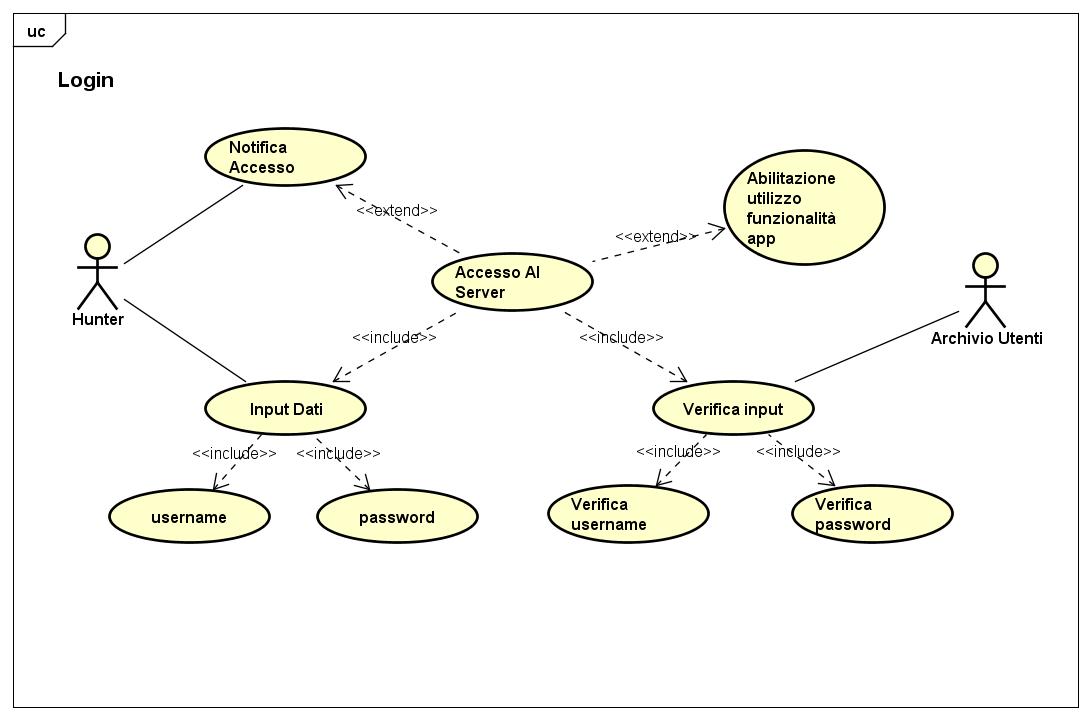
|  |  |
| --- | --- |
| **Caso d’uso: UC-03** | **Creazione di una sfida** |
| **ATTORI** | * Utente |
| **INPUT** |  |
| **PRECONDIZIONI** | * L’utente abbia effettuato l’accesso all’applicazione * L’utente sia stato geolocalizzato * La connessione a Internet sia attiva |
| **OUTPUT** | * Un oggetto sfida nel database |
| **POSTCONDIZIONI** | * La sfida creata è memorizzata sul database * La sfida può essere scelta dagli utenti quando selezionano il bottone di inizio sfida |
| **SCENARIO PRIMARIO** | 1. L’utente accede all’applicazione (vedi **UC-02**) 2. L’utente viene geolocalizzato (vedi **UC-03**) 3. L’utente seleziona la voce “Create Challenge” dal menù 4. L’utente clicca su un punto della mappa che sarà il primo checkpoint della sfida 5. L’utente si trova in una nuova actvity Question 6. L’utente clicca sullo spinner degli indovinelli per sceglierne uno da inserire 7. L’utente clicca sul tasto “ok” per associare l’indovinello al checkpoint 8. L’utente può scegliere di inserire altri checkpoint 9. Una volta inserito l’ultimo checkpoint l’utente clicca sul tasto “confirm” 10. L’utente riceve una notifica secondo cui la sfida è avvenuta con successo 11. Il caso d’uso termina |
| **SCENARIO ALTERNATIVO A** | **Annullamento durante la creazione**   1. L’utente clicca sul tasto “back” 2. Il caso d’uso termina |

