
Software Requirements Specification

for

FreeProg WebApplication

Version 1.0 approved

Prepared by Saverio Valentino

Sistemi Web e Basi di Dati

Table of Contents

Table of Contents	ii
Table of Contents	ii
Revision History	ii
1. Introduction	1
1.1 Purpose	1
1.2 Intended Audience and Reading Suggestions	1
1.3 Product Scope.....	1
1.4 References	1
2. Overall Description	1
2.1 Product Perspective	1
2.2 Product Functions.....	2
2.3 User Classes and Characteristics.....	2
2.4 Operating Environment.....	2
2.5 Design and Implementation Constraints	2
2.6 User Documentation.....	2
2.7 Assumptions and Dependencies.....	3
3. External Interface Requirements	3
3.1 User Interfaces.....	3
3.2 Hardware Interfaces	3
3.3 Software Interfaces.....	3
3.4 Communications Interfaces.....	3
4. System Features	4
5. Other Nonfunctional Requirements	8
5.1 Performance Requirements	8
5.2 Security Requirements	8
5.3 Software Quality Attributes	9
6. Progettazione del Database	10
7. Test Cases	12

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

In questo documento verrà data una panoramica su una web application richiesta da una startapp per la gestione e divulgazione dei principali linguaggi di programmazione al fine di istruire, condividere e supportare tutti gli utenti della community su questi linguaggi.

1.2 Intended Audience and Reading Suggestions

Questo documento si presta ad essere una chiara guida ai requisiti software (funzionali e non), agli obiettivi della web application e alle modalità di utilizzo dello stesso, rivolgendosi a progettisti e sviluppatori coinvolti per i processi di project management e codifica.

1.3 Product Scope

La web application è progettata per agevolare la divulgazione e l'apprendimento dei principali linguaggi di programmazione. A tal fine, la web application offre una navbar intuitiva, con specifiche funzioni e operazioni che l'utente autenticato può effettuare.

1.4 References

Lo sviluppo della suddetta web application è stata sostenuta dalla formazione appresa alla 42 Roma luiss, dal materiale fornito per approfondire determinati concetti di sviluppo e dai vari tutorial su youtube riguardanti i framework utilizzati con svariate ricerche web.

2. Overall Description

2.1 Product Perspective

Questa web application mira ad essere una soluzione per semplificare l'apprendimento e la divulgazione dei principali linguaggi di programmazione, garantendo un'esperienza utente intuitiva e semplice. Questa applicazione ha anche come scopo quello di supportare l'utente nell'apprendimento in caso abbia dubbi o perplessità sul materiale didattico caricato sulla web application.

2.2 Product Functions

L'utente finale una volta registratosi e autenticatosi correttamente avrà a disposizione tre particolari funzionalità:

- *Gestione Esercizi*
Ricerca del materiale
Upload del materiale
Download del materiale
- *Gestione Video corsi*
Ricerca del materiale video
Upload del materiale video
Download del materiale video
- *Supporto*
Effettuare domande
Rispondere alle domande

2.3 User Classes and Characteristics

L'utente non registrato può solo visualizzare la vetrina dei linguaggi di programmazione che potrebbe apprendere in caso di registrazione alla piattaforma, mentre l'utente finale può far uso di tutte le funzionalità elencate sopra.

2.4 Operating Environment

Per la funzione della web application si richiede l'utilizzo di:

- *XAMPP che comprende:*
Server Web Apache
phpMyAdmin per la gestione di database MySQL
- *React + vite*
- *NodeJS*

2.5 Design and Implementation Constraints

La web application deve essere completamente fruibile, tramite i principali browser in uso, dai soli utenti registrati. Gli utenti per accedere devono autenticarsi dietro inserimento di credenziali verificate sia lato frontend che lato backend con relativi controlli, inoltre sono protette da un token JWT e i Cookies di sessione in aggiunta a password crittografate per garantire un ulteriore strato di protezione e affidabilità all'utente.

Il registro linguistico utilizzato è l'italiano.

Il sistema deve tenere traccia di tutti i file inseriti, in particolare controllando il formato e la loro dimensione.

Il sistema deve prestarsi a una manutenzione periodica.

