

---

# **Software Requirements Specification**

**for**

## **Online T-Platform**

**Version 1.0 approved**

**Prepared by Giovanni D'Ambrosio**

**Università degli Studi della Campania Luigi Vanvitelli**

**07/06/2023**

# Table of Contents

<b>Table of Contents</b>	<b>0</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
<b>2. Overall Description</b>	<b>1</b>
2.1 Product Perspective	1
2.2 Product Functions	1
2.3 User Classes and Characteristics	2
2.4 Operating Environment	2
2.5 Design and Implementation Constraints	2
2.6 Assumptions and Dependencies	2
<b>3. External Interface Requirements</b>	<b>2</b>
3.1 User Interfaces	2
3.2 Hardware Interfaces	3
3.3 Software Interfaces	3
3.4 Communications Interfaces	3
<b>4. System Features</b>	<b>3</b>
4.1 Gestione Account	5
4.2 Ricerca, offerta e analisi	5
4.3 Gestione Profilo	6
<b>5. Other Nonfunctional Requirements</b>	<b>7</b>
5.1 Security Requirements	7
5.2 Business Rules	7
<b>Appendix A: Glossary</b>	<b>7</b>
<b>Appendix B: Analysis Models</b>	<b>8</b>
<b>Data Dictionary</b>	<b>10</b>
<b>Database Implementation</b>	<b>13</b>

# **1. Introduction**

## **1.1 Purpose**

Questo SRS fornisce le specifiche per Online T-Platform, un'applicazione PHP che permette lo scambio di oggetti tra utenti per promuovere la sostenibilità in Campania. Questa è la versione 1.0 del documento.

## **1.2 Document Conventions**

In questo documento sono usati diagrammi UML per l'analisi e la progettazione in accordo con lo standard UML 2.0.

## **1.3 Intended Audience and Reading Suggestions**

Questo documento ha come destinatari sia i progettisti di sistema, che i clienti committenti del prodotto.

## **1.4 Product Scope**

Online T-Platform ha come obiettivo la promozione della sostenibilità per i cittadini in Campania, offrendo una piattaforma su cui gli utilizzatori possono offrire gli oggetti di cui non hanno più bisogno per scambiarne degli altri con altri utenti registrati al servizio. Attraverso Online T-Platform, gli utilizzatori potranno aiutare la Campania a ridurre gli sprechi dando una nuova vita a ciò che hanno di inutilizzato. Lavorando insieme, ci si può aiutare aiutando.

# **2. Overall Description**

## **2.1 Product Perspective**

Online T-Platform è pensato come un sistema a sé stante, realizzato da zero, utilizzabile dagli utenti finali as-is.

## **2.2 Product Functions**

Il sistema deve consentire all'utente tramite interfaccia web di:

1. Registrarsi ed effettuare il login.
2. Inserire o rimuovere un annuncio relativo ad un prodotto da scambiare.
3. Cercare annunci inseriti da altri utenti tramite parole chiave e con la possibilità di filtrare per categoria.

4. Inviare una richiesta di trattativa ad un altro utente per uno degli oggetti da lui proposti nei suoi annunci.
5. Rispondere alle richieste di trattativa di altri utenti rifiutando l'offerta o inviando una proposta di chiusura trattativa, selezionando uno degli oggetti da loro proposti nei loro annunci.
6. Rifiutare o accettare una proposta di chiusura trattativa per completare uno scambio.
7. Rilasciare una recensione ad un utente con cui è stato concluso uno scambio.
8. Accedere al proprio profilo per visualizzare annunci disponibili, richieste di trattativa inviate e ricevute, proposte di chiusura trattativa inviate e ricevute, scambi completati e recensioni ricevute.
9. Visualizzare le recensioni che un utente ha ricevuto nei suoi scambi passati.

## **2.3 User Classes and Characteristics**

L'applicazione gestisce una singola tipologia di utenti, offrendo loro la possibilità di usufruire dei servizi del sistema tramite interfaccia Web.

## **2.4 Operating Environment**

L'applicazione utilizza PHP come linguaggio server side, HTML, CSS e Javascript come linguaggi client side. Per il deployment viene utilizzato XAMPP, che gestisce le seguenti componenti:

1. MySQL per la gestione della base dati.
2. Apache per la gestione del server Web.
3. PHPMyAdmin per l'interfacciamento visuale alla base dati.

## **2.5 Design and Implementation Constraints**

Il sistema deve supportare il caricamento e il salvataggio di un grande numero di immagini. L'applicazione client-side deve supportare JavaScript come linguaggio utilizzato per rendere l'esperienza utente interattiva.

## **2.6 Assumptions and Dependencies**

Lo sviluppo del sistema deve essere basato su HTML, coadiuvato da CSS per la presentazione grafica.

