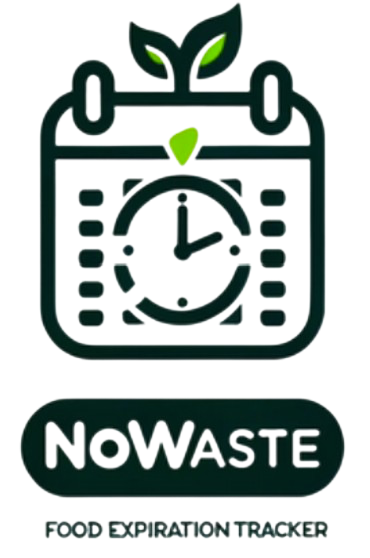
**Progetto di Ingegneria del Software 2023/24**

**Università Ca’ Foscari Venezia**

****

**Documento di Progettazione**

**2.0**

**NoWaste**

29/01/2024

*Document Informations*

|  |  |  |
| --- | --- | --- |
| NomeProgetto | Acronimo |  |
| Deliverable | Documento di Progettazione |  |
| Data di Consegna | 29 / 01 / 24 |  |
| Team Leader | Martina Ragusa | 885113@stud.unive.it |
| Team members | Daniela Spac [885386@stud.unive.it](mailto:885386@stud.unive.it)  Diego Marigo [853421@stud.unive.it](mailto:853421@stud.unive.it) |  |

*Document History*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Issue Date | Stage | Changes | Contributors |
| 1.0 | 23 / 11 / 23 | Draft | Prima stesura Documento di Progettazione | MR, DS, DM |
| 1.1 | 23 / 11 / 23 | – | Finalizzazione documento | MR, DS, DM |
| 2.0 | 29 / 01 / 24 | Final | – | MR, DS, DM |

**Indice**

[**1. Introduzione 3**](#_heading=h.6uqmpc3v0l53)

[**2. Glossario 3**](#_heading=h.30j0zll)

[**3. Architettura del sistema 4**](#_heading=h.1fob9te)

[3.1. Modello e Struttura Del Sistema 4](#_heading=h.3znysh7)

[3.2. Gestione Dei Dati 4](#_heading=h.2et92p0)

[**4. Modello dei Dati e Controllo 5**](#_heading=h.tyjcwt)

[4.1. Modello dei Dati 5](#_heading=h.3dy6vkm)

[4.2. Modello di Controllo 5](#_heading=h.1t3h5sf)

[**5. Modelli UML 6**](#_heading=h.4d34og8)

[5.1. Diagramma delle classi 6](#_heading=h.2s8eyo1)

[5.2. Diagrammi delle attività 6](#_heading=h.17dp8vu)

[5.3. Diagrammi di sequenza 10](#_heading=h.5tvp66xq04wk)

[**6. Progettazione dell’interfaccia Utente 13**](#_heading=h.3rdcrjn)

[6.1. Schermata di caricamento iniziale 13](#_heading=h.26in1rg)

[6.2. Schermata iniziale / accesso alle liste 14](#_heading=h.vdazpz55hggv)

[6.3. Vista degli ingredienti 14](#_heading=h.kti6hx4da1a3)

[6.4. Accesso utente 15](#_heading=h.q6ckowmszf0s)

[6.5. Registrazione utente 16](#_heading=h.lnxbz9)

[6.6. Accesso/registrazione via provider 16](#_heading=h.yp76icoj2qfa)

[6.7. Pagina del profilo 17](#_heading=h.4nr6gu1zwls)

[**7. Riferimenti 17**](#_heading=h.x45najfkix8f)

# Introduzione

In questo documento verranno illustrate le caratteristiche principali che compongono l’applicazione attualmente in sviluppo.

In particolare, esamineremo l’architettura complessiva dell’applicazione, le sue funzionalità principali e opzionali, nonché le decisioni progettuali fondamentali che hanno guidato il processo di sviluppo.

