WEB DEV TEAM

Applicazione Web

Sunto

Questo documento è del gruppo di sviluppo dell'applicazione web, con la descrizione di tutti i diagrammi usati per descrivere al meglio questa parte del progetto.

Enrico Lepsoy

s6714009o@studenti.itisavogadro.it

Indice

Indice 1
Introduzione 2
Ruoli 2
RACI 2
Analisi 3
Analisi dei requisiti 3
Casi d'uso 3
Diagramma spiegato 5
Analisi Design 6
Analisi DataBase 9
Entità 10
Entità delle relazioni 13
Architettura dell'app (Diagramma di Deployment) 14
Strumenti 16
Front-end 16
Back-end 16
GANTT 17

Introduzione

Questo gruppo ha l'obiettivo di sviluppare e consegnare la piattaforma web. Si compone del Project Manager **Enrico Lepsoy** che si occupa di gestire e organizzare il gruppo, **Gabriele Amara** come Back-End developer, **Samuele Tommasi** come Front-End Developer, **Luca Prisco** come Front-End Developer, **Stefano Miceli** come analista, **Federico Pinna** come Front-End Developer e Designer e infine **Luca Dobos** come Designer.

Ruoli

Back-End Developer
Svolge la funzione di gestire il database, quindi progettarlo e crearlo; nonché di poterlo popolare attraverso delle api.
Front-End Developer
Svolge la funzione di progettare la parte di sito visibile dall'utente seguendo le sue precise specifiche e preferenze.
Designer
Svolge la funzione di gestire la parte stilistica del sito basandosi sulle richieste del cliente, come ad esempio la palette colori o il posizionamento di certi attributi sulla pagina.
Analista
Svolge la funzione di redigere documenti utili alla gestione dell'intero gruppo di programmazione e di riferire eventuali svolgimenti e modifiche al capo progetto.
RACI

Responsible (R) **-** è colui che esegue e assegna l'attività.
Accountable (A) **-** è colui che ha la responsabilità sul risultato dell'attività. A differenza degli altri 3 ruoli, per ciascuna attività deve essere univocamente assegnato.
Consulted (C) **-** è la persona che aiuta e collabora con il *Responsible* per l'esecuzione dell'attività.
Informed (I) **-** è colui che deve essere informato al momento dell'esecuzione dell'attività.

RUOLI	Dei	Lepsoy	Amara	Tommasi	Prisco	Miceli	Pinna	Dobos
Back-end	I	A	R	I	I	C	I	I
Front-end	I	A	C	R	R	C	R	I
Analisi	I	A	C	C	I	R	I	I
Design	I	A	C	C	C	C	R	R

Analisi

In questa parte del documento, saranno inserite le analisi dell'applicazione in base alle richieste del cliente.

Analisi dei requisiti

I requisiti utente vengono identificati tramite un elenco di casi di utilizzo, che corrispondono alle principali funzionalità viste dall'utente che utilizzerà il sistema informatico.
Classificheremo i requisiti in due tipologie:

- **Casi d'uso funzionali**:
  sono quelli strettamente legati alle funzionalità viste dall'utente che utilizza il sistema.
- **Casi d'uso non funzionali**:
  sono quelli non visibili chiaramente dall'utente che utilizza il sistema, ma che è il caso di analizzarli e identificarli, al fine di garantire al sistema sicurezza, qualità, correttezza ed efficienza.

L'applicazione web serve a una scuola nel Camerun, la quale ha chiesto una piattaforma dove vedere dei video che trattino argomenti di informatica ed elettrotecnica.

Questi video verranno ottenuti da remoto tramite un servizio di hosting che offre una API per vedere e scaricare video.

I video sono organizzati con dei corsi, composti da delle unità che a loro volta hanno delle lezioni e delle esercitazioni. Il metodo di accesso per fruire di questi corsi è l'uso di un account, che ha come credenziali un nome utente, o una mail, e una password. Gli utenti possono essere professori o studenti. L'assegnazione del ruolo avviene tramite un codice di conferma generati in precedenza. Ogni volta che un codice viene usato, non lo si può più riutilizzare.

Esistono delle classi, che sono dei gruppi di studenti e professori, e i professori posso essere 'tutor' o 'prof' normali. In questi gruppi, ogni professore può vedere lo stato di avanzamento degli studenti su dei specifici corsi.

Gli utenti posso seguire dei corsi, indipendentemente se siano prof o studenti. I professori posso creare dei corsi, aggiungendo all'interno unità, lezioni ed esercitazioni. Possono anche appartenere a più classi.