2.6 User Documentation

La consegna del prodotto è accompagnata da una guida per l'installazione pulita della web application, ma di per sé l'utilizzo è concepito per essere user-friendly oltre alla presenza di una semplice pagina iniziale accessibile dalla homepage che fa da presentazione alla web application.

2.7 Assumptions and Dependencies

Il progetto ha superato le fasi di testing, garantendo il suo normale funzionamento al netto di aggiunte e futuri aggiornamenti. Non garantiamo il corretto funzionamento del applicativo con materiale di terze parti, inoltre si assume che vengano utilizzate le versioni più recenti delle librerie utilizzate.

3. External Interface Requirements

3.1 User Interfaces

La UI è lineare, semplice e intuitiva tramite l'uso dei vari bottoni e menù a tendina nel caso di caricamento delle risorse, la visualizzazione dell'output per le risorse è immediata all'utente grazie anche a scroll bar specifiche e ai filtri per la ricerca, filtrando addirittura i contenuti per linguaggio di programmazione, È inoltre consentito il download tramite browser delle risorse.

3.2 Hardware Interfaces

La web application viene hostato su un server Apache con tecnologia server-side PHP (Tramite XAMPP)

3.3 Software Interfaces

*Lo sviluppo della web application è stata fatta con react + vite lato frontend, un framework basato sui principali linguaggi di programmazione come: JavaScript, CSS e HTML, mirato a semplificare l'utilizzo delle risorse rendendo il tutto più dinamico possibile, soprattutto il routing delle pagine web. Inoltre, l'inclusione di Vite (tool di sviluppo) viene in soccorso per creare e ottimizzare applicazioni web moderne, inclusi progetti React. È stato progettato per essere un'alternativa più veloce e leggera rispetto ad altri strumenti.
Per quanto riguarda il lato backend è stato utilizzato PHP.*