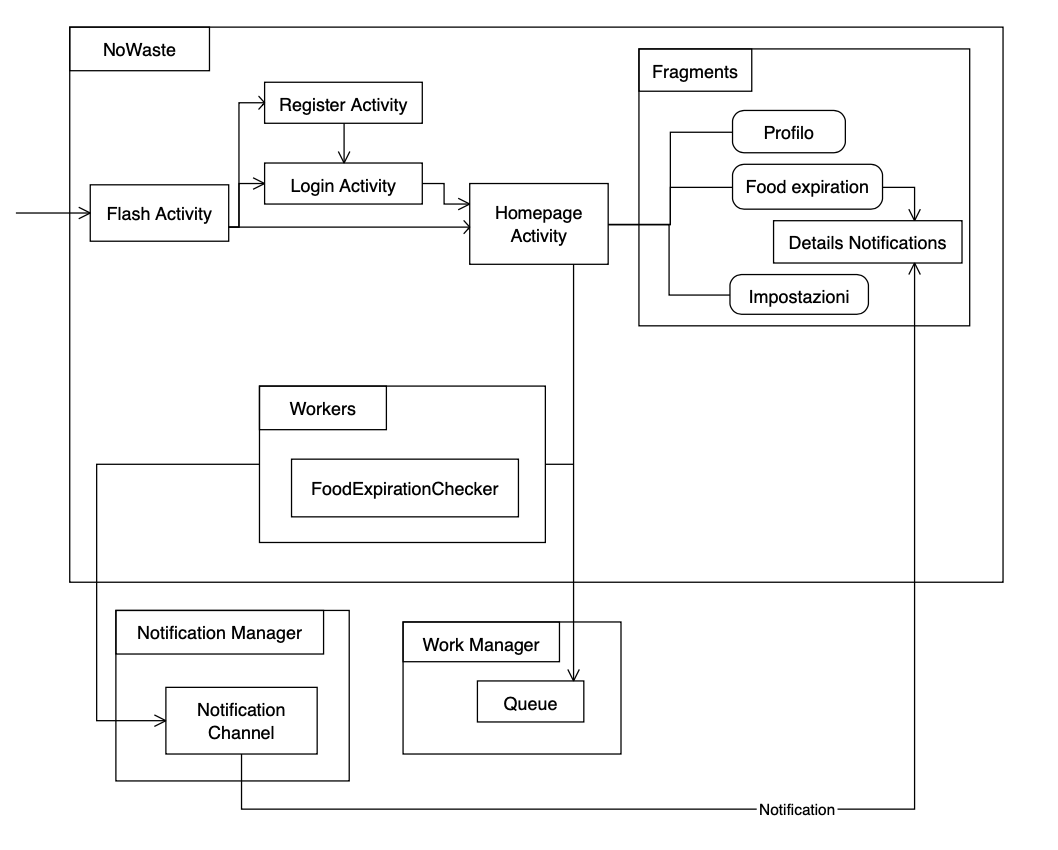
# Glossario

* **Android Studio** è un ambiente di sviluppo integrato (IDE) gratuito e open source progettato per la programmazione di app per dispositivi Android
* **Canva** Canva è una piattaforma di progettazione grafica che consente alle persone di creare elementi visivi come loghi, presentazioni, poster e altri elementi di progettazione grafica. [Fonte](https://www.canva.com/)
* **Cloud Firestore** è una piattaforma di database di Google Cloud che consente agli sviluppatori di creare, gestire e archiviare dati senza la necessità di gestire un server.
* **Firebase** è una piattaforma di sviluppo mobile e web che fornisce una serie di strumenti e SDK per lo sviluppo di applicazioni mobili e web.
* **Firebase Authenticator** è un servizio di autenticazione offerto da Google Cloud che consente agli sviluppatori di creare un sistema di autenticazione sicuro per le proprie applicazioni.
* **Firebase Storage** è un servizio di archiviazione dei file di Google Cloud che consente ai client di archiviare e accedere ai contenuti da qualsiasi dispositivo o piattaforma.
* **Fragment** è una parte di una schermata utente che può essere riutilizzata all'interno di un'applicazione Android.
* **JSON** è un (JavaScript Object Notation) è un semplice formato per lo scambio di dati. [Fonte](https://www.geekandjob.com/wiki/json" \l ":~:text=JSON%20%C3%A8%20un%20formato%20di,e%20compatta%20rispetto%20al%20rivale.)
* **Repository** è una raccolta di file, codice e documentazione che è immagazzinato e gestito su un sistema di controllo versione.
* **Top Down** I modelli top-down e bottom-up sono strategie di elaborazione dell'informazione e di gestione delle conoscenze, riguardanti principalmente il software [Fonte](https://it.wikipedia.org/wiki/Progettazione_top-down_e_bottom-up)
* **UML** Unified Modeling Language e si riferisce ad una varietà di modalità con le quali i programmi software possono essere visualizzati attraverso i diagrammi. [Fonte](https://moqups.com/it/templates/diagrams-flowcharts/uml-diagrams/)

# Architettura del sistema

## Modello e Struttura Del Sistema

Il sistema non è altro che un insieme di sottosistemi, in cui ognuno opera per risolvere un task specifico. Ogni sottosistema deve interagire e operare insieme agli altri sottosistemi per poter portare a termine l’obiettivo comune, ovvero l’esecuzione corretta dell’applicazione. La cooperazione tra i vari sottosistemi viene illustrata più chiaramente dal seguente schema:



## Gestione Dei Dati

L’insieme di tutti i dati che vengono gestiti dall’applicazione sono salvati in un database esterno e vengono interscambiati con l’applicazione per permetterne l’esecuzione.

