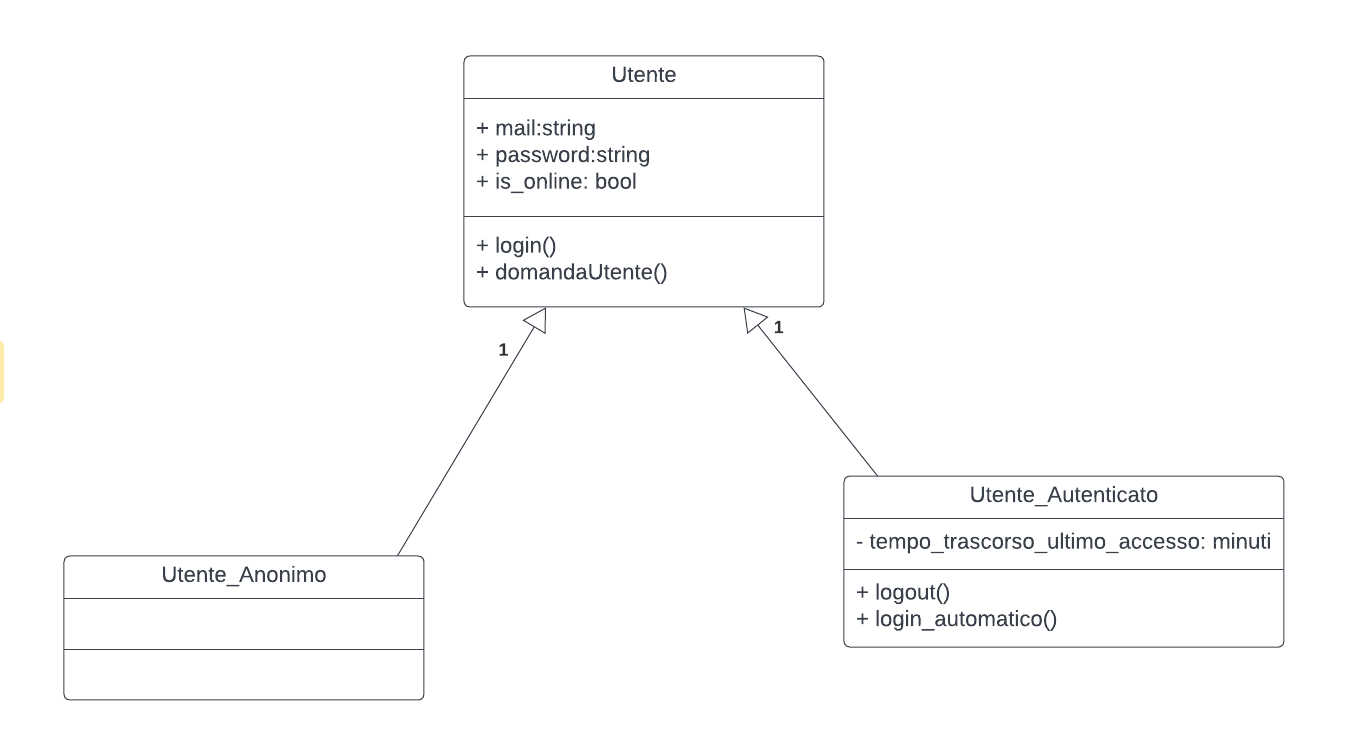
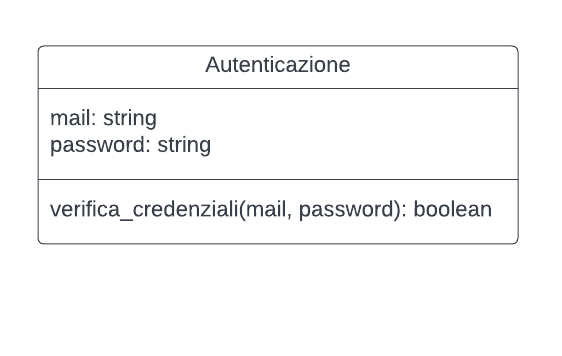
**Utenti**

Analizzando l’use case diagram realizzato per il progetto Yinco, si nota la presenza di due attori “utente anonimo” e “utente autenticato”. L’attore “utente anonimo” è colui che ha un ristretto numero di domande che può richiedere al sistema (cfr obiettivi D1), mentre “utente autenticato” è colui che ha un maggiore numero di domande che può richiedere al sistema. Entrambi questi attori hanno specifiche funzioni e attributi, ma hanno anche molto in comune. Sono state quindi individuate due classi “Utente\_Anonimo” e “Utente\_Autenticato” con attributi e funzioni specifici e una classe “utente” con funzioni e attributi in comune. Le classi “utenteAnonimo” e “utenteAutenticato” sono collegate alla classe “utente” tramite una generalizzazione. La classe utente ha come attributi l’email e la password dell’utente e un booleano, “is\_online” il quale verrà utilizzato per capire se l’utente sia autenticato o meno in quanto cambiano il modo in cui l’utente potrà interagire con il sistema. I metodi della classe “Utente” sono semplicemente un metodo “login” e un metodo “domanda\_utente”. La classe “Utente\_Anonimo” non ha ulteriori attributi o metodi in quanto sono tutti già specificati nella classe “Utente”. La classe “Utente\_Anonimo” invece ha un ulteriore attributo “tempo\_trascorso\_ultimo\_accesso” di tipo minuti il quale viene utilizzato per verificare una condizione del metodo “login\_automatico”. Il secondo metodo della classe “Utente\_Autenticato” è il metodo “logout”.



