

# Recipefy

Alessio Cernero

# Panoramica dell'applicazione

L'applicazione vuole essere una piattaforma di gestione e condivisione delle ricette. Deve quindi essere in grado di:

- Gestire le utenze
- Permettere di gestire le proprie ricette
- Permette di condividere le proprie ricette e visualizzare quelle altrui
- Prevedere utenti speciali quali Chef e Dietisti con funzioni speciali



Questa foto di Autore sconosciuto è concesso in licenza da CC BY-NC-ND



Questa foto di Autore sconosciuto è concesso in licenza da CC BY-NC-ND



# Agenda

---

1. Ingegneria dei requisiti
2. Analisi dei requisiti
3. Modellazione di dominio
4. Design dell'Architettura
5. Sviluppo software

# Ingegneria dei requisiti

- Requisiti funzionali
- Requisiti non funzionali

# Requisiti funzionali

## User Authentication

- **Requirements ID:** FR2
- **Description:** Il Sistema prevede un processo per l'*autenticazione* che dà l'accesso a tutti i propri dati personali.

## Personal Recipe Management

- **Requirements ID:** FR3
- **Description:** Il Sistema mette a disposizione modi per *creare*, *modificare*, *visualizzare* o *eliminare* le proprie **ricette**.

## Sharing Recipe

- **Requirements ID:** FR5
- **Description:** Il Sistema permette agli utenti di rendere le proprie ricette *pubbliche* e renderle *visibili agli altri utenti*.



# Sintesi dei requisiti funzionali

---

- FR1: User Creation
- FR2: User Authentication
- FR3: Personal recipe Managment
- FR4: Recipe Organization
- FR5: Sharing Recipe
- FR6: Review System
- FR7: Personalize Recipe
- FR8: Search Functionality
- FR9: 'Post your result' Functionality
- FR10: Chef Functionality
- FR11: Nutritionist Functionality

# Requisiti non funzionali

## Back-end Implementation

- **Requirements ID:** NFR1
- **Description:** Il back-end verrà implementato in **Java** e **MySQL**

## Front-end Implementation

- **Requirements ID:** NF2
- **Description:** L'interfaccia grafica sarà implementata utilizzando **Java Swing**

## Domain Constraints

- **Requirements ID:** NFR5
- **Description:** Ogni utente potrà *creare recensioni* solo per le ricette altrui e solo una per ricetta.



## Sintesi dei requisiti non funzionali

---

- NFR1: Back-end implementation
- NFR2: Front-end implementation
- NFR3: Security Requirments
- NFR4: Accessibility requirements
- NFR5: Domain Constraints



# Glossario

Termine	Significato	Sinonimi
Utente	Persona che utilizzerà l'applicazione	User
Chef	Utente specializzato nella gestione di ristoranti	-
Nutrizionista	Utente specializzato nella gestione delle diete	Dietista, Dietitian, Nutritian
Ricetta	Guida con ingredienti, istruzioni e altre informazioni relative alla preparazione di un piatto	Recipe

Termine	Significato	Sinonimi
Recensione	Valutazione generale della ricetta	Review
Tipo di Cucina	Nazionalità di appartenenza del piatto	Cuisine Type
Attributi dietetici	Etichette che descrivono le proprietà dietetiche di un piatto	Nutritional attribute

# Analisi dei requisiti

- Casi d'uso
- Diagramma delle classi
- Diagramma Entità-Relazioni
- Diagramma di sequenza

# Use Case UC1: New User creates a New recipe

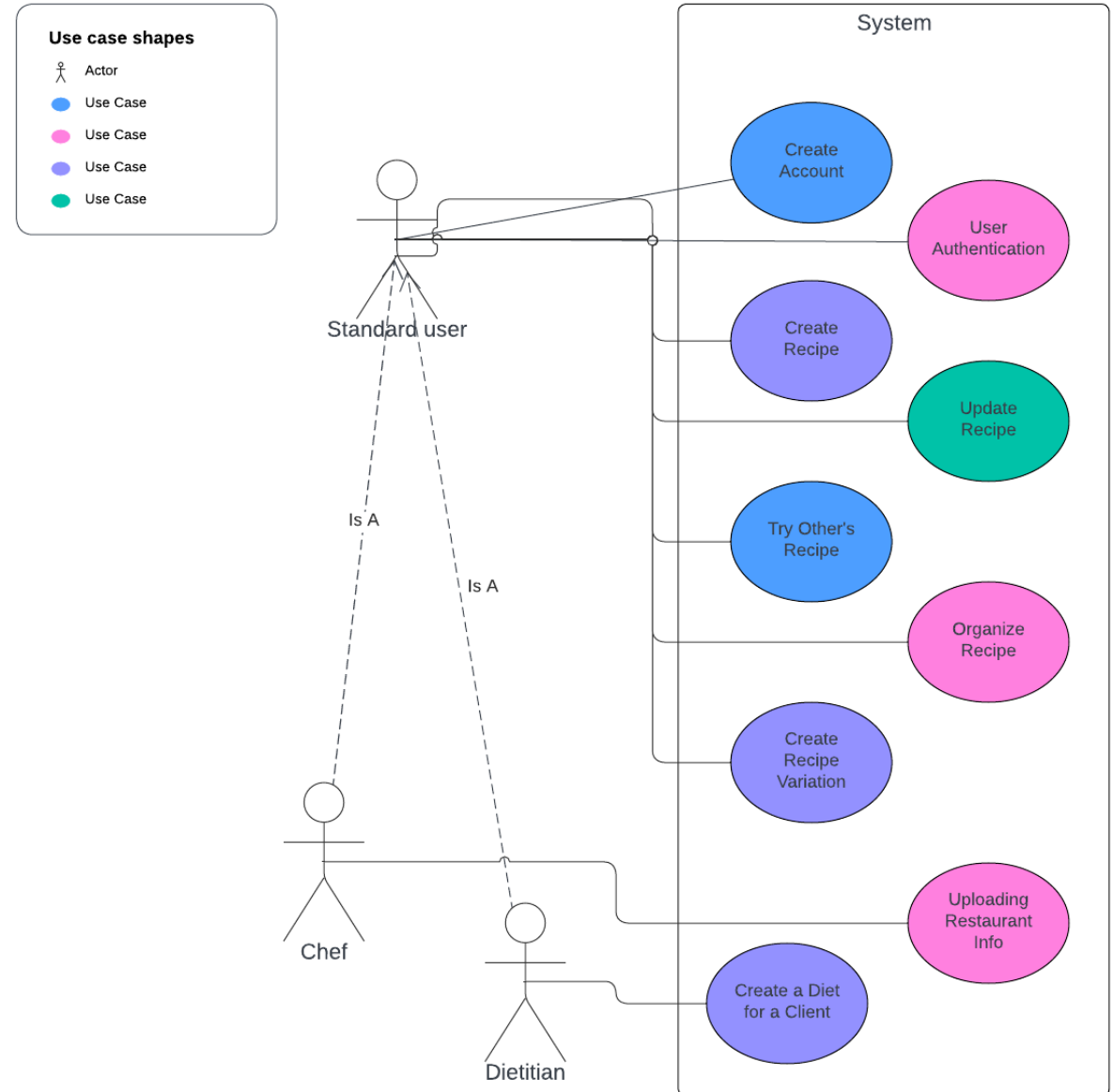
- **Primary Actor:** Standard User
- **Stakeholders and Interests:**
  - Standard User: Interests in creating an account to access the software and create a new recipe.
  - Chef (Secondary Actor): Interests in creating an account to access the software and create a new recipe.
  - Dietistic (Secondary Actor): Interests in creating an account to access the software and create a new recipe.
- **Main Success Scenario (or Basic Flow):**
  1. System renders WelcomePage with Login and Sign up forms.
  2. User fills the sign up form and clicks "Sign up".
  3. System validates data, creates the user and automatically authenticates him.
  4. System renders 'Personal recipe' for the user.
  5. User clicks on "New Recipe"
  6. System renders the 'Create Recipe' view, a form asking all the required information for the recipe.
  7. User fills in all the requested information and clicks "Save".
  8. System checks data and saves the recipe.
  9. System renders the 'Read Recipe' View of the new recipe.
- **Preconditions:** User has opened the software application.

