

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

CORSO DI LAUREA IN INFORMATICA



Titolo della tesi

Tesi di laurea

Relatore

Prof. Francesco Ranzato

Laureando

Luciano Wu

ANNO ACCADEMICO 2021-2022

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

— Oscar Wilde

Dedicato a ...

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di circa trecentoventi ore, dal laureando Luciano Wu presso l'azienda Azienda SyncLab S.p.A.

Gli obiettivi da raggiungere erano molteplici. In primo luogo era richiesto lo sviluppo di una web-app per la gestione delle ordinazione dei piatti di un ristorante sushi all-you-can-eat. Tale applicazione permette di registrare gli ordinazioni e tenere in traccia i piatti in arrivo. Le componenti fondamentali richiesti sono gestione del tavolo, gestione delle ordinazioni e infine gestione dell'accesso. Terzo ed ultimo obiettivo era l'integrazione delle varie maschere e produrre un documento tecnico che descrive le interfacce sviluppate.

“Life is really simple, but we insist on making it complicated”

— Confucius

Ringraziamenti

Innanzitutto, vorrei esprimere la mia gratitudine al Prof. Francesco Ranzato, relatore della mia tesi, per l'aiuto e il sostegno fornitomi durante la stesura del lavoro.

Ringrazio Fabio Pallaro e tutta SyncLab, per avermi dato la possibilità di svolgere lo stage e il progetto per misurare le mie capacità e imparare delle nuove.

Desidero ringraziare con affetto i miei genitori per il sostegno, il grande aiuto e per essermi stati vicini in ogni momento durante gli anni di studio.

Ho desiderio di ringraziare poi i miei amici, in modo particolare Diego e Samuele, per tutti i bellissimi anni passati insieme e le mille avventure vissute.

Padova, Luglio 2022

Luciano Wu

Indice

1	Introduzione	1
1.1	SyncLab	1
1.1.1	Prodotti	2
1.2	Organizzazione del testo	3
2	Il progetto di stage	5
2.1	L'idea	5
2.2	La soluzione individuata	5
2.2.1	Pianificazione del lavoro	6
2.3	obiettivi richiesti	7
2.3.1	Notazione	7
2.3.2	obiettivi fissati	7
2.4	Modalità di lavoro	7
2.5	Strumenti di comunicazione	8
2.5.1	Google Meet:	10
3	Analisi dei requisiti	13
3.1	Descrizione generale	13
3.1.1	interfacce della web-app	13
3.1.2	Caratteristiche degli Utenti	14
3.1.3	Tecnologie utilizzate	14
3.1.4	Descrizione delle tecnologie	14
3.2	Casi D'uso	16
3.2.1	Introduzione	16
3.2.2	Attori primari	16
3.2.3	Utente generico	17
3.2.4	UC1 - Visualizza menù	17
3.2.5	UC1.1 - Visualizza categorie	18
3.2.6	UC1.2 - Visualizza piatto	18
3.2.7	UC1.3 - Aumenta quantità	18
3.2.8	UC1.4 - Diminuisce quantità	19
3.2.9	UC1.5 - Visualizza dettaglio piatto	19
3.2.10	UC1.6 - Nascondi dettaglio piatto	19
3.2.11	UC2 - Gestione tavolo	20
3.2.12	UC2.1 - Generazione sessione tavolo	21
3.2.13	UC2.2 - Unione sessione tavolo	21
3.2.14	UC2.3 - Uscita sessione tavolo	21
3.2.15	UC2.4 - Generazione QR-code sessione tavolo	22

3.2.16	UC3 - Lista ordini	22
3.2.17	UC3.1 - Visualizza lista ordini del tavolo	23
3.2.18	UC3.2 - Visualizza lista ordini personali	23
3.2.19	UC3.3 - Visualizza lista ordini in arrivo	23
3.2.20	UC3.1.1 - Sposta la lista in arrivo	24
3.2.21	UC3.1.2 - Mostra QR-code ordini	24
3.2.22	UC3.2.1 - Visualizza in dettaglio lista ordini personali	25
3.2.23	UC3.3.1 - Ricezione piatto	26
3.3	Utente Non Autenticato	27
3.3.1	UC4 - Area personale	27
3.3.2	UC4.1 - Registrazione	28
3.3.3	UC4.2 - Login	28
3.3.4	UC4.3 - Password dimenticata	29
3.4	Utente Autenticato	29
3.4.1	UC4.4 - Logout	29
3.4.2	UC4.5 - Aggiungi ingredienti non voluti	30
3.4.3	UC4.6 - Rimuovi ingredienti non voluti	30
3.4.4	UC1.7 - Aggiungi preferiti	31
3.4.5	UC1.8 - Rimuovi preferiti	31
3.4.6	UC1.9 - Aggiungi recensione	32
3.4.7	UC5 - Visualizza lista preferiti	32
3.5	Requisiti	33
3.5.1	Introduzione	33
3.5.2	Lista dei requisiti	33
4	Progettazione e codifica	37
4.1	Progettazione	37
4.1.1	Architettura Angular	37
4.1.2	Architettura SushiLab	38
4.1.3	Progettazione API	39
4.1.4	Progettazione delle viste	40
4.2	Codifica	42
4.2.1	Interfacce	42
5	Verifica e validazione	59
5.1	Verifica	59
5.2	Validazione e collaudo	60
6	Conclusioni	61
6.1	Raggiungimento degli obiettivi	61
6.1.1	Obiettivi fissati	61
6.1.2	svolgimento del lavoro	62
6.2	Conoscenze acquisite	62
6.3	Valutazione personale	63
	Bibliografia	67

Elenco delle figure

1.1	Punti di forza di SyncLab	1
1.2	I vari prodotti di SyncLab	2
2.1	Diagramma dei componenti per la web-app	6
2.2	Canale SyncLab su Discord	8
2.3	Borad di Trello per il progetto sushi-lab	9
2.4	Calendario personale per il progetto sushi-lab	9
2.5	Repo su GitHub per il progetto sushi-lab	10
2.6	Interfaccia di Visual Studio Code	11
3.1	Logo di Angular	15
3.2	Logo di HTML e CSS	15
3.3	Logo di Stoplight	16
3.4	Logo di Angular Material	16
3.5	Use Case - UC 1	17
3.6	Use Case - UC 1.1, UC 1.2, UC 1.3, UC 1.4, UC 1.5, UC 1.6	18
3.7	Use Case - UC 2	20
3.8	Use Case - UC 2.1, UC 2.2, UC 2.3, UC 2.4	21
3.9	Use Case - UC 3	22
3.10	Use Case - UC 3.1, UC 3.2, UC 3.3	23
3.11	Use Case - UC 3.1.1, UC 3.1.2	24
3.12	Use Case - UC 3.2.1	25
3.13	Use Case - UC 3.3.1	26
3.14	Use Case - UC 4	27
3.15	Use Case - UC 4.1, UC 4.2, UC 4.3	28
3.16	Use Case - UC 4.4, UC4.5, UC4.6	29
3.17	Use Case - UC 1.7, UC1.8, UC1.9	31
3.18	Use Case - UC 5	32
4.1	Architettura Angular	38
4.2	Struttura file SushiLab	39
4.3	Stoplight SushiLab	40
4.4	Figma SushiLab	41
4.5	Menù e Menù con piatto in dettaglio di SushiLab	42
4.6	Maschera gestione tavolo SushiLab	43
4.7	Maschera unisci sessione SushiLab	44
4.8	Maschera QR-code tavolo SushiLab	45
4.9	Maschera lista ordini del tavolo e QR-code ordini SushiLab	46

4.10	Maschera ordini personali e ordini personali con piatti in dettaglio SushiLab	47
4.11	Maschera ordini in arrivo SushiLab	48
4.12	Maschera login e login con campi errati SushiLab	49
4.13	Maschera registrazione e registrazione con campi errati SushiLab	50
4.14	Maschere cambia password e cambia password con i campi errati SushiLab	51
4.15	Maschera area personae e blacklist SushiLab	52

Elenco delle tabelle

6.1	Tabella raggiungimento degli obiettivi fissati.	61
-----	---	----

Capitolo 1

Introduzione

1.1 SyncLab

SyncLab è una Innovative Company collocata in tutta Italia, è nata nel 2002 con sede principale a Napoli ed è cresciuta velocemente. Attualmente, SyncLab ha 6 sedi in tutta Italia, più di 300 dipendenti e più di 150 clienti diretti e finali.

SyncLab propone servizi innovativi che aiutano i clienti nella realizzazione, progettazione e manutenzioni di soluzioni IT. L'azienda ha collaborato con varie compagnie tra questi i più importanti sono: Tim, Trenitalia, HM, Grimaldi Lines, notartel, sky, eni, enel, Vodafone, RayWay, Poste Italiane, Intesa Sanpaolo, Ministero dell'economia delle finanze, fastweb e UniCredit.



Figura 1.1: Punti di forza di SyncLab

1.1.1 Prodotti

Come accenato, SyncLab opera nel settore IT e i suoi prodotti nascono dalle competenze acquisite e maturate durante le loro 20 anni di collaborazioni. I prodotti coprono vari ambiti come quelli delle telecomunicazioni, utilities, finanza e salute.

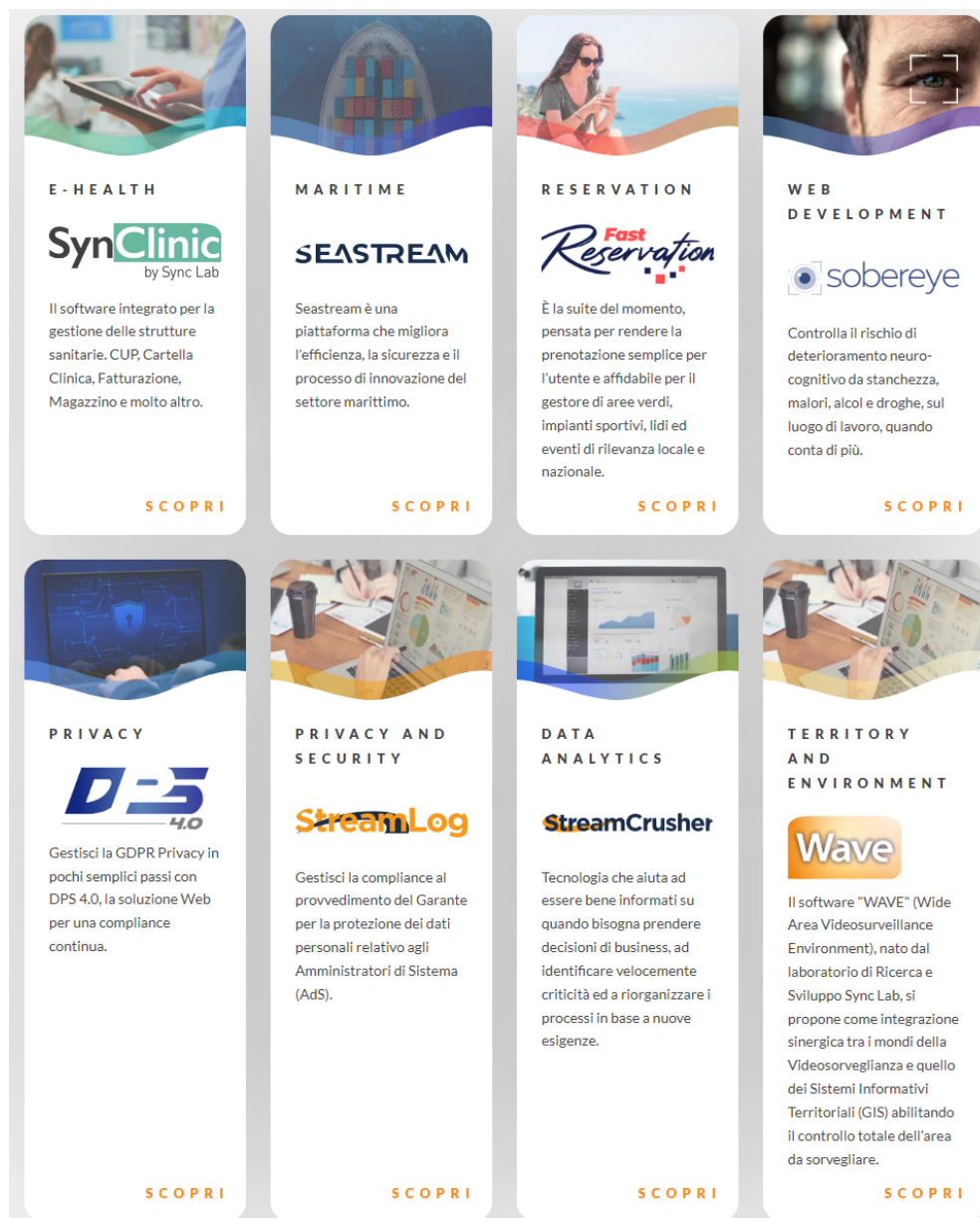


Figura 1.2: I vari prodotti di SyncLab

- * **SynClinic:** software integrato per la gestione delle strutture sanitarie, come il sistema di cartella clinica digitale, il servizio di fatturazione, la gestione informatizzata dei farmaci, i strumenti nativi di gestione amministrativa e tanto

altro.

- * **SEASTREAM:** una piattaforma nata per migliorare e potenziare le attività di business nel settore armatoriali e di altri operatori del mercato marittimo
- * **FastReservation:** applicazione realizzata per rendere il sistema di prenotazione più facile e affidabile per gli utenti.
- * **Sobereye:** soluzione per la sicurezza proattiva per la prevenzione degli incidenti nei settori di trasporti, estrazione, costruzioni e industriale.
- * **DPS 4.0:** applicativo che permette di gestire la GDPR Privacy Policy. L'applicativo, tramite una guida semplice sviluppata dagli ingegneri esperti nell'ambito dei user experience, offre la possibilità di modificare e aggiornare i documenti sulla privacy con il minimo sforzo.
- * **StreamLog:** soluzione per la protezione dei dati personali relativo agli Amministratori di Sistema, dà possibilità di effettuare il controllo degli accessi degli utenti ai sistemi in modo semplice ed efficace.
- * **StreamCrusher:** tecnologia che serve per aiutare a prendere decisioni di business, identificando velocemente i punti critici e riformando i processi in base alle nuove esigenze.
- * **Wave:** software nato per i sistemi di videosorveglianza, con l'obiettivo di avere una maggiore copertura territoriale col minor numero di telecamere installate e possibilmente utilizzare il minor numero possibile di risorse.

1.2 Organizzazione del testo

Il secondo capitolo descrive in dettaglio il progetto di stage, il problema individuato e una sua soluzione.

Il terzo capitolo descrive l'analisi dei requisiti della web-app, vengono discussi tutti i casi d'uso e i requisiti da rispettare.

Il quarto capitolo approfondisce la progettazione e codifica, elencando i vari componenti e descrivendo le loro funzionalità.

Il quinto capitolo approfondisce la fase di verifica e validazione della web-app.

Nel sesto capitolo vengono trattate le conclusioni dell'intero progetto, parlando del prodotto finale e le competenze acquisite durante lo stage.

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- * gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- * per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola*^[g];
- * i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

Capitolo 2

Il progetto di stage

In questo capitolo, vengono esaminati il problema reale incontrato nei ristoranti sushi e la soluzione individuata dall'azienda. Poi viene descritto il progetto di stage e infine la modalità di lavoro.

2.1 L'idea

Nei ristoranti sushi moderni quasi tutti applicano la formula "all-you-can-eat", con tale formula i ristoranti offrono la possibilità di ordinare senza limiti ad un prezzo fisso. I consumi dei clienti in questi locali è maggiore rispetto ai ristoranti à la carte, perché con la formula "all-you-can-eat" spesso i clienti mangiano oltre il loro livello di sazietà, quindi comporterà un elevato numero di ordinazioni da gestire creando così il problema di capire poi chi ha ordinato i piatti singoli nel momento della consegna. L'idea dunque è stata quella di creare una web-app per la gestione delle ordinazioni dei piatti.

2.2 La soluzione individuata

SyncLab ha deciso di risolvere questo problema tramite una web-app, essa è composta da due componenti la parte front-end e la parte back-end. La parte di back-end dovrà essere implementata tramite spring e ha il ruolo principale di un web server, il quale ha il compito di comunicare con il data-base, dove vengono salvati tutte le informazioni dei piatti e degli eventuali ordini dei ristoranti. Il web server deve offrire delle API, con le quali la parte front-end, che utilizza Angular, comunicherà con il server.