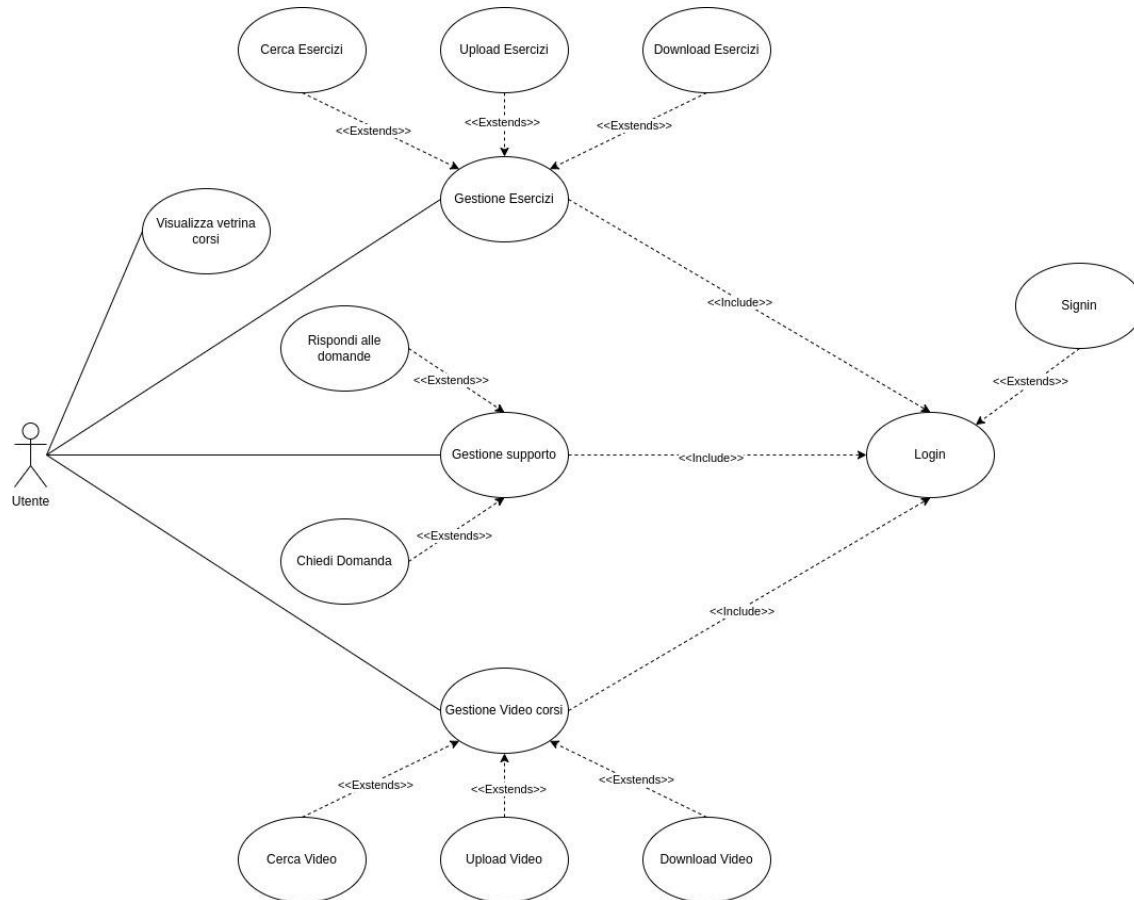
3.4 Communications Interfaces

L'infrastruttura sulla quale poggia il sistema è costituita dal web server Apache che gestisce le richieste del client web (browser) e recupera le risorse richieste dal database. MySQL, il nostro RDBMS viene gestito da PhpMyAdmin che si occupa appunto della gestione del database con una UI molto più intuitiva. Mentre la comunicazione avviene tramite scambio di messaggi HTTP.

4. System Features

Di seguito verrà illustrata nel dettaglio ogni funzionalità per l'utente.

4.1 Casi d'uso



4.2 System Feature: Sign In

4.2.1 Description and Priority

Tratterà della registrazione di un nuovo utente inserendo i vari campi richiesti.

4.2.2 Stimulus/Response Sequences

1. L'utente visualizza la home
2. L'utente clicca sul pulsante sign in
3. L'utente inserisce i dati richiesti
4. Il sistema verifica che i dati siano validi per formatazione e requisiti di sicurezza
5. Il sistema inserisce il nuovo utente nel DB.
6. Il sitema reindirizza alla pagina di login

4.2.3 Functional Requirements

REQ-1: UI interface deve presentare bottoni clicabili

REQ-2: UI interface deve presentare campi riempibili

REQ-3: UI interface deve presentare messaggi di errore in caso di errato riempimento

4.3 System Feature: LogIn

4.3.1 Description and Priority

Questa feature tratterà della autenticazione di un utente inserendo i vari campi richiesti, per poi essere confrontati per verificare se coerenti con il DB.

4.3.2 Stimulus/Response Sequences

1. L'utente visualizza la home
2. L'utente clicca sul pulsante login
3. L'utente inserisce i dati richiesti
4. Il sistema verifica che i dati siano validi per formatazione e requisiti di sicurezza.
5. Il sistema genera un token JWT
6. Il sistema reindirizza alla Homepage con le funzionalità sbloccate

4.3.3 Functional Requirements

REQ-1: UI interface deve presentare bottoni clicabili

REQ-2: UI interface deve presentare campi riempibili

REQ-3: UI interface deve presentare messaggi di errore in caso di errato riempimento

4.4 System Feature: Uploading Esercizi

4.4.1 Description and Priority

Questa feature tratterà del uploading degli esercizi, inserendo i vari campi richiesti e controllando se il file è coerente con i controlli di sicurezza.

4.4.2 Stimulus/Response Sequences

1. L'utente visualizza la pagina Esercizi
2. L'utente riempie i campi seleziona file e tipo di linguaggio poi clicca sul pulsante carica file
3. Il sistema verifica che i dati siano validi per formatazione e requisiti di sicurezza.
4. Il sistema carica il file sul DB
5. In caso di successo, il file potrà essere visualizzato sulla pagina

4.4.3 Functional Requirements

REQ-1: UI interface deve presentare bottoni e menù a tendina clicabili

REQ-2: UI interface deve presentare i file caricati

REQ-3: UI interface deve presentare una scrollbar in caso di file elevati

REQ-4: UI interface non deve presentare messaggi di errore in caso di errato riempimento (per evitare di fornire informazioni sensibili in caso di attacco)

4.5 System Feature: Uploading Video

4.5.1 Description and Priority

Questa feature tratterà del uploading dei video corsi, inserendo i vari campi richiesti e controllando se il file è coerente con i controlli di sicurezza.

