

Progetto di Sistemi
Distribuiti

Aurora Laghi

Anna Vitali

Elena Yan

Setting Up Home

Introduzione

Si è pensato di realizzare un'applicazione che consenta di gestire e pianificare diverse attività domestiche fra diversi utenti, in un contesto di convivenza.

- Un utente per poter iniziare ad utilizzare l'applicazione deve per prima cosa **registrarsi**, dopodiché avrà la possibilità di:
 - Creare un nuovo spazio rappresentante la casa,
 - Unirsi a uno spazio esistente.
- All'interno di uno spazio vi possono essere due tipi di utente: **amministratore** e **membro**
 - L'**utente creatore** di uno spazio viene designato **amministratore** di default,
 - Un **utente che si unisce ad uno spazio**, viene designato **membro** di default.

Introduzione

Un utente **amministratore** ha la possibilità di:

- Pianificare e gestire i turni di uno spazio per le diverse faccende,
- Nominare un partecipante membro amministratore,
- Eliminare un partecipante,
- Consentire a un utente di accedere ad uno spazio.

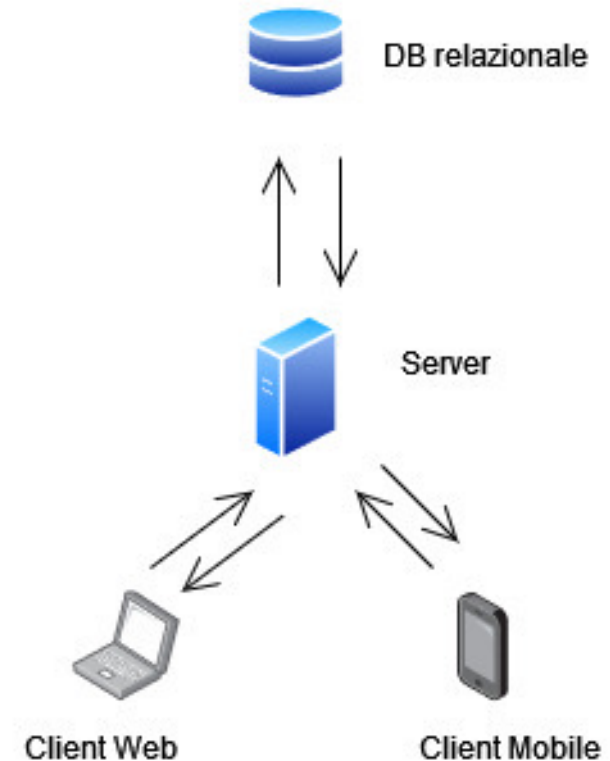
Un utente **membro** avrà la possibilità di

- Visualizzare le informazioni degli altri partecipanti, delle faccende e dei turni,
- Segnare lo svolgimento di un turno e eventualmente ricordare a un utente di svolgerlo,
- Inviare e ricevere notifiche dal sistema, relative ai turni.

Analisi e modello del dominio

Le entità base del dominio modellate sono:

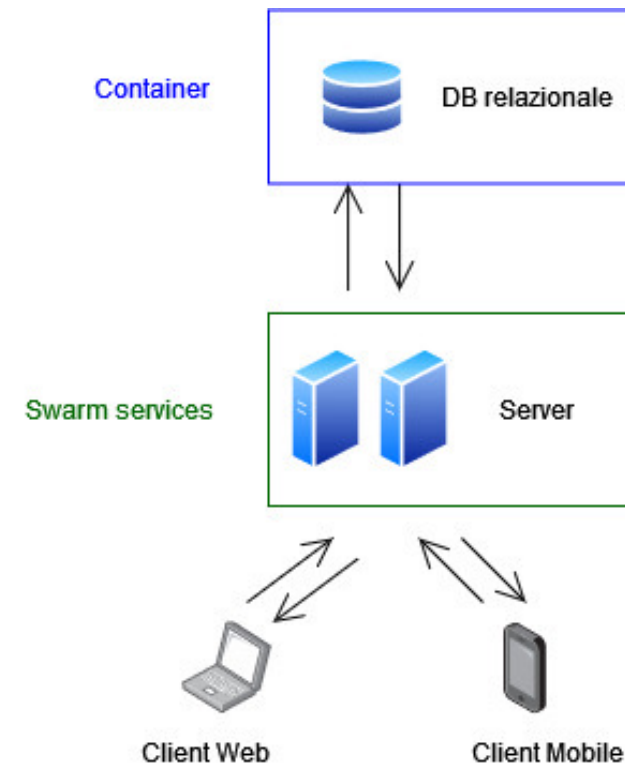
- Database relazionale,
- Server,
- Client Web,
- Client Mobile.