Questa applicazione utilizza i database Firebase, che sono gestiti nei seguenti modi:

* Realtime Database: è un servizio di archiviazione cloud in tempo reale che consente agli sviluppatori di creare applicazioni che possono sincronizzare i dati degli utenti in modo istantaneo. È particolarmente adatto per applicazioni che richiedono aggiornamenti in tempo reale e collaborativi tra gli utenti.

Alcune sue caratteristiche chiave includono:

1. Una struttura dei dati ad albero JSON, dove ogni nodo rappresenta un’entità di dati.
2. Aggiornamenti in tempo reale: una delle caratteristiche più distintive è la capacità di sincronizzare automaticamente i dati tra tutti gli utenti connessi in tempo reale. Quando i dati vengono modificati da un utente, le modifiche vengono immediatamente propagate a tutti gli altri utenti che stanno accedendo agli stessi dati.
3. Accesso sicuro: Firebase offre regole di sicurezza che consentono agli sviluppatori di definire chi può leggere o scrivere dati specifici. Le regole possono essere personalizzate per adattarsi alle esigenze dell'applicazione e proteggere i dati sensibili.

Proprio per tutte queste sue caratteristiche, abbiamo scelto il Realtime Database di Firebase per salvare i dati degli utenti e le varie liste di alimenti.

* Firebase Authentication: è un servizio fornito da Firebase che consente agli sviluppatori di implementare facilmente l'autenticazione degli utenti nelle loro applicazioni, e viene utilizzato per memorizzare mail, password e UID (è un identificatore univoco assegnato a ogni utente che si registra o accede a un progetto Firebase). Nel nostro caso, viene utilizzato per effettuare il login una volta soltanto, senza reinserire le credenziali ad ogni riapertura successiva.

# Modello dei Dati e Controllo

## Modello dei Dati

Il **modello dei dati** che verrà utilizzato è un modello a repository.

Questo si traduce in un unico punto di riferimento per la gestione di tutte le informazioni relative agli alimenti e ai profili utente. L'architettura modulare garantisce che ciascun componente del sistema, come la gestione degli alimenti o i profili utente, interagisca con il repository condiviso, mantenendo coerenza e integrità dei dati.

## Modello di Controllo

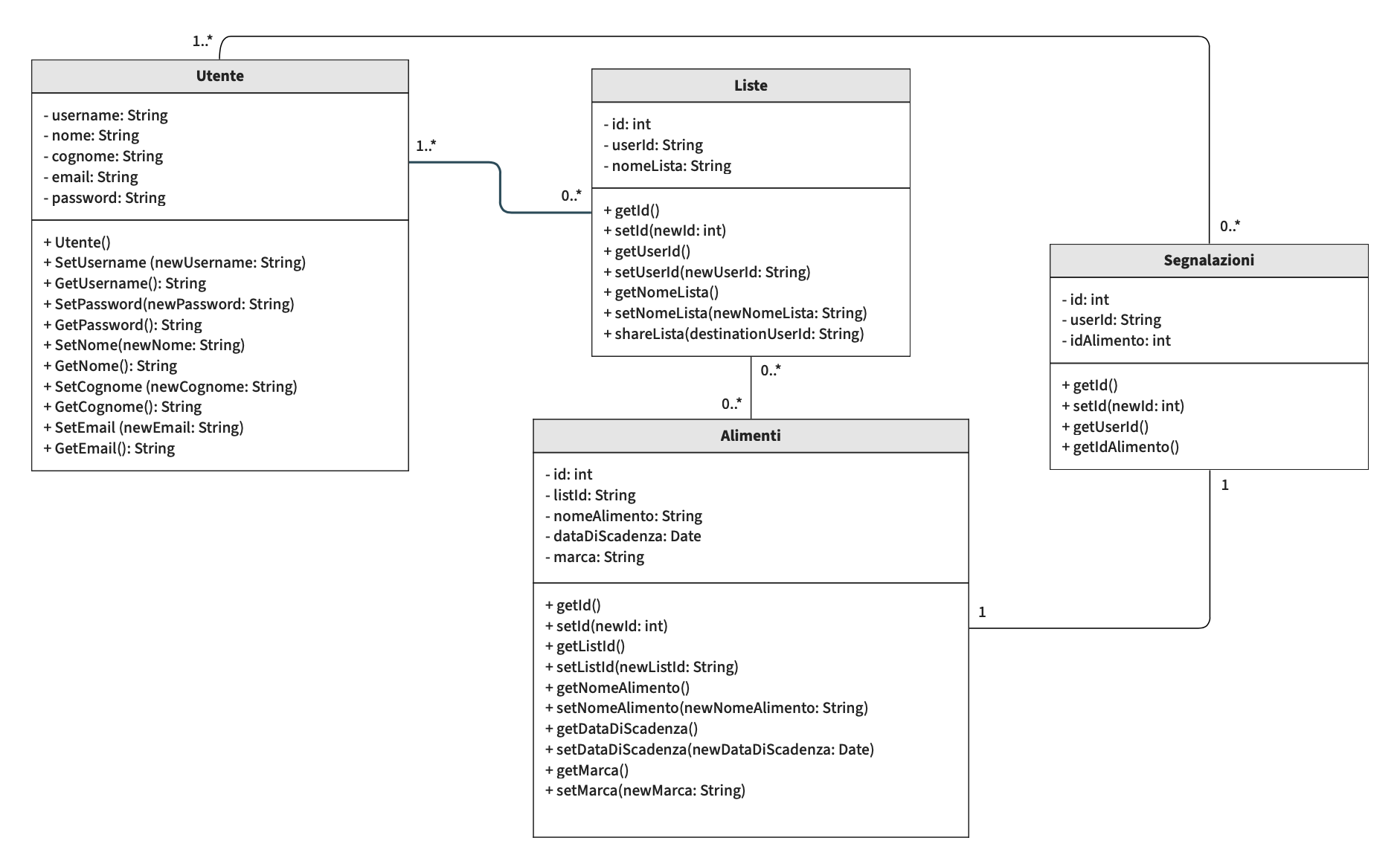
Il **modello di controllo** per il progetto è centralizzato.