# Use Case UC1: New User creates a New recipe

- **Extensions (or Alternative Flow):**
  - bis. User already register:
    2. User fills the log-in form
    3. System authenticates the user
    4. (Back to 4th point of Basic Flow)
  - tris. User registers as chef:
    4. System renders Restaurant - Create
    5. Users Fills in information and confirms
    6. System renders 'Personal Recipe' view (Back to 4th point)
- **Special Requirements:**
  - System should support a "personal information page" and also an "Update" version of it for collecting all the data not collected during the registration phase

# Sintesi dei casi d'uso

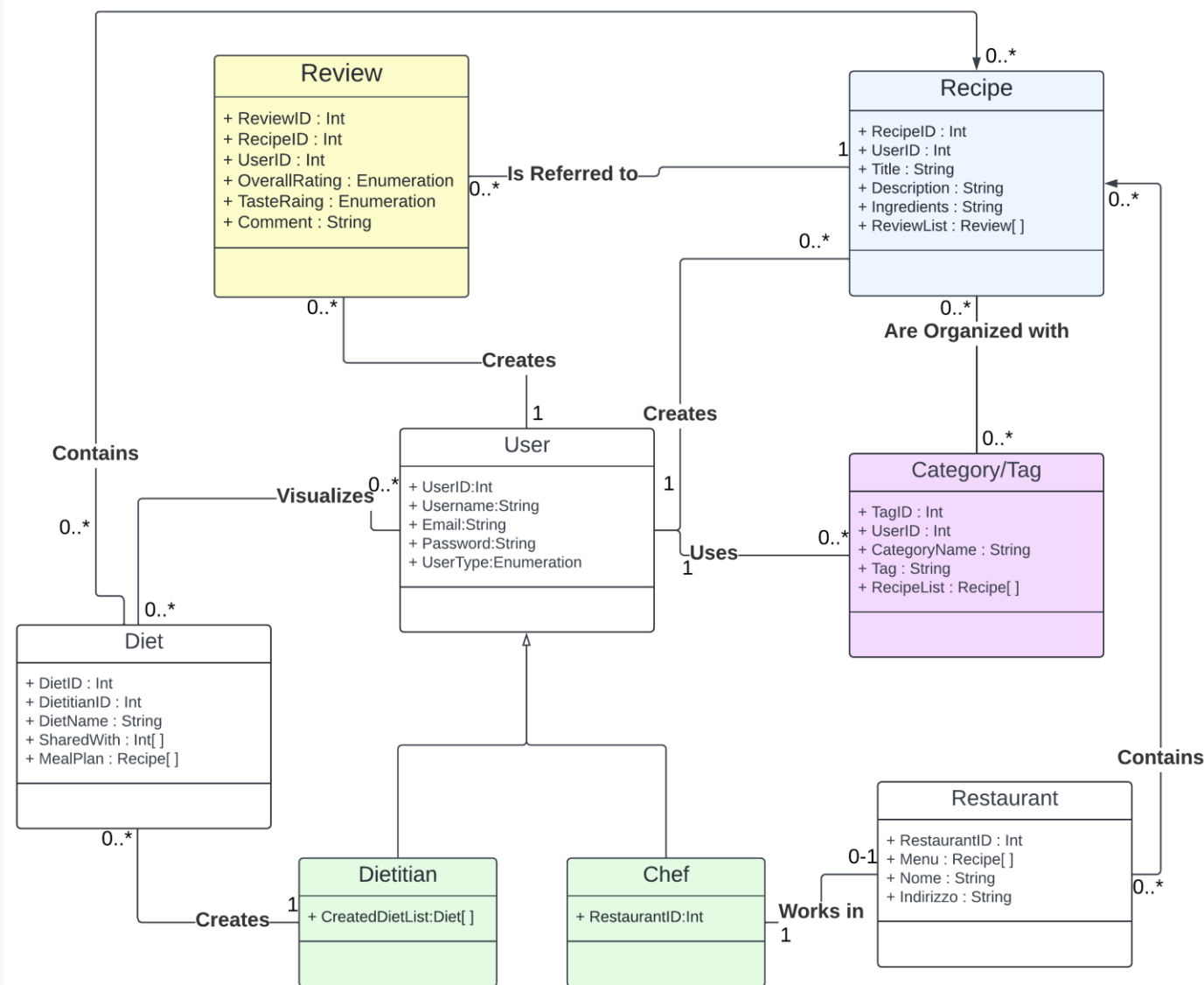
- ~~○ Create Account (OUC1)~~
  - ~~○ User Authentication (OUC2)~~
  - ~~○ Create Recipe (OUC3)~~
1. New User creates a New recipe (**UC1**)
  2. User updates his personal information (**UC2**)
  3. Visit Chef's Restaurant Page (**UC3**)
  4. Try Others' Recipe (**UC4**)
  5. Update Recipe (**UC5**)
  6. Delete Recipe (**UC6**)
  7. Organize Recipes (**UC7**)
  8. Create a Recipe Variation (**UC8**)
  9. Uploading Restaurant Information (**UC9**)
  10. Creating a Diet for a Client (**UC10**)