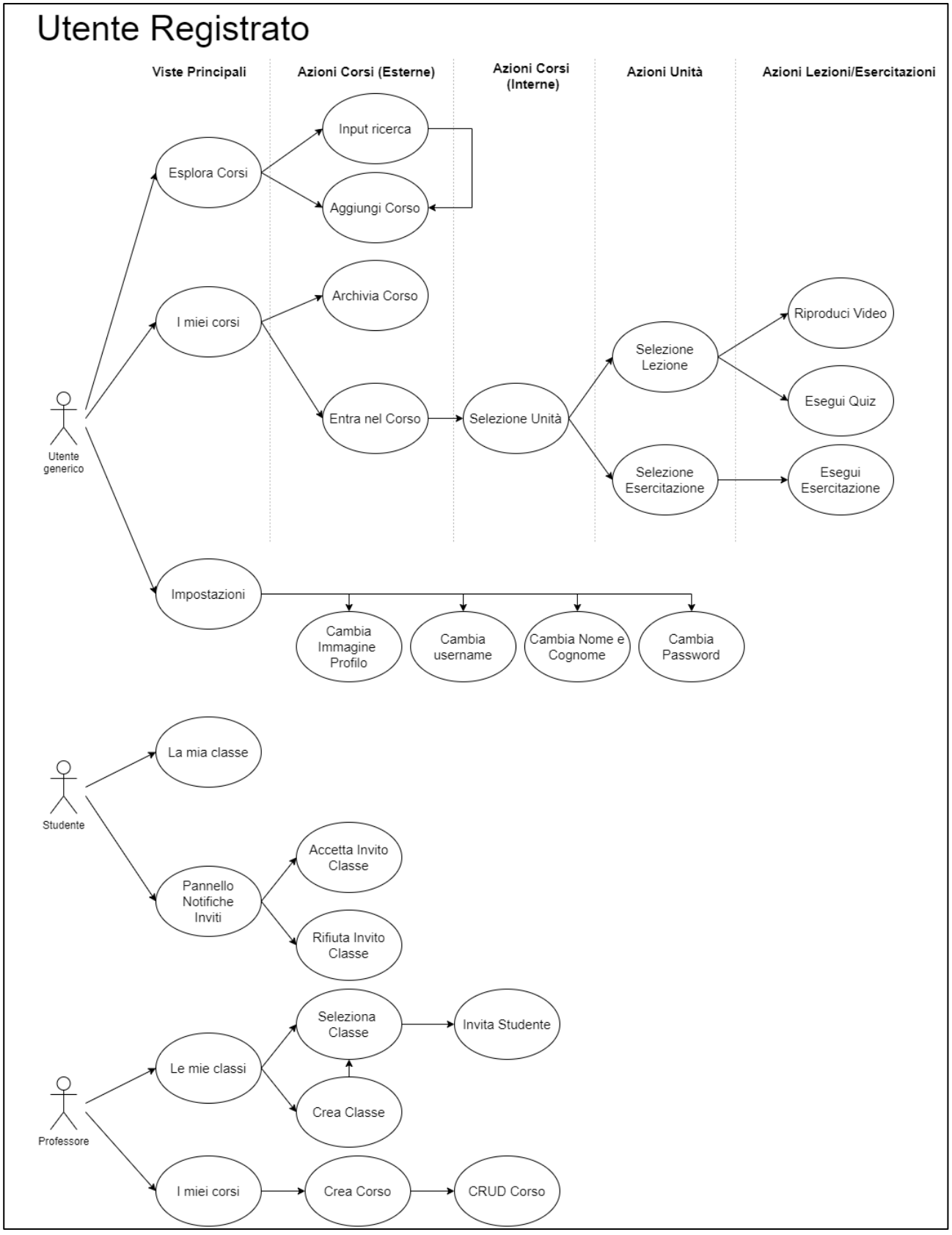
Le lezioni sono composte da un video, un nome e un quiz di conferma delle competenze, mentre le esercitazioni hanno un file di testo MarkDown e una descrizione, oltre a un suo nome.

Casi d'uso

I casi d'uso rappresentano, attraverso il linguaggio UML (Unified Modeling Language) le principali funzionalità del sistema, dal punto di vista dagli utenti che le utilizzeranno.

Descrivono le possibili interazioni tra utenti e sistema.

I diagrammi dei casi d'uso e i diagrammi di deployment inseriti nel presente documento sono stati creati mediante il tool "diagrams.net"


Utente Registrato
Viste Principali
Azioni Corsi (Esterne)
Azioni Corsi (Interne)
Azioni Unità
Azioni Lezioni/Esercitazioni
Utente generico
Esplora Corsi
Input ricerca
Aggiungi Corso
I miei corsi
Archivia Corso
Entra nel Corso
Selezione Unità
Selezione Lezione
Riproduci Video
Esegui Quiz
Selezione Esercitazione
Esegui Esercitazione
Impostazioni
Cambia Immagine Profilo
Cambia username
Cambia Nome e Cognome
Cambia Password
Studente
La mia classe
Pannello Notifiche Inviti
Accetta Invito Classe
Rifiuta Invito Classe
Professore
Le mie classi
Seleziona Classe
Invita Studente
Crea Classe
I miei corsi
Crea Corso
CRUD Corso

