

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO
II
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA ELETTRICA E
TECNOLOGIE DELL'INFORMAZIONE
"Ratatouille23"

Francesco Orlando
N86003618
francesco.orlando3@studenti.unina.it

Marco Pastore
N86003875
marco.pastore6@studenti.unina.it

Indice

1 Documento dei Requisiti Software	2
1.1 Analisi dei Requisiti	2
1.1.1 Modellazione di tutti i casi d'uso richiesti	2
1.1.2 Individuazione del target degli utenti	7
1.1.3 Descrizioni Testuali Strutturate per casi d'uso significativi	10
1.1.4 Prototipazione visuale via Mock-up dell'interfaccia utente per tutti i casi d'uso assegnati	16
1.1.5 Valutazione dell'usabilità a priori	28
1.1.6 Glossario	33
1.2 Specifica dei Requisiti	35
1.2.1 Classi, oggetti e relazioni di analisi	35
1.2.2 Diagrammi di sequenza di analisi	52
1.2.3 Prototipazione funzionale via statechart dell'interfaccia grafica	54
2 Documento di Design del Sistema	56
2.1 Analisi dell'architettura con esplicita definizione dei criteri di design	56
2.1.1 Architettura esterna	56
2.1.2 Architettura Server	56
2.1.3 Architettura Client	57
2.1.4 Servizi Cloud	58
2.2 Descrizione/motivazione delle scelte tecnologiche adottate	58
2.2.1 Perché android?	58
2.2.2 Perché AWS?	58
2.3 Diagramma delle classi di design	59
2.4 Diagrammi di sequenza di design per i casi d'uso	75
3 Testing e valutazione sul campo dell'usabilità	77
3.0.1 Descrizione delle Strategie Adottate per la Progettazione dei Test	78
3.0.2 Analisi	81
3.0.3 Analytics	83

Capitolo 1

Documento dei Requisiti

Software

1.1 Analisi dei Requisiti

L'analisi dei requisiti è una delle fasi più importanti nello sviluppo del software. Consiste nell'identificare e definire i requisiti del sistema software che si intende sviluppare. I requisiti sono le specifiche funzionali e non funzionali che il sistema deve soddisfare per essere considerato adeguato e utile per gli utenti.

L'analisi dei requisiti è un processo continuo e iterativo, che coinvolge gli stakeholders del progetto. Questo processo aiuta a garantire che il sistema software soddisfi le esigenze degli utenti, delle organizzazioni e delle parti interessate.

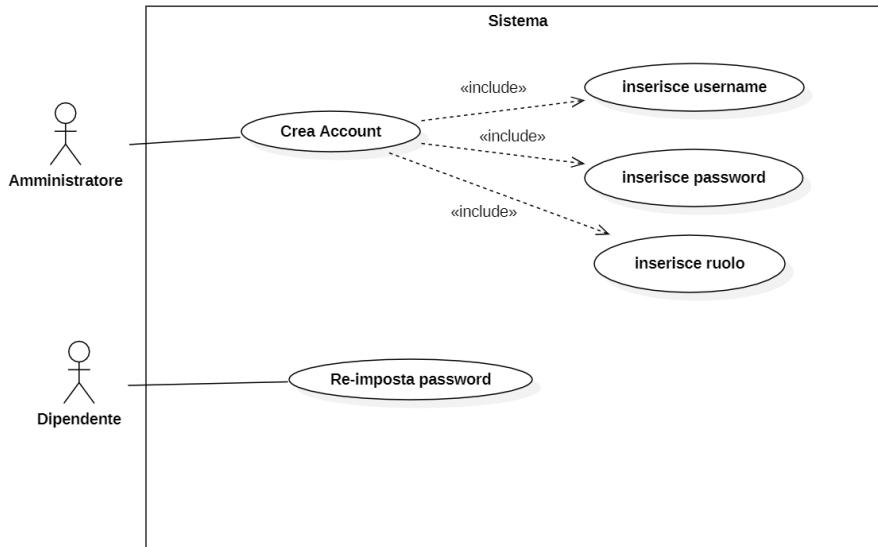
La fase di analisi dei requisiti prevede la raccolta e la documentazione dei requisiti, l'identificazione dei vincoli del sistema e la definizione dei casi d'uso. L'obiettivo è di fornire una base solida per la progettazione, lo sviluppo, la verifica e la validazione del sistema software.

1.1.1 Modelloazione di tutti i casi d'uso richiesti

Dopo un attento studio e una particolare analisi, abbiamo riassunto tutti i casi d'uso in questi 6 Use Case Diagram:

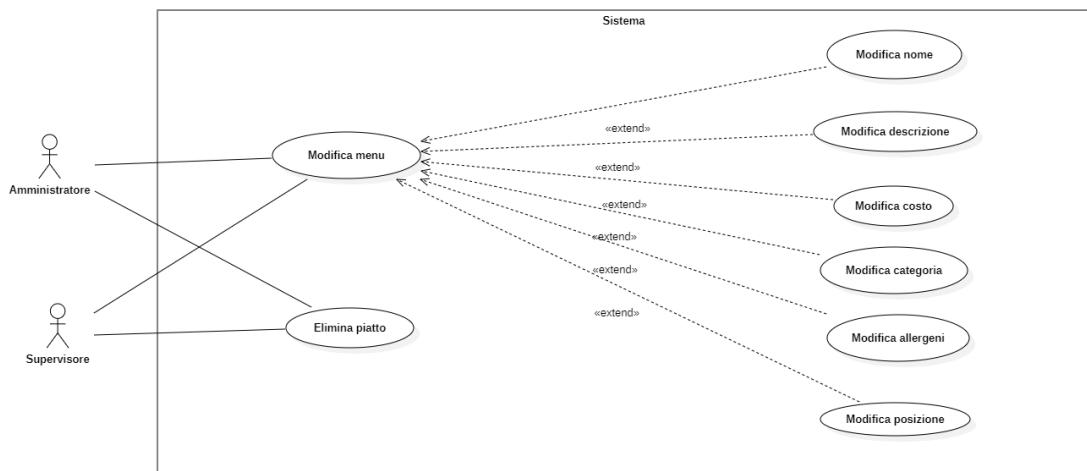
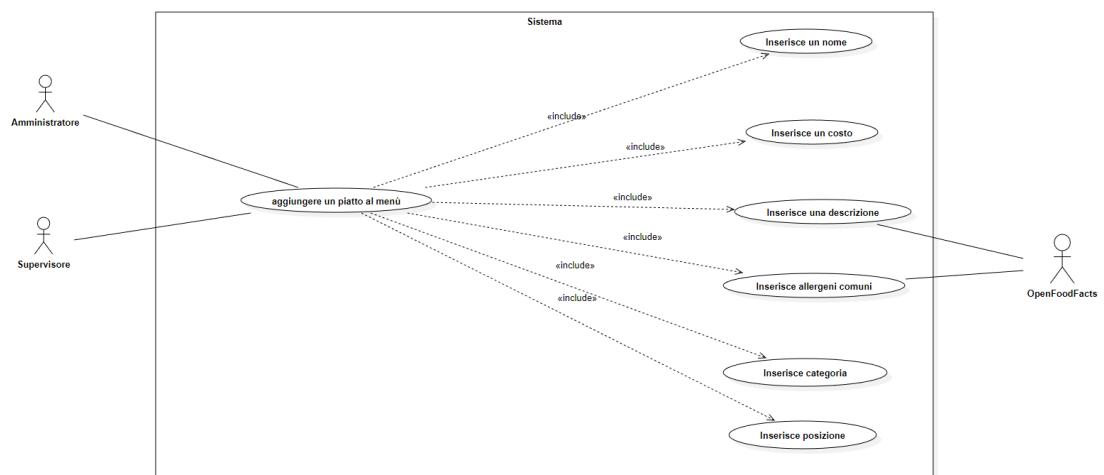
Punto 1: Creazione Utenze

Un amministratore può creare utenze per i propri dipendenti (sia addetti alla sala, che addetti alla cucina, che supervisori). Al primo accesso, ogni utente deve re-impostare la password inserita dall'amministratore, scegliendo una password diversa.



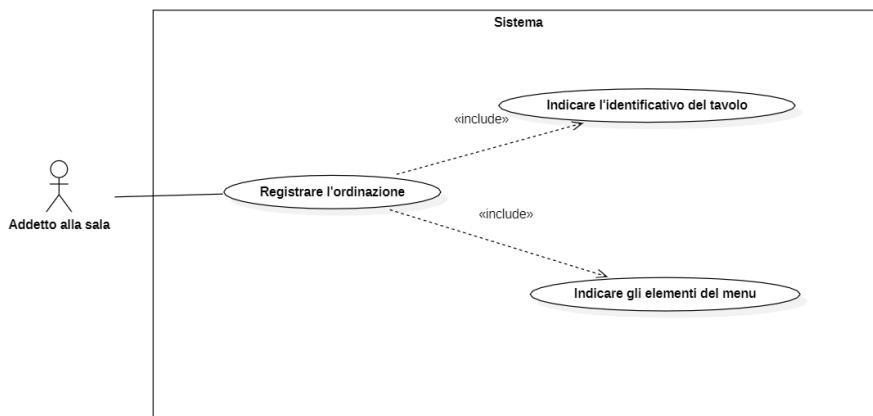
Punto 3: Personalizzazione Menù

Un amministratore (o un supervisore) può personalizzare il menù delle attività di ristorazione. Ciascun elemento è caratterizzato da un nome, un costo, una descrizione, e un elenco di allergeni comuni. Inoltre è possibile organizzare gli elementi del menù in categorie personalizzabili e definire l'ordine con cui gli elementi compaiono nel menù. In fase di inserimento, è possibile sfruttare l'autocompletamento della descrizione e degli allergeni in base al nome del piatto inserito.



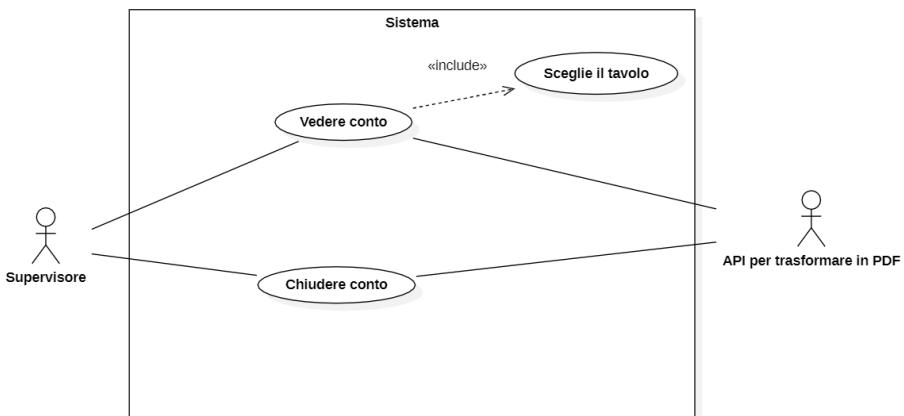
Punto 6: Registrazione Ordine

Un addetto alla sala può registrare ordinazioni indicando l'identificativo del tavolo e gli elementi del menù desiderati.



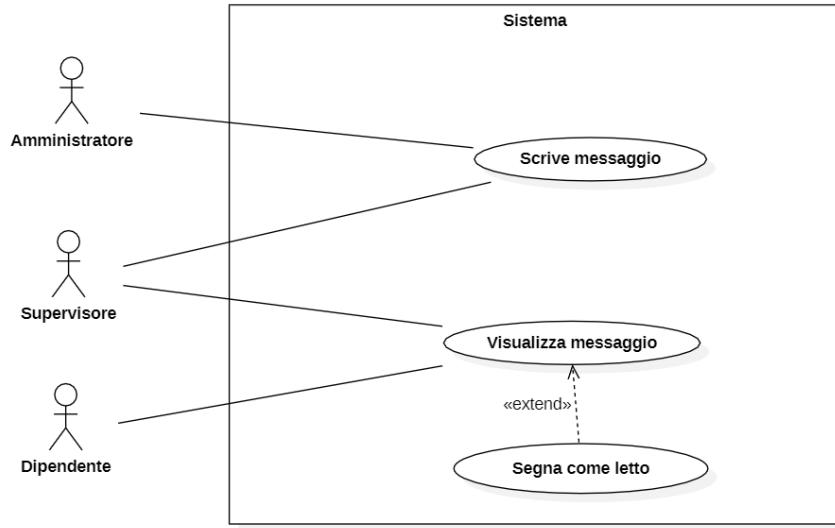
Punto 8: Visualizza/Scarica PDF

Un supervisore può visualizzare (in formato PDF) il conto di ciascun tavolo in un determinato momento. Inoltre, quando gli avventori seduti ad un certo tavolo vanno via, il supervisore può chiudere il conto e visualizzare la versione finale da sottoporre al cliente.



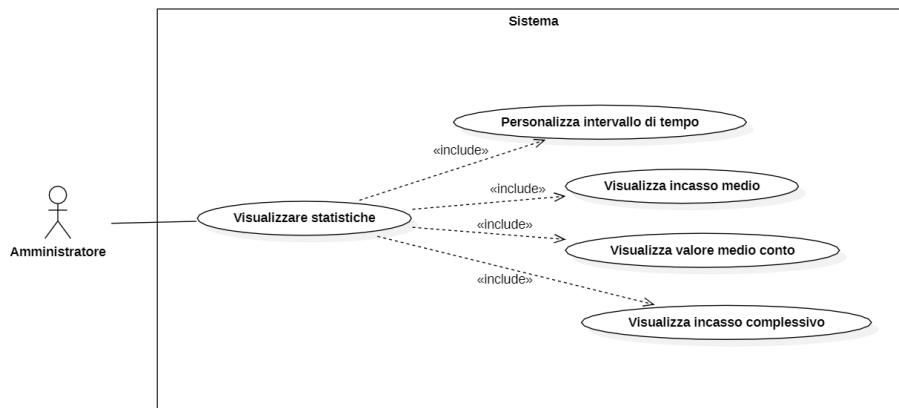
Punto 13: Avvisi

Un supervisore o un amministratore può inserire nel sistema degli avvisi, che vengono visualizzati da tutti i dipendenti. Ciascun dipendente può marcare un avviso come "visualizzato" e nasconderlo.



Punto 17: Statistiche

Un amministratore può visualizzare statistiche sugli introiti dell'attività di ristorazione. In particolare, dato un certo intervallo di tempo personalizzabile, è possibile visualizzare l'incasso medio, il valore medio di ciascun conto, l'incasso complessivo. È inoltre possibile visualizzare le statistiche tramite un grafico interattivo.



1.1.2 Individuazione del target degli utenti

Statistiche sugli utenti del sistema:

Conoscere il range di età delle varie entità coinvolte nell'uso del software è molto importante per capire chi è abituato a interfacciarsi con più o meno facilità nell'uso di applicazioni. Dopo aver consultato varie fonti, siamo arrivati alle seguenti conclusioni:

- 47% dei camerieri ha un età compresa tra i 18 e i 35 anni.
- Il 72.5% dei datori di lavoro non fa distinzioni tra uomini e donne.
- Al 36% dei nuovi assunti viene richiesto solo un diploma di scuola dell'obbligo, il 23.8% viene chiesto un diploma di scuola secondaria di secondo grado. Al 76% viene chiesto un titolo che attesti la formazione professionale.

E' anche molto importante conoscere in base alla grandezza dell'azienda, le interazioni con i camerieri:

- Le piccole aziende da 1 a 9 dipendenti sono le realtà con cui si confronteranno maggiormente (49.4%) i camerieri, il 40.7% le medie aziende e il 9.9% le grandi.

Fonte: Excelsior Unioncamere.

Personas:

Le "Personas" sono rappresentazioni semi-fittizie di persone reali che rappresentano i diversi tipi di utenti che potrebbero utilizzare un prodotto, servizio o sistema. Le Personas sono basate su ricerche di mercato, analisi dei dati degli utenti e altre fonti di informazioni per creare profili dettagliati dei comportamenti, delle esigenze, dei desideri e delle aspettative degli utenti. Questi profili aiutano i progettisti a comprendere meglio le esigenze degli utenti e a creare prodotti o servizi che soddisfino le loro necessità in modo più efficace e mirato. Le Personas possono anche aiutare a definire la strategia di marketing e comunicazione per il prodotto o servizio.