4.5.2 Stimulus/Response Sequences

1. L'utente visualizza la pagina Corsi
2. L'utente riempie i campi seleziona file, titolo e tipo di linguaggio poi clicca sul pulsante carica video
3. Il sistema verifica che i dati siano validi per formatazione e requisiti di sicurezza.
4. Il sistema carica il file sul DB
5. In caso di successo, il video potrà essere visualizzato sulla pagina

4.5.3 Functional Requirements

REQ-1: UI interface deve presentare bottoni e menù a tendina clicabili

REQ-2: UI interface deve presentare campi riempibili

REQ-3: UI interface deve presentare i file caricati

REQ-4: UI interface deve presentare una scrollbar in caso di video elevati

REQ-5: UI interface non deve presentare messaggi di errore in caso di errato riempimento (per evitare di fornire informazioni sensibili in caso di attacco)

4.6 System Feature: Downloading Esercizi

4.6.1 Description and Priority

Questa feature tratterà del downloading degli esercizi.

4.6.2 Stimulus/Response Sequences

1. L'utente visualizza la pagina Esercizi
2. L'utente clicca sul file interessato
3. Il sistema reindirizza alla pagina per visualizzare il file
4. L'utente può scaricare il file

4.6.3 Functional Requirements

REQ-1: UI interface deve presentare cards clicabili

REQ-2: UI interface deve presentare una scrollbar in caso di file elevati

4.7 System Feature: Downloading Video

4.7.1 Description and Priority

Questa feature tratterà del downloading dei video corsi.

4.7.2 Stimulus/Response Sequences

1. L'utente visualizza la pagina Corsi
2. L'utente clicca sul video interessato
3. Il sistema scarica il video interessato

4.7.3 Functional Requirements

REQ-1: UI interface deve presentare cards clicabili

REQ-2: UI interface deve presentare una scrollbar in caso di file elevati

4.8 System Feature: Ricerca delle risorse

4.8.1 Description and Priority

Questa feature tratterà della ricerca e filtraggio delle risorse, utilizzando un campo di riempimento apposito.

4.8.2 Stimulus/Response Sequences

1. L'utente visualizza la pagina Corsi o Esercizi
2. L'utente riempie il campo Ricerca file per nome o per tipologia di linguaggio
3. Il sistema verifica che se ci sono riferimenti simili o uguali nel DB
4. Il sistema rende visualizzabile solo i file cercati o filtrati

4.8.3 Functional Requirements

REQ-1: UI interface deve presentare un campo riempibile

REQ-2: UI interface deve presentare i file ricercati o filtrati

REQ-3: UI interface deve presentare una scrollbar in caso di risorse elevate

REQ-4: UI interface non deve presentare messaggi di errore in caso di errato riempimento (per evitare di fornire informazioni sensibili in caso di attacco)

4.9 System Feature: Effettuare una domanda

4.9.1 Description and Priority

Questa feature tratterà del supporto alla comunità in caso di domanda, utilizzando un campo di riempimento apposito.

4.9.2 Stimulus/Response Sequences

1. L'utente visualizza la pagina Supporto
2. L'utente riempie il campo fai una domanda
3. L'utente clicca sul pulsante domanda
4. Il sistema sposta la domanda nel campo sottostante per riferirla alla comunità
5. Il sistema rende visualizzabile la domanda alla comunità

4.9.3 Functional Requirements

REQ-1: UI interface deve presentare un campo riempibile

REQ-2: UI interface deve presentare un bottone clicabile

REQ-3: UI interface deve presentare in un campo apposito le domande alla comunità

REQ-4: UI interface deve presentare una scrollbar in caso di domande elevate

REQ-5: UI interface non deve presentare messaggi di errore in caso di errato riempimento (per evitare di fornire informazioni sensibili in caso di attacco)

4.10 System Feature: Rispondere ad una domanda

4.10.1 Description and Priority

Questa feature tratterà del supporto alla comunità in caso di risposta ad una domanda, utilizzando un campo di riempimento apposito.