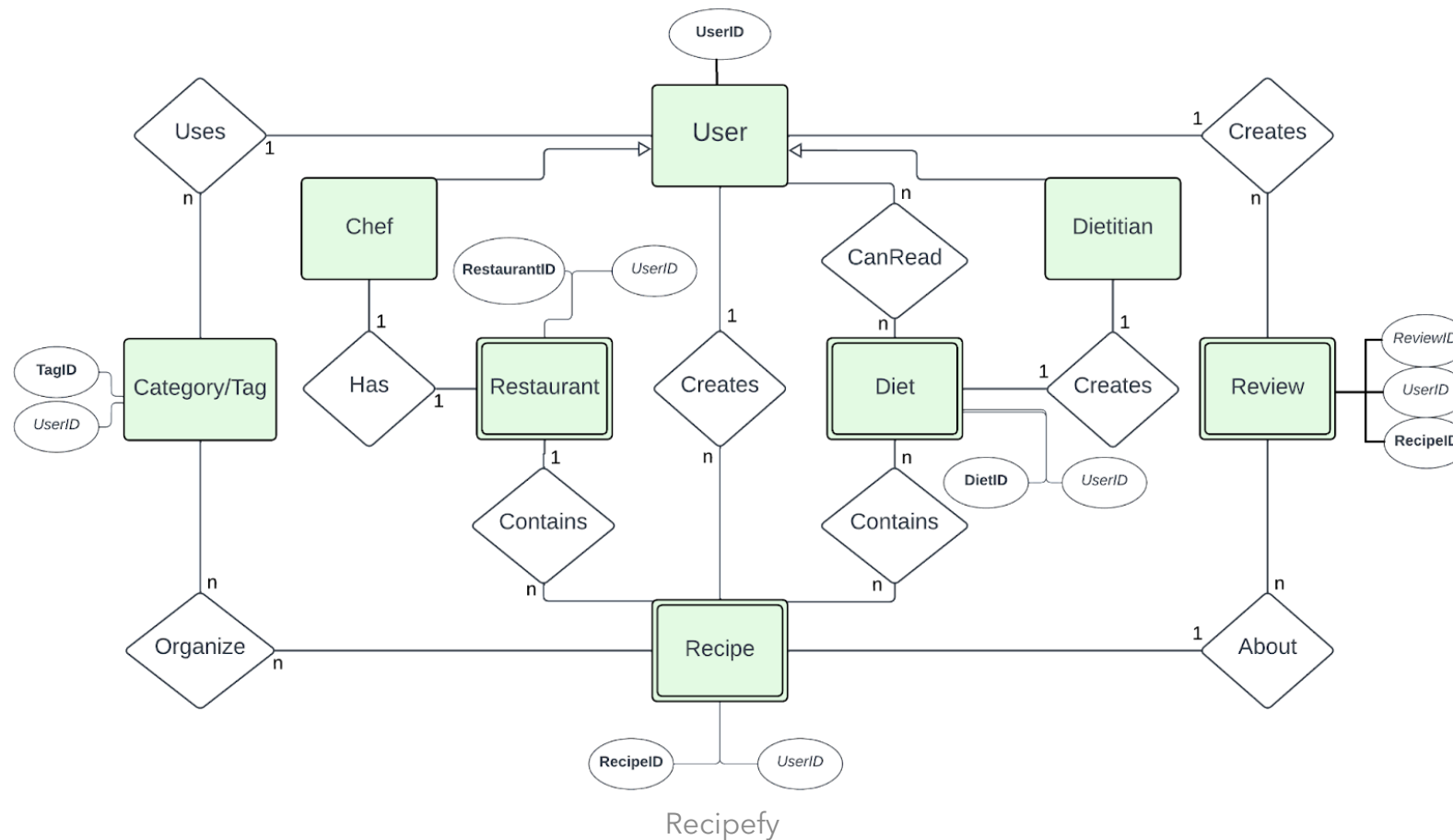
# Diagramma delle classi di dominio

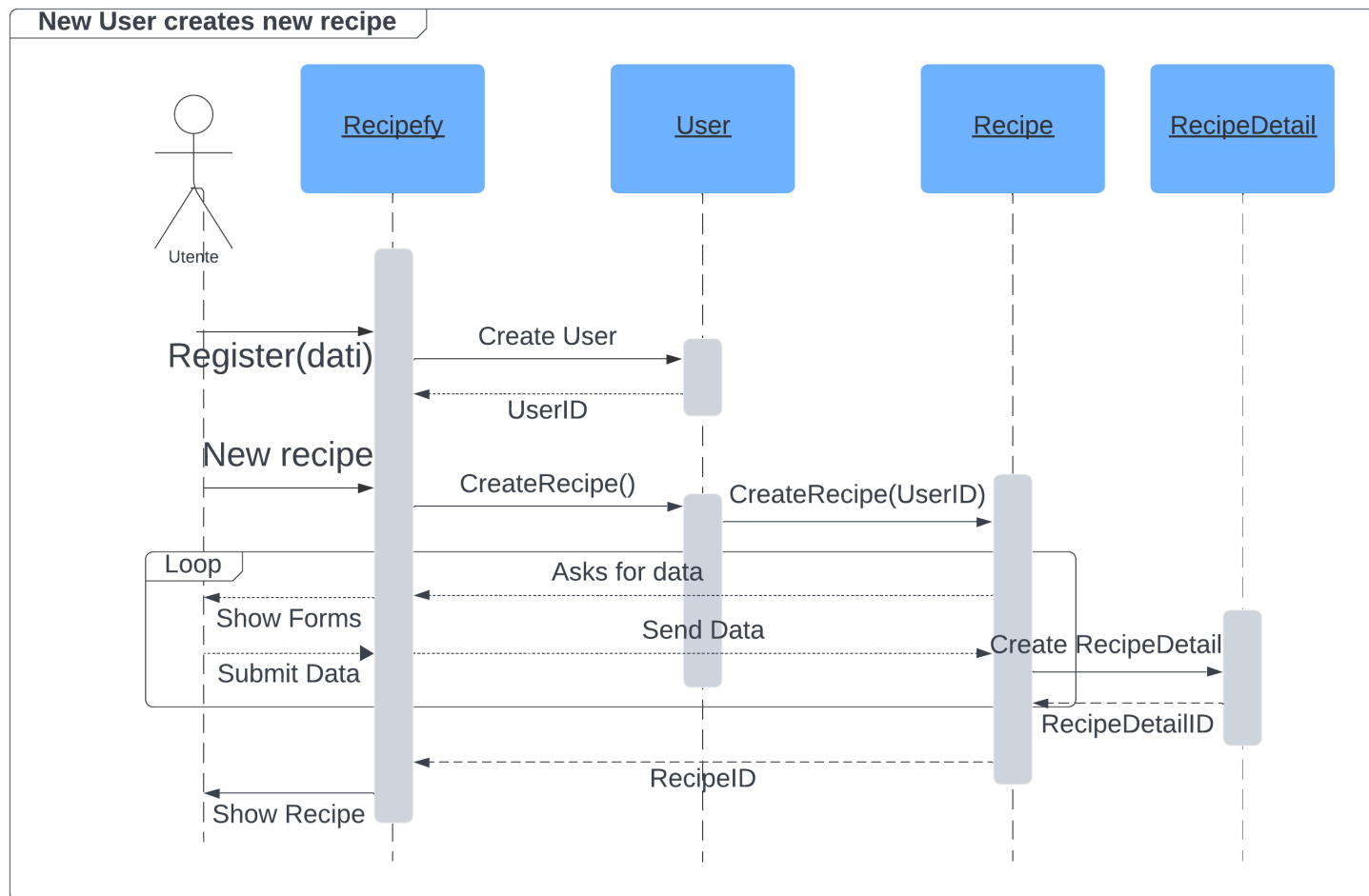
Classi identificate:

- User
- Dietista (Specializzazione di User)
- Chef (Specializzazione di User)
- Ricetta
- Categoria
- Recensione
- Dieta
- Ristoran