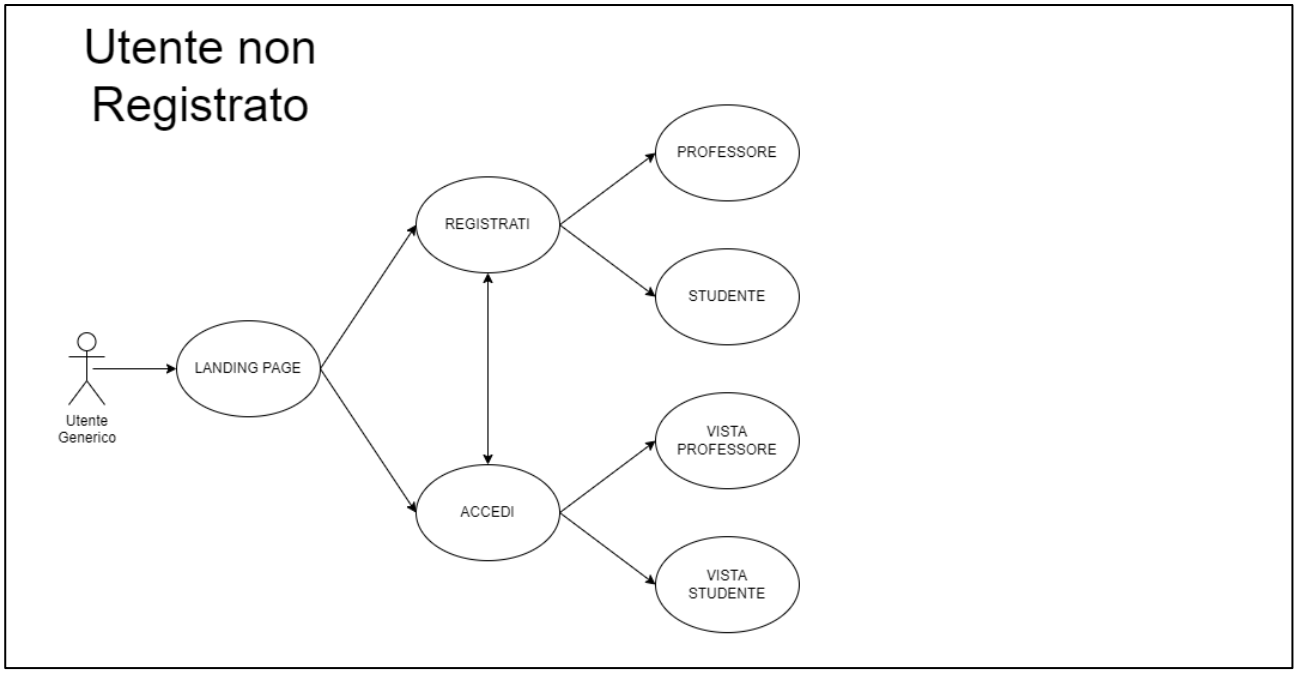


Diagramma spiegato

Come specificato prima, esistono vari tipi di attori: utente generico, professore, studente.
Se si entra per la prima volta nell'applicazione, si dà la possibilità di accedere al sito o di registrarsi. Se si accede, si diventa automaticamente professore o studente. Se invece si vuole registrarsi, si entra in una pagina di registrazione dove inserire le credenziali e il codice auto generato, come scritto in precedenza.

Gli utenti che hanno fatto l'accesso possono decidere di partecipare ai corsi. Dopo aver fatto l'accesso, esistono due attori: studenti e professori.

Un professore ha la possibilità di creare delle classi, invitando gli studenti e i vari professori, diventando tutor di quella classe e con la possibilità anche di rimuovere gli utenti di quella classe. Nella classe, tutti i professori possono vedere l'avanzamento degli studenti su dei specifici corsi. Un professore ha anche la possibilità di creare dei corsi, diventando così admin di quel corso, modificando lezioni e unità, accedendo anche alle statistiche di quel corso.

Uno studente può partecipare a una classe tramite invito da un prof tutor. Se partecipa a una classe, può seguire i corsi consigliati dalla classe, entrando così nel sistema delle statistiche.
Uno studente può seguire un corso non consigliato, senza nessuna statistica annessa.

Analisi Design

In questo capitolo vengono definiti le viste che l'utente vedrà quando navigherà nella piattaforma.

Senza aver eseguito l'accesso:

Home page

Home page con presentazione webapp e corsi presenti, possibilità di registrazione e login.

I contenuti non sono fruibili senza aver eseguito l'accesso.

Login/Register

Prima parte della web app.

Vista per la registrazione o l'entrata nell'app.

Con accesso Eseguito:

Barra superiore (Header)

L'header è la barra di intestazione della web app, presenta il logo del profilo in alto a destra, che reindirizza alle impostazioni e un'icona delle notifiche. Le notifche saranno principalmente gli inviti alle classi che potranno essere accettati, ignorati o rifiutati.

Barra laterale (Navbar)

La barra laterale è un menù di navigazione rapido per spostarsi tra le viste principali della piattaforma, che sono:

- Esplora (Ricerca dei corsi)
- I miei corsi
- Le mie classi
- Impostazioni

La barra di navigazione è sempre visibile durante la navigazione del sito a sinistra dello schermo, ed è espandibile o riducibile (con un pulsante posto sopra le opzioni di navigazione), per poter vedere i nomi delle sezioni o per risparmiare spazio e visualizzare solo le icone delle corrispondenti sezioni.

I miei corsi - Home page

La home page del sito corrisponde alla sezione "I miei corsi" dove è possibile vedere i corsi ai quali si sta partecipando o sono stati salvati.

I corsi sono elencati tramite "Tasselli" presenti al centro dello schermo e presentano le prime informazioni essenziali riguardo al corso:

- Nome del corso
- Nome del professore responsabile
- Materia
- Barra di percentuale completamento

- Immagine di copertina

Cliccando sul corso è possibile aprire la vista Corso.

Esplora

Questa sezione permette di esplorare i corsi disponibili sulla piattaforma e visualizzarne le prime informazioni. La pagina presenta una barra di navigazione superiore che permette la ricerca dei corsi tramite parole chiave, ed elenca di sotto di essa i corsi corrispondenti tramite "Tasselli".