4.10.2 Stimulus/Response Sequences

- 1. L'utente visualizza la pagina Supporto*
- 2. L'utente utilizza il campo di riempimento Rispondi*
- 3. L'utente clicca sul pulsante Rispondi*
- 4. Il sistema salva la risposta riferita alla domanda presa in esame*
- 5. Il sistema carica tutte le risposte delle corrispettive domande distinguendole*
- 6. Il sistema rende visualizzabile le risposte alla comunità*

4.10.3 Functional Requirements

REQ-1: UI interface deve presentare un campo riempibile

REQ-2: UI interface deve presentare un bottone clicabile

REQ-3: UI interface deve presentare le risposte alle domande riferite in modo coerente

REQ-4: UI interface deve presentare le risposte alla comunità

REQ-5: UI interface deve presentare una scrollbar in caso di risposte elevate

REQ-6: UI interface non deve presentare messaggi di errore in caso di errato riempimento (per evitare di fornire informazioni sensibili in caso di attacco)

4.11 System Feature: Logout

4.11.1 Description and Priority

Questa feature tratterà del logout di un utente.

4.11.2 Stimulus/Response Sequences

- 1. L'utente clicca sul pulsante Logout*
- 2. Il sistema elimina il token JWT ed eventuali cookies di sessione*
- 3. Il sistema reindirizza l'utente alla Homepage*

4.11.3 Functional Requirements

REQ-1: UI interface deve presentare un bottone clicabile

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Grazie a react il sistema risulta efficiente e reattivo, soprattutto riguardo alla sua dinamicità nel caricamento di codice, le query permesse sono tutte semplici ed efficienti, rispondono tutte in real time e non necessitano di particolari requisiti di rete o di memoria.

5.2 Security Requirements

Il sistema si avvale di utilizzo del token JWT e i cookies di sessione, in modo da creare uno strato di protezione aggiuntiva. Le password vengono sottoposte ad hashing prima di essere salvate sul DB, inoltre è stato sviluppato un solido sistema di checking e controllo dei parametri dei vari form di autenticazione sia lato frontend che lato backend, proprio per evitare errori o caratteri indesiderati

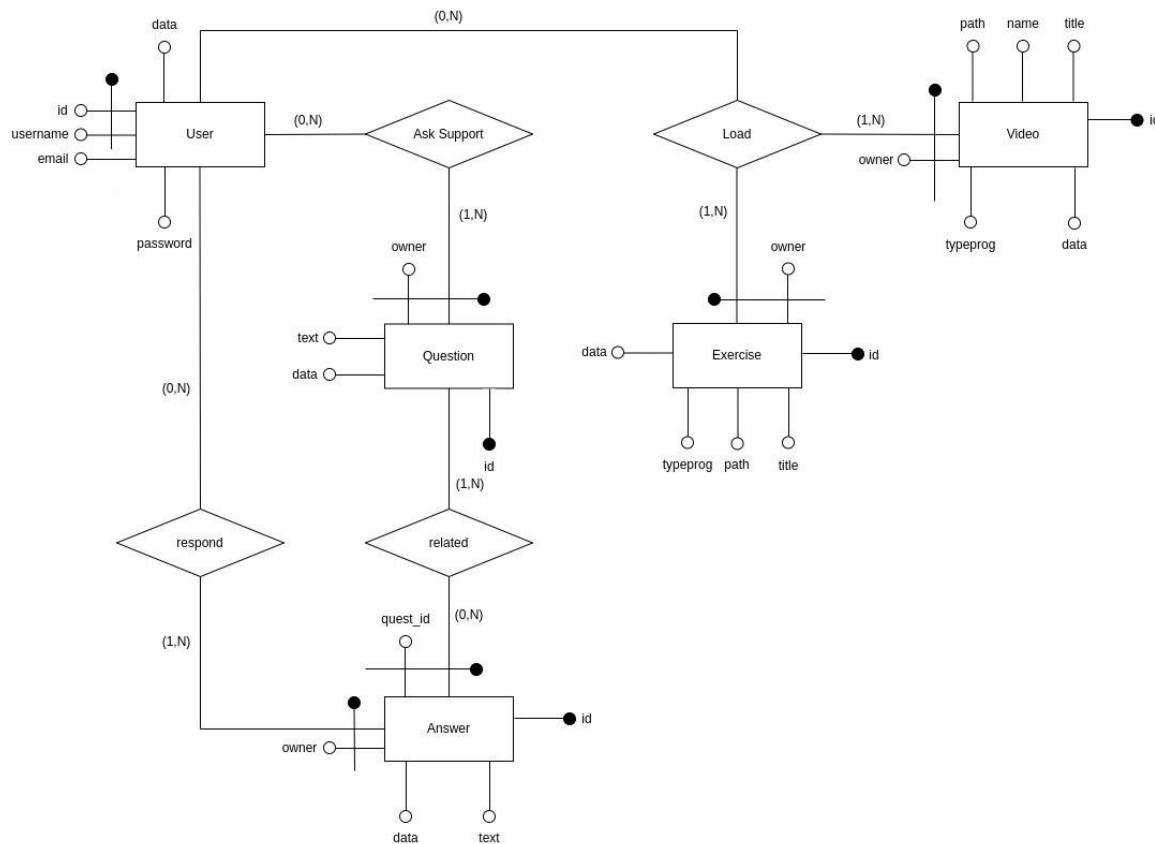
dagli utenti, il routing e i pulsanti secondari saranno attivi solo per gli utenti autenticati e i caricamenti dei file e video sono sottoposti a controlli riguardo la loro dimensione, estensione e tipo MIME. Tutto questo per evitare attacchi come XSS, SQLinjection, File inclusion e Remote Code.