Dario Franzese

Età: 55

Professione:
Cuoco



Alessia Ascolese

Età: 52

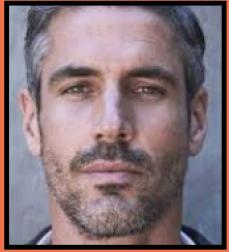
Professione:
Cameriere



Gian Marco Addati

Età: 45

Professione:
Supervisore



Luigi Penza

Età: 60

Professione:
Amministratore

1.1.3 Descrizioni Testuali Strutturate per casi d'uso significativi

Visualizza statistiche

Caratteristiche	Descrizione
Use case visualizza statistiche	Visualizzazione delle statistiche di vendita
Goal in Context	L'amministratore vuole visualizzare le statistiche di vendita del ristorante
Preconditions	L'amministratore si trova nell'interfaccia di visualizzazione statistiche
Success End Conditions	Le statistiche vengono visualizzate con successo
Failed End Conditions	Non è stato possibile visualizzare le statistiche
Primary Actor	L'amministratore
Trigger	Clicca sul bottone "STATISTICHE"

Description	Step	Amministratore	System
	1	Click sul bottone "STATISTICHE"	
	2		Mostra schermata delle statistiche
	3	Seleziona il range desiderato	
	4	Clicca il bottone "CALCOLA"	
	5		Mostra le medie e l'incasso complessivo

Extension	Step	Amministratore	System
Omissione di uno o più campi	4.1	Clicca il bottone "CALCOLA"	
	5.1		Errore: "Non hai compilato uno o più campi!"
Ritorna allo step 3			

Extension	Step	Amministratore	System
Il campo dell'inizio del range è maggiore rispetto a quello finale	4.2	Clicca il bottone "CAL-COLA"	
	5.2		Errore: "Il range è sbagliato!"
Ritorna allo step 3			

Extension	Step	Amministratore	System
Preme il bottone "LogOut"	3.3	Preme il bottone "LogOut"	
	4.3		Torna nella schermata di LogIn

Extension	Step	Amministratore	System
Preme il bottone "Aggiungi piatto"	3.4	Preme il bottone "Aggiungi piatto"	
	4.4		Va nella schermata "Aggiungi piatto"

Estensione	Step	Amministratore	System
Modifica/Elimina cibo	3.5	Preme il bottone "Modifica elimina cibo"	
	4.5		Va nella schermata "Modifica elimina cibo"

Estensione	Step	Amministratore	System
Preme il bottone "Messaggi"	3.6	Preme il bottone "Messaggi"	
	4.6		Va nella schermata "Messaggi"

Estensione	Step	Amministratore	System
Preme il bottone "Crea utente"	3.7	Preme il bottone "Crea utente"	
	4.7		Va nella schermata "Crea utente"

Visualizza conto

Caratteristiche	Descrizione
Use case visualizza statistiche	Visualizzazione e chiusura del conto di un tavolo in PDF
Goal in Context	Un supervisore vuole visualizzare il conto di un tavolo oppure vuole chiudere un conto
Preconditions	Il supervisore ha selezionato dal menù la voce "TAVOLI"
Success End Conditions	La visualizzazione del conto è avvenuta con successo o chiude il conto correttamente
Failed End Conditions	E' avvenuto un errore durante la visualizzazione del conto o durante la chiusura del conto
Primary Actor	Un supervisore
Trigger	Clicca sul bottone "STATISTICHE"

Description	Step	Amministratore	System
	1	Preme il bottone "Conti"	
	2		Mostra i tavoli e il conto del primo tavolo
	3	Clicca sul tavolo che vuole visualizzare	
	4	Mostra il conto del tavolo selezionato	
	5		Sceglie di chiudere il conto
	6		Chiude il conto
	7		Aggiorna il database
	8		Scarica il PDF

Subvariation	Step	Supervisore	System
Vuole scaricare il pdf	5.1	Sceglie di scaricare il pdf del conto per quel tavolo	
	6.1		Scarica il PDF"
Ritorna allo step 3			

Extension	Step	Supervisore	System
Preme il botto-ne "LogOut"	3.2	Preme il botto-ne "LogOut"	
	4.2		Torna nella schermata di LogIn

Extension	Step	Supervisore	System
Preme il botto-ne "Aggiungi piatto"	3.3	Preme il botto-ne "Aggiungi piatto"	
	4.3		Va nella schermata "Aggiungi piatto"

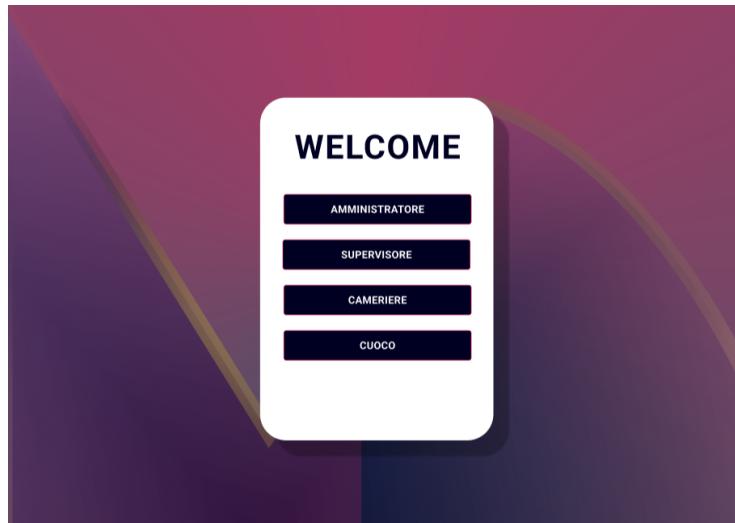
Estensione	Step	Supervisore	System
Modifica/Elimina cibo	3.4	Preme il botto-ne "Modifica elimina cibo"	
	4.4		Va nella schermata "Modifica elimina cibo"

Estensione	Step	Supervisore	System
Preme il bottone "Messaggi"	3.5	Preme il bottone "Messaggi"	
	4.5		Va nella schermata "Messaggi"

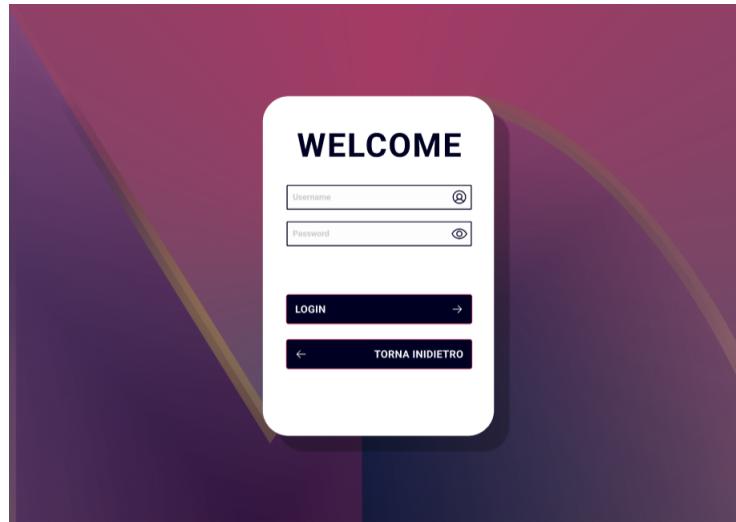
1.1.4 Prototipazione visuale via Mock-up dell'interfaccia utente per tutti i casi d'uso assegnati

Un mockup è una simulazione grafica del prodotto da sviluppare, per poterlo sviluppare abbiamo usato **Figma**, in particolare abbiamo sviluppato due prototipi: il **primo** ci è servito per poter avere una rappresentazione generale sulla grafica dell'applicazione, mentre il **secondo** è la versione definitiva con delle piccole modifiche per migliorare l'esperienza dell'utente. Di seguito vengono mostrate le varie schermate della versione finale.

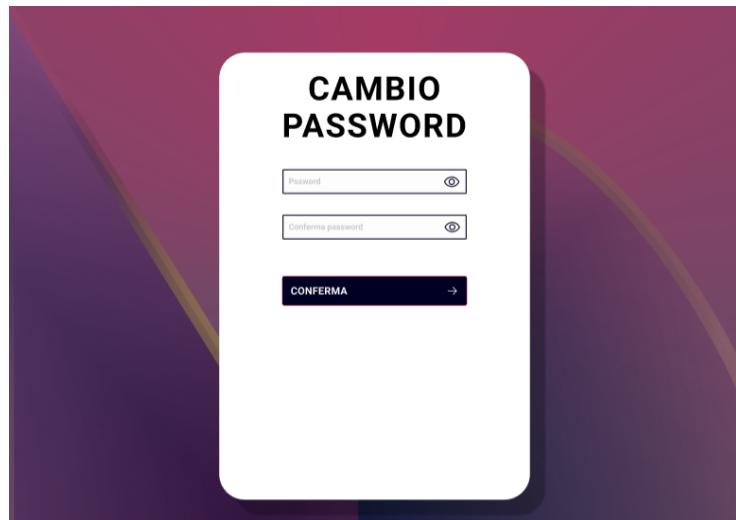
Schermata welcome



Schermata login



Schermata cambia password



Schermata ordinazioni cuoco



Schermata messaggi cuoco



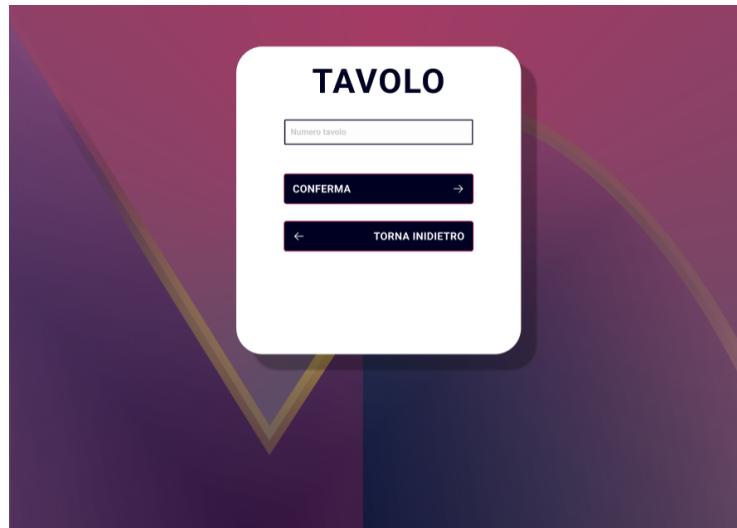
Schermata menu cameriere



Schermata visualizza ordine cameriere



Schermata conferma ordine cameriere



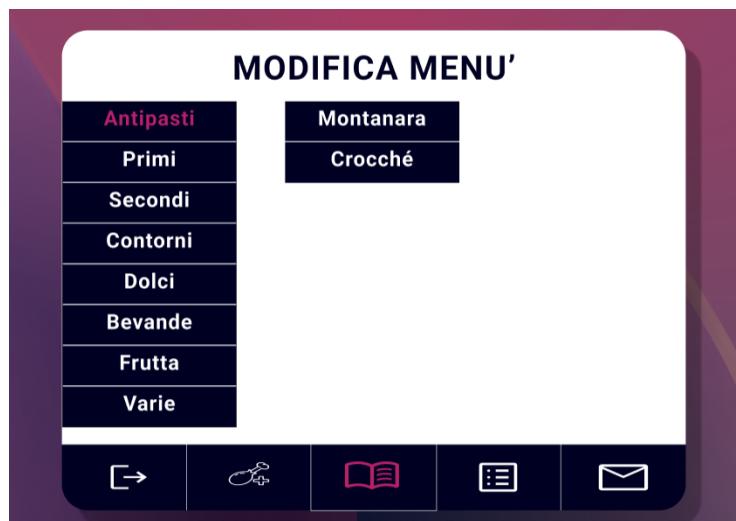
Schermata messaggi cameriere



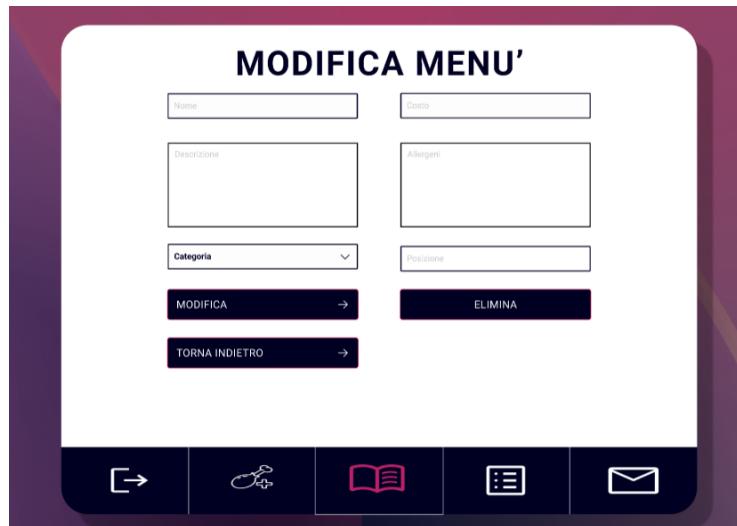
Schermata aggiungi piatto supervisore



Schermata modifica menu supervisore



Schermata modifica o elimina piatto supervisore



Schermata conto supervisore

The screenshot shows a mobile application interface titled 'CONTO'. It displays a table of items and their quantities:

Tavolo 1	Montanara	2.5€ x 2
Tavolo 2	Pasta al forno	8€ x 1
	Filetto	15€ x 1
	Caprese	5€ x 1
Tot: 30.5€		
SCARICA		CHIUDI

At the bottom of the screen are five navigation icons: a square with an arrow, a stylized fork and knife, an open book, a grid, and an envelope.

Schermata scrivi messaggio supervisore



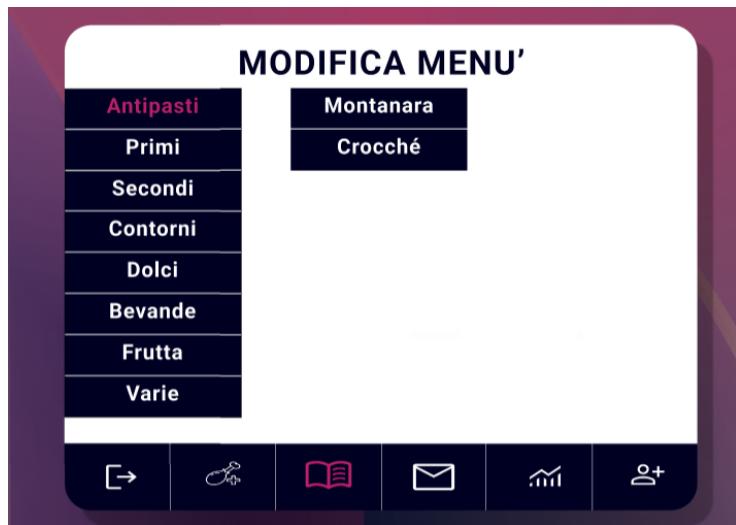
Schermata leggi messaggio supervisore



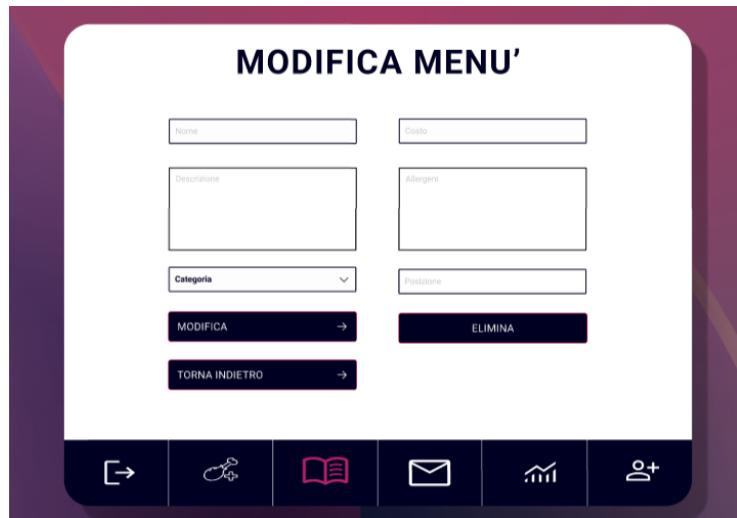
Schermata aggiungi piatto admin



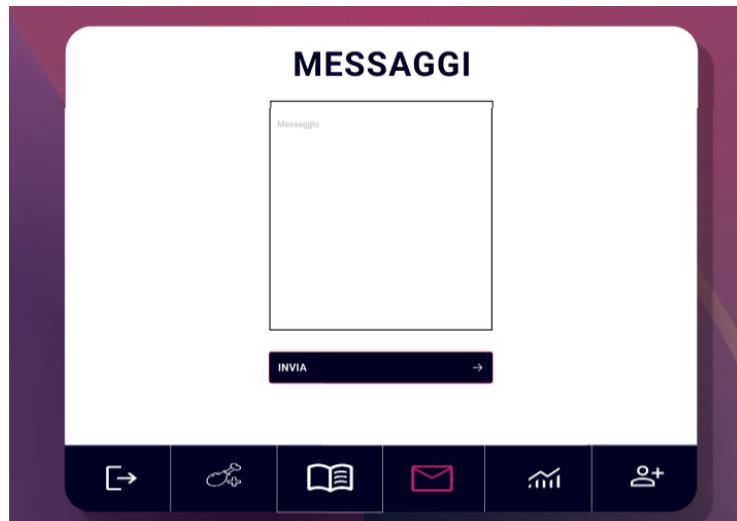
Schermata modifica menu admin



Schermata modifica o elimina piatto admin



Schermata scrivi messaggio admin



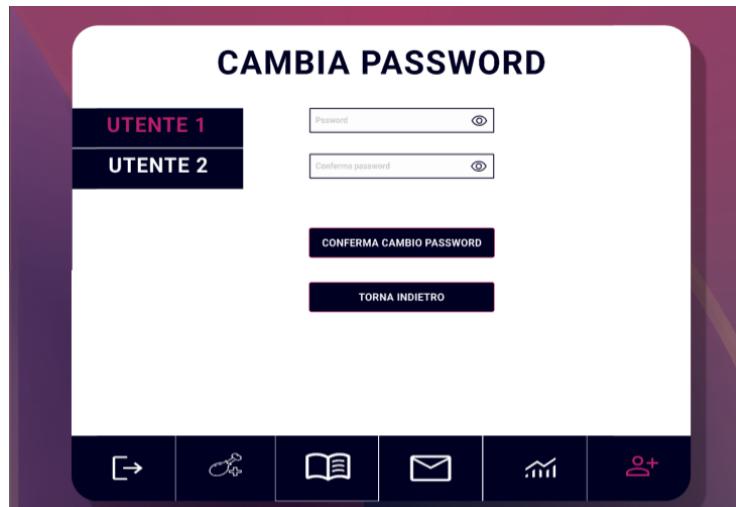
Schermata statistiche admin



Schermata crea utente admin

The screenshot shows a user creation form titled "CREA UTENTE". It includes fields for "Username" and "Password" (with an eye icon for visibility). A dropdown menu for "Ruolo" (Role) lists "Cameriere", "Cuoco", and "Supervisore". Below the form are two buttons: "CREA UTENTE" and "CAMBIA PASSWORD UTENTE". A footer navigation bar features icons for back, forward, search, file, mail, and user addition.

Schermata cambia password admin



1.1.5 Valutazione dell'usabilità a priori

Dalla realizzazione dei mock-up, abbiamo fatto sì che 4 persone (ognuna ha provato i 4 ruoli) che non hanno niente a che fare con l'informatica, li abbiano testati per misurare l'usabilità.

Da una prima analisi sono sorti problemi dal punto di vista dell'usabilità non banali. Dalla schermata di welcome, si può scegliere uno dei 4 ruoli (Amministratore, Supervisore, Cameriere e Chef). I problemi sorgono all'interno di ogni ruolo. Analizziamo i problemi per un ruolo alla volta.

1. Amministratore
 - Cambiare i nomi delle schermate
2. Supervisore
 - Cambiare i nomi delle schermate.
3. Cameriere
 - Non ci sono eventuali problemi.
4. Chef
 - Non ci sono eventuali problemi.

Problemi comuni per tutti e 4 i ruoli:

1. La schermata di logout (che sta in basso a sinistra) non porta nella schermata "Welcome" ma porta nella schermata di login del ruolo precedentemente cliccato.
2. Non esiste un tasto per "Password dimenticata", quindi è impossibile resettare la password.

