



SAPIENZA
UNIVERSITÀ DI ROMA

Inline: queues management system

Facoltà di Ingegneria informatica e automatica
Degree program in Ingegneria Informatica

Candidato

Daniele Fedeli

Matricola 170945

Relatore

Prof. Leonardo Querzoni

Anno Accademico 2018/2019

*"Il computer non è una macchina intelligente che aiuta le persone stupide,
anzi, è una macchina stupida che funziona solo nelle mani delle persone
intelligenti." Umberto Eco*

Indice

1	Introduzione	1
1.1	Nascita dell'idea	1
1.2	Organizzazione della tesi	1
2	Raccolta e analisi dei requisiti	2
2.1	Descrizione della realtà di interesse	2
2.2	Funzionalità offerte	2
2.2.1	Sign-up e sign-in	3
2.2.2	Profilo	4
2.2.3	Room	6
2.2.4	Room host side	9
2.3	Dashboard	11
2.4	Messaggi privati	12
2.5	Admin	12
2.6	User stories	13
2.6.1	Sezione utente e profilo	14
2.6.2	Sezione accesso al sito web	14
2.6.3	Sezione messaggi privati	14
2.6.4	Sezione achievements	14
2.6.5	Sezione servizi esterni	14
2.6.6	Sezione chat	15
2.6.7	Sezione ricerca	15
2.6.8	Sezione notifica	15
2.6.9	Sezione room	15
2.6.10	Sezione dashboard	16
2.6.11	Sezione admin	16
2.7	Mockup	16
3	Descrizione tecnologie usate	20
3.1	Ruby e Ruby on Rails	20
3.2	Google calendar Api	21
3.3	Google e facebook oauth	21
3.4	Mapbox	22
3.5	Mailboxer	22

3.6	Bootstrap	22
3.7	Ice cube	23
3.8	Strumenti minori	23
3.9	Metodologie usate	23
4	Analisi e progettazione concettuale	24
4.1	Analisi dei requisiti	24
4.2	Diagramma ER	26
4.3	Dizionario dei dati: Entità	27
4.4	Dizionario dei dati: Relazioni	28
4.5	Dizionario dei dati: Attributi	28
4.6	Dizionario dei dati: vincoli di cardinalità	30
5	Progettazione del sistema	31
5.1	Introduzione all'UML	31
5.2	Architettura di riferimento	31
5.3	Diagramma delle classi	32
5.4	Diagramma degli stati	35
5.5	Diagramma delle attività	36
6	Test e validazione del software	38
6.1	Installazione	38
6.2	Attraverso unit e integration testing	39
6.2.1	Unit test	39
6.2.2	Integration test	40
7	Conclusioni	41

Elenco delle figure

2.1	Schermata sign up	3
2.2	Schermata sign in	4
2.3	Schermata reset password smarrita	4
2.4	Schermata profilo	5
2.5	Schermata achievements	6
2.6	Creazione stanza	7
2.7	Room con priority queue vista da un utente	9
2.8	Room vista da un room host	10
2.9	vista dell'edit room	10
2.10	Vista della dashboard	11
2.11	Vista dei messaggi privati	12
2.12	Vista della dashboard admin	13
4.1	Diagramma ER	27
5.1	Architettura MVC schematizzata	32
5.2	Diagramma UML delle classi	34
5.3	Diagramma degli stati	35
5.4	Diagramma attività dell'operazione SignIn	37

Elenco delle tabelle

4.1	Tabella delle entità	28
4.2	Tabella delle relazioni	28
4.3	Tabella degli attributi user	29
4.4	Tabella degli attributi room	29
4.5	Tabella degli attributi message	30

Capitolo 1

Introduzione

1.1 Nascita dell'idea

La tesi nasce per l'esame di **Architetture software e sicurezza informatica**, un laboratorio svolto dal Prof. **Leonardo Querzoni**. Con una rapida visione sul mercato di software abbiamo notato, io ed il mio team, la mancanza di un sistema per gestire delle code generiche.

Confrontandoci con le aziende nel campo IT abbiamo notato l'uso assiduo di *Google calendar*, prodotto della grande multinazionale *Google*. Nonostante la grandissima qualità del prodotto, non è possibile organizzare code, ma soltanto eventi a cui è possibile partecipare o promemoria. E' qui che nasce l'opportunità e la voglia di sviluppare qualcosa che ancora non esiste.

L'utente dopo essersi registrato può partecipare alla collettività di Inline, creando o partecipando a stanze in modo facile ed intuitivo. Le room posso essere private o pubbliche, a discrezione del creatore. Se l'utente mantiene la promessa di partecipazione guadagna un punto, visibile nella classifica globale del sito.

1.2 Organizzazione della tesi

Il documento attraverserà la fase di collezionamento dei requisiti, sviluppo e testing in modalità verbosa e, talvolta, accompagnata da grafici ER o User stories.

Ci soffermeremo sulle APIs scelte, menzionando i motivi di prevalenza rispetto ad altre ed i processi di inserimento all'interno del progetto. Menzioneremo tutte le user stories del progetto con mockup di riferimento, e la suddivisione dei compiti. Tutti i test sviluppati verranno affrontati.

Capitolo 2

Raccolta e analisi dei requisiti

2.1 Descrizione della realtà di interesse

L'idea è stata quella di creare una gestione delle code senza essere presenti fisicamente, in modo da sfruttare il proprio tempo in maniera più efficiente. Quante volte è capitato di stare in fila alla posta o dal medico? Persino ad un ricevimento di un professore. Tante. Il tempo è una delle poche risorse che l'essere umano non è in grado di riciclare.

L'obiettivo che ci siamo posti come team è di creare una piattaforma user-friendly, utile e accessibile. Chiunque può iscriversi e creare una stanza a cui gli altri iscritti possono partecipare, chattare, o scambiare posti in coda.

2.2 Funzionalità offerte

Le funzionalità offerte dall'applicazione sono molteplici e mirate a raggiungere gli scopi sopracitati. Ogni visitatore può iscriversi utilizzando la propria mail, attivando il proprio account tramite un link inviato via posta, oppure attraverso i più famosi social network come **Google** o **Facebook**.

Ogni iscritto avrà a propria disposizione una sezione profilo con all'interno un avatar, il contatore delle partecipazioni, un id hashato e un link che consente di invitare visitatori in modo da guadagnare punti.

Si può creare una stanza inserendo nome, descrizione, numero partecipanti e facoltativamente luogo. Il tutto può essere pubblico o privato, nella seconda alternativa si può accedere soltanto se si conosce il link della stanza.

Si ricompensano gli utenti che mantengono la loro promessa di frequentazione assegnandogli punti, esperienza e aumentando il loro contatore nella sezione profilo. Ovviamente è presente una sezione leaderboard che contiene la classifica globale del sito.

Le stanze possono essere ricercate tramite la barra di ricerca oppure trovandole nella mappa: per posizione personale oppure tramite località. Non

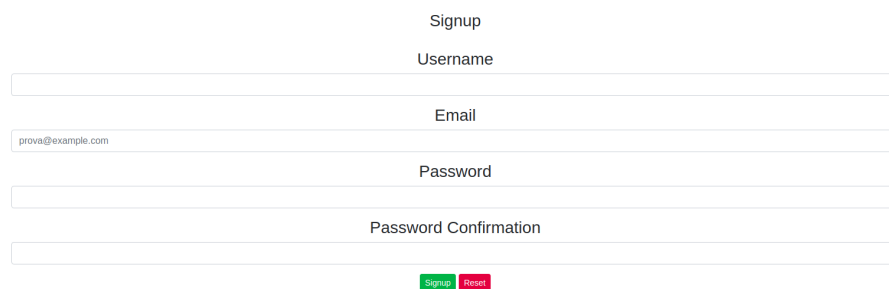
è possibile, ovviamente, prenotarsi per stanze piene ma, è possibile tramite messaggi privati, richiedere lo scambio posizione con chi è già in attesa.

Il sito prevede anche un servizio di messaggistica, come sopra citato, che può essere usato per chattare normalmente o per richiedere posti nelle code delle stanze che ci interessano.

2.2.1 Sign-up e sign-in

Nel sito si ha la possibilità di scegliere: essere un **guest** o **user**. Da ruoli diversi, derivano differenti poteri. Un guest ha la possibilità di vedere room e ricercarle tramite località, può controllare le code di una stanza ma, non può partecipare. La musica cambia nel caso in cui si autentica. Un utente ha tutti i poteri del guest e in più può creare room e partecipare, mandare messaggi e vedere la leaderboard.

La registrazione può essere effettuata tramite email, con password scelta dall'utente, oppure attraverso **Oauth** con i più famosi servizi: Google e Facebook. Nel caso si scelga la prima alternativa, il sistema genererà una mail di conferma che sarà inviata all'indirizzo immesso. Questa procedura non è necessaria nel caso si utilizzi l'oauth.



The image shows a sign-up form with the following fields and elements:

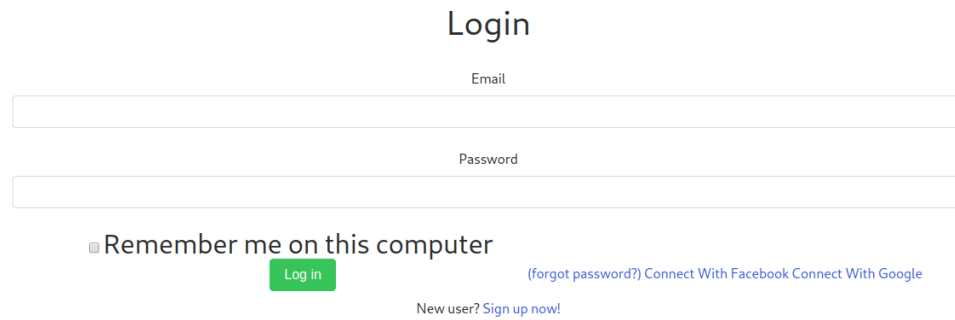
- Signup**: A heading centered above the form.
- Username**: A text input field.
- Email**: A text input field containing the placeholder text "prova@example.com".
- Password**: A text input field.
- Password Confirmation**: A text input field.
- Buttons**: Two buttons at the bottom, "Signup" (green) and "Reset" (red).

Figura 2.1. Schermata sign up

Le user-stories relative a questa sezione sono:

- 1.As an UNREGISTERED USER i want to SIGN UP WITH MY EMAIL so that i can BECOME A USER
- 2.As an UNREGISTERED USER i want to SIGN UP WITH MY FACEBOOK OR GMAIL ACCOUNT so that i can BECOME A USER

Per un utente già registrato ma senza una sessione attiva si ha la possibilità di effettuare il sign-in.



