

Elaborato ISPW 2024/2025

DI LUCA SABATINI & SMITH FIDEL VALENZUELA SANTANA

Nome del progetto: **MEAL CALENDAR**

Indice

1	Introduzione	2
1.1	Descrizione del sistema	2
1.2	Requisiti software e hardware specifici	2
1.3	User Stories	2
1.3.1	US 1:	2
1.3.2	US 2:	2
1.3.3	US 3:	2
1.3.4	US 4:	2
1.3.5	US 5:	3
1.3.6	US 6:	3
1.4	Funcional requirements	3
1.4.1	FR 1:	3
1.4.2	FR 2:	3
1.4.3	FR 3:	3
1.4.4	FR 4:	3
1.4.5	FR 5:	3
1.4.6	FR 6:	3
1.4.7	FR Termini aggiuntivi (*):	3
2	Internal steps del sistema	3
2.1	Internal steps: Choose restaurant:	3
2.2	Internal steps: Set eating time	4
3	Interfaccia grafica	5
3.1	Schermata di registrazione:	5
3.2	Schermata di login:	5
3.3	Schermata del menu principale:	6
3.4	Schermata del carrello:	6
3.5	Schermata del pagamento:	7
3.6	Schermata delle mappe:	8
4	Diagrammi UML del sistema	9
4.1	Use Case Diagram:	9
4.2	Activity Diagram	10
4.2.1	AD: Choose restaurant	10
4.2.2	Ad: Find recipe	11
4.3	State diagrams	11
4.3.1	State diagram: Choose restaurant	11
4.3.2	State diagram: Find recipe	12
4.4	Sequence diagram	12
4.4.1	Sequence diagram: Find recipe	12

4.4.2 Sequence diagram: Choose restaurant	13
4.5 Diagrammi di classe	14
4.5.1 Diagramma BCE, use case: Choose restaurant	14
4.5.2 Diagramma BCE, use case: Find recipe	14
4.6 MVC: Choose restaurant	15
4.6.1 MVC: Find recipe	15

1 Introduzione

1.1 Descrizione del sistema

Il sistema è un'applicazione che fornisce all'utente vari servizi differenti. Il primo servizio consiste nel consigliare all'utente ristoranti e/o locali nelle sue vicinanze, filtrando risultati in base alle fasce orarie da lui scelte e al tipo di dieta che egli segue (vegana, vegetariana oppure onnivora). Il secondo servizio consente all'utente di memorizzare delle ricette che poi possono essere proposte ad altri utenti creando così un sottosistema di condivisione delle ricette, il sistema consente anche di avere una dispensa digitale (il frigorifero) in cui l'utente inserisce tutti gli ingredienti che ha a sua disposizione in casa. Per finire il sistema memorizza quando l'utente pianifica un pasto e successivamente invia una email come remainder contenente tutti i dati relativi a quel pasto specifico come luogo, tipologia e orario.

1.2 Requisiti software e hardware specifici

Il sistema per funzionare necessita di una java virtual machine, un DBMS SQL su cui può effettuare operazioni di lettura e scrittura dei dati, almeno 2GB di spazio su disco fisso e almeno 1GB di RAM disponibile.

1.3 User Stories

1.3.1 US 1:

In quanto utente che lavora spesso in smart working, voglio una schermata che mi consenta di trovare ricette, così che abbia sempre in mente che cosa mangiare.

1.3.2 US 2:

In quanto utente che viaggia spesso per lavoro, voglio un calendario che in automatico mi dica dove devo mangiare, in modo tale da non pensare di dovermi organizzare dopo.

1.3.3 US 3:

In quanto utente vegano, voglio che il sistema filtri i ristoranti dove è possibile mangiare cibo vegano, così da permettermi di rispettare la mia dieta.

1.3.4 US 4:

In quanto utente, voglio che il sistema mi offra un inventario in cui possa aggiungere e rimuovere i prodotti che ho in casa, in modo da poter sapere cosa ho in frigo anche quando non sono a casa.

1.3.5 US 5:

In quanto utente, voglio che il sistema mi dia la possibilità di modificare le mie ricette, così da tenerle sempre aggiornate.

1.3.6 US 6:

In quanto utente a dieta, voglio che il sistema mi mostri la quantità di calorie totali e gli ingredienti usati nelle ricette disponibili, così da non doverli calcolare da solo.

1.4 Funcional requirements

1.4.1 FR 1:

Il sistema deve offrire due filtri* per filtrare le ricette che l'utente vuole.

1.4.2 FR 2:

Il sistema deve avvisare l'utente trenta minuti prima dell'ora scelta da lui nel calendario, attraverso una mail in cui gli ricorda dove ha scelto di mangiare.

1.4.3 FR 3:

Il sistema deve offrire un box di testo in cui inserire il nome della ricetta, il tipo di pasto, il tipo di dieta, il numero degli ingredienti, gli ingredienti e la descrizione*.

1.4.4 FR 4:

Il sistema deve avvisare l'utente con una mail quando ha programmato il pasto con il calendario indicando luogo e orario.

1.4.5 FR 5:

Il sistema deve aprire un browser ogni volta che l'utente clicca due volte sul ristorante selezionato.

1.4.6 FR 6:

Il sistema deve mostrare all'utente una lista di 20 ristoranti che soddisfano i suoi filtri di ricerca.