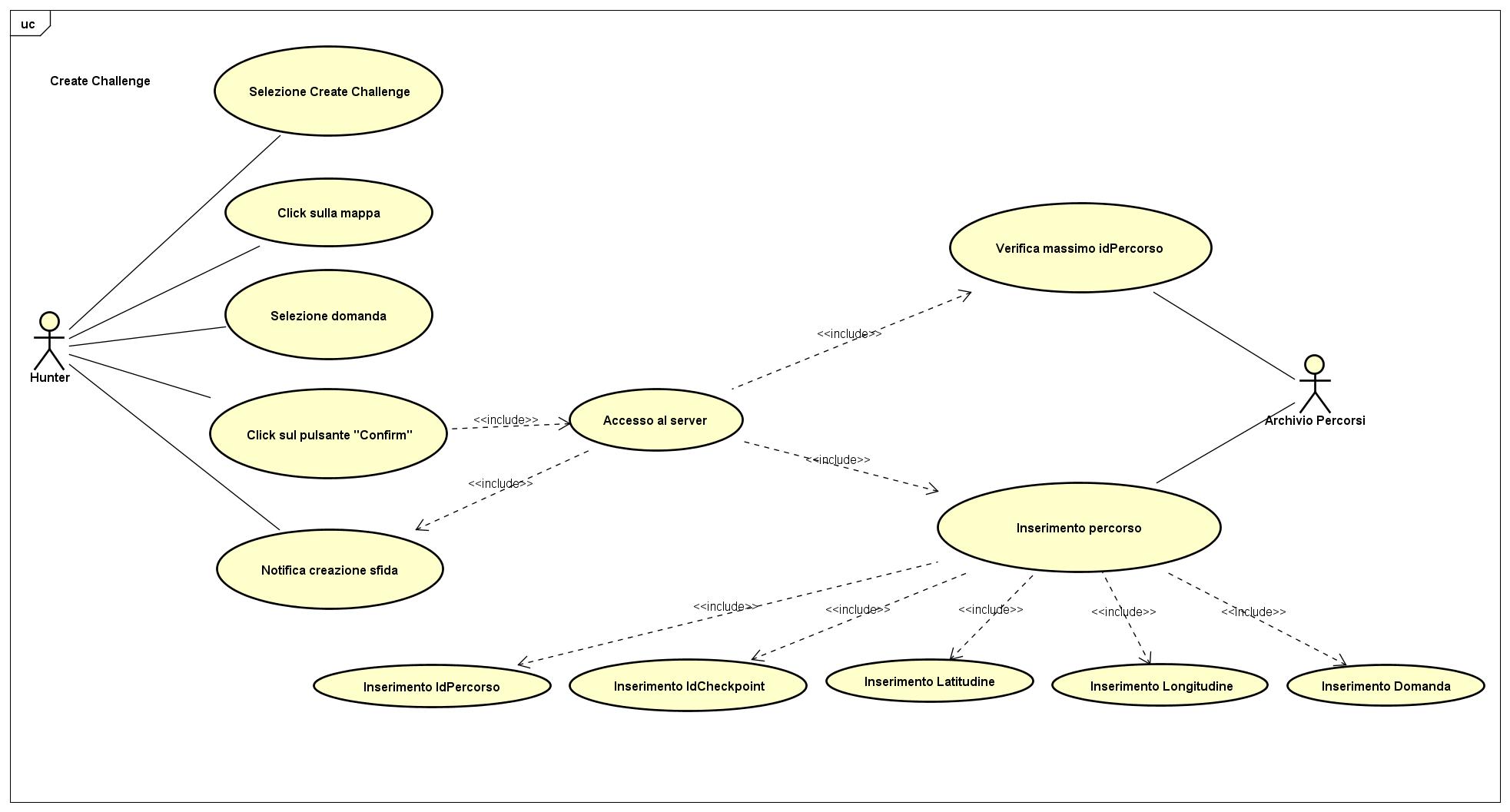
|  |  |
| --- | --- |
| **Caso d’uso: UC-04** | **Giocare una Sfida** |
| **ATTORI** | * Utente |
| **INPUT** |  |
| **PRECONDIZIONI** | * L’utente abbia effettuato l’accesso all’applicazione * L’utente sia stato geolocalizzato * La connessione a Internet sia attiva |
| **OUTPUT** | * Inserimento sul DB del tempo della sfida |
| **POSTCONDIZIONI** | * Il tempo impiegato a risolvere la sfida viene memorizzato nel database |
| **SCENARIO PRIMARIO** | 1. L’utente accede all’applicazione (vedi **UC-02**) 2. L’utente viene geolocalizzato (vedi **UC-03**) 3. L’utente seleziona la voce “Play Challenge” dal menù 4. L’utente clicca su un marker presente sulla mappa che sarà il primo checkpoint della sfida scelta 5. L’utente si trova in una nuova actvity Challenge 6. L’utente clicca sul marker visibile 7. L’utente si trova in una nuova activity Answer. 8. L’utente risponde alla domanda collegata a quel marker 9. Se l’utente risponde correttamente torna all’activity Challenge. Se risponde erroneamente torna all’activity Challenge con una penalità sul tempo impiegato per completare la sfida aumentato di 30 secondi. 10. Il marker visibile cambia mostrando all’utente il successivo punto della sfida 11. Se il marker non è l’ultimo della sfida l’utente ritorna al punto 6 altrimenti va al punto 12 12. Il tempo impiegato per risolvere la sfida viene memorizzato nel Database 13. L’utente termina la sfida e torna all’activity Main e gli viene mostrato il tempo finale impiegato 14. Il caso d’uso termina |

