PROGETTO DI INGEGNERIA DEL SOFTWARE

SKI ONLINE

Sviluppo applicazione

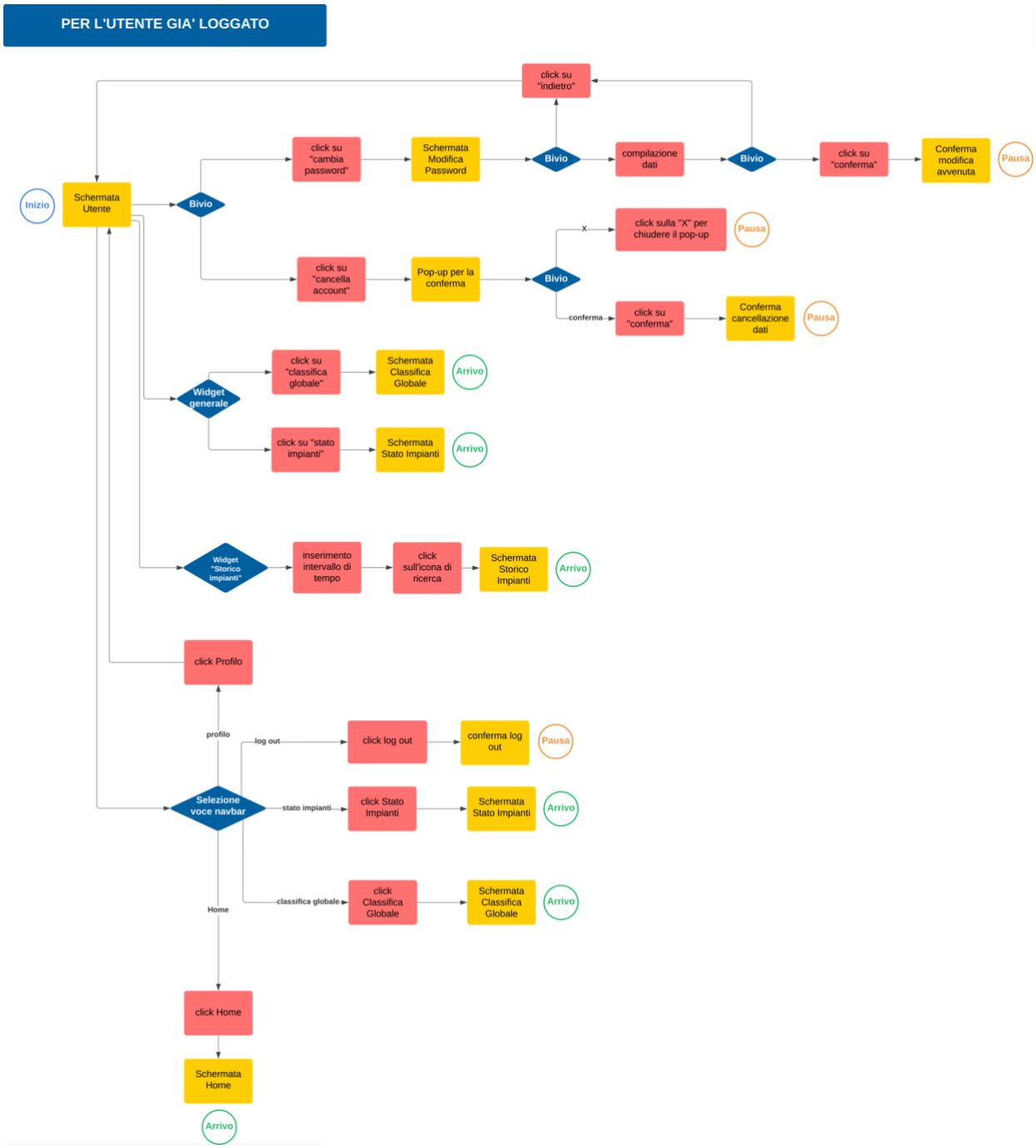
Indice dei contenuti

Scopo del documento

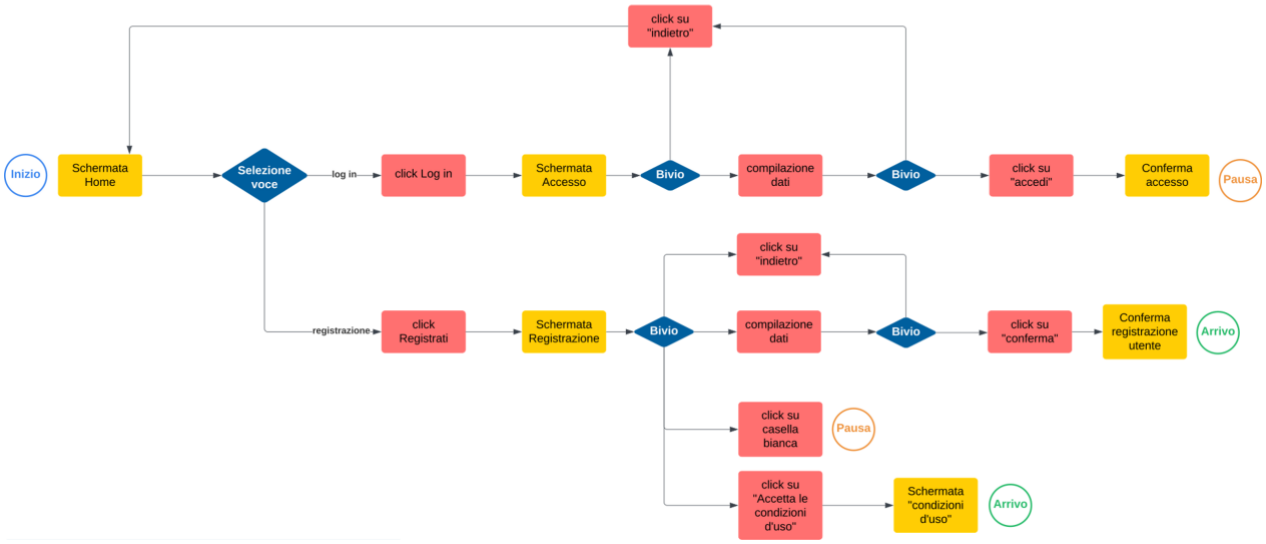
Il presente documento riporta tutte le informazioni necessarie per lo sviluppo di una parte dell'applicazione Ski Online. In particolare, presenta tutti gli artefatti necessari per realizzare i servizi di gestione *degli utenti e degli impianti di risalita dell'applicazione Ski Online*.
Partendo dalla descrizione degli user flow legate *alle azioni eseguibili dagli utenti anonimi, dagli utenti registrati e dagli utenti di sistema (operatore e gestore)*, il documento prosegue con la presentazione delle API necessarie (tramite l'API Model e il Modello delle risorse) per *effettuare il login, registrazione, cancellazione di un account, visualizzare lo stato di affollamento degli impianti e lo storico dei log degli accessi agli stessi* necessari all'applicazione Ski Online.