1.4.7 FR Termini aggiuntivi (*):

- *descrizione: descrizione della ricetta e le sue calorie
- *filtri: tipo di dieta e pasto

2 Internal steps del sistema

2.1 Internal steps: Choose restaurant:

1. Il sistema chiede all'utente di inserire tipo di dieta, tipo di pasto, e raggio in cui si vuole cercare il locale.

2. L'utente seglie i tre filtri in base alle sue preferenze
3. L'utente conferma i filtri
4. Il sistema usa Google Search Engine per cercare i locali con i filtri selezionati
5. Il sistema mostra una sequenza di 20 ristoranti trovati
6. L'utente seleziona il ristorante
7. Il sistema riporta l'utente nella sezione di selezione dei filtri

Extensions

- 3.a.** L'utente non ha inserito tutti i filtri: Il sistema non effettua la ricerca e mostra un messaggio di errore con scritto "riempire tutti i campi"
- 4.a.** Il sistema non trova alcun ristorante: Il sistema mostra un messaggio con scritto "nessun ristorante trovato"
- 6.a.** L'utente non seleziona alcun ristorante: Il sistema continua a mostrare tutti e 20 i ristoranti fino a una nuova ricerca

2.2 Internal steps: Set eating time

1. Il sistema chiede all'utente di inserire luogo, ora e data
2. L'utente inserisce il luogo, l'ora e il giorno in cui vuole mangiare
3. Il sistema carica la sezione di scelta dei ristoranti
4. L'utente inserisce i filtri
5. Il sistema mostra i 20 ristoranti disponibili
6. L'utente seleziona il ristorante
7. Il sistema carica la sezione del calendario
8. L'utente conferma la data tramite il calendario
9. Il sistema ottiene l'indirizzo e-mail dell'utente
10. L'utente conferma di voler ricevere la e-mail
11. Il sistema invia la e-mail
12. L'utente riceve la e-mail

Extensions

- 2.a.** L'utente sceglie "casa" come primo filtro invece di "ristorante": Il sistema carica la sezione delle ricette


- 4.a. L'utente non ha inserito tutti i filtri: Il sistema mostra un messaggio di errore all'utente
- 9.a. L'utente non si è registrato con un indirizzo e-mail valido: Il sistema non invia le e-mail

3 Interfaccia grafica

3.1 Schermata di registrazione:

Register Page

Welcome!



Username

Email

Password

☒ Show password

Register


login

click [here](#) to continue as a guest

3.2 Schermata di login:

Register Page

Welcome!



