



Unireview

Sviluppo e documentazione dell'applicazione

Indice

Scopo del documento

User flow

Application Implementation and Documentation

Project Structure

Project Dependencies

Modelli Database

User model

Review model

Exam model

Sviluppo API

Resources Extraction

Risorse relative agli utenti

Risorse relative alle recensioni

Risorse relative agli esami

Diagramma complessivo

Resource Diagram

User Resource Diagram

Review Resource Diagram

Email Resource Diagram

Auth Resource Diagram

Exam Resource Diagram

Resource Diagram Complessivo

Descrizione Rest API

Auth API

Accesso con Google GET /auth/google

Email API

Invio di una email di segnalazione ai moderatori POST /api/send-email

User API

Restituisci tutti gli utenti GET /api/user

Restituisci le informazioni per un utente GET /api/user/:email

Modifica dei dati di un utente PATCH /api/user/:email

Cancellazione dei dati di un utente DELETE /api/user/:email

Review API

Creazione e pubblicazione di una nuova recensione POST /api/review

Restituisci tutte le recensioni presenti in piattaforma GET /api/review

Restituisci le recensioni di un utente GET /api/review/:email

Restituisci tutte le recensioni per un professore GET /api/review/professore/:professore

Restituisce tutte le recensioni per un esame GET /api/review/esame/:esame

Elimina una recensione DELETE /api/review/:reviewId

Modifica di una recensione PATCH /api/review/:reviewId

Exam API

Ottieni gli esami GET /api/exam

Aggiungi un esame POST /api/exam

Cancella un esame DELETE /api/exam/:esame

Restituisci gli esami di un professore GET /api/exam/professore/:professore

Restituisci gli esami per nome GET /api/exam/esame/:esame

Documentazione API

Authentication Middleware

FrontEnd implementation

Testing

Risultati

Deployment

Esecuzione in locale

Scopo del documento

Questo documento comprende una descrizione delle componenti principali e dello sviluppo dell'applicazione Unireview.

Viene dapprima descritto mediante un diagramma *user flow* il flusso delle azioni effettuabili dall'utente in piattaforma.

Viene data poi un'introduzione della struttura del progetto con le sue dipendenze software e i modelli utilizzati dal database.

Vengono poi descritte tutte le componenti relative alle API e agli endpoint sviluppate a partire dal diagramma delle classi presentato nel precedente documento.

Con l'utilizzo di alcuni *resource diagram* vengono messe in relazione le api con le risorse utilizzate e consultate.

Infine vengono presentate le modalità con cui l'applicazione viene pubblicata rendendola accessibile e utilizzabile da internet, seguito da una breve spiegazione su come eseguire il codice in locale.

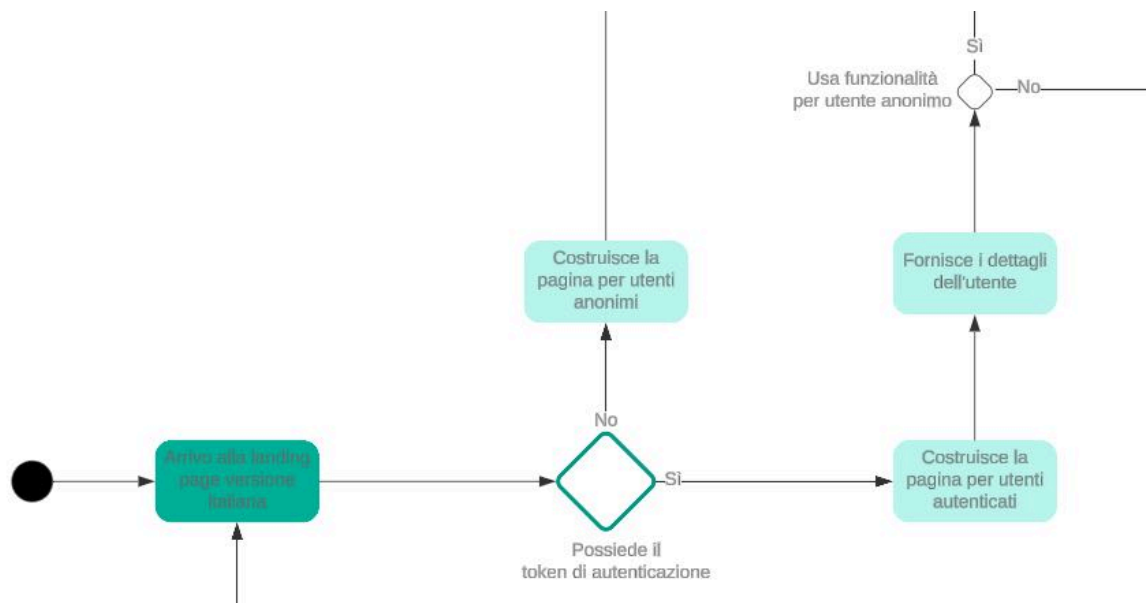
User flow

Questa sezione riporta lo user flow che descrive il flusso delle azioni operabili da un utente sull'applicazione.

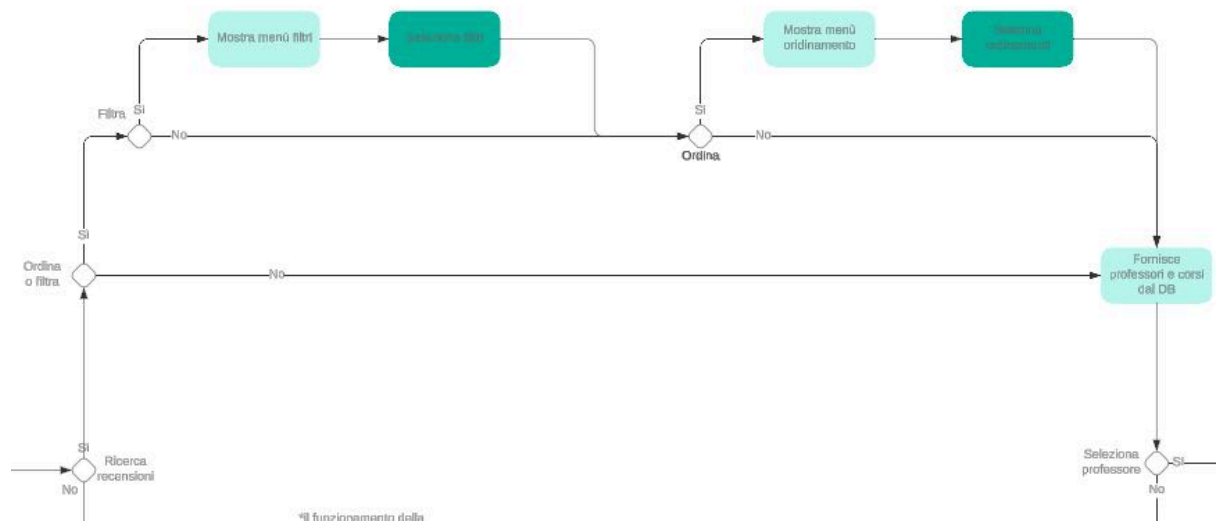
Qui sotto è riportata una didascalia dei vari componenti utilizzati nello user flow.



Si possono distinguere due tipi di bivi: il bivio semplice e il bivio non dipendente dall'utente. Il bivio semplice rappresenta una scelta dell'utente tra diverse azioni possibili in un determinato momento. Il bivio non dipendente dall'utente, invece, rappresenta i diversi stati dell'applicazione (come l'essere autenticato o meno), che determinano quali azioni sono consentite o proibite, secondo i requisiti funzionali del D1.

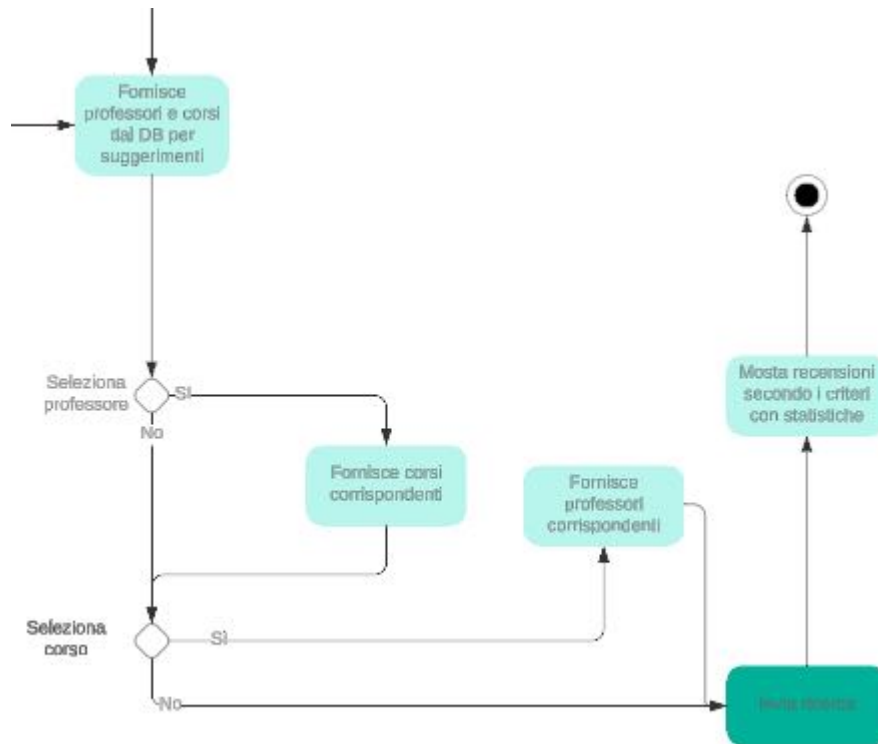


Appena entrati nella pagina si cerca il token di autenticazione per definire le funzionalità da fornire. L'utente autenticato può comunque usufruire di tutte le funzionalità di base.

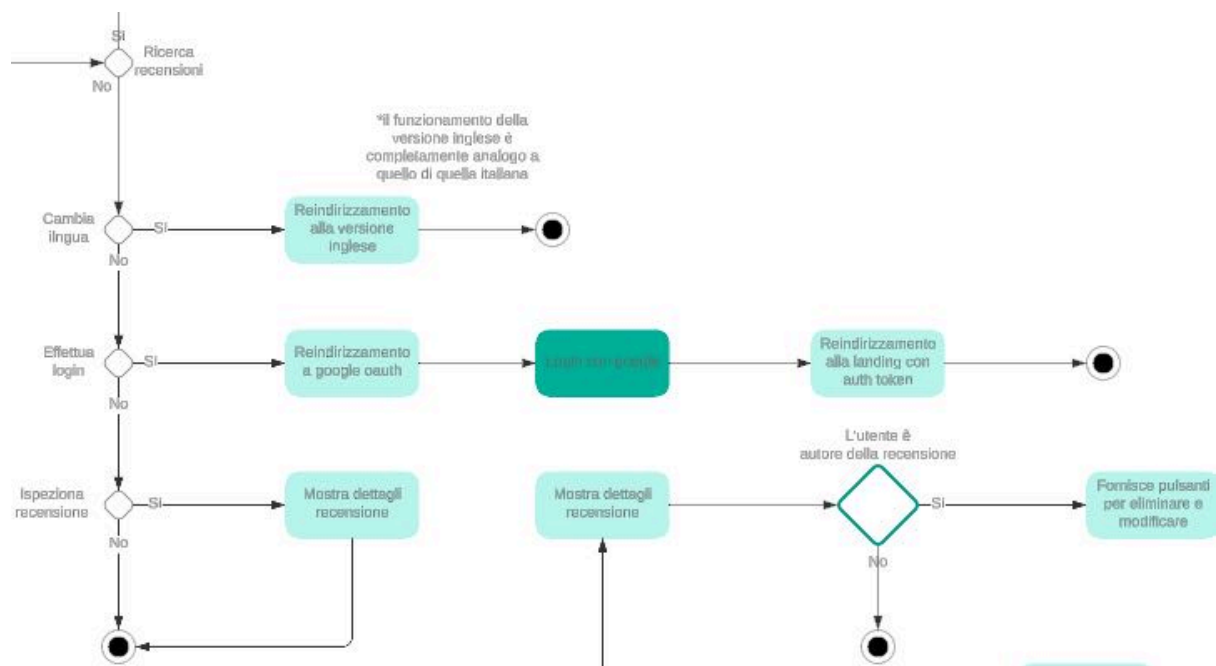


Prima di fare una ricerca è necessario selezionare eventuali filtri.