Tabella di usabilità dei 4 ruoli:

S	P	F
Successo (1 pt)	Parziale (0,5 pt)	Fallimento (0 pt)

Compiti per l'amministratore:

1. Aggiungere utente.
2. Visualizzare le statistiche.
3. Inviare messaggi.
4. Modificare/Eliminare piatto.
5. Aggiungere piatto.

Per l'amministratore i compiti sono stati abbastanza intuibili (92,5% di accuracy).

	Compito 1	Compito 2	Compito 3	Compito 4	Compito 5
Gianmarco S.	S	S	S	P	S
Mattia S.	S	S	S	S	P
Matteo F.	S	S	S	S	S
Luigi P.	S	S	S	P	S

Tabella 1.1: Tabella dei compiti amministratore.

Compiti per il supervisore:

1. Inviare messaggi.
2. Visualizzare messaggi.
3. Scaricare/Chiudere conto.
4. Modificare/Eliminare piatto.
5. Aggiungere piatto.

	Compito 1	Compito 2	Compito 3	Compito 4	Compito 5
Gianmarco S.	S	S	S	P	S
Mattia S.	S	S	S	S	S
Matteo F.	S	S	S	S	S
Luigi P.	S	S	S	P	S

Tabella 1.2: Tabella dei compiti supervisore.

Per il supervisore i compiti sono stati abbastanza intuibili (95% di accuracy).

Compiti per il cameriere:

1. Effettuare l'ordinazione.
2. Visualizzare ordinazione.
3. Confermare tavolo.
4. Visualizzare i messaggi.

	Compito 1	Compito 2	Compito 3	Compito 4
Gianmarco S.	S	S	P	S
Mattia S.	S	S	S	S
Matteo F.	S	S	P	S
Luigi P.	S	S	P	S

Tabella 1.3: Tabella dei compiti cameriere.

Per il cameriere i compiti sono stati abbastanza intuibili (90,6% di accuracy).

Compiti per il cuoco:

1. Visualizzare ordinazioni.
2. Visualizzare i messaggi.

	Compito 1	Compito 2
Gianmarco S.	S	S
Mattia S.	S	S
Matteo F.	S	S
Luigi P.	S	S

Tabella 1.4: Tabella dei compiti cuoco.

Dopo le modifiche

Per il cuoco i compiti sono veramente molto facili e intuibili (100% di accuracy).

Dopo le varie analisi con gli opportuni grafici, abbiamo effettuate le modifiche precedentemente effettuate. Rifacendo i grafici calcolando le modifiche notiamo che:

Compiti per l'amministratore:

1. Aggiungere utente.
2. Cambiare password ad un utente.
3. Visualizzare le statistiche.
4. Inviare messaggi.
5. Modificare/Eliminare piatto.
6. Aggiungere piatto.

	Compito 1	Compito 2	Compito 3	Compito 4	Compito 5	Compito 6
Gianmarco S.	S	S	S	S	S	S
Mattia S.	S	S	S	S	S	S
Matteo F.	S	S	S	S	S	S
Luigi P.	S	S	S	S	P	S

Tabella 1.5: Tabella dei compiti amministratore dopo le varie modifiche.

Per l'amministratore i compiti sono stati abbastanza intuibili (97,9% di accuracy).

Compiti per il supervisore:

1. Inviare messaggi.
2. Visualizzare messaggi.
3. Scaricare/Chiudere conto.
4. Modificare/Eliminare piatto.
5. Aggiungere piatto.

	Compito 1	Compito 2	Compito 3	Compito 4	Compito 5
Gianmarco S.	S	S	S	S	S
Mattia S.	S	S	S	S	S
Matteo F.	S	S	S	S	S
Luigi P.	S	S	S	P	S

Tabella 1.6: Tabella dei compiti supervisore dopo le varie modifiche.

Per il supervisore i compiti sono stati abbastanza intuibili (97,5% di accuracy).

Compiti per il cameriere:

1. Effettuare l'ordinazione.
2. Visualizzare ordinazione.
3. Confermare tavolo.
4. Visualizzare i messaggi.

	Compito 1	Compito 2	Compito 3	Compito 4
Gianmarco S.	S	S	S	S
Mattia S.	S	S	S	S
Matteo F.	S	S	P	S
Luigi P.	S	S	P	S

Tabella 1.7: Tabella dei compiti cameriere dopo le varie modifiche.

Per il cameriere i compiti sono stati abbastanza intuibili (93,7% di accuracy).

Compiti per il cuoco:

1. Visualizzare ordinazioni.
2. Visualizzare i messaggi.

	Compito 1	Compito 2
Gianmarco S.	S	S
Mattia S.	S	S
Matteo F.	S	S
Luigi P.	S	S

Tabella 1.8: Tabella dei compiti cuoco dopo le varie modifiche.

Per il cuoco i compiti sono veramente molto facili e intuibili (100% di accuracy).

1.1.6 Glossario

Dominio Informatico

- **Requisiti funzionali** - elenchi di funzionalità o servizi che il sistema deve fornire.
- **Requisiti non funzionali** - I requisiti non funzionali rappresentano i vincoli e le proprietà/caratteristiche relative al sistema.
- **Ergonomia** - la scienza che studia l'usabilità
- **Usabilità** - l'usabilità è il "grado in cui un prodotto può essere usato da particolari utenti per raggiungere certi obiettivi con efficacia, efficienza e soddisfazione in uno specifico contesto d'uso.
- **Robustezza** - La robustezza di un sistema è la misura in cui il sistema si comporta in modo ragionevole in situazioni impreviste, non contemplate dalle specifiche.
- **Affidabilità** - Un sistema è tanto più affidabile quanto più raramente, durante l'uso del sistema, si manifestano malfunzionamenti.
- **Sicurezza** - Il sistema deve essere protetto da accessi non autorizzati.
- **Performance** - Il sistema deve essere in grado di gestire il numero richiesto di utenti senza alcun degrado delle prestazioni.
- **Scalabilità** - Il sistema deve essere in grado di aumentare o diminuire in base alle esigenze.
- **Disponibilità** - Il sistema deve essere disponibile quando necessario.
- **Manutenzione** - Il sistema deve essere di facile manutenzione e aggiornamento.
- **Portabilità** - Il sistema deve essere in grado di funzionare su piattaforme diverse con modifiche minime.
- **Stakeholders** - Tutti i soggetti coinvolti nello sviluppo del software.
- **Personas** - Sono dei personaggi immaginari creati sulla base di una ricerca condotta su utenti reali allo scopo di identificare gli utenti-tipo di un prodotto/servizio.
- **Design Pattern** - è una soluzione progettuale generale ad un problema ricorrente.
- **Use Case Diagram** - rappresenta una funzione o servizio offerto dal sistema a uno o più attori.
- **Tabelle di CockBurn** - Tabelle di rappresentazione di un caso d'uso create da Alistair Cockburn.

- **Mock-up** - Una rappresentazione iniziale e teorica che illustrano una rappresentano una bozza, senza le funzionalità del prodotto finale.
- **Class Diagram** - diagrammi previsti in un modello UML che consentono di descrivere tipi di entità, con le loro caratteristiche e le eventuali relazioni.
- **Sequence Diagram** - diagramma previsto dall'UML utilizzato per descrivere una determinata sequenza di azioni.
- **StateChart** - descrive il comportamento di sistemi reattivi. Le transizioni descrivono come l'entità modellata reagisce a eventi (generati dall'esterno o da se stesso).
- **Android** - Sistema Operativo mobile.
- **Front-end** - parte visibile all'utente di un programma e con cui egli può interagire.
- **Back-end** - parte di sviluppo che non opera sul client e che fornisce un servizio.
- **Server** - macchina che ha il compito di offrire servizi ai client che li richiedono.
- **White Box** - Modalità di testing in cui un metodo viene testato in base al codice che contiene, alla struttura dell'applicazion.
- **Black Box** - Modalità di testing in cui un'unità viene testata in base ai requisiti di quest'ultima.

Dominio Ristorazione

- **Amministratore** - Colui che crea utenti, personalizza il menù, invia e visualizza gli avvisi, visualizza statistiche sugli introiti dell'attività di ristorazione, dato un certo intervallo di tempo personalizzabile.
- **Supervisore** - Colui che personalizza il menù, invia e visualizza gli avvisi, può visualizzare e chiudere i conti dei tavoli.
- **Cameriere** - Colui che prende le comande, visualizza gli avvisi, aggiunge un tavolo per le ordinazioni e visualizza il conto.
- **Chef** - Colui che visualizza le comande per poter effettuare i piatti richiesti dai clienti e visualizza gli avvisi.
- **Messaggio** - Serve per segnare eventuali avvisi che vengono comunicati dall'amministratore o dal supervisore.
- **Statistiche** - dato un certo intervallo di tempo personalizzabile, è possibile visualizzare l'incasso medio, il valore medio di ciascun conto, l'incasso complessivo.

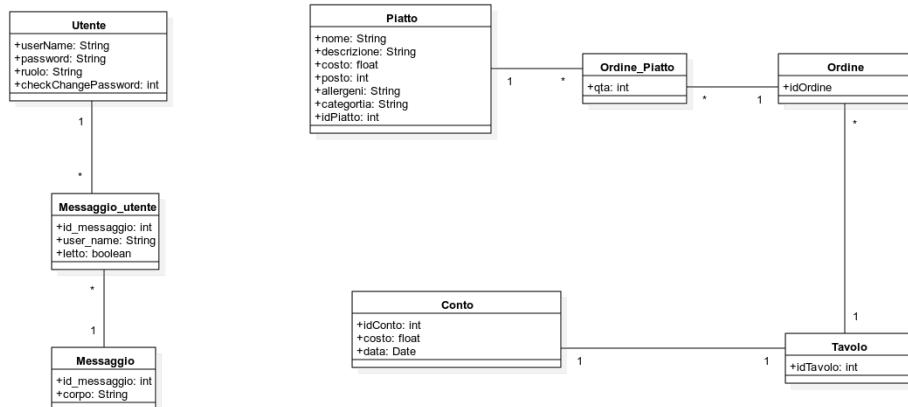
- **Piatto** - Pietanza ordinata dal cliente.
- **Menù** - Lista di piatti che possono essere ordinati dai clienti.
- **Conto** - La lista di tutti i piatti ordinati dal cliente con il corrispettivo costo.
- **Ordine** - La lista di tutti i piatti ordinati dal cliente.
- **Tavolo** - Viene occupato dai clienti per mangiare e si libera nel momento in cui i commensali pagano il conto.

1.2 Specifica dei Requisiti

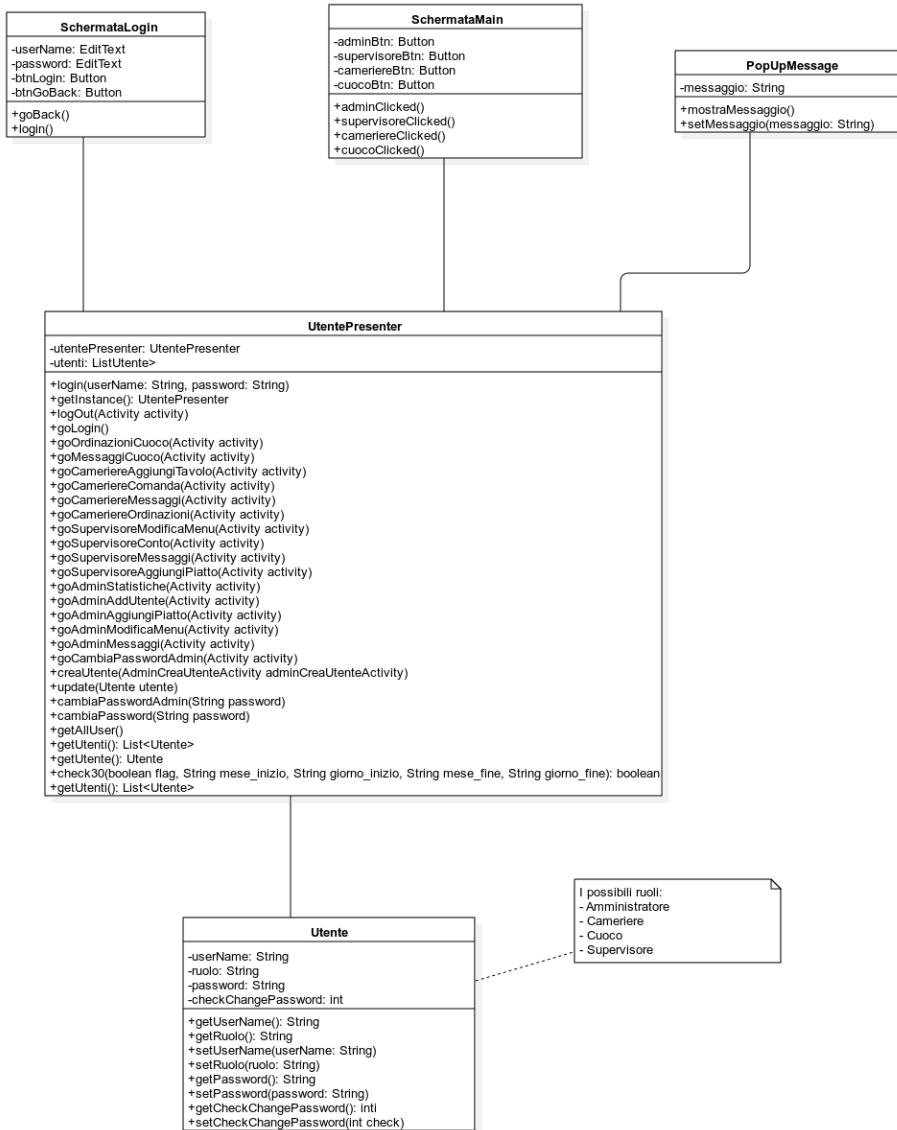
In questa fase andiamo a schematizzare le varie funzionalità senza scendere in troppi dettagli, in questo modo grazie a dei **class diagram** riusciamo a individuare quali classi dobbiamo implementare e le loro relazioni, quali sono e in quale ordine vengono eseguite le operazioni degli oggetti tramite i **sequence diagram** e infine per descrivere il comportamento delle classi utilizziamo degli **statechart**.