# Diagramma entità-relazione





## Diagramma di sequenza



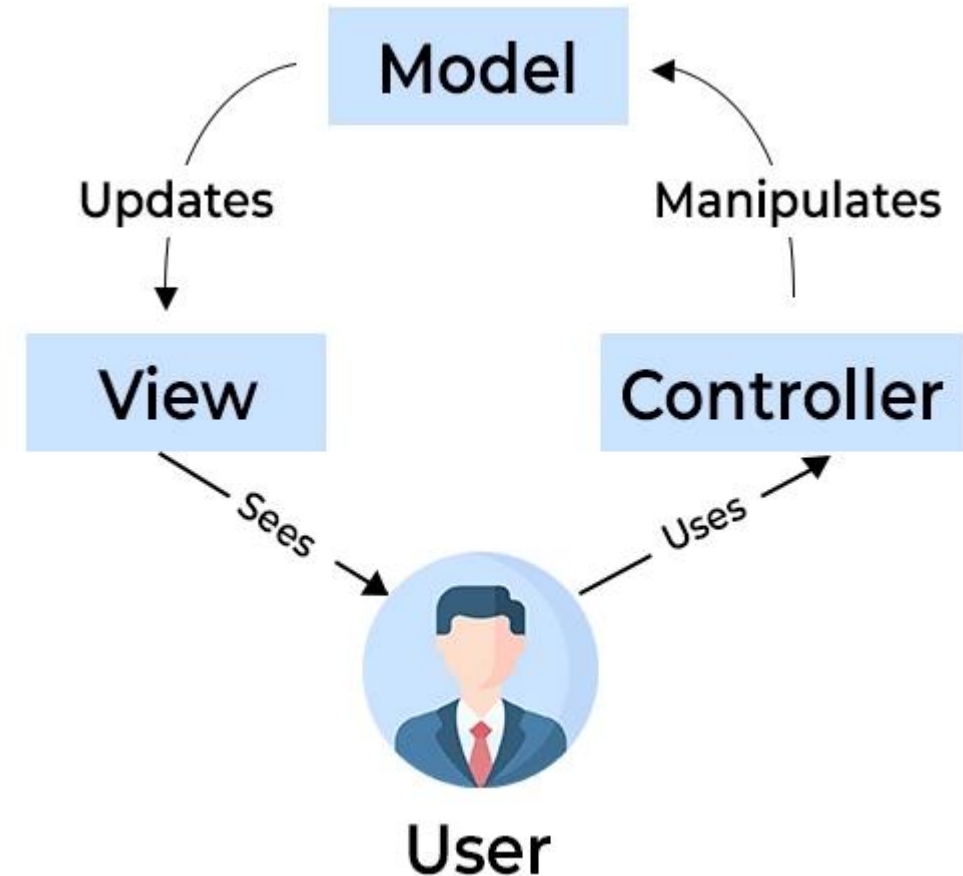


# Progettazione

- 
- Architettura

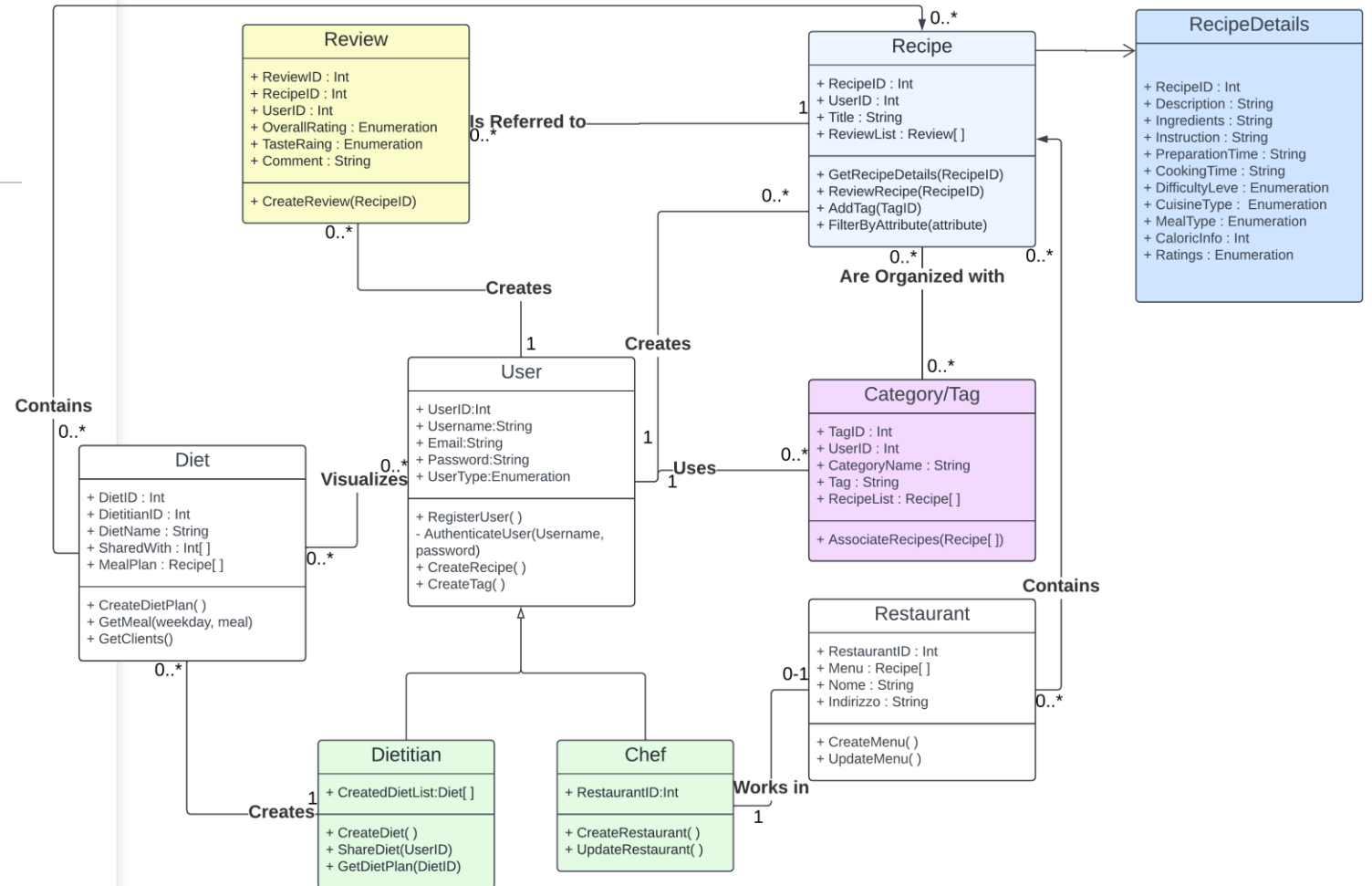
# Architettura MVC

- L'interazione Applicazione-Utente si basa tutta sul **Collezionare** e **Presentare** Dati
- Tutta l'applicazione gira attorno a poche entità e diverse funzionalità semplice che le collegano
- Non c'è bisogno di particolare sicurezza e quindi distanza tra i vari livelli



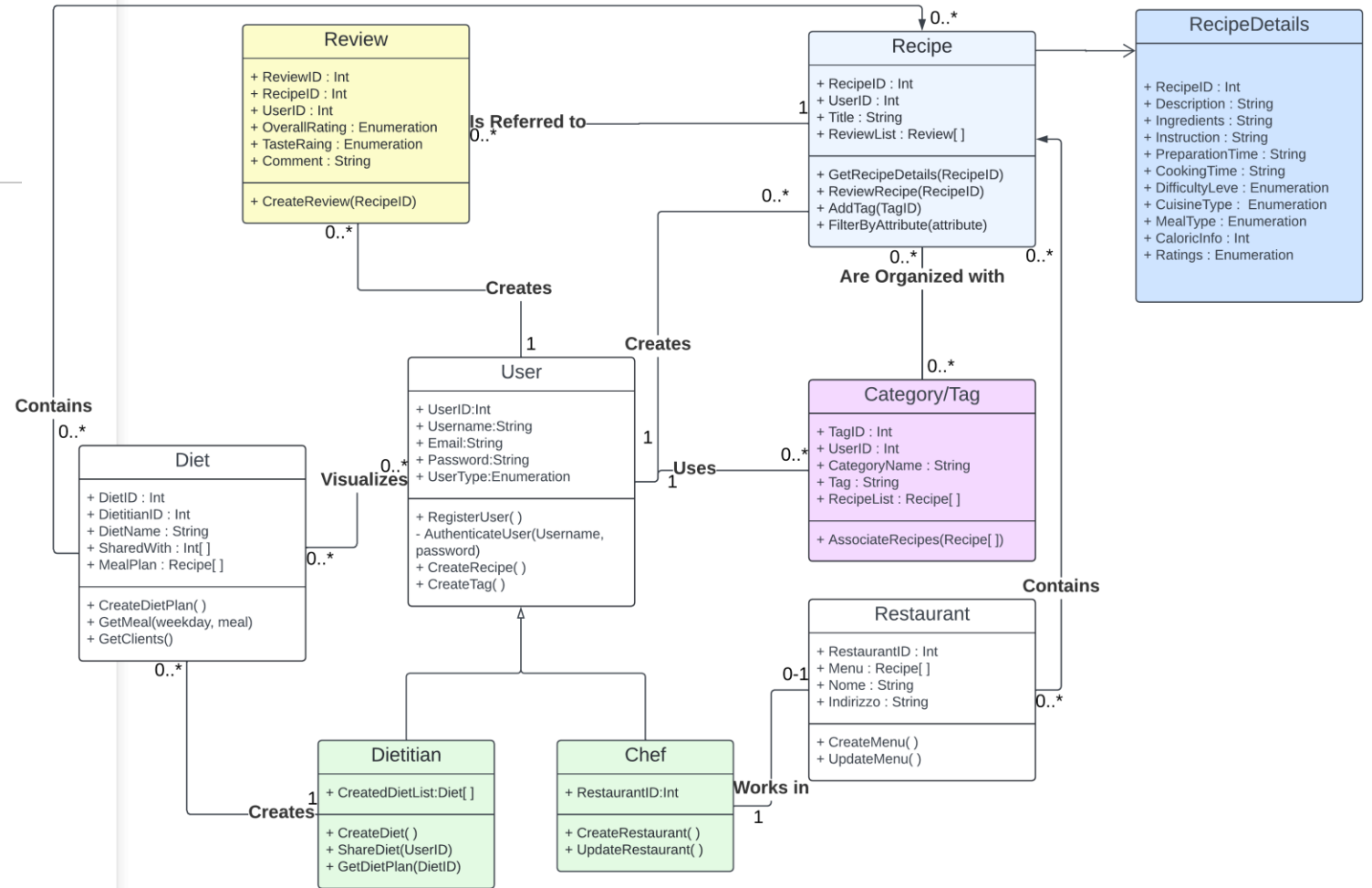
# Layer dei Modelli e Diagramma delle classi

