

# Elaborato di ISPW

Luca Sabatini & Smith Valenzuela

## 1) Requisiti software e hardware specifici:

L'applicazione per girare necessita una Java Virtual Machine, aver installato sul proprio dispositivo un data base SQL .Il dispositivo de avere almeno 2GB liberi sul disco fisso e almeno un 1 giga di RAM .

### 1.1) Introduzione

Il sistema è un'applicazione che fornisce all'utente vari servizi differenti. Il primo servizio consiste nel consigliare all'utente una serie di ristoranti e/o locali nelle sue vicinanze, filtrando i risultati in base alle sue fasce di orario preferite e a quale tipo di dieta segue (vegana, vegetariana o onnivora). Il secondo servizio consente all'utente di memorizzare delle ricette che poi possono essere proposte ad altri utenti creando così un sistema di condivisione delle ricette. Il sistema consente anche di avere una dispensa digitale (il frigorifero) che raccoglie tutti gli ingredienti che l'utente potrebbe avere in casa. Come ultimo servizio il sistema consente di memorizzare quando un utente ha intenzione di mangiare a casa o ad un eventuale locale inviando una e-mail come remainder.

### 1.2) User stories:

#### 1.2.1) US-1:

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

#### 1.2.2) 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.

#### 1.2.3) 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.

#### 1.2.4) 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.

#### 1.2.5) US-5

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

#### 1.2.5) 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.

### 1.3) Functional requirements

#### 1.3.1) FR-1:

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

#### 1.3.2) 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.

#### 1.3.3) 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\*.

#### 1.3.4) FR-4

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

#### 1.3.5) FR-5

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

#### 1.3.6) FR-6

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

\*descrizione=descrizione + calorie della ricetta, \*filtri=tipo dieta e pasto.

## 2) Internal steps :Choose Restaurant:

- 1.il sistema mostra un'schermata con tre spazi in cui si deve inserire il tipo di dieta , il tipo di pasto e il raggio in cui si vuole cercare
- 2.l'utente sceglie con le sue preferenze i tre filtri
- 3.l'utente conferma i filtri
- 4.il sistema usa Google Search engine per cercare i ristoranti con i filtri scelti
- 5.il sistema mostra in una schermata di venti ristoranti trovati
6. l'utente sceglie il ristorante attraverso due click su di esso
7. il sistema carica su Google Maps l'indirizzo del ristorante scelto
8. il sistema riporta l'utente alla schermata iniziale

## Extensions

3.a. se l'utente si è dimenticato di riempire uno spazio il sistema non cerca i ristoranti ,ma mostra una scritta "Riempire tutti i campi"

4.a. se non si trova nessun ristorante il sistema mostra la scritta "Nessun ristorante trovato"

6.a. se l'utente non fa due click il sistema continua a mostrare la schermata con i ristoranti trovati finche non cambia e conferma i filtri

## 2.2) Internal steps: Set eating time:

1. L'utente visualizza una schermata con tre spazi in cui inserire il luogo, l'ora e la data
2. L'utente inserisce il luogo l'orario e il giorno in cui mangiare
3. Il sistema carica la schermata del ristorante
4. L'utente seleziona i filtri
5. Il sistema mostra i ristoranti disponibili
6. L'utente seleziona il ristorante
7. Il sistema carica la schermata del calendario
8. L'utente conferma la sua scelta
9. Il sistema ottiene la mail dell'utente
10. L'utente conferma di voler ricevere la e-mail
11. Il sistema mostra il caricamento della e-mail all'utente
12. L'utente riceve la e-mail