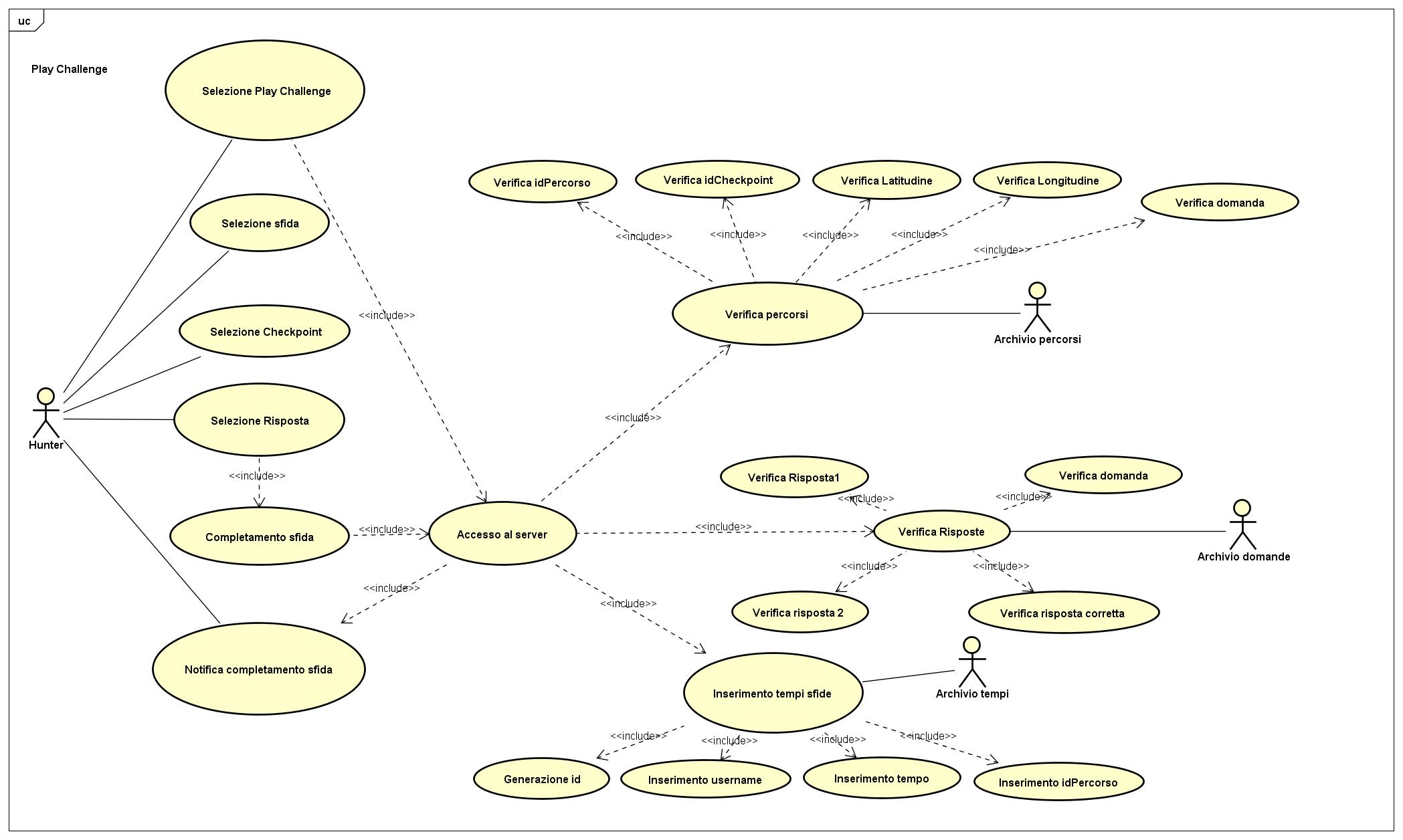
|  |  |
| --- | --- |
| **SCENARIO ALTERNATIVO A** | **Annullamento durante la creazione**   1. L’utente clicca sul tasto “Back” nell’activity StartChallenge annulla la sfida tornando all’activity Main 2. Il caso d’uso termina |
| **SCENARIO ALTERNATIVO B** | **Annullamento durante la creazione**   1. L’utente clicca sul tasto “Back” nell’activity Challenge annulla la sfida tornando all’activity Main 2. Il caso d’uso termina |