- User Model
- Review Model
- Diet Model
- Restaurant Model
- Category Model
- Recipe Model
- RecipeDetail Model
- Recipefy Model (non mostrato)



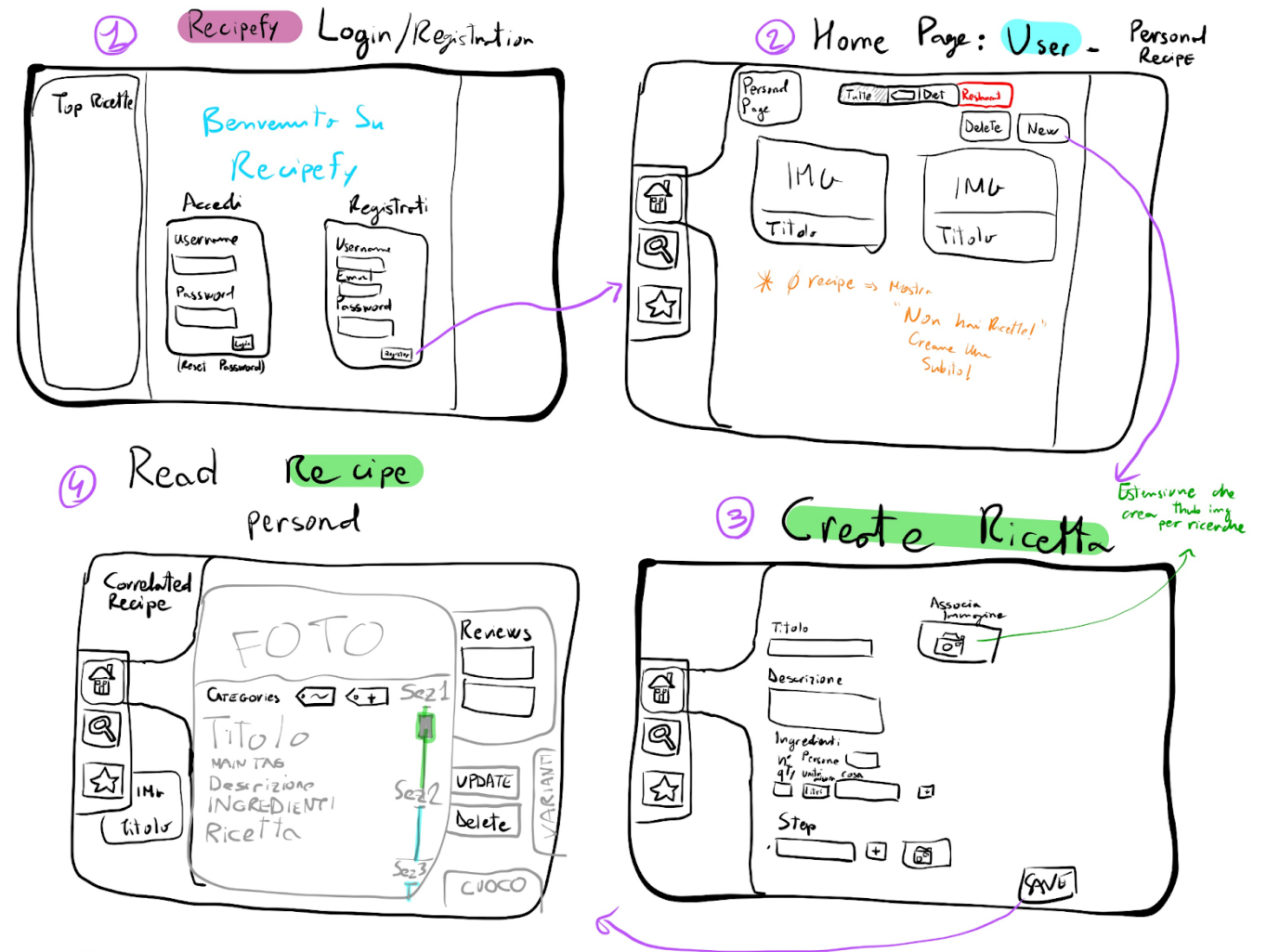
# Layer dei Modelli e Diagramma delle classi

- User Model
- Review Model
- Diet Model
- Restaurant Model
- Category Model
- Recipe Model
- RecipeDetail Model
- Recipefy Model (non mostrato)
- (+) UpdateableModel (Interface)



# Use case studies for View Layer

UC1: New User creates a New Recipe



# View identificate

- **User:**
  - HomePage/Personal CookBook (UC1,UC2,UC4, UC7,UC9,UC10)
  - Read (UC2)
  - Update (UC2)
  - PersonalCategory (UC7)
  - DietList (UC10)
- **Recipe:**
  - Create (UC1)
  - Read (UC1,UC4,UC7,UC7bis) [Personal-Others']
  - \_UploadTrial - Modal (UC4)
  - \_Ask for Review - Modal (UC4)
  - \_Select - Modal (UC4,UC9,UC10)
  - Search (UC4)
  - Update (UC5) UserNavBar
- **Recipefy:**
  - Login/Registration (UC1)
- **Review:**
  - Create (UC4)
  - \_\_\_\_MISSING\_\_\_\_ Update
  - \_\_\_\_MISSING\_\_\_\_ Read
  - \_\_\_\_MISSING\_\_\_\_ Delete
- **Category:**
  - \_Manage - Modal (UC7)
  - \_Select - Modal (UC7bis)
- **Restaurant:**
  - Read (UC9)
  - Update (UC9)
  - Create (UC1)
- **Diet:**
  - \_Create - Modal (UC10)
  - Create (UC10)
  - Read (UC10)
  - \_\_\_\_MISSING\_\_\_\_ Update
- **Components:**
  - NavBar
  - RecipeShower
  - FormPanel