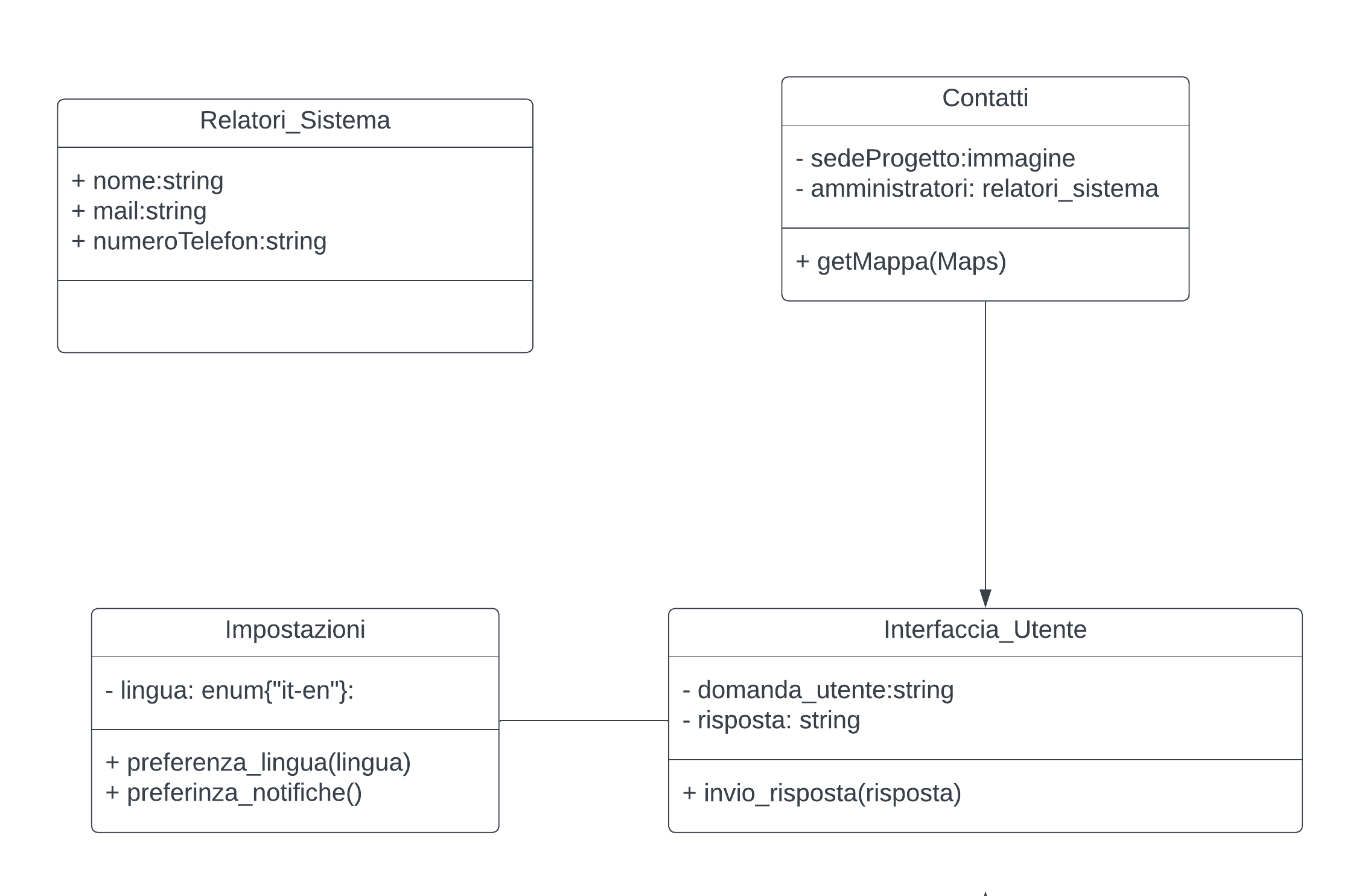
**Gestione autenticazione**

L’use case diagram analizzato presenta un sistema subordinato denominato “sistema credenziali universitarie”. Questo elemento rappresenta il meccanismo di autenticazione degli utenti attraverso un sistema esterno che gestisce le credenziali universitarie. È stata quindi identificata una classe “autenticazione” la quale si interfaccia con il sistema di gestione delle credenziali dell’ateneo. Il software sviluppato per il progetto Yinco non memorizzerà le e-mail e le password di ogni singolo utente, ma passerà i dati di autenticazione ad un sistema paritario esterno, il sistema di credenziali universitarie, che valuterà questi dati e risponderà specificando se le credenziali inserite sono valide o meno. Gli attributi di questa classe sono semplicemente la mail e la password dell’utente. Il metodo “verifica\_credenziali” verificherà se la mail e la password inserite siano corrette.