Email

Password

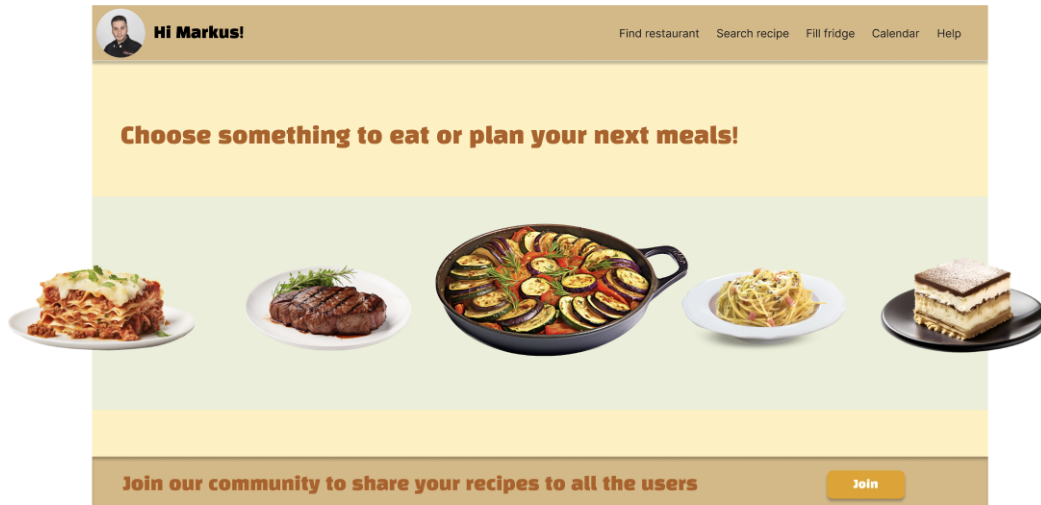
☒ Show password

Log in

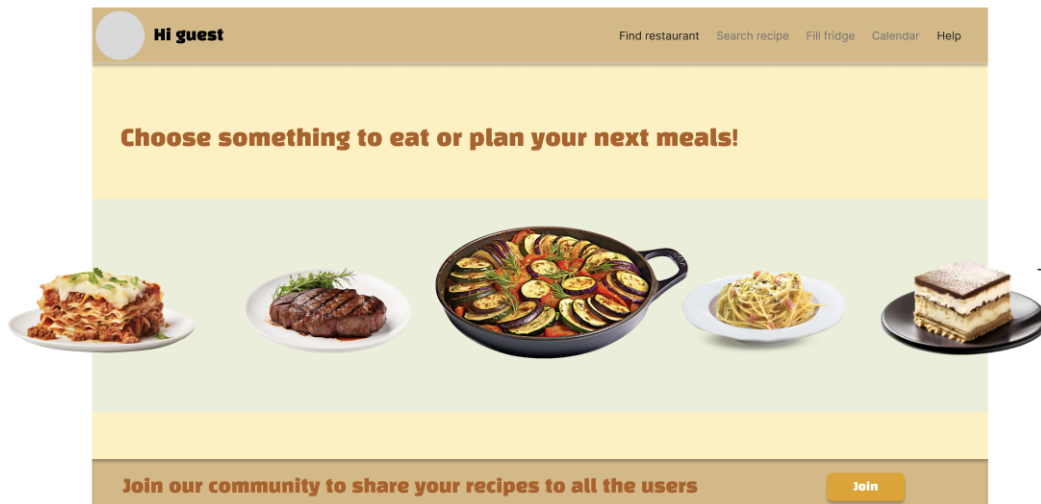
click [here](#) to continue as a guest

3.3 Schermata del menu principale:

Home page User

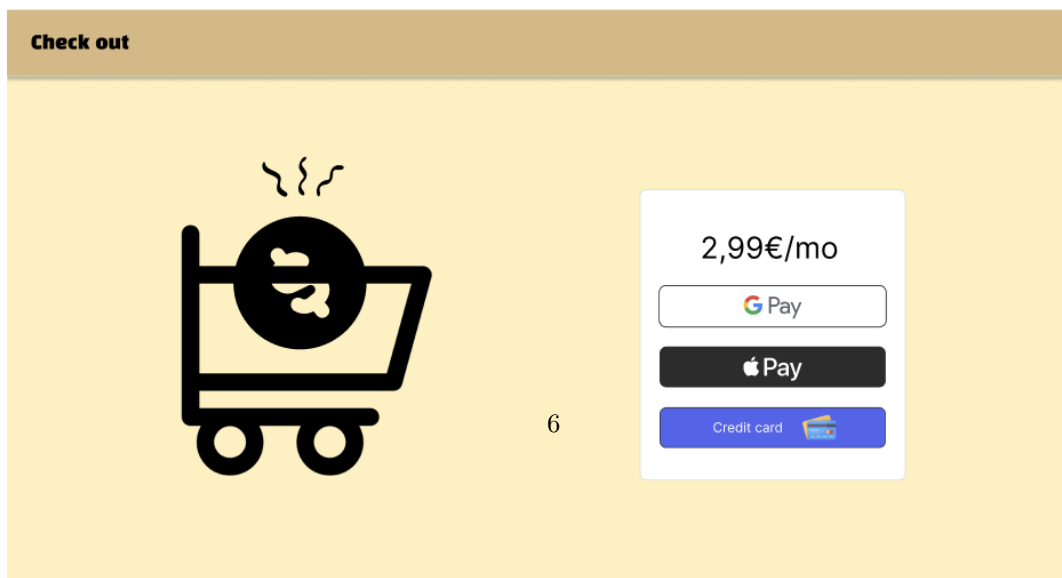


Home page guest



3.4 Schermata del carrello:


Payment page



3.5 Schermata del pagamento:

Payment page

Pay with credit card

A stylized black and white illustration of a person with curly hair, wearing a tank top and pants, jumping over a large credit card. The person is holding a smartphone in their right hand. The credit card has a black stripe and a small chip in the bottom right corner.

Credit card number

12 Numbers

Expiration date

MM/YY

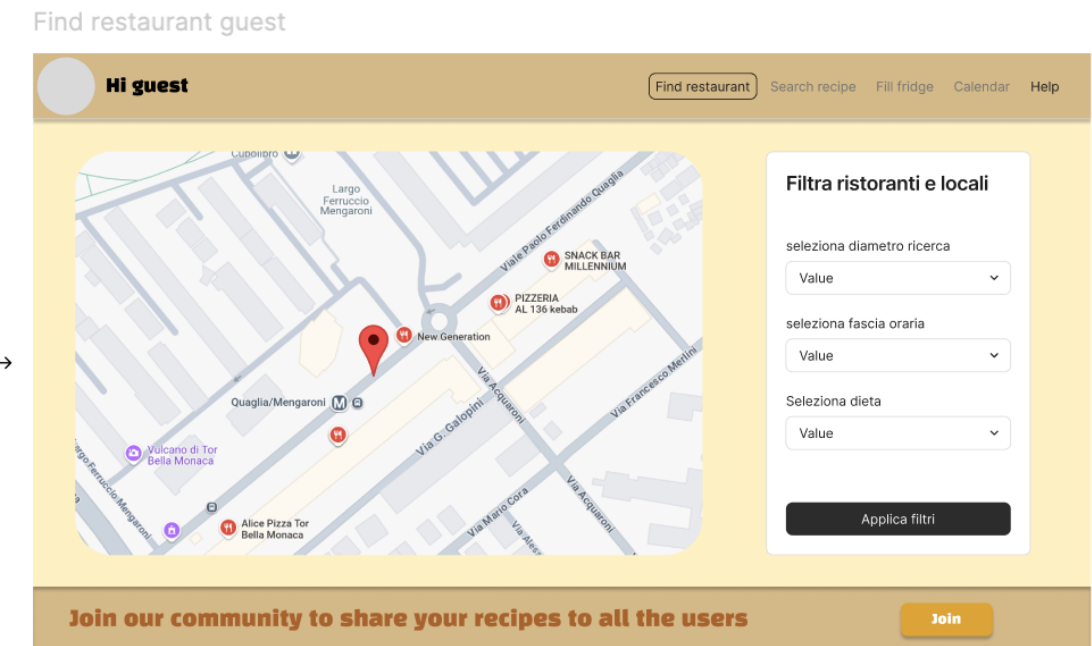
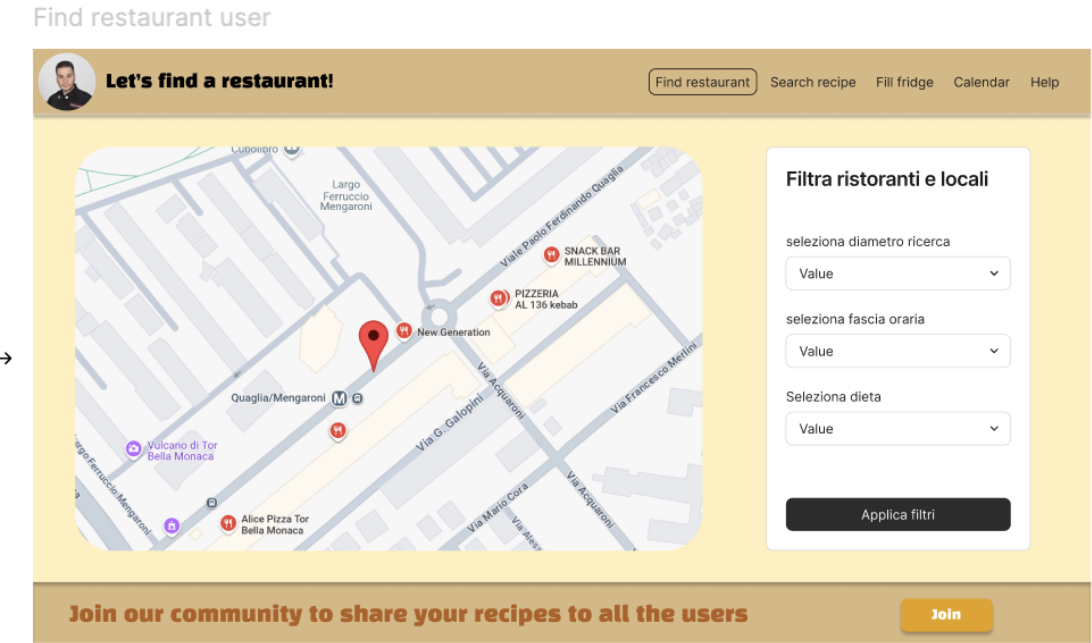
CVV