1.2.1 Classi, oggetti e relazioni di analisi

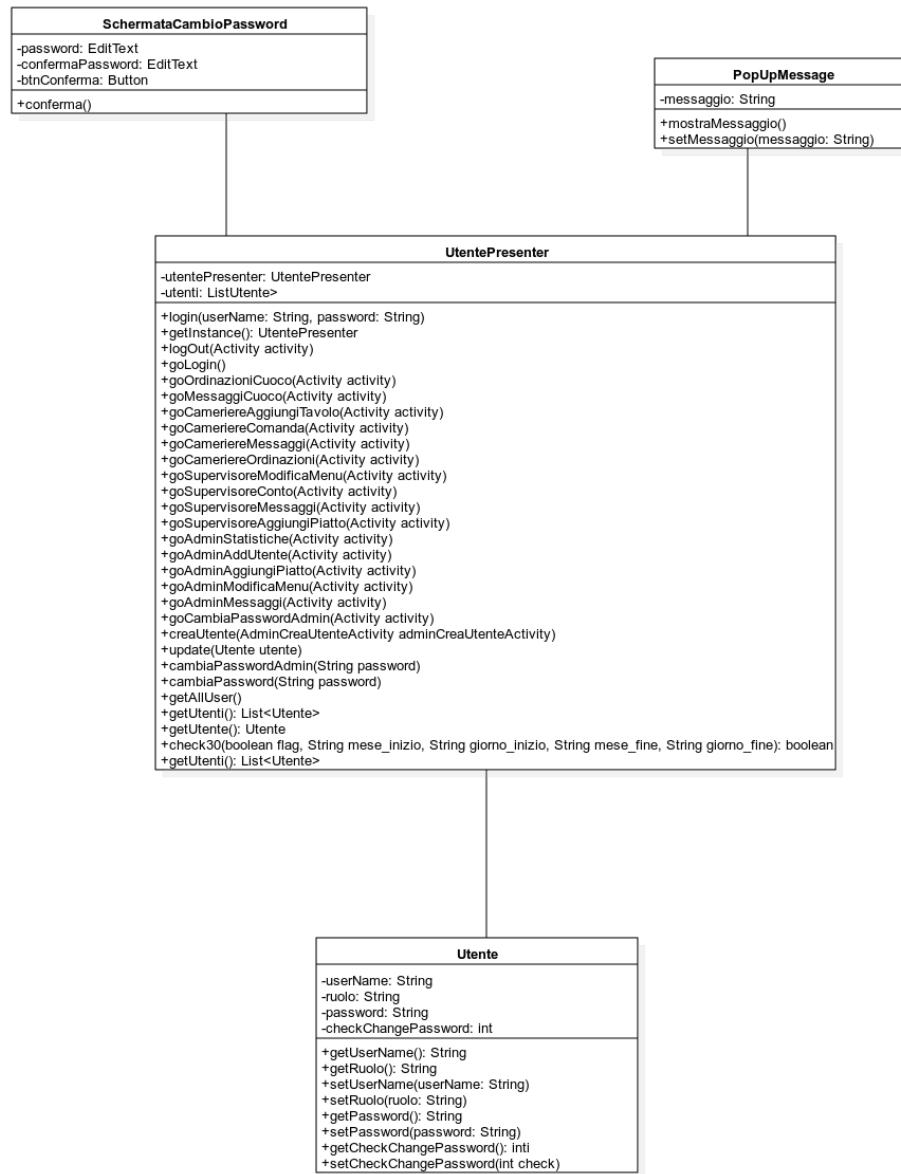
Entità



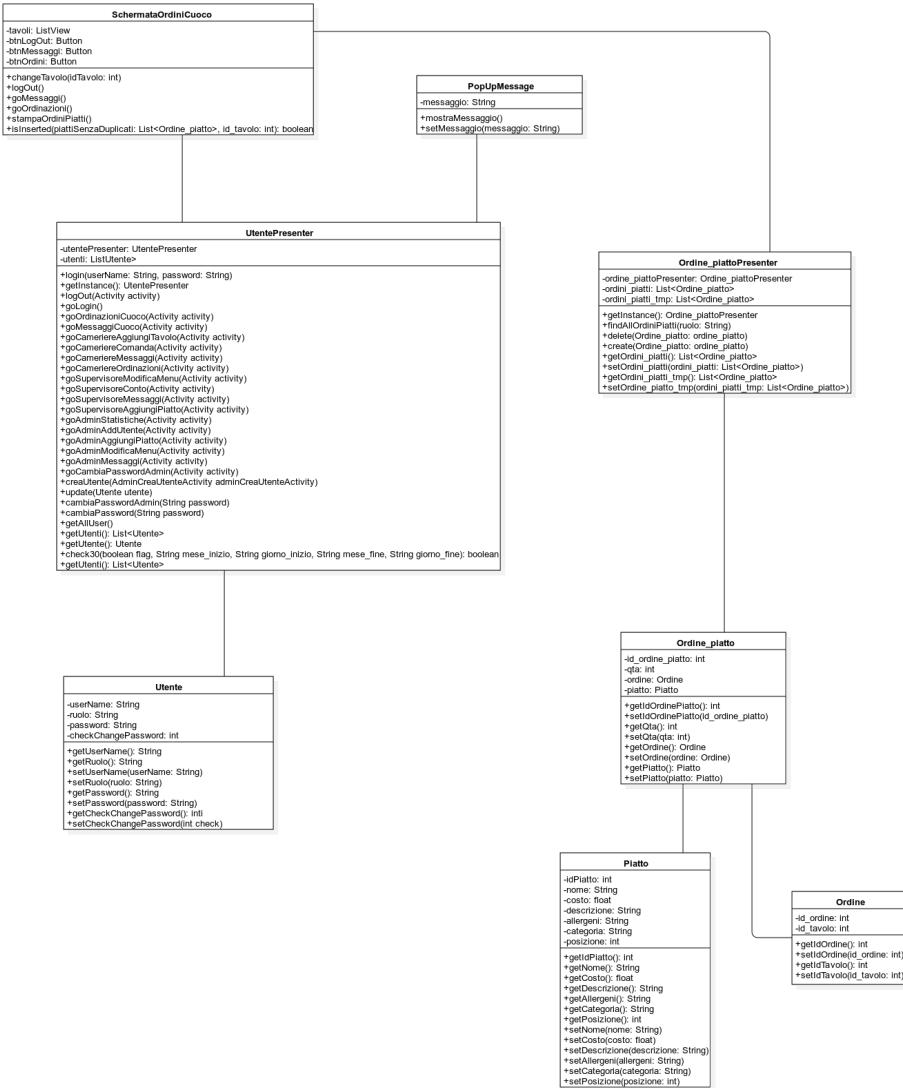
Login



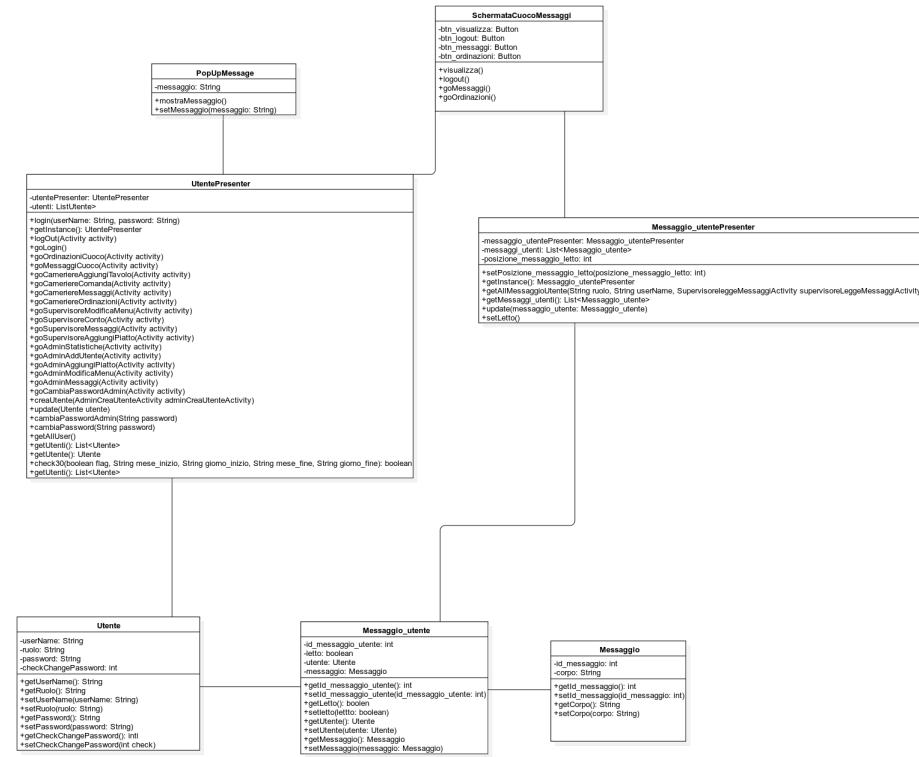
Cambia password



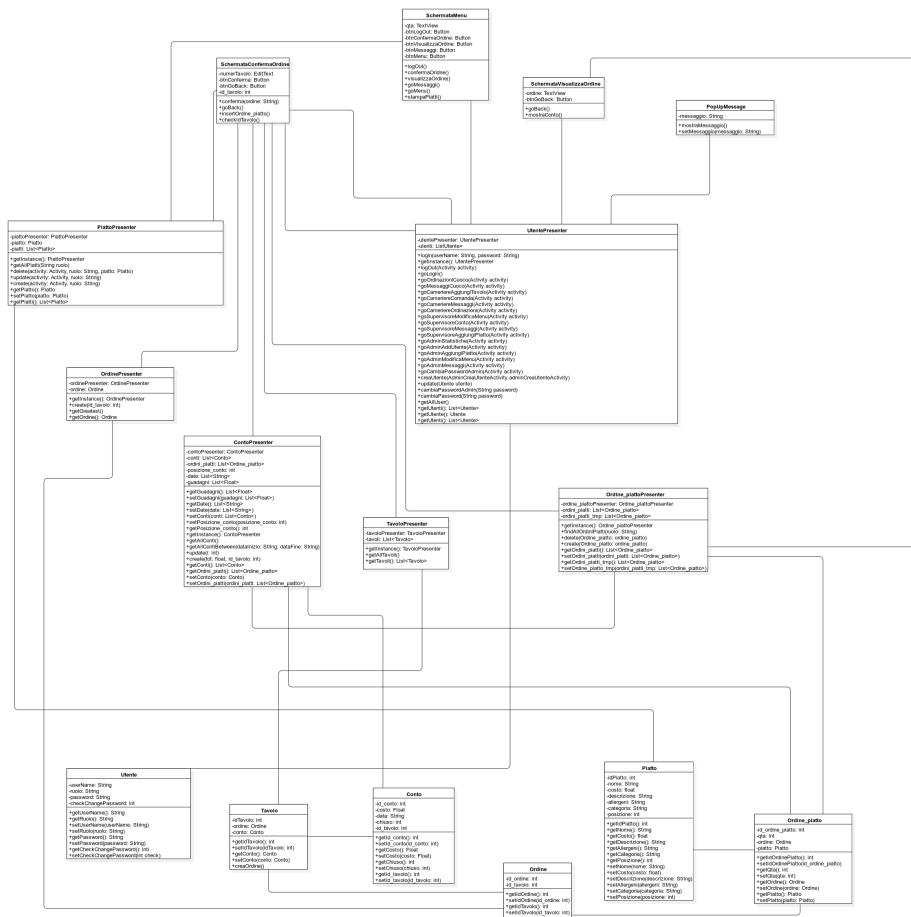
Ordinazioni cuoco



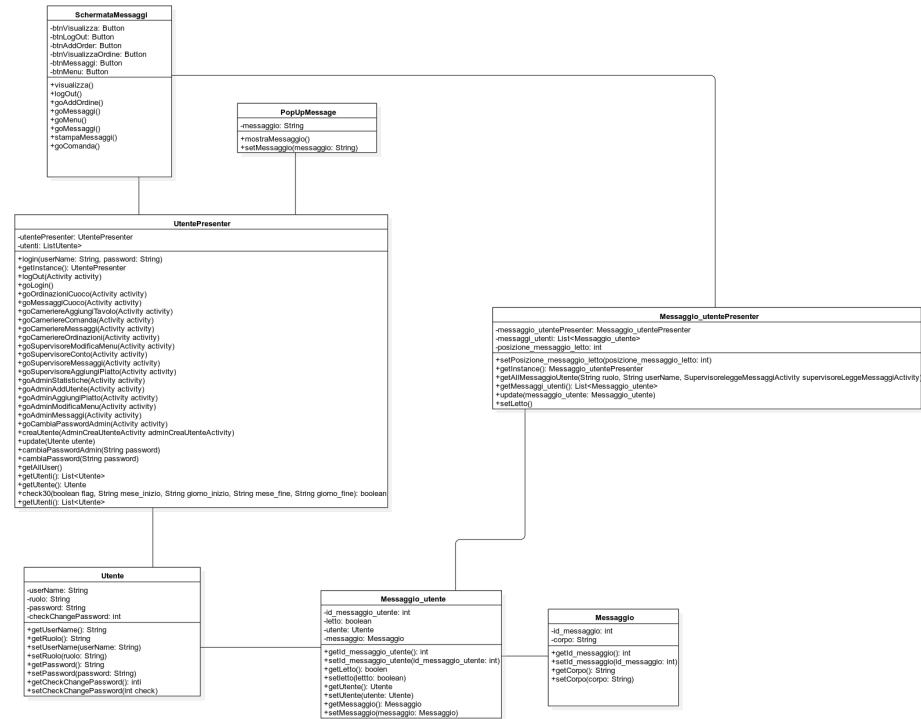
Messaggi cuoco



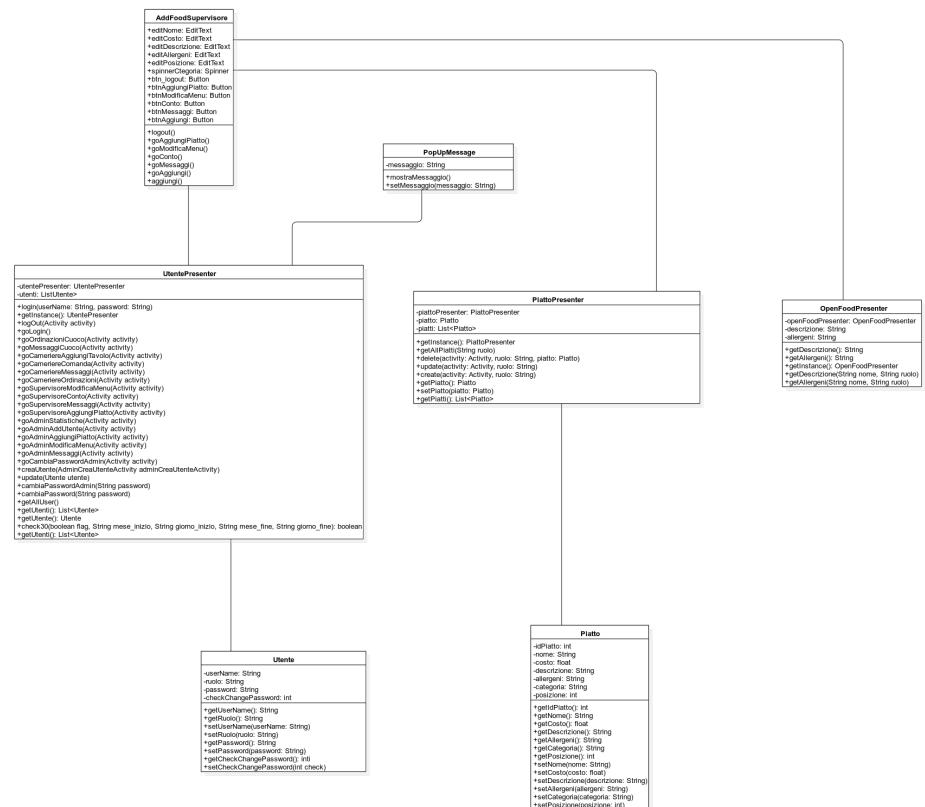
Ordinazione cameriere



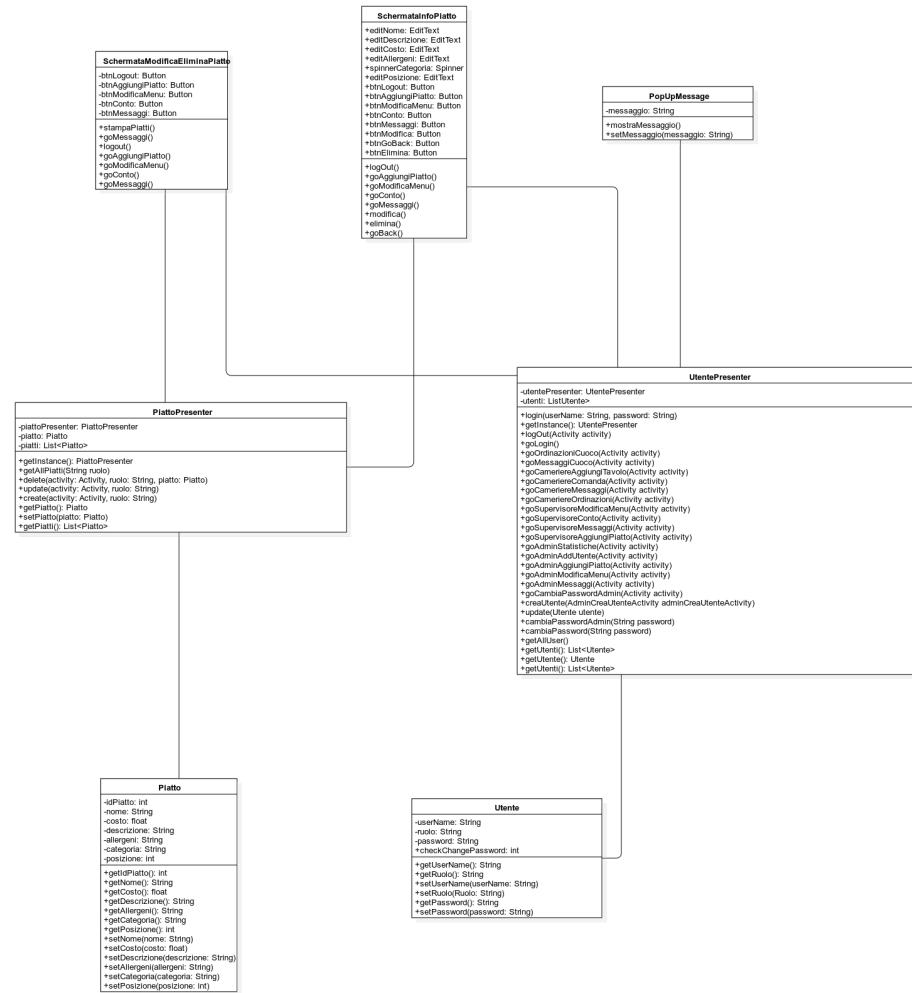
Messaggi cameriere



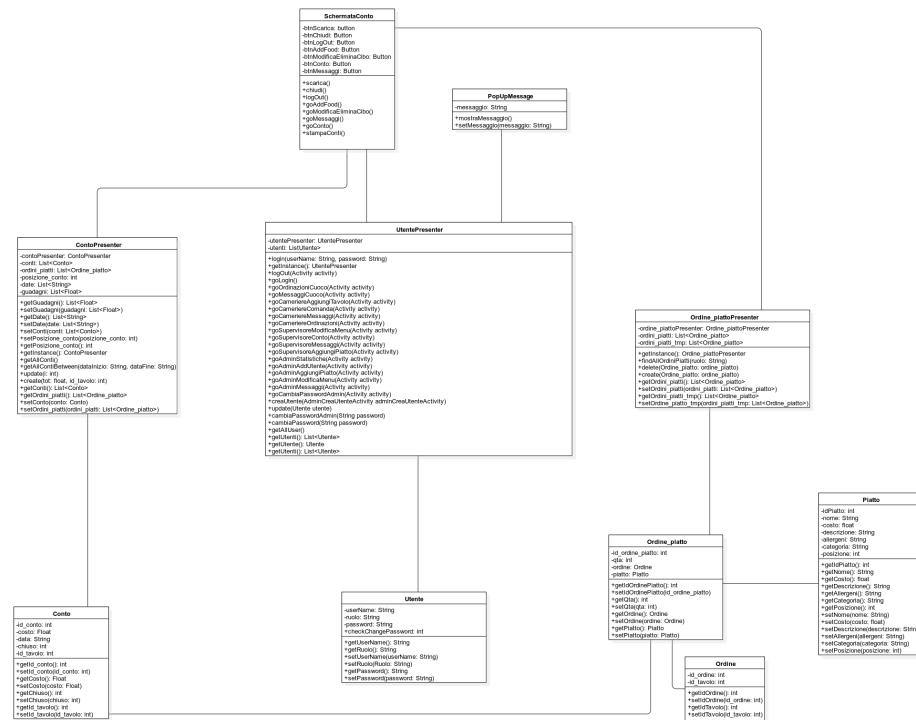
Crea piatto supervisore



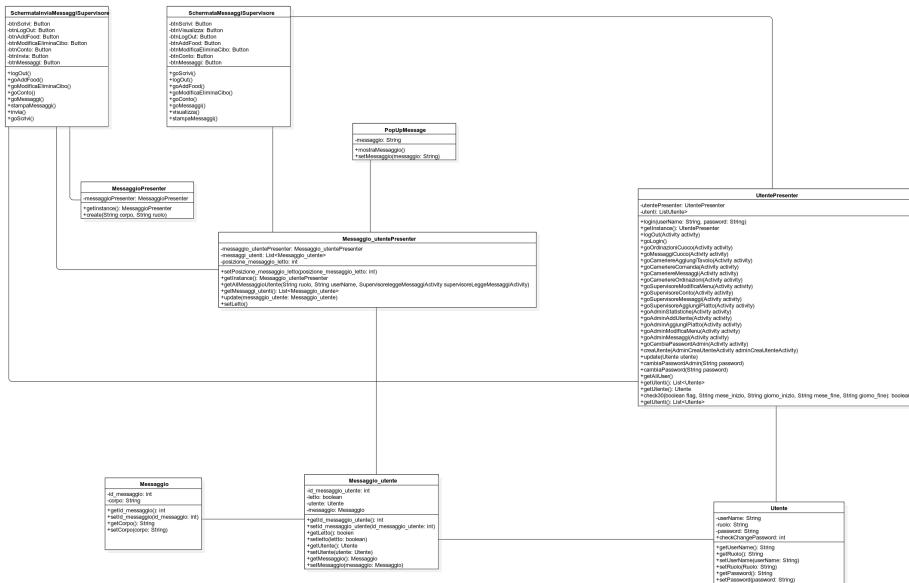
Modifica menu supervisore



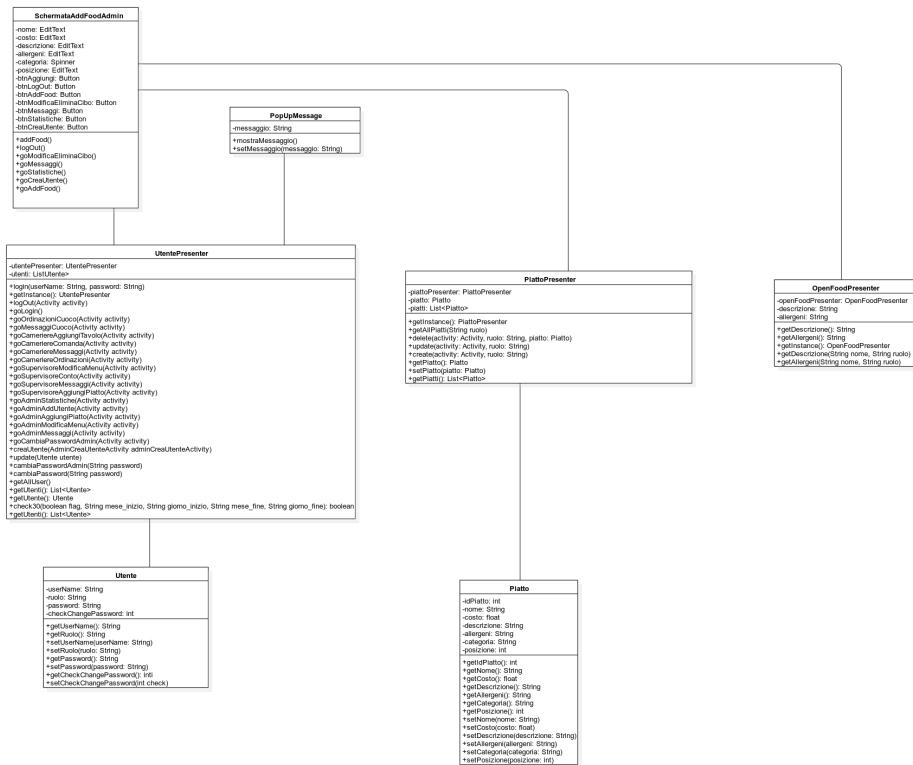
Conto supervisore



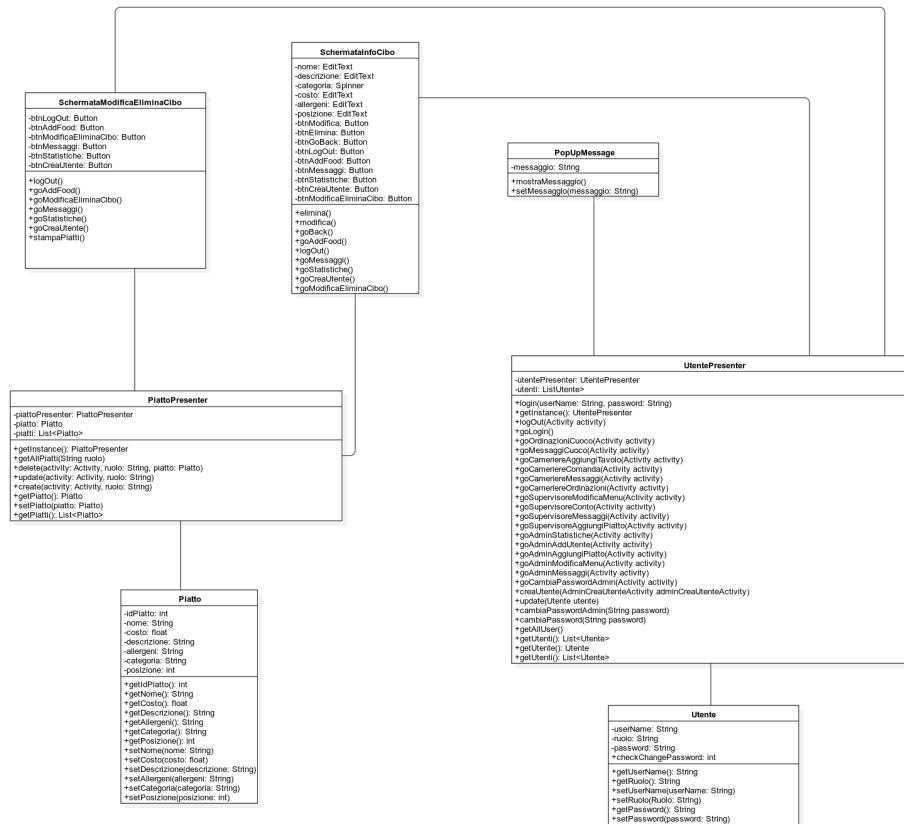
Messaggi supervisore



Aggiungi piatto admin



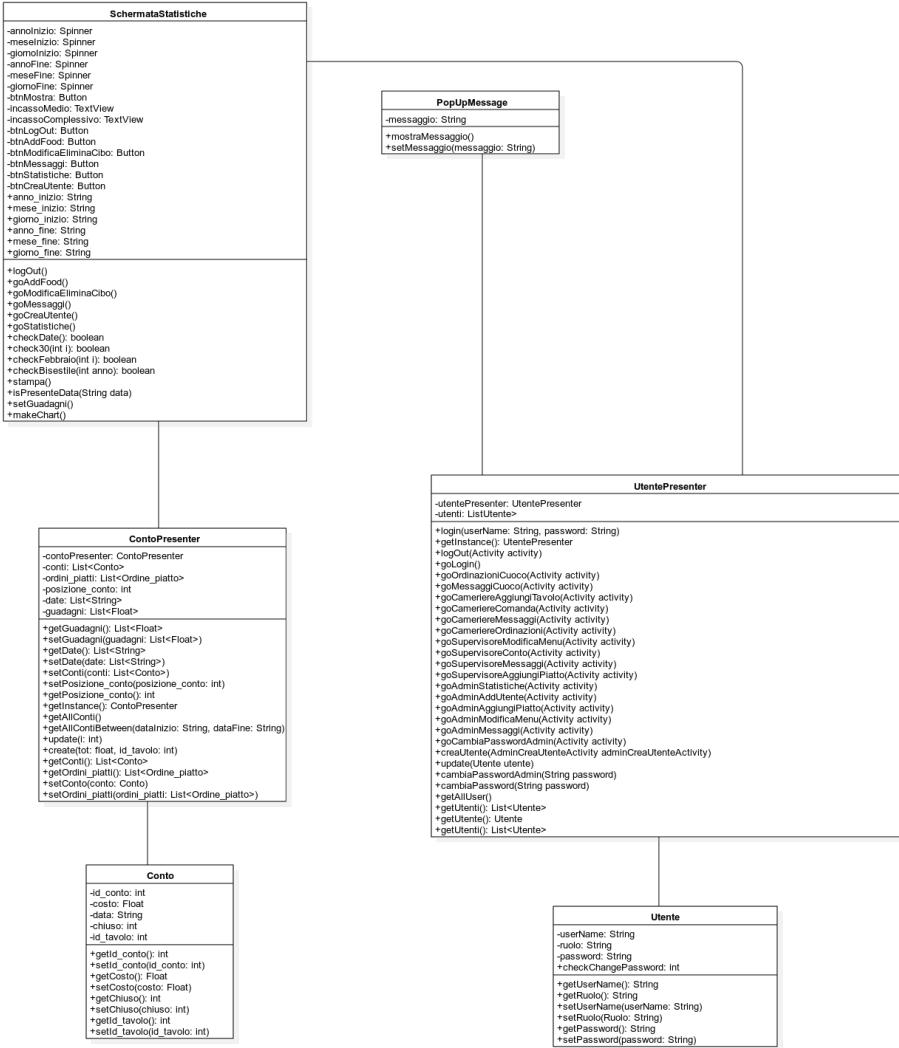
Modifica menu admin



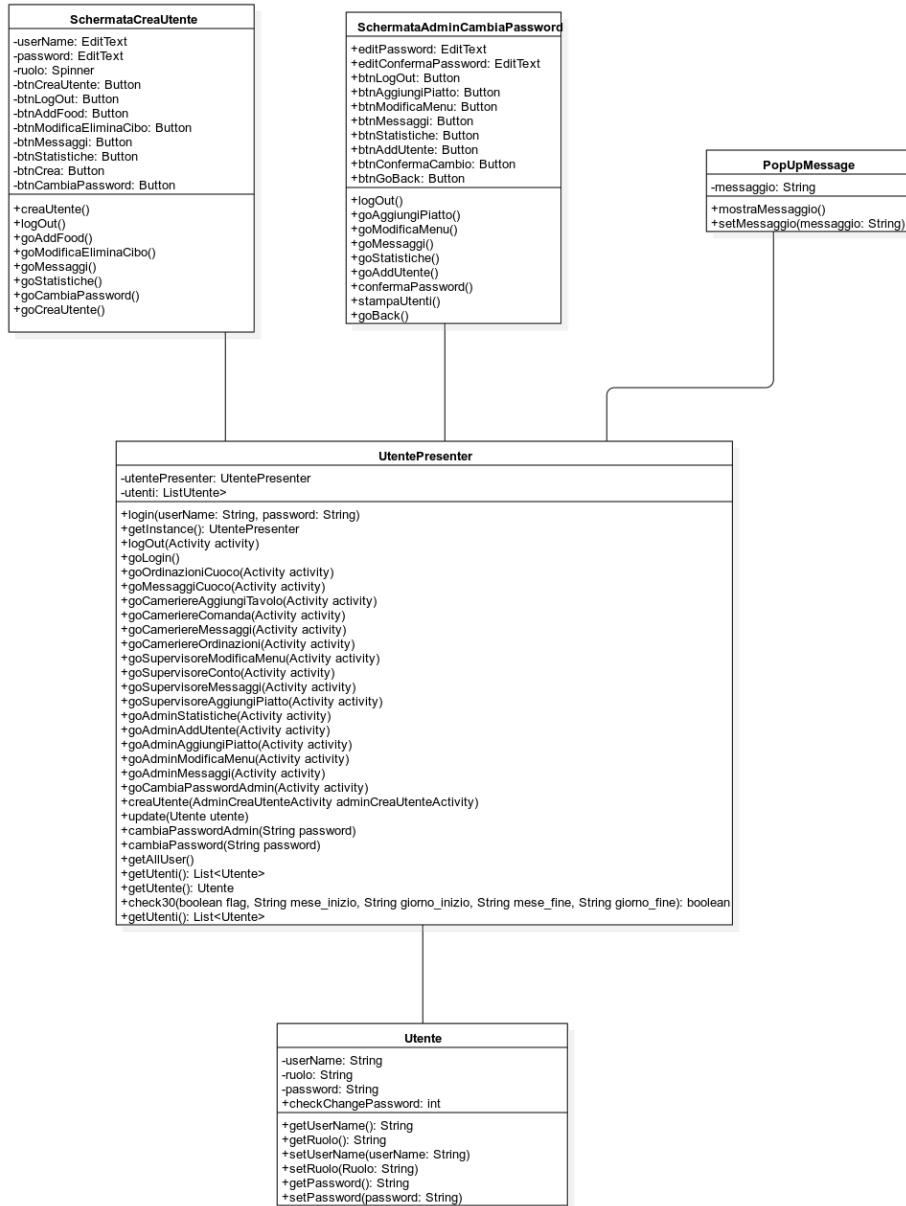
Messaggi admin



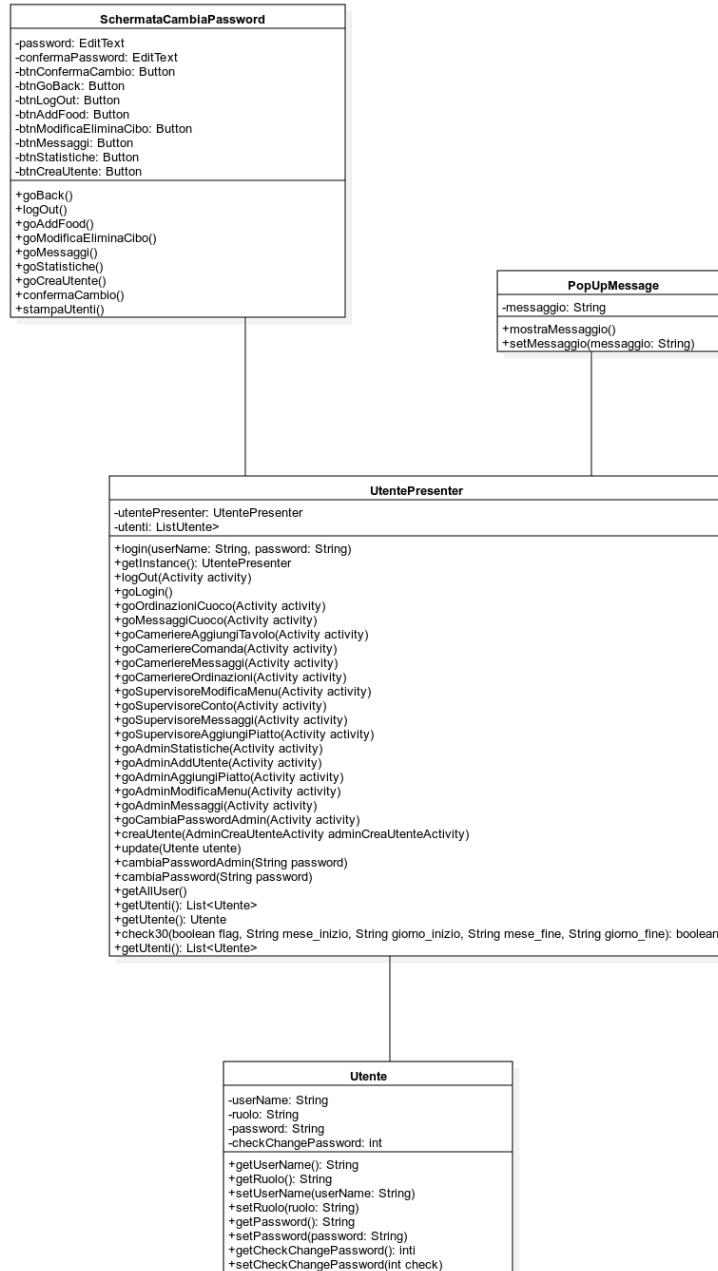
Statistiche admin



Crea utente admin

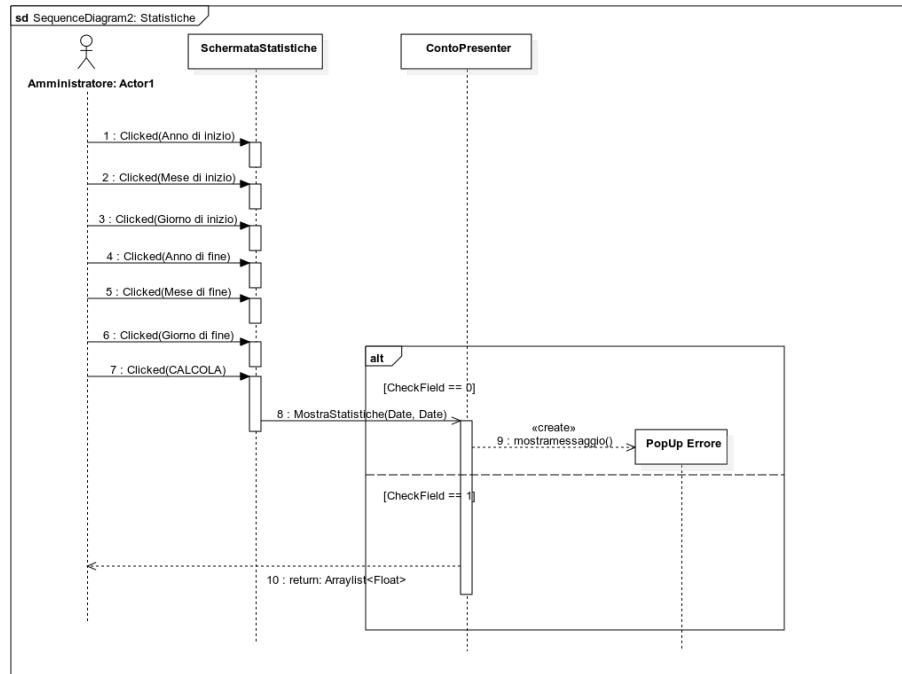


Cambia password admin

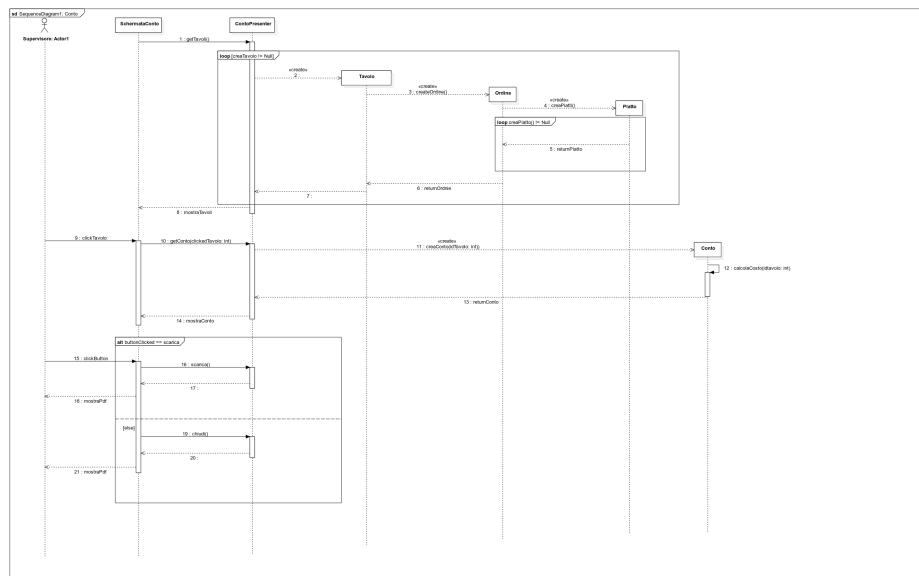


1.2.2 Diagrammi di sequenza di analisi

Statistiche

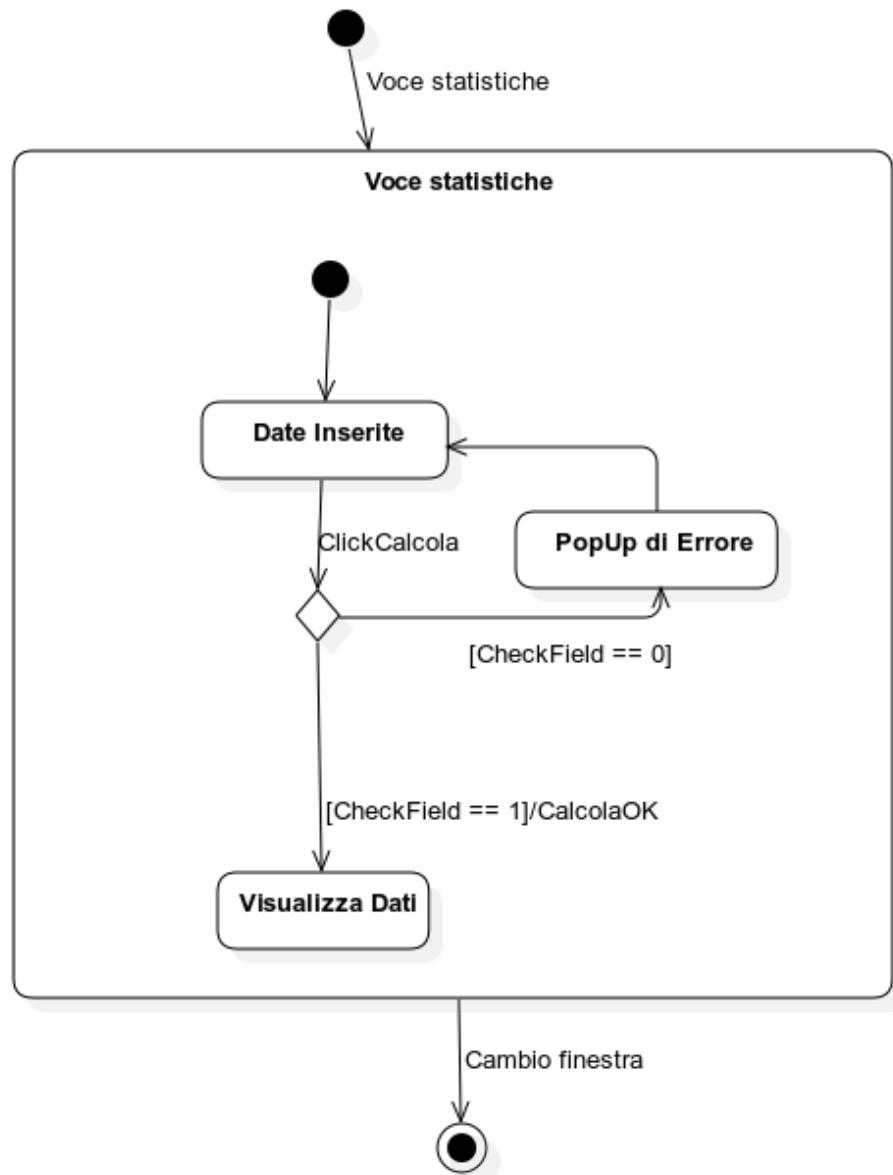


Conto

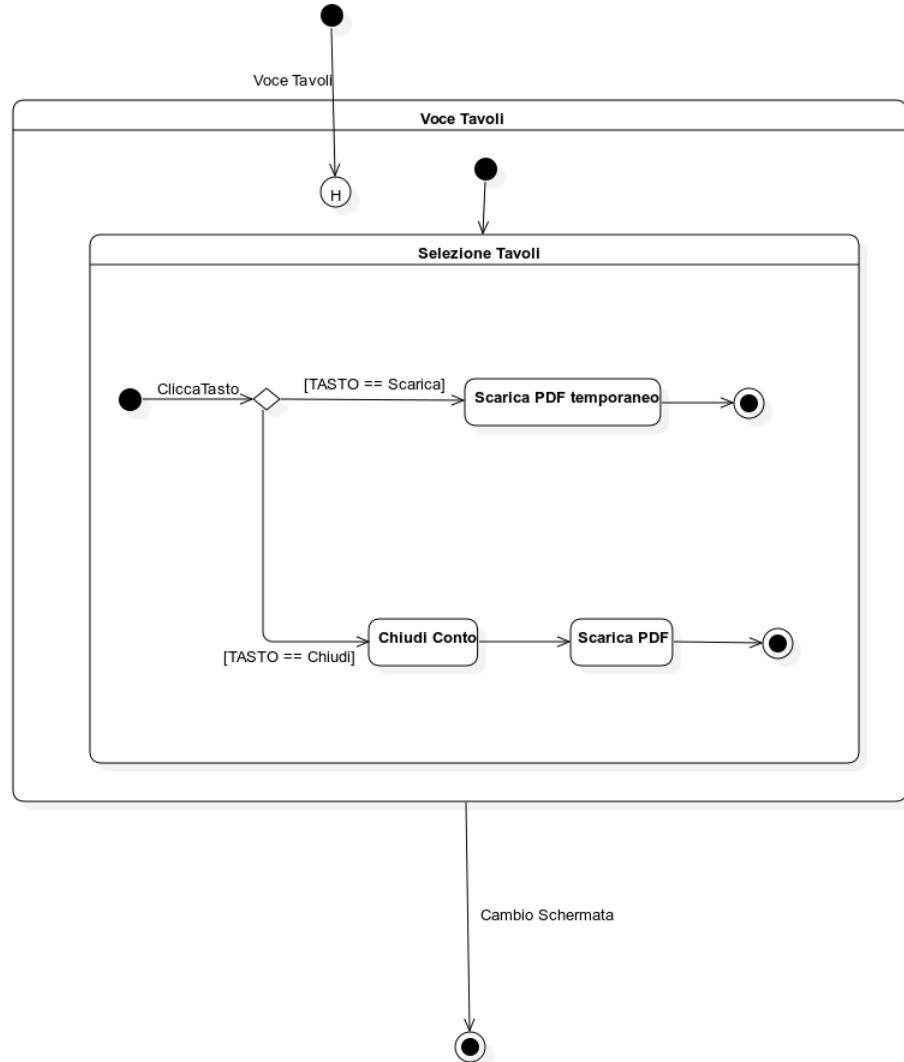


1.2.3 Prototipazione funzionale via statechart dell'interfaccia grafica

Visualizza statistiche



Visualizza statistiche



Capitolo 2

Documento di Design del Sistema

2.1 Analisi dell'architettura con esplicita definizione dei criteri di design

2.1.1 Architettura esterna

L'architettura esterna del sistema è suddivisa in tre differenti layers, ognuno dei quali può accedere esclusivamente al layer immediatamente sottostante, rendendo dunque l'architettura chiusa. I clienti inviano delle richieste al server il quale le elabora e le proroga al database, in questo modo non vi è una comunicazione diretta tra client e database rendono il sistema più sicuro.

2.1.2 Architettura Server

Il back-end è distribuito come container Docker in opera utilizzando il servizio di public cloud computing di AWS. In particolare l'architettura del server è implementata nel seguente modo:

- **Controller:** i controller gestiscono le richieste HTTP per recuperare i dati dal database e inviarli al client, sfrutta quindi i Model che converte in ModelDto per inviare al client le informazioni più utili.