Per ogni API realizzata, oltre ad una descrizione delle funzionalità fornite, il documento presenta la sua documentazione e i test effettuati. Infine una sezione è dedicata alle informazioni del Git Repository e il deployment dell'applicazione stessa.

User flows

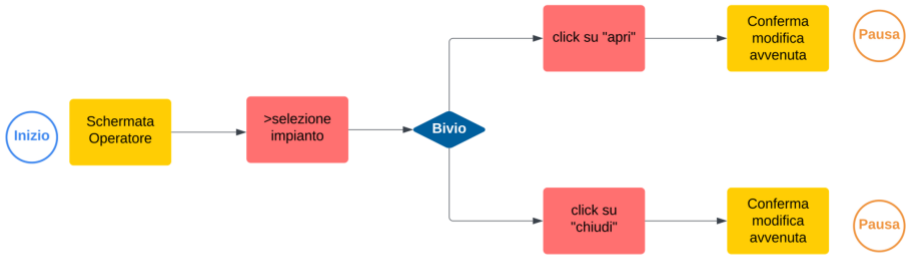
In questa sezione del documento di sviluppo riportiamo gli "user flows" per il ruolo sia dell'utente registrato che per quello anonimo.
La prima figura descrive gli user flows relativi alle operazioni di log-in e registrazione da parte dell'utente anonimo. Diversamente, la seconda si riferisce all'utente registrato e alle azioni che può effettuare a partire dalla "Schermata Utente" a lui dedicata; mentre, la terza descrive lo stato di apertura degli impianti gestita dall'Utente Gestore o Operatore.