# View identificate

- **User:**
  - AllPanel
  - ReadPanel
  - UpdatePanel
  - CategoryPanel
  - DietPanel
  - (+) UserNavBar
  - (+) UserPanel
- **Recipe:**
  - CreatePanel
  - ReadPanel
  - \_UploadTrial - Modal (UC4)
  - \_Ask for Review - Modal (UC4)
  - \_Select - Modal (UC4,UC9,UC10)
  - Search (UC4)
  - UpdatePanel
  - (+) CardGridShower
  - (+) CardPanel
- **MainFrame(Recipefy):**
  - (+) MainFrame
  - (+) NavBar
  - (+) MainPanel (Interface)
- **Review:**
  - Create (UC4)
  - \_\_\_\_MISSING\_\_\_\_ Update
  - \_\_\_\_MISSING\_\_\_\_ Read
  - \_\_\_\_MISSING\_\_\_\_ Delete
- **Category:**
  - \_Manage - Modal (UC7)
  - \_Select - Modal (UC7bis)
- **Restaurant:**
  - Read (UC9)
  - Update (UC9)
  - Create (UC1)
- **Diet:**
  - \_Create - Modal (UC10)
  - Create (UC10)
  - Read (UC10)
  - \_\_\_\_MISSING\_\_\_\_ Update
- **Components:**
  - NavBar
  - RecipeShower
  - FormPanel
  - (+) AttributeShower
  - (+) ImageUplodaerPanel
  - (+) FormPanel
  - (+) UpdateFormPanel
- **Interfaces:**
  - (+) UpdateableView (Interfaces)



Controller  
identificati

---

**RecipefyController**

---

**UserController**

---

**RecipeController**

---

**ReviewController**

---

**CategoryController**

---

**RestaurantController**

---

**DietController**



# Controller identificati

---

**Recipefy (MainFrameController)**

---

**UserController**

---

**RecipeController**

---

**ReviewController**

---

**CategoryController**

---

**RestaurantController**

---

**DietController**

---

(+) **Update Controller**

---

(+) **Register Controller**

---

(+) **Login Controller**

---

# Altri Package aggiunti

## Factory

- MainPanelFactory
- UserNavBarFactory
  - UserNavBarActionListenerInterface (intercae)
  - UserNavBarActionListenerFactory
  - AllPanelListener
  - AllCategoryListener
  - DietPanelListener
  - ReadPanelListener

## Utility

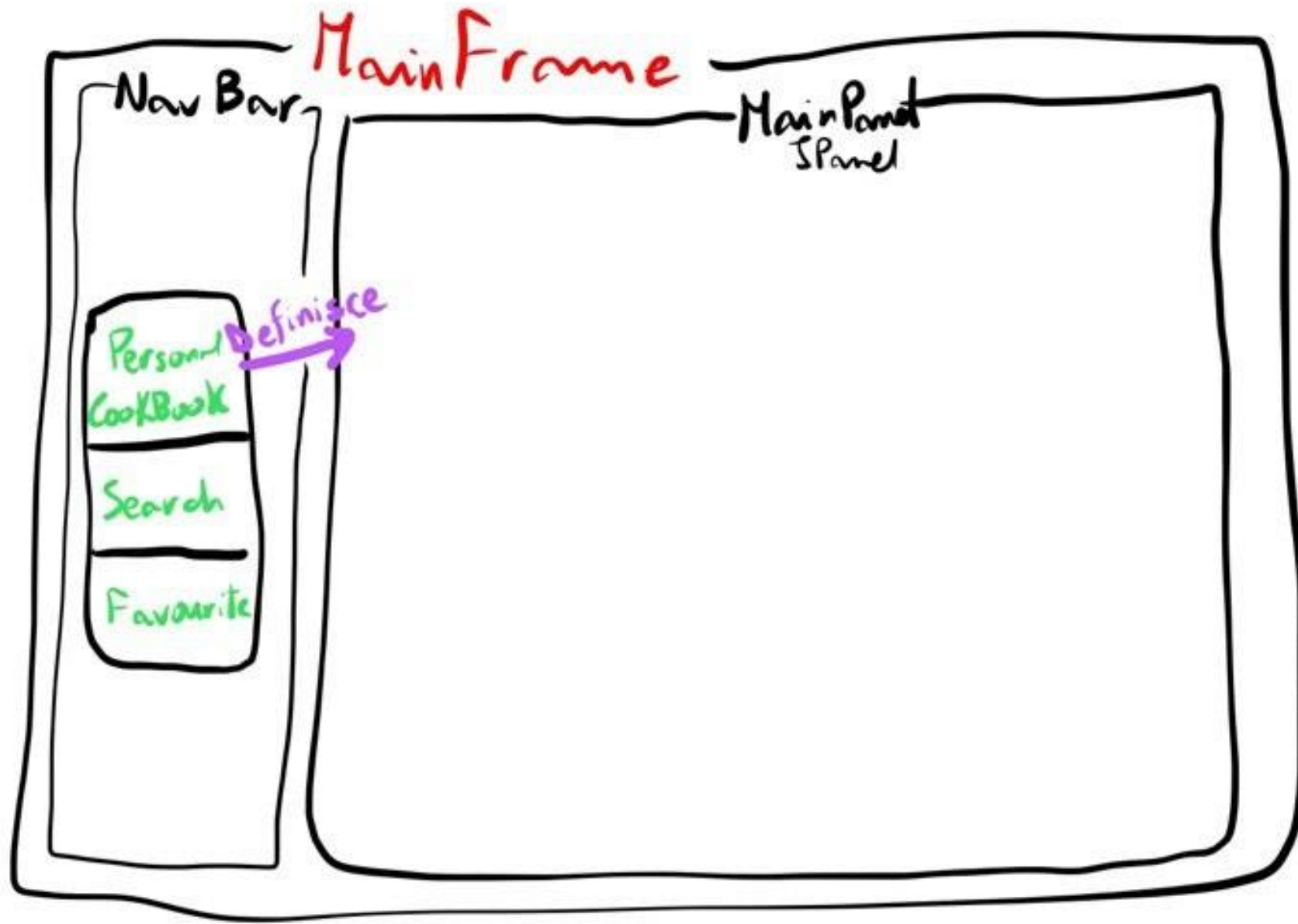
- DatabaseConnection
- DataCollector
- DataValidator

## Beans

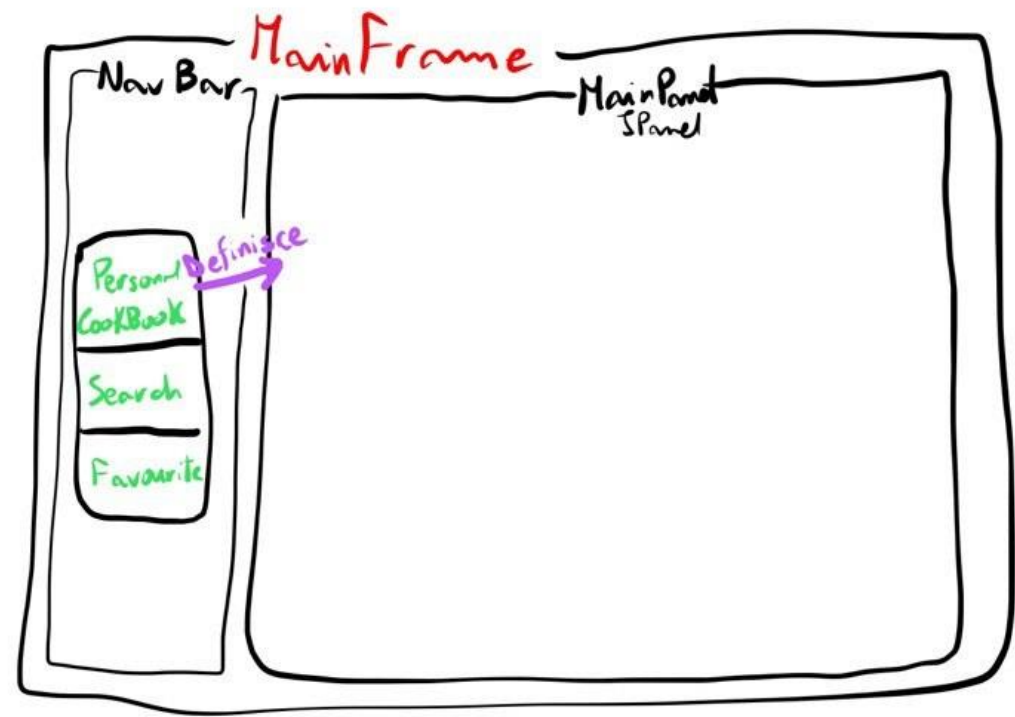
- Category
- Recipe
- RecipeDetail
- User