The screenshot shows a login interface. At the top, the word "Login" is centered. Below it are two input fields: "Email" and "Password". Under the "Email" field is a checkbox labeled "Remember me on this computer". Below the "Password" field is a green "Log in" button. To the right of the button are links: "(forgot password?)", "Connect With Facebook", and "Connect With Google". At the bottom, there is a link "New user? Sign up now!"

Figura 2.2. Schermata sign in

Le user-stories relative a questa porzione sono:

3. As a USER i want to LOG IN WITH MY EMAIL so that i can ACCESS THE WEBSITE
4. As a USER i want to LOG IN WITH MY FACEBOOK OR GMAIL ACCOUNT so that i can

E' possibile, in caso di smarrimento, chiedere il reset della password. Verrà inviata una mail contenente un link che rimanda alla apposita sezione.



The screenshot shows a "Forgot password" form. At the top left is the "inline" logo. At the top right are "Sign up" and "Log in" buttons. Below the logo is the text "Forgot password". Underneath is an "Email" input field containing "mario.rossi@gmail.com". At the bottom left is a blue "Submit" button.

Figura 2.3. Schermata reset password smarrita

La user-story relativa a questa sezione sono:

5. As a USER i want to GENERATE A LINK TO MODIFY MY PASSWORD so that i can RECOVER MY PASSWORD

2.2.2 Profilo

E' presente una sezione profilo personale, che contiene tutti i dati immessi nella fase di registrazione. Nella fase di [sign-up](#) non è richiesto immettere dati reali eccetto per l'email, salvaguardando quindi la privacy dell'utente.

Si è scelto di rendere l'id utente un hash per una maggiore sicurezza: non è così possibile scorrere gli utenti del sito in maniera cronologica ma soprattutto, non

è possibile individuare gli admin, caso banale in ci fossero stati gli id numerici.

Con il referral link è possibile invitare conoscenti ad iscriversi, guadagnando così punti nella leaderboard. All'invitato arriverà una mail con il referral link dell'utente invitante.

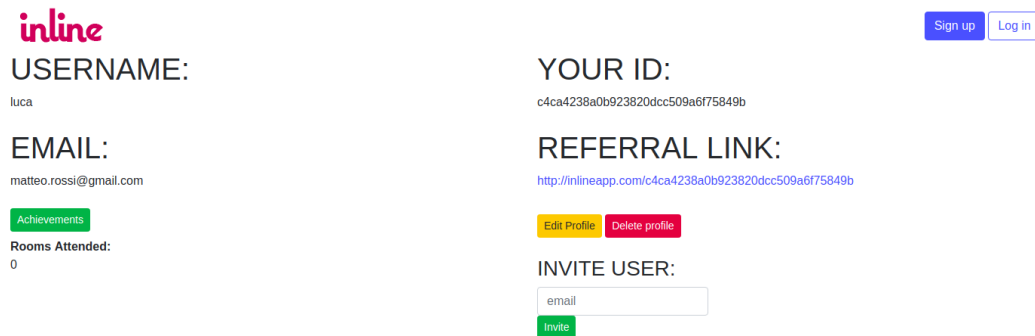


Figura 2.4. Schermata profilo

La user-story relativa a questa sezione sono:

6. As a USER i want to SHARE MY REFERRAL LINK so that i can INVITE MY FRIENDS AND INCREASE MY RANK

E' possibile modificare i propri dati all'interno del sito, in modo da mantenere aggiornati con la realtà. Si può cambiare e-mail, password e per i casi più estremi si ha la possibilità di eliminare ogni traccia del proprio account sul sito.

La user-story relativa a questa sezione sono:

7. As a USER i want to HAVE SETTINGS so that i can UPDATE MY PROFILE

8. As a USER i want to HAVE SETTINGS so that i can CHANGE MY PASSWORD

9. As a USER i want to HAVE SETTINGS so that i can DELETE MY PROFILE

Porzione collegata al referral link è la leadeboard, che tiene traccia delle stanze a cui hai partecipato o che hanno riscosso tanto successo, fornendo esperienza. I modi per ottenere punti sono i seguenti:

- Invitare amici tramite l'apposito input nel profilo.
- Partecipare ad una stanza, con successiva verifica.
- Una stanza creata da te ha riscosso un notevole successo.

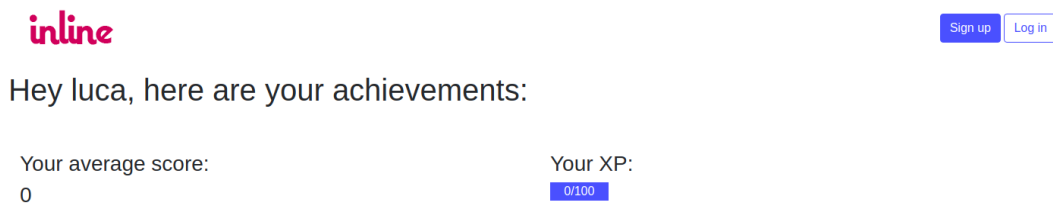


Figura 2.5. Schermata achievements

La user-story relativa a questa sezione sono:

- 12. As a USER i want to HAVE ACHIEVEMENTS so that i can SEE MY RATING
- 13. As a USER i want to HAVE ACHIEVEMENTS so that i can SEE HOW MANY ROOMS I ATTENDED
- 14. As a USER i want to HAVE ACHIEVEMENTS so that i can SEE MY RANK

2.2.3 Room

Creazione

Le stanze sono il cuore pulsante del progetto, possono essere create da tutti coloro che sono registrati, quindi anche admin. Una room può essere creata tramite il bottone nella toolbar, inserendo nome, descrizione, ora e data. Le rooms che riscuotono successo genereranno esperienza per i room host collegati.

Le room private, rese tali dalla decisione del proprietario, non compaiono nella homepage, barra di ricerca oppure mappa. L'unico modo di accere è tramite invito o link. Dopo la sottoscrizione ad una stanza nascosta l'utente è in grado di vederla normalmente.

E' possibile ricercare una località a cui la room fa riferimento tramite l'api di **mapbox** compilando l'apposito input, in caso di località non presente è consigliato aggiungerla nella descrizione. La ricorrenza delle stanze è un'idea che è stata resa possibile tramite un gem *recurring-select*, che permette di iterare un evento ogni porzione di tempo che si vuole. Per comodità sono state inserite nel sito le più usate: *Every day, every week, every month, every year*.

I radio buttons in alto segnalano l'opzione di scegliere il tipo di coda, argomentata nella prossima sottosezione, che può essere una fifo o una priority queue.

The screenshot shows the 'Crea la stanza' (Create the room) form in the 'inline' application. The form is titled 'Crea la stanza' and has a header with the 'inline' logo and navigation links: 'Create a room' (green button) and 'Logout' (red button). The form fields are as follows:

- Nome Stanza: A text input field.
- Tipo di coda: Two radio buttons: 'First in first out' (selected) and 'Choose your position'.
- Max partecipanti: A text input field.
- Descrizione: A text input field.
- Location: A search bar with a magnifying glass icon and the text 'Search'.
- Visibilità stanza: Two radio buttons: 'Pubblica' (selected) and 'Privata'.
- Definizione temporale: Two text input fields: 'Data inizio...' and 'Data fine...'.
- Ricorrenza evento: A dropdown menu with the option '- not recurring -'.

At the bottom of the form are two buttons: 'Submit' (blue) and 'Cancel' (grey).

Figura 2.6. Creazione stanza

La user-story relativa a questa sezione sono:

18. As a ROOM HOST i want to USE MAPBOX so that i can PIN AN ADDRESS ON A MAP
19. As a USER i want to USE GOOGLE CALENDAR so that i can SEE INCOMING EVENTS
20. As a ROOM HOST i want to USE GOOGLE CALENDAR so that i can TRACK AND SET RECURRENT EVENTS

Gestione coda

Essendo una parte core, abbiamo deciso di impostare la coda secondo due metodologie differenti, basandoci su due strutture dati ben note nel mondo informatico. **FIFO**, first-in-first-out, e la **Priority queue**, erano le due migliori alternative candidate a risolvere questo problema.

Nella prima soluzione, la più semplice poi da implementare, è soltanto necessario scegliere di partecipare alla stanza, il sistema farà il restante. E' possibile, tramite la visione della coda, capire approssimativamente il momento del proprio turno. Nel caso in cui l'antecedente cancelli la propria prenotazione, viene rimosso dalla queue facendo avanzare i successori.

La seconda scelta, invece, si differenzia dalla prima per poter scegliere il proprio slot in modo da poter assecondare le proprie necessità. Questo tipo di coda è più consigliata ad incontri con medici, professori o persino alla posta.

Entrambe le opzioni prevedono la possibilità di scambio, avanzata, tramite messaggio privato cliccando l'icona accanto al proprio nickname nella coda. L'utente verrà reindirizzato nella pagina dei messaggi privati, come oggetto

verrà utilizzato il seguente template [**utenteRichiedente NomeStanza in DateTime swapRequest from PosNum to PosNum**].

Il destinatario riceverà una proposta che potrà rifiutare o accettare tramite pulsanti presente nella comunicazione.

La user-story relativa a questa sezione sono:

- 26. As a USER i want to JOIN A ROOM so that i can RESERVE A SPOT
- 27. As a USER i want to UNJOIN A ROOM so that i can CANCEL MY RESERVATION
- 28. As a USER i want to REQUEST ANOTHER USER'S SPOT IN THE QUEUE so that i can SWITCH MY POSITION
- 29. As a USER i want to REPORT A ROOM OR USER so that i CAN ALERT AN ADMIN

User side

L'utente, una volta effettuata la sottoscrizione alla stanza ha a disposizione la chat associata room che consente di restare in contatto con gli organizzatori, in caso di imprevisti, o di comunicare con gli altri partecipanti. L'utente non può in nessun effettuare azioni di *delete* o *update* dei messaggi. Questo compito spetta rigorosamente ai moderatori e ai room-hosts.

In caso di scorrettezze è possibile segnalare qualsiasi cosa ad un admin del sito, che accoglierà la segnalazione e verificherà di conseguenza.

Alla sinistra della schermata si può notare la mappa con relativo pin sulla posizione esatta dell'evento, nel caso in cui la mappa non fosse presente è perché il room host ha preferito non dare un indirizzo fisico alla stanza. Due input ineditabili alla sinistra mostrano la data e ora dell'evento.