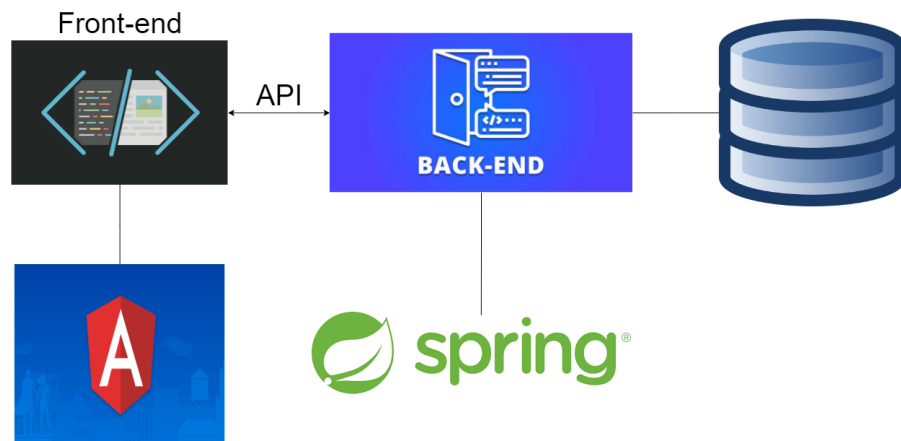


Figura 2.1: Diagramma dei componenti per la web-app

Angular è un framework open source gratuita per lo sviluppo di applicazione web. Il linguaggio di programmazione utilizzato è TypeScript, TypeScript estende la sintassi di JavaScript, quindi qualsiasi codice scritto in JavaScript è eseguibile anche tramite TypeScript senza nessuna modifica o aggiunta di codice. Grazie a TypeScript il codice generato da Angular gira su tutti i principali web browser comuni come Google Chrome, Firefox, Safari, Opera, Microsoft Edge e tanti altri. La web-app, per ciascuna parte dell'applicazione è stata affidata a più persone. Discutendo con il tutor aziendale, Fabio Pallaro, abbiamo individuato i principali obiettivi per realizzare la web-app ed a me è stato assegnato il compito di sviluppare la visuale dell'applicazione.

2.2.1 Pianificazione del lavoro

Pianificazione settimanale

- * Prima Settimana (40 ore)
 - Incontro con persone coinvolte nel progetto per discutere i requisiti e le richieste relativamente al sistema da sviluppare;
 - Presentazione strumenti di lavoro per la condivisione del materiale di studio e per la gestione dell'avanzamento;
 - Condivisione scaletta di argomenti;
 - Ripasso del linguaggio Java SE;
 - Ripasso concetti Web (Servlet, servizi Rest, Json ecc.).
- * Seconda Settimana (40 ore)
 - Studio principi generali di Spring Core (IOC, Dependency Injection);
 - Studio SpringBoot;
 - Studio Spring Data/DataRest.
- * Terza Settimana (40 ore)
 - Ripasso linguaggio Javascript;
 - Studio del linguaggio TypeScript.
- * Quarta Settimana (40 ore)
 - Studio piattaforma NodeJS e AngularCLI;
 - Studio framework Angular.
- * Quinta Settimana (40 ore)
 - Analisi e studio del progetto SushiLab;

- Progettazione ed implementazione della nuova maschera di accesso.
- * Sesta Settimana (40 ore)
 - Progettazione ed implementazione nuova maschera "Inserimento Ordine e Gestione Tavolo".
- * Settima Settimana (40 ore)
 - Progettazione ed implementazione nuova maschera "Merge Ordini e Visualizzazione Ordine unico".
- * Ottava Settimana - Conclusione (40 ore)
 - Verifica del funzionamento della web-app;
 - Validazione della web-app;
 - Termine integrazioni e collaudo finale.

2.3 obiettivi richiesti

2.3.1 Notazione

Si farà riferimento ai requisiti secondo le seguenti notazioni:

- * O per i requisiti obbligatori, vincolanti in quanto obiettivo primario richiesto dal committente;
- * D per i requisiti desiderabili, non vincolanti o strettamente necessari, ma dal riconoscibile valore aggiunto;
- * F per i requisiti facoltativi, rappresentanti valore aggiunto non strettamente competitivo.

Le sigle precedentemente indicate saranno seguite da una coppia sequenziale di numeri, identificativo del requisito.

2.3.2 obiettivi fissati

Si prevede lo svolgimento dei seguenti obiettivi:

- * Obbligatori:
 - O01: Acquisizione competenze sulle tematiche sopra descritte;
 - O02: Capacità di raggiungere gli obiettivi richiesti in autonomia seguendo il cronoprogramma;
 - O03: Portare a termine le implementazioni previste con una percentuale di superamento pari al 80%.
- * Desiderabili:
 - D01: Portare a termine le implementazioni previste con una percentuale di superamento pari al 100%.
- * Facoltativi:
 - F01: Apportare un valore aggiunto al gruppo di lavoro durante le fasi di progettazione delle interfacce.

2.4 Modalità di lavoro

L'azienda utilizza lo sviluppo agile del software. La modalità agile aiuta a ridurre il rischio di fallimento sviluppando il software in finestre di tempo limitate chiamate iterazioni che, in genere, durano qualche settimana. Ogni iterazione è un piccolo progetto a sé stante e deve contenere tutto ciò che è necessario per rilasciare un piccolo incremento nelle funzionalità del software: pianificazione, analisi dei requisiti, proget-

tazione, implementazione, test e documentazione. Ogni settimana viene effettuato un incontro con il tutor aziendale per discutere dei problemi trovati durante lo sviluppo e il punto di situazione del progetto.

2.5 Strumenti di comunicazione

Per avere una buona comunicazione con l'azienda si è deciso di utilizzare:

Discord:

una piattaforma di VoIP, messaggistica istantanea e distribuzione digitale. Su discord gli utenti comunicano con chiamate vocali, video chiamate ed è anche possibile condividere lo schermo.

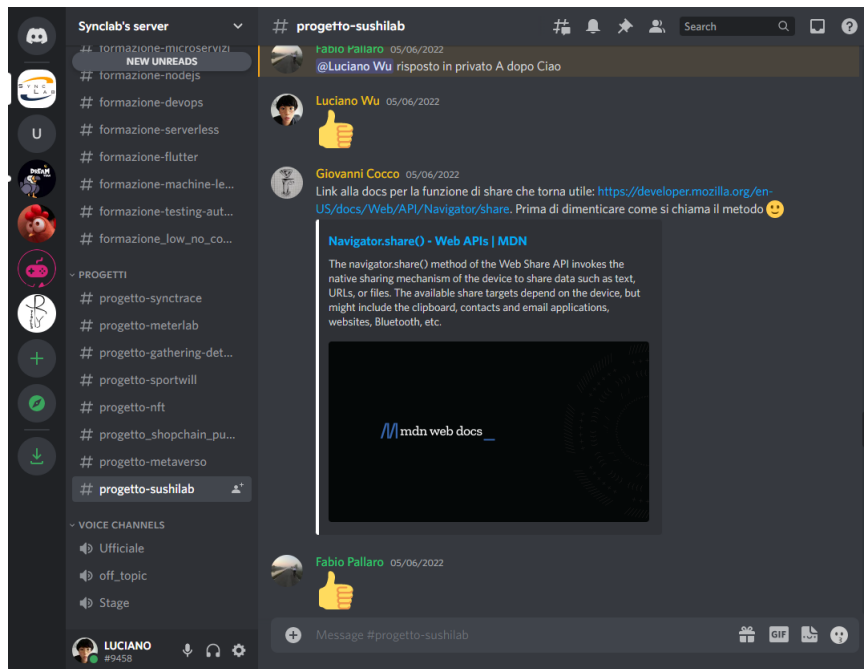


Figura 2.2: Canale SyncLab su Discord

Trello:

un software gestionale in stile Kanban, in cui è possibile pianificare il progetto, condividere lo stato di svolgimento di una card con altri collaboratori, spostare vari card tra le liste e assegnare ad un utente.

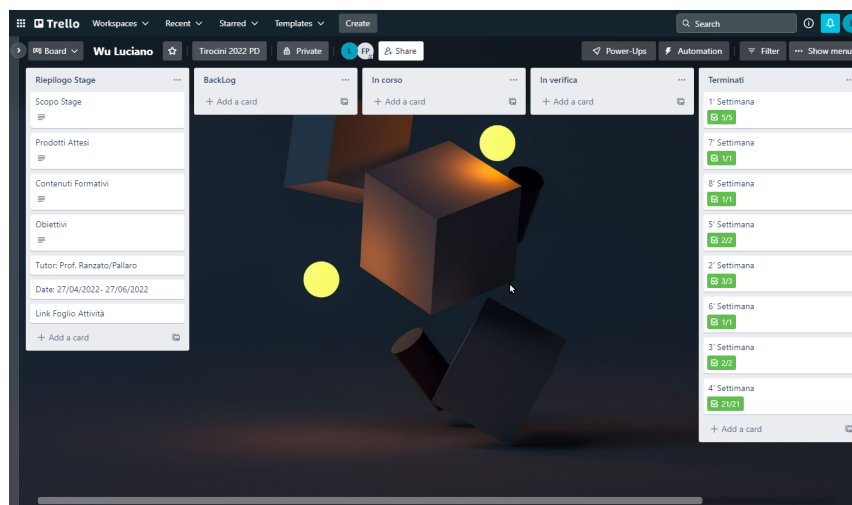


Figura 2.3: Borad di Trello per il progetto sushi-lab

Google Sheets:

una web-app che fornisce tutte le funzionalità di un foglio elettronico, lo abbiamo utilizzato come un diario giornaliero, dove vengono descritti il compito svolto durante una certa gioranata.

Luciano WU - Calendario

File Edit View Insert Format Data Tools Extensions Help Last edit was made yesterday at 5:57...

A1	A	B	C	D	E	F
17	2022-05-18	Studio su lifecycle hooks e comunicazione tra i componenti di angular	✓			
18	2022-05-19	Studio su chiamate http, rest, build e deploy	✓			
19	2022-05-20	Studio su lazy loading e reactive forms	✓			
20	2022-05-23	Progettazione maschera accesso	✓			
21	2022-05-24	Progettazione maschera tavolo	✓			
22	2022-05-25	Progettazione maschera ordini	✓			
23	2022-05-26	Progettazione maschera password dimenticata	✓			
24	2022-05-27	Progettazione guard	✓			
25	2022-05-30	Progettazione maschera gestione tavolo(creazione sessione tavolo)	✓			
26	2022-05-31	Progettazione maschera gestione tavolo(unione sessione tavolo)	✓			
27	2022-06-01	Progettazione maschera gestione ordini del tavolo	✓			
28	2022-06-02	Progettazione maschera gestione ordini personali	✓			
29	2022-06-03	Progettazione maschera gestione ordini in arrivo	✓			
30	2022-06-06	Progettazione merge ordini tavolo	✓			
31	2022-06-07	Progettazione visualizza ordine unico	✓			
32	2022-06-08	Progettazione sposta ordini in arrivo	✓			
33	2022-06-09	Abbellimento css maschera tavolo	✓			
34	2022-06-10	Abbellimento css maschera ordini	✓			
35	2022-06-14	Abbellimento css maschera password dimenticata	✓			
36	2022-06-15	Controllo completo su i file di stile	✓			
37	2022-06-16	Controllo completo su file di guard	✓			
38	2022-06-17	Controllo completo su file di service	✓			
39	2022-06-20	Controllo completo su file di type script	✓			
40	2022-06-21	Stesura documento analisi tecnica	✓			
41	2022-06-22	Completamento documento analisi tecnica	✓			
42	2022-06-23	Preparazione collaudo finale	✓			
43	2022-06-24	Collaudo finale	✓			
44						
45						
46						

Sheet1 Explore

Figura 2.4: Calendario personale per il progetto sushi-lab

2.5.1 Google Meet:

un software nata per le videochiamate sviluppato da Goolge, in cui è possibile mandare messaggi e condividere lo schermo e video contemporaneamente, lo abbiamo utilizzato per alcuni incontri con alcuni collaboratori esterni per analisi dei requisiti.

GitHub:

servizio di hosting per il sviluppo software, è implementato insieme con lo strumetento di cotrollo versione distribuito Git, nel mio caso è utilizzato per condividere e tracciare i file con gli altri collaboratori del progetto.

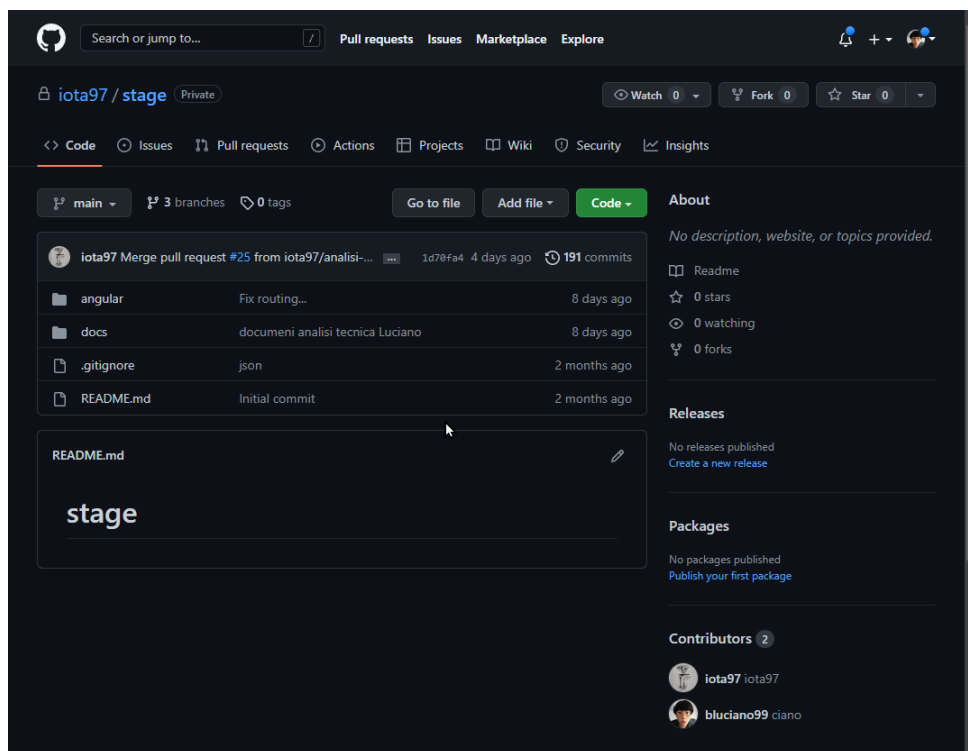


Figura 2.5: Repo su GitHub per il progetto sushi-lab

Visual Studio Code

un editor di codice sorgente sviluppato da Microsoft. Contiene tutte le funzionalità di un editor ed completamente gratuita. Le maggiori funzionalità utilizzate per il progetto sono:

- * Le estensioni di VS CODE per il codice html, css e TypeScript;
- * Il source control di Git;
- * La serach dove è possibile ricercare il termine dentro tutti i file e fare la sostituzione velocemente;

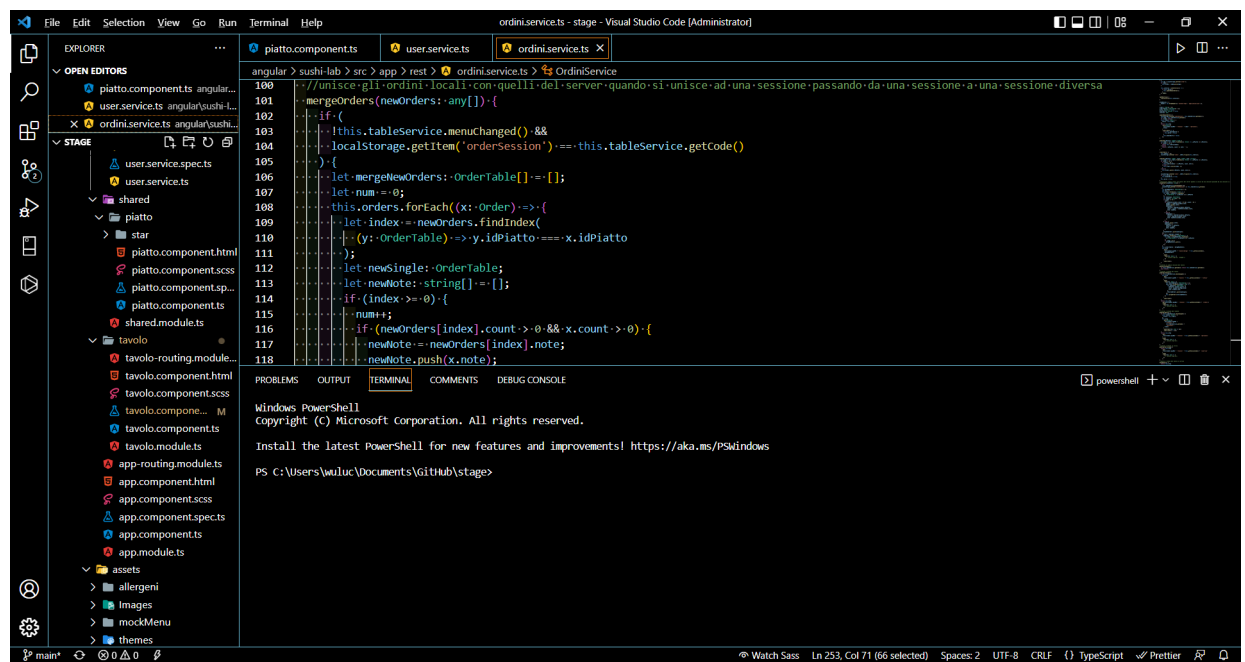


Figura 2.6: Interfaccia di Visual Studio Code

Capitolo 3

Analisi dei requisiti

In questo capitolo vengono trattati le analisi dei requisiti del modulo di front-end della web-app, con i vari casi d'uso ed elenco dei requisiti. Infine vengono elencati i requisiti richiesti dall'azienda.