PER L'UTENTE GIA' LOGGATO
Inizio
Schermata Utente
Bivio
click su "cambia password"
Schermata Modifica Password
Bivio
compilazione dati
Bivio
click su "conferma"
Conferma modifica avvenuta
Pausa
click su "indietro"
click su "cancella account"
Pop-up per la conferma
Bivio
X
click sulla "X" per chiudere il pop-up
Pausa
conferma
click su "conferma"
Conferma cancellazione dati
Pausa
Widget generale
click su "classifica globale"
Schermata Classifica Globale
Arrivo
click su "stato impianti"
Schermata Stato Impianti
Arrivo
Widget "Storico impianti"
inserimento intervallo di tempo
click sull'icona di ricerca
Schermata Storico Impianti
Arrivo
click Profilo
profilo
log out
click log out
conferma log out
Pausa
Selezione voce navbar
stato impianti
click Stato Impianti
Schermata Stato Impianti
Arrivo
Home
classifica globale
click Classifica Globale
Schermata Classifica Globale
Arrivo
click Home
Schermata Home
Arrivo

PER L'UTENTE NON ANCORA LOGGATO



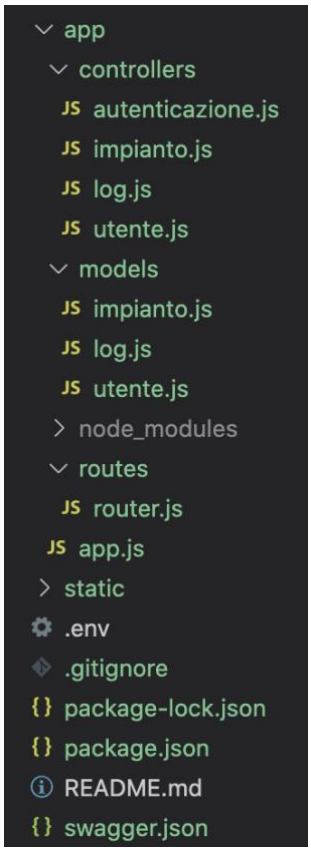
PER L'UTENTE OPERATORE/GESTORE



Application Implementation and Documentation

Project Structure

La struttura del progetto è presentata in figura ed è composta di una cartella API per la gestione delle API locali, di una cartella static per la parte del front-end, dei file di sistema come il file .env, .gitignore e package.json e del file swagger.json per la documentazione delle API che verrà discussa in seguto



Project Dependencies

I seguenti moduli Node sono stati utilizzati e aggiunti al file package.Json

- Bcrypt
- Dotenv
- Express
- Jsonwebtoken
- Mongoose
- Mongoose-express-api
- Swagger-ui-express

Project Data or DB

Per la gestione dei dati utili all'applicazione abbiamo definito tre principali strutture dati come illustrato in figura. Una collezione di "Utenti", una collezione di "Impianti" e una collezione di "Log" che lega il passaggio di un Utente per un Impianto.

LOGICAL DATA SIZE: 2.86KB STORAGE SIZE: 108KB INDEX SIZE: 108KB TOTAL COLLECTIONS: 3 CREATE COLLECTION

Collection Name	Documents	Logical Data Size	Avg Document Size	Storage Size	Indexes	Index Size	Avg Index Size
impiantos	3	256B	86B	36KB	1	36KB	36KB
logs	16	1.97KB	126B	36KB	1	36KB	36KB
utentes	3	658B	220B	36KB	1	36KB	36KB

Per rappresentare gli Utenti, Impianti e Log abbiamo definito i seguenti tipi di dati di esempio:

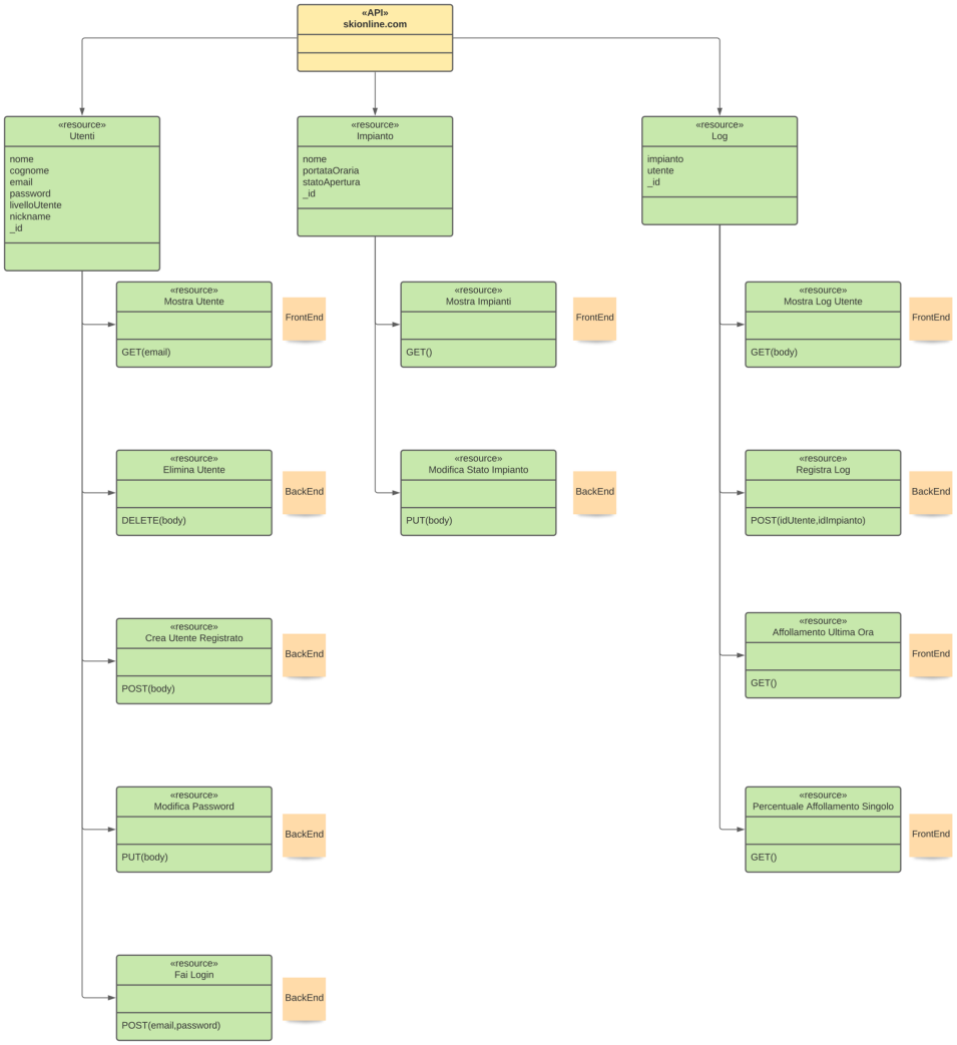
```
_id: ObjectId('639c64292f351b13f99e0137')
nome: "Mario"
cognome: "Rossi"
email: "mariorossi@gmail.com"
password: "$2b$05$K1za2US7NwGZOCwAIXrvlO.r2Eqy7.fzgD6V5gSifSOtaQjUY3xrG"
livelloUtente: "Base"
__v: 0

 _id: ObjectId('63a4de9eec8ee30dee706855')
 nome: "Piz Boe"
 portataOraria: 1500
 statoApertura: false
 __v: 0

_id: ObjectId('63a44607994b249035e12c57')
impianto: "63987e3b498173d611a5c667"
utente: "63972a8231b4484c9fe6eb97"
timestamp: 1671710215700
__v: 0
```