Ogni tassello presenta le seguenti informazioni:

- Nome del corso
- Nome del professore responsabile
- Immagine di copertina
- *Frase di presentazione*
- Pulsante "Aggiungi corso"
- Pulsante "Anteprima"

Tramite il pulsante "Aggiungi corso" è possibile salvare il corso nella sezione "I miei corsi" in modo tale da essere facilmente accessibile.
Il pulsante "Anteprima" visualizza la lezione 0 del corso che mostra un breve video di presentazione, dove vengono illustrate le unità, lezioni, test, esercitazioni e argomenti che ne compongono il programma.

I corsi già aggiunti a "I miei corsi" sono comunque visualizzati in questa sezione ma contraddistinti da un colore grigio/opaco per evidenziarne lo stato, e il tassello rimane comunque interagibile.

Spostandosi con il cursore sopra il tassello del corso (hover) è possibile visualizzare un'anteprima del corso nel quale sarà presente una breve descrizione del corso.

La mia classe

La mia classe è una sezione che si differenzia se la tipologia di utente è uno studente o un professore:

Professore

Il professore è la tipologia di utente che dispone della possibilità di essere in più classi.

La vista è quindi simile a quella dei corsi, con dei tasselli sui quali è presente nome e immagine profilo della classe.

Cliccando sui tasselli è possibile espandere la vista della classe.

Nella parte superiore ci sono tre caroselli di grafici che danno una visione generale dell'andamento della classe nei corsi. Essi saranno:

Tendenza % risposte corrette per singolo quiz

Nella parte inferiore ci sono i tasselli dei professori e degli studenti che partecipano alla classe.
Cliccando sullo studente si espande la vista per vedere le sue statistiche.

Singolo corso (dati numerici):

- Avanzamento corso (%)
- Unità completate su totale (frazione)
- Esercitazioni completate su totale (frazione)
- Statistica riguardante i quiz

Studente

Lo studente può essere partecipe di una sola classe, per cui la sua unica vista sarà quella che presenta i tasselli con nome e immagine profilo degli studenti e professori.

Corso

La vista all'interno del corso è formata da riquadri:

- Zona di navigazione (sinistra): questa parte è utile a navigare attraverso le diverse sezioni del corso, e presenta di conseguenza diversi menù a tendina che rappresentano l'incapsulamento delle lezioni del corso. Esse sono:

Nel caso sia una lezione:

- Zona video (destra superiore): occupa la maggior parte della schermata ed è la zona nella quale è visualizzato il video della lezione.
- Zona descrizione (destra inferiore): è inferiore alla zona video ed occupa la stessa larghezza di quella zona, rappresenta la descrizione del video in primo piano.
- *Zona test (destra inferiore): nella zona destra è possibile scorrere sotto alla descrizione per rispondere a dei brevi test.*

Nel caso di esercitazione:

- Zona esercitazione (destra): è il riquadro complementare alla zona navigazione e sarà la zona nella quale sarà presente l'esercitazione riferita all'unità selezionata.

Interfaccia utente

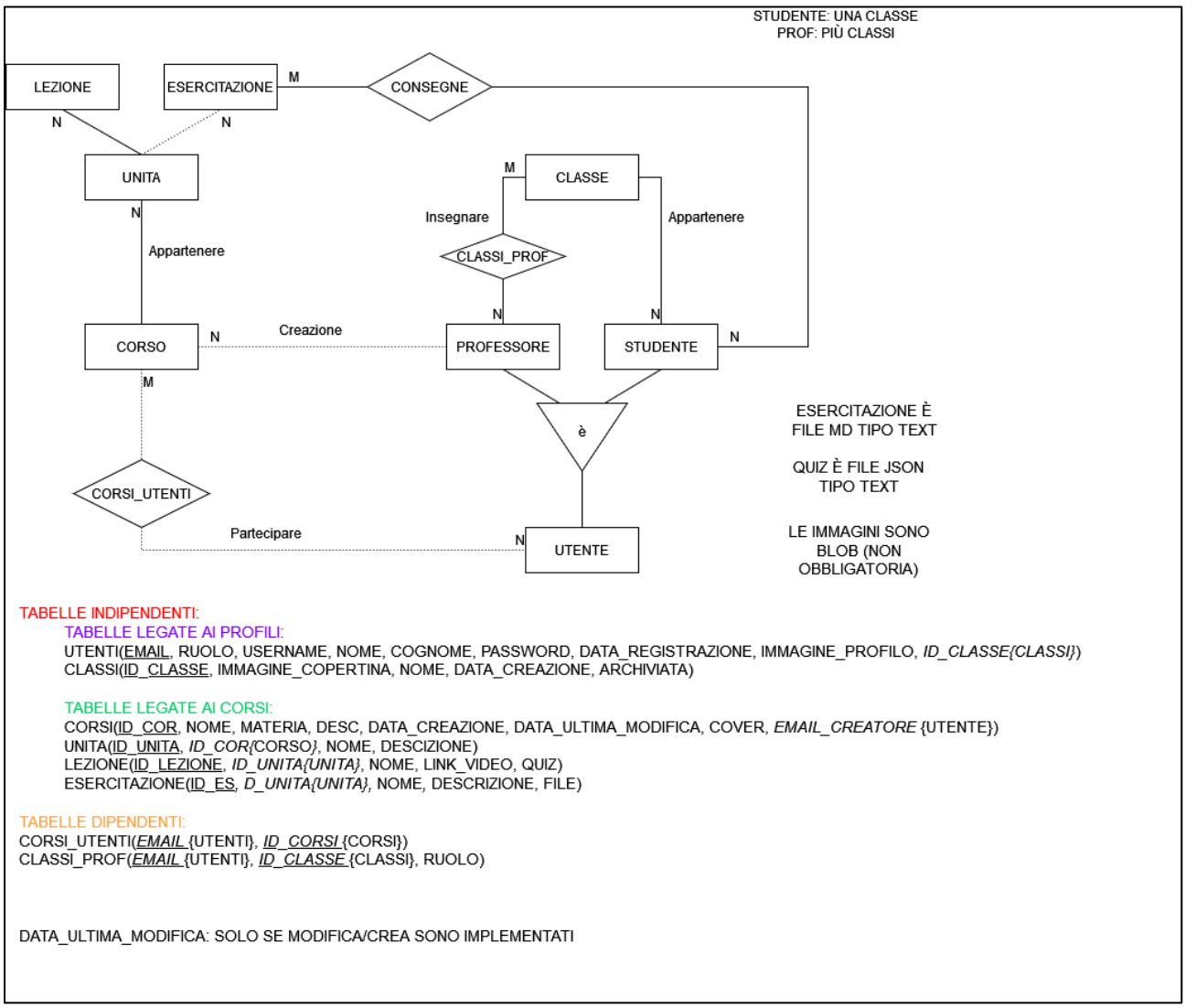
L’interfaccia utente riguarda la parte di visualizzazione dei contenuti sulla piattaforma. Le videate qui presenti sono la realizzazione di quanto spiegato nell’Analisi Design.