3.1 Descrizione generale

3.1.1 interfacce della web-app

Interfaccia menù

L'utente può visualizzare il menù di un ristorante dopo aver scansionato il QR-code di un ristorante presente sul tavolo. Il menù è composto da un insieme di categorie, in cui ci sono tutti i piatti appartenenti a quella categoria. I piatti sono ordinati in base al suo id che è un numero univoco dentro ogni menù.

Interfaccia lista ordini

L'utente può vedere i piatti ordinati del suo tavolo di appartenenza, dove ci sono anche i piatti ordinati dalle altre persone del tavolo, inoltre può vedere i suoi piatti personali e i piatti in arrivo.

Interfaccia gestione tavolo

In questa maschera utente può creare una sessione di tavolo se non appartiene ad nessuna sessione, altrimenti può visualizzare il QR-code della sua sessione in modo da fare entrare gli altri nella sua sessione di tavolo.

Interfaccia area personale

Qui l'utente può effettuare la login, di seguito se ha degli allergeni potrà inserire degli ingredienti nella blacklist in modo tale di non visualizzare i piatti contenenti quegli ingredienti nella sezione menù.

3.1.2 Caratteristiche degli Utenti

In questa sezione vengono descritti tutte le Caratteristiche degli utenti che possono utilizzare la web-app.

Utente non autenticato

Con il termine utente non autenticato ci si riferisce ad una qualsiasi persona non autenticata nel sistema, che può sfruttare le funzionalità di base offerte dalla piattaforma, ossia:

- * Visualizzare il menù del ristorante;
- * Visualizzare i singoli piatti in modalità dettaglio;
- * Creare una sessione di tavolo;
- * Unire ad una sessione di tavolo già esistente;
- * Uscire dalla sessione di tavolo;
- * Aggiungere piatti negli ordini;
- * Aggiungere note ai piatti ordinati;
- * Spostare ordini in arrivo;
- * Marcare i piatti in arrivo come arrivato;
- * Registrare nella piattaforma;
- * Effettuare la login.

Utente autenticato

Invece, con il termine “utente autenticato” ci si riferisce ad una persona registrata nel database e che ha effettuato l’accesso nella piattaforma, la quale, oltre a sfruttare le funzionalità dell’utente non autenticato, può anche:

- * Aggiungere piatti nei preferiti;
- * Rimuovere piatti dai preferiti;
- * Dare una recensione ad un piatto;
- * Aggiungere ingredienti non voluti;
- * Rimuovere gli ingredienti non voluti;
- * Visualizzare la lista dei preferiti;
- * Effettuare logout.

3.1.3 Tecnologie utilizzate

Per sviluppare la piattaforma verranno utilizzare le seguenti tecnologie:

- * Angular: per la creazione dell’interfaccia utente;
- * HTML5: per creare la struttura dell’interfaccia utente;
- * CSS3: per lo stile dell’interfaccia, viene utilizzato la sintassi SCSS;
- * Stoplight: per simulare le chiamate Rest API;
- * Angular Material: per la creazione dei componenti.

3.1.4 Descrizione delle tecnologie

Angular

Angular è un framework open source per sviluppare applicazioni web, permette di dividere l’applicazione in più componenti, grazie a questo è possibile riutilizzare lo stesso modulo in più parti della web-app, oltre a questo garantisce una maggiore manutenibilità e espandibilità.



Figura 3.1: Logo di Angular

HTML5

L'HyperText Markup Language, noto come HTML, è un linguaggio di markup più popolare, utilizzato per progettare le strutture dei siti web. Viene utilizzato da Angular per creare la struttura iniziale per le varie interfacce, poi queste strutture vengono modificate da Angular per generare la pagina dinamicamente.



Figura 3.2: Logo di HTML e CSS

CSS3

Il CSS, sigla di Cascading Style Sheets, è il linguaggio utilizzato per modificare il layout delle pagine web, le regole vengono applicate nel ordine in cui vengono scritte. Per il progetto è stato utilizzato una sua estensione SCSS, la quale è compatibile con tutte le versioni di CSS. Tramite la SCSS è possibile dichiarare le regole CSS in blocchi quindi ci aiuta a scrivere regole CSS in più velocemente e comprensibile.

TypeScript

TypeScript è un linguaggio di programmazione open source sviluppato da Microsoft, che estende il classico JavaScript quindi qualsiasi codice scritto in JavaScript è anche eseguibile con TypeScript direttamente senza nessuna modifica. TypeScript rende molto più flessibile e flessibile JavaScript aggiungendo la firma dei metodi, classi, tipi di dato e tanto altro, grazie a queste caratteristiche utilizzando TypeScript ci garantisce controlli automatici, rilevando in automatico i bug prima della compilazione.

Stoplight

Stoplight è una piattaforma che offre la possibilità di progettare le API velocemente, grazie alla sua interfaccia user friendly. Offre un buon spazio per collaborare con gli altri, condividendo tutte le API con le sue descrizioni e risposte in modo chiaro.



Figura 3.3: Logo di Stoplight

Angular Material

Angular Material è un insieme di componenti UI già implementati, questi componenti possono essere direttamente utilizzati in Angular. Tutti i componenti offerti sono già responsive e sono facili da customizzare con le proprie preferenze, come il colore dei bottoni.



Figura 3.4: Logo di Angular Material

3.2 Casi D'uso

3.2.1 Introduzione

In questa sezione verranno presentati i casi d'uso individuati durante la fase di analisi dei requisiti, i quali fanno riferimento a tutte le funzionalità che la web-app SushiLab dovrà offrire ad ogni utente che vorrà interfacciarsi con essa.

3.2.2 Attori primari

- * Utente non Autenticato: utente che non ha ancora effettuato la fase di autenticazione sulla piattaforma. Può essere in possesso o meno delle credenziali per l'autenticazione. Avrà funzionalità limitate rispetto ad un utente autenticato;
- * Utente Autenticato: utente che ha effettuato l'autenticazione alla piattaforma tramite le proprie credenziali. Ha accesso ad ogni funzionalità messa a disposizione dalla piattaforma;
- * Utente Generico: può essere sia un utente autenticato che un utente non autenticato.

3.2.3 Utente generico

3.2.4 UC1 - Visualizza menù

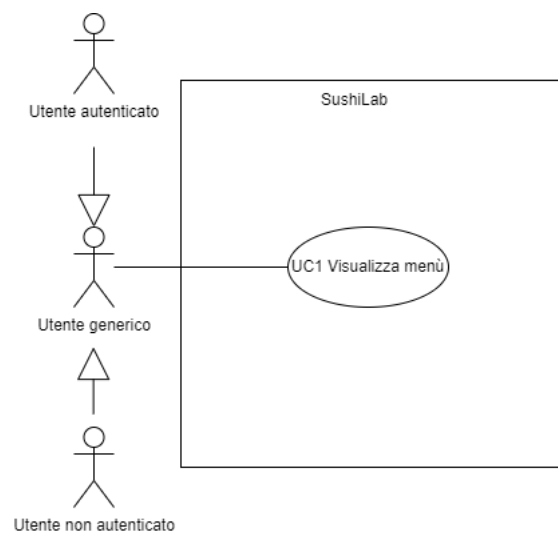


Figura 3.5: Use Case - UC 1

- * **Descrizione:** L'utente visualizza il menù del ristorante.
- * **Attore Primario:** Utente generico.
- * **Precondizione:** L'utente si trova dentro la web-app sushiLab.
- * **Postcondizione:** Viene visualizzato il menù del ristorante.
- * **Scenrio principale:**
 - L'utente si trova dentro il sistema;
 - L'utente clicca sul bottone menù;
 - Viene mostra il menù del ristorante.

3.2.5 UC1.1 - Visualizza categorie

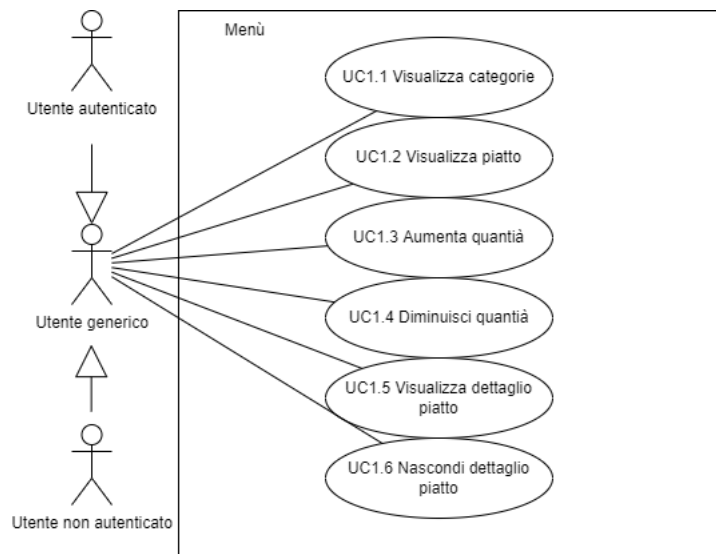


Figura 3.6: Use Case - UC 1.1, UC 1.2, UC 1.3, UC 1.4, UC 1.5, UC 1.6

- * **Descrizione:** L'utente visualizza le categorie del menù.
- * **Attore Primario:** Utente generico.
- * **Precondizione:** L'utente si trova dentro la sezione menù.
- * **Postcondizione:** Viene visualizzato i nomi delle categorie.
- * **Scenrio principale:**
 - L'utente si trova sezione menù;
 - Viene mostrato le categorie del menù.

3.2.6 UC1.2 - Visualizza piatto

- * **Descrizione:** L'utente visualizza i piatti del menù mostrando il numero, nome, prezzo, ingredienti, allergeni, limatazioni e la quantità. La quantità di default è 0 che indica che non è ancora stato ordinato.
- * **Attore Primario:** Utente generico.
- * **Precondizione:** L'utente si trova dentro la sezione menù.
- * **Postcondizione:** Viene visualizzato i piatti del menù.
- * **Scenrio principale:**
 - L'utente si trova sezione menù;
 - Viene mostrato i piatti del menù.

3.2.7 UC1.3 - Aumenta quantità

- * **Descrizione:** L'utente aumenta la quantità di un piatto.
- * **Attore Primario:** Utente generico.
- * **Precondizione:** L'utente si trova dentro la sezione menù o lista degli ordini personali.
- * **Postcondizione:** Viene aggiunto il piatto specifico con la quantità aggiornata negli ordini.

- * **Scenrio principale:**
 - L'utente si trova sezione menù;
 - L'utente clicca sul bottone + di un piatto;
 - Viene aggiunto il piatto negli ordini.
- * **Scenrio alternativo:**
 - L'utente si trova sezione menù o lista degli ordini personali;
 - L'utente clicca sul bottone + di un piatto che è già presente negli ordini;
 - Viene aumentato la quantità del piatto negli ordini.

3.2.8 UC1.4 - Diminuisce quantità

- * **Descrizione:** L'utente diminuisce la quantità di un piatto.
- * **Attore Primario:** Utente generico.
- * **Precondizione:** L'utente si trova dentro la sezione menù o lista degli ordini personali.
- * **Postcondizione:** Viene diminuito la quantità del piatto specifico negli ordini.
- * **Scenrio principale:**
 - L'utente si trova sezione menù o lista degli ordini personali;
 - L'utente clicca sul bottone - di un piatto con quantità maggiore di 1;
 - Viene diminuito la quantità del piatto negli ordini.
- * **Scenrio alternativo:**
 - L'utente si trova sezione menù o lista degli ordini personali;
 - L'utente clicca sul bottone - di un piatto con quantità uguale a 1;
 - Viene rimosso il piatto dagli ordini.

3.2.9 UC1.5 - Visualizza dettaglio piatto

- * **Descrizione:** L'utente visualizza i dettagli di un piatto nel menù, mostrando la recensione del piatto e il text-box per inserire una nota.
- * **Attore Primario:** Utente generico.
- * **Precondizione:** L'utente si trova dentro la sezione menù.
- * **Postcondizione:** Viene visualizzato i dettagli di un piatto specifico.
- * **Scenrio principale:**
 - L'utente si trova sezione menù;
 - L'utente clicca sul bottom mostra dettagli;
 - Viene mostrato i dettagli di un piatto.

3.2.10 UC1.6 - Nascondi dettaglio piatto

- * **Descrizione:** L'utente nasconde i dettagli di un piatto specifico.
- * **Attore Primario:** Utente generico.
- * **Precondizione:** L'utente si trova dentro la sezione menù con un piatto in modalità dettaglio.
- * **Postcondizione:** Viene nascosto i dettagli del piatto specifico.
- * **Scenrio principale:**
 - L'utente si trova sezione menù;
 - L'utente clicca sul bottom nascondi dettagli;
 - Viene nascosto i dettagli del piatto.

3.2.11 UC2 - Gestione tavolo

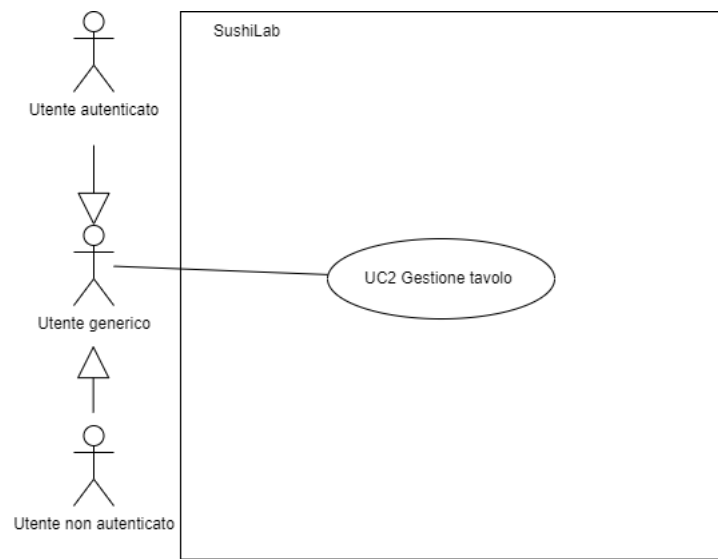


Figura 3.7: Use Case - UC 2

- * **Descrizione:** L'utente visualizza la maschera di gestione tavolo.
- * **Attore Primario:** Utente generico.
- * **Precondizione:** L'utente si trova dentro la web-app sushiLab.
- * **Postcondizione:** Viene visualizzato la maschera di gestione tavolo.
- * **Scenrio principale:**
 - L'utente si trova dentro il sistema;
 - L'utente clicca sul bottone gestione tavolo nella navbar;
 - Viene mostrato la maschera di gestione tavolo.

3.2.12 UC2.1 - Generazione sessione tavolo

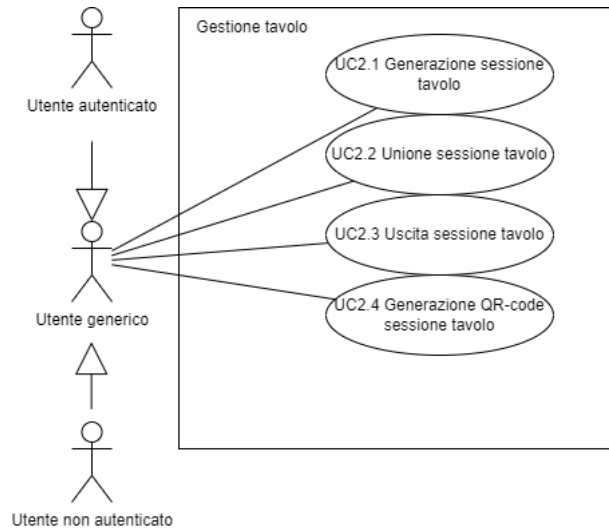


Figura 3.8: Use Case - UC 2.1, UC 2.2, UC 2.3, UC 2.4

- * **Descrizione:** L'utente genera la sessione del tavolo.
- * **Attore Primario:** Utente generico.
- * **Precondizione:** L'utente si trova dentro la sezione gestione tavolo.
- * **Postcondizione:** L'utente entra nella sessione generata del tavolo.
- * **Scenrio principale:**
 - L'utente si trova dentro la sezione gestione tavolo;
 - L'utente clicca sul bottone crea sessione;
 - Viene creata la sessione;
 - L'utente viene inserito nella sessione creata.