Unireview - D4

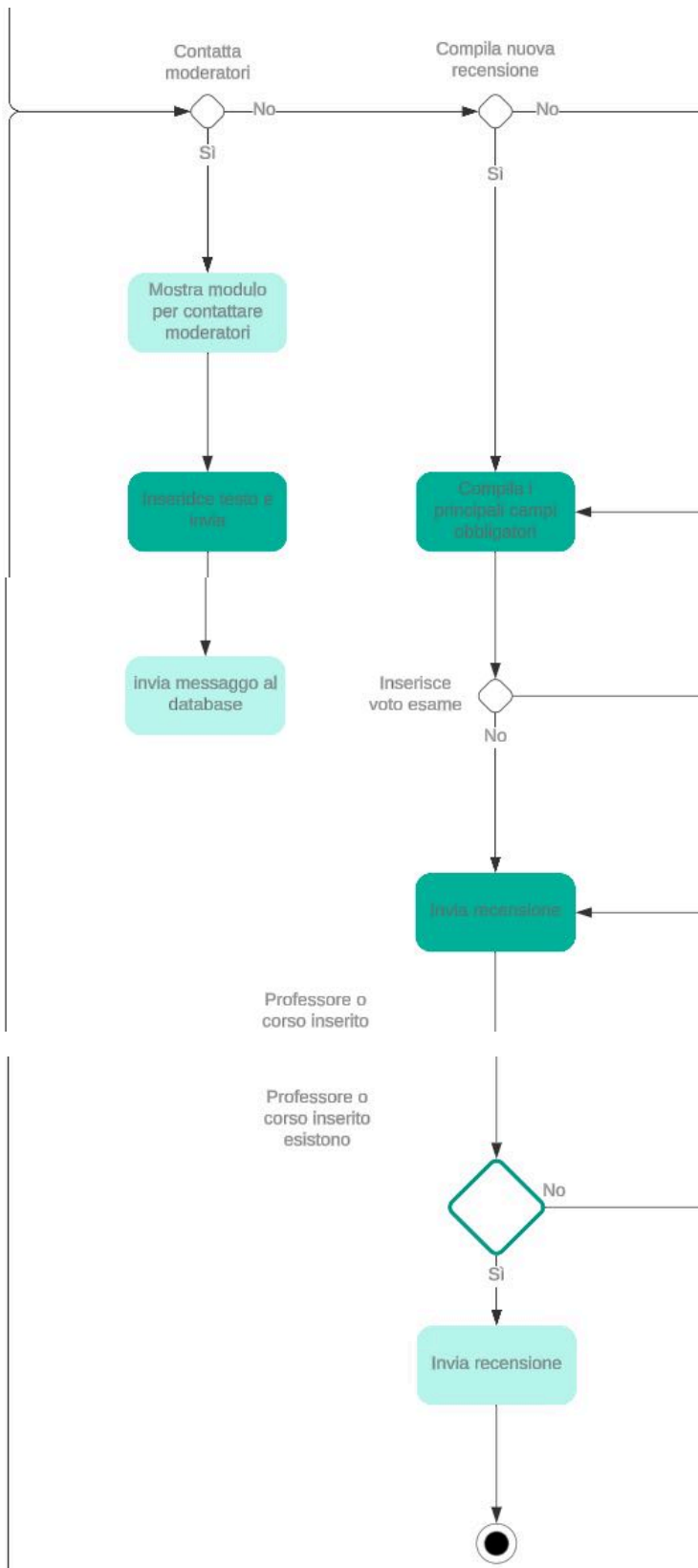


Selezione dei suggerimenti per la ricerca



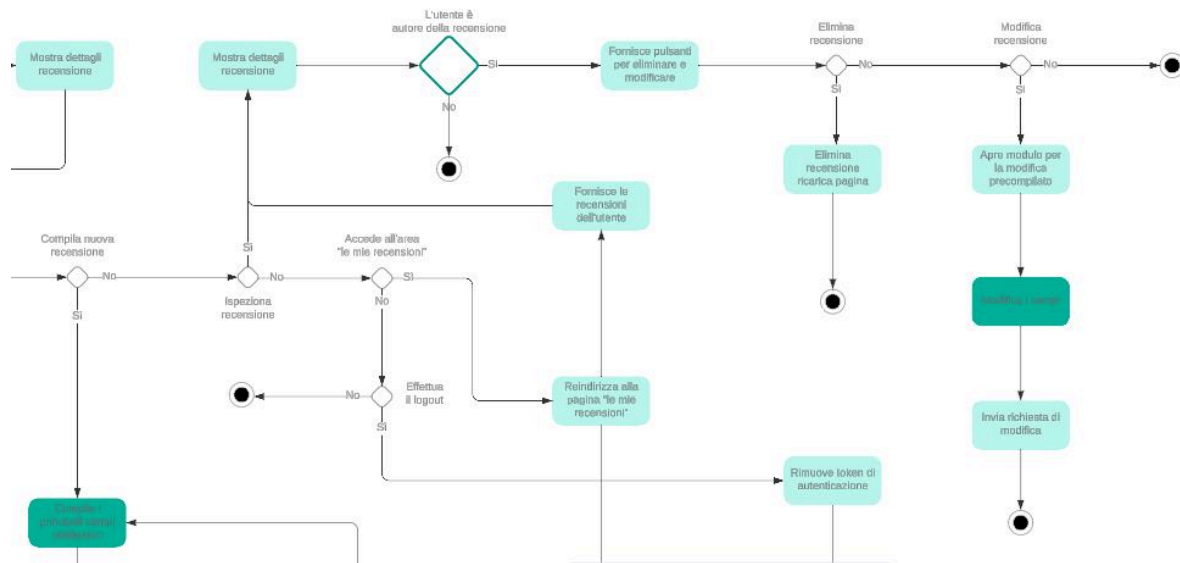
Restanti funzioni dell'utente non autenticato

Unireview - D4

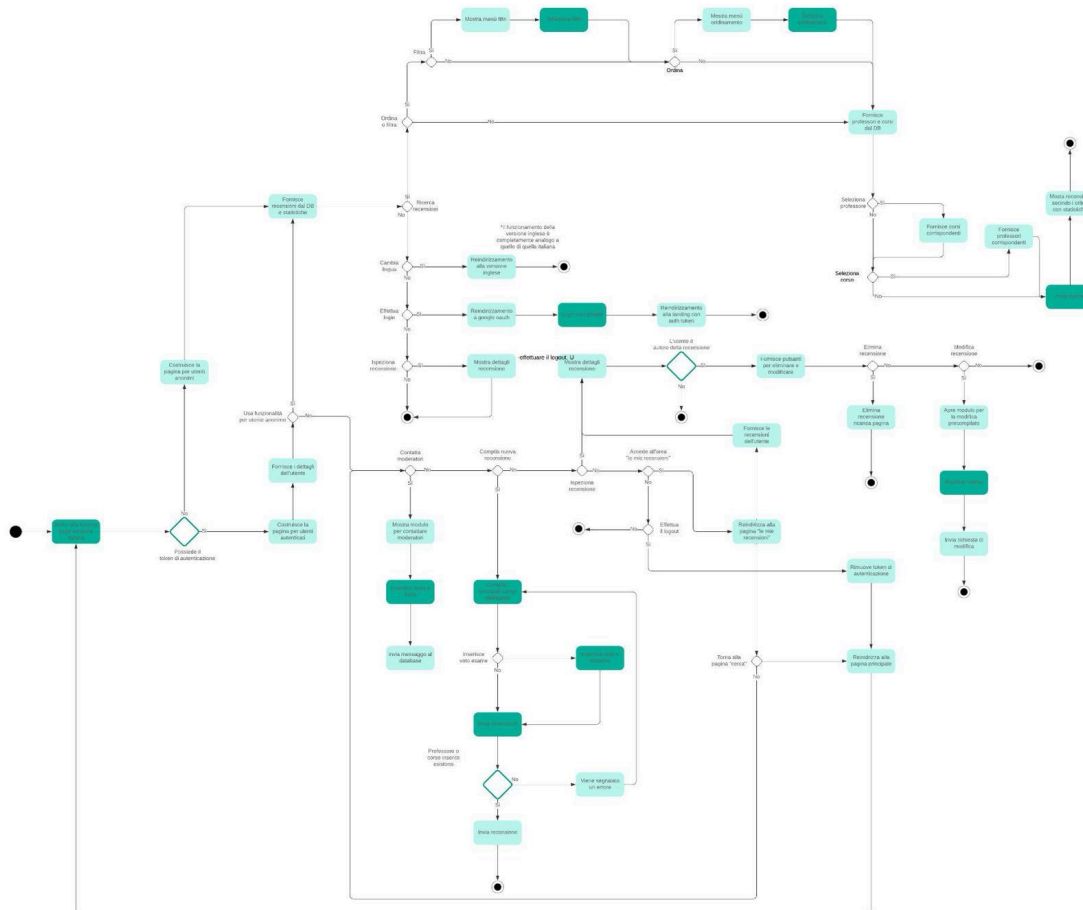


Funzionalità utente autenticato 1

Unireview - D4



Funzionalità utente autenticato 2



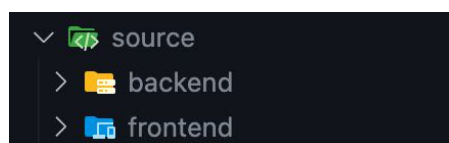
Schema complessivo

Application Implementation and Documentation

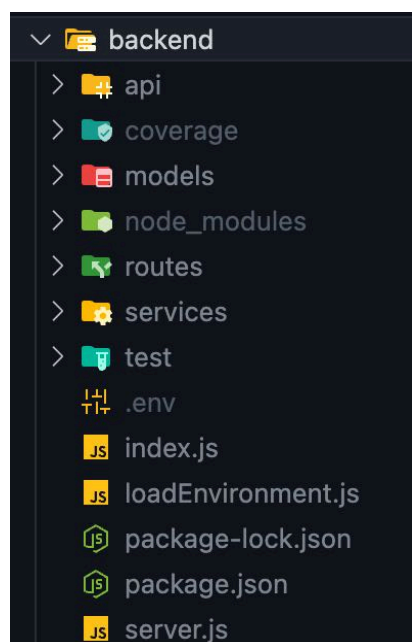
Unireview è stato sviluppato con NodeJS per la parte di backend e con NextJS per la parte di frontend. Viene utilizzato il servizio Atlas offerto da MongoDB per la parte di database e memorizzazione permanente dei dati degli utenti. Il login è reso possibile con le credenziali di ateneo attraverso l'autenticazione mediante Google OAuth che permette di sfruttare le stesse credenziali dei servizi Unitn, anche per Unireview, senza bisogno di creare un nuovo account. Ci siamo serviti di due servizi separati per il deployment dell'applicazione, Vercel per il frontend in NextJS e Railway per il backend in NodeJS.

Project Structure

Il progetto viene diviso in due subdirectory dedicate a ospitare separatamente i componenti di frontend e backend. Questo permette di avere una chiara divisione logica tra le parti.