Struttura finale realizzata

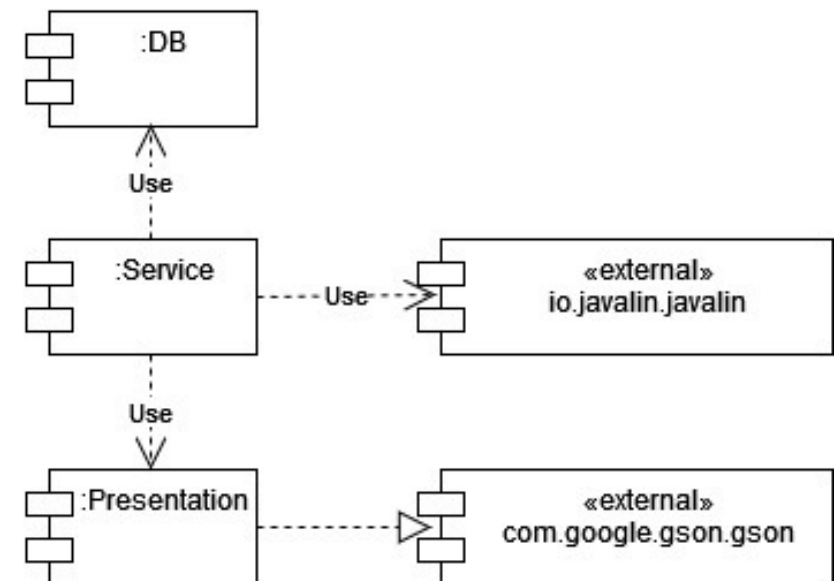
Per la realizzazione del servizio si è pensato di effettuare più **repliche** del servizio offerto dal Server per far fronte alle diverse richieste dei clients, attraverso l'utilizzo di:

- Docker compose,
- Docker swarm -> il quale mette a disposizione un servizio di **load balancing**



Server

- Il compito principale del Server è quello di **gestire le richieste** provenienti dai clients, attraverso **l'interazione** con il database
- Il Server realizzato si compone di **tre** moduli
- Si tratta di un Server di tipo **stateless**,
 - Ogni richiesta effettuata dai clients deve contenere **tutte le informazioni**, per intraprendere l'azione giusta
- L'interfaccia del servizio offerto è stata formalizzata tramite **OpenAPI**