3.2.13 UC2.2 - Unione sessione tavolo

- * **Descrizione:** L'utente si unisce alla sessione del tavolo.
- * **Attore Primario:** Utente generico.
- * **Precondizione:** L'utente si trova dentro la sezione gestione tavolo.
- * **Postcondizione:** L'utente entra nella sessione che è stata inserita.
- * **Scenrio principale:**
 - L'utente si trova dentro la sezione gestione tavolo;
 - L'utente clicca sul bottone unisciti a una sessione;
 - L'utente inserisce il numero della sessione;
 - L'utente clicca sul bottone unisciti;
 - L'utente viene inserito nella sessione.

3.2.14 UC2.3 - Uscita sessione tavolo

- * **Descrizione:** L'utente esce dalla sessione del tavolo.
- * **Attore Primario:** Utente generico.
- * **Precondizione:** L'utente si trova dentro la sezione gestione tavolo ed è dentro ad una sessione.

- * **Postcondizione:** L'utente esce dalla sessione generata del tavolo.
- * **Scenrio principale:**
 - L'utente si trova dentro la sezione gestione tavolo;
 - L'utente clicca sul bottone esci dalla sessione;
 - L'utente viene rimosso dalla sessione.

3.2.15 UC2.4 - Generazione QR-code sessione tavolo

- * **Descrizione:** L'utente genera il QR-code dalla sessione del tavolo per mostrarlo agli altri, che li permetterà di unire alla sessione direttamente scansionando il QR-code.
- * **Attore Primario:** Utente generico.
- * **Precondizione:** L'utente si trova dentro la sezione gestione tavolo ed è dentro ad una sessione.
- * **Postcondizione:** L'utente genera il QR-code dalla sessione del tavolo.
- * **Scenrio principale:**
 - L'utente si trova dentro la sezione gestione tavolo;
 - L'utente genera il QR-code della sessione;
 - Vine mostrato il QR-code;

3.2.16 UC3 - Lista ordini

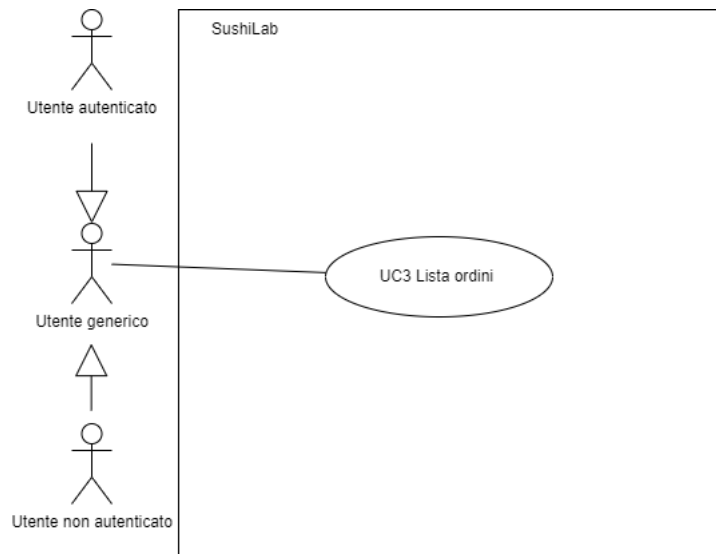


Figura 3.9: Use Case - UC 3

- * **Descrizione:** L'utente visualizza la maschera di gestione ordini.
- * **Attore Primario:** Utente generico.
- * **Precondizione:** L'utente si trova dentro ad una sessione di tavolo.
- * **Postcondizione:** Viene visualizzato la maschera di gestione ordini.
- * **Scenrio principale:**
 - L'utente si trova dentro il sistema con una sessione di tavolo attiva;
 - L'utente clicca sul bottone lista ordini nella navbar;

- Viene mostrato la maschera di gestione ordini.

3.2.17 UC3.1 - Visualizza lista ordini del tavolo

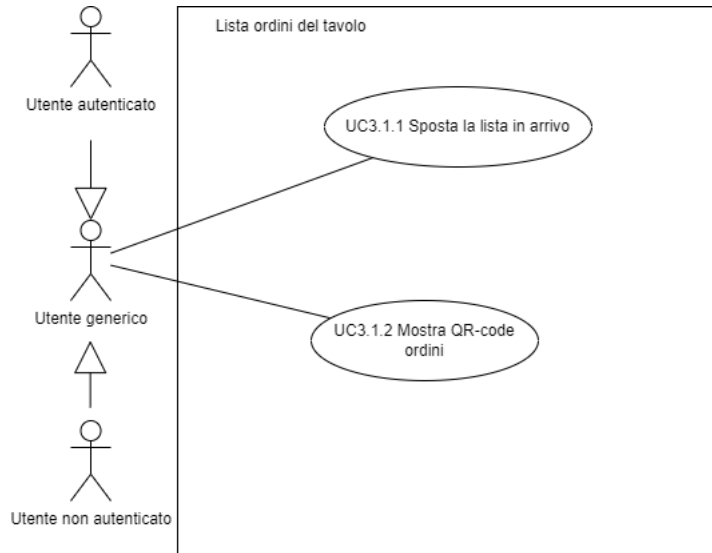


Figura 3.10: Use Case - UC 3.1, UC 3.2, UC 3.3

- * **Descrizione:** L'utente visualizza la lista degli ordini della sessione di tavolo in cui si trova.
- * **Attore Primario:** Utente generico.
- * **Precondizione:** L'utente si trova dentro la sezione lista ordini.
- * **Postcondizione:** Viene visualizzato la lista degli ordini del tavolo.
- * **Scenrio principale:**
 - L'utente si trova dentro la sezione gestione ordini;
 - L'utente clicca sul bottone "tavolo";
 - Viene mostrato la lista dei piatti ordinati del tavolo.

3.2.18 UC3.2 - Visualizza lista ordini personali

- * **Descrizione:** L'utente visualizzato la lista degli ordini personali.
- * **Attore Primario:** Utente generico.
- * **Precondizione:** L'utente si trova dentro la sezione lista ordini.
- * **Postcondizione:** Viene visualizzato la lista degli ordini personali.
- * **Scenrio principale:**
 - L'utente si trova dentro la sezione gestione ordini;
 - L'utente clicca sul bottone "personali";
 - Viene mostrato la lista dei piatti ordinati dall'utente stesso.

3.2.19 UC3.3 - Visualizza lista ordini in arrivo

- * **Descrizione:** L'utente visualizza la lista degli ordini in arrivo.
- * **Attore Primario:** Utente generico.

- * **Precondizione:** L'utente
- * **Postcondizione:** Viene visualizzato la lista lista degli ordini in arrivo.
- * **Scenrio principale:**
 - L'utente si trova dentro la sezione gestione ordini;
 - L'utente clicca sul bottone "in arrivo";
 - Viene mostrato la lista dei piatti in arrivo.

3.2.20 UC3.1.1 - Sposta la lista in arrivo

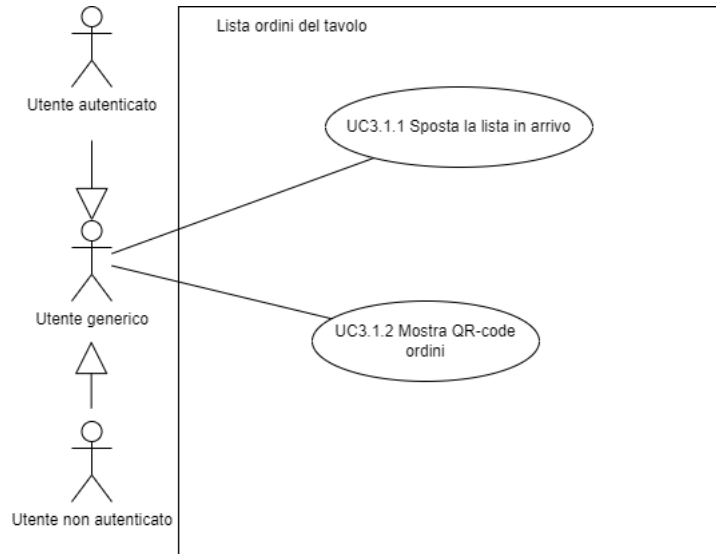


Figura 3.11: Use Case - UC 3.1.1, UC 3.1.2

- * **Descrizione:** L'utente sposta la lista degli ordini in arrivo.
- * **Attore Primario:** Untente generico.
- * **Precondizione:** L'utente si trova dentro la sezione lista ordini del tavolo.
- * **Postcondizione:** Viene spostato la lista degli ordini del tavolo in arrivo.
- * **Scenrio principale:**
 - L'utente si trova dentro la sezione gestione ordini del tavolo;
 - L'utente clicca sul bottone sposta la lista in arrivo;
 - Viene spostato la lista degli piatti ordinati personali in modalità dettaglio;
 - Viene mostrato all'utente il messaggio "ordini spostati correttamente".

3.2.21 UC3.1.2 - Mostra QR-code ordini

- * **Descrizione:** L'utente genera il QR-code della lista ordini per dopo mostrarlo al cameriere.
- * **Attore Primario:** Untente generico.
- * **Precondizione:** L'utente si trova dentro la sezione lista ordini del tavolo.
- * **Postcondizione:** Viene mostrato il QR-code della lista degli ordini.
- * **Scenrio principale:**
 - L'utente si trova dentro la sezione gestione ordini del tavolo;
 - L'utente clicca sul bottone QR-code;

- Viene generato il QR-code degli ordini;
- Viene mostrato il QR-code.

3.2.22 UC3.2.1 - Visualizza in dettaglio lista ordini personali

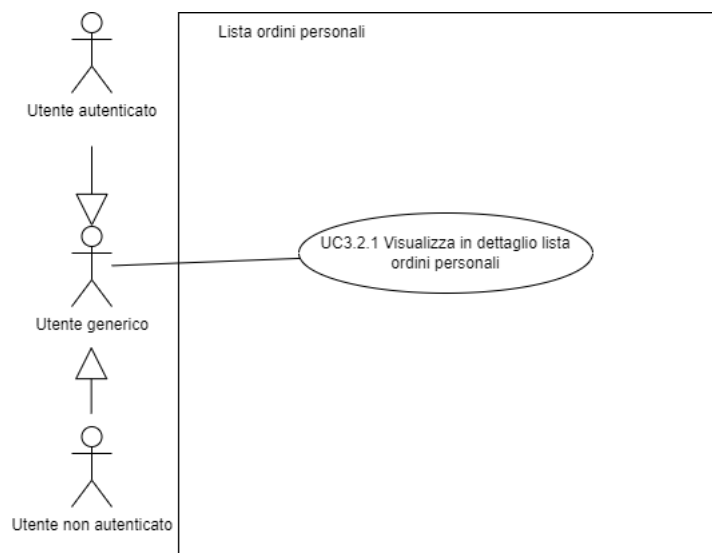


Figura 3.12: Use Case - UC 3.2.1

- * **Descrizione:** L'utente visualizza la lista degli ordini personali in modalità dettaglio.
- * **Attore Primario:** Utente generico.
- * **Precondizione:** L'utente si trova dentro la sezione lista ordini personali.
- * **Postcondizione:** Viene visualizzato la lista degli ordini personali con i piatti in modalità dettaglio.
- * **Scenrio principale:**
 - L'utente si trova dentro la sezione gestione ordini personali;
 - L'utente clicca sul bottone "lente" con il +;
 - Viene mostrato la lista dei piatti ordinati personali in modalità dettaglio.

3.2.23 UC3.3.1 - Ricezione piatto

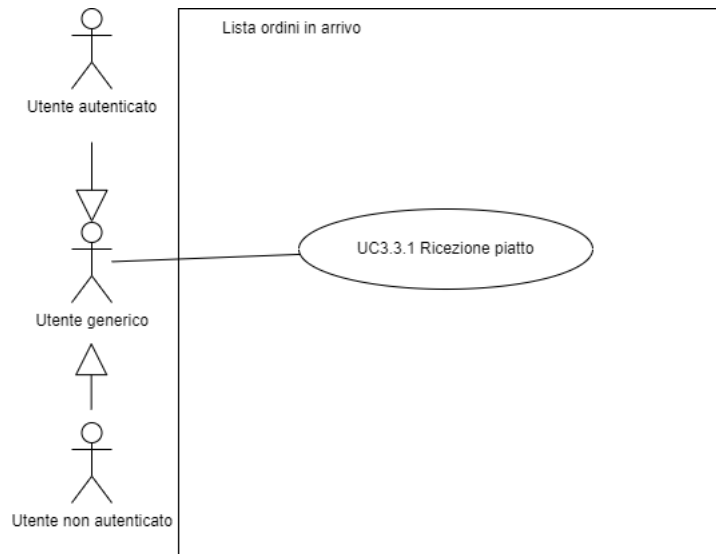


Figura 3.13: Use Case - UC 3.3.1

- * **Descrizione:** L'utente marca un piatto in arrivo come ricevuto.
- * **Attore Primario:** Utente generico.
- * **Precondizione:** L'utente si trova dentro la sezione gestione ordini in arrivo e ha almeno un piatto nella lista in arrivo.
- * **Postcondizione:** L'utente marca il piatto come arrivato diminuendo di 1 la sua quantità.
- * **Scenrio principale:**
 - L'utente si trova dentro la sezione gestione lista ordini in arrivo;
 - L'utente clicca sul bottone "v" di un piatto;
 - Viene diminuito di 1 la sua quantità.
- * **Scenrio alternativo:**
 - L'utente si trova dentro la sezione gestione lista ordini in arrivo;
 - L'utente clicca sul bottone "v" di un piatto con quantità uguale a 1;
 - Viene diminuito di 1 la quantità del piatto e viene disabilitato il bottone.

3.3 Utente Non Autenticato

3.3.1 UC4 - Area personale

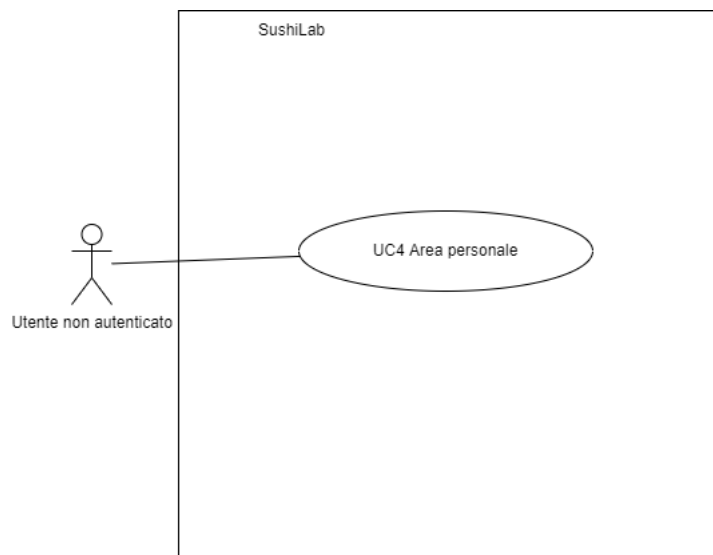


Figura 3.14: Use Case - UC 4

- * **Descrizione:** L'utente visualizza la maschera dell'area personale.
- * **Attore Primario:** Utente non autenticato.
- * **Precondizione:** L'utente si trova dentro la web-app sushiLab.
- * **Postcondizione:** Viene visualizzato la maschera dell'area personale.
- * **Scenrio principale:**
 - L'utente si trova dentro il sistema;
 - Viene mostrato la maschera dell'area personale.

