

UniBoQuest

Relazione di Progetto

Alessandro Chierici (Matricola: 0001126326)
Lorenzo Marchegiani (Matricola: 0001126380)

Corso di Laurea in Ingegneria e Scienze Informatiche
Tecnologie Web – Anno Accademico 2025/2026

Indice

1	Introduzione	3
2	Analisi dei requisiti	3
2.1	Obiettivi	3
2.2	Funzionalità richieste	3
2.3	Requisiti non funzionali	3
3	Fondamenti teorici	4
4	Progettazione e architettura del sistema	4
4.1	Frontend	4
4.2	Backend	4
4.3	Database e modello dati	4
5	Design e interfaccia	5
5.1	Design visivo	5
5.2	Interfaccia utente	5
6	Scelte implementative	5
6.1	Tecnologie utilizzate	5
6.2	Motivazioni delle scelte	5
7	Implementazione	6
7.1	Flusso principale utente	6
7.2	Sistema missioni e progressi	6
7.3	Ruolo admin e gestione missioni	6
8	Utenti e Personas	6
9	Testing e valutazione	7
10	Conclusioni e sviluppi futuri	7

1 Introduzione

UniBoQuest è una piattaforma web che applica meccaniche di *gamification* al contesto universitario, con l'obiettivo di rendere più coinvolgente e strutturata l'esperienza dello studente all'interno del campus.

L'idea alla base del progetto è reinterpretare alcune attività tipiche della vita universitaria (partecipazione a eventi, sessioni di studio, orientamento e vita sociale) come **missioni** di un videogioco, premiando l'utente con **punti esperienza (XP)**, **livelli** e **badge**. Attività spesso percepite come dispersive vengono così presentate come obiettivi chiari e misurabili.

Il progetto è stato sviluppato rispettando i vincoli del corso: utilizzo di **PHP** lato server e tecnologie web standard, con attenzione a responsività e accessibilità.

2 Analisi dei requisiti

2.1 Obiettivi

Gli obiettivi principali di UniBoQuest sono:

- aumentare il coinvolgimento degli studenti nelle attività universitarie;
- fornire uno strumento semplice per orientarsi tra missioni e progressi personali;
- dimostrare l'efficacia di tecniche di gamification in un contesto non ludico;
- realizzare un'interfaccia coerente e utilizzabile sia su desktop che su dispositivi mobili.

2.2 Funzionalità richieste

Le funzionalità implementate e/o previste includono:

- registrazione e autenticazione utenti (sessioni);
- creazione e modifica del profilo (nickname, nome visualizzato, campus, preferenze, avatar, privacy);
- dashboard personale con livello, XP e missioni in corso;
- catalogo missioni e pagina dettaglio missione con azioni (partecipa / completa);
- check-in dimostrativo per missioni evento (QR/codice fallback);
- gestione missioni da parte dell'**admin** (CRUD: crea, aggiorna, cancella).

2.3 Requisiti non funzionali

- **Responsività**: layout adattivo tramite Bootstrap e CSS custom.
- **Accessibilità**: struttura semantica, *skip-link*, focus visibile, testi leggibili.
- **Sicurezza di base**: password hash, query parametrizzate, controlli di sessione e ruolo.

3 Fondamenti teorici

Le scelte progettuali di UniBoQuest si basano su principi noti di motivazione e design dell'interazione:

- **Teoria dell'Aspettativa (Vroom)**: il legame tra impegno e ricompensa (XP, livelli) incentiva la partecipazione.
- **Teoria dell'Autodeterminazione (Deci & Ryan)**: la scelta delle missioni supporta autonomia; la progressione visibile rafforza la competenza.
- **Riduzione del carico cognitivo**: missioni brevi e ben definite facilitano comprensione e onboarding.

4 Progettazione e architettura del sistema

UniBoQuest è strutturato come applicazione web con separazione logica tra frontend e backend, mantenendo un'organizzazione modulare.

4.1 Frontend

Il frontend utilizza:

- **HTML5** semantico (header, main, footer);
- **CSS3** e **Bootstrap 5** per layout e responsività;
- **JavaScript Vanilla** per interazioni mirate (es. check-in, manuale, micro-interazioni).

4.2 Backend

Il backend è sviluppato in **PHP** e gestisce:

- autenticazione e sessioni;
- guardie di accesso (utente loggato / profilo completato / ruolo admin);
- persistenza dati tramite **MySQL** (utenti, profili, missioni, progressi).

4.3 Database e modello dati

Il database MySQL (schema importabile via `schema.sql`) include almeno le seguenti tabelle:

- **users**: credenziali e ruolo (`role = user` o `admin`);
- **profiles**: dati profilo utente, preferenze, avatar, privacy;
- **missions**: anagrafica missioni (categoria, difficoltà, XP, check-in, attivazione, sort order);
- **user_missions**: relazione tra utente e missione con stato (`active/completed`).

Questo modello consente:

- catalogo missioni comune per tutti;
- progressi personalizzati per utente;
- calcolo XP totale e livello a partire dalle missioni completate.

5 Design e interfaccia

5.1 Design visivo

UniBoQuest adotta uno stile **retro-gaming** ispirato agli anni '80:

- font a tema (*Press Start 2P* per titoli e *Share Tech Mono* per testi);
- estetica “pixel/CRT” dosata per non compromettere leggibilità e usabilità;
- coerenza tra pagine pubbliche e area privata, con elementi condivisi (header/footer).