Analisi DataBase

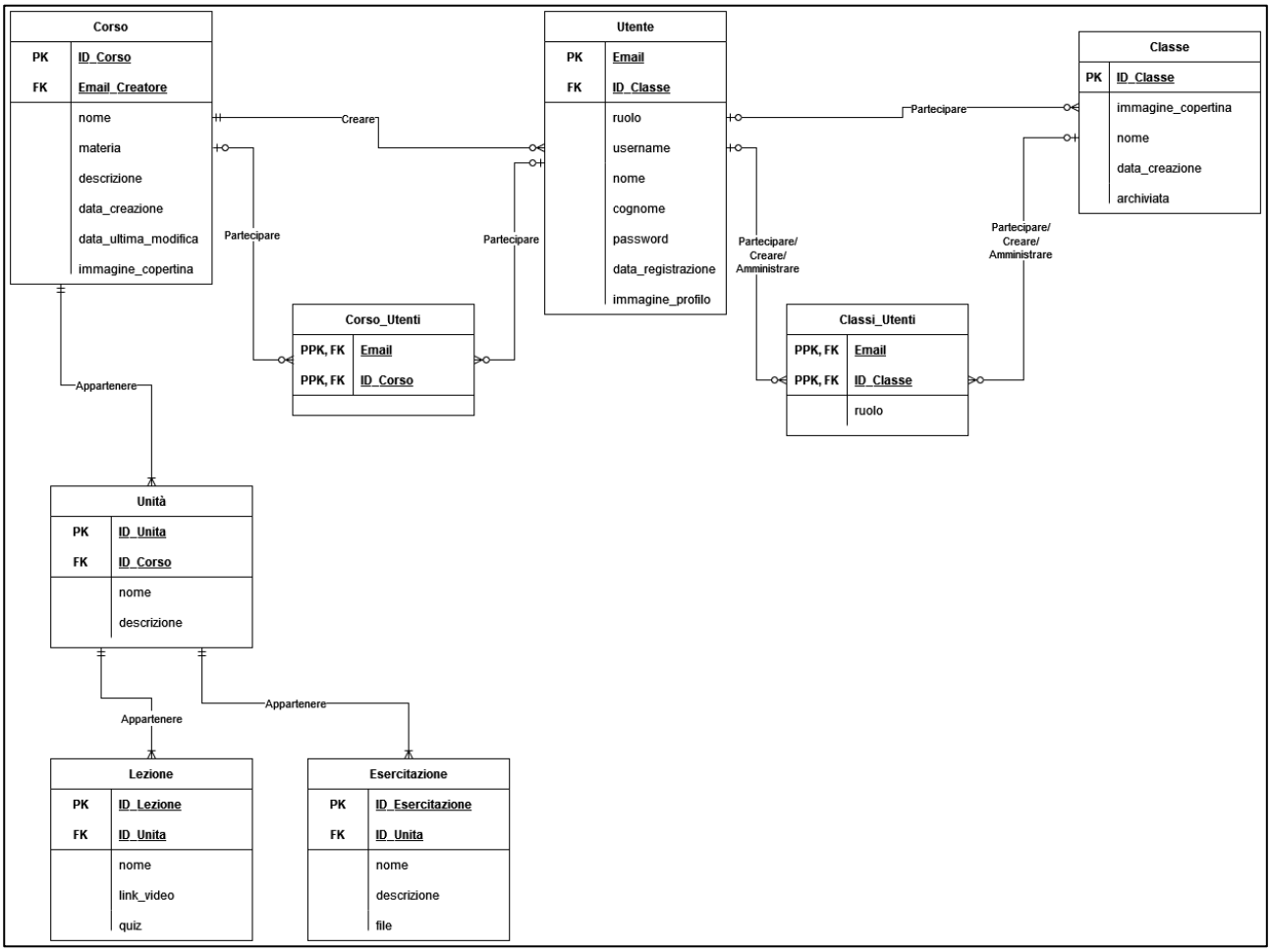
Il database è costruito per poter contenere le informazioni necessarie al funzionamento di una piattaforma web, il quale scopo è permettere di usufruire, ad una scuola in Camerun, di videolezioni asincrone di informatica ed elettrotecnica.