# **3. External Interface Requirements**

## **3.1 User Interfaces**

L'applicazione espone un sito web accessibile tramite un browser. L'interfaccia è moderna e responsiva.

Attraverso interfaccia grafica, l'utente deve avere la possibilità di accedere al suo profilo, alla pagina di caricamento annunci e alla homepage. L'utente deve avere sempre a portata di click la possibilità di performare una ricerca tramite un'interfaccia non invasiva.

Il sistema vede come colore protagonista delle interfacce il verde, nelle sue diverse sfumature.

### 3.2 Hardware Interfaces

Non ci sono vincoli sulla macchina per l'hosting del sito, esclusa la compatibilità con il software XAMPP.

### 3.3 Software Interfaces

Per la gestione del sistema sono stati utilizzati:

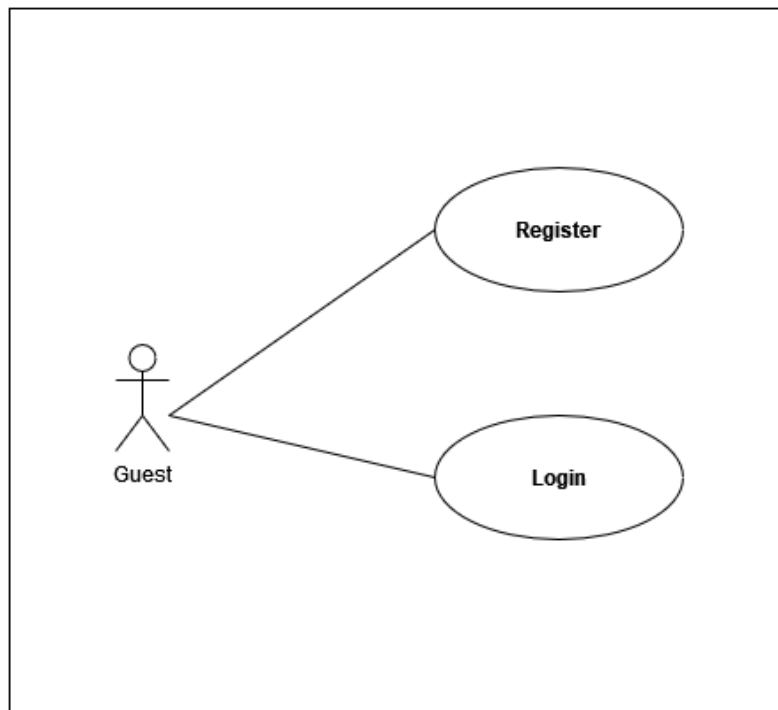
1. Server Web Apache/2.4.58
2. MySQL 10.4.32.0 come DBMS
3. phpMyAdmin 5.2.1 come interfaccia per il DBMS
4. PHP 8.2.12 per lo scripting lato server