## Extensions

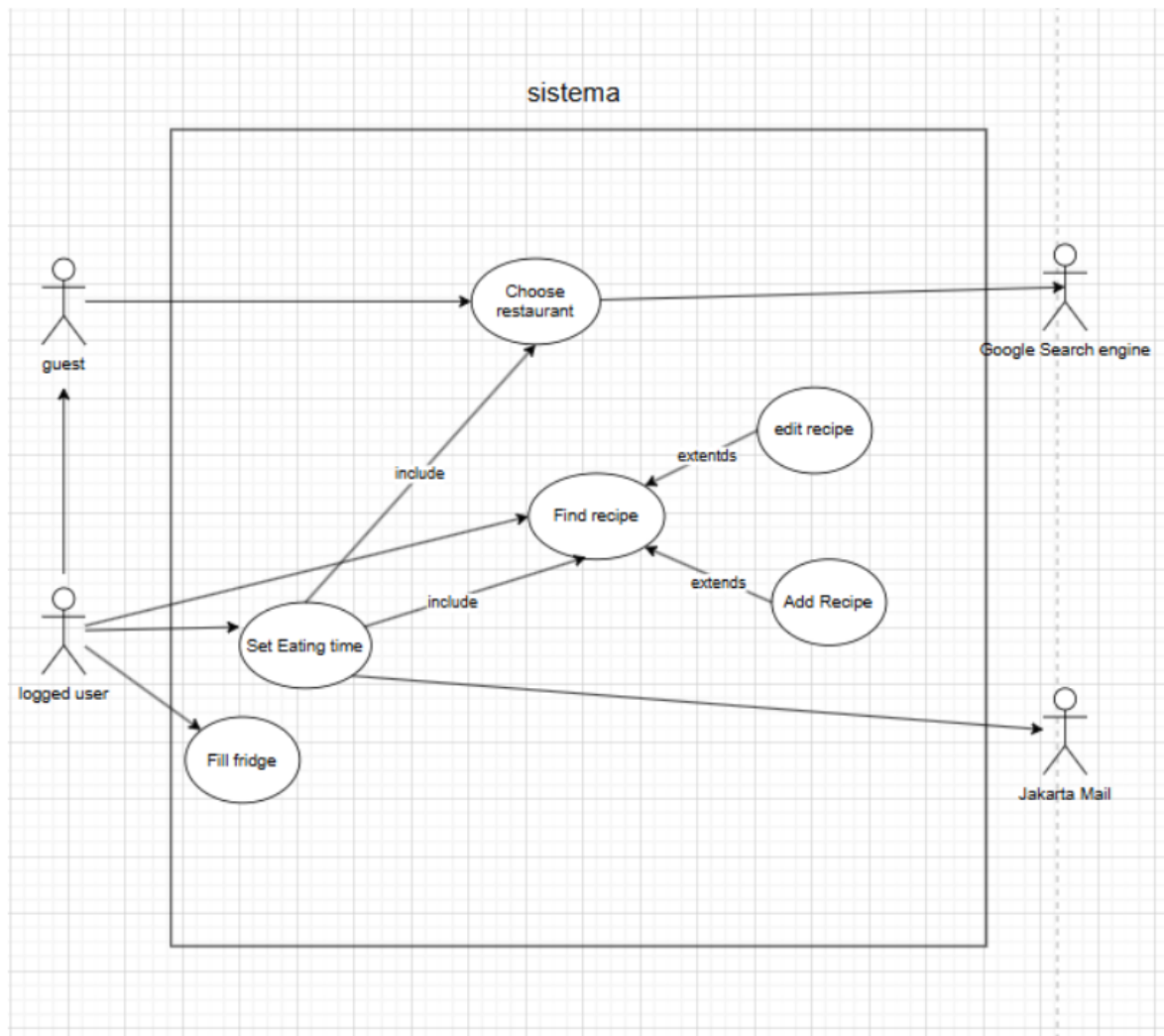
2.a. l'utente seleziona casa nello spazio del luogo invece che ristorante: il sistema carica la schermata di ricerca delle ricette.

4.a. l'utente non ha selezionato 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 non valida: il sistema non manda la e-mail