Client Mobile e Web

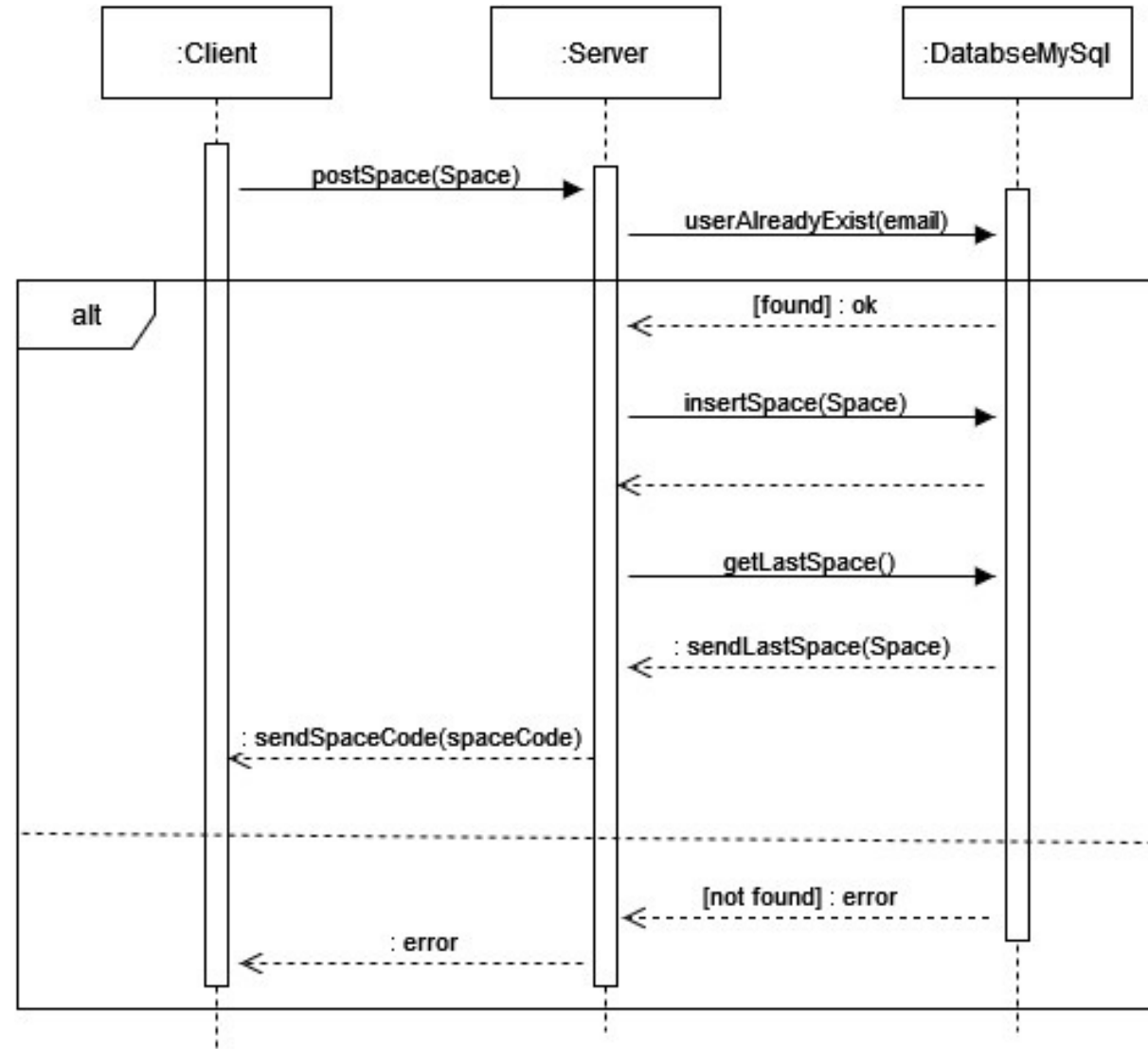
Il Client si occupa di **gestire l'interazione** con l'utente e di compiere le operazioni che esso richiede tramite l'**invio** di apposite richieste al Server e la **gestione** delle risposte.

Le operazioni che l'utente effettua determinano **il comportamento** del Client.