5.3 Software Quality Attributes

La qualità principale del software è la sua semplicità d'uso e di gestione per l'utente. Data la sua adattabilità si possono progettare sviluppi futuri in base alle esigenze della community scalando le proprietà del software in funzione dello studio committente.

6. Progettazione del Database

6.1 Diagramma E-R



6.2 Modello concettuale del modello E-R

La relazione che intercorre tra utente e esercizio/video è di tipo molti a molti poiché gli utenti possono decidere se non caricare nulla, oppure di caricare uno o più esercizi/video. Mentre un esercizio/video o più possono essere caricati da uno o più utenti.

La relazione che intercorre tra utente e domanda è di tipo molti a molti poiché gli utenti possono decidere di non chiedere nulla, oppure di effettuare una o più domande. Mentre una domanda o più possono essere chieste da uno o più utenti.

La relazione che intercorre tra utente e risposta è di tipo molti a molti poiché gli utenti possono decidere di non rispondere alle domande, oppure di rispondere ad una o più domande. Mentre una risposta o più possono essere date da uno o più utenti.

La relazione che intercorre tra domanda e risposta è di tipo molti a molti poiché a una o più domande poste possono essere correlate nessuna o più risposte. Mentre nessuna risposta o più possono essere correlate a più domande.

6.3 Traduzione dello schema concettuale in schema logico

Utente(ID, username, email, password, data)
Esercizio(ID, possessore, utente, titolo, percorso, tipo_di_linguaggio, data)
Video(ID, possessore, utente, titolo, nome, percorso, tipo_di_linguaggio, data)
Domanda(ID, possessore, utente, testo, data)
Risposta(ID, ID_domanda, domanda, possessore, utente, testo, data)