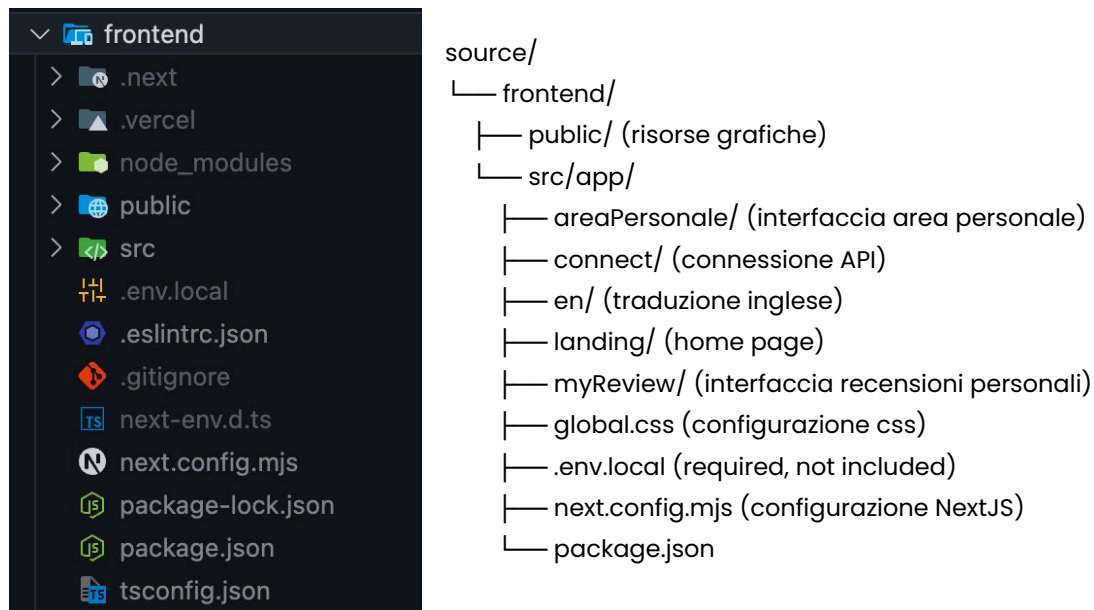
La directory *backend* ospita le componenti relative al database e alla sua connessione, le route e la definizione di tutte le api del progetto, i modelli utilizzati dal database, la documentazione delle api e i componenti relativi al testing.



```
source/  
├── backend/  
│   ├── api/ (documentazione YAML relativa alle API)  
│   ├── coverage/ (risultati relativi ai test Jest)  
│   ├── models/ (definizione dei modelli utilizzati dal database)  
│   ├── node_modules/  
│   ├── routes/ (implementazione delle API)  
│   ├── services/ (componenti relative all'autenticazione)  
│   ├── test/ (definizione dei testcase Jest)  
│   ├── index.js (avvio dell'applicazione e connessione al db)  
│   ├── server.js (registro dei API endpoints)  
│   ├── loadEnvironment.js (carica le variabili d'ambiente)  
│   ├── .env (required, not included)  
└── package.json
```


Unireview - D4

La directory *frontend* ospita le componenti relative all'interfaccia grafica dell'applicazione



Project Dependencies

Qui vengono elencate le dependencies del modulo backend:

- cors: supporto al il Cross-Origin Resource Sharing protocol
- dotenv: implementa le variabili d'ambiente definite in .env
- express: framework web
- jsonwebtoken: creazione e verifica dei token di accesso
- mongoose: permette di interagire con il database su MongoDB
- nodemailer: permette l'invio di email
- passport: middleware di autenticazione
- passport-google-oauth20: permette l'autenticazione mediante OAuth Google
- swagger-ui-express: permette di documentare le API con lo standard openAPI
- jest: esegue i test dell'applicazione e ne genera un report
- supertest: componente per testare le Rest API

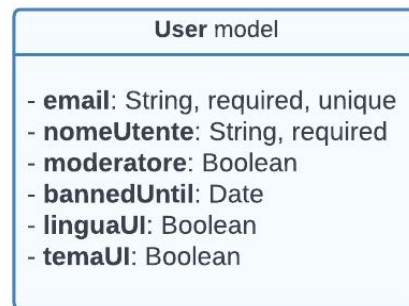
Di seguito sono elencate le dependencies del modulo frontend:

- next
- react
- react-dom
- antd: libreria grafica
- jwt-decode: decodifica i token JWT
- xhr2: framework per la connessione con le API

Modelli Database

In questa sezione vengono elencati gli oggetti json utilizzati dal database per gestire le componenti principali dell'applicazione, tra cui gli utenti, le recensioni e gli esami.

User model



Questo è l'oggetto che viene memorizzato nel database per ogni utente che effettua l'accesso sulla piattaforma.

Contiene dei parametri atti a identificare univocamente l'utente come l'indirizzo email.

Vengono salvate lato server anche informazioni come l'eventuale data di fine ban, se l'utente è moderatore o meno e altre impostazioni dell'interfaccia come la lingua e il tema.

Viene creato un oggetto *user* per ogni utente che accede alla piattaforma con le credenziali Google e il sistema si occupa di riempire automaticamente i parametri come l'indirizzo email e il nome utente, ricavando i dati direttamente dall'account Google utilizzato per l'accesso.

Review model

Review model
<ul style="list-style-type: none">- autore: String, required- professore: String, required- esame: String, required- data: Date, required- valutazioneProfessore: Number, required (1-5)- valutazioneFattibilità: Number, required (1-5)- valutazioneMateriale: Number, required (1-5)- testo: String- tentativo: Number- voto: Number- frequenza: String, required, [0%, <50%, >50%]- anonima: Boolean, required

L'oggetto *review* è responsabile della gestione dei dati delle recensioni create dagli utenti.

Conserva dati come il testo della recensione, il voto conseguito dallo studente, i parametri di valutazione espressi da uno a cinque, ma anche altri parametri relativi alla compilazione della stessa, come l'autore, la data di creazione e la preferenza di anonimità.

Anche nel caso di *review*, alcuni parametri vengono inseriti automaticamente dal sistema nel momento dell'invio, come la data di compilazione e l'autore della recensione.

Exam model

Exam model
<ul style="list-style-type: none">- professore: String, required- esame: String, required

L'oggetto *exam* è il componente fondamentale che associa gli esami con i relativi professori.

L'insieme di questi oggetti crea la lista di esami e professori per cui è possibile compilare una recensione nella piattaforma.

Solamente i moderatori del sistema possono creare e eliminare oggetti di questo tipo, in quanto sono componenti funzionali al sistema.

Sviluppo API

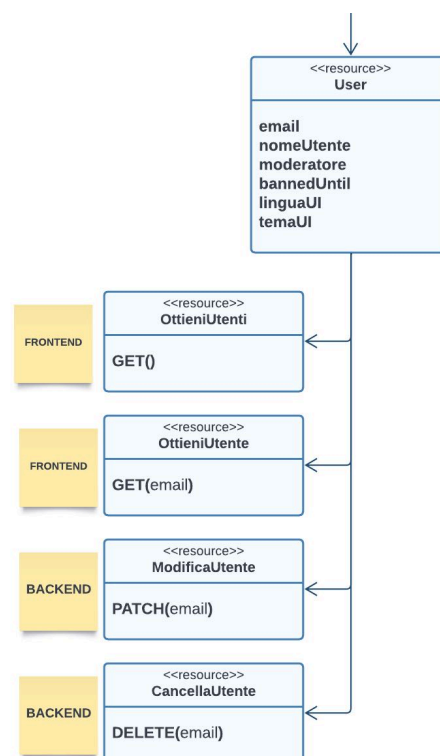
Resources Extraction

In questa sezione viene mostrato come siamo passati allo sviluppo delle componenti del sistema partendo dal Diagramma delle Classe documentato nel terzo documento.

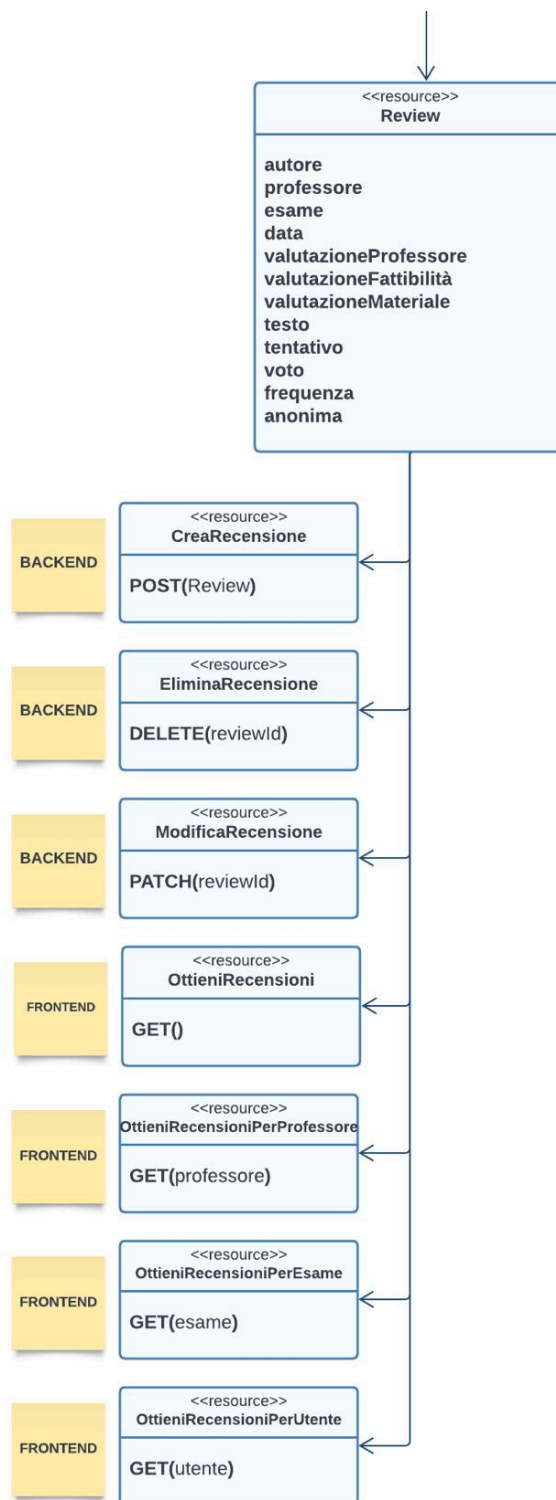
Ogni risorsa utilizzabile è affiancata da un *label* che indica se essa ha principalmente un riscontro visivo [*frontend*] oppure lavora maggiormente sul server side [*backend*].

Vengono prima presentati dei sotto diagrammi per facilitare la lettura e poi il diagramma complessivo che permette di avere una visione di insieme del progetto.