**Interfaccia utente**

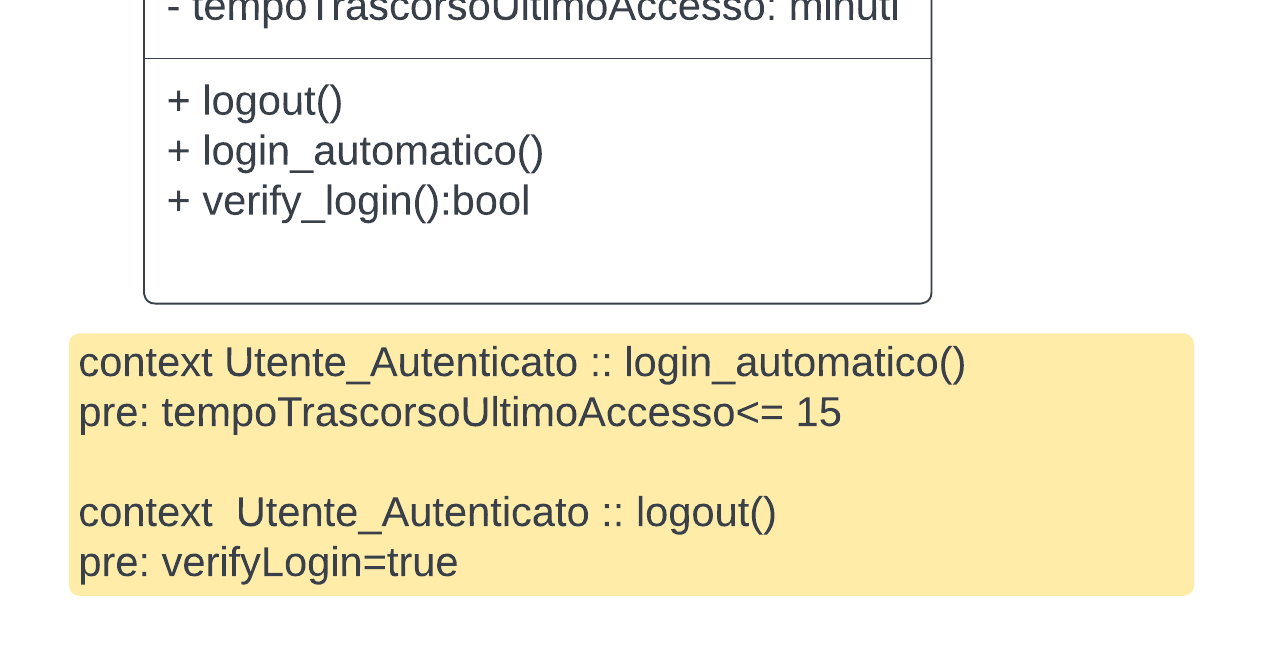
L’interfaccia utente è il cuore di questo progetto, in quanto da essa si può accedere alle impostazioni, si può interagire con il sistema Yinco e si può trovare informazioni riguardo ai relatori del sistema. Quindi, vista la complessità della componente, sono state create 4 classi opportunamente relazionate. La classe relatori sistema viene utilizzata come ausiliaria rispetto alla classe ‘Contatti’ ed è stata ideata per poter identificare il tipo dell’attributo “amministratori”. Nella classe contatti possiamo trovare due attributi: l’attributo “amministratori” di tipo relatori\_sistema e l’attributo “sede\_progetto” di tipo immagine. La sede del progetto viene ottenuta utilizzando il metodo “get\_mappa” utilizzando l’API di Google Maps. La classe “impostazioni” ha un attributo “lingua” del tipo enum in quanto l’utente avrà la possibilità di scegliere se preferisce avere il sistema in lingua italiana o inglese. Questa classe presenta due metodi, un metodo “preferenza\_lingua”, il quale prende come parametro l’attributo lingua e un metodo “preferenza\_notifiche”. Il metodo preferenza\_notifiche verrà poi utilizzato per capire se l’utente vuole ricevere nella propria mail istituzionale delle mail riguardanti gli esami e le tasse. Questa funzione verrà vista più approfonditamente ….(**Qui Daniele inserisci un riferimento alla tua parte**). Per finire la classe “Interfaccia\_Utente” ha due attributi: “domanda\_utente” e “risposta” entrambi di tipo string. Questi due attributi vanno a indicare ciò che il sistema riceverà (domanda\_utente) e invierà (risposta) quando l’utente anonimo o autenticato che sia interagirà con il sistema. Il metodo di questa classe (invio\_risposta) serve a poter inviare all’utente una risposta alla domanda ricevuta.



**Codice Object Costraint Language**

Login Automatico

Il login automatico da parte di un utente è possibile se e solo il tempo trascorso dall’ultimo accesso delll’utente in questione è inferiore a 15 minuti. Questa condizione è espressa in OCL attraverso una precondizione



Logout

Il logout da parte di un utente è possibile se e solo l’utente ha già effettuato il login. Questa condizione è espressa in OCL attraverso una precondizione