E' possibile utilizzare **Google Calendar** per tenere traccia delle proprie sottoscrizioni, infatti troviamo un link che aggiunge un evento al calendario personale. Il codice verrà esaminato nei capitoli successivi, per ora ci limitiamo a listare le varie features.

Quando un evento è imminente, si riceverà un email che ricorderà data, ora e luogo se presente della stanza a cui ci si è sottoscritti.

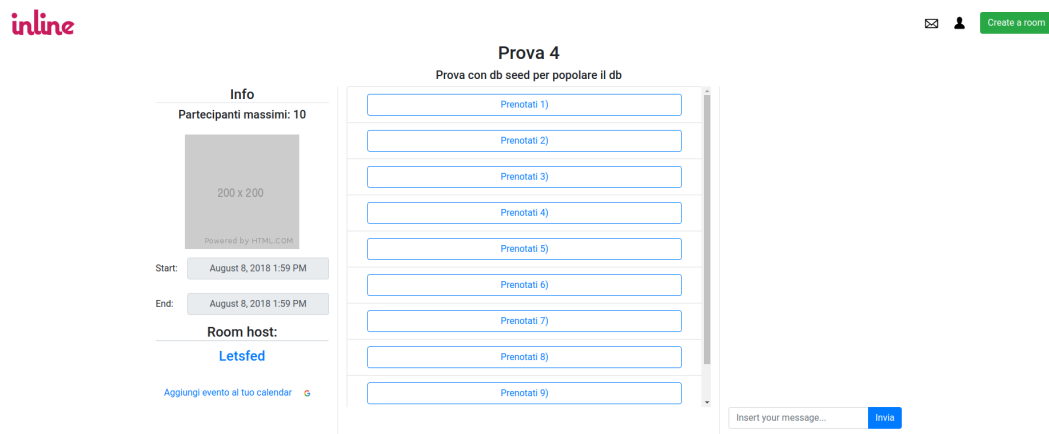


Figura 2.7. Room con priority queue vista da un utente

2.2.4 Room host side

Appena creata la stanza un messaggio flash, ci avviserà della corretta fondazione. Un room-host avrà poteri amministrativi relativi solo alle room che creato o alla quale è stato nominato tale da altri room-hosts.

I messaggi della chat presentano una croce rossa e una matita in alto a destra costruendo, in questo modo, un sistema di moderazione a cui partecipano solo gli utenti con poteri.

Nonostante si hanno tutti i poteri necessari per far funzionare al meglio la stanza, un room-host non potrà *mai e poi mai*: cambiare, cancellare o creare prenotazioni.

Nella parte alta della pagina è presente un input che offre la funzionalità di invitare utenti nella stanza, questo funziona sia se la room è pubblica o privata. Appena sotto è presente un link che cliccando, ci reindirizzerà nella pagina di edit, consentendoci di modificare i principali campi.


In caso di presenza, che essa sia fisica o virtuale, l'host della room provvederà a votare l'utente, con un pollice al fianco della prenotazione della persona interessata.

Dottor schemilazzi
Dottor schemilazzi riceve tutti i giorni dalle 15 alle 18 tranne la domenica. Portare con se cartella clinica

[Edit room](#) [Invita](#)


Info

Partecipanti massimi: 15



200 x 200

Powered by HTML5.COM



Dottor schemilazzi

via Del Babuino, 00187 Roma Roma, Italy

Start: October 2, 2019 6:45 AM

End: October 2, 2019 6:00 PM

Room host:

[Antonio](#)

[Aggiungi evento al tuo calendar](#)

Insert your message... [Invia](#)


Figura 2.8. Room vista da un room host

inline

[Go back to the room](#) [Create a room](#) [Logout](#)

Modifica la room

[Ripristina immagine predefinita](#)



200 x 200

Powered by HTML5.COM

[Scogli file](#) | Nessun file selezionato

Immagine caricata verrà usata solo come thumbnail, dimensione consigliata 200x200

[Upload](#)

Room host username [Add](#)

[Cancella la room](#)

[Antonio](#)

Personal info

Nome Stanza:

Descrizione:

Location:

Max unjoin time:

Visibilità stanza: ☒ Pubblica ☐ Privata

[Save changes](#) [Cancel](#)

Figura 2.9. vista dell'edit room

Le user stories relative a queste due sottosezioni sono:

21. As a USER i want to HAVE A CHAT IN THE ROOM so that i can COMMUNICATE WITH OTHER ROOM PARTICIPANTS
22. As a ROOM HOST i want to HAVE SPECIAL PRIVILEGES so that i can EDIT MESSAGES
23. As a ROOM HOST i want to HAVE SPECIAL PRIVILEGES so that i can PIN A MESSAGE
25. As a USER i want to BE NOTIFIED OF MY INCOMING EVENT so that i can HAVE A REMINDER SENT VIA EMAIL
30. As a ROOM HOST i want to HAVE SPECIAL SETTINGS so that i can EDIT A ROOM'S PARAMETERS

31. As a ROOM HOST i want PROMOTE ANOTHER USER TO ROOM HOST to so that HE CAN HAVE HOST PRIVILEGES
32. As ROOM HOST i want to CHECK THE EVENT ATTENDEES so that i can RATE A USER
33. As ROOM HOST i want to INVITE USERS TO MY ROOM so that i can ALERT THEM OF THE EVENT

2.3 Dashboard

Una persona, che sia registrata o un visitatore, riesce a vedere la dashboard, che comprende le principali funzioni di ricerca e listing delle room.

Si presenta con una mappa in alto, contenente, tutte le stanze vicino alla propria posizione o, occasionalmente, si può ricercare, tramite la barra di ricerca, le room per nome e per raggio o per località. Le stanze che vengono ricercate compariranno sulla mappa, di conseguenza, scompariranno quelle che non corrispondono alla query.

Vicino alla barra è presente un pulsante che consente di centrare la propria posizione sulla mappa, fornendo le opportune autorizzazioni al browser.

Le room sono listate al di sotto della mappa in un riquadro per ciascuna che contiene descrizione, luogo, un link per visitare la stanza e, al passaggio del mouse, rivelerà i room hosts.

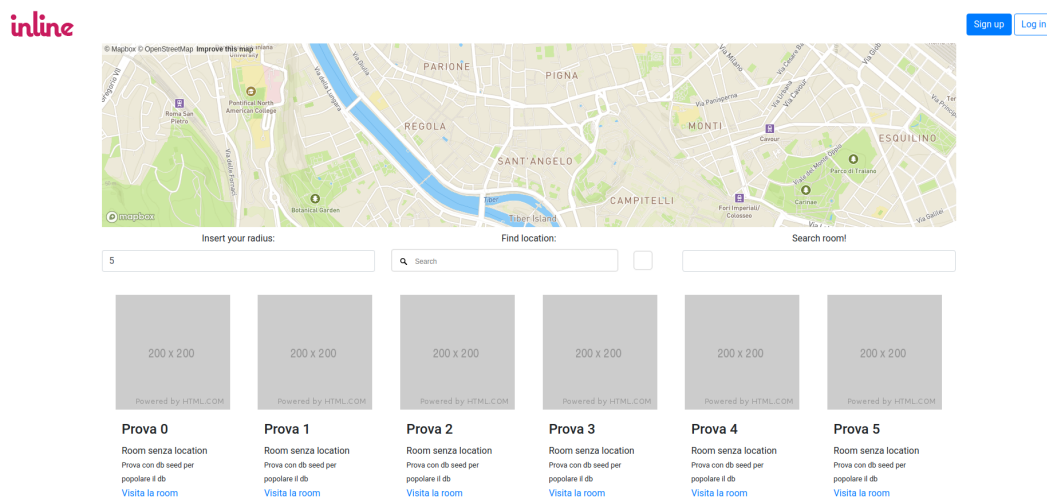


Figura 2.10. Vista della dashboard

Le user stories relative a questa sezione sono:

15. As a USER i want to USE MAPBOX so that i can SEE WHERE THE EVENT IS LOCATED ON A MAP
16. As a USER i want to USE MAPBOX so that i can SEE ROOMS IN A DETERMINED RADIUS NEAR ME
17. As a USER i want to USE MAPBOX so that i can SEE ROOMS IN A

CERTAIN LOCATION

24. As a USER i want to USE THE SEARCH BAR so that i can FIND ROOMS

34. As an UNREGISTERED USER / USER i want to USE THE DASHBOARD so that i can SEE PUBLICLY LISTED ROOMS

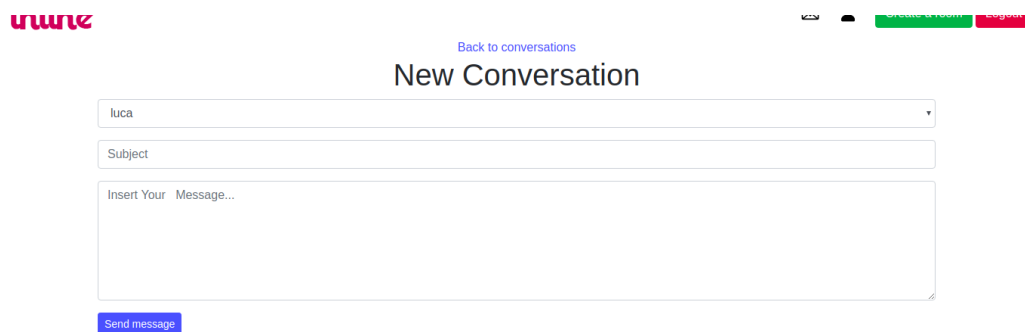
35. As a USER i want to SEE THE ROOMS I'M CURRENTLY ATTENDING so that i can KEEP TRACK OF THEM

2.4 Messaggi privati

E' stato concesso agli utenti la possibilità di interagire tra di loro al di fuori delle room, tutto grazie ai messaggi privati. In una qualsiasi pagina del sito è possibile cliccare sull'icona della posta e creare un DM per un qualsiasi utente iscritto.

La schermata presenta tre input diversi:

- Destinatario, un qualsiasi utente iscritto al sito, a patto che abbia l'opzione di ricevere messaggi privati attivata.
- Oggetto, un titolo per l'argomento del messaggio.
- Testo, il corpo del messaggio o l'invito a scambiare un posto alla coda.



The screenshot shows the 'New Conversation' interface. At the top left is the 'unine' logo. At the top right are icons for a bell, a person, and a battery level indicator. Below these is a link 'Back to conversations'. The main title is 'New Conversation'. There are three input fields: a dropdown menu with 'luca' selected, a 'Subject' field, and a large text area with the placeholder 'Insert Your Message...'. At the bottom left is a blue 'Send message' button.