### 6.1 Use Case Diagram UML





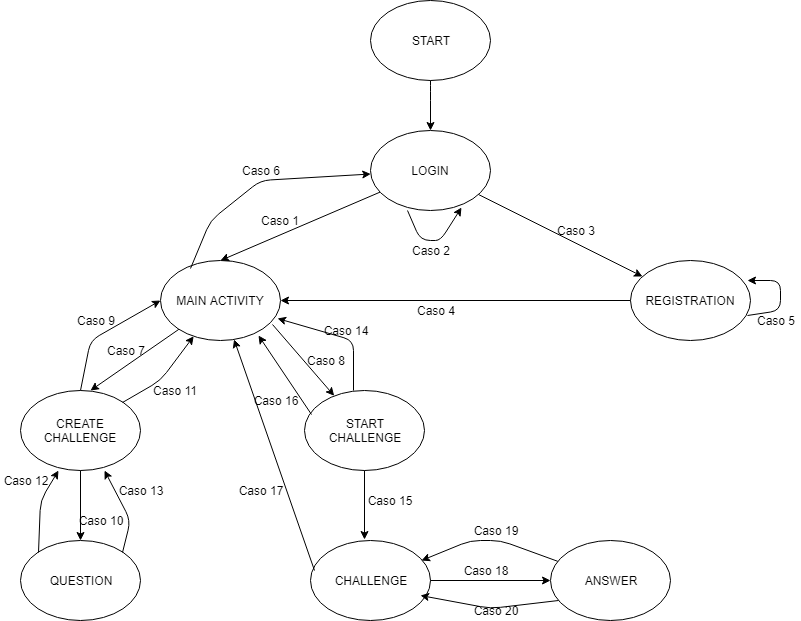




## 7 Design architetturale

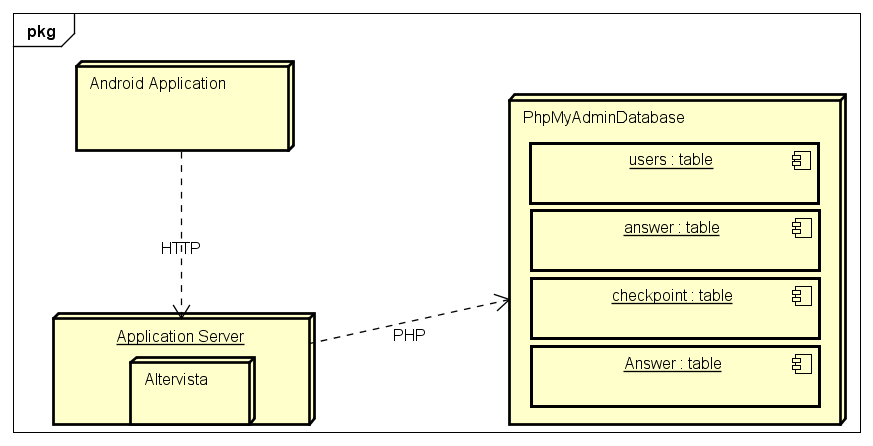
### 7.1 Automa a stati finiti

Dopo aver elencato i vari casi d’uso è stato ritenuto necessario realizzare un automa a stati finiti per descrivere tutti i possibili comportamenti dell’applicazione. I tondini rappresentano le varie Activity e le frecce rappresentano come l’app si muove da una activity all’altra. Su ogni freccia è indicato il numero di un caso.



|  |  |
| --- | --- |
| **Elenco Activity:** | **Elenco casi:** |
| LOGIN | Caso1: Login effettuato |
| LOGIN | Caso2: Login fallito |
| LOGIN | Caso3: Registrazione |
| REGISTRATION | Caso4: Registrazione effettuata |
| REGISTRATION | Caso5: Registrazione fallita |
| MAIN ACTIVITY | Caso6: Logout |
| MAIN ACTIVITY | Caso7: Creare una nuova sfida e andare su Create Challenge |
| MAIN ACTIVITY | Caso8: Giocare ad una sfida e andare su Start Challenge |
| CREATE CHALLENGE | Caso9: Annullare la creazione di una sfida |
| CREATE CHALLENGE | Caso10: Inserire un indovinello e quindi aprire l’activity Question |
| CREATE CHALLENGE | Caso11: Concludere la creazione e tornare sulla Main Activity |
| QUESTION | Caso12: Premere su OK e inserire la domanda |
| QUESTION | Caso13:Premere su CANCEL e tornare su Create Challenge |
| STAR CHALLENGE | Caso14:Premere back e tornare alla main activity |
| STAR CHALLENGE | Caso15:premere sul marker per scegliere la sfida da iniziare |
| STAR CHALLENGE | Caso16:la sfida è terminata e si torna sulla Main |
| CHALLENGE | Caso17:Premere back e uscire dalla sfida senza terminarla |
| CHALLENGE | Caso18:Rispondere alla domanda |
| ANSWER | Caso19:Risposta giusta e ritorna alla sfida |