3.3.2 UC4.1 - Registrazione

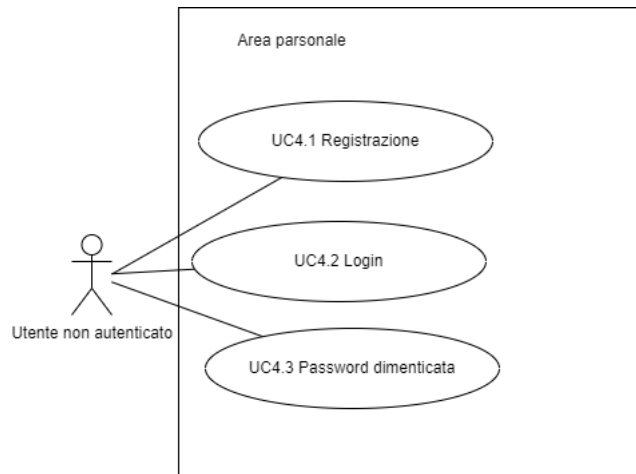


Figura 3.15: Use Case - UC 4.1, UC 4.2, UC 4.3

- * **Descrizione:** L'utente viene registrato nella piattaforma.
- * **Attore Primario:** Utente non autenticato.
- * **Precondizione:** L'utente si trova dentro la web-app sushiLab.
- * **Postcondizione:** Viene salvato i dati dell'utente inseriti durante la fase di registrazione nel data-base.
- * **Scenrio principale:**
 - L'utente si trova dentro l'area personale;
 - L'utente clicca sul bottone registrati;
 - Vine mostrato il form di registrazione;
 - L'utente inserisce l'email;
 - L'utente inserisce la password;
 - L'utente ripete la password;
 - L'utente clicca sul bottone registrati;
 - Viene registrato correttamente l'account.

3.3.3 UC4.2 - Login

- * **Descrizione:** L'utente effettua login nella piattaforma.
- * **Attore Primario:** Utente non autenticato.
- * **Precondizione:** L'utente si trova dentro la web-app sushiLab.
- * **Postcondizione:** Viene effettuato il login.
- * **Scenrio principale:**
 - L'utente si trova dentro l'area personale;
 - L'utente inserisce l'email;
 - L'utente inserisce la password;
 - L'utente clicca sul bottone login;
 - Viene effettuato il login correttamente.

3.3.4 UC4.3 - Password dimenticata

- * **Descrizione:** L'utente reimposta la password del proprio account.
- * **Attore Primario:** Utente non autenticato.
- * **Precondizione:** L'utente si trova dentro la web-app sushiLab.
- * **Postcondizione:** Viene aggiornato la nuova password nel data-base.
- * **Scenrio principale:**
 - L'utente si trova dentro l'area personale;
 - L'utente clicca sul bottone password dimenticata;
 - Vine mostrato il form di recupero password;
 - L'utente inserisce l'email;
 - L'utente clicca sul bottone ottieni codice;
 - L'utente arriva nel secondo form tramite il link mandato tramite email;
 - L'utente inserisce la password;
 - L'utente ripete la password;
 - L'utente clicca sul bottone cambia password;
 - Viene cambiato correttamente la password.

3.4 Utente Autenticato

3.4.1 UC4.4 - Logout

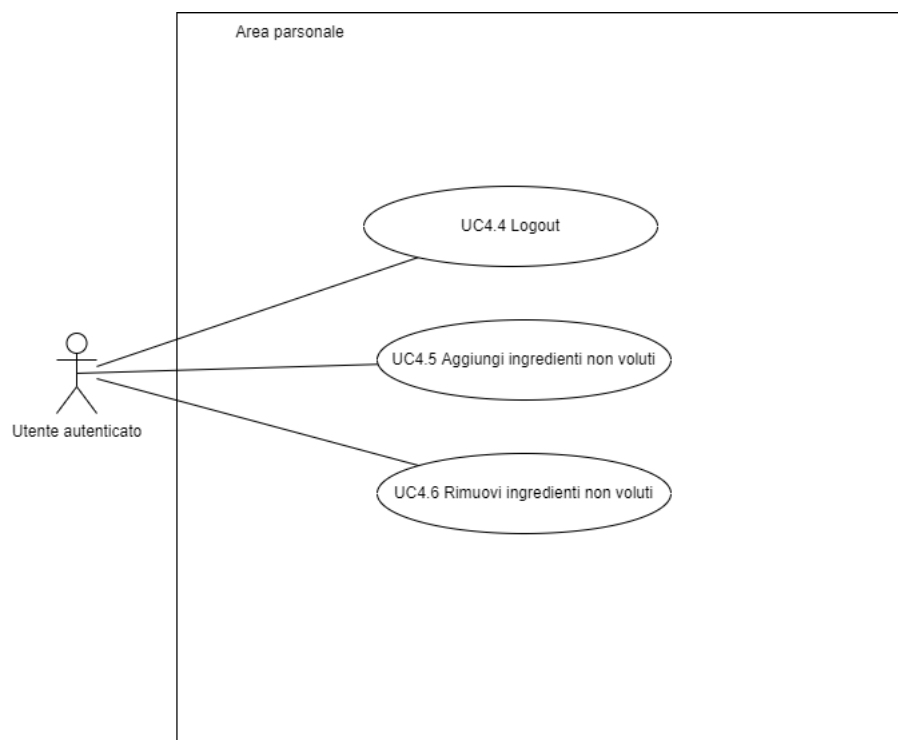


Figura 3.16: Use Case - UC 4.4, UC4.5, UC4.6

- * **Descrizione:** L'utente effettua logout.

- * **Attore Primario:** Utente autenticato.
- * **Precondizione:** L'utente si trova dentro la web-app sushiLab ed ha effettuato la login.
- * **Postcondizione:** Viene effettuato il logout.
- * **Scenrio principale:**
 - L'utente si trova dentro l'area personale;
 - L'utente clicca sul bottone logout;
 - Viene effettuato il logout dell'utente.

3.4.2 UC4.5 - Aggiungi ingredienti non voluti

- * **Descrizione:** L'utente inserisce un ingrediente nella blacklist.
- * **Attore Primario:** Utente autenticato.
- * **Precondizione:** L'utente si trova dentro la web-app sushiLab ha effettuato la login.
- * **Postcondizione:** Viene inserito l'ingrediente nella blacklist.
- * **Scenrio principale:**
 - L'utente si trova dentro l'area personale;
 - L'utente clicca sul bottone blacklist ingredienti;
 - L'utente inserisce il nome del ingrediente;
 - L'utente clicca sul bottone +;
 - Viene inserito ingrediente nella blacklist.

3.4.3 UC4.6 - Rimuovi ingredienti non voluti

- * **Descrizione:** L'utente rimuove un ingrediente dalla blacklist.
- * **Attore Primario:** Utente autenticato.
- * **Precondizione:** L'utente si trova dentro la web-app sushiLab ha effettuato la login.
- * **Postcondizione:** Viene rimosso l'ingrediente dalla blacklist.
- * **Scenrio principale:**
 - L'utente si trova dentro l'area personale;
 - L'utente clicca sul bottone blacklist ingredienti;
 - L'utente clicca sul bottone - di un ingrediente già esistente;
 - Viene rimosso ingrediente dalla blacklist.

3.4.4 UC1.7 - Aggiungi preferiti

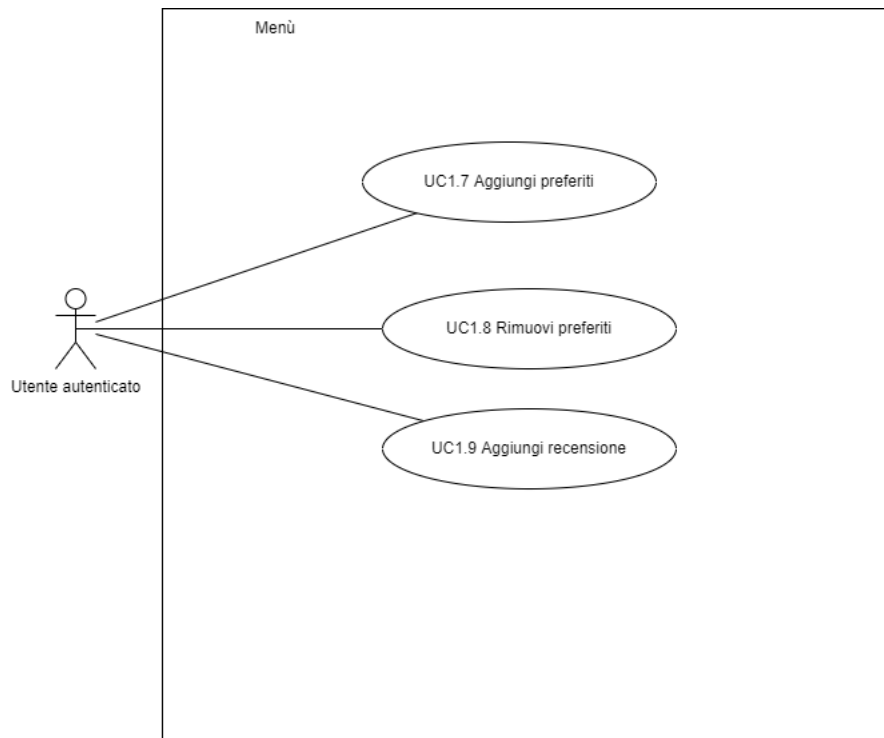


Figura 3.17: Use Case - UC 1.7, UC1.8, UC1.9

- * **Descrizione:** L'utente aggiunge un piatto nella lista dei preferiti.
- * **Attore Primario:** Utente autenticato.
- * **Precondizione:** L'utente si trova dentro nella sezione menù, lista ordini personali o lista ordini in arrivo.
- * **Postcondizione:** Viene inserito il piatto nei preferiti.
- * **Scenrio principale:**
 - L'utente si trova nella sezione menù, lista ordini personali o lista ordini in arrivo;
 - L'utente clicca sul bottone "cuoricino grigio";
 - Viene inserito il piatto nella lista dei preferiti.

3.4.5 UC1.8 - Rimuovi preferiti

- * **Descrizione:** L'utente rimuove un piatto nella lista dei preferiti.
- * **Attore Primario:** Utente autenticato.
- * **Precondizione:** L'utente si trova dentro nella sezione menù, lista ordini personali, lista ordini in arrivo o lista preferiti.
- * **Postcondizione:** Viene rimosso il piatto dalla lista dei preferiti.
- * **Scenrio principale:**
 - L'utente si trova nella sezione menù;
 - L'utente clicca sul bottone "cuoricino rosa";
 - Viene rimosso il piatto nella lista dei preferiti.

3.4.6 UC1.9 - Aggiungi recensione

- * **Descrizione:** L'utente Aggiungi una recensione per un piatto.
- * **Attore Primario:** Utente autenticato.
- * **Precondizione:** L'utente ha selezionato un piatto e il piatto è in modalità dettaglio.
- * **Postcondizione:** Viene aggiornato la recensione del piatto.
- * **Scenrio principale:**
 - L'utente ha selezionato un piatto e il piatto è in modificata dettaglio;
 - L'utente clicca su una delle 5 stelle;
 - Viene inserito la recensione del piatto.

3.4.7 UC5 - Visualizza lista preferiti

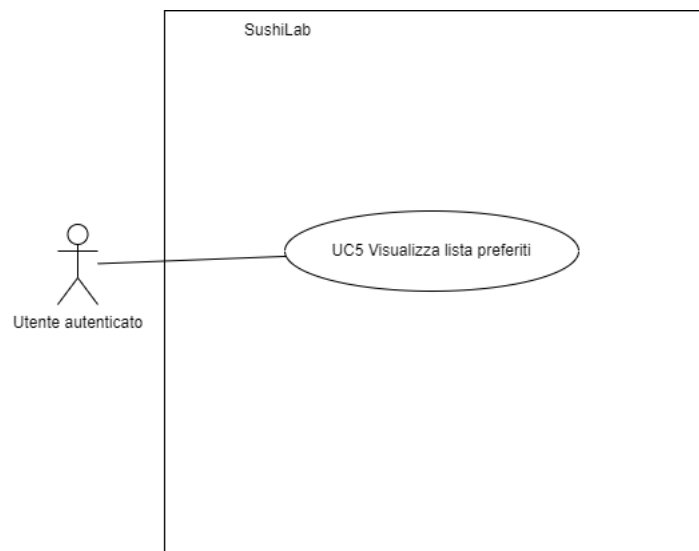


Figura 3.18: Use Case - UC 5

- * **Descrizione:** L'utente visualizza la lista dei preferiti.
- * **Attore Primario:** Utente autenticato.
- * **Precondizione:** L'utente si trova dentro la web-app sushiLab ed ha effettuato la login.
- * **Postcondizione:** Viene mostrato la lista dei preferiti.
- * **Scenrio principale:**
 - L'utente si trova dentro la web-app;
 - L'utente entra nella sezione lista preferiti;
 - Viene mostrato la lista dei preferiti.

3.5 Requisiti

3.5.1 Introduzione

In base a quanto definito nell'analisi dei requisiti sono stati individuati una lista dei requisiti. Nella tabella si trovano tutti requisiti del progetto. Il codice dei requisiti è così strutturato R(F/Q/V)(N/D/O) dove:

R = requisito

F = funzionale

Q = qualitativo

V = di vincolo

N = obbligatorio (necessario)

D = desiderabile

Z = opzionale

3.5.2 Lista dei requisiti

Tabella dei requisiti		
Codice	Requisito	Fonte
ROF1	L'utente può accedere al menù del ristorante	UC1
ROF2	All'utente viene mostrato il menù del ristorante con tutte le categorie	UC1.1
ROF3	All'utente viene mostrato il menù del ristorante con tutti i piatti delle rispettive categorie di appartenenza	UC1.2
ROF4	L'utente può aumentare la quantità di un piatto nel menù	UC1.3
ROF5	L'utente può diminuire la quantità di un piatto nel menù	UC1.4
ROF6	L'utente può impostare la visualizzazione dei piatti del menù nella modalità dettaglio	UC1.5
ROF7	L'utente può impostare la visualizzazione dei piatti del menù nella modalità normale	UC1.6
ROF8	L'utente può accedere alla maschera per la gestione tavolo	UC2
ROF9	L'utente può generare la sessione del tavolo in cui si trova	UC2.1
ROF10	L'utente può unirsi alla sessione di un tavolo	UC2.2
ROF11	L'utente può uscire dalla sessione di un tavolo	UC2.3
ROF12	L'utente può generare il QR-code della sessione del tavolo in cui si trova	UC2.4
ROF13	L'utente può accedere alla maschera per la gestione della lista ordini	UC3
ROF14	All'utente viene mostrato la lista ordini del tavolo	UC3.1
ROF15	All'utente viene mostrato la lista ordini personali	UC3.2
ROF16	All'utente viene mostrato la lista ordini in arrivo	UC3.3

Tabella dei requisiti		
Codice	Requisito	Fonte
ROF17	L'utente può spostare la lista ordini del tavolo in arrivo	UC3.3.1
ROF18	L'utente può generare il QR-code della lista ordini del tavolo in cui si trova	UC3.1.2
ROF19	L'utente può impostare la visualizzazione dei piatti della lista ordini personali in modalità dettaglio	UC3.2.1
ROF20	L'utente può marcare un piatto della lista ordini in arrivo come ricevuto	UC3.3.1
ROF21	L'utente non autenticato può accedere all'area personale	UC4
ROF22	L'utente non autenticato deve riuscire ad inserire email, password e conferma password nel form di registrazione per effettuare la registrazione	UC4.1
ROF23	L'utente non autenticato deve riuscire ad inserire email e password nel form di login per effettuare la login	UC4.2
ROF24	L'utente non autenticato deve riuscire ad inserire la email e la nuova password nel form del password dimenticata	UC4.3
ROF25	L'utente autenticato deve riuscire ad effettuare la logout nell'area personale	UC4.4
ROF26	L'utente autenticato può aggiungere ingredienti non voluti nell'area personale	UC4.4
ROF27	L'utente autenticato può rimuovere ingredienti non voluti nell'area personale	UC4.4
ROF28	L'utente autenticato può aggiungere un piatto nella lista dei preferiti	UC4.4
ROF29	L'utente autenticato può rimuovere un piatto dalla lista dei preferiti	UC4.4
ROF30	L'utente autenticato può aggiungere una recensione per un piatto	UC1.7
ROF31	L'utente autenticato può visualizzare la propria lista dei preferiti	UC5
ROV1	L'interfaccia utente del sistema dovrà essere sviluppato sfruttando il framework Angular	SyncLab
ROV2	Lo stile dell'interfaccia utente del sistema dovrà essere sviluppato sfruttando CSS-3	SyncLab
ROV3	Le chiamate API devono essere implementate tramite Stoplight	SyncLab
ROV4	È necessario dividere le varie maschere in componenti diversi di Angular	SyncLab

Tabella dei requisiti		
Codice	Requisito	Fonte
ROV5	La web-app dovrà funzionare sul browser Microsoft Edge dalla versione più recente	SyncLab
ROV6	La web-app dovrà funzionare sul browser Google Chrome dalla versione più recente	SyncLab
ROV7	La web-app dovrà funzionare sul browser Firefox dalla versione più recente	SyncLab
ROV8	La web-app dovrà funzionare sul browser Safari dalla versione più recente	SyncLab
RDV9	Il codice sorgente dovrà essere commentata	SyncLab
ROQ1	Il codice sorgente della piattaforma sarà reperibile su GitHub	SyncLab
ROQ2	Fornire una sezione tutorial che spieghi come si utilizza la web-app	SyncLab

Capitolo 4

Progettazione e codifica

In questo capitolo vengono trattati la progettazione e codifica della parte front-end della web-app. Vengono elencati tutti i componenti della web-app e vengono descritti la loro funzionalità.