- **Repository:** le varie repository si occupano delle comunicazioni con il database con delle operazioni CRUD, tramite JPA è possibile usare delle query implicite andando a definire solo la signature del metodo, in questo modo è possibile astrarre l'applicativo dall'implementazione del database. In ogni caso è possibile anche definire delle query esplicite.
- **Service:** le varie classi service fanno da ponte tra i controller e le repository per poter avere un livello di astrazione maggiore e facilitare un eventuale modifica del database.

- **Model:** qui vengono definite le varie classi model che rispecchiano le entità del database seguendo però delle regole di JPA per poter permettere la comunicazione.
- **ModelDto:** le classi di ModelDto sono come quelle del Model ma con degli attributi in meno o diversi per poter mandare al client solo le informazioni che gli sono utili (trascurando quindi alcuni attributi inseriti per il funzionamento di JPA).

Tecnologie usate per il server

Le tecnologie utilizzate per l'implementazione del server sono le seguenti:

- **SpringBoot:** è un framework java che semplifica e velocizza lo sviluppo di applicazioni web tramite delle configurazioni automatiche.
- **JPA:** è un framework che si occupa della gestione e della persistenza dei dati in un database relazionale.
- **Docker:** strumento utilizzato per creare il container del Server e del Database che tramite una rete condivisa il server può essere eseguito e comunicare con il database in un ambiente sicuro e facilmente distribuibile.

2.1.3 Architettura Client

Il nostro applicativo mobile è realizzato in Android, nello specifico, è realizzato facendo riferimento al pattern MVP (Model–View–Presenter). In particolare il Model è lo strato di Data Access per la gestione dei dati, la View lavora con il Presenter per mostrare i dati e notificare le azioni dell'utente. Il Presenter lavora tra il Model e la View. A differenza del classico MVC (Model - View - Controller), il pattern MVP si adatta meglio all'ambiente Android poiché la View non comunica con il Model ed il Presenter è responsabile dell'associazione del Model alla View. Inoltre permette di scrivere test con più facilità e facilita la manutenzione e la leggibilità. In particolare i package creati sono:

- **Presenter:** permette la comunicazione tra le varie view e con model, inoltre saranno i presenter a chiamare i metodi dei service per poter comunicare con il server.