In questo modello, un sottosistema centrale o "controllore" gestisce l'esecuzione e l'interazione degli altri componenti del sistema. Il controllo procede in modo gerarchico: inizia dal livello superiore e si propaga verso i livelli inferiori attraverso l'esecuzione di funzioni o metodi specifici. Questo approccio garantisce un'organizzazione chiara del flusso di controllo e facilita la gestione delle interazioni tra i vari componenti.

# Modelli UML

## Diagramma delle classi

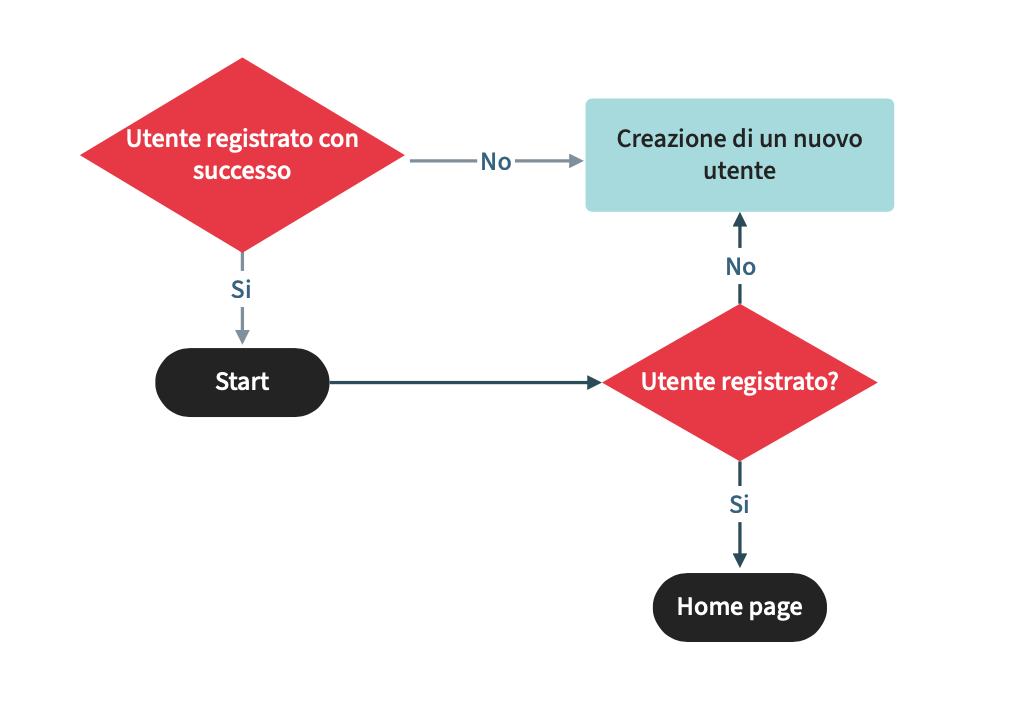
Il diagramma sotto riportato rappresenta le classi di oggetti del nostro sistema con i loro attributi e operazioni, mostrando inoltre le relazioni tra le classi.