Risorse relative agli utenti



Risorse relative alle recensioni



Risorse relative agli esami

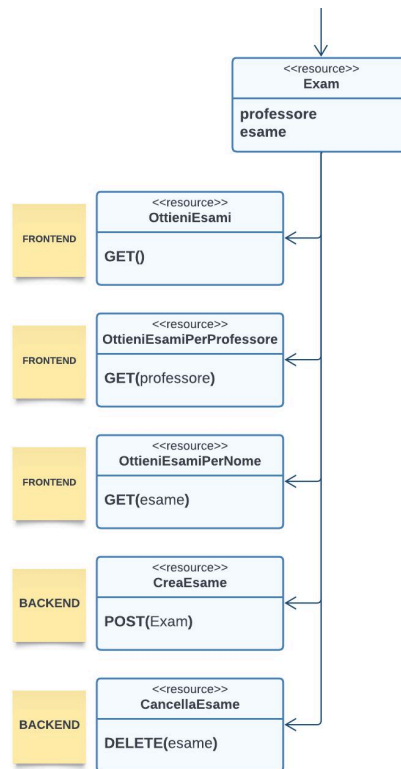
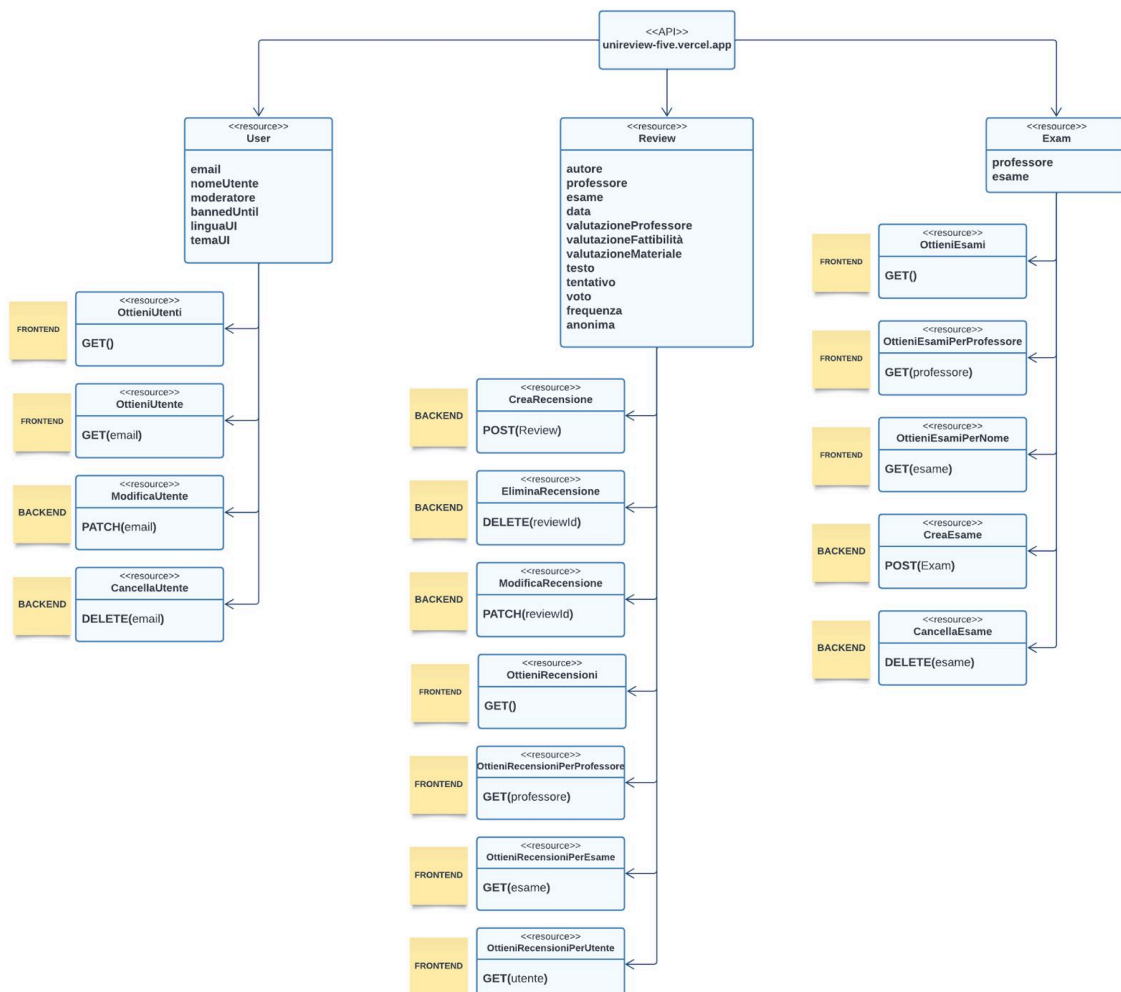


Diagramma complessivo



Resource Diagram

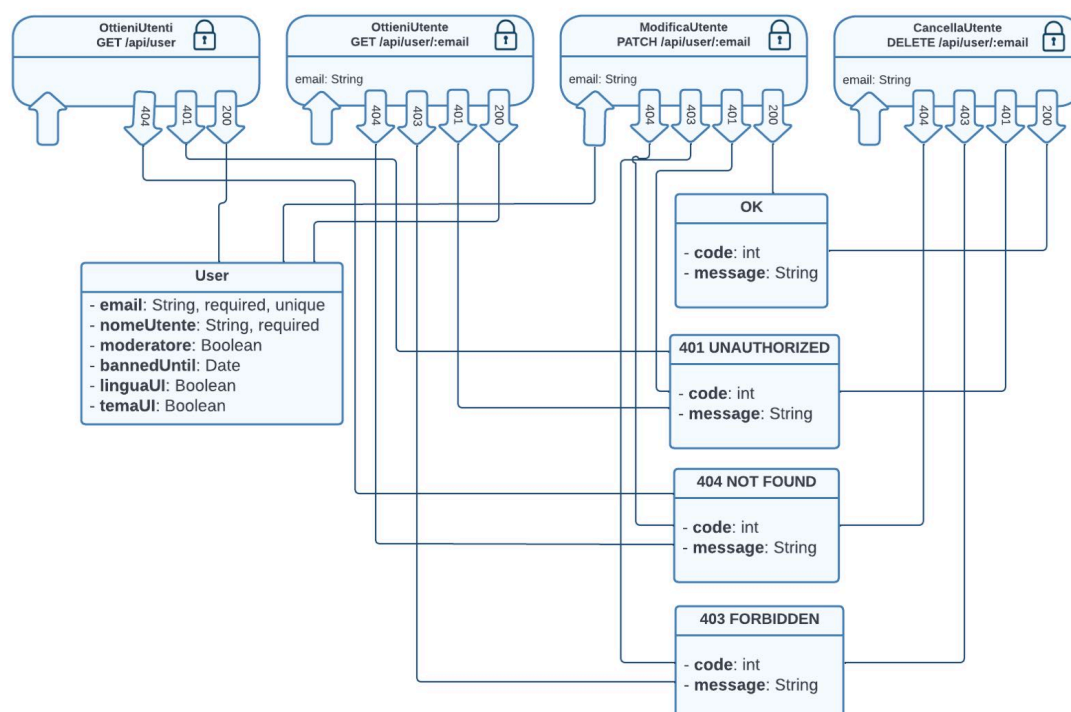
In questa sezione vengono rappresentate sotto forma di diagramma le API utilizzate dalla piattaforma Unireview.

Le API sono state suddivise in categorie per semplificare la vista, seguendo la logica dei modelli descritti in precedenza e la struttura dei file del progetto. Per semplicità viene omesso da tutti i diagrammi l'errore generico del server (500) che comprende un messaggio di descrizione.

