

Bachelor's Degree in Computer Engineering and Science
A.A. 2025/26

WEB25

Grazia Bochdanovits de Kavna

matr. 0001117082

Alessandro Rebosio

matr. 0001130557

Contents

1	Analisi	2
1.1	Descrizione	2
1.2	Analisi dei requisiti	2
1.2.1	Requisiti funzionali	2
1.2.2	Requisiti non funzionali	3
2	Scenari d'uso e Personas	4
2.1	Attori e casi d'uso	4
2.2	Personas	6
3	Mockup dell'applicazione	7
4	Ambiente di sviluppo	8
4.1	Tecnologie utilizzate	8
4.2	Stack di sviluppo	8
A	Guida Utente	9
A.1	Requisiti di sistema	9
A.2	Installazione e Setup	9
A.2.1	Passo 1: Configurare le credenziali del database	9
A.2.2	Passo 2: Importare lo schema e i dati di esempio	9
A.2.3	Passo 3: Avviare il server PHP	10
A.3	Accesso all'applicazione	10

Chapter 1

Analisi

1.1 Descrizione

UniMatch è un'applicazione web progettata per facilitare la connessione tra studenti universitari in cerca di compagni per progetti, studio di gruppo o preparazione agli esami. Il progetto nasce dall'esigenza di trovare partner affidabili con interessi e competenze simili, cosa spesso difficile nelle università con numerosi corsi e studenti. L'app adotta un approccio simile a un social network: gli studenti creano un account e possono proporre progetti indicando corso di studio, numero di collaboratori e competenze richieste. Gli altri utenti possono interagire con i post, e l'autore può contattare direttamente chi mostra interesse. In questo modo, UniMatch semplifica la ricerca di partner di studio compatibili e favorisce la collaborazione accademica.

1.2 Analisi dei requisiti

1.2.1 Requisiti funzionali

Utente Studente

- **Registrazione e login:** possibilità di creare un account personale o accedere con credenziali già registrate.
- **Creazione e pubblicazione di post:** possibilità di proporre progetti indicando titolo, descrizione, corso di studio, numero di collaboratori richiesti e competenze necessarie.
- **Visualizzazione dei post:** possibilità di visualizzare i post pubblicati da altri utenti nella home.
- **Interazione con i post:** possibilità di mettere "mi piace" o "skip" ai post nella home.
- **Gestione profilo:** visualizzazione e modifica dei dati personali, elenco dei post pubblicati e lista degli utenti che hanno messo "mi piace" ai propri post.
- **Sistema di messaggistica e chat:** possibilità di inviare e ricevere messaggi privati con altri utenti interessati ai propri post.

Utente Admin

- Accesso a tutte le funzionalità dell'utente studente.
- Monitoraggio dei post pubblicati da tutti gli utenti.
- Possibilità di gestire contenuti inappropriati o segnalati.

1.2.2 Requisiti non funzionali

- **Usabilità:** l'interfaccia deve essere semplice, intuitiva e facilmente navigabile.
- **Accessibilità:** l'applicazione deve essere accessibile a utenti con disabilità, rispettando le linee guida WCAG.
- **Prestazioni:** l'applicazione deve garantire tempi di risposta rapidi durante la visualizzazione dei post e l'invio/ricezione dei messaggi.
- **Compatibilità:** l'applicazione deve essere fruibile su diversi browser web e dispositivi, inclusi desktop, smartphone e tablet.
- **Manutenibilità:** il codice deve essere chiaro e modulare per facilitare aggiornamenti e miglioramenti futuri.

Chapter 2

Scenari d'uso e Personas

2.1 Attori e casi d'uso

Attori

- **Utente Studente:** utilizza l'applicazione per cercare e proporre progetti, interagire con i post, gestire il proprio profilo e comunicare con altri utenti.
- **Amministratore:** gestisce la piattaforma, monitora i contenuti pubblicati e amministra gli account degli utenti.

Casi d'uso principali

- Registrazione / Login
- Visualizzazione dei post
- Creazione di un post
- Interazione con i post
- Gestione profilo
- Sistema di messaggistica
- Gestione utenti (solo per Admin)
- Monitoraggio contenuti (solo per Admin)

Il diagramma dei casi d'uso riassume le principali funzionalità offerte dall'applicazione e le interazioni tra gli attori e il sistema.