## Diagrammi delle attività

Di seguito, vengono inseriti i diagrammi delle attività che consentono di illustrare il processo interno del sistema, delineando l’ordine con cui le diverse attività vengono eseguite ed evidenziando i punti decisionali che collegano tali attività.

### MAIN

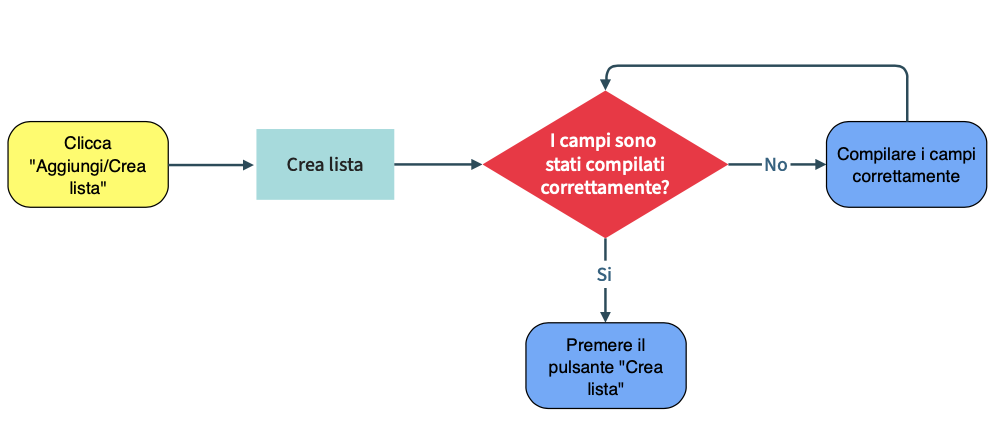


### HOMEPAGE

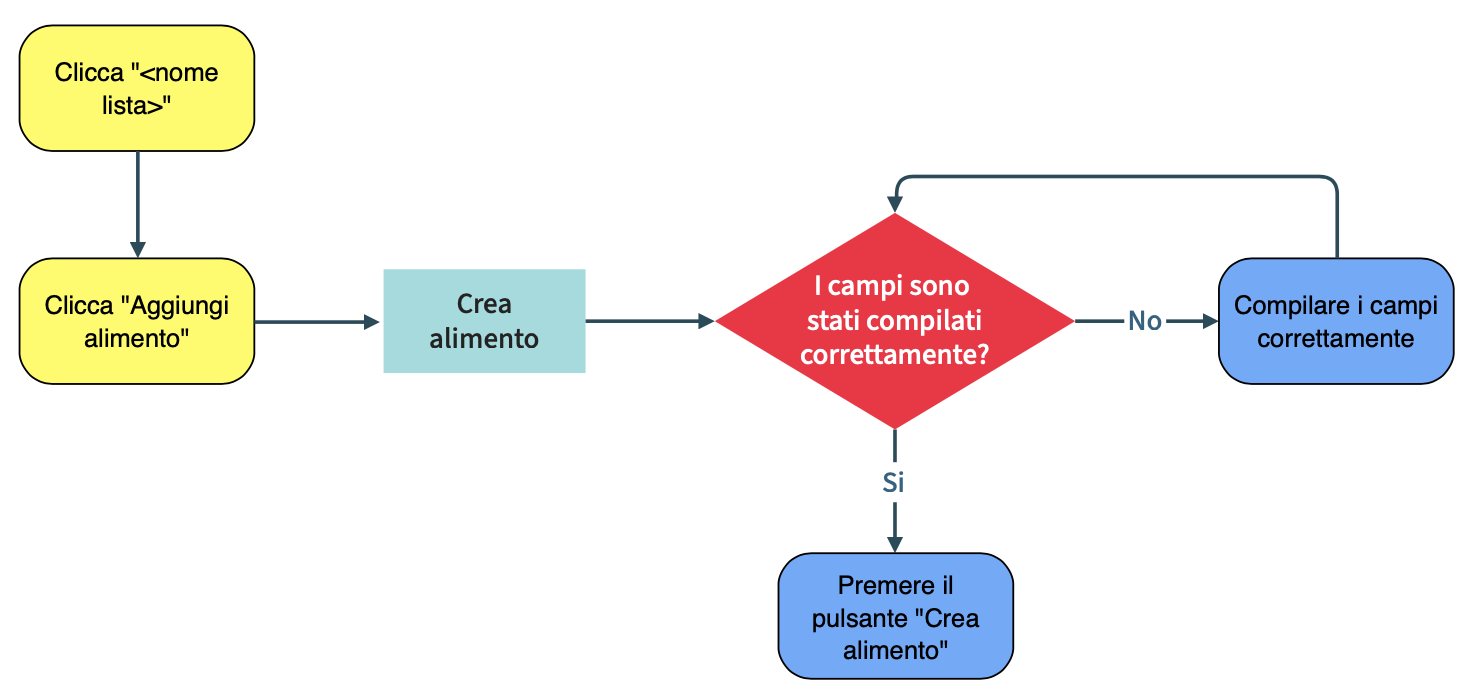