### 3) Diagrammi UML

#### 3.1) Diagrama dei casi d'uso del sistema:

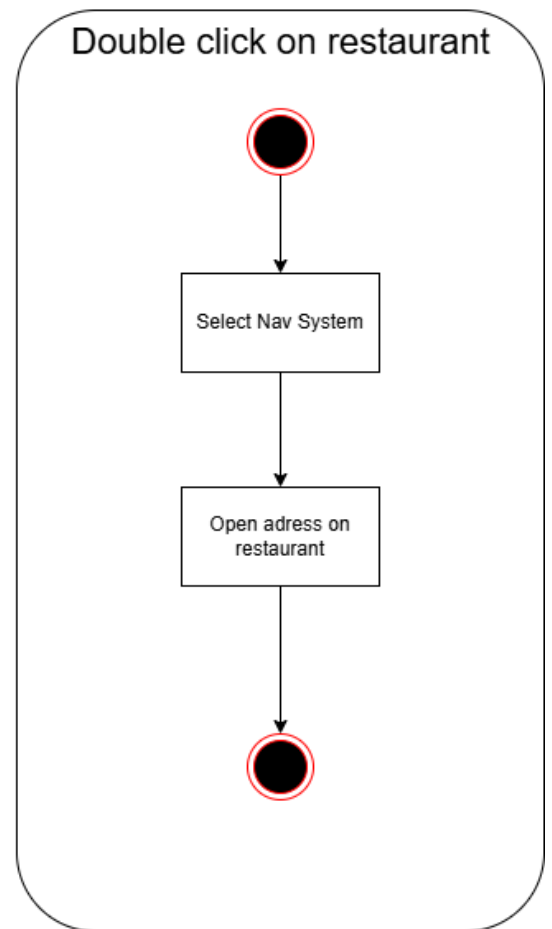
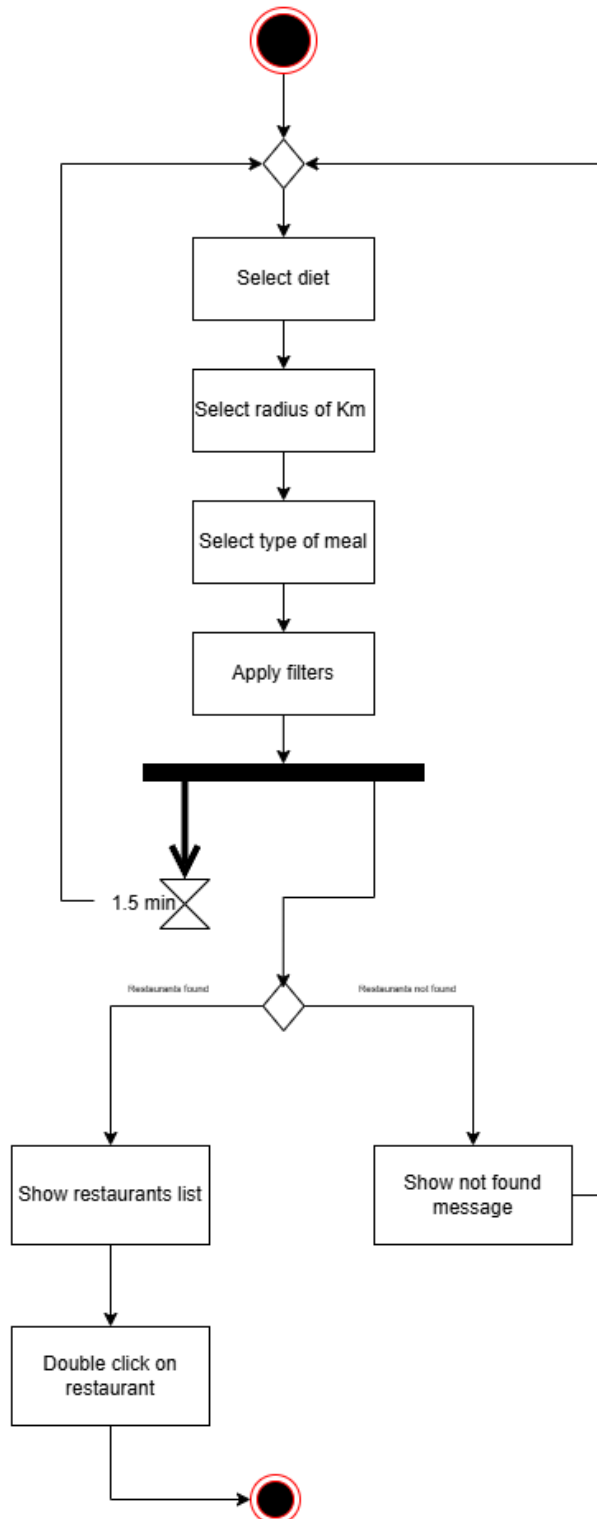