4.1 Progettazione

4.1.1 Architettura Angular

Un'applicazione Angular è formato da un insieme di moduli, dove il modulo principale è il modulo root, chiamato AppModule, che contiene più moduli di funzionalità. Un modulo di funzionalità è composto da un componente, che devinisce la vista dell'utente. Ogni componente possiede un template di HTML, dove viene definita il modello di vista, combinata con le direttive di Angular consente di modificare il codice HTML, una volta cambiata la struttura della pagina viene fatto il rendering e infine viene cambiata la vista.

I componenti utilizzano dei servizi che forniscono funzionalità specifiche come la login, ma non sono correlate direttamente alla vista, essi sono inseriti come delle dipendenze e grazie a questo rende il codice efficiente. Non solo i servizi sono riutilizzabili ma anche i componenti lo sono, quindi un componente può contenere più componenti così rende l'applicazione Angular più semplice da comprendere e manutenibile in futuro.

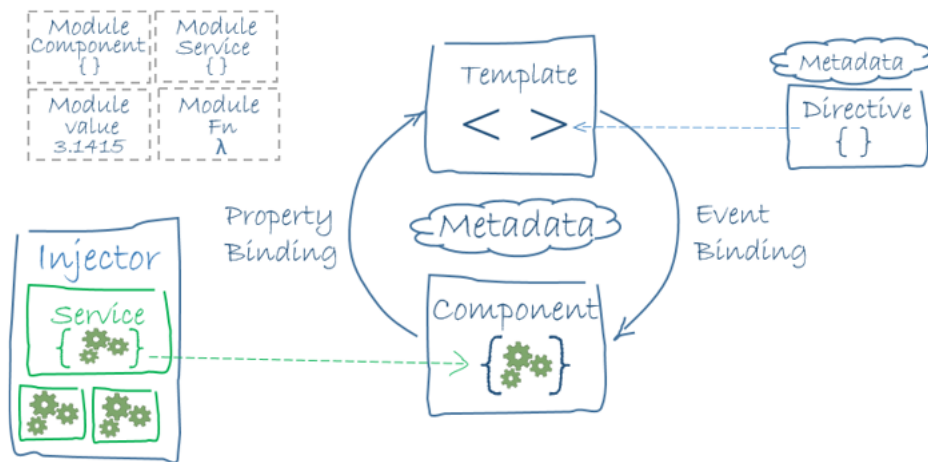


Figura 4.1: Architettura Angular

4.1.2 Architettura SushiLab

La web-app segue l'architettura spiegata precedentemente che è anche quello consigliato dal sito ufficiale di Angular.

La cartella principale della web-app è app che contiene tutti i componenti, i servizi e il root.

Per ogni componente si è creato una cartella per essa, in cui contiene il suoi file .ts per la logica, .html per la struttura, .scss per il layout di grafica e infine i suoi componenti figli. Per i componenti condivisi si è creato una cartella shared dove vengono salvati i componenti che sono utilizzati in più parti dell'applicazione

Nella cartella rest vengono salvati tutti i file service, in cui ci sono dei metodi che sono chiamati in più parti dell'applicazione al fine di massimizzare il riutilizzo del codice.

Nella cartella assets vengono salvati le immagini e le icone utilizzate, in modo da fare utilizzare da tutti i componenti.

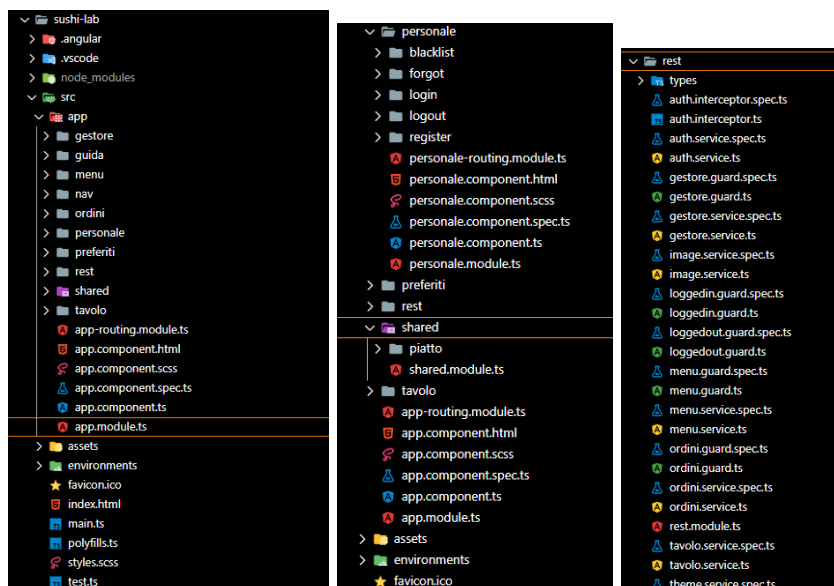


Figura 4.2: Struttura file SushiLab

4.1.3 Progettazione API

La API sono implementate tramite la piattaforma Stoplight che è uno dei requisiti obbligatori richiesti dall'azienda.

La progettazione delle API sono molto semplici grazie all'interfaccia semplice di Stoplight, che permette di definire i path e i rispettivi metodi facilmente.

Per ogni chiamata bisogna avere:

- * **Nome:** individua le API;
- * **Descrizione:** spiega in dettaglio la funzione della API;
- * **Metodo:** definisce il tipo di chiamata, è stato usato:
 - **get:** per richiedere dei dati al server come la chiamata per ottenere il menù;
 - **post:** per inviare dei dati sintetici al server come i dati per login;
 - **delete:** per eliminare dei dati dal server come eliminazione della sessione di tavolo.
- * **Path:** definisce il percorso finale della API;
- * **Risposta:** configura la risposta che ritorna la API, è stato usato:
 - **200:** richiesta andata a buon fine;
 - **201:** creazione andata a buon fine;
 - **204:** richiesta andata a buon fine ma il contenuto è vuoto;
 - **401:** non autorizzato;
 - **404:** non trovato;
 - **406:** non accettato;
 - **500:** errore interno;

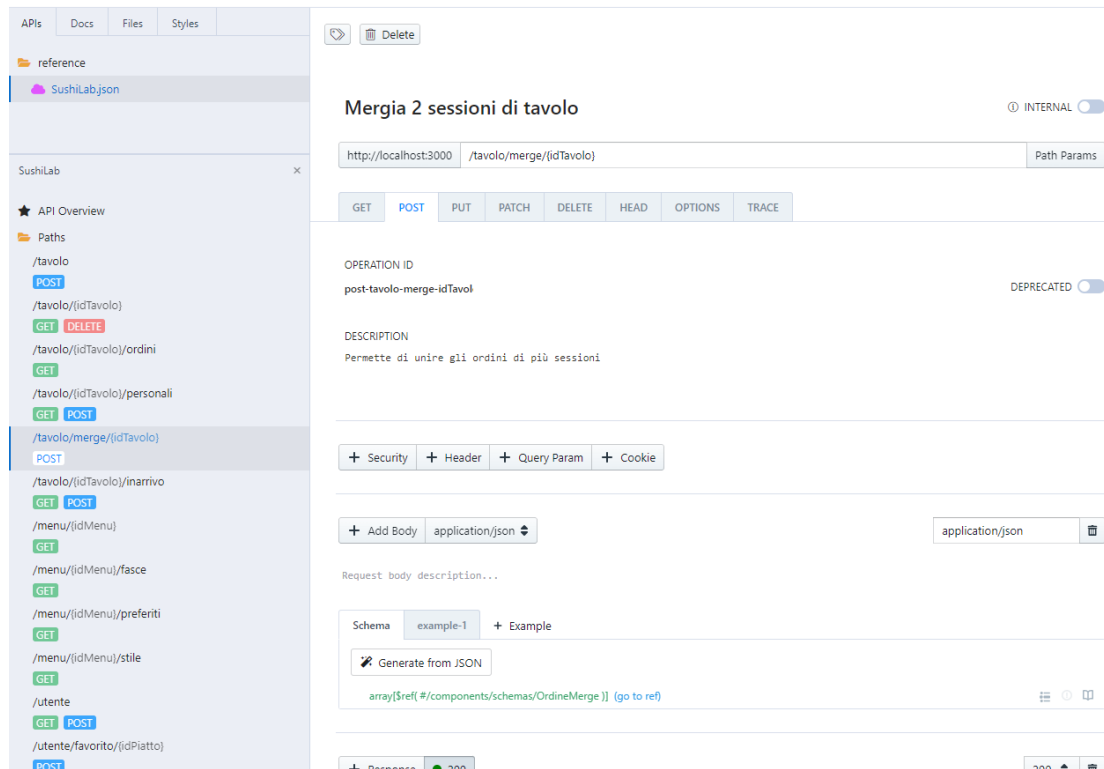


Figura 4.3: Stoplight SushiLab

4.1.4 Progettazione delle viste

All'inizio si è fatto un meeting con il tutor aziendale per chiarire le funzionalità che deve avere la web-app, dopo di che si è iniziato la progettazione delle viste tramite la editor di grafica online Figma.

Tramite il sistema di progettazione di Figma sono stati creati le bozze delle viste per chiarire i collegamenti tra di loro e i posizionamenti dei componenti. Il posizionamento dei componenti sono scelti in base al loro numero di volte che viene premuto e si è anche basato alle web-app più popolari. È stato deciso i seguenti aspetti:

- * I colori principali e lo sfondo della applicazione;
- * Il bottone per la navbar è in alto a destra;
- * Il logo della piattaforma in alto a sinistra;
- * I bottoni, testi e form devono avere lo stesso stile e colore in base alla loro funzionalità;
- * Lo stile del piatto in modalità dettaglio in menù e nella lista degli ordini personali è la stessa;
- * Tutti le maschere hanno una visuale che utilizza la card.



Figura 4.4: Figma SushiLab

4.2 Codifica

4.2.1 Interfacce

Menù

Viene mostrata il menù del ristorante che comprende tutte le categorie e le loro piatti.

Funzionalità:

- * L'utente può aumentare la quantità di un piatto per aggiungerlo o aumentare la sua quantità nell'ordine del tavolo;
- * L'utente può diminuire la quantità di un piatto per rimuoverlo o diminuire la sua quantità nell'ordine del tavolo;
- * L'utente può visualizzare il piatto in modalità dettaglio per dare una recensione al piatto o inserire una nota per il piatto;
- * L'utente può aggiungere il piatto nella lista dei preferiti o rimuoverlo dalla lista.

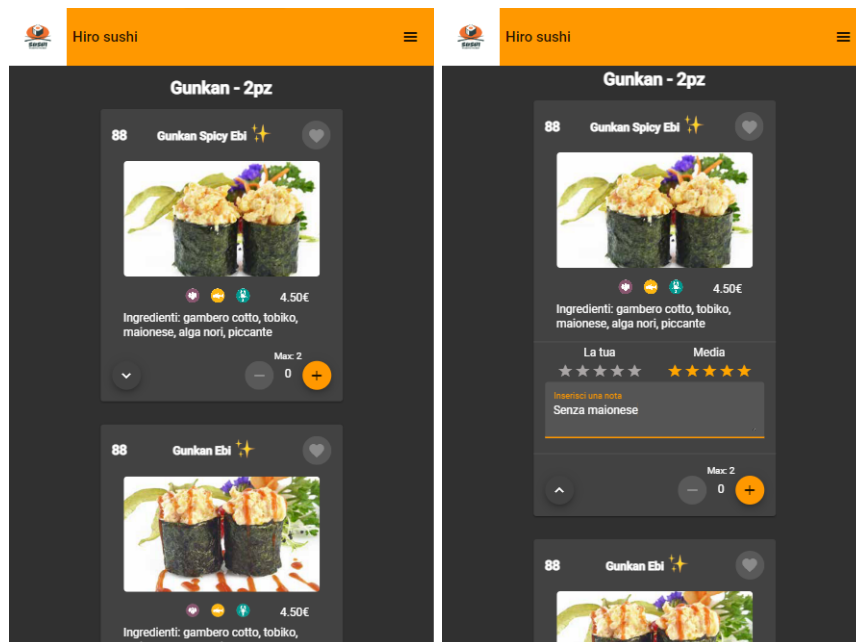


Figura 4.5: Menù e Menù con piatto in dettaglio di SushiLab

Gestione tavolo

Viene mostrata la maschera di gestione tavolo.

Funzionalità:

- * L'utente può creare una sessione di tavolo;
- * L'utente può andare al form per unire ad una sessione.

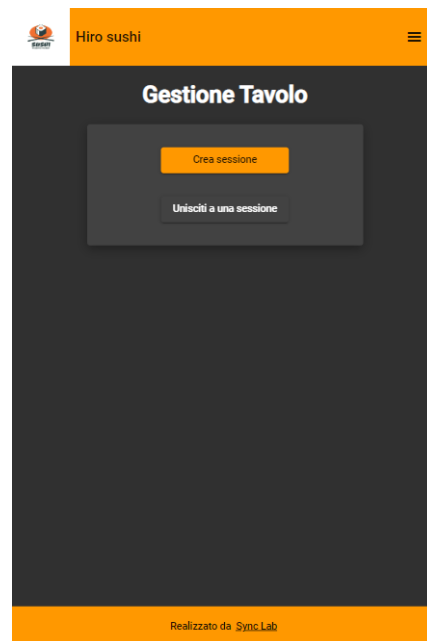


Figura 4.6: Maschera gestione tavolo SushiLab

Unisci sessione

Interfaccia tramite la quale è possibile unire ad una sessione di tavolo.

Funzionalità:

- * L'utente può unirsi ad una sessione inserendo il codice di sessione se il codice non rispetta la validazione del form non si abilita il bottone unisci;
- * L'utente può tornare nella pagina gestione tavolo.

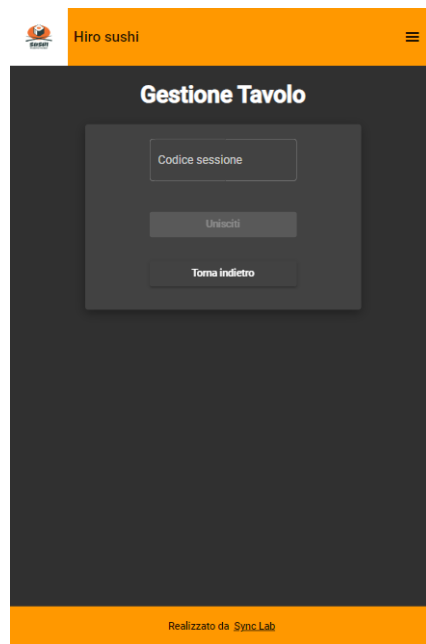


Figura 4.7: Maschera unisci sessione SushiLab

QR-code tavolo

Viene mostrata il QR-code del tavolo che permette agli altri utenti che si trovano sullo stesso tavolo dell'utente di unirsi.

Funzionalità:

- * L'utente può uscire dalla sessione premendo il bottone esci dalla sessione.

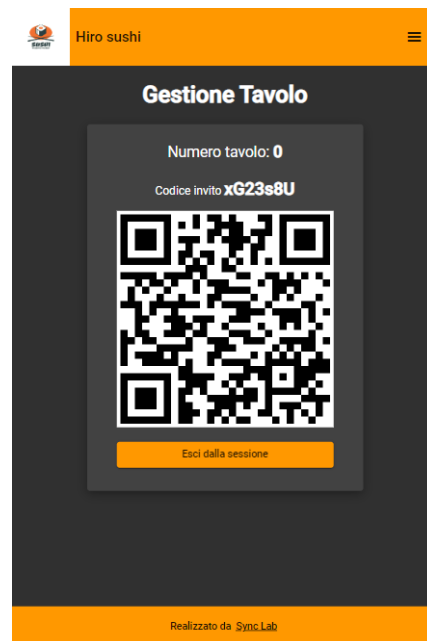


Figura 4.8: Maschera QR-code tavolo SushiLab

Lista ordini del tavolo

Viene mostrato la lista degli ordini della sessione del tavolo in cui si trova l'utente.