2.2 Personas

Persona 1: Studente in cerca di collaboratori

Nome Marco Rossi

Età 22 anni

Corso di studio Informatica (Triennale)

Marco è uno studente universitario che fatica a trovare compagni affidabili con cui svolgere i progetti assegnati durante il percorso universitario. Attraverso UniMatch, spera di connettersi con altri studenti che condividono i suoi stessi interessi accademici.

Obiettivi: Trovare partner per progetti di gruppo e collaborare con studenti del proprio corso in un ambiente strutturato e organizzato.

Bisogni: Accedere facilmente ai progetti disponibili, contattare rapidamente altri studenti e gestire le comunicazioni all'interno della piattaforma.

Persona 2: Studente organizzatore di progetti

Nome Giulia Bianchi

Età 24 anni

Corso di studio Informatica (Magistrale)

Giulia deve spesso organizzare progetti di gruppo e cerca collaboratori con competenze specifiche. Utilizza UniMatch per pubblicare post dettagliati e selezionare i candidati più adatti.

Obiettivi: Pubblicare progetti chiari e mirati e selezionare collaboratori affidabili.

Persona 3: Amministratore

Nome Luca Verdi

Età 35 anni

Ruolo Amministratore della piattaforma

Luca si occupa della gestione e del corretto funzionamento della piattaforma UniMatch. Controlla i contenuti pubblicati e gestisce eventuali segnalazioni da parte degli utenti.

Obiettivi: Garantire un ambiente sicuro e collaborativo, e gestire utenti e contenuti inappropriati.

Le personas descritte rappresentano i principali profili di utenti dell'applicazione e permettono di comprendere le diverse esigenze che hanno guidato la definizione dei casi d'uso e delle funzionalità del sistema.

Chapter 3

Mockup dell'applicazione

Chapter 4

Ambiente di sviluppo

L'applicazione UniMatch è stata sviluppata come applicazione web basata su architettura client-server. Per lo sviluppo è stato utilizzato lo stack **XAMPP**, che fornisce un ambiente di sviluppo locale completo.

4.1 Tecnologie utilizzate

- **HTML5** e **CSS3**: utilizzati per la struttura e la presentazione dell'interfaccia utente.
- **Bootstrap**: framework CSS utilizzato per semplificare la realizzazione di un'interfaccia responsive.
- **JavaScript**: utilizzato per la gestione delle interazioni lato client.
- **PHP**: utilizzato per l'implementazione della logica applicativa lato server.
- **MySQL**: utilizzato per la gestione e la persistenza dei dati.

4.2 Stack di sviluppo

Lo stack XAMPP include i seguenti componenti principali: **Apache** come web server, **PHP** come linguaggio server-side e **MySQL** come database relazionale.

Appendix A

Guida Utente

A.1 Requisiti di sistema

Per eseguire UniMatch in locale, assicurati di avere i seguenti componenti installati:

- **PHP 8+** con estensione `mysqli`
- **MySQL** o **MariaDB** server
- Un **web browser** moderno (Chrome, Firefox, Safari, Edge, ecc.)

A.2 Installazione e Setup

A.2.1 Passo 1: Configurare le credenziali del database

Apri il file `config/database.php` nella radice del progetto e modifica i seguenti parametri secondo le tue impostazioni locali:

```
<?php
$host = 'localhost';      // Indirizzo del server MySQL
$user = 'root';           // Utente MySQL
$pass = 'password';       // Password MySQL
$dbname = 'unimatch';     // Nome del database
$port = 3306;             // Porta (di default 3306)
```

A.2.2 Passo 2: Importare lo schema e i dati di esempio

Dalla root del progetto, esegui i seguenti comandi per creare il database e popolare i dati:

```
mysql < resources/database.sql
mysql < resources/demo.sql
```

In alternativa, puoi utilizzare phpMyAdmin o il client MySQL per importare i file SQL.

A.2.3 Passo 3: Avviare il server PHP

Dalla root del progetto, avvia il server web integrato di PHP:

```
php -S localhost:8000 -t src/public
```

Questo comando avvierà il server sulla porta 8000, servendo i file della cartella `src/public`.

A.3 Accesso all'applicazione

Una volta che il server è in esecuzione, apri il tuo browser web e accedi all'indirizzo:

```
http://localhost:8000/
```

Verrai reindirizzato alla pagina di login se non hai ancora effettuato l'accesso.