### 3.2) AD 1

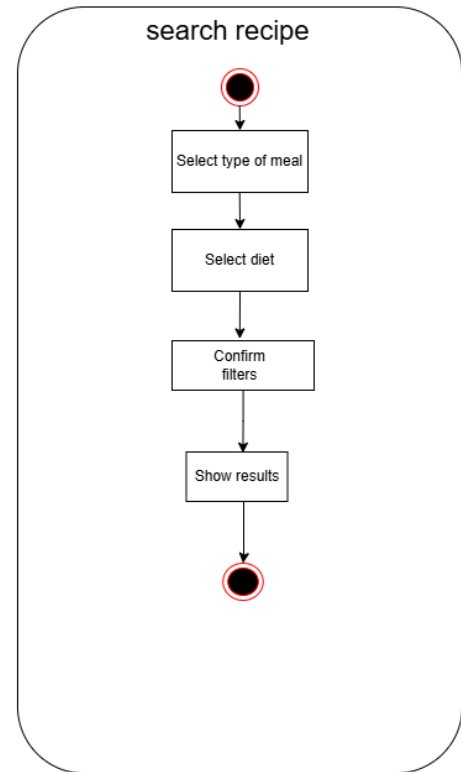
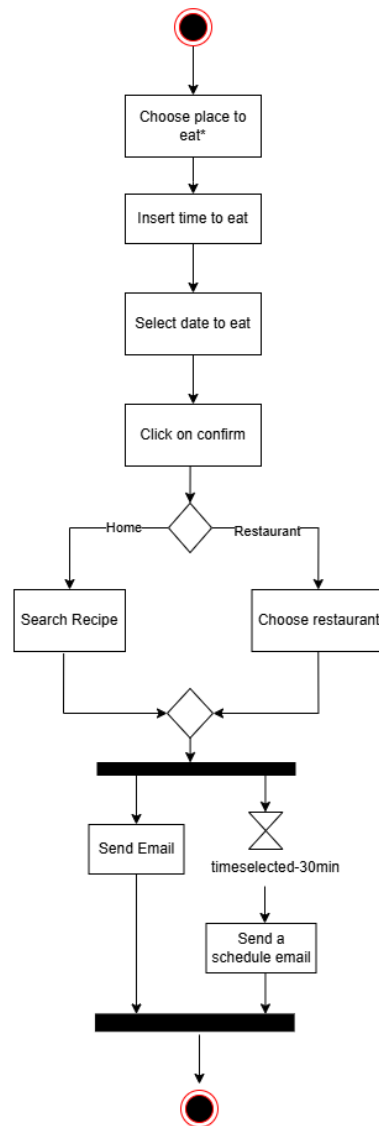
#### Activity diagram: Choose restaurant



### 3.2.1) AD 2

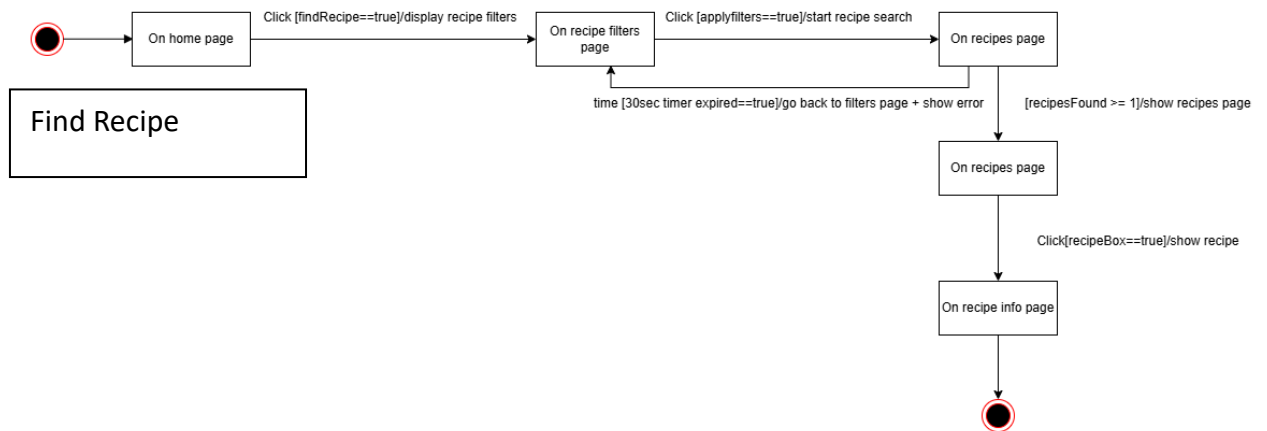
\*By saying choose we mean between Home, Restaurant