Sono stati realizzati due frontend:

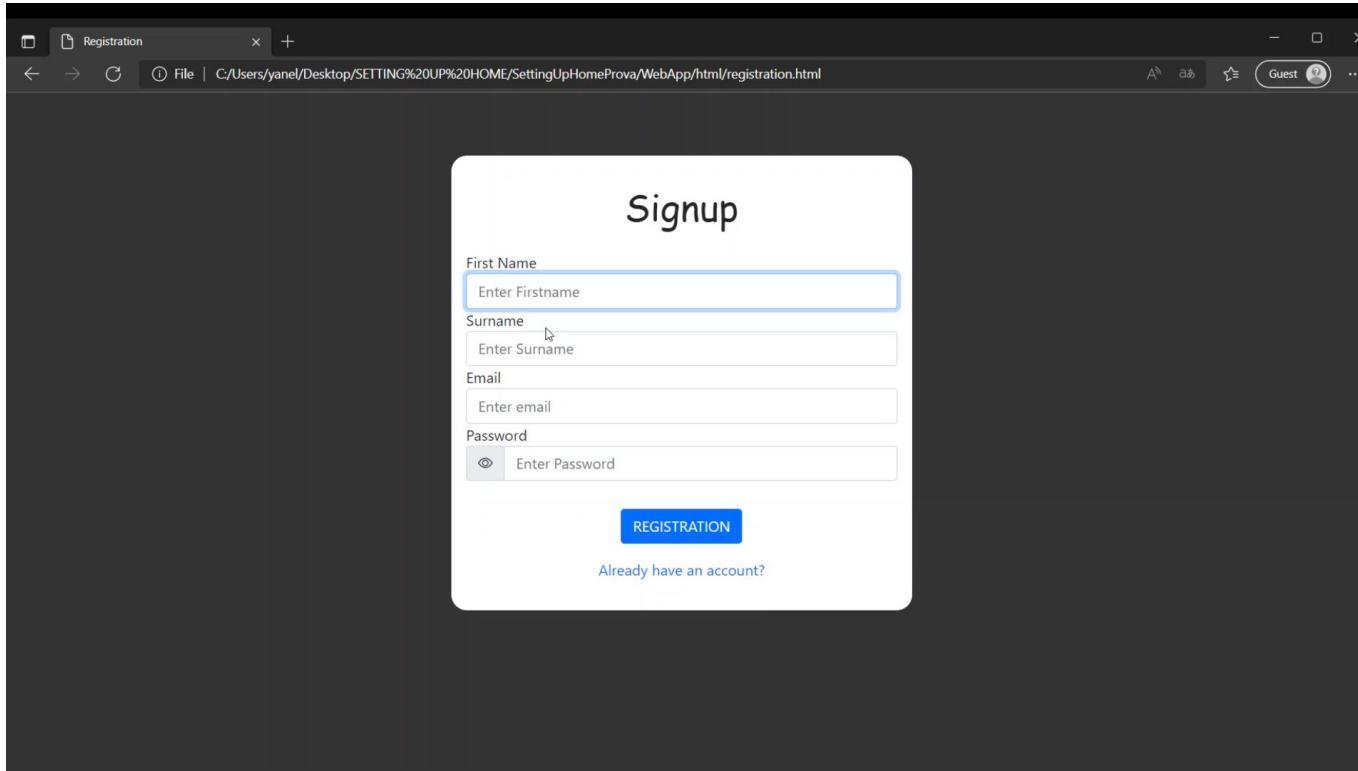
- **Mobile**: è stata realizzata un' applicazione **Android** sviluppata con il linguaggio Java.
 - Suddivisa in **moduli** e aderente al pattern **MVVM**,
 - Si è fatto utilizzo della libreria **Volley** per la gestione degli aspetti di networking.
- **Web**: è stata realizzata un' applicazione Web utilizzando HTML, CSS e Javascript.
 - Si è fatto utilizzo della libreria **jQuery** ed il supporto ad **AJAX** che essa offre per la gestione degli aspetti di networking.