6.4 Struttura del database

```
CREATE TABLE IF NOT EXISTS user (  
  id INT AUTO_INCREMENT,  
  user VARCHAR(50) NOT NULL PRIMARY KEY,  
  email VARCHAR(50) NOT NULL,  
  pass VARCHAR(255) NOT NULL,  
  upload_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  UNIQUE (id)  
);
```

```
CREATE TABLE IF NOT EXISTS videos (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  typeprog VARCHAR(50) NOT NULL,  
  _owner VARCHAR(50),  
  title VARCHAR(255) NOT NULL,  
  names VARCHAR(255) NOT NULL,  
  paths VARCHAR(255) NOT NULL,  
  upload_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  FOREIGN KEY (_owner) REFERENCES user(user) ON DELETE CASCADE  
);
```

```
CREATE TABLE IF NOT EXISTS exercises (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  _owner VARCHAR(50),  
  typeprog VARCHAR(50) NOT NULL,  
  title VARCHAR(255) NOT NULL,  
  paths VARCHAR(255) NOT NULL,  
  upload_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  FOREIGN KEY (_owner) REFERENCES user(user) ON DELETE CASCADE  
);
```

```
CREATE TABLE IF NOT EXISTS questions (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  _owner VARCHAR(50),  
  question_text TEXT NOT NULL,  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  FOREIGN KEY (_owner) REFERENCES user(user) ON DELETE CASCADE  
);
```

```
CREATE TABLE IF NOT EXISTS answers (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  _owner VARCHAR(50),  
  question_id INT,  
  answer_text TEXT NOT NULL,  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  FOREIGN KEY (question_id) REFERENCES questions(id) ON DELETE CASCADE,
```

FOREIGN KEY (_owner) REFERENCES user(user) ON DELETE CASCADE
);

7. Test Cases

AZIONE	COMPORTAMENTO ATTESO	OSSERVAZIONE
L'utente lascia i campi vuoti oppure inserisce caratteri non consentiti o coerenti con le norme di sicurezza nel SignIn.	Viene negata la registrazione all'utente da parte del sistema.	Il sistema mostra un messaggio di errore esplicativo.
L'utente lascia i campi vuoti oppure inserisce caratteri non consentiti o coerenti con le norme di sicurezza nel Login.	Viene negato l'accesso all'utente da parte del sistema.	Il sistema nega l'accesso mostrando un messaggio di errore esplicativo.
Un utente prova a visitare una pagina senza essersi autenticato.	Il sistema vieta la visione della pagina all'utente non autenticato.	Il sistema controlla che l'utente non si sia autenticato e di conseguenza lo reindirizza alla pagina di login.
L'utente prova ad usare i pulsanti secondari senza essersi autenticato	Il sistema vieta l'uso dei pulsanti secondari all'utente non autenticato.	
L'utente prova ad inserire username o email già in uso nel form di SignIn.	Viene negata la registrazione all'utente da parte del sistema.	Il sistema mostra all'utente un messaggio di errore esplicativo.
L'utente prova ad utilizzare una password inferiore a 8 caratteri nel form SignIn.	Viene negata la registrazione all'utente da parte del sistema.	Il sistema mostra all'utente un messaggio di errore esplicativo.
L'utente prova ad utilizzare una email che non rispetta i parametri generici di una mail nel form SignIn (tipo la presenza di una @ oppure del .it o .com).	Viene negata la registrazione all'utente da parte del sistema.	Il sistema mostra all'utente un messaggio di errore esplicativo.
L'utente dopo essersi autenticato prova a caricare un file o video che non rispetta la dimensione prestabilita.	Il sistema nega l'aggiunta dell'file o video al database.	Il sistema non mostra all'utente nessun messaggio di errore esplicativo, per evitare informazioni sensibili per eventuali attacchi al sistema.
L'utente dopo essersi autenticato prova a caricare un file o video senza aver riempito tutti i campi.	Il sistema nega l'aggiunta dell'file o video al database.	Il sistema mostra all'utente un messaggio di errore esplicativo.
L'utente dopo essersi autenticato prova a caricare un file o video che non	Il sistema nega l'aggiunta dell'file o video al database.	Il sistema non mostra all'utente nessun messaggio di errore esplicativo, per evitare informazioni

rispetta l'estensione prestabilita.		sensibili per eventuali attacchi al sistema.
L'utente dopo essersi autenticato prova a caricare un file o video che non rispetta il tipo MIME.	Il sistema nega l'aggiunta dell'file o video al database.	Il sistema non mostra all'utente nessun messaggio di errore esplicativo, per evitare informazioni sensibili per eventuali attacchi al sistema.
L'utente dopo essersi autenticato prova a caricare un file o video che è già presente nel database.	Il sistema nega l'aggiunta dell'file o video al database.	Il sistema non mostra all'utente nessun messaggio di errore esplicativo, per evitare informazioni sensibili per eventuali attacchi al sistema.