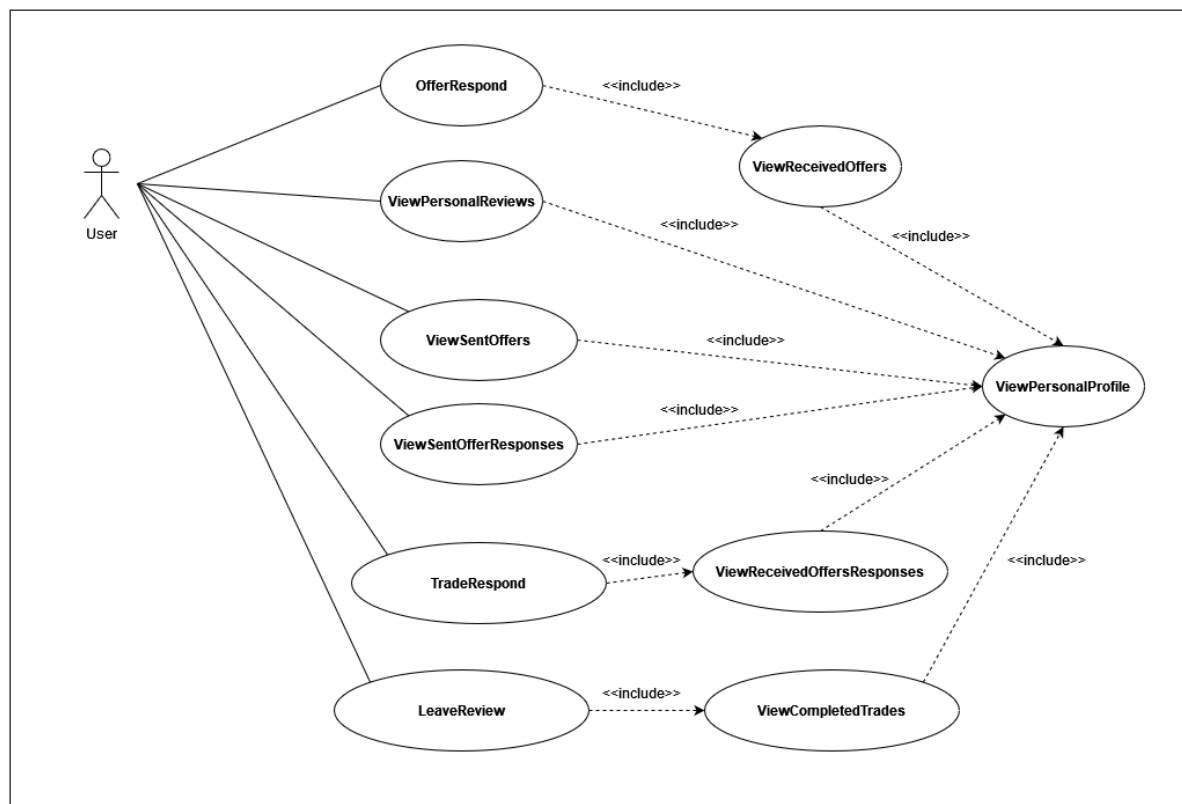
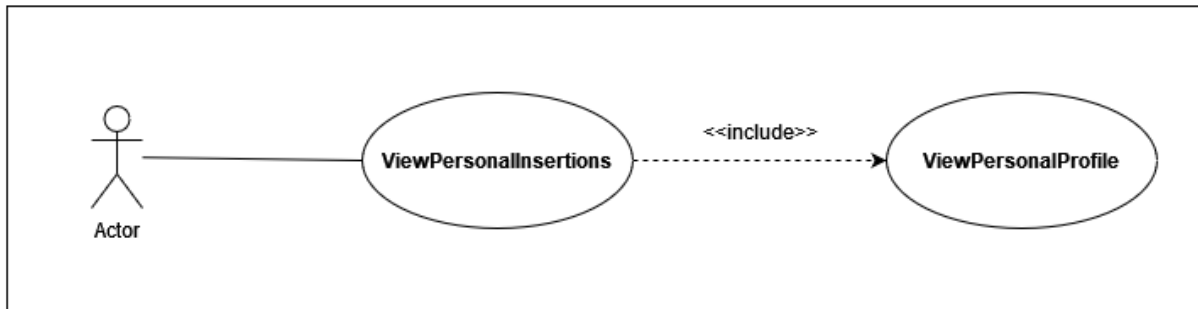
### 3.4 Communications Interfaces

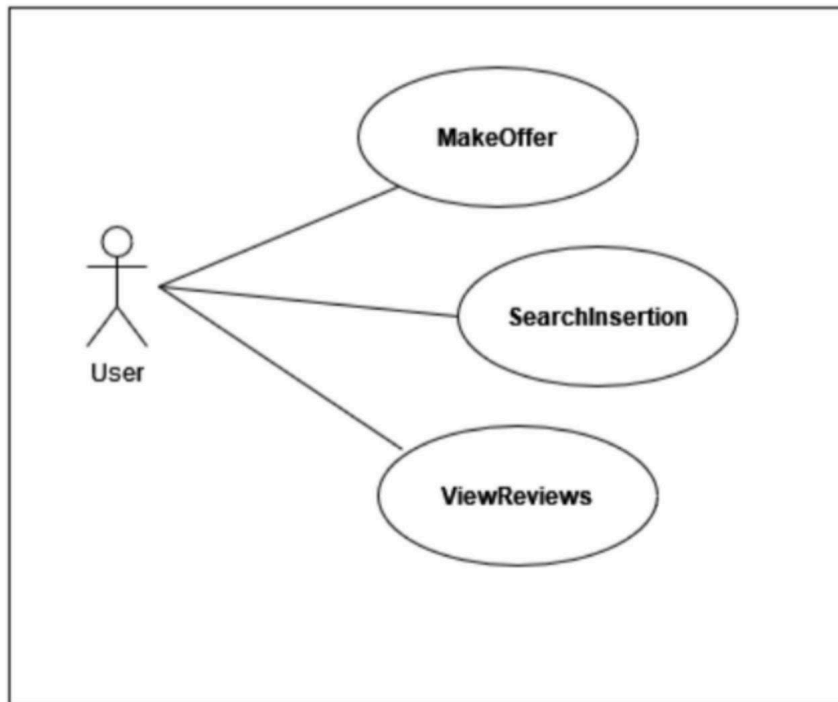
Il sistema utilizza lo standard di comunicazione HTTP, sia per l'elaborazione delle richieste, sia per l'interazione con l'utente finale. Lo scambio di dati e richieste è gestito dai metodi GET e POST del protocollo.