Activity diagram: Set Eating Time

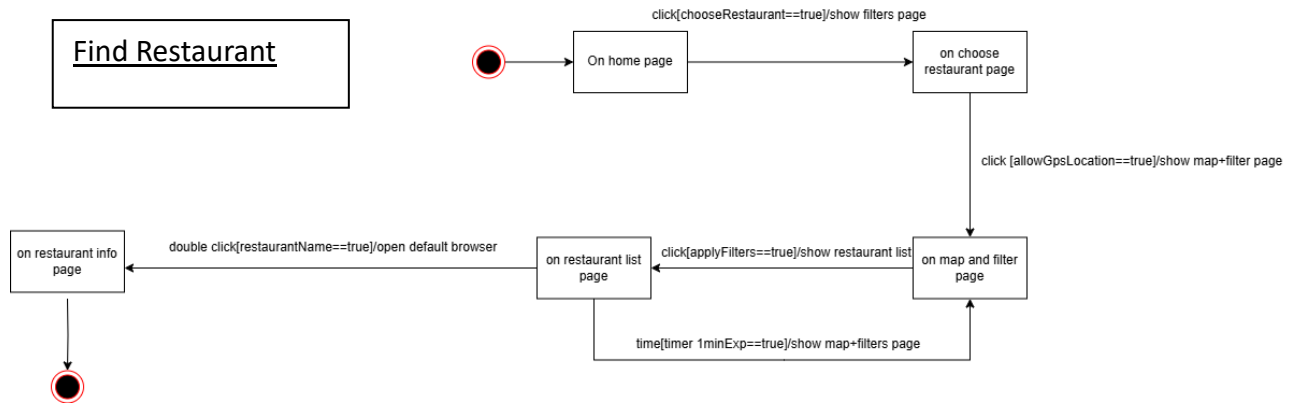


### 3.3) State diagrams

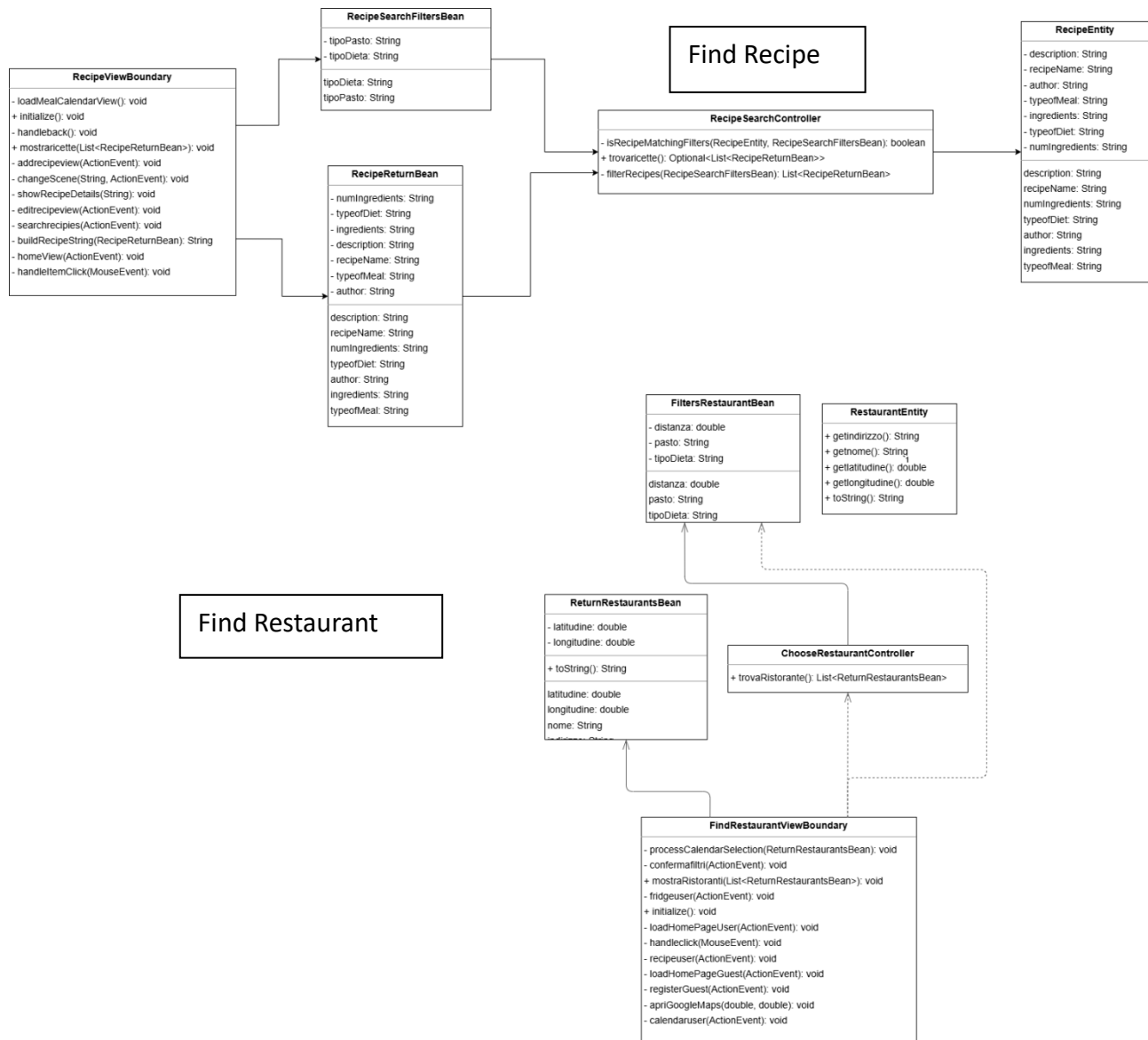