Project APIs

RESOURCES EXTRACTION FROM THE CLASS DIAGRAM



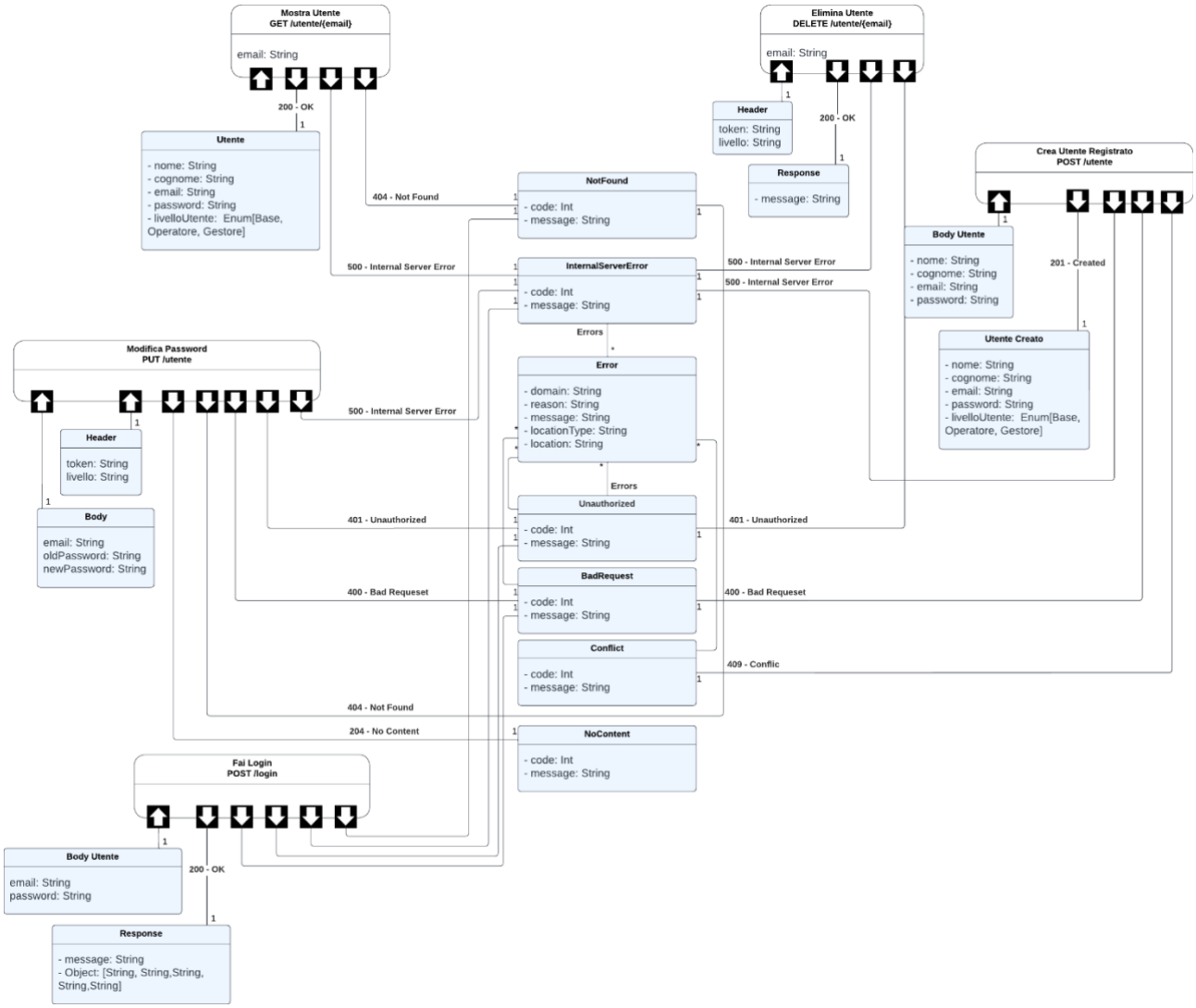
Nota bene: il parametron _id indicato nelle risorse si riferisce all'identificativo univoco generato da MongoDB all'atto di registrazione dei dati

RESOURCES MODELS

Basandoci sulla sezione precedente sono state realizzate cinque Resources Models suddivisi per tipologia.