## 4. System Features

*Di seguito sono illustrati i casi d'uso del sistema e una loro descrizione più dettagliata.*







## 4.1 Gestione Account

### 4.1.1 Description and Priority

*L'utente può registrarsi o effettuare l'accesso al sistema.*

### 4.1.2 Stimulus/Response Sequences

*L'utente può performare queste operazioni dalla pagina di benvenuto del sistema.*

### 4.1.3 Functional Requirements

1. Register: L'utente si registra al sistema inserendo il proprio nome, cognome, un indirizzo email valido tra i domini accettati, un username tra gli 8 e i 30 caratteri e una password che rispetta i requisiti di sicurezza.
2. Login: L'utente fa l'accesso al sistema inserendo il proprio username e la propria password.

## 4.2 Ricerca, offerta e analisi

### 4.2.1 Description and Priority

*L'utente può ricercare prodotti a cui è interessato, visualizzare gli annunci degli utenti, inviare richieste di trattativa per gli oggetti di interesse e visualizzare le valutazioni che un utente ha ricevuto da altri utenti.*

### 4.2.2 Stimulus/Response Sequences

*L'utente può performare queste operazioni dalla homepage e dalle pagine di visualizzazione degli annunci.*

#### 4.2.3 Functional Requirements

1. SearchInsertion: l'utente effettua una ricerca per parola chiave, con la possibilità di filtrare i risultati per categoria.
2. MakeOffer: l'utente invia una richiesta di trattativa dalla pagina di visualizzazione dell'annuncio a cui è interessato.
3. ViewReviews: l'utente visualizza le recensioni di un utente ricevute da altri utenti con valutazione da 1 a 5 stelle.

### 4.3 Gestione Profilo

#### 4.3.1 Description and Priority

*L'utente può accedere al proprio profilo e visualizzare i propri annunci, le proprie recensioni, le richieste di trattativa inviate e ricevute, le proposte di chiusura trattativa inviate e ricevute, gli scambi completati. Se non già inviata, può lasciare una recensione agli utenti con cui ha completato uno scambio.*

#### 4.3.2 Stimulus/Response Sequences

*L'utente può performare queste operazioni dalla pagina del suo profilo personale.*

#### 4.3.3 Functional Requirements

1. ViewPersonalProfile: L'utente accede al proprio profilo, da cui può visualizzare le sue informazioni personali.
2. ViewPersonalInsertions: L'utente visualizza i propri annunci caricati e visibili.
3. ViewSentOffers: L'utente visualizza le richieste di trattativa da lui inviate agli altri utenti.
4. ViewReceivedOffers: L'utente visualizza le richieste di trattativa ricevute dagli altri utenti.
5. OfferRespond: L'utente risponde ad una richiesta di trattativa ricevuta da un altro utente, rifiutando oppure inviando una proposta di chiusura trattativa, scegliendo uno degli annunci messi a disposizione dall'offerente.
6. ViewSentOfferResponses: L'utente visualizza le proposte di chiusura trattativa da lui inviate agli altri utenti.
7. ViewReceivedOffersResponses: L'utente visualizza le proposte di chiusura trattativa ricevute dagli altri utenti.
8. TradeRespond: L'utente risponde ad una proposta di chiusura trattativa ricevuta da un altro utente, rifiutando oppure accettando lo scambio.
9. ViewCompletedTrades: L'utente visualizza gli scambi completati con gli altri utenti.
10. LeaveReview: L'utente, se non già fatto, lascia una recensione a un utente con cui ha completato uno scambio.
11. ViewPersonalReviews: L'utente visualizza le recensioni ricevute dagli altri utenti.
- 12.



## 5. Other Nonfunctional Requirements

### 5.1 Security Requirements

Il salvataggio delle password degli utenti dovrà avvenire dopo la loro codifica in hash MD5 per salvaguardare la confidenzialità degli utilizzatori.