Interazione

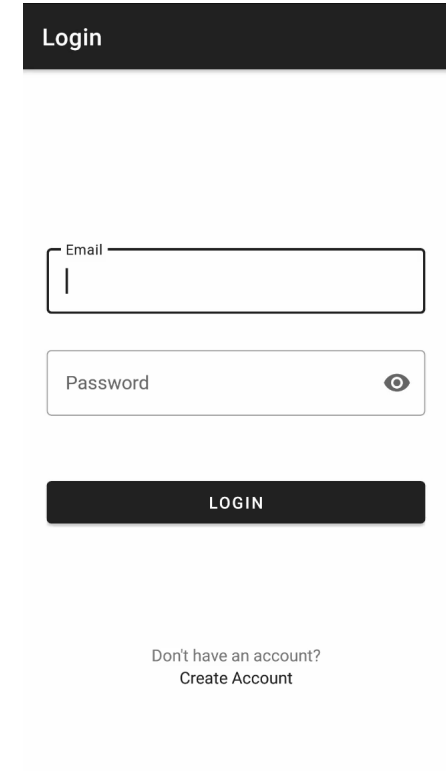


Esempio di utilizzo

Supponiamo che Alice vuole utilizzare l'applicazione per gestire le faccende della casa con Bob.



The screenshot shows a web browser window with a dark theme. The address bar displays the file path: C:/Users/yanel/Desktop/SETTING%20UP%20HOME/SettingUpHomeProva/WebApp/html/registration.html. The page content features a white 'Signup' form with the following fields: 'First Name' (placeholder: Enter Firstname), 'Surname' (placeholder: Enter Surname), 'Email' (placeholder: Enter email), and 'Password' (placeholder: Enter Password with a toggle icon). A blue 'REGISTRATION' button is at the bottom of the form, and a link 'Already have an account?' is positioned below it.

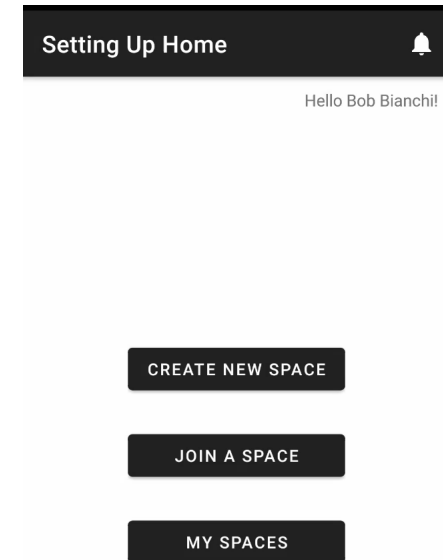
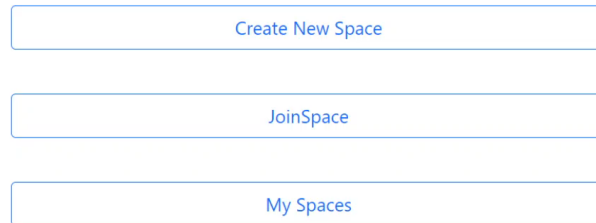
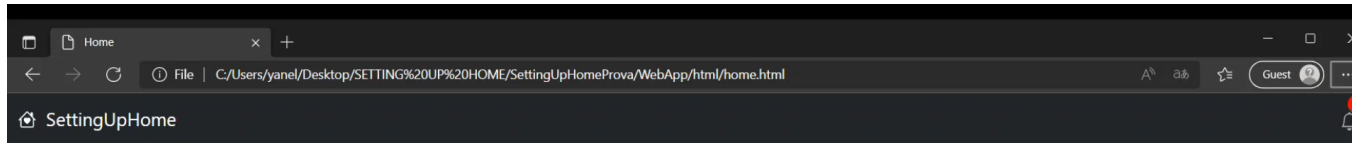


The screenshot shows a web browser window with a dark theme. The page content features a white 'Login' form with the following fields: 'Email' and 'Password' (with a toggle icon). A black 'LOGIN' button is located below the password field. At the bottom of the form, there is a link 'Don't have an account? Create Account'.

Se sono dei nuovi utenti, si dovranno **registrare all'applicativo**, altrimenti potranno procedere direttamente con il **login**.