| ANSWER | Caso20: Risposta errata e ritorna sulla sfida |

### 7.2 Deployment Diagram



### 7.3 Sequence Diagram

## 8.Testing

### 8.1 Black Box Testing

Il testing black box è stato effettuato creando per ogni Activity un caso di Test e simulando il comportamento dell’applicazione in risposta agli input dell’utente che sono stati automatizzati utilizzando la libreria Espresso.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Registrazione | | | | | | | |
| **Choice** | **Caso d’uso** | **Input parameters** | | | **Enviroment characteristics** |  | |
|  | | **Email** | **Username** | **Password** | **GPS/Network** | **Valore atteso** | **Output** |
| #1 | **UC-01.C** | “salvo” | “salvo” | “salvo” | Attivo/Attivo | “Dati inseriti non validi” | “Dati inseriti non validi” |
| #2 | **UC-01.B** | “[salvo@s](mailto:walter@gmail.com)” | “salvo” | “salvo” | Attivo/Attivo | “Email o username esistente” | “Email o username esistente” |
| #3 | **UC-01.A** | “[salvo@s](mailto:walter@gmail.com)” | “null” | “salvo” | Attivo/Attivo | “Dati inseriti non validi” | “Dati inseriti non validi” |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Login | | | | | | |
| **Choice** | **Caso d’uso** | **Input parameters** | | Enviroment characteristics | | |
|  | | **Username** | **Password** | **GPS/Network** | **Valore atteso** | **Output** |
| #1 | **UC-02.B** | “salv” | “salvo” | Attivo/Attivo | “Username o password errate” | “Username o password errate” |
| #2 | **UC-02.A** | “salv” | “null” | Attivo/Attivo | “Dati inseriti non validi” | “Dati inseriti non validi” |
| #3 | **UC-02** | “salvo” | “salvo” | Attivo/Attivo | “Login avvenuto con successo” | “Login avvenuto con successo” |