Figura 2.11. Vista dei messaggi privati

2.5 Admin

Ogni sito che si rispetti ha una persona a comando che gestisce e controlla gli iscritti o semplicemente modifica i contenuti. Nel nostro caso un admin ha la possibilità di cancellare ogni traccia di un utente oppure bannarlo, a seconda dall'azione scorretta compiuta.

E' presente una dashboard accessibile solo da admin o utenti eletti da essi. Si ha accesso a tutte le funzioni sopracitate e in più si può sospendere il sito, per manutenzione o motivi tecnici.

L'admin ha il potere di cambiare la maggior parte dei parametri all'interno della piattaforma come, ad esempio, quelli delle room, se violano le politiche del sito oppure campi utenti per gli stessi motivi.

L'amministratore riesce a vedere tutte le stanze presenti, in modo da favorire il proprio compito di moderazione.

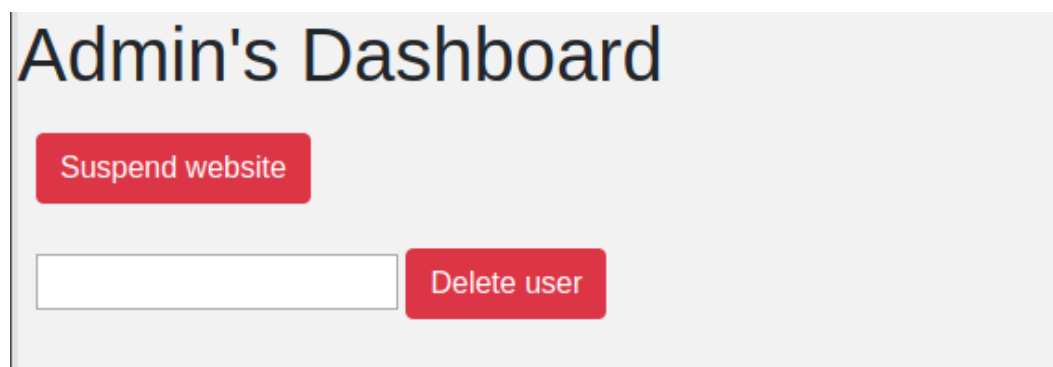


Figura 2.12. Vista della dashboard admin

Le user stories relative a questa sezione sono:

- 36. As an ADMIN i want to HAVE SPECIAL SETTINGS so that i can DELETE A USER
- 37. As an ADMIN i want to HAVE SPECIAL SETTINGS so that i can HANDLE A USER'S STATUS
- 38. As an ADMIN i want to HAVE SPECIAL PRIVILEGES so that i can SEE ALL ROOMS
- 39. As an ADMIN i want to HAVE SPECIAL SETTINGS so that i can SUSPEND THE WEBSITE
- 40. As an ADMIN i want to HAVE SPECIAL PRIVILEGES so that i can MODIFY ANY ROOM

2.6 User stories

Per la progettazione del sito abbiamo usato le user stories che, un linguaggio informale per descrivere le features della piattaforma dal punto di vista dell'end user, il tutto può essere paragonato a dei post-it. Per chiarezza esporremo le user storie raggruppate nelle sezioni appena affrontate.

2.6.1 Sezione utente e profilo

1. As an UNREGISTERED USER i want to SIGN UP WITH MY EMAIL so that i can BECOME A USER
2. As an UNREGISTERED USER i want to SIGN UP WITH MY FACEBOOK OR GMAIL ACCOUNT so that i can BECOME A USER
3. As a USER i want to LOG IN WITH MY EMAIL so that i can ACCESS THE WEBSITE
4. As a USER i want to LOG IN WITH MY FACEBOOK OR GMAIL ACCOUNT so that i can

2.6.2 Sezione accesso al sito web

5. As a USER i want to GENERATE A LINK TO MODIFY MY PASSWORD so that i can RECOVER MY PASSWORD
6. As a USER i want to SHARE MY REFERRAL LINK so that i can INVITE MY FRIENDS AND INCREASE MY RANK
7. As a USER i want to HAVE SETTINGS so that i can UPDATE MY PROFILE
8. As a USER i want to HAVE SETTINGS so that i can CHANGE MY PASSWORD
9. As a USER i want to HAVE SETTINGS so that i can DELETE MY PROFILE

2.6.3 Sezione messaggi privati

10. As a USER i want to HAVE PRIVATE MESSAGES so that i can MESSAGE ANOTHER USER
11. As a USER i want to HAVE PRIVATE MESSAGES so that i can RECEIVE MESSAGES FROM OTHER USERS

2.6.4 Sezione achievements

12. As a USER i want to HAVE ACHIEVEMENTS so that i can SEE MY RATING
13. As a USER i want to HAVE ACHIEVEMENTS so that i can SEE HOW MANY ROOMS I ATTENDED
14. As a USER i want to HAVE ACHIEVEMENTS so that i can SEE MY RANK

2.6.5 Sezione servizi esterni

15. As a USER i want to USE MAPBOX so that i can SEE WHERE THE EVENT IS LOCATED ON A MAP

16. As a USER i want to USE MAPBOX so that i can SEE ROOMS IN A DETERMINED RADIUS NEAR ME
17. As a USER i want to USE MAPBOX so that i can SEE ROOMS IN A CERTAIN LOCATION
18. As a ROOM HOST i want to USE MAPBOX so that i can PIN AN ADDRESS ON A MAP
19. As a USER i want to USE GOOGLE CALENDAR so that i can SEE INCOMING EVENTS
20. As a ROOM HOST i want to USE GOOGLE CALENDAR so that i can TRACK AND SET RECURRENT EVENTS

2.6.6 Sezione chat

21. As a USER i want to HAVE A CHAT IN THE ROOM so that i can COMMUNICATE WITH OTHER ROOM PARTICIPANTS
22. As a ROOM HOST i want to HAVE SPECIAL PRIVILEGES so that i can EDIT MESSAGES
23. As a ROOM HOST i want to HAVE SPECIAL PRIVILEGES so that i can PIN A MESSAGE

2.6.7 Sezione ricerca

24. As a USER i want to USE THE SEARCH BAR so that i can FIND ROOMS

2.6.8 Sezione notifica

25. As a USER i want to BE NOTIFIED OF MY INCOMING EVENT so that i can HAVE A REMINDER SENT VIA EMAIL

2.6.9 Sezione room

26. As a USER i want to JOIN A ROOM so that i can RESERVE A SPOT
27. As a USER i want to UNJOIN A ROOM so that i can CANCEL MY RESERVATION
28. As a USER i want to REQUEST ANOTHER USER'S SPOT IN THE QUEUE so that i can SWITCH MY POSITION
29. As a USER i want to REPORT A ROOM OR USER so that i CAN ALERT AN ADMIN
30. As a ROOM HOST i want to HAVE SPECIAL SETTINGS so that i can EDIT A ROOM'S PARAMETERS
31. As a ROOM HOST i want PROMOTE ANOTHER USER TO ROOM HOST to so that HE CAN HAVE HOST PRIVILEGES

- 32. As ROOM HOST i want to CHECK THE EVENT ATTENDEES so that i can RATE A USER
- 33. As ROOM HOST i want to INVITE USERS TO MY ROOM so that i can ALERT THEM OF THE EVENT

2.6.10 Sezione dashboard

- 34. As an UNREGISTERED USER / USER I want to USE THE DASHBOARD so that i can SEE PUBLICLY LISTED ROOMS
- 35. As a USER i want to SEE THE ROOMS I'M CURRENTLY ATTENDING so that i can KEEP TRACK OF THEM

2.6.11 Sezione admin

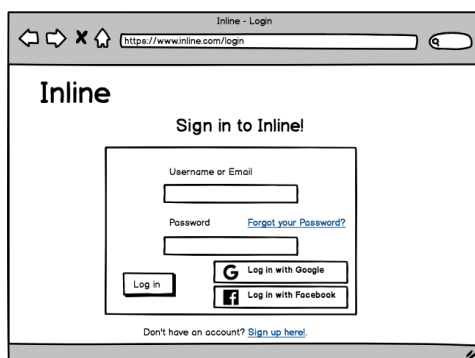
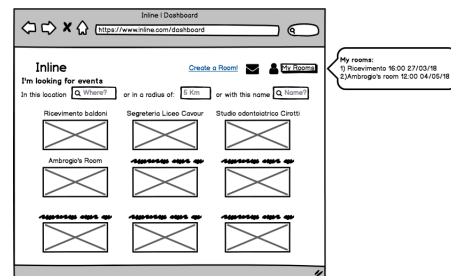
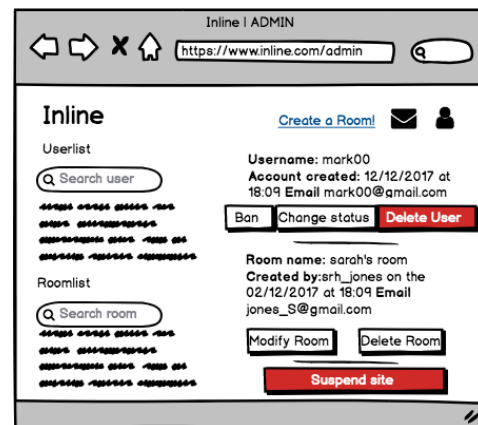
- 36. As an ADMIN i want to HAVE SPECIAL SETTINGS so that i can DELETE A USER
- 37. As an ADMIN i want to HAVE SPECIAL SETTINGS so that i can HANDLE A USER'S STATUS
- 38. As an ADMIN i want to HAVE SPECIAL PRIVILEGES so that i can SEE ALL ROOMS
- 39. As an ADMIN i want to HAVE SPECIAL SETTINGS so that i can SUSPEND THE WEBSITE
- 40. As an ADMIN i want to HAVE SPECIAL PRIVILEGES so that i can MODIFY ANY ROOM

2.7 Mockup

Un mockup è una riproduzione stilizzata, solitamente effettuata in fase di progettazione , della struttura che assumeranno le pagine dell'applicazione.

In questo progetto sono state prima stilate le user stories e successivamente i mockup, attraverso un software online gratuito chiamato **Balsamiq**. Con facilità, questo programma permette di inserire moltissimi elementi arricchendo la bozza di utilissimi dettagli.

Di seguito verranno listati i mockup utilizzati in questo progetto.



Inline - Change your password
https://www.inline.com/one_time_link

Inline



Insert your new password

Password

Confirm Password

Inline | Messages
https://www.inline.com/message

Inline

[Create a Room!](#)  

From **luca#d78f9k** Hey man, do you need a lift to University?

From **Jake1994#d78b8k** What you up to after the exam? Wanna grab a drink?