Funzionalità:

- * L'utente può spostare la lista in arrivo;
- * L'utente può generare il QR-code premendo sul bottone QR-code;
- * L'utente può andare in altre sezioni della lista ordini utilizzando la navbar interna.

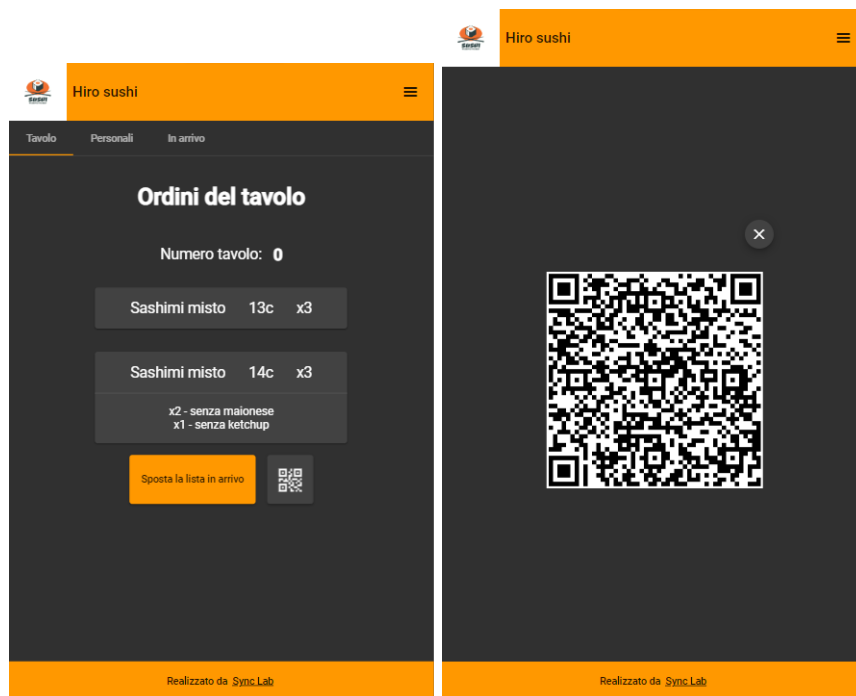


Figura 4.9: Maschera lista ordini del tavolo e QR-code ordini SushiLab

Lista ordini personali

Viene mostrata la lista degli ordini personali.

Funzionalità:

- * L'utente può visualizzare i piatti in modalità dettaglio;
- * L'utente può visualizzare i piatti in modalità normale;
- * L'utente può aggiungere un piatto nella lista dei preferiti o rimuoverlo dalla lista;
- * L'utente può dare una recensione ad un piatto.

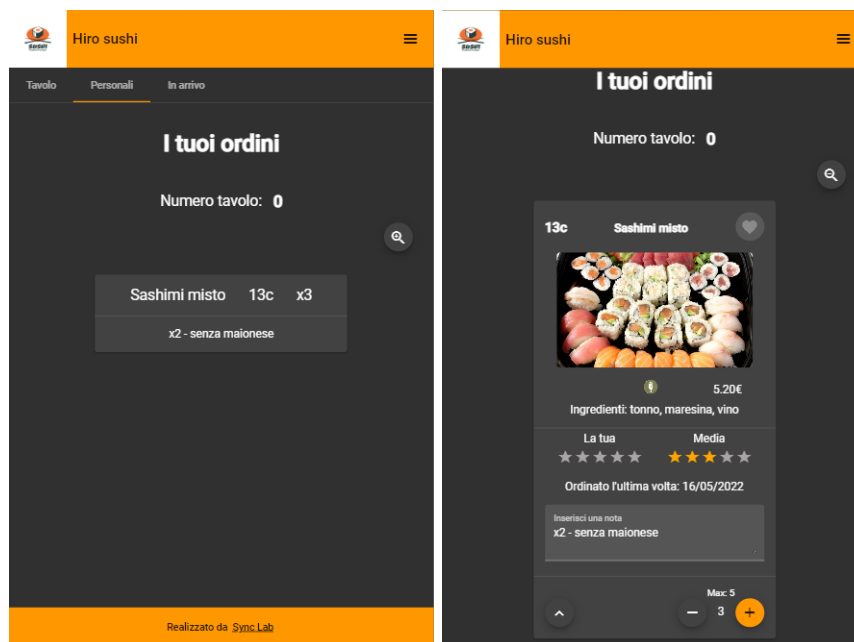


Figura 4.10: Maschera ordini personali e ordini personali con piatti in dettaglio SushiLab

Lista ordini in arrivo

Viene mostrata la lista degli ordini in arrivo.

Funzionalità:

- * L'utente può visualizzare i piatti in modalità dettaglio;
- * L'utente può visualizzare i piatti in modalità normale;
- * L'utente può marcare un piatto come arrivato;
- * L'utente può aggiungere un piatto nella lista dei preferiti o rimuoverlo dalla lista;
- * L'utente può dare una recensione ad un piatto.

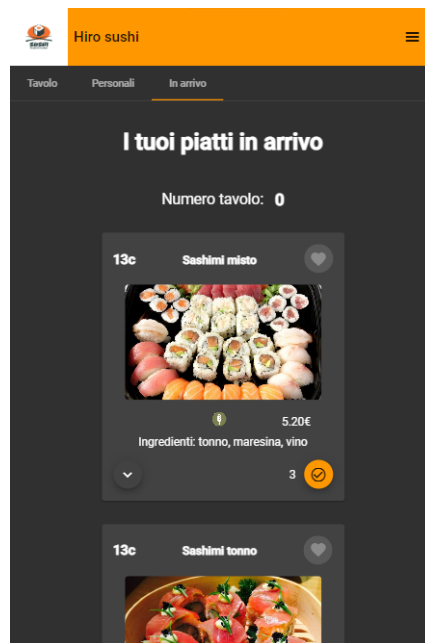


Figura 4.11: Maschera ordini in arrivo SushiLab

Login

Viene mostrato il form per effettuare il login dove è possibile effettuare la login nella piattaforma.

Funzionalità:

- * L'utente può effettuare il login inserendo i campi correttamente altrimenti viene evidenziato in rosso i campi sbagliati;
- * L'utente può andare al form di registrazione;
- * L'utente può andare al form di password dimenticata.

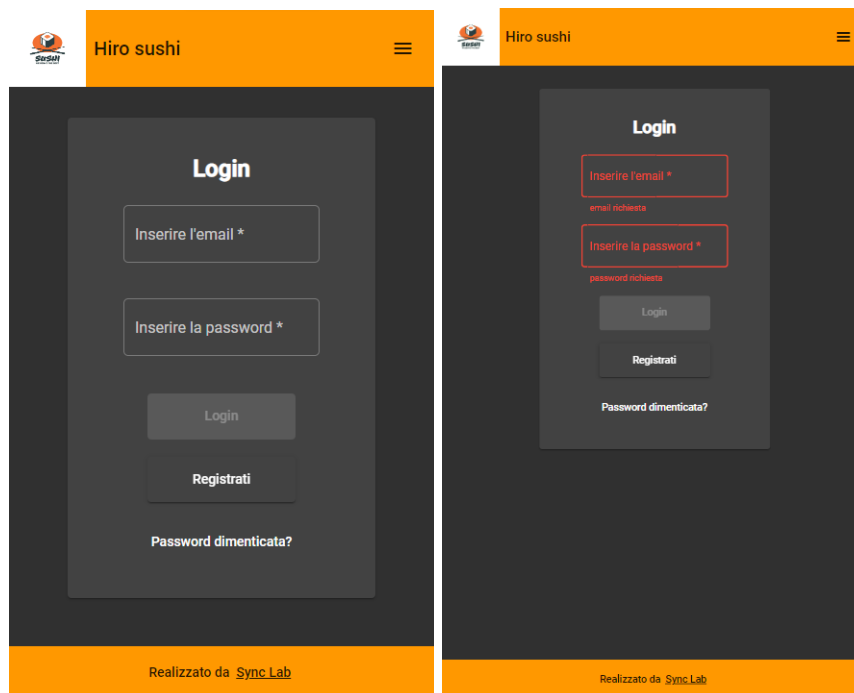


Figura 4.12: Maschera login e login con campi errati SushiLab

Registrazione

Viene mostrato il form di registrazione dove è possibile registrare nella piattaforma.

Funzionalità:

- * L'utente può effettuare la registrazione inserendo i campi correttamente altrimenti viene evidenziato in rosso i campi sbagliati;
- * L'utente può andare al form di login.

The figure displays two versions of the registration form for 'Hiro sushi'. Both versions have an orange header with the logo and a menu icon, and an orange footer with the text 'Realizzato da Sync Lab'. The form itself is a dark gray box with the title 'Registrati'. It contains three input fields: 'Inserire l'email *', 'Inserire la password *', and 'Ripetere la password *'. Below these fields are a 'Registrati' button and a 'Già registrato? Login' link. In the left screenshot, all fields are correctly filled. In the right screenshot, each field has a red border and a red error message below it: 'email richiesta', 'password richiesta', and 'ripetizione richiesta'.

Figura 4.13: Maschera registrazione e registrazione con campi errati SushiLab

Password Dimenticata

Viene mostrato il form di cambia password dove è possibile reimpostare la password.

Funzionalità:

- * L'utente può reimpostare la password inserendo i campi correttamente altrimenti viene evidenziato in rosso i campi sbagliati;
- * L'utente può andare al form di registrazione;
- * L'utente può andare al form di Login.

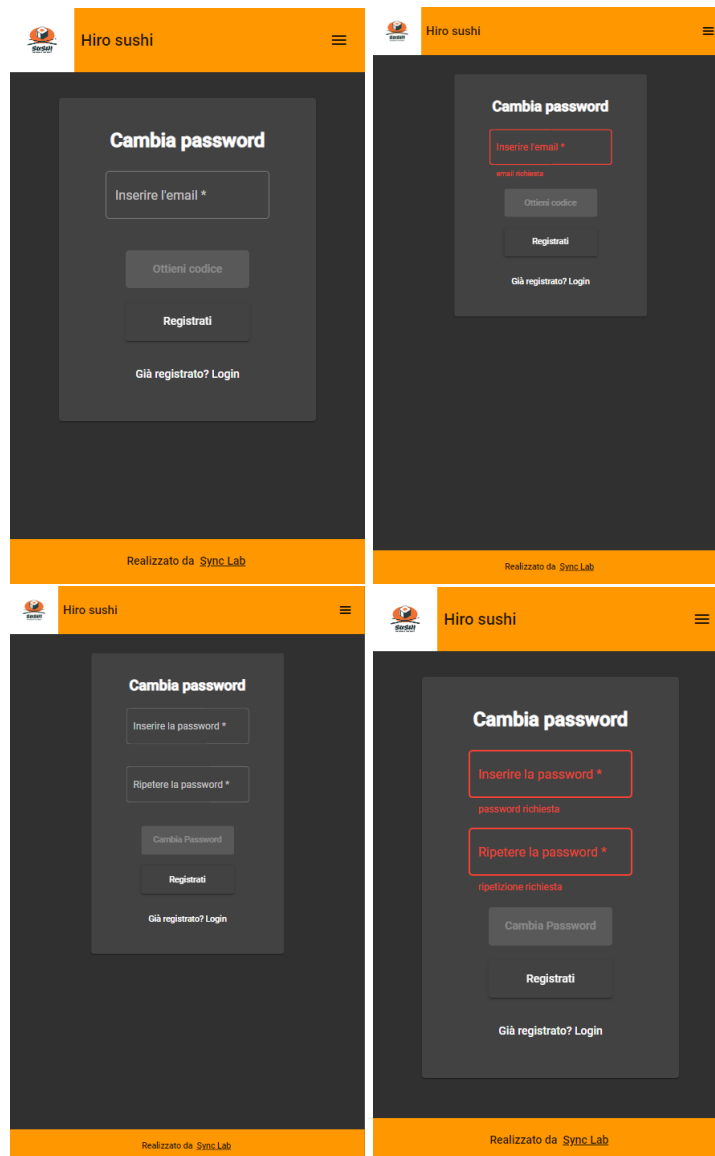


Figura 4.14: Maschere cambia password e cambia password con i campi errati SushiLab

Area Personale

Viene mostrata l'area personale dell'utente dove si può vedere i dati dell'utente.

Funzionalità:

- * L'utente può effettuare il logout;
- * L'utente può andare nella sezione di blacklist ingredienti per inserire o rimuovere delle allergeni o preferenze.

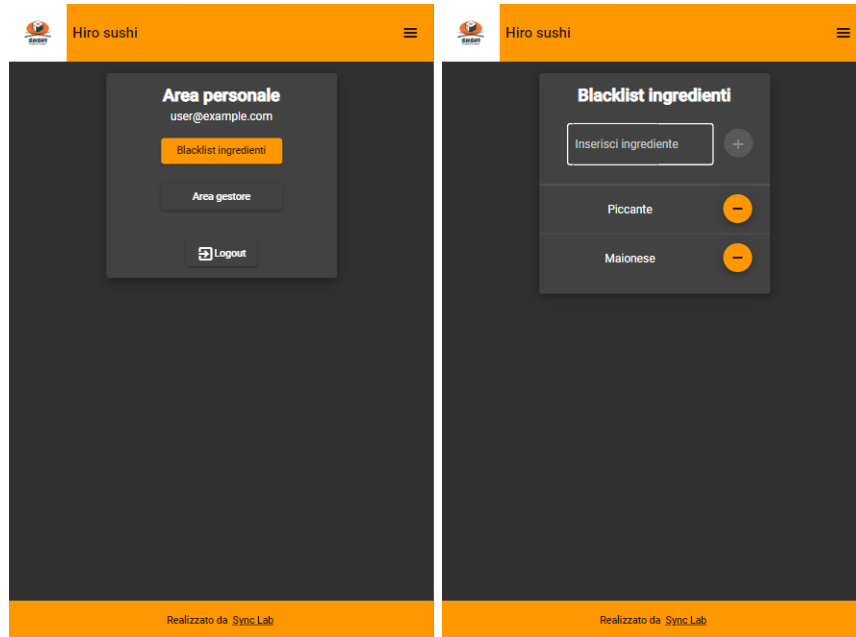


Figura 4.15: Maschera area personae e blacklist SushiLab

Menu component

Componente usato per creare la maschera del menù. Composto da un insieme di piatti creati automaticamente in base all'array menù di ritorno.

Componenti e servizi usati:

- * `piatto.component`;
- * `menu.service`;
- * `ordini.service`.

Nav component

Componente usato per creare la maschera della navbar. Composto da un insieme di link che porta l'utente in varie sezioni della web-app.

Metodi:

- * `isLoggedIn()`: ritorna true se l'utente è loggato, false altrimenti;
- * `haveMenu()`: ritorna true se è presente il menù nel local storage, false altrimenti;
- * `haveTable()`: ritorna true se è presente la sessione di tavole nel local storage, false altrimenti;
- * `close()`: chiude il menu a tendina.

Componenti e servizi usati:

- * auth.service;
- * menu.service;
- * table.service.

Ordini component

Componente usato per creare la maschera della lista degli ordini. Composto da tre sezioni che sono ordini del tavolo, ordini personali e ordini in arrivo.

Metodi:

- * moveOrders(): sposta la lista degli ordini del tavolo in arrivo;
- * changeZoom(): cambia la modalità di presentazione dei piatti, cambiando da modificata normale in modificata dettaglio o viceversa;
- * showQRCode(): mostra il QR-code della lista degli ordini;
- * closeQRCode(): chiude la maschera di QR-code;
- * getAllOrders(): inizializza tutti gli ordini predendo i piatti dal server.

Componenti e servizi usati:

- * piatto-arrivo.component;
- * piatto.component;
- * piatto-ordine.component;
- * auth.service;
- * menu.service;
- * table.service.

Piatto-arrivo component

Componente usato per creare il piatto della sezione lista degli ordini in arrivo.

Metodi:

- * toggleFav(): cambia lo stato di preferito di un determinato piatto;
- * updateStar(): aggiorna la recensione di un determinato piatto;
- * decreaseCount(): chiamato quando utente marca un piatto come arrivato, diminuisce la quantità di un piatto in arrivo.

Componenti e servizi usati:

- * auth.service;
- * menu.service.

Piatto-ordine component

Componente usato per creare il piatto in modalità normale per la sezione ordini dove viene mostrato solamente il nome, numero, quantità e la nota.