3 Numbers behind card

2,99€/mo

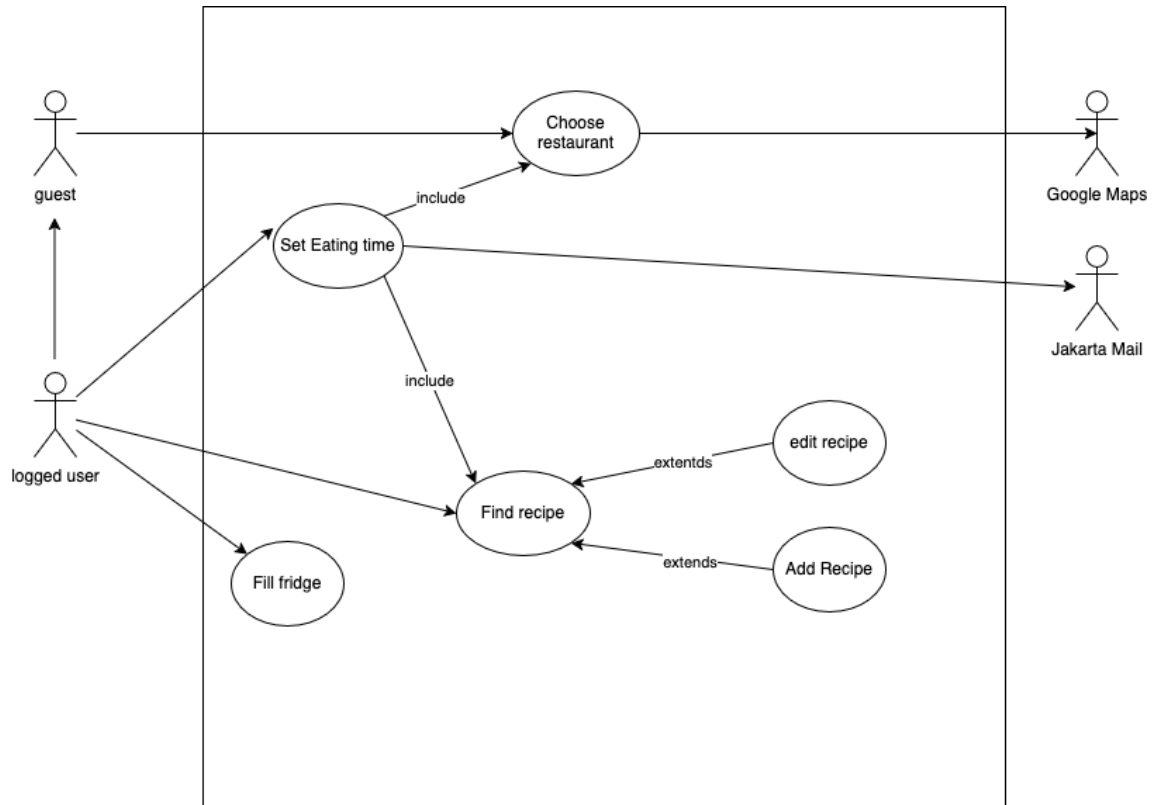
Pay

3.6 Schermata delle mappe:



4 Diagrammi UML del sistema

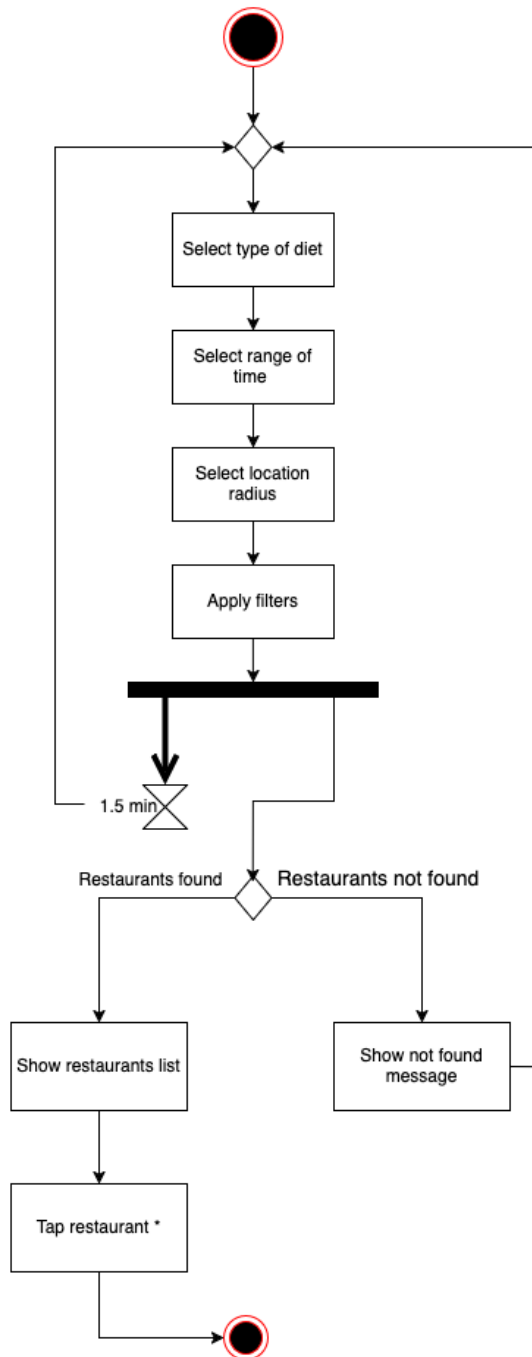
4.1 Use Case Diagram:



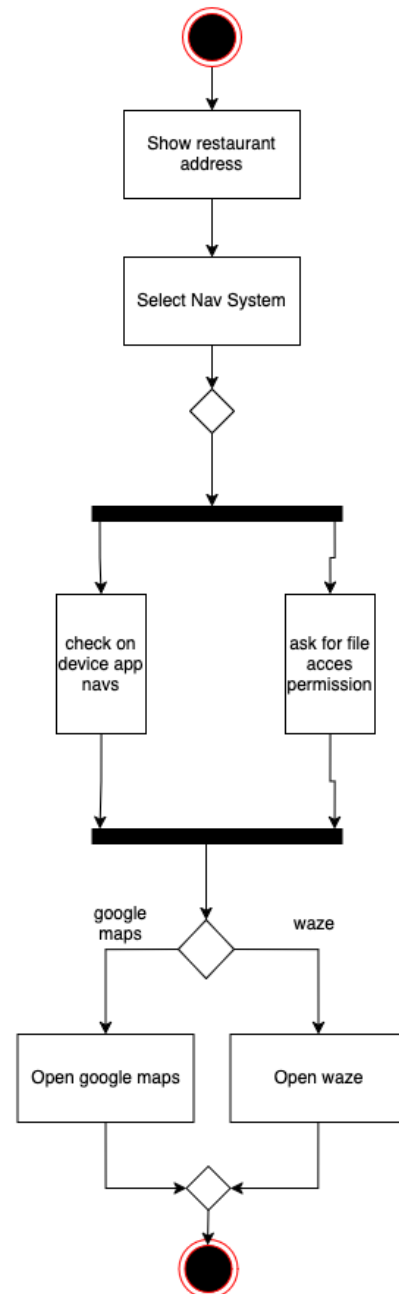
4.2 Activity Diagram

4.2.1 AD: Choose restaurant

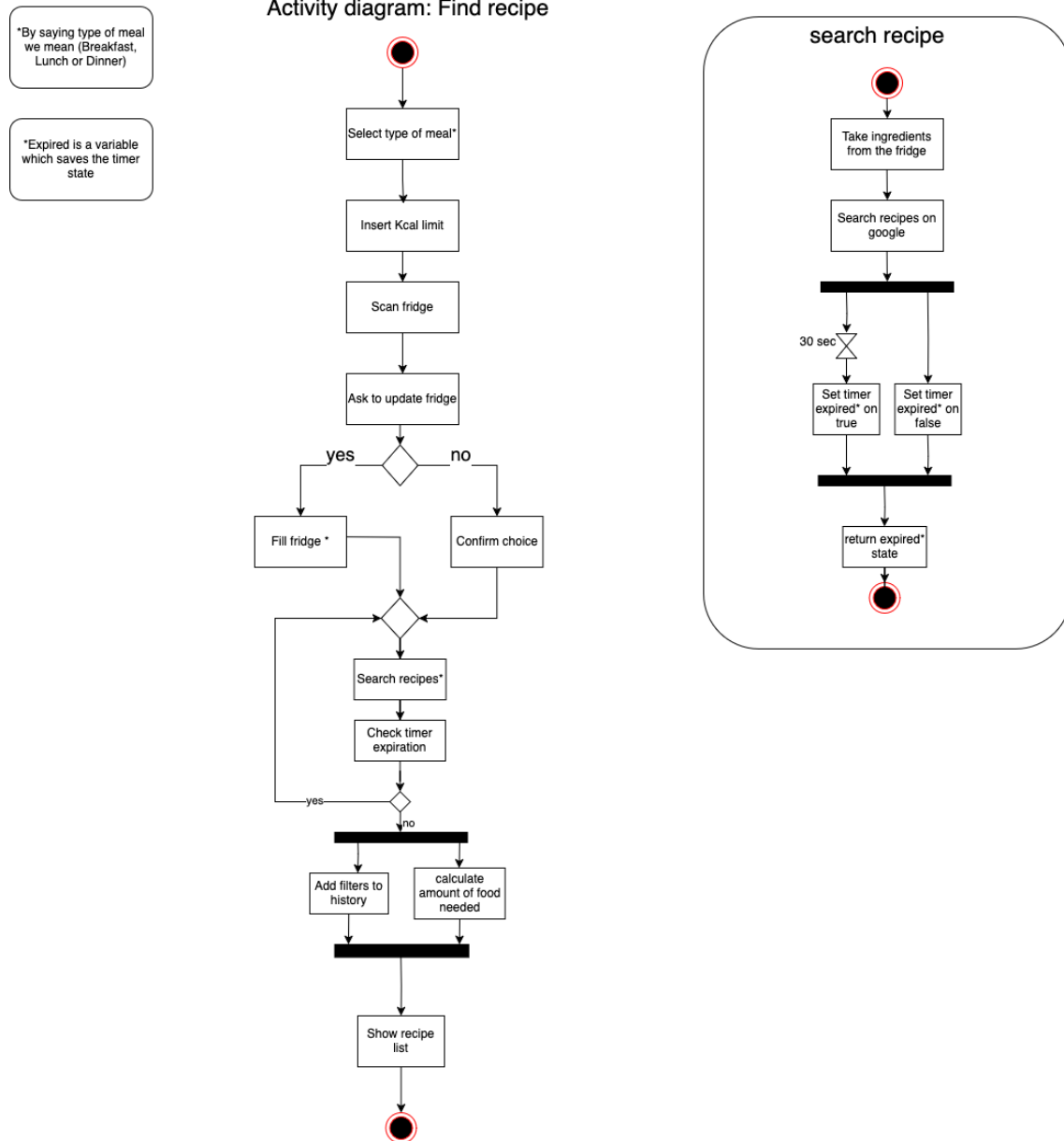
Activity diagram: Choose restaurant



Tap restaurant

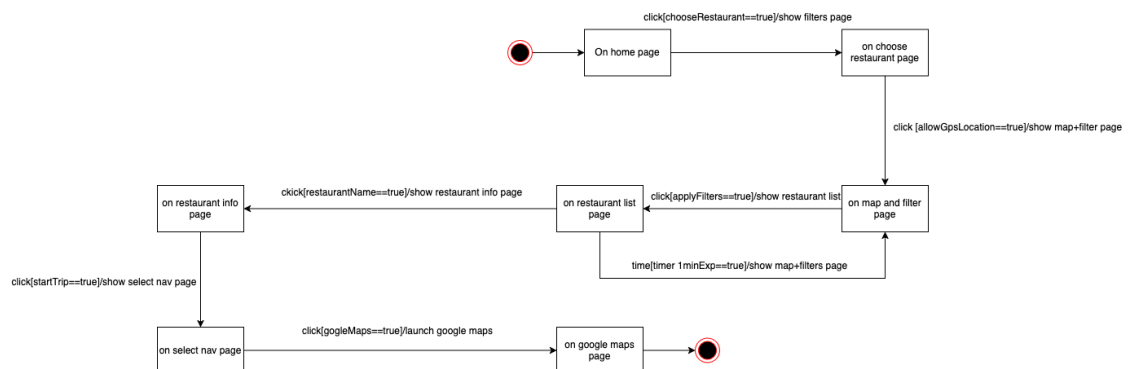