A diagram of a website

Description automatically generated

### AGGIUNGI LISTA



### AGGIUNGI ALIMENTO

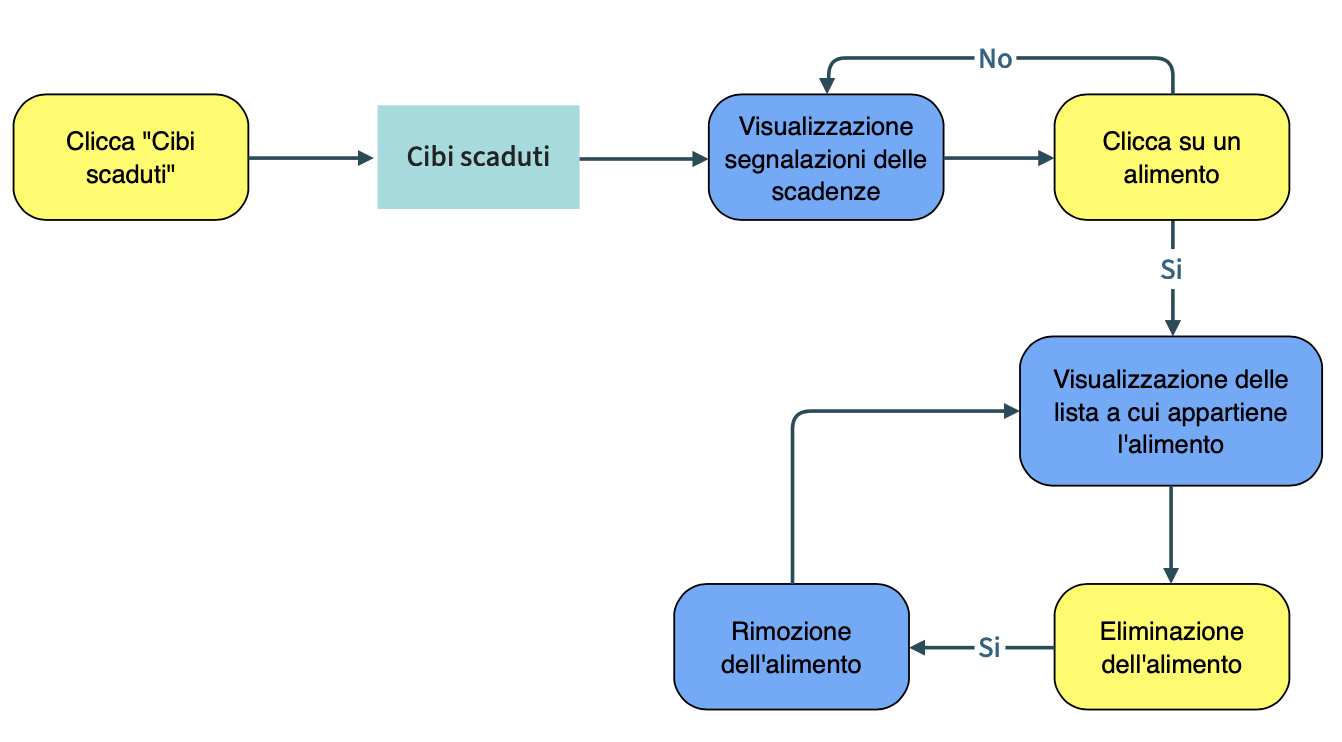


### ELENCO DELLE LISTE E DEGLI ALIMENTI PER CIASCUNA LISTA

A diagram of a diagram

Description automatically generated

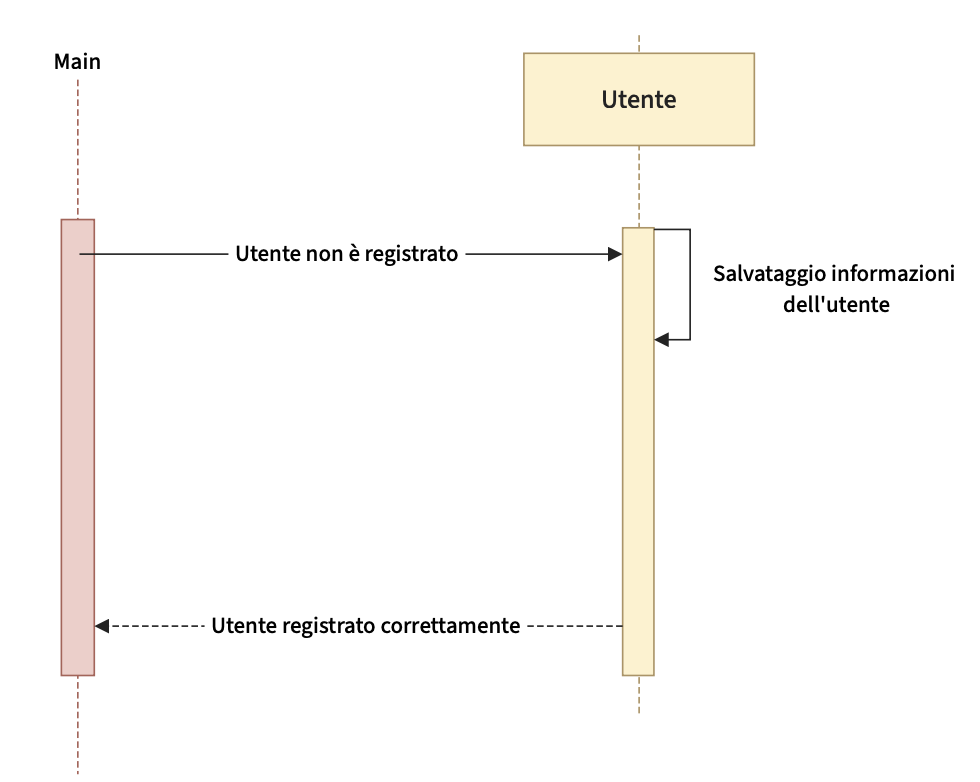
### FOOD EXPIRATION + ELIMINAZIONE ALIMENTO