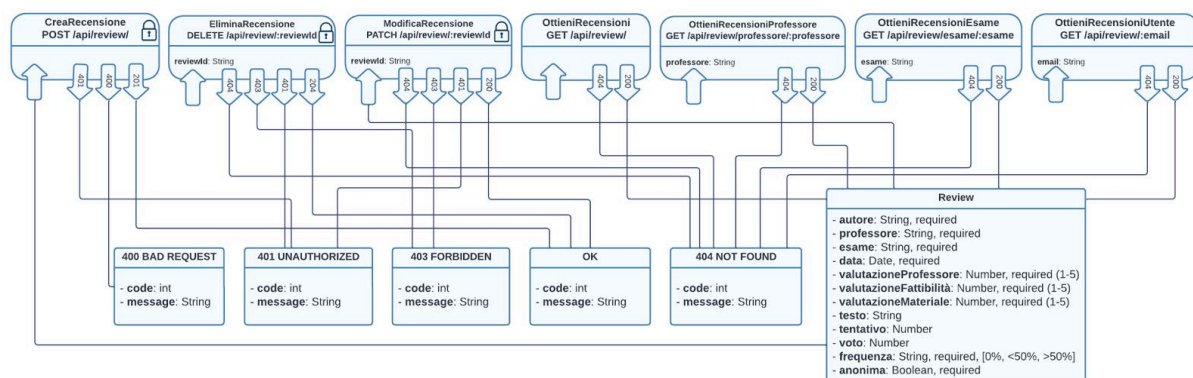


Indica che deve essere presente il token JWT nell'header della richiesta

User Resource Diagram

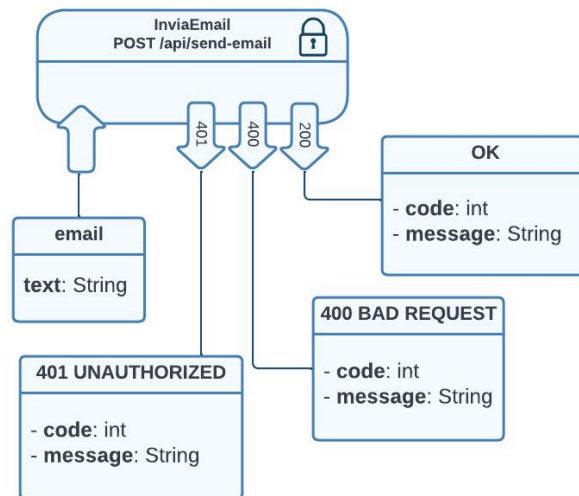


Review Resource Diagram

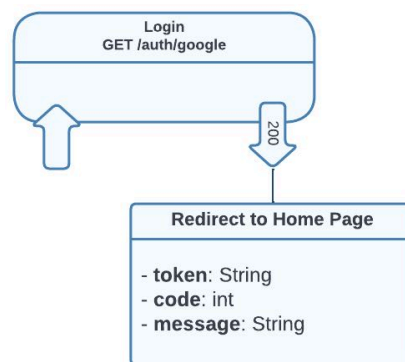


Unireview - D4

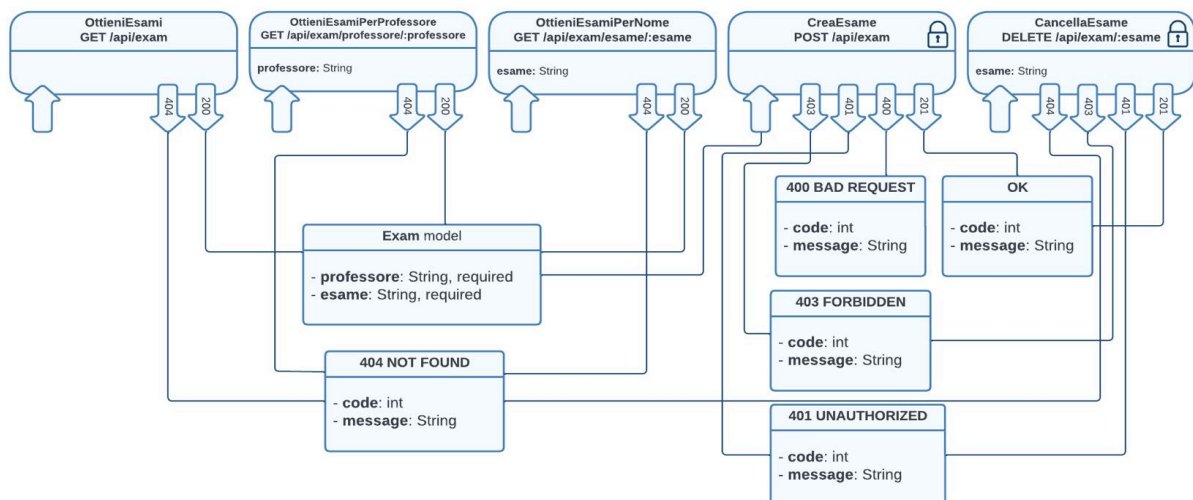
Email Resource Diagram



Auth Resource Diagram

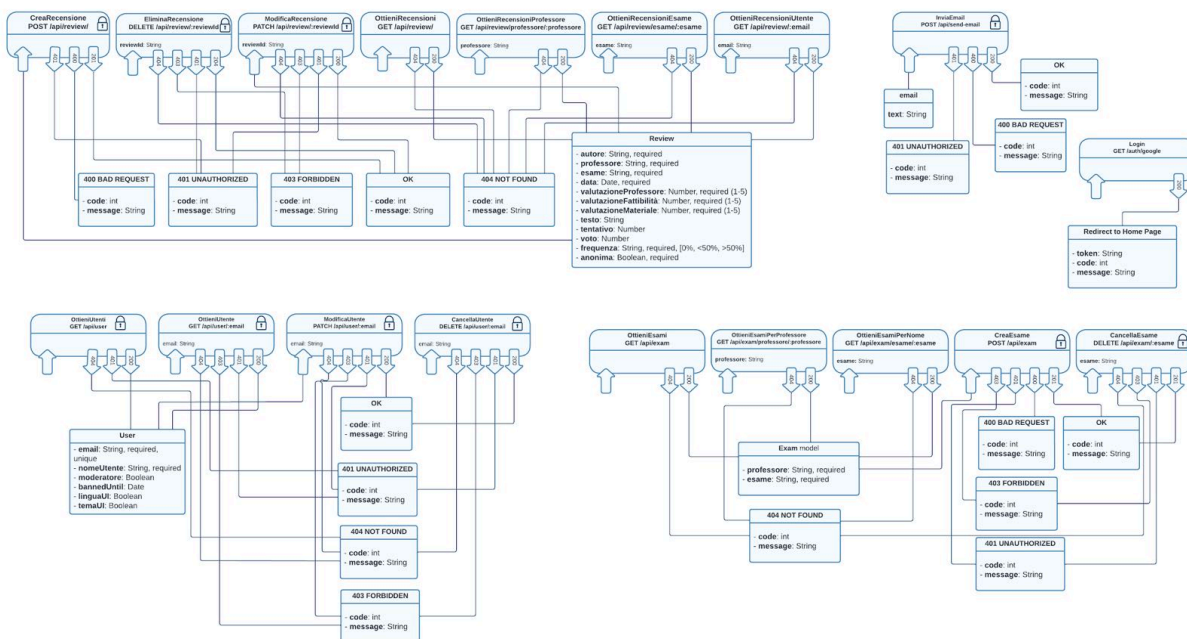


Exam Resource Diagram



Resource Diagram Complessivo

Questa visualizzazione non comprende nulla di nuovo rispetto ai diagrammi precedenti, ma ha lo scopo di restituire una visione d'insieme del progetto.



Descrizione Rest API

In questa sezione verranno elencati e commentati tutti gli endpoint delle API utilizzate dall'applicazione Unireview.

Gli endpoint sono stati divisi in categorie che li raggruppano per funzionalità e utilizzo.

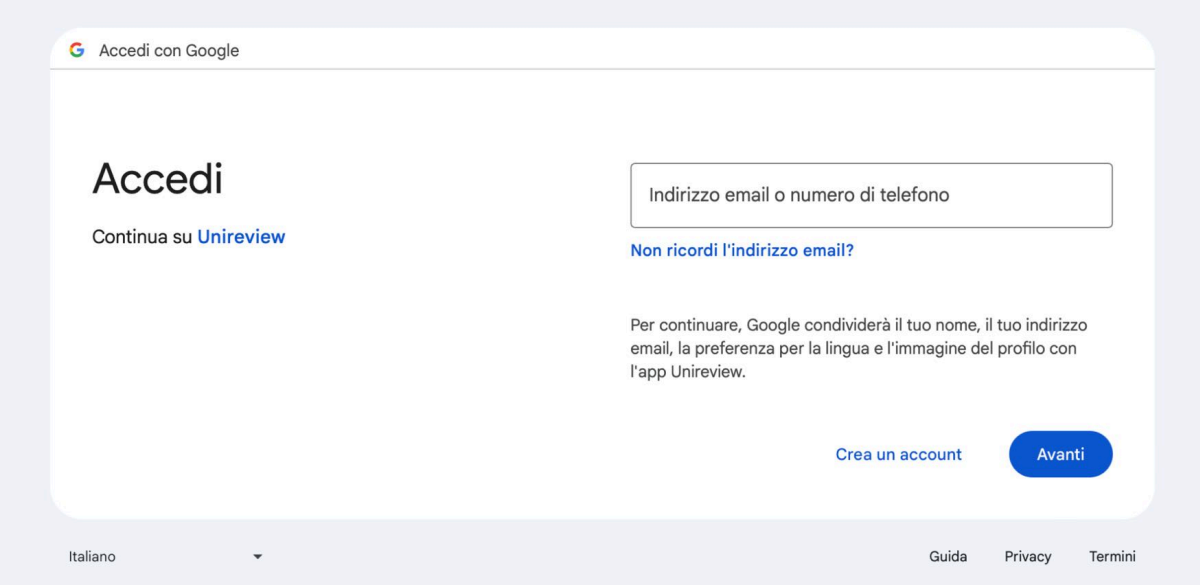
Auth API

Questa categoria comprende tutti gli endpoint destinati a permettere il login e l'autenticazione degli utenti alla piattaforma con l'utilizzo dello standard OAuth Google.

Come definito in RF1 documento D1 viene permesso agli utenti di registrarsi mediante le proprie credenziali di Ateneo, senza bisogno di creare un nuovo account apposito per la piattaforma.

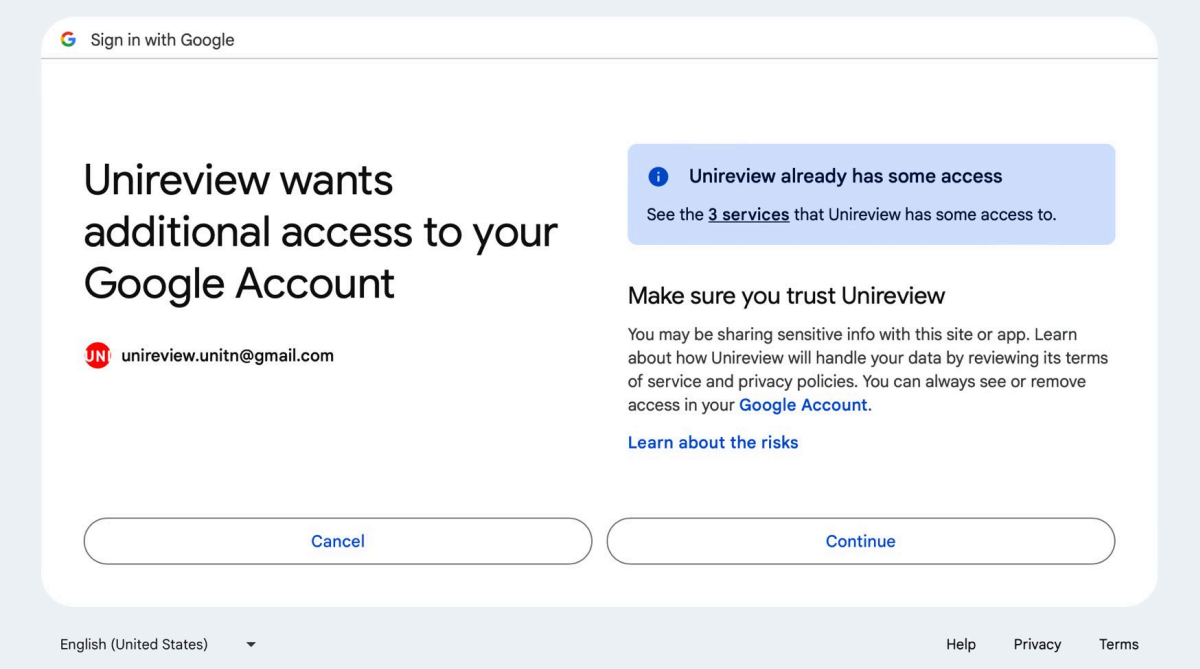
Accesso con Google GET /auth/google

Una volta raggiunto questo endpoint, l'utente verrà reindirizzato ad una pagina che gli permetterà di eseguire l'accesso con le credenziali del suo account Google accademico, esattamente come farebbe per accedere ad altri servizi di ateneo come Gmail.



The screenshot shows the Google login interface for Unireview. At the top, it says "Accedi con Google". The main heading is "Accedi" with a subtext "Continua su Unireview". There is a text input field for "Indirizzo email o numero di telefono". Below the field, there is a link "Non ricordi l'indirizzo email?". A paragraph explains that Google will share the user's name, email address, language preference, and profile picture with the Unireview app. At the bottom right, there are two buttons: "Crea un account" and "Avanti". The footer includes the language "Italiano" and links for "Guida", "Privacy", and "Termini".

Dopo aver inserito email e password, viene chiesta l'autorizzazione di condividere con Unireview il proprio email e il nome utente. Queste informazioni verranno utilizzate dalla piattaforma per identificare l'utente e assegnargli un nome utente.



The screenshot shows the Google account authorization screen. The heading is "Unireview wants additional access to your Google Account". Below this, the email "unireview.unitn@gmail.com" is displayed. A blue box contains the message "Unireview already has some access" and a link to "See the 3 services that Unireview has some access to." The section "Make sure you trust Unireview" explains that sensitive information may be shared and provides a link to "Learn about the risks". At the bottom, there are two buttons: "Cancel" and "Continue". The footer shows the language "English (United States)" and links for "Help", "Privacy", and "Terms".

Dopo aver concesso l'autorizzazione e quindi acconsentito l'informativa privacy, l'utente verrà automaticamente reindirizzato alla home page dell'applicazione, in cui risulterà correttamente registrato e abilitato a tutte le azioni disponibili.

Durante questo processo, il server emetterà un token JWT firmato e assegnato all'utente.

Sarà poi questo token, che inserito come Header delle richieste HTTP, permetterà all'utente di eseguire tutte le operazioni che richiedono l'autenticazione.

Email API

Questa categoria comprende le route che rendono disponibile un servizio di comunicazione di posta elettronica tra gli utenti e i moderatori della piattaforma. Questo è reso possibile dal servizio esterno *NodeMailer* che permette l'inoltro di email attraverso i servizi offerti da Google, come dichiarato nel diagramma delle componenti al punto 4.1.6 .

Invio di una email di segnalazione ai moderatori POST `/api/send-email`

Richiede di aver eseguito il login.

Questa api permette agli utenti di contattare e inviare una segnalazione ai moderatori della piattaforma.

Viene richiesto di inserire nel *body* della richiesta il testo della segnalazione, come segue:

```
Unset
{
  "text": "Ho bisogno di aiuto con la mia recensione."
}
```

Una volta inviata, i moderatori riceveranno una notifica contenente non solo il testo del messaggio, ma anche l'autore della richiesta, in modo da poterlo identificare e ricontattare.

User API

Questa categoria comprende tutte le api relative alla gestione degli utenti e delle loro funzioni.

Restituisci tutti gli utenti GET /api/user

Richiede di aver eseguito il login.

Il comportamento di questa api varia a seconda di chi la esegue, se ad invocarla è un moderatore, restituisce una lista che comprende tutti gli utenti e le relative informazioni della piattaforma. Se ad invocarla è invece un semplice utente, verrà restituito solo il suo profilo con le sue informazioni personali.

Viene consultato il database MongoDB per effettuare la ricerca delle informazioni mediante il metodo `find()`.

Restituisci le informazioni per un utente GET /api/user/:email

Richiede di aver eseguito il login.

Questa api richiede un parametro *email* che serve a selezionare l'utente di cui recuperare le informazioni dal database attraverso il metodo `findOne()`.

Vengono restituite le informazioni richieste solo nel caso in cui l'utente che effettua la richiesta è un moderatore oppure è l'account stesso di cui vengono richieste le informazioni.

Non sarà pertanto possibile ad un utente generico ricevere informazioni personali relative ad un altro utente, qualora una richiesta di questo tipo fosse ricevuta dal server, esso risponderebbe con un'autorizzazione negata.

Modifica dei dati di un utente PATCH /api/user/:email

Può essere eseguita solo da un moderatore.