4.2.2 Ad: Find recipe

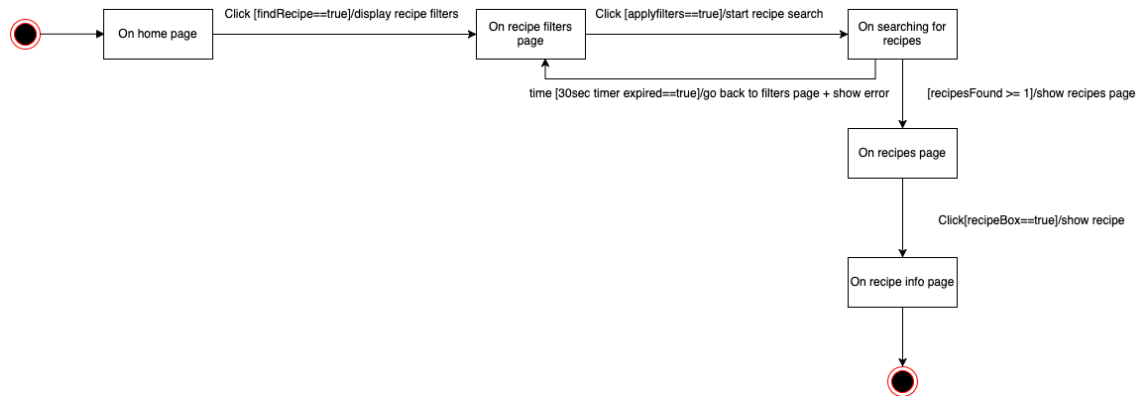


4.3 State diagrams

4.3.1 State diagram: Choose restaurant

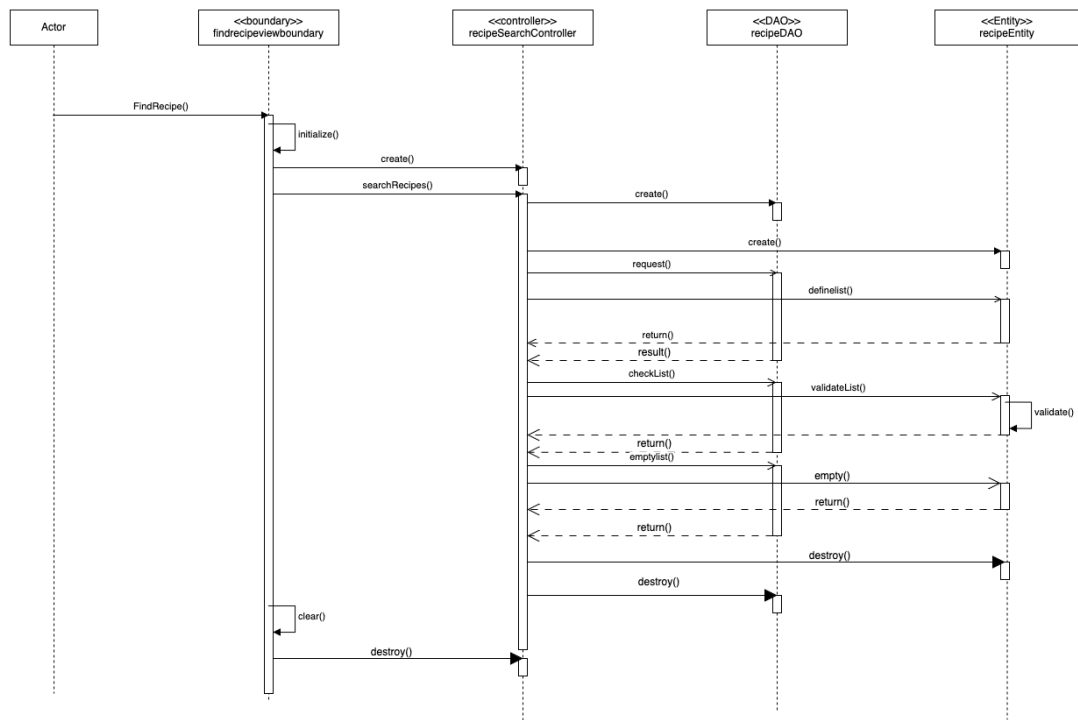


4.3.2 State diagram: Find recipe