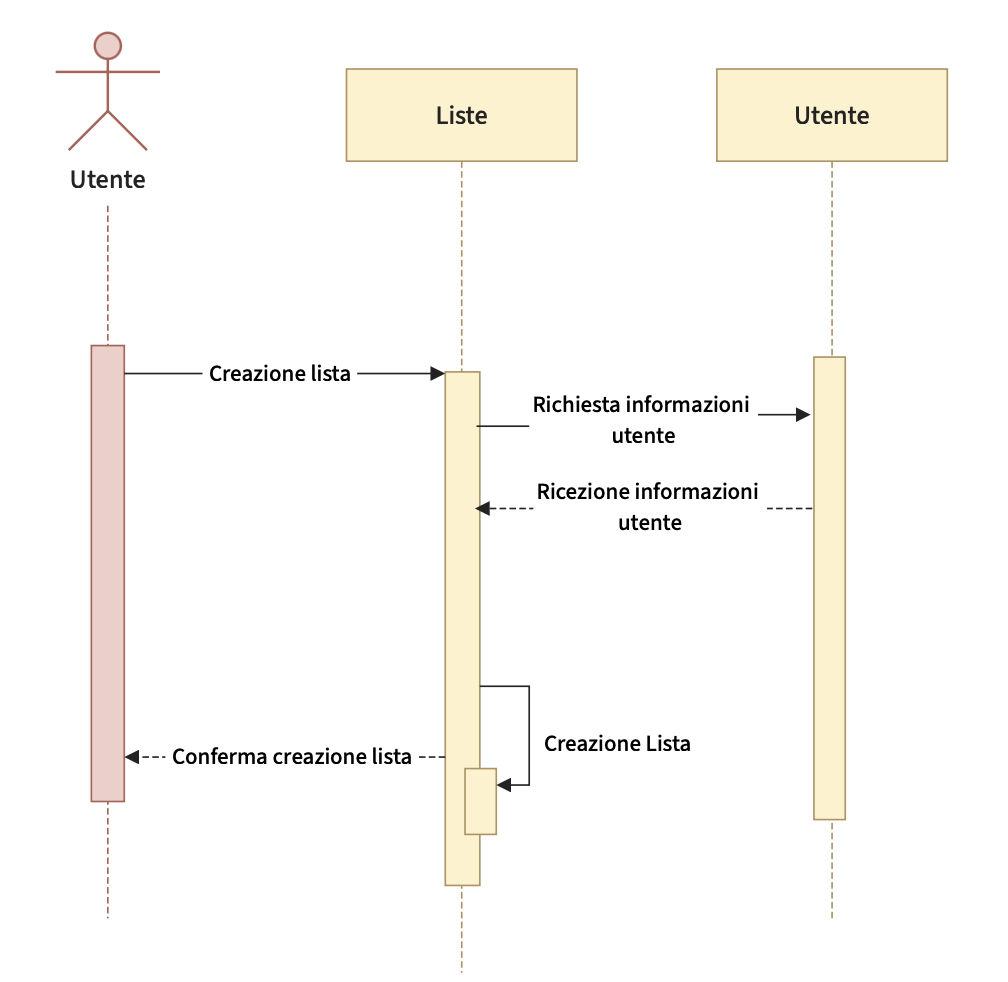
## Diagrammi di sequenza

Qui di seguito saranno mostrati i diagrammi di sequenza che esplicitano e definiscono la logica di diversi scenari all’interno del sistema con delle specifiche sequenze di eventi. Verrà chiarita la sequenza