Find recipe



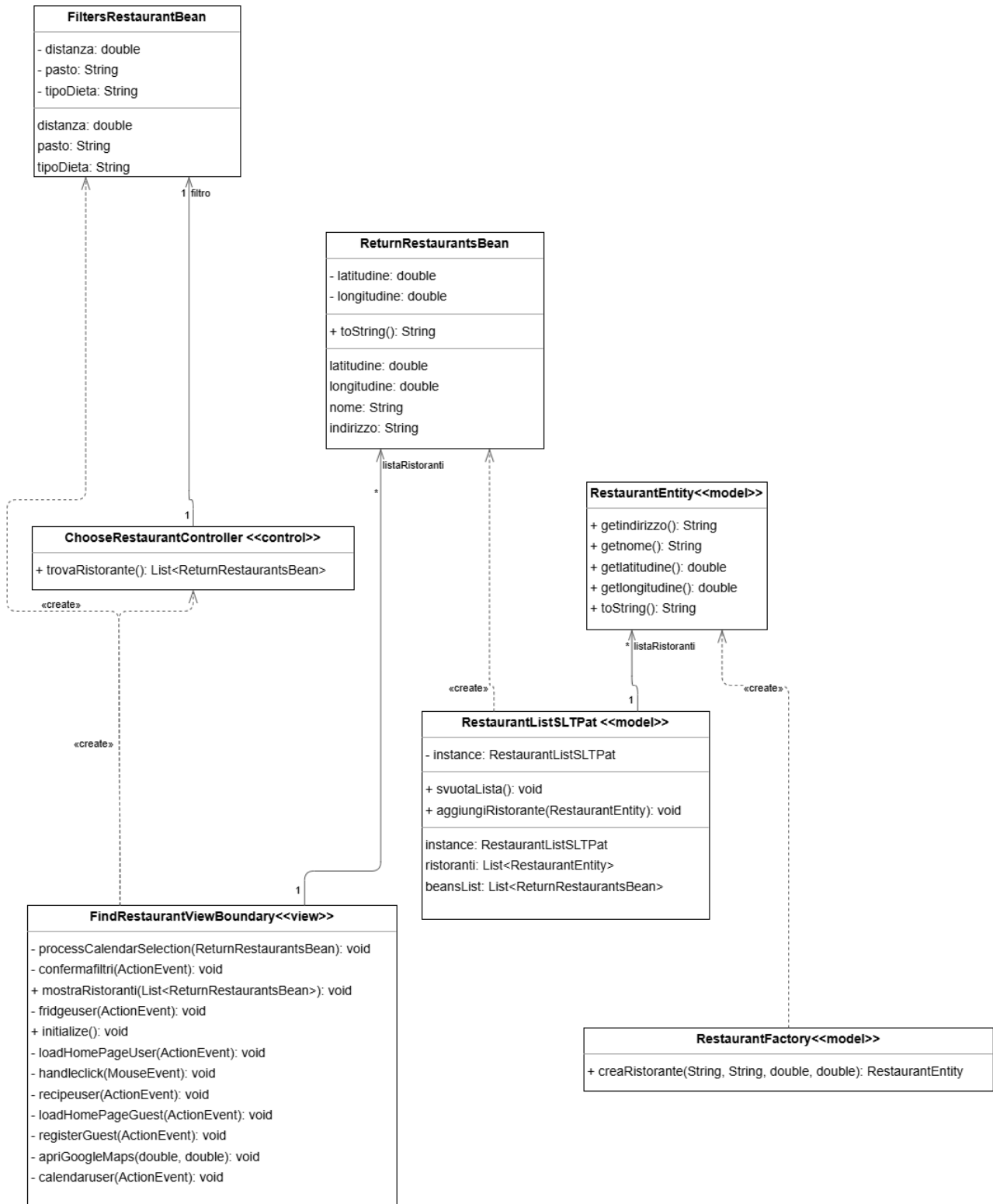
Find Restaurant



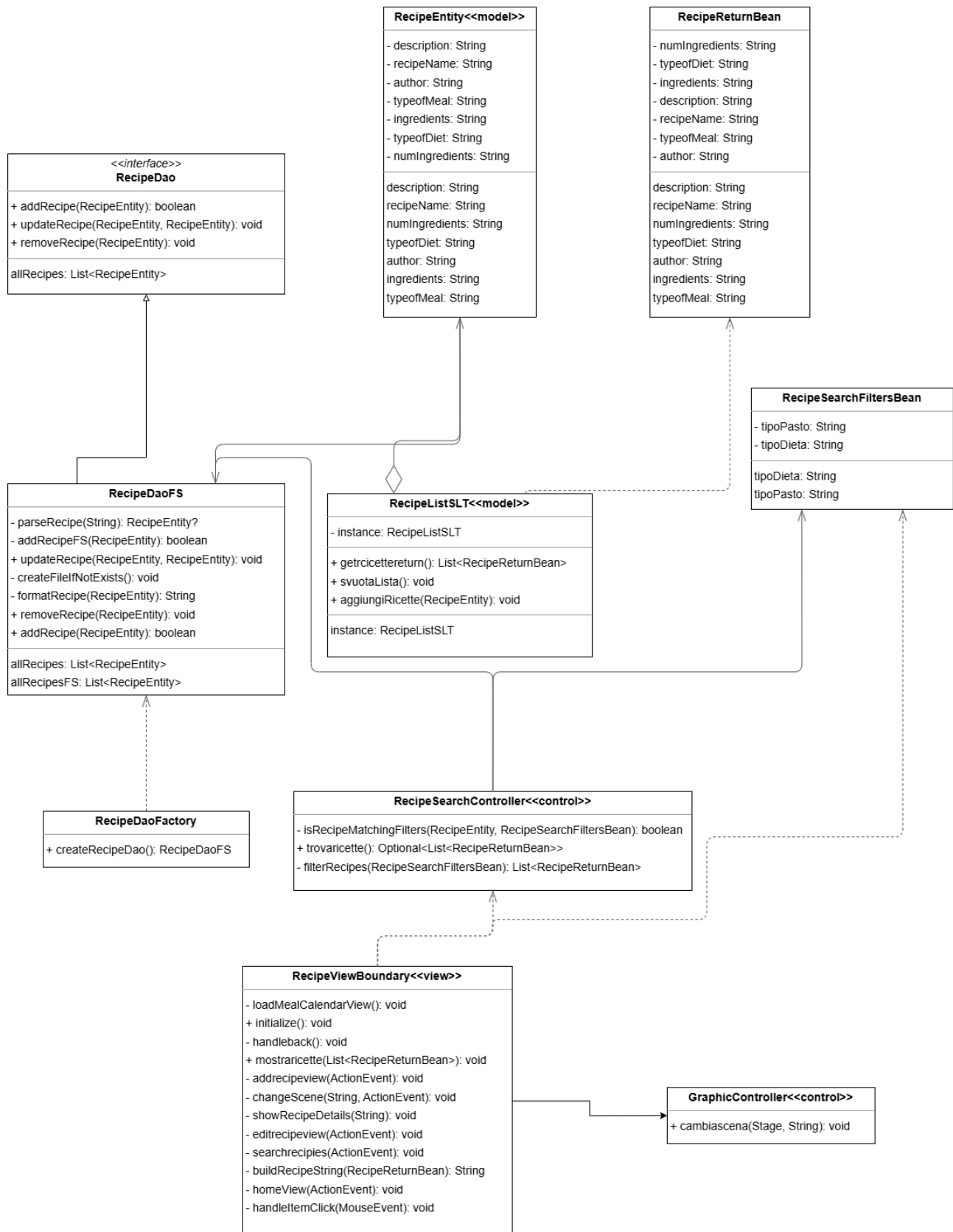
### 3.4) Diagramma delle classi