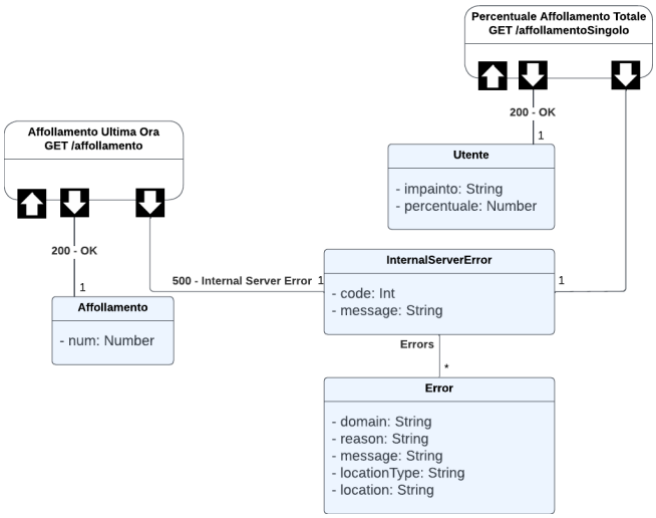
UTENTI

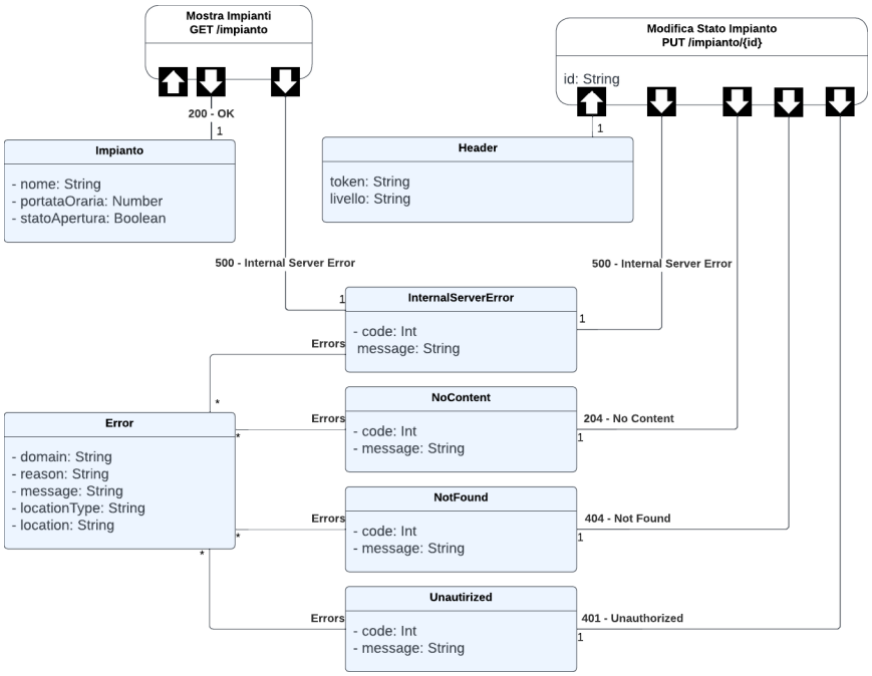
<descrizione>



IMPIANTI

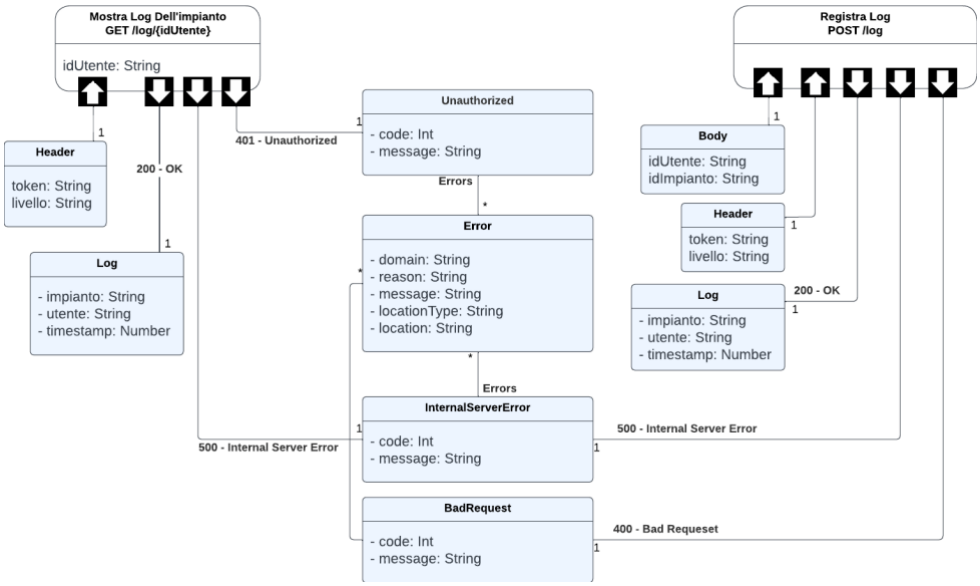
<descrizione>





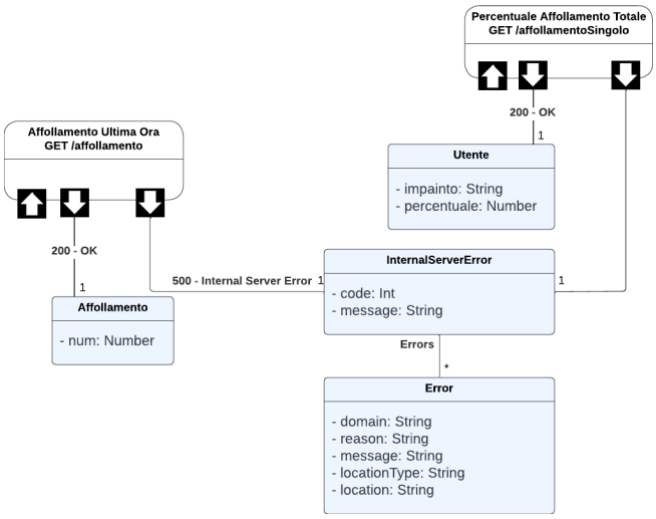
LOG

<descrizione>



AFFOLLAMENTO

<descrizione>



Sviluppo API

UTENTE

API documentation

Le API Locali fornite dall'applicazione Ski Online e descritte nella sezione precedente sono state documentate utilizzando il modulo NodeJS chiamato Swagger UI Express. In questo modo la documentazione relativa alle API è direttamente disponibile a chiunque veda il codice sorgente. Per poter generare l'endpoint dedicato alla presentazione delle API abbiamo utilizzato Swagger Swagger UI in quanto crea una pagina web dalle definizioni delle specifiche OpenAPI. In particolare, di seguito mostriamo la pagina web relativa alla documentazione che presenta le 3 API (GET, POST and DELETE) per la gestione dei dati della nostra applicazione. La GET viene utilizzata per visualizzare i dati in una pagina HTML. La POST per inserire un nuovo dato nel nostro sistema. La DELETE per cancellare un dato dal nostro sistema. L'endpoint da invocare per raggiungere la seguente documentazione è: http://localhost:8080/api-docs

Ski Online project

1.0.0

[Base URL: localhost:8080/]

Documentazione delle API Ski Online project

MIT