From **Stephanie_12#d7999k** I'll be there at 12 o'clock! See you then!

Send to:

Message

Inline | Profile
https://www.inline.com/profile

Inline

[Create a Room!](#)  

Username Your ID: ok89iP0Z

Email Referral link: inline.com/invite/ok89iP0Z

Reset Password Confirm Password

Inline - Track event
https://www.inline.com/track

Inline

Your Calendar

18th May 2018

MAY 2018						
S	M	T	W	T	F	S
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

11:00 Ricevimento Baldoni
12:00
13:00
14:00
15:00
16:00 Ambrogio's room
17:00
18:00
19:00

Inline - Sign Up
https://www.inline.com/signup

Inline

Join Inline!

Username

Email

Password

Inline - Sign Up
https://www.inline.com/signup

Inline

Join Inline using your Google or Facebook account!

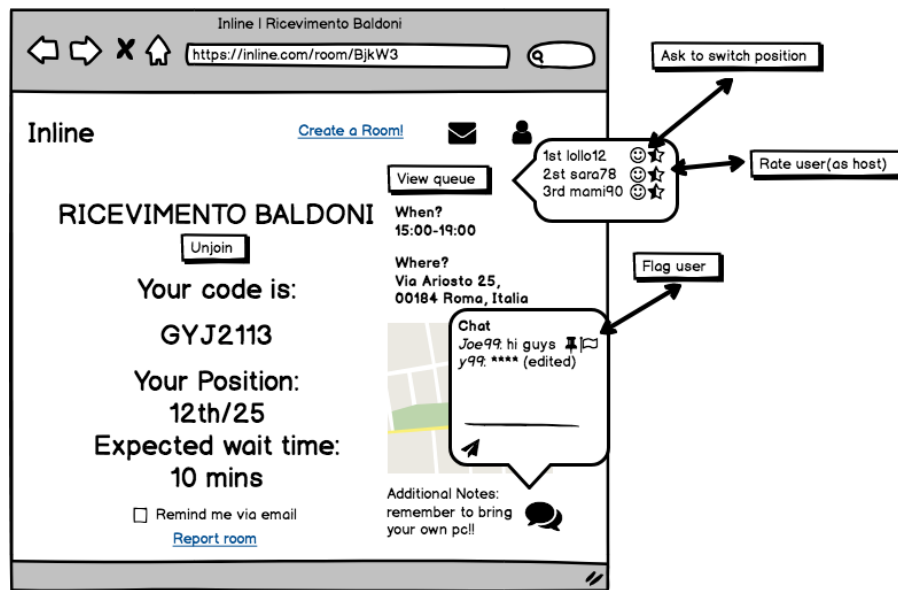
Username

Email

Password

We will need the following info from your Facebook/Gmail account:

By clicking "Create account" you accept that we may use your Facebook/Gmail account's info.



Capitolo 3

Descrizione tecnologie usate

Nel seguente capitolo illustrerò le varie metodologie e tecnologie utilizzate, con attenzione al framework e le api.

3.1 Ruby e Ruby on Rails

Ruby è un linguaggio di programmazione nato nel 1993 che, pur essendo ad oggetti, presenta alcune caratteristiche tipiche dei paradigmi imperativo e funzionale.

Il paradigma ad oggetti di **Ruby** è puro, ossia ogni componente del linguaggio, dalle costanti numeriche alle classi, è un oggetto, e come tale può possedere metodi; a differenza dei linguaggi come C++ e derivati, tuttavia, gli oggetti in Ruby sono qualcosa di molto più dinamico, in quanto è possibile aggiungere o modificare metodi a run-time. Il tipo di un oggetto, perciò, non è definito tanto dalla classe che lo ha istanziato, quanto dall'insieme dei metodi che possiede, o dei messaggi a cui sa rispondere.

In Ruby, dunque, è fondamentale il *duck typing* (dall'inglese *if it looks like a duck, and quacks like a duck, it must be a duck*), ovvero il principio secondo il quale il comportamento di una funzione sui suoi argomenti non deve essere determinato dal tipo di questi (come accade in C++ e altri linguaggi staticamente tipizzati), bensì da quali messaggi essi sono in grado di gestire.

Un'altra caratteristica fondamentale di Ruby è costituita dai cosiddetti blocchi, che sono sostanzialmente delle chiusure (ovvero funzioni dotate di ambiente), e che consentono di sostituire i cicli espliciti, frequenti nei linguaggi a basso livello, con l'utilizzo di iteratori, nascondendo così al chiamante i meccanismi interni del ciclo in questione.

Ruby è il linguaggio del framework open source **Ruby on rails**, per applicazioni web. La sua architettura è una **MVC** (Model - view - controller). I suoi obiettivi sono la semplicità e la possibilità di Ruby on rails è distribuito

attraverso **RubyGems**, l'unico canale ufficiale di distribuzione per librerie ed applicazioni Ruby.

I principi guida di Ruby on Rails comprendono *don't repeat yourself* e *convention over configuration*.

Don't repeat yourself significa che le definizioni devono essere poste una volta soltanto. Poiché Ruby On Rails è un framework *full-stack*, i componenti sono integrati in modo tale che i collegamenti fra di essi non devono essere impostati manualmente. Ad esempio in Active Record, le definizioni delle classi non devono specificare i nomi delle colonne; Ruby può estrarli direttamente dal database, dunque riportarli anche nel programma sarebbe ridondante.

Convention over configuration significa che il programmatore ha bisogno di metter mano alla configurazione soltanto per ciò che differisce dalle convenzioni. Ad esempio, se un modello è costituito dalla classe Post, la corrispondente tabella nel database deve chiamarsi posts, o altrimenti deve essere specificata manualmente.

3.2 Google calendar Api

E' stato deciso a monte che il progetto fosse di tipo gestionale, come potevamo non includere il calendario di google?

Google calendar è un sistema calendari concepito da Google ed offre la possibilità creare più calendari, di dividerli e importarli da altri servizi, il che, calza a pennello con l'obiettivo di *Inline*, creare eventi e ottimizzare la gestione del tempo.

L'implementazione è fatta utilizzando la gemma *googleauth*, si avrebbe potuto scegliere anche una soluzione di basso livello scrivendo le varie chiamate **HTTP** from scratch, ma si è preferito utilizzare una libreria che con poche chiamate fa il suo lavoro.

L'applicazione dispone di un account di servizio in modo che ogni presente sul sito lo sia anche sul calendario. Diverso invece è per gli utenti: l'evento non viene aggiunto automaticamente al proprio calendario, ma è necessario cliccare sul link ipertestuale inserito nella room. Per semplicità si è evitato di incorporare evento ricorsivi nel calendario degli utenti.

3.3 Google e facebook oauth

Facebook e *Google* mettono a disposizione le proprie API per poter ottenere le autorizzazioni per effettuare sign-in o sign-up. E' necessario essere iscritto ad entrambi i siti e ottenere un account sviluppatore.

Dal 2018 Facebook ha introdotto il vincolo per l'utilizzo dell'api. Ogni applicazione che utilizzi tali servizi necessita di un certificato *SSL*. Per superare

questo problema in fase di development abbiamo dovuto utilizzare un nostro certificato. E' doveroso bootare il server con il comando:

```
rails s -b  
      'ssl://127.0.0.1:3000?key=localhost.key&cert=localhost.crt'
```

3.4 Mapbox

Mapbox è uno dei più grandi fornitori di mappe online, abbiamo preferito l'utilizzo di questa API rispetto a *Google maps* poichè già molto servizi erano dipendenti dal grande colosso americano. In caso di guasto ci sarebbero stati molti problemi. L'api è stata implementata solo a livello di front-end, tramite librerie *JQuery* e *Javascript*. Per ottenere la chiave per usare questo applicativo abbiamo bisogno di un account sviluppatore nel sito di mapbox.

Nel sito è presente la mappa nella dashboard, dove presenta le room vicine alla posizione ricercata o a quella attuale, ottenuta tramite GPS, nella creazione della room che però, a differenza della precedente, è nascosta all'utente finale. Questo perchè è necessario che sia presente per far funzionare il ricercatore di località Mapbox. Ed infine, è presente nelle stanze che hanno una posizione, nella colonna sinistra.

3.5 Mailboxer

Mailboxer è una gemma Rails che fa parte del framework *social stream* per la creazione di social network. È un sistema di messaggistica generico che permette di rendere messaggiabile qualsiasi modello, dotandolo di alcuni strumenti versatili. Con Mailboxer, puoi creare conversazioni con uno o più destinatari e notificare via e-mail. È anche possibile inviare messaggi tra diversi modelli e aggiungere allegati. In poche parole, è lo strumento che ci ha permesso di creare i messaggi privati all'interno della piattaforma.

3.6 Bootstrap

Bootstrap è una raccolta di libreria per la creazione di siti e applicazione per il web. E' stata sviluppata presso *Twitter* come un framework che uniformasse i vari componenti che ne realizzavano l'interfaccia web, dato che la presenza di diverse libreria aveva portato ad incoerenze ed elevati oneri di manutenzione.

Contiene modelli basati su *CSS* e *HTML* per l'interfaccia grafica, contenendo quindi, bottoni, moduli e pulsanti di navigazioni. E' stato incluso con la gemma *bootstrap* e poi chiamato tramite *SASS*.

3.7 Ice cube

Ice cube è una gemma di rails che gestire eventi ripetuti facilmente. L'API è modellata sugli eventi iCalendar in una piacevole sintassi di Ruby. Il potere sta nella capacità di specificare regole multiple e fare in modo che Ice cube capisca rapidamente l'evento schedulato cada in una determinata data o in quali orari si verifica.

E' stata presentata in un tutorial di *Go rails*, un famoso canale che porta notizie sul framework in questione, dove ne mostrava l'utilizzo.

3.8 Strumenti minori

Altri strumenti che non *meritano* una sezione a parte, poichè non hanno avuto un ruolo fondamentale nello sviluppo del sito sono:

- **Paperclip**, una gemma utilizzata per ritagliare gli avatar delle Room.
- **Bcrypt**, una gemma utilizzata per encryptare le password degli utenti in fase di creazione
- **Devise invitable**, fa parte del pacchetto devise e serve per invitare persone via mail
- **Friendly ID**, genera un hash e lo inserisce al posto dell'ID utente e delle rooms, per questioni di sicurezza.

3.9 Metodologie usate

Nello sviluppo dell'applicazione sono stati utilizzati diversi strumenti. Il principale è stato *Git-Hub*, un famoso sito per sviluppatori comprato di recente dalla *Microsoft*. Ci ha permesso di essere sempre aggiornato con il lavoro degli altri componenti del team.