Esempio di utilizzo

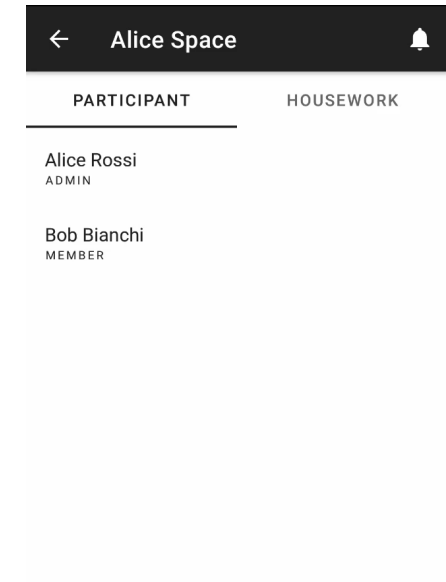
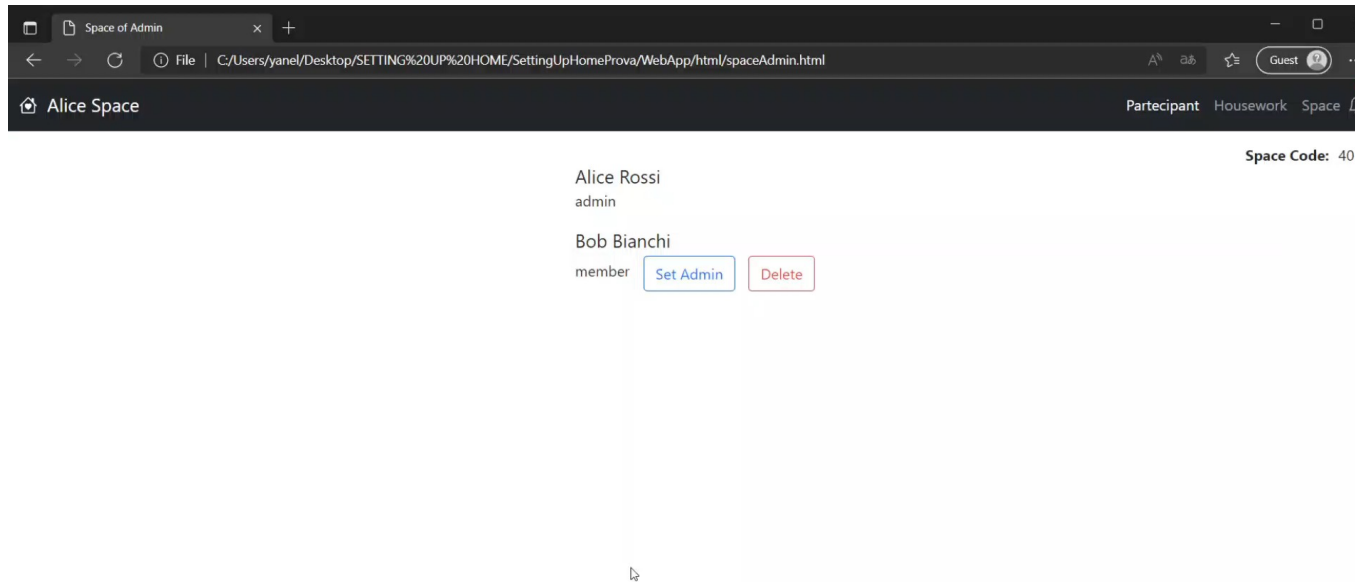
Una volta entrati nell'applicazione, Alice procede alla **creazione di un nuovo spazio**



Successivamente condivide il codice dello spazio a Bob in modo che Bob possa **partecipare**.