5.2 Interfaccia utente

L'interfaccia è pensata per guidare l'utente:

- onboarding tramite registrazione + completamento profilo;
- dashboard con informazioni immediate (livello, XP, missioni attive);
- percorsi chiari verso missioni e check-in;
- per l'admin, una pagina dedicata (`admin_missions.php`) con form CRUD.

6 Scelte implementative

6.1 Tecnologie utilizzate

- HTML5
- CSS3 + Bootstrap 5.3.x
- JavaScript Vanilla
- PHP 8.x
- MySQL (phpMyAdmin / XAMPP per ambiente locale)

6.2 Motivazioni delle scelte

- Bootstrap accelera la responsività senza introdurre framework complessi.
- JavaScript vanilla mantiene il progetto leggibile e adatto a un contesto didattico.
- MySQL rende persistenti missioni e progressi in modo robusto (rispetto a soluzioni solo client-side).
- Query parametrizzate e password hash migliorano la sicurezza base.

7 Implementazione

7.1 Flusso principale utente

1. Registrazione e login.
2. Se profilo mancante: redirect su `edit_profile.php`.
3. Completamento profilo (nickname, avatar, preferenze, ecc.).
4. Accesso a `dashboard.php` con livello, XP e missioni.
5. Navigazione su missioni, dettaglio missione e completamento.

7.2 Sistema missioni e progressi

Le missioni sono definite in `missions` e rese disponibili agli utenti. Quando un utente partecipa a una missione, viene creata una riga in `user_missions` con stato `active`. Al completamento, lo stato passa a `completed` e l'XP viene conteggiato nel totale utente. Il livello è calcolato da soglie cumulative (ad es. 100 XP per livello).

7.3 Ruolo admin e gestione missioni

È stato introdotto un ruolo `admin` nella tabella `users`. Un utente admin visualizza la voce **ADMIN** nella navbar e può accedere a `admin_missions.php`, dove può:

- **creare** nuove missioni;
- **aggiornare** missioni esistenti;
- **cancellare** missioni (“cancella”).

Queste operazioni agiscono direttamente sulla tabella `missions`. Di conseguenza, una missione creata diventa disponibile anche agli utenti standard (a parità di condizioni, es. `active=1`).

8 Utenti e Personas

Durante la progettazione sono state individuate alcune tipologie di utenti:

- **Studente al primo anno:** usa spesso lo smartphone, cerca una guida iniziale (onboarding chiaro) e obiettivi semplici.
- **Studente attivo:** già coinvolto in attività extracurricolari, vuole monitorare progressi (XP/livelli) e avere feedback immediato.
- **Studente di psicologia (focus questionari):** alcuni studenti di psicologia che hanno testato il prototipo hanno evidenziato che una piattaforma come UniBoQuest potrebbe essere davvero utile per **somministrare questionari** a fini didattici (tesi, laboratori, raccolta dati), integrandoli come missioni con ricompense e tracciamento.

Queste personas hanno guidato le scelte su interfaccia, chiarezza dei passi iniziali e coerenza dell'esperienza su mobile.

9 Testing e valutazione

Il prototipo è stato testato in modo informale da studenti esterni al progetto. I tester hanno:

- compreso rapidamente l'idea generale di missioni e progressione;
- apprezzato la dashboard come punto di partenza e il manuale/onboarding;
- suggerito di rendere evidenti i prossimi passi (es. missioni consigliate, pulsanti chiari).

Il feedback è stato utilizzato per migliorare navigazione, coerenza delle pagine private e gestione degli errori (es. link missioni/dettaglio, guardie di accesso, percorsi di ritorno).

10 Conclusioni e sviluppi futuri

UniBoQuest rappresenta un prototipo coerente che dimostra come la gamification possa essere applicata al contesto universitario in modo chiaro e motivante, mantenendo attenzione a responsabilità, accessibilità e struttura del codice.

Possibili sviluppi futuri:

- integrazione reale di check-in (QR effettivo) e validazione lato server;
- missioni dinamiche e personalizzazione avanzata in base al profilo;
- estensione del modulo **questionari** (per studenti di psicologia e contesti didattici), con form compilabili e raccolta dati;
- strumenti admin aggiuntivi (gestione utenti, moderazione, statistiche).
- Un'ulteriore estensione interessante riguarda la gestione dei ruoli. Oltre alla distinzione attuale tra *user* e *admin*, si potrebbe introdurre una figura **intermedia** (ad esempio *teacher* o *moderator*), pensata per docenti o responsabili di attività didattiche. Questo ruolo avrebbe permessi limitati rispetto all'admin, ma sufficienti per **creare e pubblicare missioni** legate a corsi, laboratori o iniziative accademiche, inclusa la **somministrazione di questionari** per ricerche, raccolta dati o attività di tesi. In questo scenario UniBoQuest diventerebbe non solo uno strumento di gamification per studenti, ma anche una piattaforma a supporto di processi didattici e sperimentali, mantenendo un controllo centralizzato dell'admin sulle funzionalità critiche (gestione utenti, moderazione e configurazioni globali).