Io, personalmente, ho utilizzato Visual studio code della *Microsoft*, come editor di testo. Sebbene sia uno strumento pesante, mi ha aiutato molto alla stesura di un codice pulito e facile da leggere.

E' stato utilizzato uno sito esterno, *Taiga*, il tracking delle user stories e per gli sprint finali. E' stato lo strumento che ci ha permesso di dividere il lavoro per quattro persone in modo equivalente, sia per difficoltà che lunghezza.

Capitolo 4

Analisi e progettazione concettuale

In questo capitolo verranno analizzati la struttura e lo sviluppo della base di dati su cui si appoggia l'applicazione, useremo un diagramma molto usato nella progettazione, chiamato **diagramma ER** (Entità-Relazione)

4.1 Analisi dei requisiti

Inizialmente, per la progettazione di una base di dati è necessario raccogliere i requisiti che il database deve soddisfare. Si intende così, l'individuazione dei problemi che l'applicazione deve risolvere oltre alle caratteristiche che dovrà assumere.

I requisiti vengono raccolti in specifiche espresse generalmente in linguaggio naturale e in modo confusionale, per questo l'analisi dei requisiti viene in aiuto chiarendo e organizzando le specifiche.

I requisiti raccolti durante la progettazione di Inline sono:

User

- ID, identificativo unico incrementante
- Username, Il nome scelto per l'utente
- Password, Una parola che rispetti i requisiti scelta dall'utente
- RoomAttended, Il numero delle stanze a cui questo utente ha partecipato
- Admin, Un campo booleano che denota se l'utente è amministratore
- Rating, Un campo che specifica il rating utente
- CreatedAt, Campo che specifica la data di creazione

- UpdatedAt, Campo che specifica l'ultima modifica del profilo
- Varie colonne JSON per la rappresentazione dei social network.
- Altre colonne accessorie come 'ResetDigest', 'Invitations', 'ResetSentAt'

Room

- ID, identificativo unico incrementante
- Nome, Il nome scelto per la room
- Fifo, Un boolean che descrive la tipologia delle code
- MaxParticipants, Un campo integer che contiene il massimo numero di partecipanti
- Latitude, Latitudine del luogo in cui si terrà la room
- Longitude, Longitudine del luogo in cui si terrà la room
- Private, Un boolean che descrive se la room è privata
- CreatedAt, Campo che specifica la data di creazione
- UpdatedAt, Campo che specifica l'ultima modifica della room
- Description, Campo che contiene una descrizione della room
- Address, Campo che contiene l'indirizzo in cui si terrà la room

Prenotazione

- IDUser, Identificativo dell'utente
- IDRoom, Identificativo della room
- CreatedAt, Campo che specifica la data di creazione della prenotazione
- UpdatedAt, Campo che specifica l'ultima modifica della prenotazione

Mailbox

- IDUser, Identificativo del mittente
- IDRoom, Identificativo della room
- Body, Il corpo del messaggio
- ReceiverType, Il tipo del destinatario
- ReceiverId, L'id del destinatario
- Una serie di campi per confermare la lettura, modifica e cancellazione del messaggio

4.2 Diagramma ER

Nell'ambito della progettazione dei database il modello Entità-Relazione è un modello per la rappresentazione concettuale e grafica dei dati a un alto livello di astrazione.

Il modello ER viene spesso utilizzato nella prima fase della progettazione di una base di dati in cui è necessario tradurre le informazioni risultanti dall'analisi di un determinato dominio in uno schema concettuale.

Nell'ambito della progettazione ingegneristica delle basi di dati si distinguono in tre livelli indipendenti e consecutivi di progettazione: *Progettazione concettuale*, *progettazione logica*, *progettazione fisica*.

I principali costrutti del diagramma sono:

- Le entità
- Le associazioni
- Gli attributi
- La cardinalità

Nella prossima pagina sarà riportato il diagramma entità-relazione di Inline, nel quale si descrive la creazione di una room da parte dell'utente. Ogni diagramma deve essere accompagnato da una documentazione che lo descrive in tutte le sue componenti.

Le successive tabelle mostrano ciò che viene chiamato il dizionario dei dati associato al diagramma ER, costituito dalle tabelle dell'entità, delle relazioni e degli attributi.

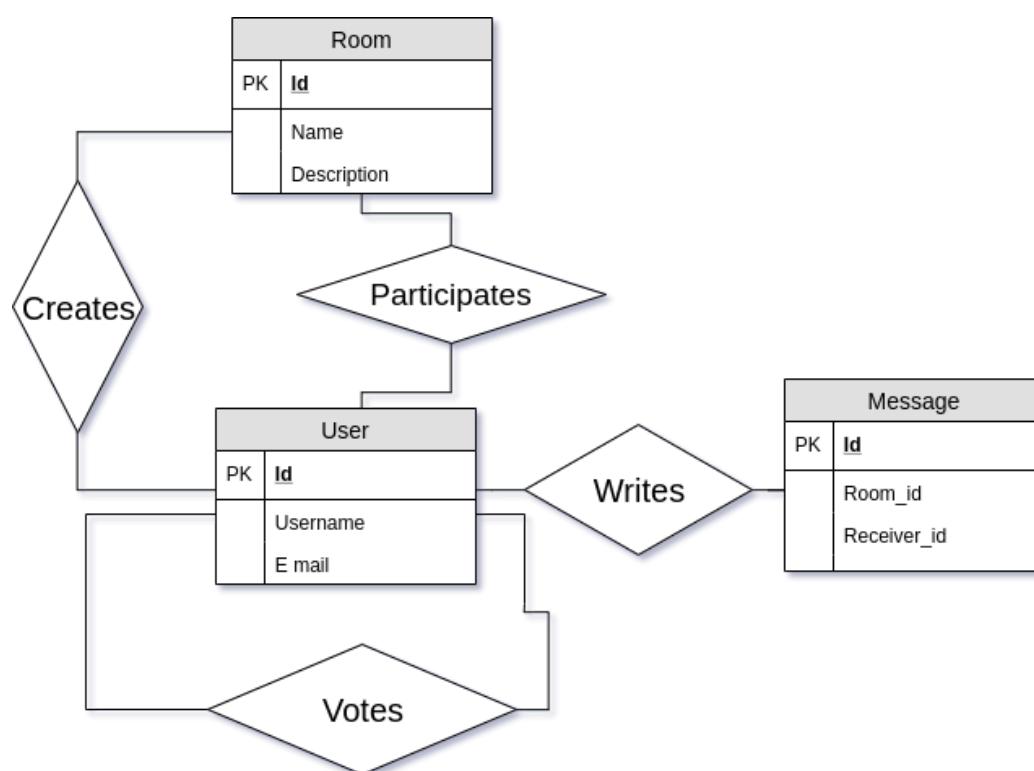


Figura 4.1. Diagramma ER

4.3 Dizionario dei dati: Entità

Nella prima colonna troviamo le entità, cioè gli elementi cardine dello schema, nonché tabelle principali nella base di dati dell'applicazione. Nella seconda colonna è presente una descrizione che aiuta a comprendere meglio l'elemento precedente. Nella terza colonna troviamo gli attributi che rappresentano le colonne nel database. Per ultimo, ma non per importanza, troviamo l'identificatore della riga.

Tabella 4.1. Tabella delle entità

Entità	Identificativo	Attributi
User	ID	ID , Username, Email, Password, Admin, Rating, Invitations, Rooms attended
Room	ID	ID , Nome, Descrizione, Fifo, MaxParticipants, Address, Latitude, Longitude, Private, MaxUnjoinTime, TimeFrom, TimeTo, Recurrence, EventId, UserId
Message	ID	ID , RoomId, ReceiverId, NotificationId, IsRead, isDelivered, Trashed, Deleted, MailboxType, deliveryMethod, MessageId

4.4 Dizionario dei dati: Relazioni

Le relazioni, rappresentate con dei rombi nel diagramma ER costituiscono i legami che ci sono tra *minimo* due entità. In base al numero di entità coinvolte possiamo definire il grado della relazione.

Tabella 4.2. Tabella delle relazioni

Relazione	Descrizione	Componenti
Creates	Creazione di una room da parte di un utente	User, Room
Participates	Partecipazione di un utente ad una room	User, Room
Writes	Scrittura e pubblicazione di un messaggio da parte di un utente	User, Message
Votes	Votazione di un utente verso un altro utente	User

4.5 Dizionario dei dati: Attributi

Attraverso tre tabelle nomineremo tutti gli attributi presenti nello schema ER menzionando il dominio, l'entità. La prima colonna citerà il nome dell'attributo, sarà in grassetto se è un identificatore, la seconda l'entità alla quale appartiene ed infine, la terza definirà il dominio.

Tabella 4.3. Tabella degli attributi user

Attributo	Entità	Dominio
ID	User	Integer
Username	User	String
Email	User	String
Password	User	String
Rating	User	Json
Invitations	User	Json
RoomsAttended	User	Integer
Admin	User	Boolean

Tabella 4.4. Tabella degli attributi room

Attributo	Entità	Dominio
ID	Room	Integer
Name	Room	String
Description	Room	String
Fifo	Room	Boolean
MaxParticipants	Room	Integer
Address	Room	String
Latitude	Room	Float
Longitude	Room	Float
Private	Room	Boolean
MaxUnjoinTime	Room	Datetime
TimeFrom	Room	Datetime
TimeTo	Room	Datetime
Recurrence	Room	Text
EventId	Room	String
UserId	Room	String

Tabella 4.5. Tabella degli attributi message

Attributo	Entità	Dominio
ID	Room	Integer
RoomId	Message	Integer
UserId	Message	Integer
ReceiverType	Message	String
ReceiverId	Message	Integer
NotificationId	Message	Integer
IsRead	Message	Boolean
IsDelivered	Message	Boolean
Trashed	Message	Boolean
MailboxType	Message	String
DeliveryMethod	Message	String
MessageId	Message	String

4.6 Dizionario dei dati: vincoli di cardinalità

Un vincolo di cardinalità associa ad un ruolo **U** corrispondente ad un entità **E** in una relazione **R**, ed impone un limite minimo e massimo di istanze della relazione a cui può partecipare ogni istanza dell'entità **E** nel ruolo **U**. In inline abbiamo tutti i tipi di vincoli del tipo 0..*

Capitolo 5

Progettazione del sistema

5.1 Introduzione all'UML

In questo capitolo verrà trattata la progettazione e la modellazione del sistema attraverso uno strumento molto diffuso nello sviluppo delle applicazioni: UML.