# Implementazione

- Design Pattern



- NavBar: **Dependency Injection**(frame)
- MainFrame**Controller**:  
CreateAndDisplay()
- Main Panel: **Factory Method**



Personal Cookbook

< Main Panel >

User Panel

Scroll Pane

UserNav Bar

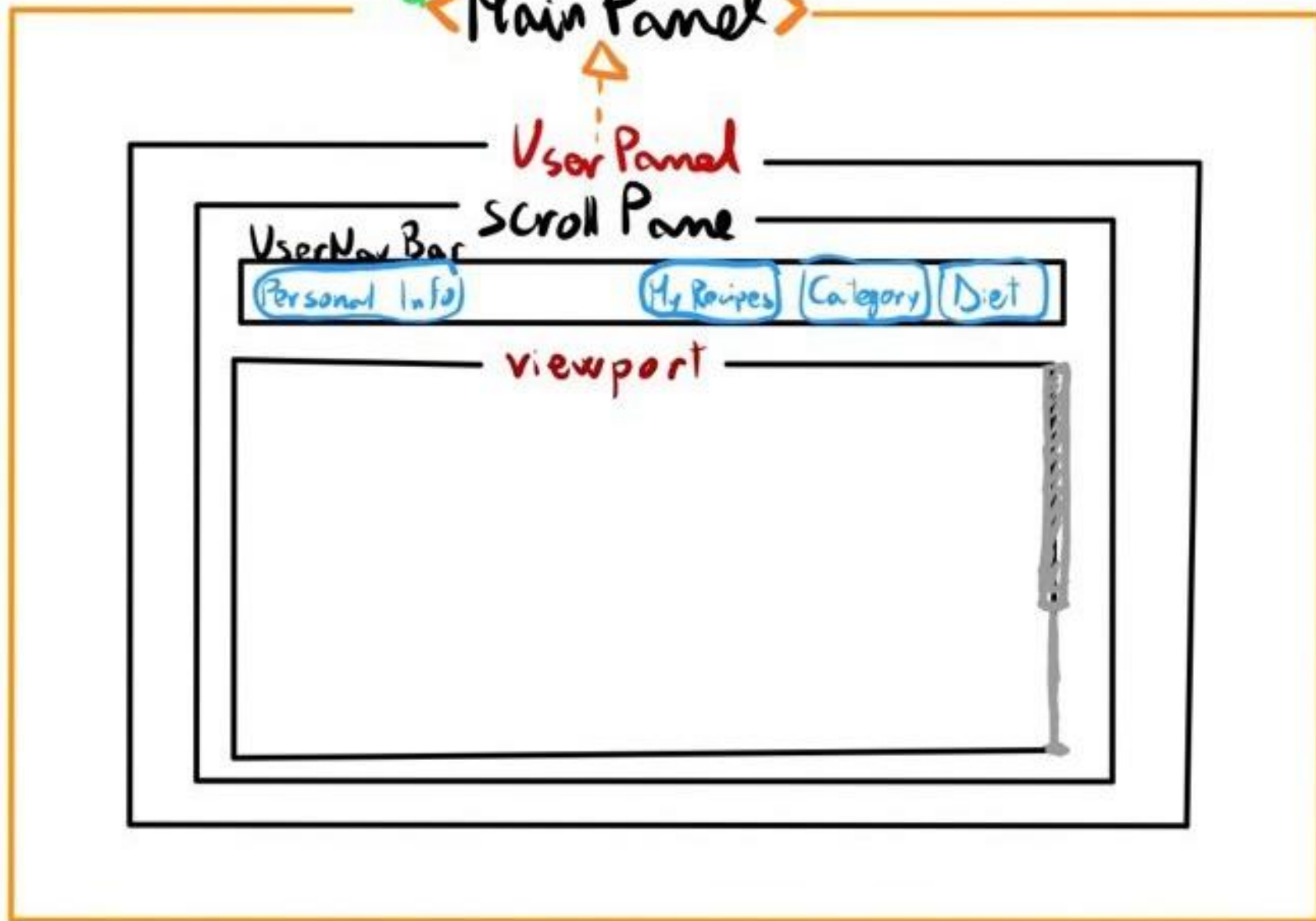
Personal Info

My Recipes

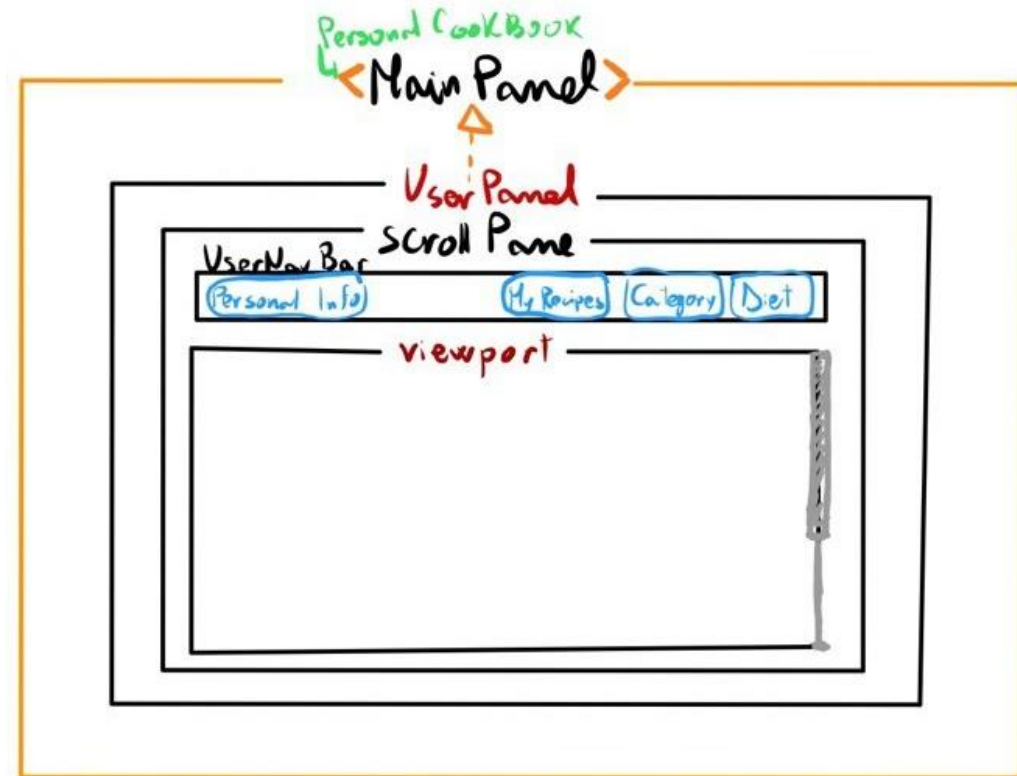
Category

Diet

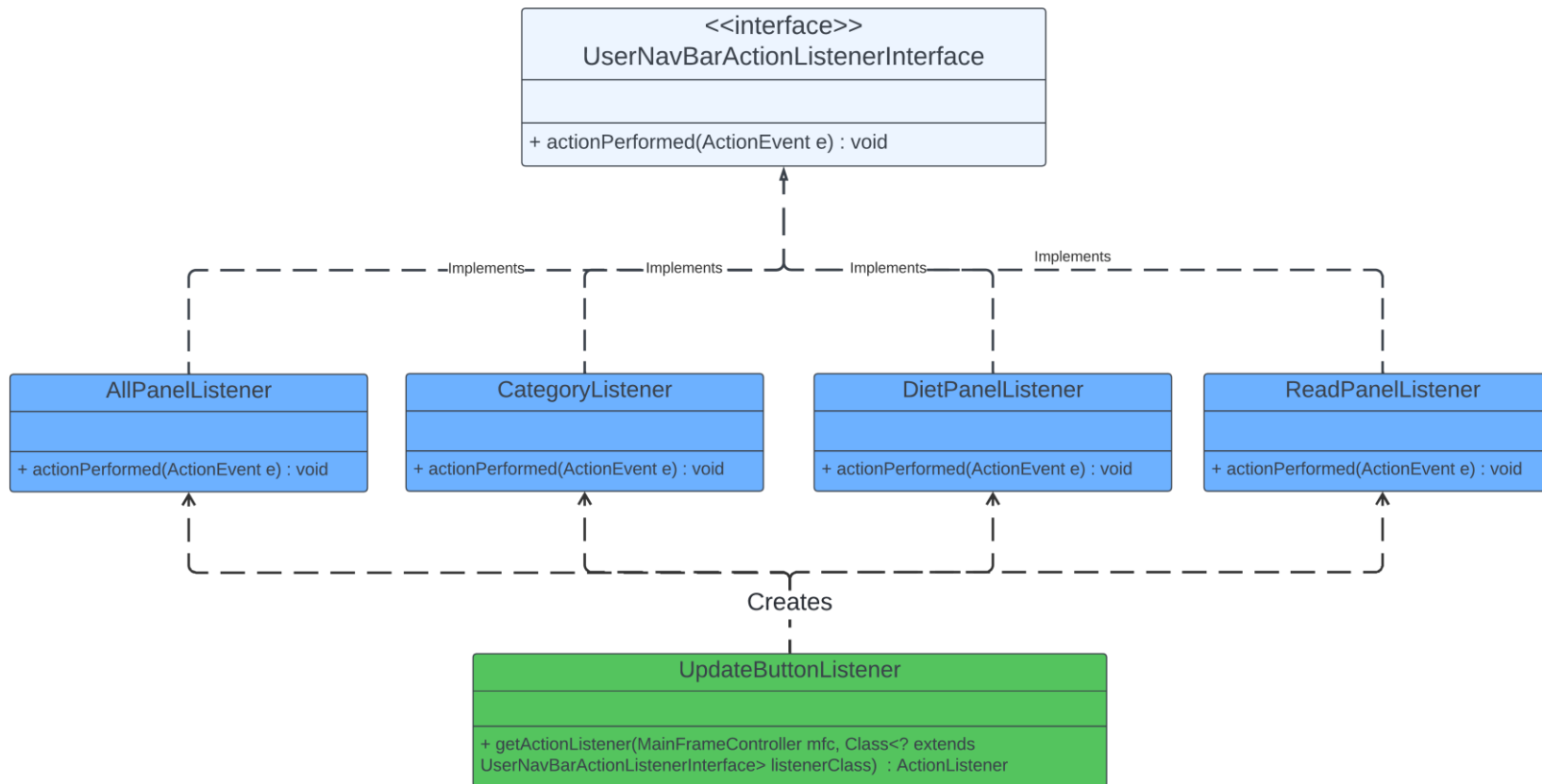
viewport



- UserNavBar:  
**FactoryMethod** for  
listener
- Interface for action  
listener



# Factory Method





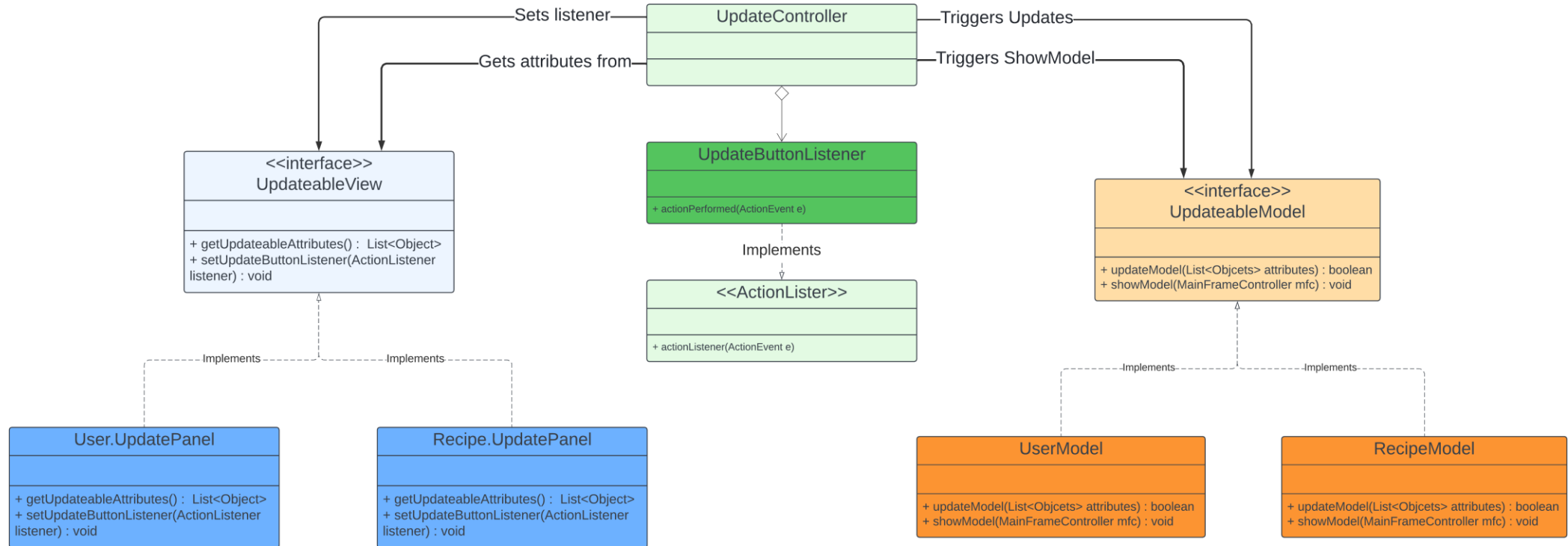
# Singleton for databaseConnection


```
2  usages  GokuGunZ
public class DatabaseConnection {
    3 usages
    private static Connection instance;
    1 usage
    private static final String url = "jdbc:mysql://localhost:3306/recipefy?serverTimezone=UTC";
    1 usage
    private static final String username = "admin";
    1 usage
    private static final String password = "RecipefyAdmin";

    no usages  GokuGunZ
    private DatabaseConnection() {}

    18 usages  GokuGunZ
    public static Connection getInstance() throws SQLException {
        if (instance == null){
            instance = DriverManager.getConnection(url, username, password);
        }
        return instance;
    }
}
```

# Controller and pseudo-Strategy for Updates





Grazie per l'attenzione