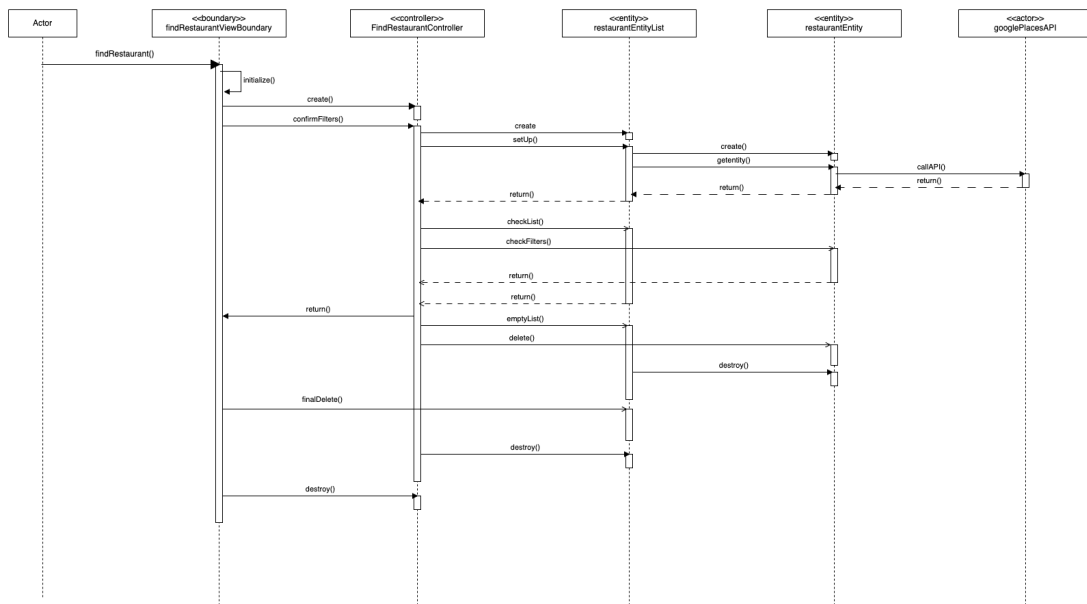


4.4 Sequence diagram

4.4.1 Sequence diagram: Find recipe

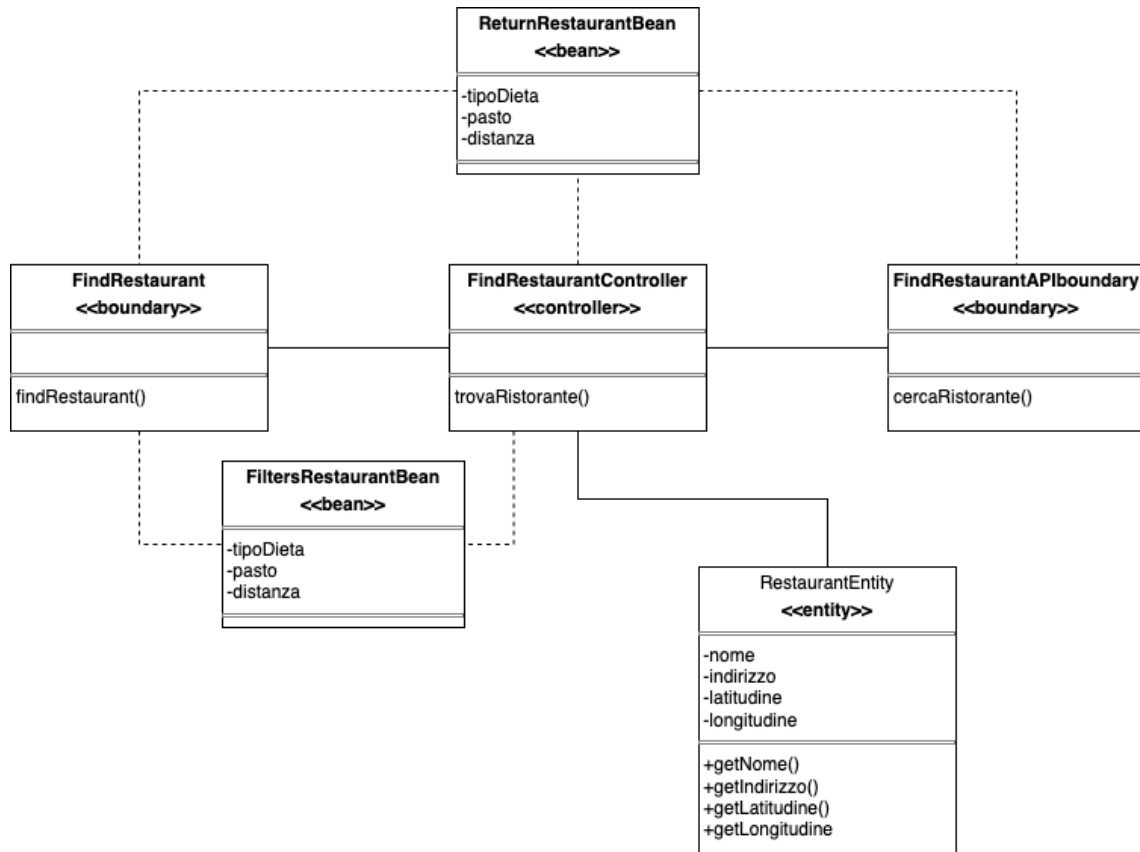


4.4.2 Sequence diagram: Choose restaurant

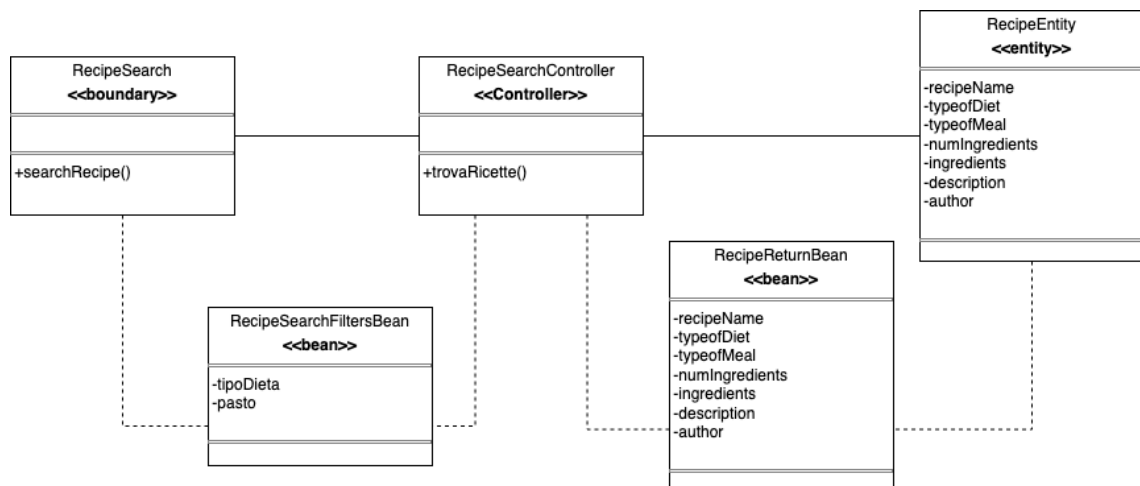