In ingegneria del software, UML è un linguaggio di modellazione e di specifica basato sul paradigma orientato agli oggetti. Il linguaggio nacque con l'intento di unificare approcci precedenti, raccogliendo le migliori prassi nel settore e definendo così uno standard industriale unificato.

5.2 Architettura di riferimento

Nell'ingegneria del software, il termine architettura multi-tier o architettura multi-strato indica un'architettura software in cui le varie funzionalità del software sono logicamente separate ovvero suddivise su più strati o livelli software differenti in comunicazione tra loro.

Ciascuno strato è in comunicazione diretta con quelli adiacenti ovvero richiede ed offre servizi allo strato adiacente in maniera concettualmente simile a quanto accade con le architetture di rete a strati.

Nello specifico, in questa applicazione è stato usato il pattern MVC, ovvero il *Model-View-Controller* che, in informatica, è un pattern architetturale molto diffuso nello sviluppo di sistemi software, in particolare nell'ambito della programmazione orientata agli oggetti, in grado di separare la logica di presentazione dei dati dalla logica di business.

Questo pattern si posiziona nel livello logico o di business e di presentazione in una architettura multi-tier.

Il componente centrale del MVC, il modello, cattura il comportamento dell'applicazione in termini di dominio del problema, indipendentemente dall'interfaccia utente. Il modello gestisce direttamente i dati, la logica e le regole dell'applicazione. Una vista può essere una qualsiasi rappresentazione in output di informazioni, come un grafico o un diagramma. Sono possibili viste multiple

delle stesse informazioni, come ad esempio un grafico a barre per la gestione e la vista tabellare per l'amministrazione. La terza parte, il controller, accetta l'input e lo converte in comandi per il modello e/o la vista.

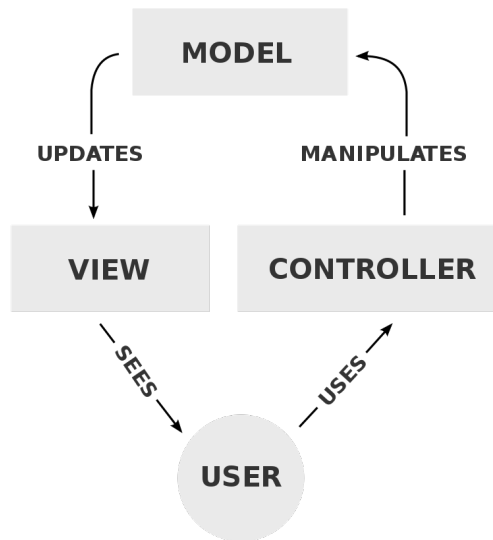


Figura 5.1. Architettura MVC schematizzata

5.3 Diagramma delle classi

Il diagramma delle classi consente di descrivere le entità del progetto, rappresentandoli con le loro caratteristiche e mettendole in relazione tra loro.

In questa sezione viene presentato il diagramma delle classi di Inline con i relativi requisiti e le specifiche necessarie alla gestione della classe Utente e delle varie classi ad essa connesse. Degli utenti interessa:

- Id
- Username
- Email
- Password
- Numero delle room a cui ha partecipato
- Il suo rating
- Data creazione
- Ultima modifica

In oltre, qualsiasi utente può in qualsiasi momento eliminare il proprio account, creare una room e inviare un messaggio.

Delle rooms interessa:

- Id
- Name
- Descrizione
- Tipo di coda
- Numero massimo partecipanti
- Indirizzo
- Longitudine
- Latitudine
- Se è privata
- Data creazione
- Data terminazione
- Id del creatore
- Id dell'evento

Dei messaggi interessa:

- Id
- Id della room
- Id dell'utente
- Tipo del destinatario
- Id del destinatario
- Notification id
- Campi per denotare se il messaggio è stato letto, cancellato
- Campo per la creazione del messaggio
- Campo per la modifica del messaggio

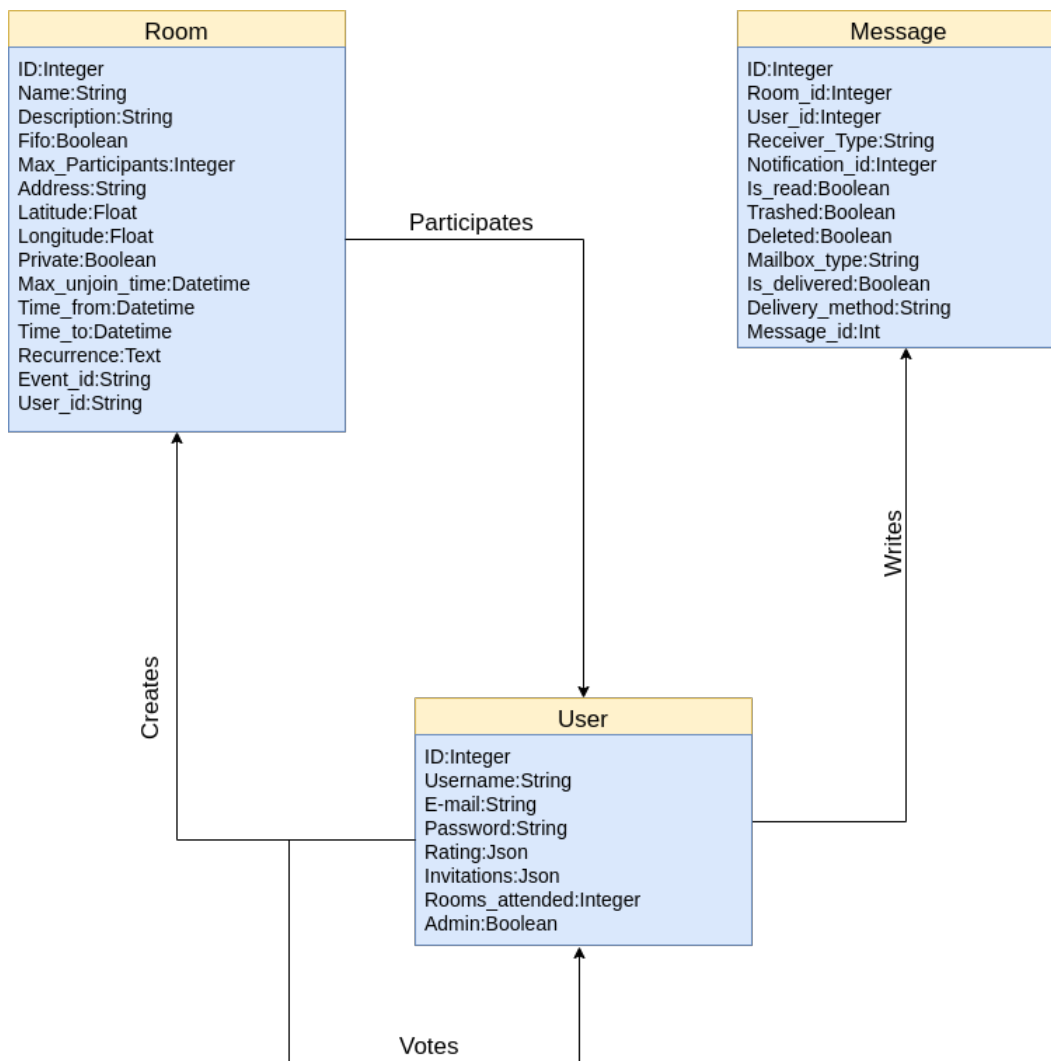


Figura 5.2. Diagramma UML delle classi

Adesso definiremo due operazioni: La delete per l'user e per il messaggio.

```

InizioSpecificaOperazioneClasse User
  Delete(Id:Integer): User
    Pre:Nessuno
    Post:Utente cancellato
Fine specifica
  
```

```

InizioSpecificaOperazioneClasse Message
  Delete(Id:Integer): Message
    Pre:Nessuno
    Post:Messaggio cancellato
Fine specifica
  
```

5.4 Diagramma degli stati

Il diagramma degli stati è un diagramma previsto dall'UML per descrivere il comportamento di classi, mostrando gli stati che sono assunti dall'entità in risposta ad eventi. Presentiamo il diagramma degli utenti di Inline: Un utente si trova inizialmente in uno stato *Visitatore*. Se un utente effettua il *Signup* diventa registrato. Se l'utente ha effettuato il *Signin*, può mandare un evento *Logout* ed uscire dal sito rimanendo un utente registrato. Se l'utente registrato non ha effettuato l'accesso, può mandare un evento *Signin* ed accedere al sito rimanendo utente registrato. Un utente può passare da uno stato di *Registrato* a *Ospite* in seguito ad un'operazione di delete.

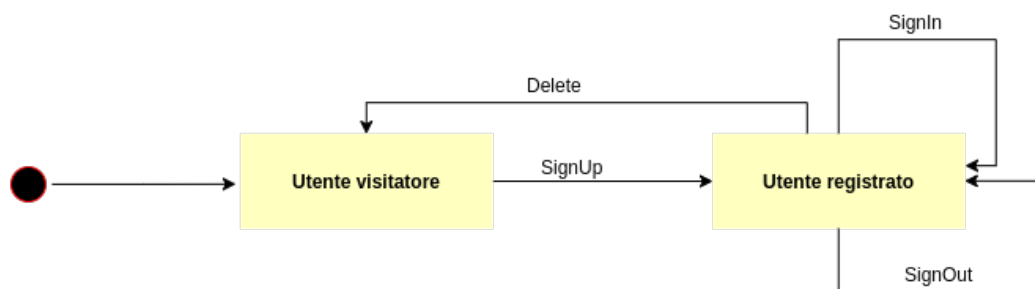


Figura 5.3. Diagramma degli stati

```
InizioSpecificaStatiClasse User
```

```
Stato: {Visitatore, Registrato}
```

```
Variabili di stato ausiliarie:
```

```
    UtenteRegistrato = User
```

```
    UtenteLog = User
```

```
Stato Iniziale:
```

```
    statoCorrente = Ospite
```

```
    utenteRegistrato = Nil
```

```
    utenteLog = Nil
```

```
Fine specifica
```

```
InizioSpecificaTransazioni User
```

```
Transazione Ospite -> Registrato (Signup)
```

```
    Evento: Signup
```

```
    Condizione: Nessuna
```

```
    Azione:
```

```
        Pre: Nil
```

```
        Post: UtenteRegistrato = This
```

```
Transazione Registrato -> Registrato (Signin)
```

```
    Evento: Sign in
```

```
    Condizione: UtenteLog = Nil
```

```
Azione:
  Pre: Nil
  Post: UtenteLog = This

Transazione Registrato -> Registrato (SignOut)
  Evento: SignOut
  Condizione: UtenteLog = This
  Azione:
    Pre: Nil
    Post: UtenteLog = Nil

Transazione Registrato -> Visitatore (Delete)
  Evento: Delete
  Condizione: Nessuna
  Azione:
    Pre: Nil
    Post: UtenteRegistrato = Nil
FineSpecifica
```

5.5 Diagramma delle attività

Il diagramma delle attività è un tipo di diagramma che permette di descrivere un processo attraverso dei grafi in cui i nodi rappresentano le attività e gli archi l'ordine con cui vengono eseguite. Nel nostro caso definiremo uno schema per le attività del *SignIn* effettuato sulla piattaforma, attraverso la quale, un utente può autenticarsi tramite *Email e password* o *OAuth* con servizi esterni.

