# 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”.

L’applicazione contiene anche una componente social che consiste nel confronto dei tempi di risoluzione dei percorsi tra più hunters, permettendo così agli utilizzatori dell’app di sfidarsi tra di loro e confrontare le proprie capacità fisiche ed intellettive.

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.

## 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. Il ciclista inserisce le proprie credenziali: 2. Email 3. Username 4. Password 5. Il sistema verifica che il form sia riempito in tutti i suoi campi 6. Il sistema verifica che nel campo mail sia presente in carattere ‘@’ 7. Il sistema verifica che l’email non sia presente nell’archivio 8. Il sistema verifica che l’username non sia presenta nell’archivio 9. Il sistema informa il ciclista della corretta registrazione 10. Il caso d’uso termina |
| **SCENARIO ALTERNATIVO** | **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 1 |
| **SCENARIO ALTERNATIVO** | **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 1 |
| **SCENARIO ALTERNATIVO** | **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 1 |
| **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** | **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** | **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** | **Acquisizione della geoposizione** |
| **ATTORI** | * Hunter |
| **INPUT** | * Google Maps |
| **PRECONDIZIONI** | * L’hunter abbia effettuato l’accesso all’applicazione * Il GPS sia attivo * La connessione a Internet sia attiva |
| **OUTPUT** | * All’hunter viene notificato della sua posizione sulla mappa |
| **POSTCONDIZIONI** | * Registrazione di una nuova geoposizione all’interno dell’archivio |
| **SCENARIO PRIMARIO** | 1. L’hunter accede all’applicazione 2. L’hunter vede un marker relativo alla sua posizione sulla mappa 3. Il caso d’uso termina |
| **SCENARIO ALTERNATIVO** | **Il GPS non è stato attivato**   1. L’hunter non ha attivato il GPS 2. L’hunter viene informato che il GPS non è attivo 3. Il caso d’uso termina |
| **Caso d’uso: UC-04** | **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** | **Annullamento durante la creazione**   1. L’utente clicca sul tasto “back” 2. Il caso d’uso termina |

CATEGORY PARTITION

### 6.1 Use Case Diagram UML