4.5 Diagrammi di classe

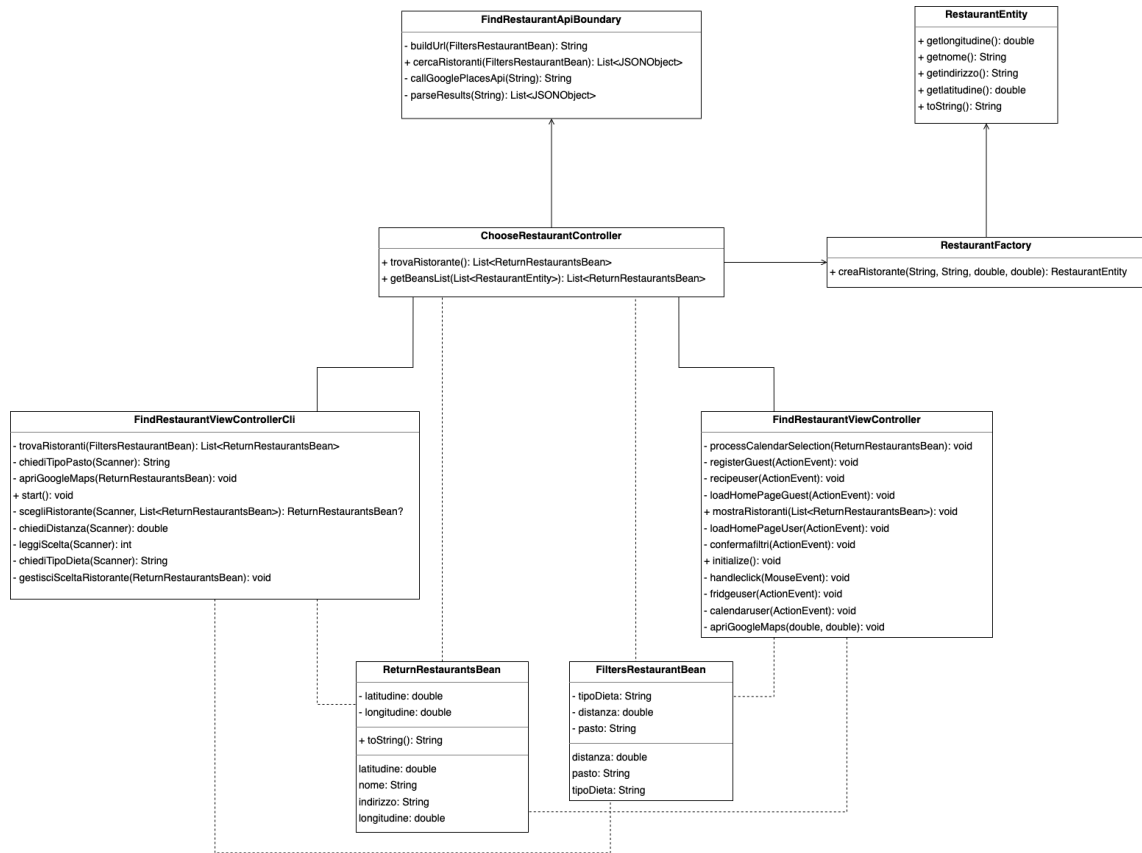
4.5.1 Diagramma BCE, use case: Choose restaurant



4.5.2 Diagramma BCE, use case: Find recipe



4.6 MVC: Choose restaurant



4.6.1 MVC: Find recipe