- **View:** vengono definite le varie activity, quindi hanno solo scopo grafico per mostrare i dati e raccogliere le interazioni dell'utente.
- **Model:** definizione delle varie entità.
- **API:** interfacce per la comunicazione con il server, tramite loro vengono inviate le richieste HTTP.
- **Service:** fanno da ponte tra il presenter e le API.

Tecnologie usate per il client

Le tecnologie usate per implementare il client sono le seguenti:

- **Retrofit:** framework usato per accedere ai servizi HTTP e quindi per comunicare con il server.
- **MAndroidChart:** libreria usata per generare grafici per le statistiche dell'admin.
- **iTextpdf:** libreria usata per creare file pdf che contengono le varie informazioni sul conto di un tavolo.

Il client oltre a comunicare con il server comunica anche con OpenFoodFact per ricevere le informazioni dei piatti per l'autocompletamento.

2.1.4 Servizi Cloud

- **EC2:** servizio di amazon che fornisce macchine virtuali dove poter far eseguire l'applicazione web, è tramite un'istanza ec2 che viene eseguito il server e il database tramite docker.

2.2 Descrizione/motivazione delle scelte tecnologiche adottate

2.2.1 Perché android?

- I dispositivi android sono meno costosi. Quindi in un ottica in cui un imprenditore dovesse comprare dei dispositivi per i suoi dipendenti, in proporzione un dispositivo android è abbastanza economico.
- Android gradisce le notifiche in stile “pop-up” e il multitasking.
- Il testing è molto intuitivo e simile anche ad altri linguaggi di programmazione (come ad esempio Java).

2.2.2 Perché AWS?

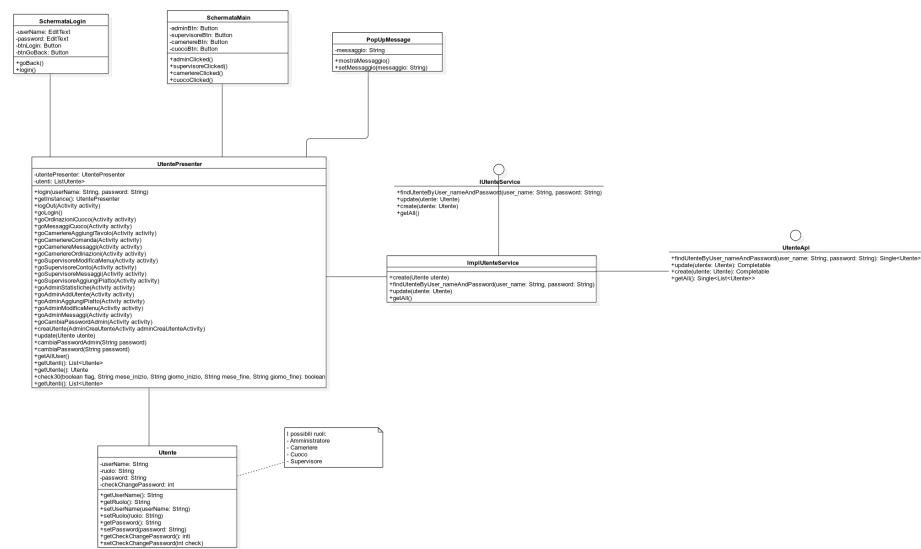
- AWS permette di selezionare il sistema operativo, il linguaggio di programmazione, la piattaforma di applicazione Web, il database e altri servizi secondo necessità.
- I costi si basano esclusivamente sulla potenza di elaborazione, lo storage e le risorse utilizzate, senza contratti a lungo termine né impegni anticipati.
- Grazie all'infrastruttura di Amazon, potremo ricalibrare le risorse di calcolo e di storage in base alle esigenze.

- AWS adotta un approccio alla sicurezza completo che garantisce l'affidabilità dell'infrastruttura, sia dal punto di vista fisico e operativo, sia a livello di software.

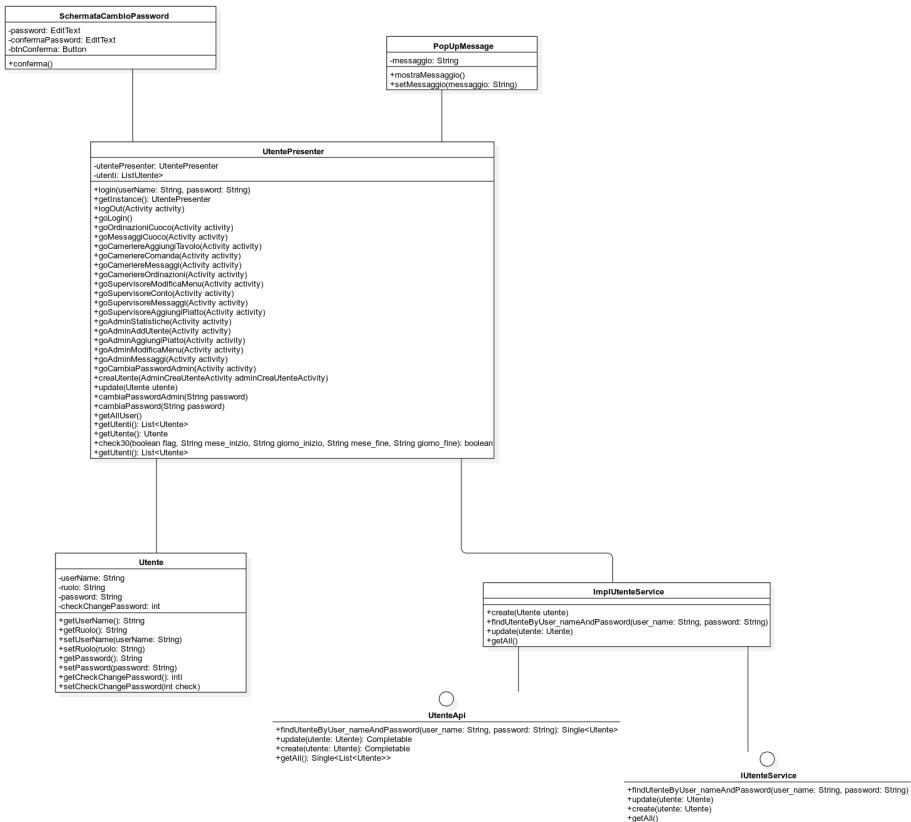
2.3 Diagramma delle classi di design

In questa fase mostriamo i class diagram e i sequence diagram ad un livello più dettagliato.