Questa api rende disponibile la modifica dei dati di un utente per un moderatore della piattaforma, attraverso la richiesta di un parametro *email*.

Unset

```
{ "email": "string",  
  "moderatore": true,  
  "nomeUtente": "string",  
  "bannedUntil": "2024-07-06",  
  "linguaUI": true,  
  "temaUI": true }
```

Unireview – D4

La modifica dei dati di un utente deve avvenire inoltrando un oggetto json con i dati aggiornati come body della richiesta.

Qualora l'utente per cui viene richiesta una modifica dei dati non esistesse in piattaforma, verrà restituito un errore 404 not found.

Cancellazione dei dati di un utente DELETE /api/user/:email

Può essere eseguita da un utente autenticato o da un moderatore.

Questa api rende disponibile agli utenti la cancellazione dei propri dati dal database della piattaforma. Viene data inoltre la possibilità ai moderatori di cancellare i dati di qualunque utente nel database.

Deve venire passato come parametro l'indirizzo email dell'utente da cancellare.

Review API

Questa categoria comprende tutte le api relative alla creazione, cancellazione e gestione delle recensioni, oggetto principale della piattaforma Unireview.

Creazione e pubblicazione di una nuova recensione POST /api/review

Richiede di aver eseguito il login.

Questa api permette la compilazione di una nuova recensione da pubblicare sulla piattaforma.

Devono venire passati come corpo della richiesta almeno i campi obbligatori di una recensione come nel modulo seguente:

```
Unset
{
  "professore": "string", [required]
  "esame": "string", [required]
  "valutazioneProfessore": 5, [required]
  "valutazioneFattibilita": 5, [required]
  "valutazioneMateriale": 5, [required]
  "testo": "string",
  "tentativo": 0,
  "voto": 0,
  "frequenza": "0%", [required]
  "anonima": true
}
```

La struttura della richiesta è in completo accordo con il [modello Review](#) descritto in precedenza, tenendo a mente che alcuni parametri della recensione vengono compilati automaticamente dal server al momento della pubblicazione.

Qualora vi fosse un errore nell'inserimento di uno o più parametri, questo verrà segnalato con un messaggio di errore 400 che riporterà non solo quali sono i parametri che lo hanno generato, ma anche quali tipi di dati dovrebbero venire inseriti.

Restituisci tutte le recensioni presenti in piattaforma GET `/api/review`

Questa è una delle api che è invocabile senza dover effettuare l'accesso alla piattaforma come descritto in RF2 documento D1, infatti, è possibile a chiunque consultare le recensioni senza registrarsi.

Viene restituita una lista di tutte le recensioni presenti in piattaforma, indipendentemente dall'esame relativo o dal professore.

Restituisci le recensioni di un utente GET `/api/review/:email`

Questa api permette di restituire una lista di tutte le recensioni pubblicate da un utente specificato tramite il parametro *email* che deve essere presente nella richiesta HTTP.

Qualora l'utente non dovesse avere pubblicato alcuna recensione, verrà restituito un messaggio relativo con errore 404.

Restituisci tutte le recensioni per un professore GET `/api/review/professore/:professore`

Questa api restituisce una lista di recensioni che coinvolgono uno specifico professore, indicato dal parametro *professore* passato nella richiesta.

Qualora non dovessero essere presenti recensioni relative al professore richiesto nel database, verrà restituito un messaggio di errore 404.

Restituisce tutte le recensioni per un esame GET `/api/review/esame/:esame`

Questa api restituisce una lista di recensioni che coinvolgono uno specifico esame, indicato dal parametro *esame* passato nella richiesta.

Qualora non dovessero essere presenti recensioni relative all'esame richiesto nel database, verrà restituito un messaggio di errore 404.

Elimina una recensione DELETE `/api/review/:reviewId`

Può essere eseguita da un utente autenticato o da un moderatore.

Questa api permette ad un utente di cancellare solo le proprie recensioni, mentre permette ai moderatori di cancellare qualunque recensione, passando l'id della recensione come parametro della richiesta.

Modifica di una recensione PATCH `/api/review/:reviewId`

Può essere eseguita da un utente autenticato o da un moderatore.

Questa api permette ad un utente di modificare i dati delle proprie recensioni, mentre permette ai moderatori di modificare qualunque recensione, passando l'id della recensione come parametro della richiesta.

Exam API

In questa categoria ricadono le api che permettono di gestire gli esami per cui è ammesso pubblicare una recensione.

Ottieni gli esami GET `/api/exam`

Questa api restituisce una lista di tutti gli esami per cui è attualmente permesso pubblicare una recensione sulla piattaforma.

Aggiungi un esame POST `/api/exam`

Può essere eseguita solo da un moderatore.

Questa api permette ai moderatori di aggiungere un esame all'elenco utilizzando la seguente struttura:

```
Unset
{
  "professore": "string",
  "esame": "string"
}
```

In pieno accordo con il [modello exam](#), viene effettuata un'associazione tra il nuovo esame e il professore che tiene il relativo corso.

Cancella un esame DELETE /api/exam/:esame

Può essere eseguita solo da un moderatore.

Questa api permette ai moderatori di eliminare un esame dall'elenco specificando il nome dell'esame come parametro della richiesta.

Restituisci gli esami di un professore GET /api/exam/professore/:professore

Questa api restituisce un elenco degli esami trovati nel database tenuti dal professore inoltrato come parametro.

Restituisci gli esami per nome GET /api/exam/esame/:esame

Questa api restituisce un elenco degli esami trovati nel database con il nome inoltrato come parametro della richiesta.

Documentazione API

La documentazione interattiva in standard OpenAPI è stata realizzata mediante il modulo Swagger UI Express.

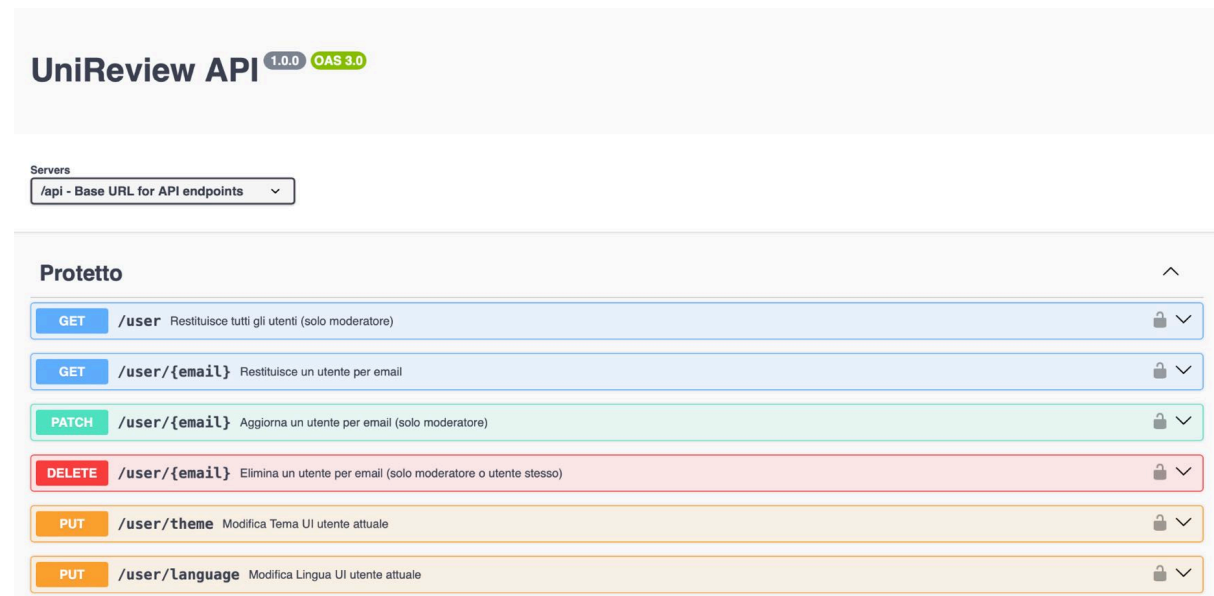
La documentazione è raggiungibile da internet accedendo al seguente link:

<https://unireview.up.railway.app/api-docs>

Oppure in locale eseguendo l'applicazione sul proprio computer attraverso questo indirizzo:

<http://localhost:8080/api-docs>


Nella documentazione OpenAPI vengono raccolti non solo tutti gli endpoint descritti nel paragrafo precedente, ma anche i modelli utilizzati nel database, anch'essi descritti precedentemente.



La schermata principale ci avvisa innanzitutto che il basepath è `/api` per tutti gli endpoint seguenti, pertanto se vorremo effettuare una richiesta `GET /user` per elencare tutti gli utenti della piattaforma, dovremo ricordarci di apporre la basepath `GET /api/user`.

Vediamo inoltre una divisione in due macro categorie, una dedicata ad ospitare le route protette da autenticazione e l'altra che comprende tutte le altre, che non richiedono particolari permessi per accedervi.

Come dichiarato RF2 documento D1, alcuni endpoint sono raggiungibili anche da utenti che non si sono registrati alla piattaforma. Ad esempio per ottenere la lista di tutte le recensioni presenti non è richiesta alcuna autenticazione.

Questa differenziazione è visibile anche notando il simbolo di un lucchetto  accanto ad ogni route protetta.

Ogni route possiede la definizione del metodo HTTP (`GET`, `DELETE`, `POST`, ...), una breve descrizione testuale, eventuali parametri richiesti e un esempio di corpo della richiesta (*body*) che chiarisce i tipi di dato ammessi. Vengono elencate infine le response HTTP che l'api può restituire, associate ad una breve descrizione.

POST /review Componi nuova recensione (richiede autenticazione)

Parameters

Try it out

No parameters

Request body

application/json

Example Value

Schema

```
{
  "professore": "string",
  "esame": "string",
  "valutazioneProfessore": 5,
  "valutazioneFattibilita": 5,
  "valutazioneMateriale": 5,
  "testo": "string",
  "tentativo": 0,
  "voto": 0,
  "frequenza": "0%",
  "anonima": true
}
```

Responses

Code	Description	Links
201	Recensione creata con successo	No links
400	Errore nell'inserimento dei dati	No links
401	Token mancante, autorizzazione negata	No links
500	Errore del server	No links

Nell'immagine precedente vi è l'esempio della route POST /api/review che permette di comporre una recensione da pubblicare sulla piattaforma. Viene chiaramente mostrato che per questa api è richiesta l'autenticazione, come descritto in RF2.2.1 documento D1.

È presente anche un esempio in formato json di una possibile recensione di cui è possibile visualizzare lo schema (modello) premendo su *Schema*, accanto a *Example Value*.

Tra le possibili risposte vi sono il successo della richiesta (201), l'errore nell'inserimento dei dati (400), l'autenticazione fallita (401) e il generico errore lato server (500).

L'ultima sezione della pagina è dedicata ai modelli che stabiliscono la struttura degli oggetti utilizzati dal database.

Schemas

^

User

{

email

▼

string

moderatore

▼

boolean

nomeUtente

▼

string

bannedUntil

▼

string(\$date)

linguaUI

▼

boolean

temaUI

▼

boolean

}

Review

>

Exam

>

Nell'immagine precedente viene mostrato il modello *User* che si occupa di definire la struttura salvata nel database per ogni utente che decide di registrarsi alla piattaforma.

Per ogni parametro dell'oggetto viene anche dichiarato il tipo di dato ammesso.

Authentication Middleware

Un componente importante per il funzionamento complessivo dell'applicazione è il middleware che effettua la verifica del token di autenticazione per le funzioni che richiedono di aver eseguito l'accesso.

Il componente è unico e comune per tutte le api protette.

Il suo compito è quello di verificare innanzitutto se la richiesta HTTP ricevuta comprende il token JWT nell'Header *Authentication*.