### 5.2 Business Rules

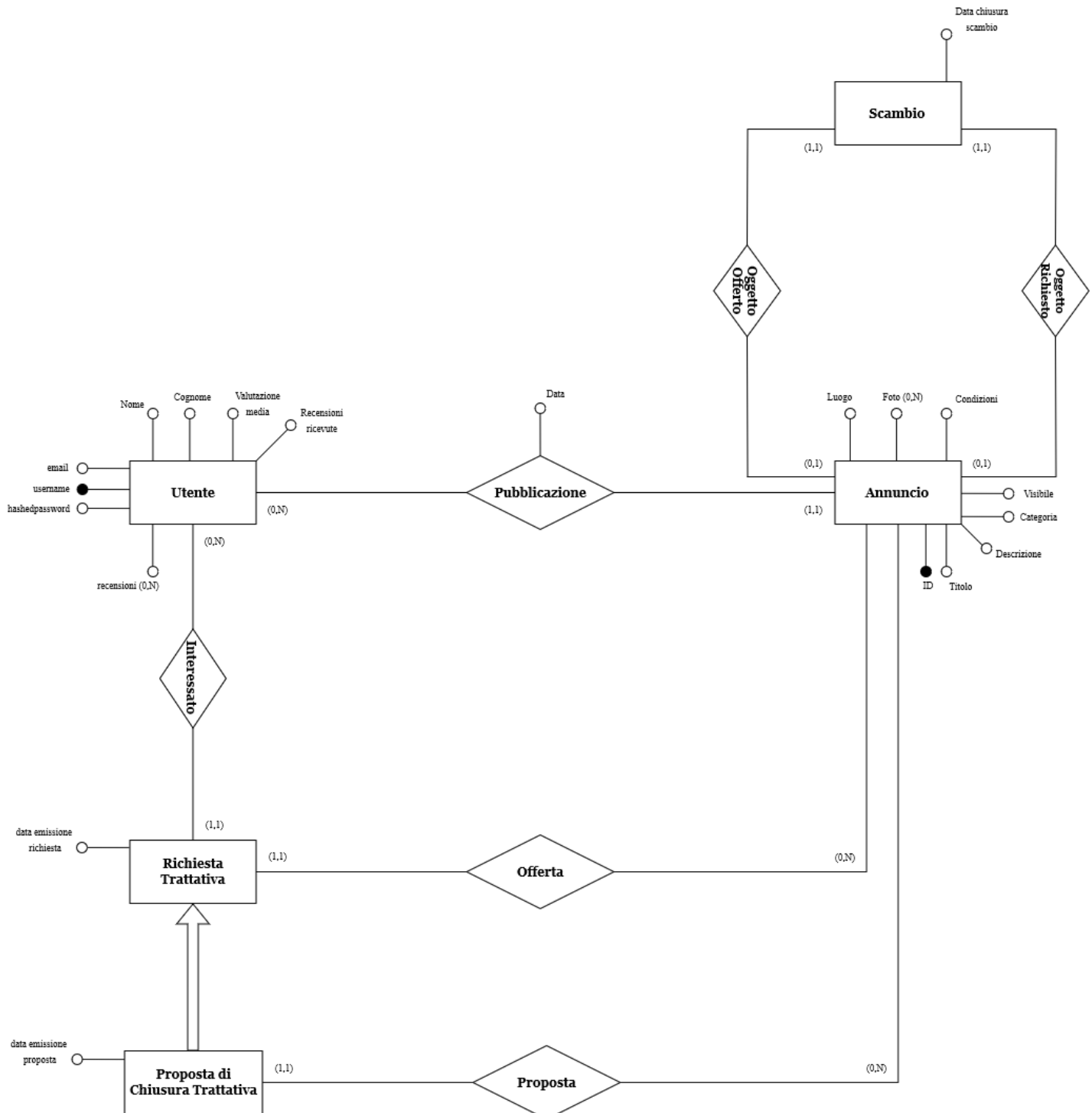
Un utente potrà inviare richieste di trattativa o proposte di chiusura trattativa solo se ha a disposizione almeno un annuncio caricato sul proprio profilo.