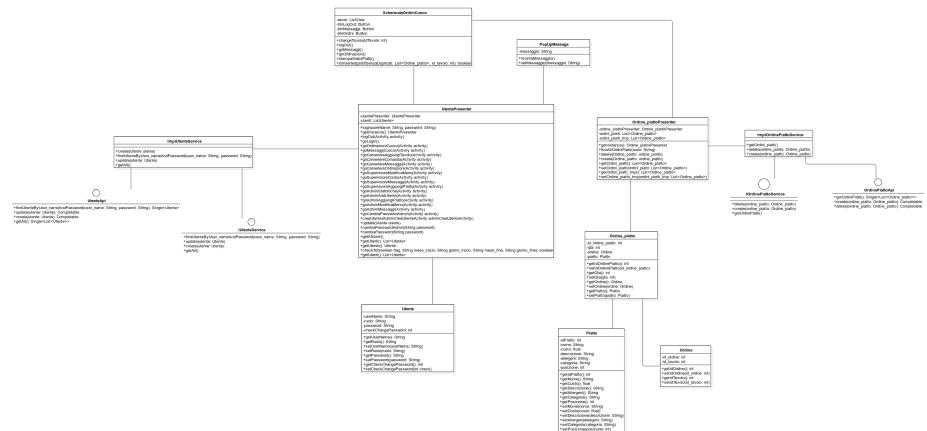
Login



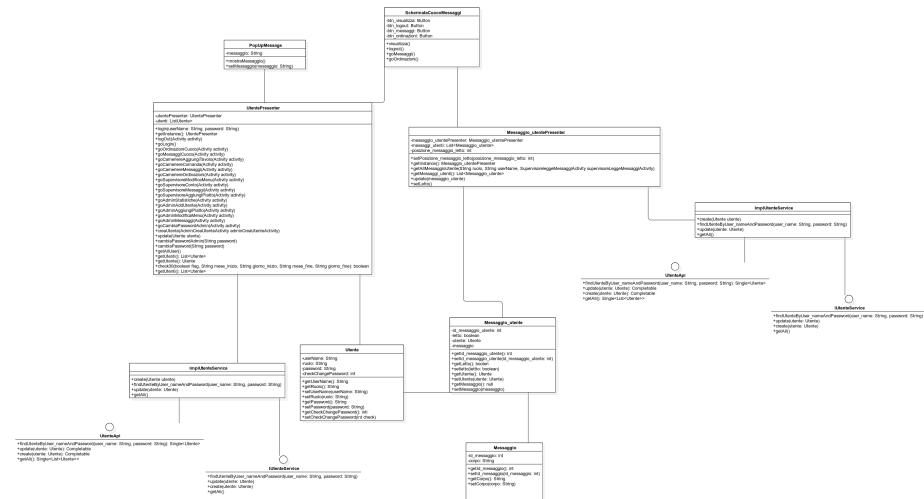
Cambia password



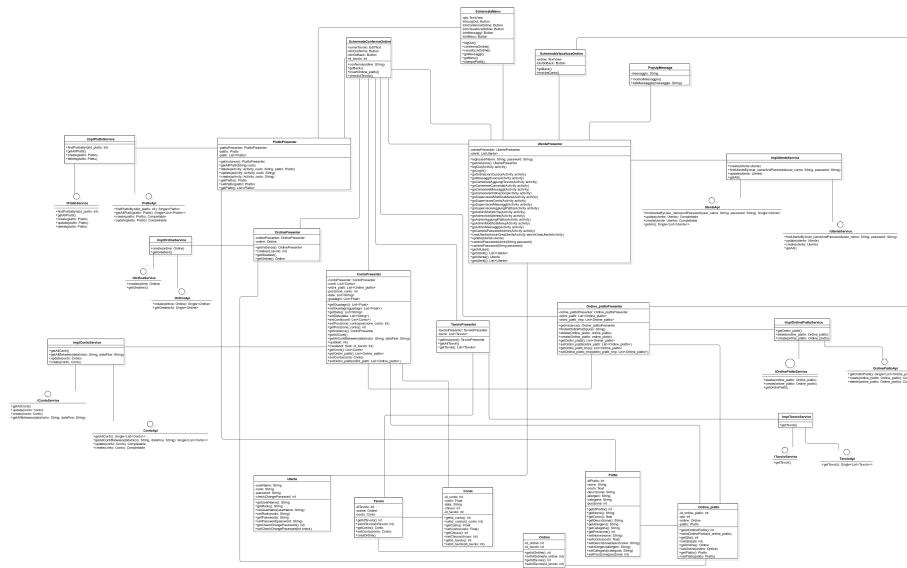
Ordinazioni cuoco



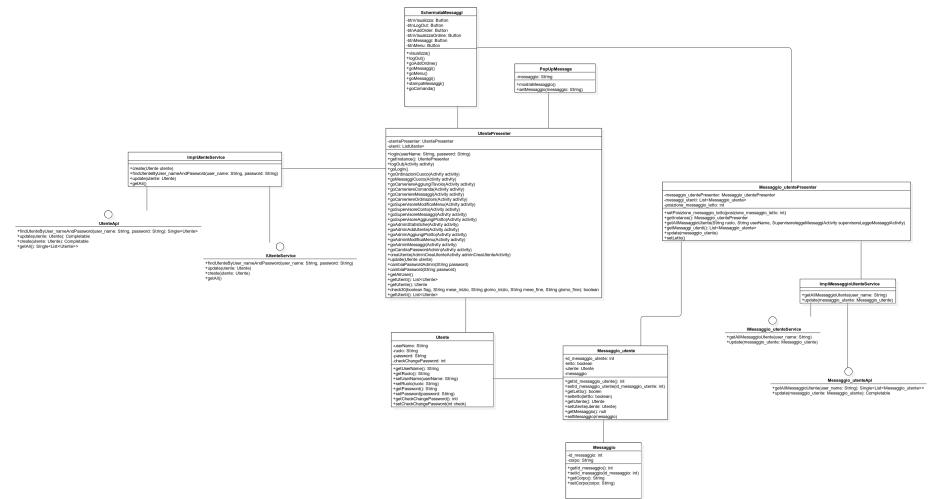
Messaggi cuoco



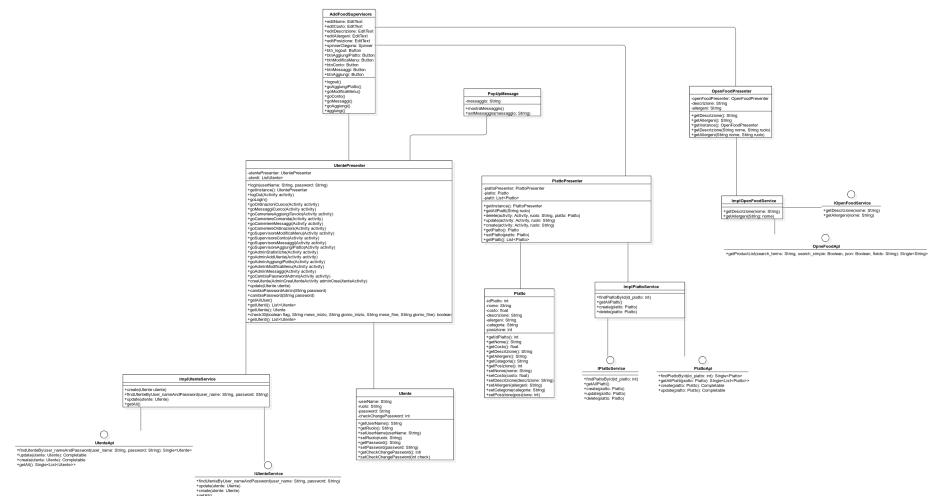
Ordinazione cameriere



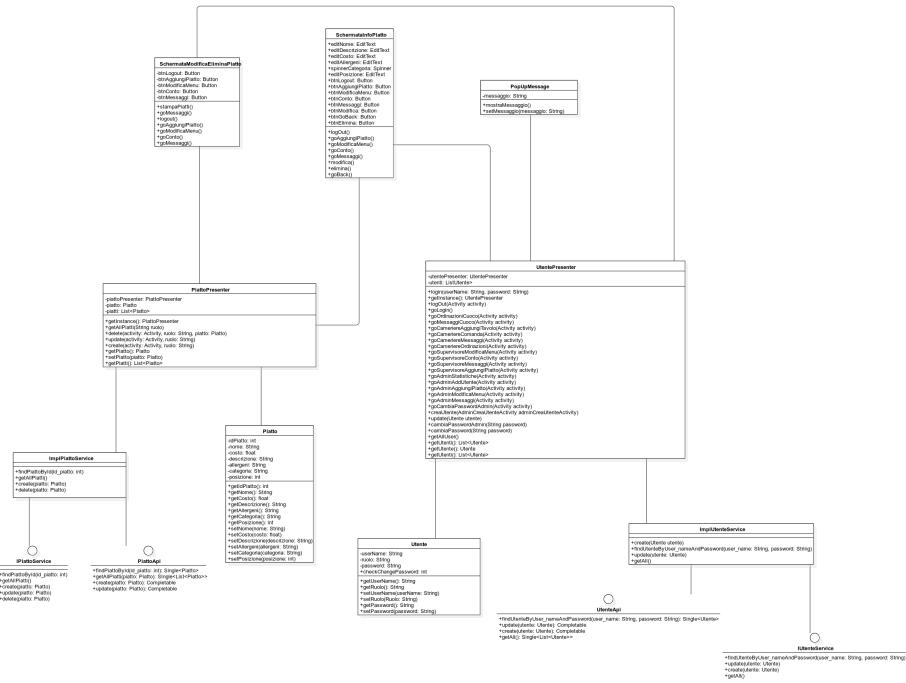
Messaggi cameriere



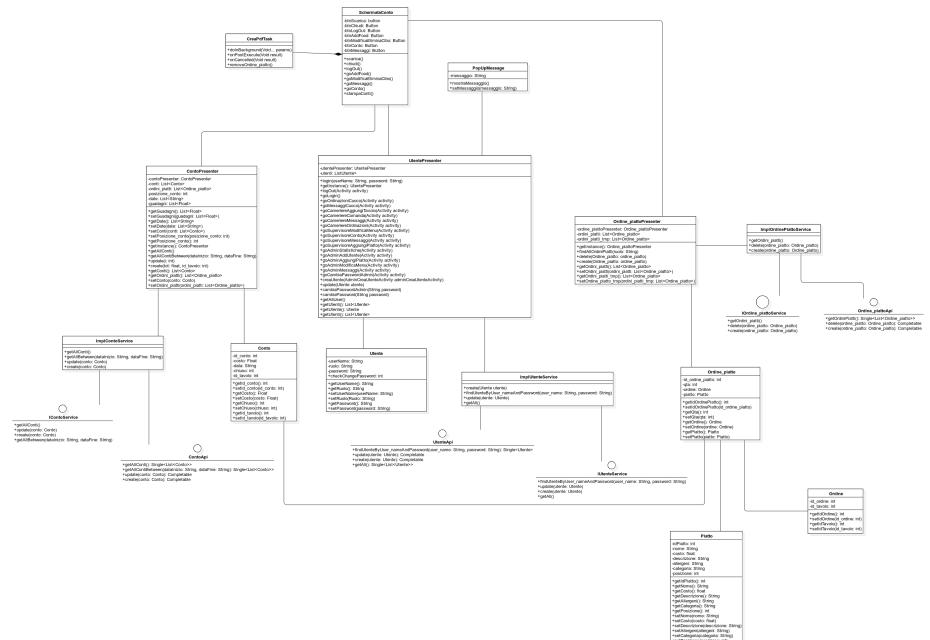
Crea piatto supervisore



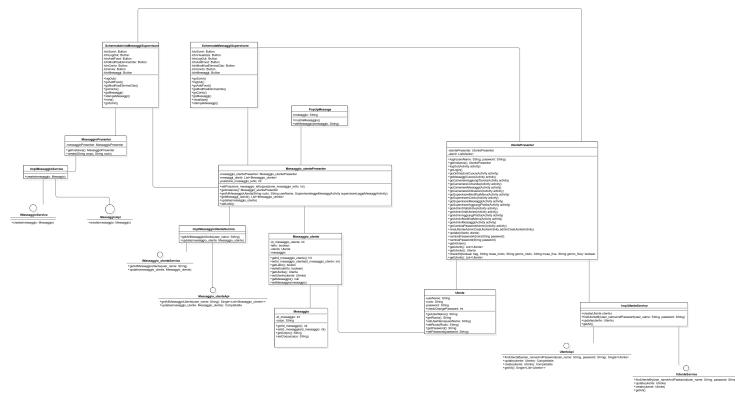
Modifica menu supervisore



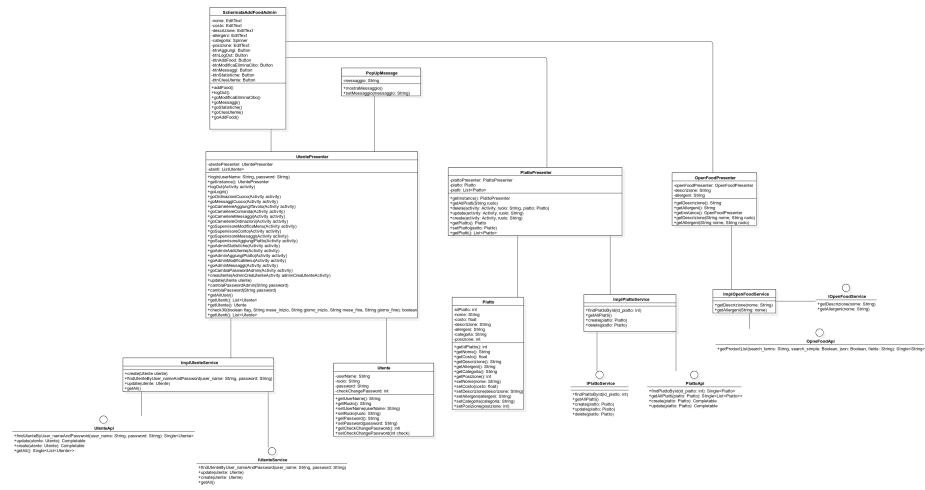
Conto supervisore



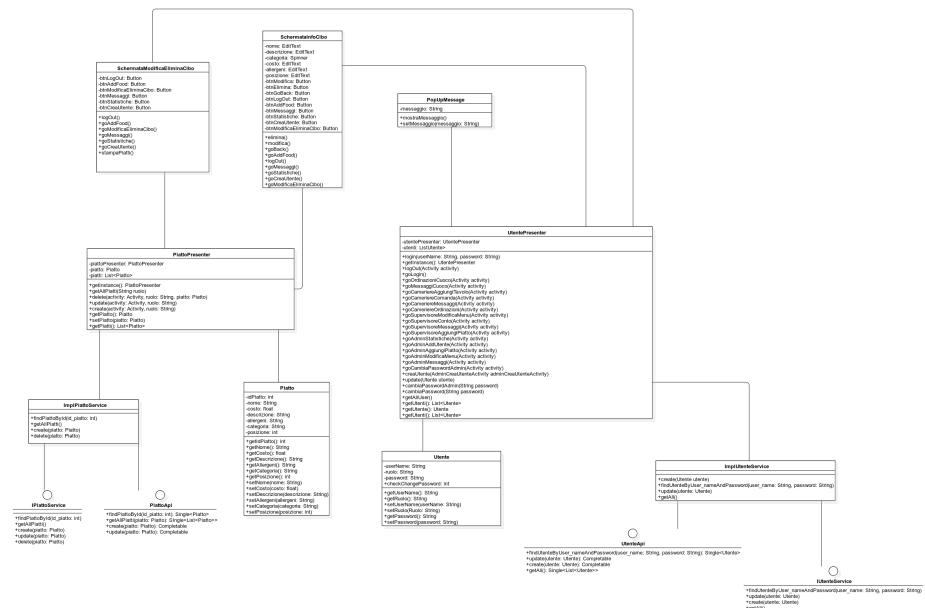
Messaggi supervisore



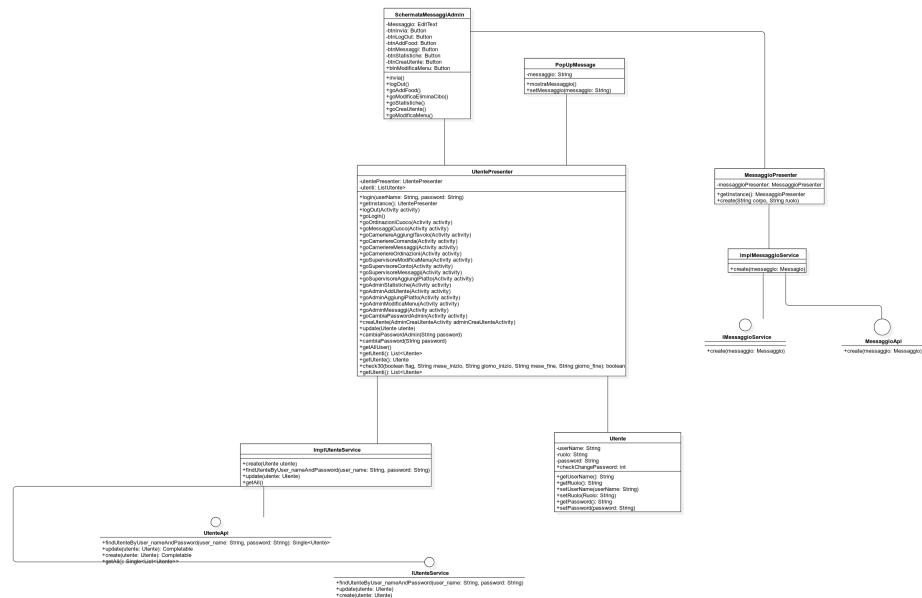
Aggiungi piatto admin



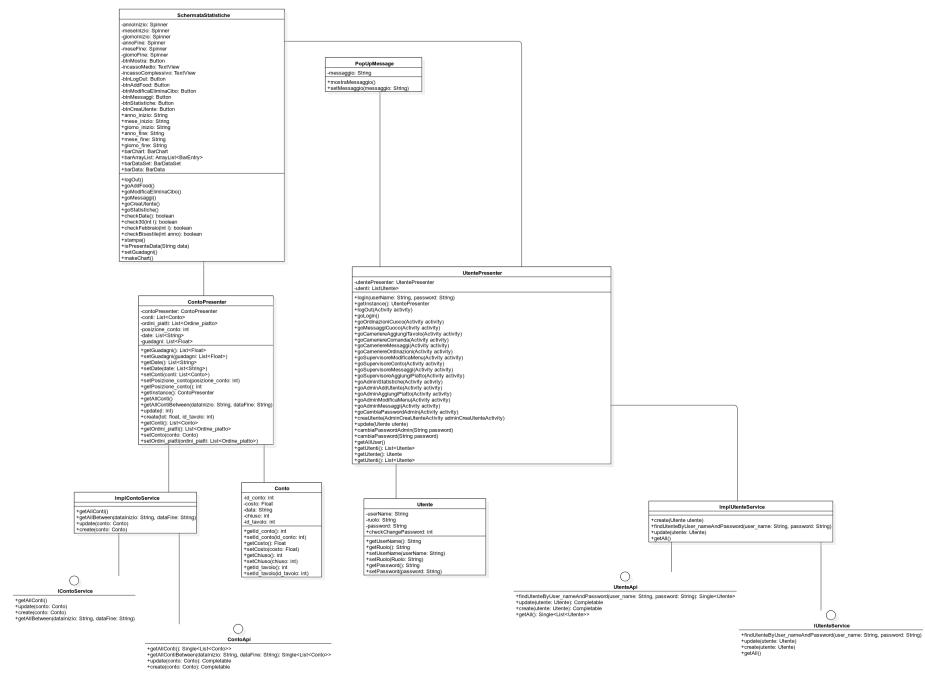
Modifica menu admin



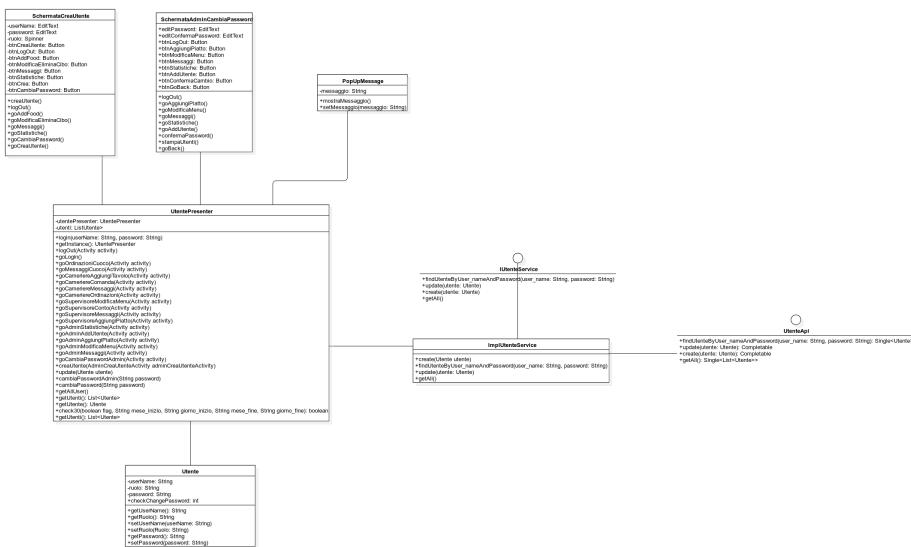
Messaggi admin



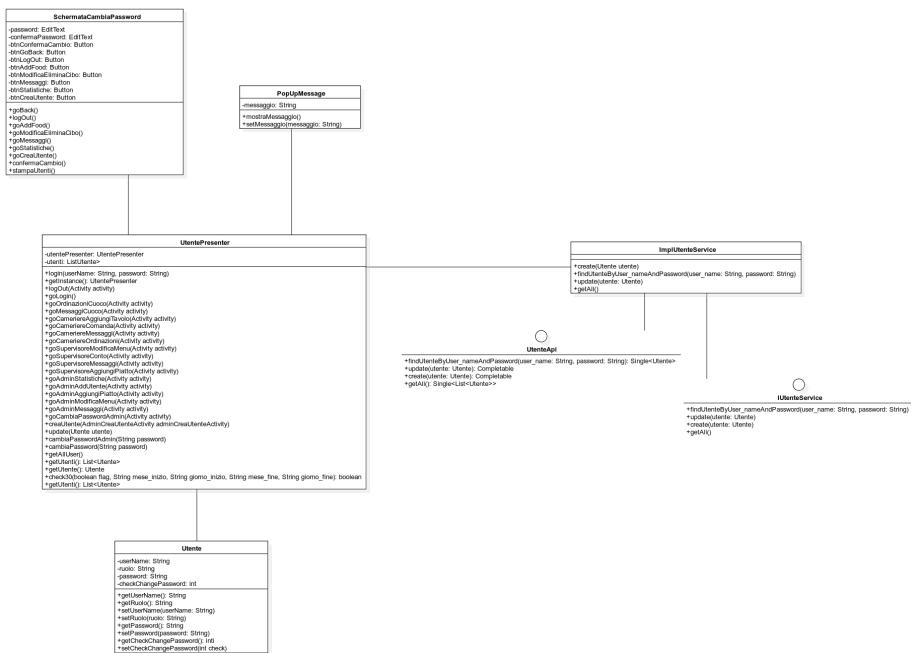
Statistische admin



Crea utente admin

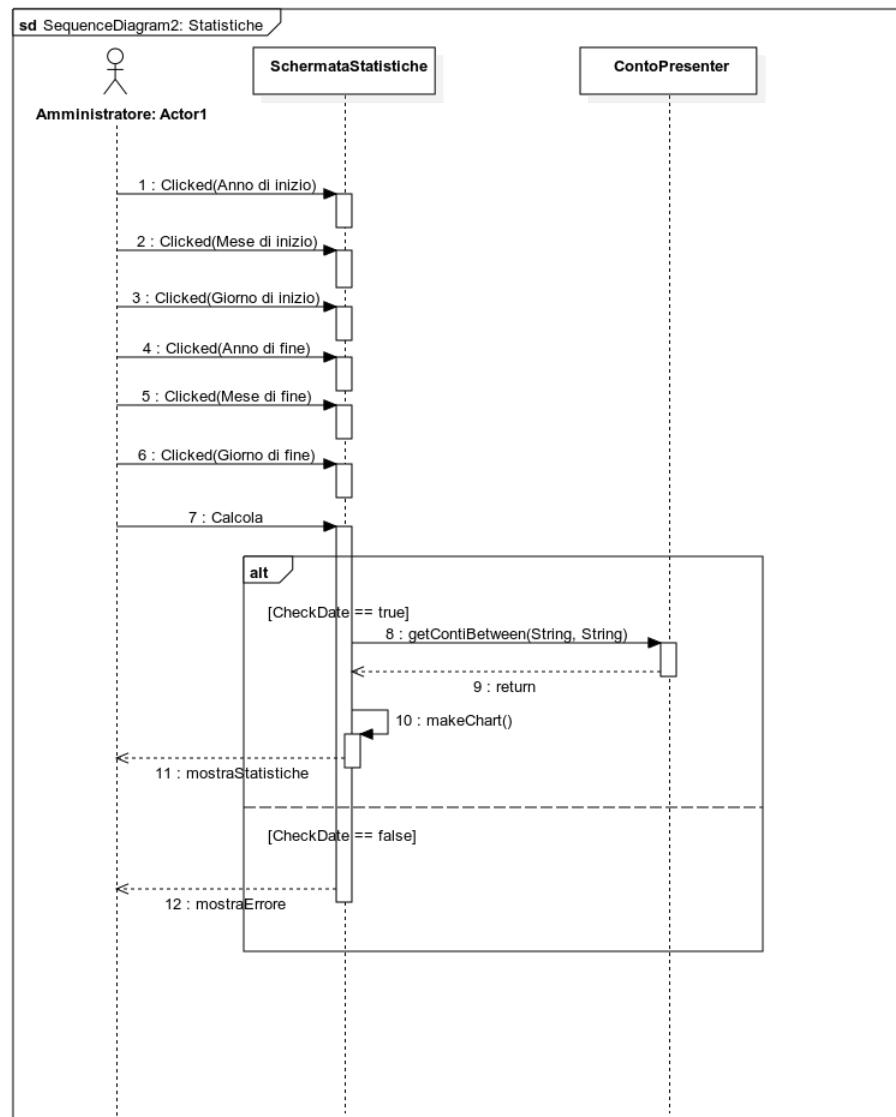


Cambia password admin

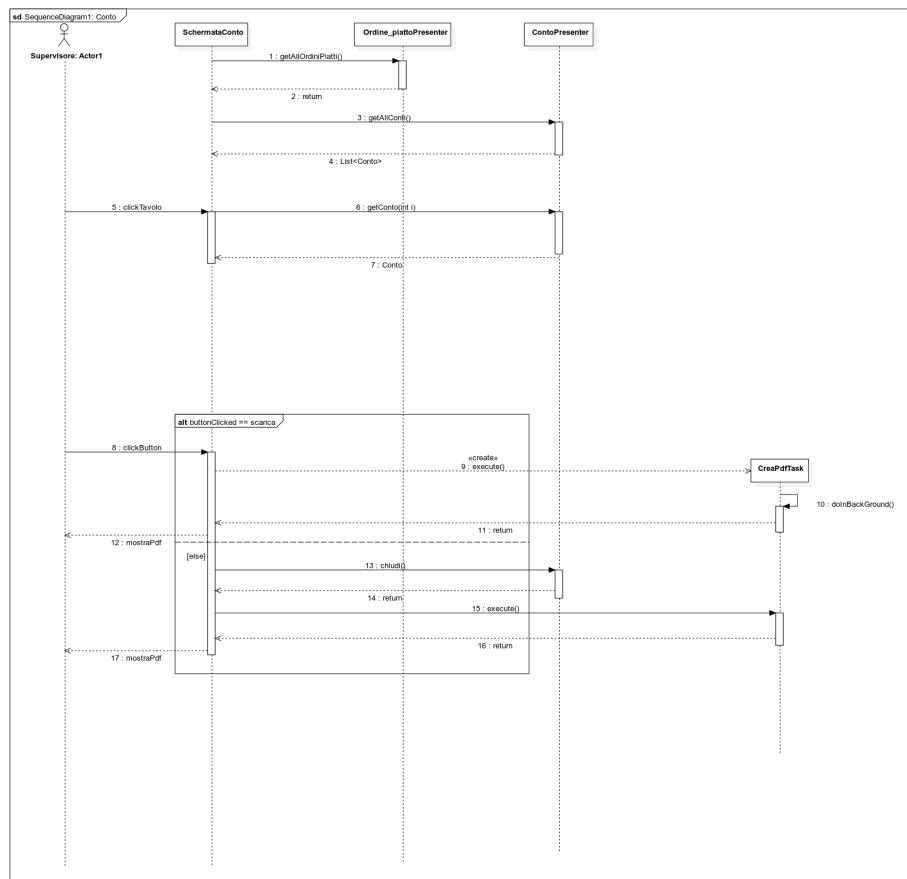


2.4 Diagrammi di sequenza di design per i casi d'uso

Statistiche



Conto



Capitolo 3

Testing e valutazione sul campo dell'usabilità

Per quanto riguarda la fase di Testing del sistema Ratatouille23, abbiamo scelto di testare i seguenti due metodi, con strategia Black Box di tipo WECT (Weak Equivalence Class Testing).

- **checkBisestile:**

La funzione checkBisestile è una funzione Java che verifica se un dato anno è bisestile. L'anno è considerato bisestile se è divisibile per 4 ma non per 100, oppure se è divisibile per 400. La funzione restituisce true se l'anno è bisestile e false altrimenti. La sintassi della funzione è la seguente:

```
public boolean checkBisestile(int anno){  
    return (anno % 4 == 0 && anno % 100 != 0) || (anno % 400 == 0);  
}
```

- **check30:**

La funzione check30 è una funzione Java che verifica se una data inizia o finisce con il 30 del mese in mesi specifici. La funzione restituisce true se la data soddisfa la condizione e false altrimenti. La funzione accetta cinque parametri: un valore booleano flag2 che indica se la data da verificare è quella di inizio o di fine, e quattro stringhe rappresentanti il mese e il giorno di inizio e fine.

La sintassi della funzione è la seguente:

```
public boolean check30(boolean flag2 , String mese_inizio , String giorno_iniz  
boolean flag = true;  
if(flag2){  
    if(mese_inizio.equals("11") || mese_inizio.equals("04") ||  
    mese_inizio.equals("06") || mese_inizio.equals("09")) {
```

```

        if (giorno_inizio.equals("31")) flag = false;
    }
} else {
    if (mese_FINE.equals("11") ||
    mese_FINE.equals("04") ||
    mese_FINE.equals("06") ||
    mese_FINE.equals("09")) {
        if (giorno_FINE.equals("31")) flag = false;
    }
}
return flag;
}

```

3.0.1 Descrizione delle Strategie Adottate per la Progettazione dei Test

Dopo una attenta analisi, abbiamo individuato le Classi di Equivalenza per ogni classe Test.

- checkBisestile: Avendo come parametro di input solo un numero di tipo intero, che corrisponde a un anno, il nostro dominio da considerare è da [0 a +MaxInt].