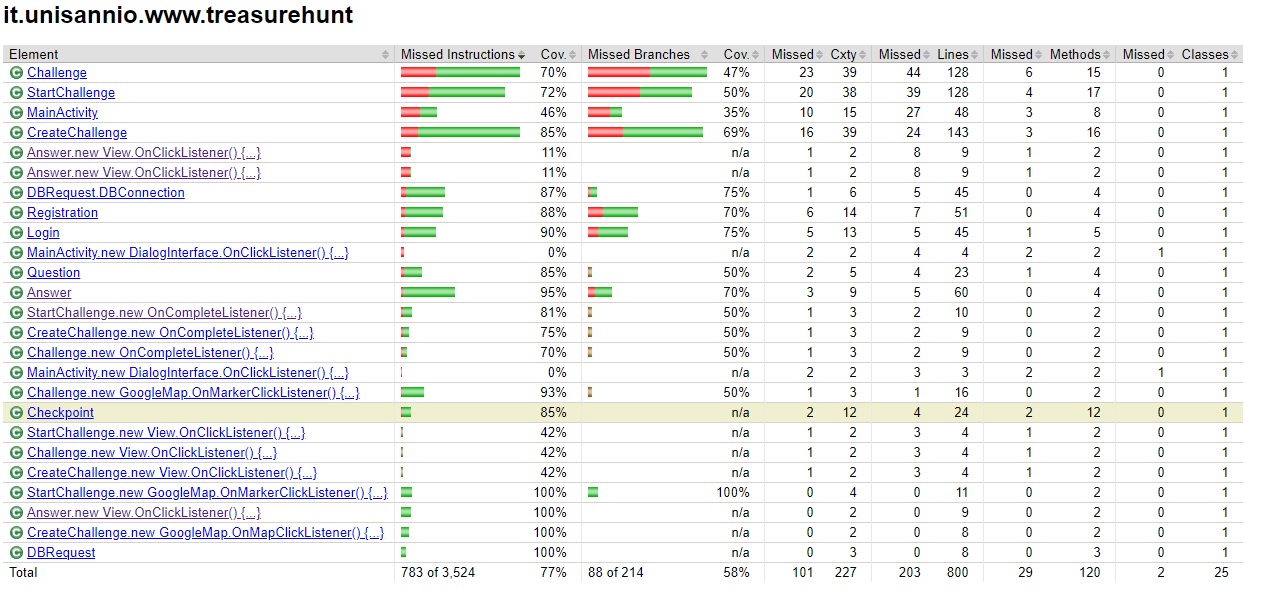
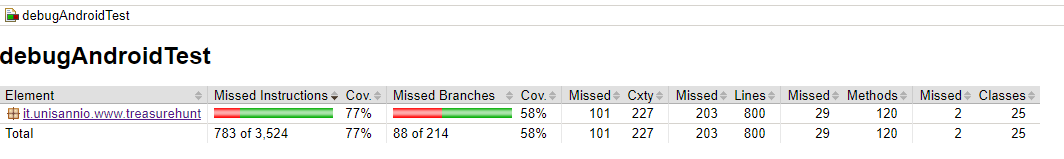
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Creazione di una sfida | | | | UC-03 |
| **#Operazioni** | **Azioni** | **GPS/Network** | Valore atteso | **Output** |
| #1 | Effettua il Login | Attivo/Attivo | Apertura dell’activity principale | Apertura dell’activity principale |
| #2 | Clicca sul button Create Challenge | Attivo/Attivo | Apertura della mappa per creare una sfida | Apertura della mappa per creare una sfida |
| #3 | Clicca su un punto della mappa | Attivo/Attivo | Apertura dell’activity che permette di scegliere la domanda | Apertura dell’activity che permette di scegliere la domanda |
| #4 | Clicca sull’elenco delle domande | Attivo/Attivo | Apertura menu a tendina con elenco domande | Apertura menu a tendina con elenco domande |
| #5 | Clicca sulla settima domanda | Attivo/Attivo | Seleziona la domanda “Qual è il logo della Apple” alla posizione 6 | Seleziona la domanda “Qual è il logo della Apple” |
| #6 | Preme sul bottone OK | Attivo/Attivo | Ritorna alla mappa della creazione sfida | Ritorna alla mappa della creazione sfida |
| #7 | Preme sul bottone CONFIRM | Attivo/Attivo | “Creazione sfida avvenuta con successo” | “Creazione sfida avvenuta con successo” |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Giocare una sfida | | | | UC-04 |
| **#Operazioni** | **Azione** | **GPS/Network** | Valore atteso | Output |
| #1 | Effettua il Login | Attivo/Attivo | Apertura dell’activity principale | Apertura dell’activity principale |
| #2 | Clicca sul button Play Challenge | Attivo/Attivo | Visualizzazione delle sfide disponibili | Visualizzazione delle sfide disponibili |
| #3 | Clicca su uno dei marker per iniziare la sfida | Attivo/Attivo | Apertura dell’activity di sfida | Apertura dell’activity di sfida |
| #4 | Clicca sul marker che indica l’indovinello del primo checkpoint | Attivo/Attivo | Visualizzazione della domanda e delle varie risposte | Visualizzazione della domanda e delle varie risposte |
| #5 | Clicca sulla risposta corretta “Mela” | Attivo/Attivo | Ritorna sull’activity della sfida | Ritorna sull’activity della sfida |
| #6 | Clicca sul marker che indica l’indovinello del secondo checkpoint | Attivo/Attivo | Visualizzazione della domanda e delle varie risposte | Visualizzazione della domanda e delle varie risposte |
| #7 | Clicca sulla risposta corretta “Orwell” | Attivo/Attivo | “HAI COMPLETATO LA SFIDA” | “HAI COMPLETATO LA SFIDA” |