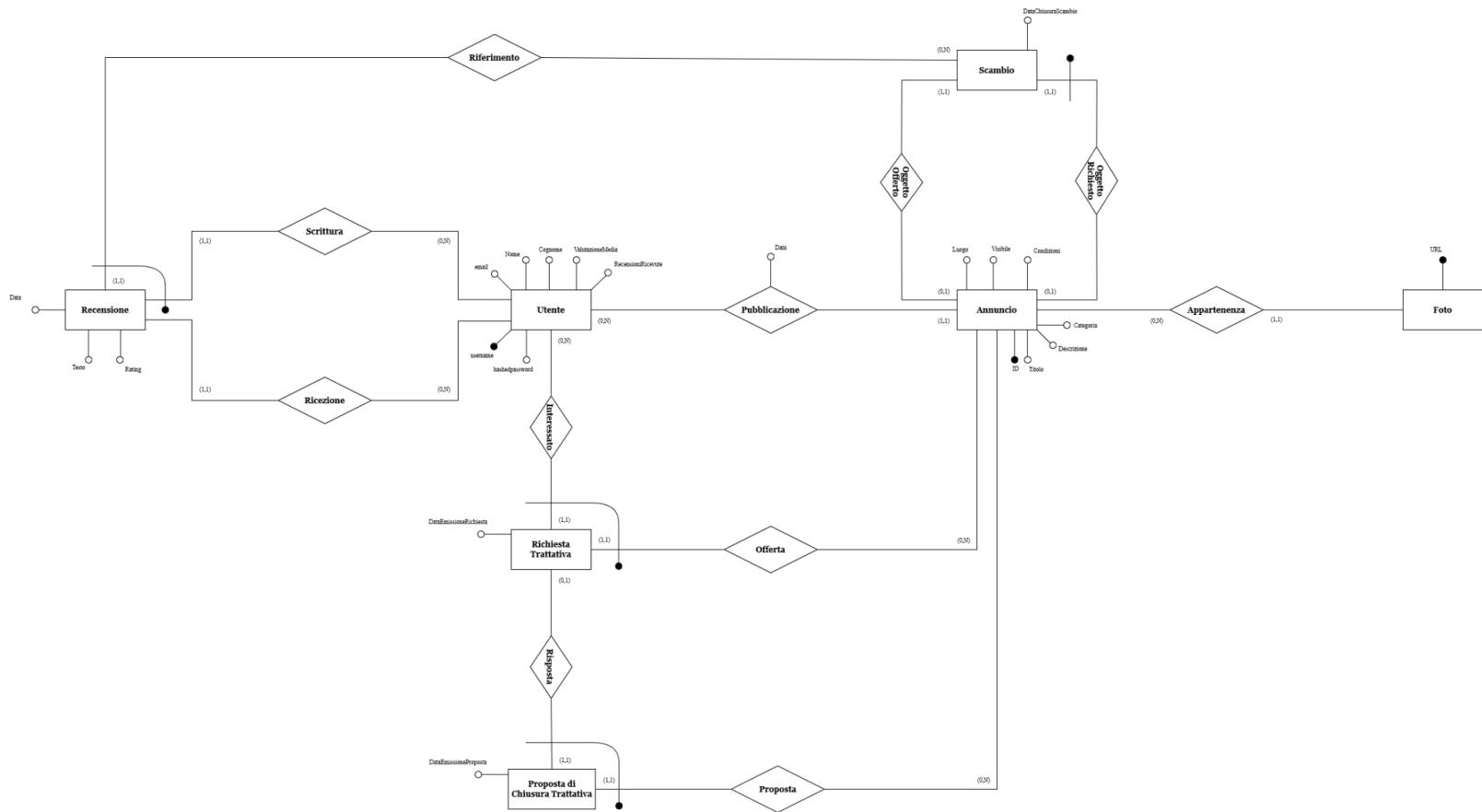
## Appendix A: Glossary

Termine	Descrizione	Collegamenti
Utente	Persona iscritta all'app di scambio oggetti	
Profilo	Profilo personale associato ad ogni utente. Contiene gli annunci dell'utente, gli scambi effettuati con altri utenti, le recensioni, le richieste di trattativa inviate e ricevute, le proposte di chiusura trattativa inviate e ricevute	Utente, Annuncio, Scambio, Richiesta di trattativa, Proposta di chiusura trattativa
Annuncio	Annuncio pubblico sul profilo dell'utente per lo scambio di un oggetto	Oggetto, Utente
Oggetto	Entità messa a disposizione per lo scambio in ogni annuncio	Annuncio
Richiesta di trattativa	Richiesta inviata da un utente ad un altro per l'avvio di una trattativa per uno degli oggetti da lui messo a disposizione in un annuncio	Proposta di chiusura trattativa, Scambio
Proposta di chiusura trattativa	Proposta inviata da un utente in risposta ad una richiesta di trattativa per un proprio oggetto, che offre di completare lo scambio con uno degli oggetti messo a disposizione dall'utente offerente	Richiesta di trattativa, Scambio
Scambio	Istanza di uno scambio tra due utenti caratterizzato dagli annunci degli oggetti che sono stati scambiati	Oggetto, Annuncio, Utente
Recensione	Valutazione rilasciata da un utente su un altro utente a seguito di uno scambio con lui effettuato	Scambio, Utente, Annuncio

## Appendix B: Analysis Models

L'analisi dei requisiti ha portato alle seguenti considerazioni. Gli **utenti** saranno identificati dal proprio username univoco e potranno caricare **annunci**, identificati da un ID, inserendo informazioni descrittive dell'oggetto dello scambio. Di seguito è rappresentato lo schema concettuale del sistema completo, incluse le componenti per la gestione degli scambi:





## Data Dictionary

Di seguito, il dizionario dei dati:

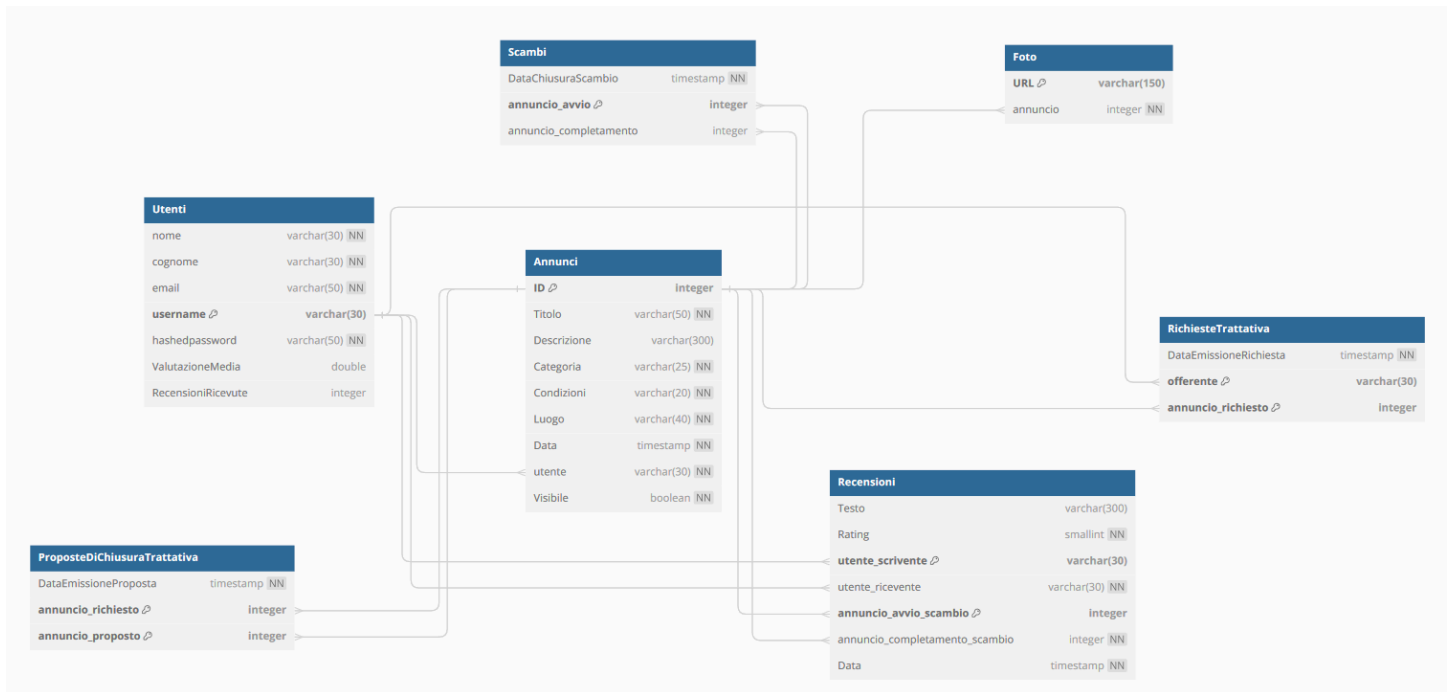
Entità	Descrizione	Attributi	Identificatore
Utente	Utilizzatore del sistema	Nome, Cognome, email, username, hashedpassword, ValutazioneMedia, RecensioniRicevute	username
Annuncio	Annuncio per lo scambio di un oggetto	ID, Titolo, Descrizione, Categoria, Condizioni, Visibile, Luogo	ID
Foto	Foto di un oggetto presente in un annuncio	URL	URL
Richiesta Trattativa	Richiesta inviata ad un utente per iniziare una trattativa di scambio	DataEmissioneRichiesta	Offerente, AnnuncioRichiesto
Proposta di Chiusura Trattativa	Proposta di risposta ad una richiesta trattativa per il completamento dello scambio	DataEmissioneProposta	AnnuncioOfferta, AnnuncioRisposta
Scambio	Entità rappresentante uno scambio completato tra due utenti	DataChiusuraScambio	AnnuncioPartenza
Recensione	Recensione di un utente verso un altro con cui ha completato uno scambio	Testo, Rating, Data	AnnuncioPartenza, UtenteScrittore