Esempio di utilizzo

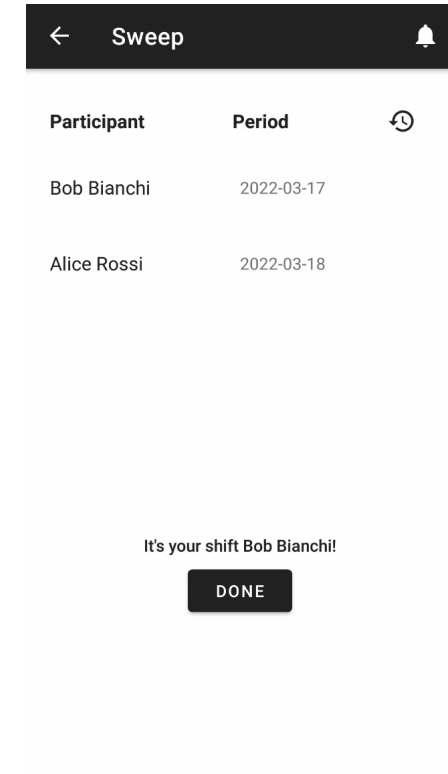
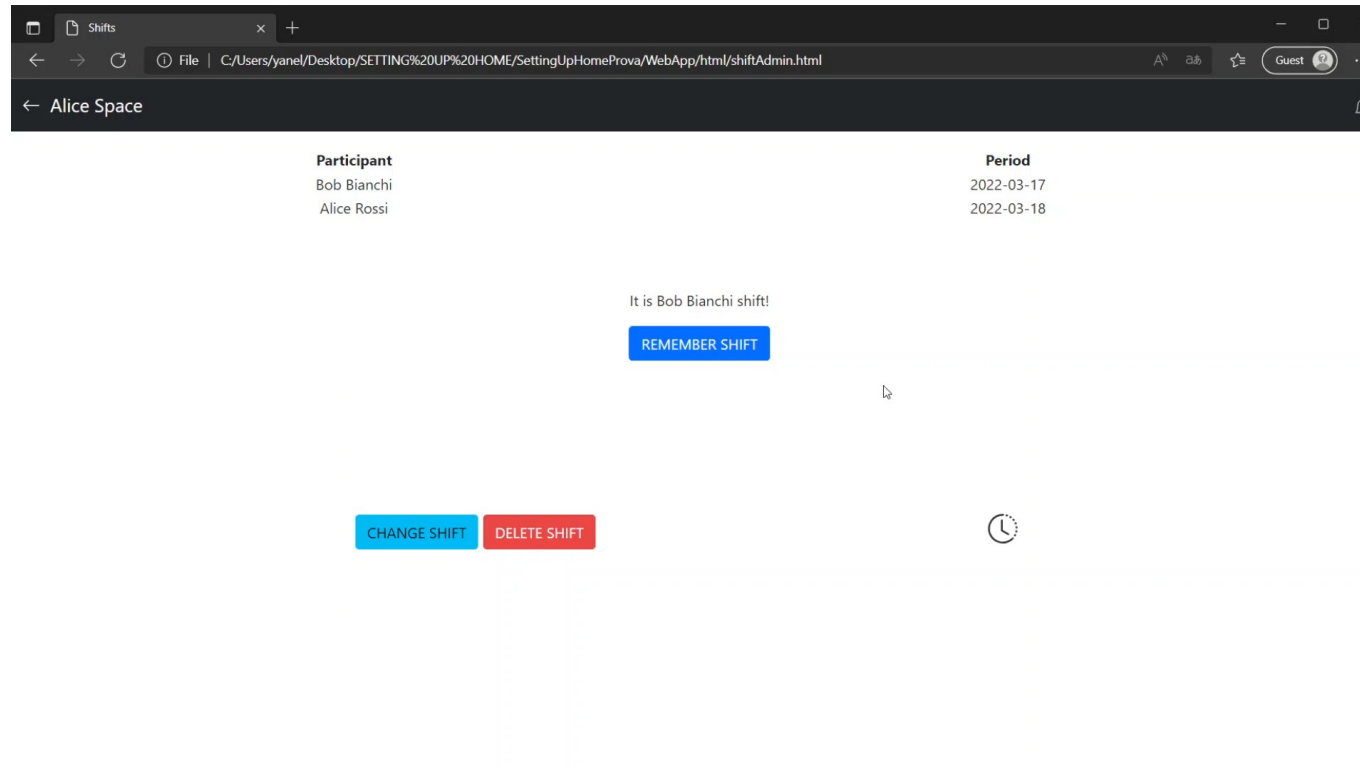
Alice vede che Bob è entrato e decide di **assegnare una nuova faccenda**.



