

Gestione Collaborativa Unità Abitative

1. Visione e Obiettivi

L'applicazione nasce per risolvere le criticità tipiche della convivenza studentesca. L'obiettivo è centralizzare in un'unica piattaforma la contabilità delle spese, la turnazione delle pulizie e la ricerca di nuovi inquilini.

2. Architettura e Stack Tecnologico

| Componente | Tecnologia | Ruolo |
|------------|-------------------|--|
| Backend | PHP (Native) | Gestione sessioni e comunicazioni con il DB. |
| Frontend | HTML5, JS Vanilla | Interfaccia dinamica e manipolazione del DOM per aggiornamenti asincroni. |
| Styling | CSS3 / Bootstrap | Design responsivo con approccio Mobile-First per l'utilizzo su smartphone. |
| Database | MySQL | Gestione dei dati per utenti, unità abitative, spese e turni. |

3. Funzionalità Implementative

3.1 Gestione Finanziaria e "Smart Split"

Il sistema integra un modulo per il calcolo automatico delle quote pro-capite. Inserendo il totale di una bolletta o di una spesa comune, l'algoritmo ripartisce l'importo tra i membri dell'unità, visualizzando in tempo reale il saldo debiti/crediti nella dashboard utente.

3.2 Modulo Collaborazione e Gamification

Per incentivare la partecipazione attiva, i turni di pulizia e le attività comuni sono gestiti in modo dinamico.

- **Tracking:** Monitoraggio dello stato dei task (Completato/In attesa).
- **Gamification:** Possibilità di implementare un sistema di badge o punteggi per premiare i coinvilgimenti più puntuali, trasformando la gestione domestica in un'esperienza coinvolgente.

3.3 Marketplace e Matching Annunci

L'applicazione funge da ponte tra residenti e potenziali inquilini.

- **Annunci Pubblici:** Visibilità esterna per utenti non registrati.
- **Tag Lifestyle:** Utilizzo di metadati (es. preferenze fumatori, animali) per migliorare il matching e la compatibilità tra i coinvilgimenti.

4. Accessibilità e UI/UX

L'interfaccia è progettata seguendo il paradigma **Mobile-First**.

- **Feedback Utente:** Utilizzo di notifiche non invasive (Toast) per confermare l'avvenuta registrazione, modifiche o cancellazioni.

5. Struttura dei Dati e Sicurezza

L'architettura del database è organizzata per garantire l'integrità dei dati tra le diverse entità:

- **Ruoli Gerarchici:** Distinzione tra Super Admin (moderazione globale), Admin Interno (gestione casa) e Studente ed utente non registrato.
- **Sicurezza:** Gestione delle sessioni e protezione degli accessi alle aree private delle unità abitative attraverso controlli lato server in PHP.

6. Scenari d'Uso (Personas)

1. **Ricerca:** L'utente non registrato consulta il forum, filtra per zona e contatta l'unità tramite form dedicato.
2. **Operatività Quotidiana:** Lo studente accede alla dashboard per verificare quale turno di pulizia gli spetti e aggiornare la lista della spesa.
3. **Coordinamento:** L'amministratore interno pubblica l'annuncio per una stanza vacante e monitora le scadenze delle utenze imminenti.