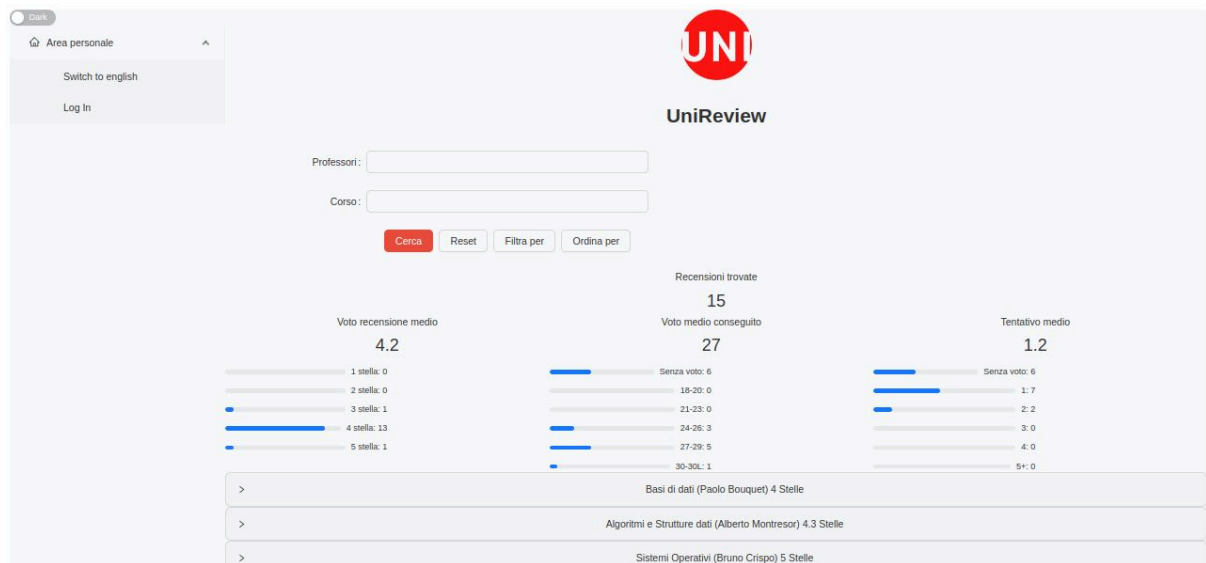
Dopodiché verifica che il token sia valido, firmato e attualmente non scaduto.

Se tutte le verifiche precedenti hanno avuto esito positivo, la richiesta viene autorizzata e viene dato l'accesso alle risorse richieste.

Se invece anche solo un requisito non viene rispettato, l'accesso viene negato e viene restituito un messaggio di errore 401 UNAUTHORIZED.

FrontEnd implementation

Questa sezione comprende tutte le principali schermate dell'applicazione Unireview, con una descrizione dei suoi elementi e delle azioni effettuabili dall'utente.

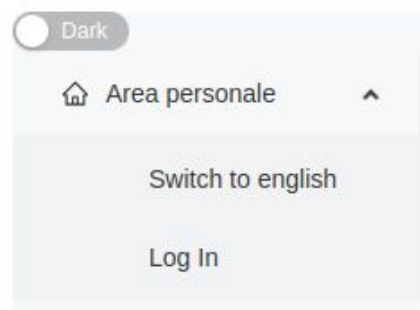


Landing Page

La prima schermata con cui l'utente avrà a che fare è la landing page.

In questa schermata abbiamo due elementi principali, una barra laterale a sinistra e l'elenco delle recensioni al centro.

In un primo momento vengono mostrate solo le azioni non protette, in quanto l'utente non ha ancora effettuato l'accesso.



Barra laterale

Sulla barra di sinistra in alto viene mostrato il pulsante che permette di cambiare il tema cromatico dell'interfaccia, permettendo di personalizzarlo tra colori chiari *light mode* e colori scuri *dark mode*, adatto per le ore serali.

Voto recensione medio		Voto medio conseguito		Tentativo medio	
1 stella: 0	2 stella: 0	Senza voto: 6	18-20: 0	Senza voto: 6	1: 7
3 stella: 1	4 stella: 13	21-23: 0	24-26: 3	2: 2	3: 0

Dark mode

Subito sotto è presente l'area personale dell'utente che propone l'opzione del cambio lingua, per passare alla versione dell'interfaccia tradotta in inglese e il pulsante login, che permette di iniziare la procedura di accesso.

Al centro dello schermo vengono mostrate, inizialmente, tutte le recensioni presenti in piattaforma, ordinate in modo anticronologico, accompagnate da dei parametri di statistica riassuntivi.

Sempre in questa sezione è presente la barra di ricerca per professori e corsi, in cui è possibile cercare l'esame di interesse premendo sull'apposito pulsante *Cerca*.

Oltre al pulsante *Reset*, che permette di reimpostare i parametri di ricerca, sono presenti i pulsanti *Filtra per* e *Ordina per* che permettono rispettivamente di filtrare le recensioni per contenuto e di ordinarle secondo dei parametri personalizzati.

Professori:

Corso:

Cerca Reset Filtra per Ordina per

Barre di ricerca

Unireview – D4

Nel momento in cui viene selezionata una barra di ricerca, vengono mostrati i relativi suggerimenti.

Nel caso della barra *Professore*, verranno suggeriti dei nomi di professori di cui cercare un esame, come analogamente accade per il *Corso*.

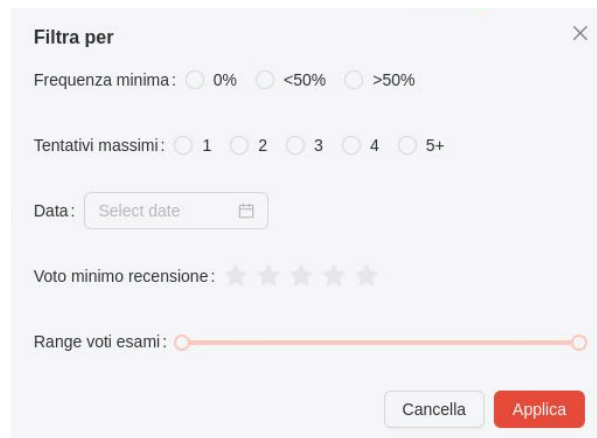
Suggerimenti di ricerca

Nel momento in cui viene selezionato un Professore, nella sezione *Corso*, verranno proposti solamente i corsi tenuti dal suddetto professore, per facilitarne la ricerca. La medesima logica vale anche per la sezione *Corso*.

Restringimento della ricerca

Il pop-up che viene visualizzato una volta premuto sul pulsante *Filtra per* comprende i seguenti parametri per filtrare i risultati della ricerca:

- Selezione della frequenza alle lezioni dello studente
- Numero di tentativi per superare l'esame
- Data della recensione
- Voto minimo della recensione
- Selettore intervallo di voti conseguiti dagli studenti



The 'Filtra per' modal window contains the following elements:

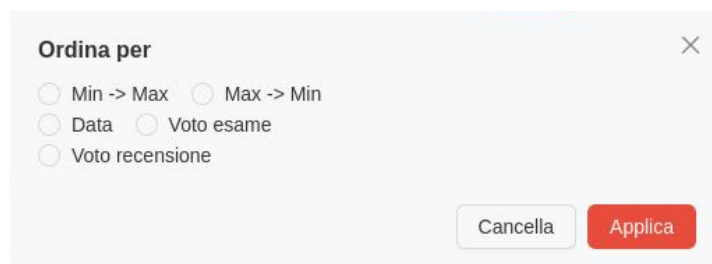
- Filtra per** (title) and a close button (X).
- Frequenza minima:** Radio buttons for 0%, <50%, and >50%.
- Tentativi massimi:** Radio buttons for 1, 2, 3, 4, and 5+.
- Data:** A date selection field with a calendar icon.
- Voto minimo recensione:** Five star icons.
- Range voti esami:** A horizontal range slider.
- Buttons:** 'Cancella' (Cancel) and 'Applica' (Apply).

Schermata "Filtra per"

Il pop-up visualizzato dopo aver premuto sul pulsante *Ordina per* comprende dei parametri su cui agire per modificare l'ordine della presentazione dei risultati della ricerca.

Permette di ordinare in modo crescente o decrescente:

- Data della recensione
- Il voto conseguito dallo studente all'esame
- Il voto complessivo della recensione



The 'Ordina per' modal window contains the following elements:

- Ordina per** (title) and a close button (X).
- Sort direction:** Radio buttons for 'Min -> Max' and 'Max -> Min'.
- Sort criteria:** Radio buttons for 'Data', 'Voto esame', and 'Voto recensione'.
- Buttons:** 'Cancella' (Cancel) and 'Applica' (Apply).

Schermata "Ordina per"

Dopo aver premuto il pulsante *Login* presente nella barra laterale sinistra, l'utente verrà reindirizzato ad una pagina che gli permetterà di eseguire l'accesso con le credenziali del suo account Google unitn.

The screenshot shows a Google login interface for the Unireview application. At the top, it says "Accedi con Google". The main heading is "Accedi". Below it, there is a link "Continua su Unireview". To the right, there is a text input field labeled "Indirizzo email o numero di telefono". Below the input field, there is a link "Non ricordi l'indirizzo email?". Further down, a paragraph explains that Google will share the user's name, email, language preference, and profile picture with the Unireview app. At the bottom right, there are two buttons: "Crea un account" and "Avanti". At the bottom left, there is a language selector set to "Italiano". At the bottom right, there are links for "Guida", "Privacy", and "Termini".

Dopo aver inserito email e password, viene chiesta l'autorizzazione di condividere con Unireview il proprio email e il nome utente.

The screenshot shows a Google account authorization page. The heading is "Unireview wants additional access to your Google Account". Below the heading, there is a small icon and the email address "unireview.unitn@gmail.com". To the right, there is a blue box with an information icon and the text "Unireview already has some access" and "See the 3 services that Unireview has some access to." Below this, there is a section titled "Make sure you trust Unireview" with a paragraph explaining that sensitive information may be shared and that users should review the terms of service and privacy policies. There is a link "Learn about the risks". At the bottom, there are two buttons: "Cancel" and "Continue". At the bottom left, there is a language selector set to "English (United States)". At the bottom right, there are links for "Help", "Privacy", and "Terms".

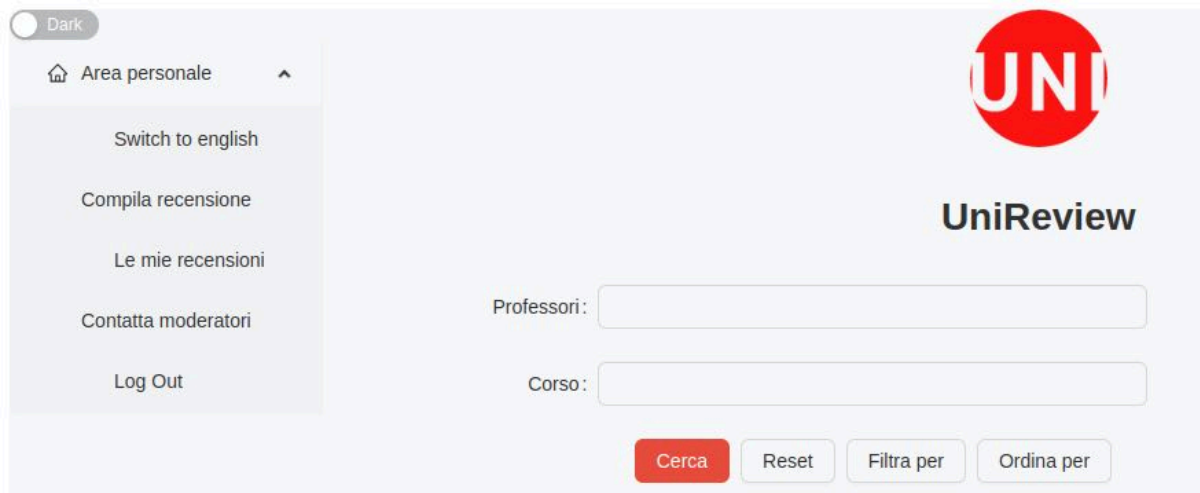
Dopo aver concesso l'autorizzazione, l'utente verrà automaticamente reindirizzato alla landing page dell'applicazione, in cui risulterà correttamente registrato e abilitato a tutte le azioni disponibili.