Bob **riceve la notifica** della faccenda e può visualizzarne i turni

Esempio di utilizzo

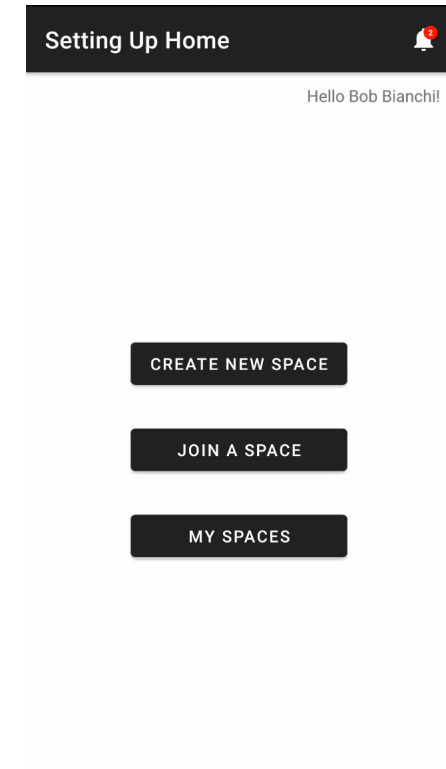
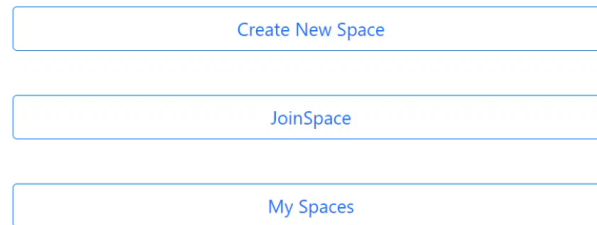
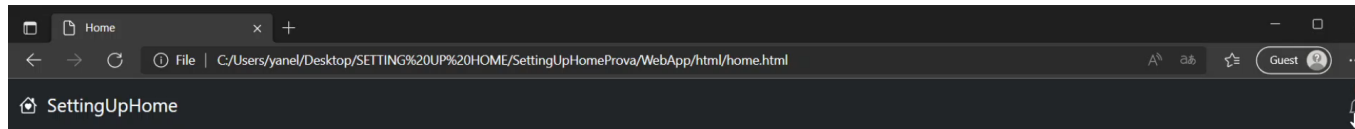
Se Bob non ha ancora svolto la faccenda, Alice glielo può **ricordare**.



Una volta che Bob ha svolto la faccenda, può **segnare il turno come fatto**.

Esempio di utilizzo

Infine è possibile visualizzare le notifiche dell'utente nell'apposita pagina.



Conclusioni

Tutte le diverse funzionalità realizzate sono state **testate** sia mediante l'utilizzo di test **automatici** che **manuali** e analizzando quello che è stato fatto, possiamo affermare che il sistema soddisfa gli obiettivi e i requisiti individuati e che si è complessivamente soddisfatti del risultato raggiunto.

Possibili funzionalità aggiuntive dell'applicazione, che possono essere realizzate in futuro possono essere le seguenti:

- Consentire agli utenti di avere un **calendario condiviso**,
- Consentire la **gestione delle spese** della casa,
- Mettere a disposizione una **chat** per la comunicazione.