Schemes
HTTP

Utenti API per gli utenti che utilizzano il sistema

- GET /utente/{email} Mostra l'utenza associata ad uno specifico indirizzo email
- DELETE /utente/{email} Elimina l'utenza associata all'indirizzo email specificato
- POST /utente Crea un nuovo Utente di livello Utente Registrato nel sistema
- PUT /utente Modifica la password di un utente, richiedendo prima la password precedente alla modifica
- POST /login Effettua l'autenticazione di una utenza

Impianti API per la gestione degli impianti sciistici nella località sciistica

- GET /impianto Fornisce la lista di tutti gli impianti memorizzati
- PUT /impianto/{id} Modifica lo stato di apertura di un impianto da aperto a chiuso o vice versa

Log API per la gestione dei log degli accessi degli utenti agli impianti sciistici

- GET /log/{idUtente} Fornisce la lista di tutti i log di accesso ad un impianto effettuati da un utente
- POST /log Registra un nuovo log di passaggio di un utente per un impianto

Affollamento

- GET /affollamento Fornisce il numero di utenti con almeno un accesso ad un impianto nell'ultima ora
- GET /affollamentoSingolo Fornisce per ogni impianto la percentuale di affollamento (num accessi/portata max nell'ultima mezz'ora)

Models

- Utente
- Impianto
- Log

FrontEnd Implementation

GitHub Repository and Deployment info

Testing