Non considero tutti i numeri negativi perchè siccome l'anno viene selezionato da input, nel db i numeri negativi non li ha. Quindi i nostri Campi di Esistenza sono:

- CE1 : Anno divisibile per 4
- CE2 : Anno non divisibile per 100
- CE3 : Anno divisibile per 400

Il codice è il seguente:

```

private ContoPresenter contoPresenter = ContoPresenter.getInstance();

@Test
public void testAnnoNonBisestile() {
    assertFalse(contoPresenter.checkBisestile(2001));
}

@Test
public void testAnnoBisestileDivisibilePer4() {
    assertTrue(contoPresenter.checkBisestile(2020));
}

```

```

    @Test
    public void testAnnoNonBisestileDivisibilePer100() {
        assertFalse(contoPresenter.checkBisestile(1900));
    }

    @Test
    public void testAnnoBisestileDivisibilePer400() {
        assertTrue(contoPresenter.checkBisestile(2000));
    }

```

- check30: Come parametro di input abbiamo un boolean che ci serve a considerare se prendiamo in esame le prime due stringhe o le ultime 2. Quindi i nostri Campi di Esistenza sono:

- Mesi d'inizio e di fine e sono:
 - * CE1 : [1-12]
- Giorni d'inizio e di fine e sono:
 - * CE2 : [1-30]
 - * CE3 : [31]

Il codice è il seguente:

```

private UtentePresenter utentePresenter = UtentePresenter.getInstance();

@Test
public void Mese_di_inizio_e_giorno_di_inizio_rispettano_il_requisito(){
    boolean result1 = utentePresenter.check30(true, "11", "30", "12", "31");
    assertTrue(result1);
}

@Test
public void Mese_di_fine_e_giorno_di_fine_rispettano_il_requisito(){
    boolean result2 = utentePresenter.check30(false, "10", "15", "11", "30");
    assertTrue(result2);
}

@Test
public void Mese_di_inizio_rispetta_il_requisito_ma_il_giorno_di_inizio_no(){
    boolean result3 = utentePresenter.check30(true, "06", "31", "07", "31");
    assertFalse(result3);
}

```

```
@Test
public void Mese_di_fine_rispetta_il_requisito_ma_il_giorno_di_fine_no(){
    boolean result4 = utentePresenter.check30(false, "09", "01", "09", "31");
    assertFalse(result4);
}
```

3.0.2 Analisi

Soggetti coinvolti:

I soggetti che hanno provato la nostra app sono i seguenti:



Victor Oshimen

Età: 22

Professione: Supervisore



Zambo Anguissa

Età: 20

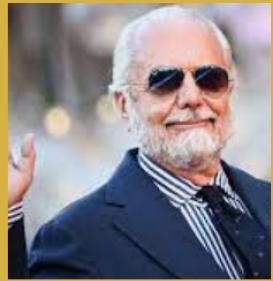
Professione: Cuoco



Tanguy Ndombele

Età: 26

Professione: Cameriere



Aurelio De Laurentiis

Età: 55

Professione: Amministratore

3.0.3 Analytics

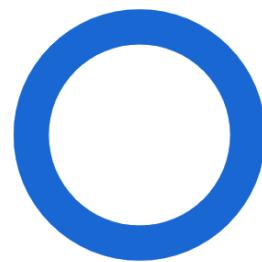
Tramite **Firebase Analytics** abbiamo potuto monitorare l'utilizzo dell'applicazione, di seguito mostriamo i dati raccolti.

Persone coinvolte:



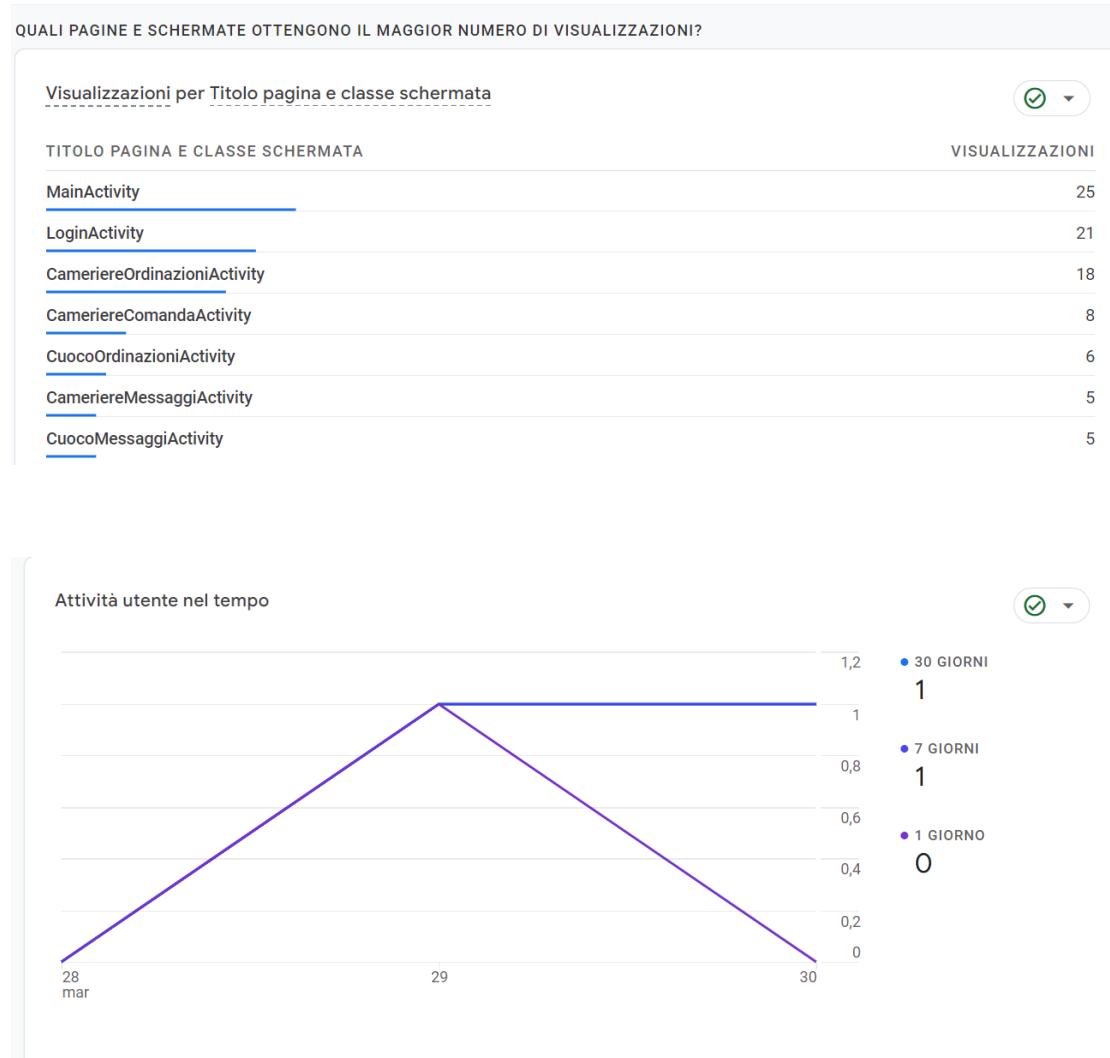
QUALI SONO I DATI DELLE ATTIVITÀ PER LE DIVERSE PIATTAFORME?

Conversioni ▾ per Piattaforma



● ANDROID
100,0%

Schermate:



Tempo di utilizzo:

Titolo pagina e...asse schermata	Visualizzazioni per utente	Durata media del coinvolgimento	Conteggio eventi
	105,00 Uguale alla media	56 m 31 s Uguale alla media	Tutti gli eventi 107 100% del totale
11 CameriereMessaggiActivity	5,00	0 m 41 s	5
12 CuocoMessaggiActivity	5,00	11 m 03 s	5
13 CuocoOrdinazioniActivity	6,00	0 m 29 s	6
14 CameriereComandaActivity	8,00	1 m 02 s	8
15 CameriereOrdinazioniActivity	18,00	4 m 02 s	18
16 LoginActivity	21,00	2 m 34 s	21
17 MainActivity	25,00	33 m 26 s	25