Le video-lezioni saranno organizzate in corsi.

Modello ER



Modello logico



Entità

Corso

Un corso è l'entità nella quale sono fruibili tutti i contenuti riguardanti un argomento specifico.

É suddiviso in Unità, ed è caratterizzato da:

- ID {PK}
- Nome
- Materia
- Descrizione
- Data di creazione
- Data di ultima modifica
- Immagine di copertina
- E-mail del creatore {FK}

Un corso è creato da un solo utente.
Un corso contiene più unità.

Un corso può essere seguito da più utenti.

Unità

L'unità è l'entità che suddivide in argomenti un corso.

Ogni unità è composta da un numero indefinito di lezioni e può avere delle esercitazioni o quiz.

É caratterizzato da:

- ID {PK}
- Nome
- Descrizione
- ID corso di appartenenza {FK}

Un'unità appartiene ad un solo corso.
Un'unità contiene una o più lezioni.

Un'unità può contenere più esercitazioni.

Lezione

La lezione è l'entità che contiene effettivamente uno dei video del corso. Una lezione contiene un solo video. Il video sarà immagazzinato come link. Ogni lezione può avere al suo interno un quiz a risposte chiuse, immagazzinate come file di tipo JSON, e avranno un sistema di autocorrezione.

É caratterizzato da:

- ID {PK}
- Nome
- Descrizione
- Link video
- ID unita di appartenenza {FK}

Una lezione appartiene ad una sola unità.

Esercitazione

L'esercitazione è un contenuto facoltativo di un'unità. Un'esercitazione è immagazzinata come file MD, e non sarà autocorretta, ma inviata al professore della classe.

É caratterizzato da:

- ID {PK}
- Nome
- Descrizione
- File
- ID corso di appartenenza {FK}

Un'esercitazione appartiene ad una sola unità.

Classe

La classe è l'entità che rappresenta un gruppo di studenti e professori. Ogni classe avrà almeno un professore con il ruolo di amministratore. Una classe è l'entità che permette ai professori di visualizzare le attività degli studenti attraverso un pannello di controllo della web app.

É caratterizzato da:

- ID {PK}
- Immagine copertina
- Nome
- Data creazione
- Stato archiviazione

Le relazioni della classe si differenziano tra gli utenti nel caso questi siano professori o studenti:
Una classe può contenere diversi studenti.

Una classe può contenere diversi professori.

Utente

L'utente è l'entità che interagisce direttamente con i corsi e classi della piattaforma. É una generalizzazione delle due specializzazioni Professore e Studente. Le due specializzazioni sono immagazzinate nella stessa tabella utente identificate dal campo ruolo. Un utente può frequentare i corsi indipendentemente dalle specializzazioni.

É caratterizzato da:

- E-mail {PK}
- Ruolo
- Username
- Nome
- Cognome
- Password
- Data di registrazione
- Immagine di profilo
- ID classe di appartenenza {FK}

Professore

Il professore può creare dei corsi e può appartenere a diverse classi. Nella tabella utenti ha il campo ID della classe impostato a NULL, poiché la relazione è risolta in una tabella di relazione.

Un professore può creare più corsi.

Un professore può partecipare a più corsi.

Un professore può appartenere a più classi.

Studente

Uno studente non può creare i corsi e può appartenere ad una sola classe.

Uno studente appartiene ad una sola classe.

Uno studente può partecipare a più corsi.

Entità delle relazioni

Classi_Prof

Questa entità è utile a gestire la relazione M:N tra le entità Classe e Professore.

É caratterizzato da:

- E-mail professore {PPK} {FK}
- ID classe {PPK} {FK}
- Ruolo del professore nella classe

Corsi_Utente

Questa entità è utile a gestire la relazione M:N tra le entità Corso e Utente.

É caratterizzato da:

- E-mail utente {PPK} {FK}
- ID classe {PPK} {FK}

Architettura dell'app (Diagramma di Deployment)

Il diagramma di deployment serve a descrivere il sistema di tipo hardware che verrà usato per la comunicazione dei messaggi tra l'utente e l'applicazione.

Analisi Diagramma Architetturale

Nodi

Avogadro Server:

Questo nodo rappresenta l'insieme dei servizi forniti dal server dell'Avogadro.

Fornisce due servizi:

- Web Server: contiene le APIs che permettono il funzionamento della web app. Le APIs sono formate da componenti in Express.js e utilizzano la crittografia JWT per mascherare i token di autenticazione.
- Database: il database permette di immagazzinare tutti i dati e le informazioni riguardanti lo stato della web app. Il RDBMS sarà MariaDB.

End User Device:

Rappresenta il device dell'utente il quale accede alla web app tramite un web browser.

La web app sarà scritta in HTML, CSS e Javascript con il framework Vue.js.