Il cerchio chiuso rappresenta l'inizio dell'attività, mentre quello con i bordi la fine. I vari rettangoli le attività e le frecce il flusso con cui vengono svolte.

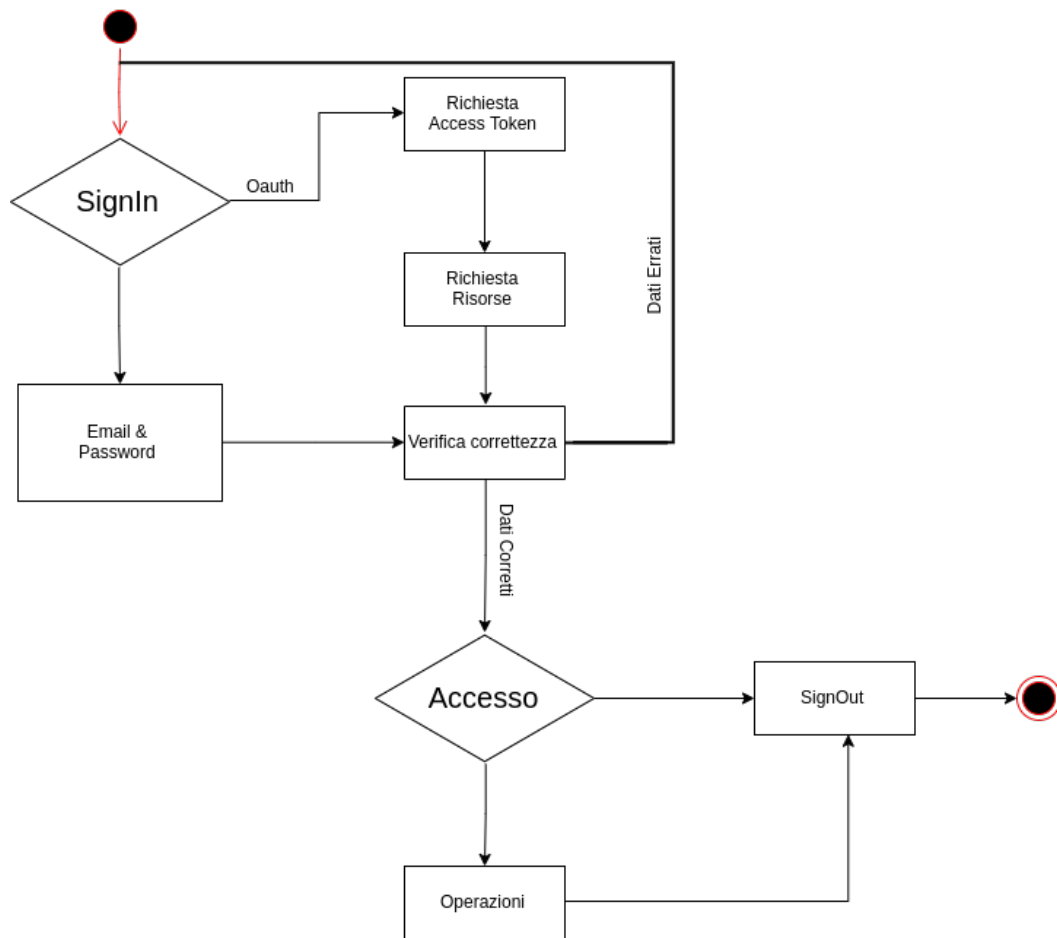


Figura 5.4. Diagramma attività dell'operazione SignIn

```

InizioSpecificaAttivita Accesso
  Accesso()
VariabiliProcesso
  Dati: String
InizioProcesso
  begin
  InserimentoDati(): (Dati);
  if (Dati.isCorrect)
    Accesso(Dati);
    Operazioni();
  else goto begin;
  FineProcesso
  FineSpecifica
  
```

Capitolo 6

Test e validazione del software

6.1 Installazione

Inline è stato pensato e costruito sul web framework **Ruby on Rails**, pertanto, la compatibilità con i sistemi dipende esclusivamente dal framework. Esso è disponibile per i più popolari come Linux, Mac OS e Windows; E' possibile utilizzare terze parti o il terminale per installarlo.

Abbiamo bisogno di **Ruby** installato nel sistema operativo, alla versione **2.5.5** come specificato nel Gemfile del software. Successivamente si necessita di alcune librerie fondamentali per lo sviluppatore installabili con il comando

```
sudo apt-get update && sudo apt-get install build-essential -y
```

Successivamente, dobbiamo scaricare la cartella che contiene il software. E' necessario avere git installato nel proprio sistema operativo, se non lo è al momento, bisogna scrivere nel terminale:

```
sudo apt-get update && sudo apt-get install git -y
```

Dopocì, bisogna effettuare il clone della cartella e abbiamo due alternative: Scaricarla da git attraverso il link: <https://github.com/Mattens15/Inline-LASESI>, oppure da terminale digitare:

```
git clone https://github.com/Mattens15/Inline-LASESI
```

Adesso, in ordine, scrivere i seguenti comandi dal terminale per installare le gemme e avviare il server di rails.

```
cd Inline-LASESI  
bin/bundle install  
bin/rake db:create  
bin/rails s
```

6.2 Testing

Il testing è una pratica fondamentale che si svolge dopo lo stesura del codice. Serve a prevenire e correggere importanti bug che minano la sicurezza dell'applicativo. Ci affidiamo a due strumenti forniti dalla gemme di RoR, **Rspec** e **Cucumber**.

Il periodo di testing è durato all'incirca 1 settimana, attraverso il quale sono stati riscontrati errori di sicurezza nella validazione dell'utente. Una qualsiasi persona, con una POST poteva tranquillamente validare un utente.

6.2.1 Unit test

In *Ingegneria del software*, per unit test si intende l'attività di testing di singole unità di software. Per unità si intende normalmente il minimo componente di un programma dotato di funzionamento autonomo.

In Inline i test di unità sono stati svolti con **Rspec** e si trovano in `./spec/`. Quindi ogni controller, model e view è stata testata con Unit test.

Prendiamo per esempio il codice di test del controller powers

file `./spec/controller/powers_controller_spec.rb`

```
RSpec.describe PowersController, type: :controller do
  before(:each) do
    @owner = FactoryBot.create(:owner)
    @user = FactoryBot.create(:user)
    @room = @owner.rooms.create!(attributes_for(:valid_room))
  end

  describe "POST #create as room host" do
    it "returns http success" do
      allow(controller).to
        receive(:current_user).and_return(@owner)

      expect{post :create,
        params: {room_id: @room.id, power: {user_id:
          @user.username}}}.to change
        {@user.powers.count}.by(1)
      expect(response).to
        redirect_to(edit_room_path(@room.hash_id))
    end
  end

  //Other tests
end
```

Dalla prima riga, capiamo che si sta testando un controller, in particolare PowersController.

`before(:each)` do implica che prima di ogni test descritto all'interno di questa funziona crei prima un proprietario, un utente e una stanza.

Nel test soprastante stiamo testando il fatto che un utente che abbia creato la stanza in precedenza abbia la possibilità di aggiungere altri admin. Abbiamo supposto che il `current_user` fosse il proprietario.

Nell' `expectation` abbiamo effettuato una `post`, nei parametri è presente l'id della room e l'id dell'utente che vogliamo far diventare admin.

Alla fine ci aspettiamo che il controller reindirizza nel percorso `edit_room_path`.

6.2.2 Integration test

I test di integrazione sono stati svolti con Cucumber, un tool che permette di effettuare BDD, *Behavior driven development*. Questi tests permettono di verificare l'integrità e l'insieme delle parti, testate nel test di unità, quando devono compiere un'operazione. Queste operazioni sono per noi le *User stories* descritte nel capitolo 2.

I test descritti da cucumber si trovano in `./features`. La peculiarità di questo strumento è quella di scrivere test in linguaggio di alto livello, quasi parlato; Successivamente, un modulo in linguaggio ruby preleverà gli step con delle espressioni regolari e effettuare dei test.

Prendiamo per esempio il test della user story 18.

As a User I want to use Mapbox so that i can see rooms in a certain location.

file `PASSED_find_room_by_location.feature`

```
@javascript
Scenario:
  Given I am a registered user
  And I log in
  When I create a room
  And I visit dashboard
  And I fill Search with Via del babuino
  And I pick the first one
  Then I should see marker
```

`@Javascript` indica al motore che esegue i test che avrà bisogno dell'appoggio di Javascript nel browser per far sì che i test vengano superati.

Given, *When* e *Then* rappresentano le parole chiave per i test. In ordine raffigurano il preambolo dello scenario, poi l'azione e alla fine la conseguenza.

In questo caso: **Dato** che sono registrato e loggato, **quando** creo una stanza e visito la dashboard e scrivo *Via del Babuino* nella barra di ricerca, **dovrei** vedere un marker sulla mappa.

Capitolo 7

Conclusioni

Inline è un applicazione e soprattutto accessibile. Non ha bisogno di particolari dati dell'utente per funzionare.

E' stata la prima volta che ho effettuato qualcosa in un team, ho incontrato tanto difficoltà a far combaciare il mio codice con quello degli altri *Coff coff*, *git merge*. E' stata sicuramente un esperienza positiva che prima o poi, in questo campo deve avvenire. Mi ha permesso di avvicinarmi ad un campo che non avevo idea mi piacesse, come quello del web development.

Bibliografia

- M. Hartl, Ruby on Rails tutorial, WordlBridge Education.
- G. De Giacomo, Dispense del corso Progettazione del software.
- M. Lenzerini, Dispense del corso Basi di Dati.
- D. Chelimsky, D. Astels. The RSpec Book: Behaviour-Driven Development with RSpec, Cucumber, and Friends.
- <https://www.mvmnet.com/it/ssl>, Certificato SSL
- Stackoverflow