Personale component

Componente usato per creare la maschera dell'area personale, dove si può vedere email dell'utente e andare nella sezione blacklist.

Metodi:

- * isLoggedIn(): ritorna true se l'utente non è loggato, false altrimenti.

Componenti e servizi usati:

- * blacklist.component;
- * forgot.component;

- * login.component;
- * logout.component;
- * register.component;
- * auth.service;
- * user.service.

blacklist component

Componente usato per creare gestire la sezione blacklist degli ingredienti, viene mostrato la blacklist ed è possibile rimuovere o aggiungere un ingrediente.

Metodi:

- * onAdd(): metodo chiamato quando si aggiunge un ingrediente nella blacklist;
- * onRemove(): metodo chiamato quando si rimuove un ingrediente dalla blacklist;
- * isValid(): ritorna true se ingrediente inserito è valido, false altrimenti.

Componenti e servizi usati:

- * user.service.

Forgot component

Componente usato per creare la maschera per reimpostare la password.

Metodi:

- * sendCode(): chiama il server per mandare il link dove porta l'utente alla pagina per reimpostare la password;
- * cambia(): metodo chiamato quando l'utente conferma il form, viene controllato i dati inseriti e in fine viene chiamato API per cambiare la password.

Componenti e servizi usati:

- * user.service.

Login component

Componente usato per creare la il form per effettuare la login.

Metodi:

- * login(): metodo chiamato quando l'utente conferma il form, viene controllato vari campi e infine viene chiamato API per effettuare la login.

Componenti e servizi usati:

- * auth.service.

Logout component

Componente usato per creare il bottone di logout e serve per effettuare il logout.

Metodi:

- * logout(): effettua il logout dell'utente.

Componenti e servizi usati:

- * auth.service.

Register component

Componente usato per creare il form per la registrazione.

Metodi:

- * register(): metodo chiamato quando l'utente conferma il form, viene controllato vari campi e infine viene chiamato API per la registrazione.

Componenti e servizi usati:

- * user.service.

Preferiti component

Componente usato per creare la maschera dei preferiti dove si può visualizzare la lista dei preferiti e rimuovere un piatto non più interessante.

Componenti e servizi usati:

- * piatto.component;
- * menu.service.

Piatto component

Componente usato per creare il piatto, viene utilizzato per la visualizzazione del menù e della lista dei preferiti.

Componenti e servizi usati:

- * menu.service.

Preferiti component

Componente usato per creare la maschera dei preferiti dove si può visualizzare la lista dei preferiti e rimuovere un piatto non più interessante.

Metodi:

- * toggleExpand(): espande la visuale del piatto mostrando tutti i dettagli del piatto o nasconde i suoi dettagli;
- * toggleFav(): cambia lo stato di preferito di un determinato piatto;
- * updateStar(): aggiorna la recensione di un determinato piatto;
- * increaseCount(): aumenta la quantità di un determinato piatto presente negli ordini personali;
- * decreaseCount(): diminuisce la quantità di un determinato piatto presente negli ordini personali;
- * noteChanged(): cambia la nota di un piatto presente negli ordini personali.

Componenti e servizi usati:

- * star.component;
- * ordini.service;
- * auth.service;
- * menu.service.

Tavolo component

Componente usato per creare la maschera gestione tavolo, dove si può creare una sessione di tavolo, unire ad una sessione o uscire da una sessione.

Metodi:

- * changeState(): cambia lo stato di visualizzazione della sezione gestione tavolo in base al table.service se è presente o no una sessione;
- * removeSession(): rimuove la sessione del tavolo;
- * setSession(): imposta la sessione del tavolo chiamando il server e infine imposta il numero di tavolo;
- * updateURL(): aggiorna il percorso del link per generare il QR-code del tavolo;
- * createSession(): crea la sessione di tavolo.

Componenti e servizi usati:

- * table.service;
- * menu.service.

auth.service

Servizio utilizzato per gestire le autenticazioni. **Metodi:**

- * login(email: string, password: string): metodo chiamato per effettuare la login dell'utente;
- * setSession(): salva i dati dell'utente dopo la login nel local storage per mantenere la sessione di login;
- * logout(): metodo chiamato per effettuare il logout dell'utente;
- * isLoggedIn(): ritorna true se è presente nel local storage la sessione dell'utente, false altrimenti;
- * getExpirations(): ritorna true se la sessione dell'utente non è ancora scaduto, false altrimenti.

menu.service

Servizio utilizzato per gestire tutte le funzionalità del menù. **Metodi:**

- * title(): ritorna il nome del menù;
- * haveMenu(): ritorna true se è presente un menù nel local storage, false altrimenti;
- * getMenuId(): ritorna l'id del menù;
- * setMenuId(id: number): salva l'id del menù nel local storage;
- * applyBlacklist(menu: Menu): rimuove i piatti che contengono gli ingredienti presenti nella blacklist dell'utente;
- * setFavorite(idPiatto:number, favorite:boolean): aggiunge un piatto del menù nella lista dei preferiti o lo toglie dalla lista dei preferiti;
- * setRating(idPiatto: number, rate: number): salva la recensione del piatto;
- * getPreferiti(): richiede al server la lista dei preferiti dell'utente.

ordini.service

Servizio utilizzato per gestire le ordinazioni. **Metodi:**

- * updateRemoteOrder(): aggiorna gli ordini del server mandandogli la lista degli ordini ogni lasso di tempo;
- * getOrdered(idPiatto:number): ritorna indice nel array di ordini del piatto passato;
- * clearOrder(): svuota array di ordini presente nel local storage;
- * setOrder(idPiatto: number, count: number, note: string): imposta la quantità del piatto con id idPiatto passato presente nella lista degli ordini con count;
- * mergeOrder(newOrders: any[]): unisce gli ordini locali con quelli del server quando l'utente si unisce ad una sessione passando da una sessione a una sessione diversa;
- * getSessionCode(): ritorna il codice sessione del tavolo;
- * getOrdersTable(): ritorna l'ordine del tavolo;
- * getOrdersUser(): ritorna l'ordine dell'utente;
- * getOrdersArriving(): ritorna l'ordine in arrivo;
- * moveOrders(): sposta l'ordine del tavolo in arrivo;
- * toQRCode(): ritorna la stringa per generare il qr code per la lista dell'ordine.

tavolo.service

Servizio utilizzato per gestire le sessioni di tavolo. **Metodi:**

- * `createSession()`: chiama il server per creare una sessione di tavolo;
- * `getSession()`: ritorna il numero del tavolo;
- * `getCode()`: ritorna il codice di sessione;
- * `haveSession()`: ritorna true se esiste un codice di sessione false altrimenti;
- * `setSession(code: string)`: imposta il codice di sessione con il code passato come parametro per unirsi alla sessione del tavolo;
- * `menuChanged()`: controlla se il menu della sessione corrente è uguale a quello della sessione a cui si vuole unirsi se è false non è possibile fare il merge dei piatti;
- * `removeSession()`: rimuove i dati della sessione di tavolo.

user.service

Servizio utilizzato per gestire le funzionalità dell'area personale. **Metodi:**

- * `pushBlacklist()`: chiama il server aggiornare la blacklist degli ingredienti;
- * `getUtente()`: richiede al server per ottenere le informazione dell'utente attualmente collegato;
- * `setUtente(user: User)`: manda al server i dati dell'utente per completare la registrazione;
- * `verifyCode(code: string)`: verifica se il link con cui è passato al form per reimpostare la password è corretto o no;
- * `getChangeCod(email: string)`: chiede al server per mandare un'email per al email passato per reimpostare la password;
- * `changePassword(password: string)`: manda al server la password nuova per aggiornare quella vecchia.

Capitolo 5

Verifica e validazione

In questo

5.1 Verifica

Viene riportato una tabella dei test di unità sui vari componenti e sulle loro funzionalità, per ogni test viene identificato il codice del test, il requisito, una descrizione e lo stato.

Tabella dei requisiti			
Test	Requisito	Descrizione	Stato
T1	ROF1	Il componente menù viene creato	Passato
T2	ROF3	Il componente piatto viene creato	Passato
T3	ROF4	La quantità viene aumentata	Passato
T4	ROF5	La quantità viene diminuita	Passato
T5	ROF6	Il componente piatto viene visualizzato in modalità dettaglio	Passato
T6	ROF8	Il componente gestione tavolo viene creato	Passato
T7	ROF9	La sessione viene generata	Passato
T8	ROF10	L'utente viene unito ad una sessione	Passato
T9	ROF11	L'utente esce dalla sessione	Passato
T10	ROF12	Il QR-code del tavolo viene generato	Passato
T11	ROF13	Il componente gestione lista ordini viene creato	Passato
T12	ROF17	La lista ordini del tavolo viene spostato in arrivo	Passato
T13	ROF18	Il QR-code della lista ordini viene generato	Passato
T14	ROF20	Il piatto delle lista ordini in arrivo viene marcato come arrivato	Passato

Tabella dei requisiti			
Test	Requisito	Descrizione	Stato
T15	ROF21	Il componente area personale viene creato	Passato
T16	ROF22	Il componente register viene creato	Passato
T17	ROF22	I campi sbagliati del form di registrazione vengono evidenziati	Passato
T18	ROF22	Il submit del form deve registrare l'utente	Passato
T19	ROF23	Il componente login viene creato	Passato
T20	ROF23	I campi sbagliati del form di login vengono evidenziati	Passato
T21	ROF23	Il submit del form deve effettuare il login dell'utente	Passato
T22	ROF24	Il componente forgot viene creato	Passato
T23	ROF24	I campi sbagliati del form di password dimenticata vengono evidenziati	Passato
T24	ROF24	Il submit del form deve reimpostare la password	Passato
T25	ROF25	Il componente logout viene creato	Passato
T26	ROF25	Il click sul componente logout deve effettuare il logout dell'utente	Passato
T27	ROF26	Il componente blacklist viene creato	Passato
T28	ROF26	L'ingrediente viene inserito nella blacklist	Passato
T29	ROF27	L'ingrediente viene rimosso dalla blacklist	Passato
T30	ROF28	Il piatto viene aggiunto nella lista dei preferiti	Passato
T31	ROF29	Il piatto viene rimosso dalla lista dei preferiti	Passato
T32	ROF30	Il componente star viene creato	Passato
T34	ROF30	La recensione viene impostato correttamente	Passato
T35	ROF31	Il componente lista preferiti viene creato	Passato

5.2 Validazione e collaudo

Nell'ultima settimana dello stage ho controllato tutte le funzionalità implementate durante intero progetto, dove ho testato tutte le attività che sono svolgibili da un utente generale. Infine è stato mostrato il prodotto finale del progetto sushiLab al tutor, in cui viene fatto il demo completo.

Capitolo 6

Conclusioni

6.1 Raggiungimento degli obiettivi

6.1.1 Obiettivi fissati

Tutti gli obiettivi fissati all'inizio dello con il tutor aziendale sono stati raggiunti, oltre ai requisiti obbligatori sono stati soddisfatti anche i requisiti desiderabili e i facoltativi. Infatti la web-app fonisce tutte le funzionalità richieste e offre inoltre la possibilità di salvare i piatti nella lista dei preferiti, filtrare il menù rimuovendo i piatti contenenti gli ingredienti presenti nella blacklist e fornire una recensione ad un piatto.

Obiettivi fissati		
Codice	Descrizione	Esito
O01	Acquisizione competenze sulle tematiche sopra descritte	Superato
O02	Capacità di raggiungere gli obiettivi richiesti in autonomia seguendo il cronoprogramma	Superato
O03	Portare a termine le implementazioni previste con una percentuale di superamento pari al 80%	Superato
D01	Portare a termine le implementazioni previste con una percentuale di superamento pari al 100%	Superato
F01	Apportare un valore aggiunto al gruppo di lavoro durante le fasi di progettazione delle interfacce	Superato

Tabella 6.1: Tabella raggiungimento degli obiettivi fissati.

6.1.2 svolgimento del lavoro

I lavori pianificati per le prime settimane di stage sono state svolte senza particolari difficoltà, infatti le attività di studio e formazione sono terminate con un piccolo anticipo rispetto alla pianificazione e il tempo risparmiato viene utilizzato alla fine per implementare le funzionalità aggiuntive. Questo guadagno di tempo è dovuto ad una buona familiarità con i linguaggi di programmazione acquisita durante gli otti anni di studio nell'ambito di informatica.

Avendo lavorato nei ristoranti sushi con il ruolo di cameriere, quindi sapendo di preciso i processi che ci sono, la fase di progettazione ed implementazione delle maschere sono anche loro andati con un pò di anticipo.

La parte che ha impiegato più ore di quelli previsti è stato la fase di integrazione tra le varie maschere dove bisogna mantenere i dati salvati di ogni maschera, questo problema è stato risolto grazie ai componenti service forniti da Angular. Un'altro strumento che ha dato un grande aiuto è discord dove ci ha permesso di comunicare facilmente con altri collaboratori del progetto.

6.2 Conoscenze acquisite

Durante lo stage nell'azienda SyncLab, ho acquisito molte nuove tecnologie più utilizzati nel mondo.

Ho imparato ad usare Angular che è uno dei più famosi framework per lo sviluppo front-end per le applicazioni, progettare il front-end è sempre stato il mio ambito più amato. Apprendimento di Angular è tutto svolto in autonomia, per fare questo ho dovuto consultare vari siti web e video di tutorial su YouTube, così mi ha fatto crescere la mia capacità di imparare nuovi skills da solo.

Per l'implementazione delle maschere il css è uno dei linguaggi più importanti, Angular ha una buona integrazione con la sintassi SCSS, infatti, ho dovuto imparare ad utilizzare questa nuova estensione.

La progettazione delle API tramite Stoplight mi ha fatto apprendere spring, che sarà il back-end della web-app quindi per avere una buona integrazione tra i due componenti bisogna avere una buona conoscenza di spring.

Lavorando con Angular mi ha dato la possibilità di imparare bene TypeScript, utilizzando TypeScript mi ha fatto notare i problemi che ha JavaScript, tramite TypeScript tutti gli errori sintattici vengono rilevati prima della compilazione perciò mi ha risparmiato molto tempo per il debugging.

Ogni settimana viene fatto un incontro in sede con il tutor, dove si discute l'avanzamento della web-app, se ci sono stati delle difficoltà e viene ripianificato il lavoro della settimana seguente in caso di cambiamenti, svolgendo in questo modo lo stage mi ha fatto familiarizzare con il metodo Agile.

Infine il tirocinio mi è stato molto fondamentale per conoscere il mondo del lavoro dell'informatica, difatti SyncLab è uno delle aziende molto importante nell'ambito IT. Non solo questo, lo stage mi ha anche fatto vedere come si è cambiato il lavoro dopo la pandemia covid-19, tanti lavori adesso possono svolti in remoto e tutto questo è possibile grazie alle nuove applicazioni sviluppate da programmatori, ciò mi ha fatto capire importanza dell'informatica.

Tutto questo mi ha consolidato le mie competenze di programmazione comprese durante gli anni di studio e mi ha dato tanti notevoli insegnamenti.

6.3 Valutazione personale

Lo stage mi ha ampiamente soddisfatto. Mi ha fatto conoscere meglio i miei limiti ma soprattutto il reale lavoro di programmatore, in effetti durante le 320 ore di tirocinio dove si lavora come un vero e proprio dipendente 8 ore al giorno, mi ha messo in prova dove posso usare tutte le competenze acquisite durante il corso di studio universitario. Ho visto intero processo di sviluppo di un software. Si è partito da un problema reale che si trova nei ristoranti sushi, si è analizzato tutti i requisiti che servono per risolverlo infine si è arrivato alla progettazione e implementazione del programma.

Durante il processo di sviluppo, dove si è svolto il progetto con il tutor e un altro studente, mi ha rafforzato le capacità di lavoro in gruppo, in particolare nell'aspetto organizzativo come ad esempio il `ordini.service`, che è utilizzato in più maschere dell'applicazione, ho dovuto iniziarlo prima, così facendo il mio compagno lo poteva utilizzare e testare il suo corretto funzionamento.

Arrivando alla fine, intera esperienza dello stage è più che positiva, ritengo che il tirocinio sia molto importante per uno studente, permettendolo di sfruttare e valutare le proprie conoscenze acquisite e particolarmente per capire e riflettere il corso studio scelto se è quello che desiderato o meno, infatti lo stage mi ha affermato di continuare la carriera lavorativa nell'ambito di programmazione.

Bibliografia

Riferimenti bibliografici

James P. Womack, Daniel T. Jones. *Lean Thinking, Second Editon*. Simon & Schuster, Inc., 2010.

Siti web consultati

Manifesto Agile. URL: <http://agilemanifesto.org/iso/it/>.