A questo punto la sezione *Area Personale* nella barra sinistra sarà aggiornata con le nuove funzionalità disponibili.

Oltre ai pulsanti presenti in precedenza, è presente un pulsante *Compila recensione*, che permette di scrivere una nuova recensione, un pulsante *Le mie*

Unireview – D4

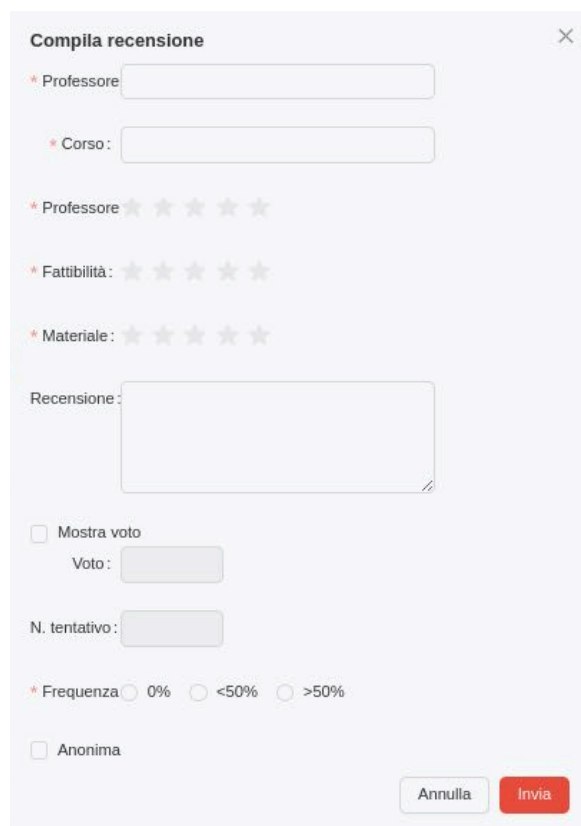
recensioni, che visualizza una lista delle recensioni compilate in precedenza dall'utente e un pulsante *Contatta i moderatori*, che permette di inviare una segnalazione ai moderatori del sistema.



The screenshot shows the UniReview landing page. On the left, there is a sidebar menu with a 'Dark' toggle at the top. The menu items are: 'Area personale' (with a home icon), 'Switch to english', 'Compila recensione', 'Le mie recensioni', 'Contatta moderatori', and 'Log Out'. The main area features the UniReview logo (a red circle with 'UNI' in white) and the text 'UniReview'. Below the logo, there are two input fields labeled 'Professori:' and 'Corso:'. At the bottom right, there are four buttons: 'Cerca' (red), 'Reset', 'Filtro per', and 'Ordina per'.

Landing page dopo aver effettuato l'accesso

Come detto, il pulsante *Compila recensione*, permette di compilare una nuova recensione attraverso un form che viene visualizzato dopo averlo premuto.



The screenshot shows a 'Compila recensione' (Write review) pop-up form. It has a title bar with a close button (X). The form contains several fields and options: 'Professore:' (with a red asterisk), 'Corso:' (with a red asterisk), 'Professore:' (with a red asterisk and five stars), 'Fattibilità:' (with a red asterisk and five stars), 'Materiale:' (with a red asterisk and five stars), 'Recensione:' (a large text area), 'Mostra voto' (checkbox), 'Voto:' (input field), 'N. tentativo:' (input field), 'Frequenza:' (with a red asterisk and three radio buttons: 0%, <50%, >50%), and 'Anonima' (checkbox). At the bottom right, there are two buttons: 'Annulla' and 'Invia' (red).


Pop-up per la compilazione di una recensione

Unireview – D4

Tra i diversi parametri, solo alcuni sono obbligatori e vengono contrassegnati con un asterisco rosso.

Qualora l'utente decidesse di inserire il voto conseguito all'esame, dovrebbe allora inserire anche il numero del tentativo dopo il quale ha ottenuto il voto. Se dovessero venire inseriti dei valori non validi, il sistema notificherebbe l'utente con un errore, accompagnato da un suggerimento per correggere l'errore.

Il pulsante *Contatta i moderatori*, una volta premuto, mostra un pop-up che permette di inviare una segnalazione ai moderatori del sistema.



The image shows a modal dialog box titled "Contatta moderatore" with a close button (X) in the top right corner. Inside the dialog, there is a label "Messaggio:" followed by a large, empty text input field with a red border. At the bottom right of the dialog, there are two buttons: "Cancel" and "OK".

Pop-up per contattare i moderatori

Premendo sul pulsante *Le mie recensioni*, viene mostrata una lista delle recensioni che l'utente ha già pubblicato in precedenza.



The screenshot displays the user interface for "Le mie recensioni". On the left is a sidebar menu with a "Dark" theme toggle and options: "Area personale", "Switch to english", "Compila recensione", "Cerca", "Contatta moderatori", and "Log Out". The main content area features the UNI logo and the title "Le mie recensioni". Below the title is a list of reviews, each with a right-pointing chevron icon and the review text:

>	Basi di dati (Paolo Bouquet) 4 Stelle
>	Algoritmi e Strutture dati (Alberto Montresor) 4.3 Stelle
>	Sistemi Operativi (Bruno Crispo) 5 Stelle
>	Ingegneria del software (Paolo Giorgini) 4.7 Stelle

Schermata *Le mie recensioni*

Premendo sul pulsante *Log Out* viene effettuata la disconnessione della sessione.

Cliccando su una recensione è possibile espanderla, rivelando maggiori informazioni su di essa.

Inoltre, qualora l'utente fosse l'autore della recensione, oppure un moderatore di sistema, gli verranno date anche le possibilità di eliminare o modificare la recensione.

Unireview - D4

Reti (Paolo Casari) 4.7 Stelle	Reti (Paolo Casari) 4.7 Stelle
Anonimo	Anonimo
Fattibilità	Fattibilità
★★★★☆	★★★★☆
Professore	Professore
★★★★★	★★★★★
Materiale	Materiale
★★★★★	★★★★★
Frequenza: >50%	Frequenza: >50%
	<button>Elimina</button> <button>Modifica</button>

L'utente non è l'autore

L'utente è l'autore

Premendo su *Modifica*, viene visualizzato un pop-up che permette la modifica dei dati precedentemente inseriti.

✕

Modifica recensione

Professore:

Corso:

Professore: ★★★★★

Fattibilità: ★★★★★

Materiale: ★★★★★

Recensione:

☐ Mostra voto
Voto:

N. tentativo:

* Frequenza ☐ 0% ☒ <50% ☐ >50%

☐ Anonima

Annulla

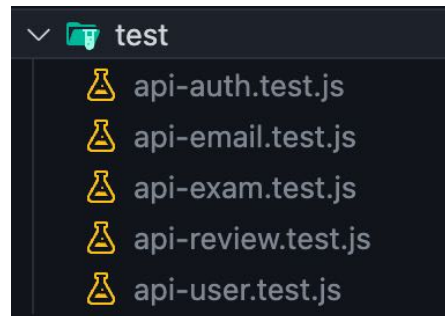
Modifica

Modulo precompilato per modificare la recensione

Testing

Per il testing dell'applicazione è stato utilizzato il modulo *Jest* e *superjest* per la verifica delle api.

Tutti i file che definiscono le casistiche di test sono raggruppati in una cartella *test/* presente nella directory *backend*.



Anche in questo caso è stata seguita una suddivisione logica, pari a quella utilizzate in tutte le precedenti sezioni di questo documento.

In alcuni casi è stato utilizzato il comando `/* istanbul ignore next */` per ignorare il test dell'errore generico lato server 500, in quanto non possibile da riprodurre artificialmente.

I test vengono eseguiti lanciando il comando `npm test` che esegue il seguente script:

```
node --experimental-vm-modules node_modules/jest/bin/jest.js --detectOpenHandles --coverage
```

dove:

`--experimental-vm-modules` è necessario poiché nel codice è stato utilizzato lo standard ES6 che utilizza *import* invece di *required*.

`--detectOpenHandles` rileva eventuali espressioni aperte che potrebbero prevenire la corretta terminazione dei test.

`--coverage` esegue un test di copertura generando un report complessivo riportato di seguito.

Risultati

I risultati dei test di copertura sono consultabili autonomamente nella cartella *coverage/* aprendo il file `index.html` dopo averla generata eseguendo `npm test`.

Unireview – D4

backend	<div><div></div></div>	100%	21/21	100%	0/0	100%	2/2	100%	21/21
backend/models	<div><div></div></div>	100%	6/6	100%	0/0	100%	0/0	100%	6/6
backend/routes	<div><div></div></div>	96.24%	205/213	82.85%	58/70	100%	26/26	96.22%	204/212
backend/services	<div><div></div></div>	100%	9/9	100%	3/3	100%	2/2	100%	9/9

La categoria più interessante è sicuramente quella data dall'insieme di api presenti in *backend/routes* che restituisce un valore di >96%, testando tutte le 26 funzioni che la compongono.

All files

96.78% Statements 241/249 **83.56%** Branches 61/73 **100%** Functions 30/30 **96.77%** Lines 240/248

Complessivamente vengono testate tutte le 30 funzioni del progetto in 5 file di test differenti per un totale di 44 *test cases*.

```
Test Suites: 5 passed, 5 total
Tests:      44 passed, 44 total
Snapshots:  0 total
Time:       9.96 s
Ran all test suites.
```

Deployment

L'applicazione viene resa accessibile pubblicamente mediante l'utilizzo di due servizi, Vercel per quel che concerne il frontend e Railway per il backend.

La home page dell'applicazione è raggiungibile da internet seguendo questo link:
<https://unireview-five.vercel.app>

Anche la documentazione relativa alle API è raggiungibile da internet accedendo al seguente indirizzo:

<https://unireview.up.railway.app/api-docs>

I file del progetto e tutta la sua cronologia di sviluppo è presente su Github al seguente indirizzo:

<https://github.com/G14-ABN/Unireview>

Esecuzione in locale

In questa sezione vengono trattate le istruzioni per eseguire l'applicazione sulla propria macchina in locale, senza l'utilizzo di servizi internet.

Per prima cosa è necessario scaricare l'ultima versione del progetto direttamente da github con il comando da terminale:

```
git clone https://github.com/G14-ABN/Unireview.git
cd Unireview/
```

`npm i` per installare le dipendenze dell'applicazione

Prima di avviare il programma è necessario creare due file con le variabili d'ambiente necessarie per eseguire il progetto.

`.env` nella cartella **backend**

```
ATLAS_URI="mongodb+srv://sasha:8EiUhrjjTVvKPohT@unireview.wzmoezy.mongodb.net/?retryWrites=true&w=majority"
PORT=8080
DB_NAME="unireview"
CLIENT_ID="115683779443-s5j34cf9dlapup77ic1qrq39es70blme.apps.googleusercontent.com"
CLIENT_SECRET="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"
FRONTEND_URI="http://localhost:3000"
CALLBACK_URI="http://localhost:8080/auth/google/callback"
JWT_SECRET="banana"
```

`.env.local` nella cartella **frontend**

```
NEXT_PUBLIC_BACKEND_URI="http://localhost:8080"
```

Dopodichè possiamo eseguire l'applicazione con il comando `npm start`

Visitando <http://localhost:3000> si accede direttamente alla home page dell'applicazione, da cui è possibile effettuare tutte le operazioni dall'interfaccia utente.

Visitando <http://localhost:8080/api-docs> si può consultare la documentazione in locale delle api presentate in precedenza.