BCE



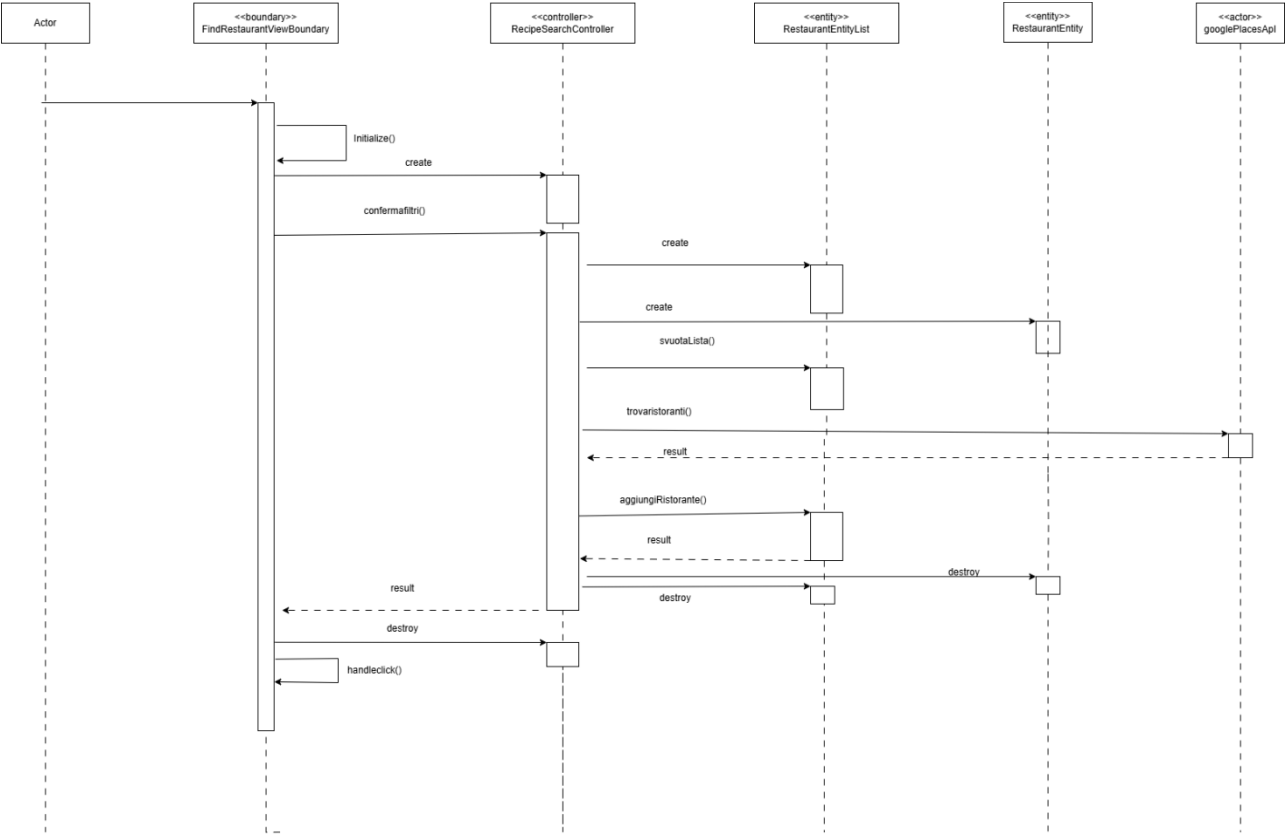
### 3.5) Diagramma di classe MVC del caso d'uso Find Restaurant



### 3.6) Diagramma di classe MVC del caso d'uso find recipe



3.7) Sequence Diagram del caso d’uso Find Recipe:



### 3.8) Sequence diagram del caso d'uso Find Restaurant.