### 8.2 Coverage Test

Un’altra fase del testing è stata quella di Coverage Testing realizzata tramite il tool Jacoco. Questo tool ci ha un report con la percentuale di codice coperta dai test.

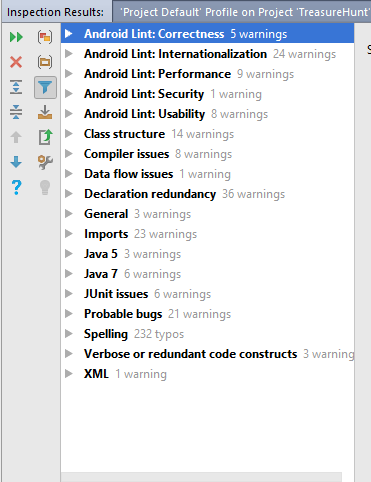
Nella prima figura viene mostrato la percentuale complessiva di codice coperto dai test relativa a tutto il pakage. Nella seconda immagine la percentuale di cui sopra viene mostrata in dettaglio per activity, e relativi metodi, implementata.



### 8.3 Android Lint

Android Studio usa Lint come strumento per rilevare problemi strutturali nelle applicazioni, questo è incluso nella cartella tools dell’Android SDK e deve essere lanciato specificando la directory del progetto.

L’output di Lint fornisce informazioni sui vari problemi riscontrati classificandoli come Error o Warning. La descrizione di ogni aspetto rilevato include il file di collocazione (completo di percorso all’interno del progetto), numero di riga, una breve spiegazione ed un’etichetta che lo classifica.



Il primo report ottenuto con Lint riportava un ‘error’ che è stato subito aggiustato. Nel report seguente che è quello riportato in figura l’ispezione del codice ha riportato diversi warning presenti in vari livelli dell’applicazione che possono essere d’aiuto nel generare un progetto più sicuro.