Backup Camerun Server:

Questo nodo rappresenta un qualsiasi dispositivo collocato localmente in Camerun, dove in caso di necessità verranno immagazzinati i video. Le necessità possono sorgere in caso di difficoltà di connessione con il server o di rallentamenti che impediscono la fruizione scorrevole dei video.

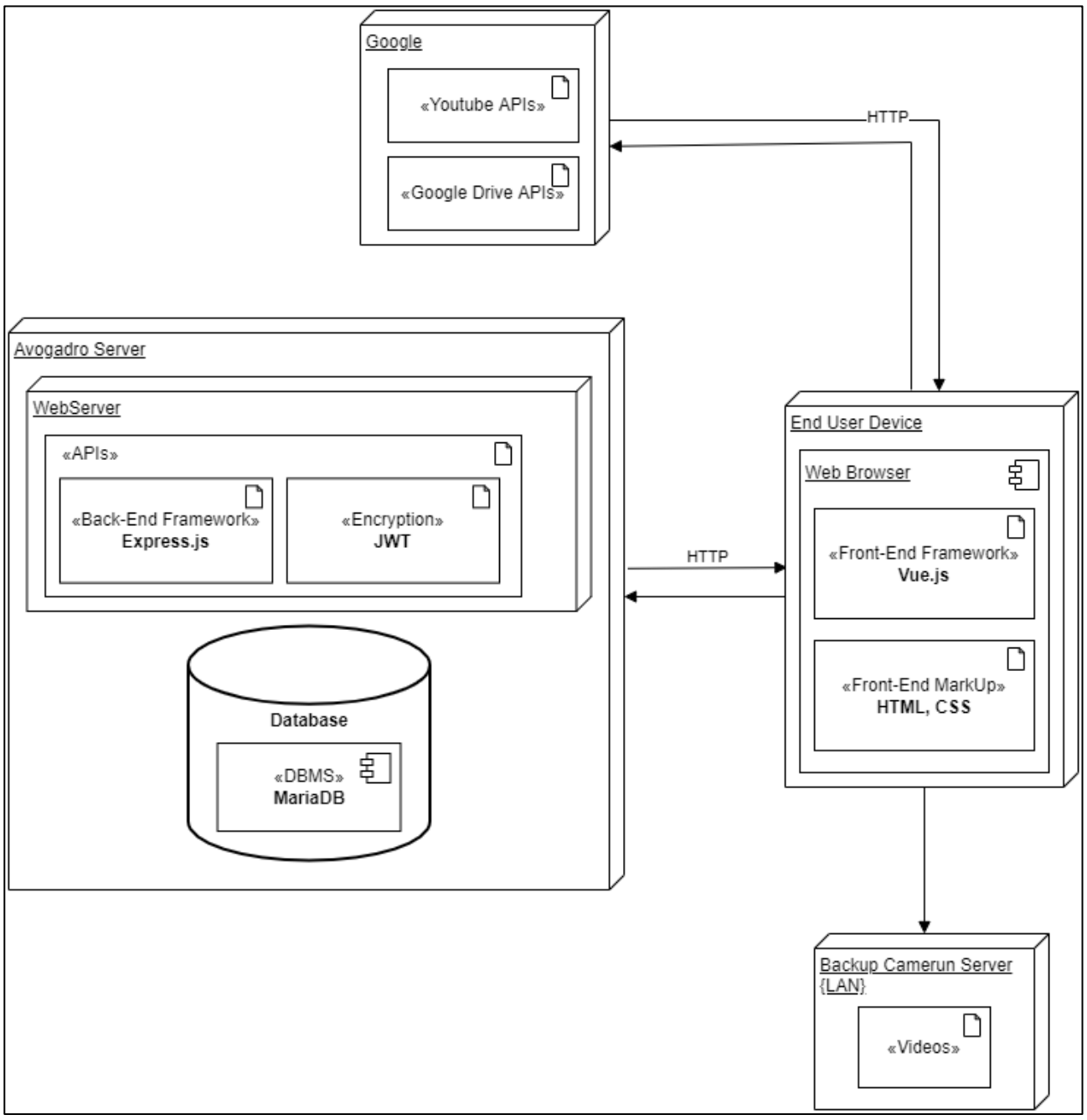
Il nodo non è necessario al funzionamento di base dell'applicazione.

Google Server:

Rappresenta il nodo che fornisce i servizi di Google, quali la YouTube API per la fruizione dei video in streaming e **da discutere e implementare** le Google Drive API che permettono il download del video.

Relazioni

Le connessioni tra i nodi avvengono tramite HTTP.


Google
«Youtube APIs»
«Google Drive APIs»
HTTP
Avogadro Server
WebServer
«APIs»
«Back-End Framework»
Express.js
«Encryption»
JWT
HTTP
End User Device
Web Browser
«Front-End Framework»
Vue.js
«Front-End MarkUp»
HTML, CSS
Database
«DBMS»
MariaDB
Backup Camerun Server
{LAN}
«Videos»

Strumenti

Di seguito ci sono le tecnologie e i linguaggi usati per la piattaforma web.

Front-end

HTML, CSS, Javascript

Linguaggi più diffusi per la creazione di web app e pagine internet. Supporto dalla maggior parte dei web browser.