<b>Relazione</b>	<b>Descrizione</b>	<b>Componenti</b>	<b>Attributi</b>
Pubblicazione	Legame tra utente e annuncio da lui pubblicato	Utente, Annuncio	Data
Appartenenza	Legame di appartenenza di una foto a un annuncio	Annuncio, Foto	
Interessato	Legame di interesse per un oggetto tra utente e richiesta trattativa	Utente, Richiesta Trattativa	
Offerta	Legame tra richiesta trattativa e annuncio di interesse per la trattativa	Richiesta Trattativa, Annuncio	
Risposta	Legame di specificazione che lega una proposta di chiusura trattativa con una richiesta di trattativa	Richiesta Trattativa, Proposta di Chiusura Trattativa	
Proposta	Legame tra proposta di chiusura trattativa e annuncio scelto per tale proposta	Proposta di Chiusura Trattativa, Annuncio	
Oggetto Offerto	Legame tra annuncio da cui è partita una richiesta trattativa e lo scambio con esso completato	Scambio, Annuncio	
Oggetto Richiesto	Legame tra annuncio richiesto per una proposta di chiusura trattativa e lo scambio a esso associato	Scambio, Annuncio	
Scrittura	Legame tra utente che scrive una recensione e la recensione stessa	Utente, Recensione	
Ricezione	Legame tra utente che riceve una	Utente, Recensione	

	recensione e la recensione stessa		
Riferimento	Legame tra scambio completato e recensione riferente a quello scambio	Recensione, Scambio	

## Database Implementation

Dallo schema concettuale si è realizzato lo schema logico rappresentato in seguito:



Che ha portato alla realizzazione di uno script SQL per la creazione delle tabelle:

```

CREATE TABLE Utenti(
    nome VARCHAR(30) NOT NULL,
    cognome VARCHAR(30) NOT NULL,
    email VARCHAR(50) NOT NULL UNIQUE,
    username VARCHAR(30) PRIMARY KEY,
    hashedpassword VARCHAR(50) NOT NULL,
    ValutazioneMedia DOUBLE,
    RecensioniRicevute INT DEFAULT 0
)
  
```

```
CREATE TABLE Annunci(  
    ID INT PRIMARY KEY AUTO_INCREMENT,  
    Titolo VARCHAR(50) NOT NULL,  
    Descrizione VARCHAR(300),  
    Categoria VARCHAR(25) NOT NULL,  
    Condizioni VARCHAR(50) NOT NULL,  
    Luogo VARCHAR(40) NOT NULL,  
    Data TIMESTAMP NOT NULL,  
    Utente VARCHAR(30) NOT NULL,  
    Visibile BOOLEAN NOT NULL,  
    FOREIGN KEY (Utente) REFERENCES utenti(username)  
)
```

```
CREATE TABLE Foto(  
    URL VARCHAR(150) PRIMARY KEY,  
    Annuncio INT,  
    FOREIGN KEY (Annuncio) REFERENCES Annunci(ID) ON DELETE CASCADE  
)
```

```
CREATE TABLE RichiesteTrattativa(  
    DataEmissioneRichiesta TIMESTAMP NOT NULL,  
    offerente VARCHAR(30),  
    annuncio_richiesto INT,  
    PRIMARY KEY (offerente, annuncio_richiesto),  
    FOREIGN KEY (offerente) REFERENCES utenti(username),  
    FOREIGN KEY (annuncio_richiesto) REFERENCES Annunci(ID) ON DELETE CASCADE  
)
```

```
CREATE TABLE ProposteDiChiusuraTrattativa(  
    DataEmissioneProposta TIMESTAMP NOT NULL,  
    annuncio_richiesto INT,  
    annuncio_proposto INT,  
    PRIMARY KEY (annuncio_richiesto, annuncio_proposto),  
    FOREIGN KEY (annuncio_richiesto) REFERENCES annunci(ID) ON DELETE CASCADE,  
    FOREIGN KEY (annuncio_proposto) REFERENCES annunci(ID) ON DELETE CASCADE  
)
```

```
CREATE TABLE Scambi(  
    DataChiusuraScambio TIMESTAMP NOT NULL,  
    annuncio_avvio INT PRIMARY KEY,
```



```
    annuncio_completamento INT NOT NULL,  
    FOREIGN KEY (annuncio_avvio) REFERENCES annunci(ID),  
    FOREIGN KEY (annuncio_completamento) REFERENCES annunci(ID)  
)  
  
CREATE TABLE Recensioni(  
    Testo VARCHAR(300),  
    Rating SMALLINT NOT NULL,  
    utente_scrivente VARCHAR(30),  
    utente_ricevente VARCHAR(30) NOT NULL,  
    annuncio_avvio_scambio INT,  
    annuncio_completamento_scambio INT NOT NULL,  
    Data TIMESTAMP NOT NULL,  
    PRIMARY KEY (utente_scrivente, annuncio_avvio_scambio),  
    FOREIGN KEY (utente_scrivente) REFERENCES utenti(username),  
    FOREIGN KEY (utente_ricevente) REFERENCES utenti(username),  
    FOREIGN KEY (annuncio_avvio_scambio) REFERENCES annunci(ID),  
    FOREIGN KEY (annuncio_completamento_scambio) REFERENCES annunci(ID)  
)
```