Vue.js

Framework JavaScript per la creazione comoda e veloce di Web App con il paradigma MVC (Model View Controller).
Ci permette di legare il modello dati alla interfaccia grafica, ottimizzare il render, e tanto altro.

Back-end

MariaDB

Database relazionale, ottimizzato, open source, documentazione completa, facile e veloce da utilizzare, controlli sui dati (carico di lavoro minore su client), versioni collaudate e stabili.
Non utilizziamo MongoDB perché non strutturato, non facile da utilizzare con competenze scolastiche, anche se più ottimizzato, in continua evoluzione.

Node.js

Basso livello, performante, documentazione completa, facile da utilizzare con competenze di javascript pregresse, uso di codice asincrono, ottimo gestore di pacchetti, librerie e framework pronte e veloci da utilizzare.

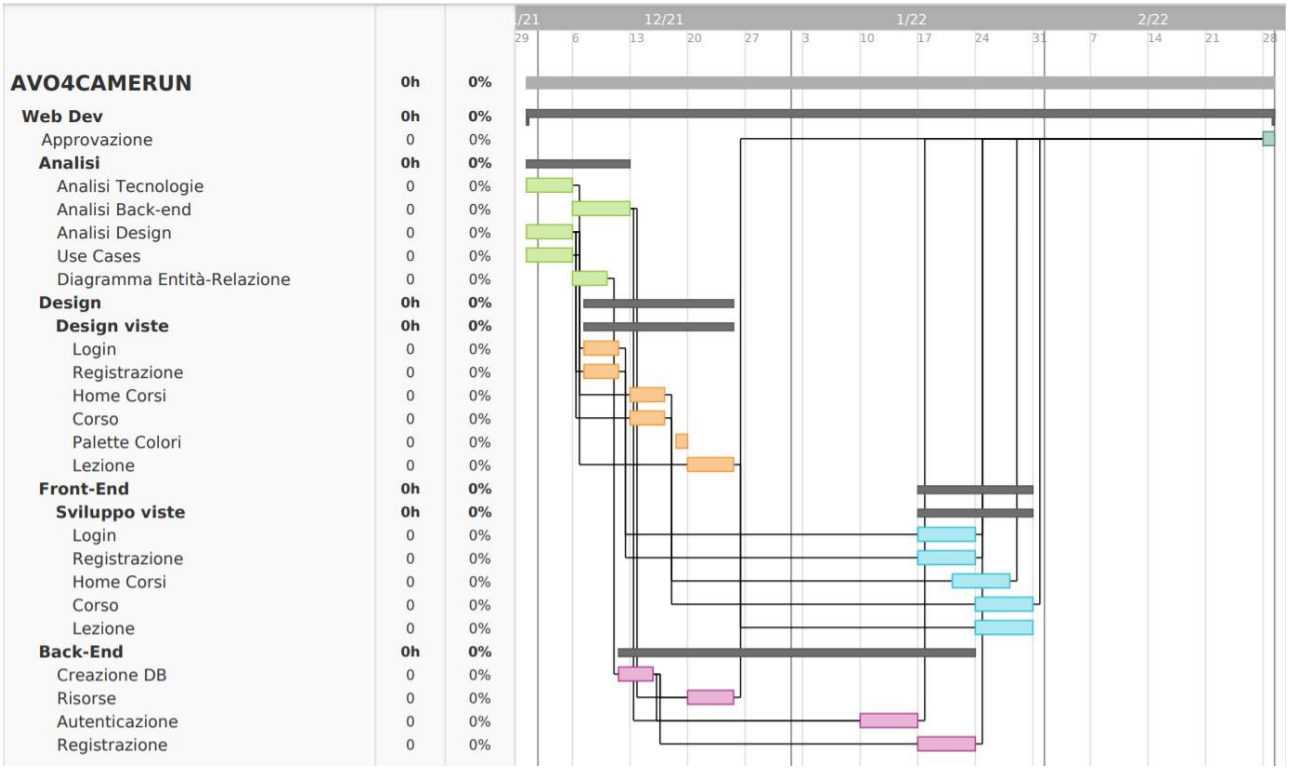
Express.js

Framework leggero per Node.js, astrae le logiche di basso livello di Node, rendendo lo sviluppo più agevole, standard per la gestione e l'instradamento delle richieste HTTP; maschera i percorsi del file system del server.

JWT (JSON Web Token)

Standard per gestire l'autenticazione e di conseguenza i permessi attraverso i token trasmessi tramite HTTP.

GANTT


/21
12/21
1/22
2/22
29 6 13 20 27 3 10 17 24 31 7 14 21 28
AVO4CAMERUN 0h 0%
Web Dev 0h 0%
Approvazione 0 0%
Analisi 0h 0%
Analisi Tecnologie 0 0%
Analisi Back-end 0 0%
Analisi Design 0 0%
Use Cases 0 0%
Diagramma Entità-Relazione 0 0%
Design 0h 0%
Design viste 0h 0%
Login 0 0%
Registrazione 0 0%
Home Corsi 0 0%
Corso 0 0%
Palette Colori 0 0%
Lezione 0 0%
Front-End 0h 0%
Sviluppo viste 0h 0%
Login 0 0%
Registrazione 0 0%
Home Corsi 0 0%
Corso 0 0%
Lezione 0 0%
Back-End 0h 0%
Creazione DB 0 0%
Risorse 0 0%
Autenticazione 0 0%
Registrazione 0 0%