temporale e degli oggetti coinvolti, e gli scambi che avverranno tra di essi. Ogni diagramma analizzerà una casistica al cui interno sarà presente una collaborazione tra più oggetti per svolgere un compito.

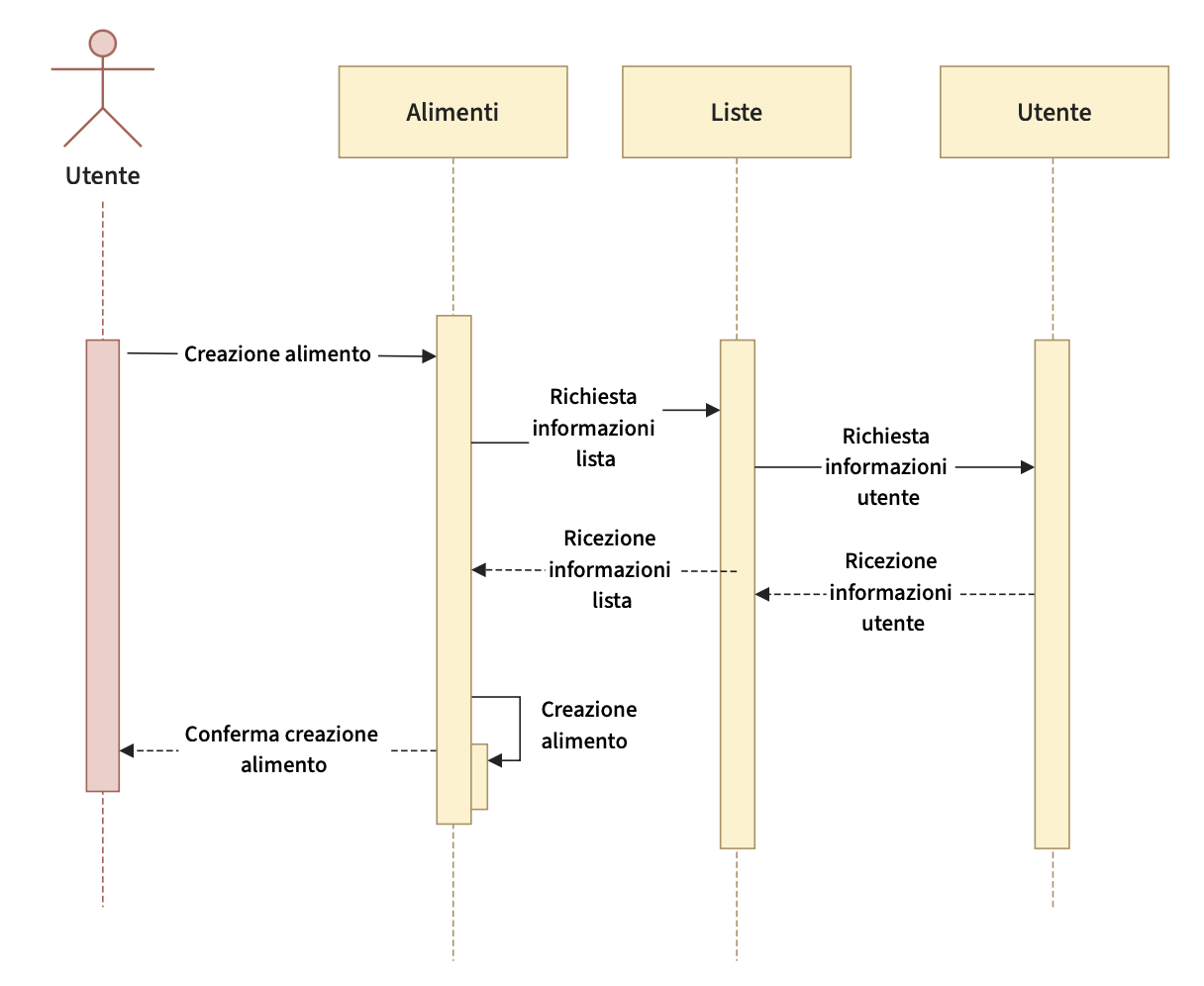
### ESISTENZA/CREAZIONE UTENTE



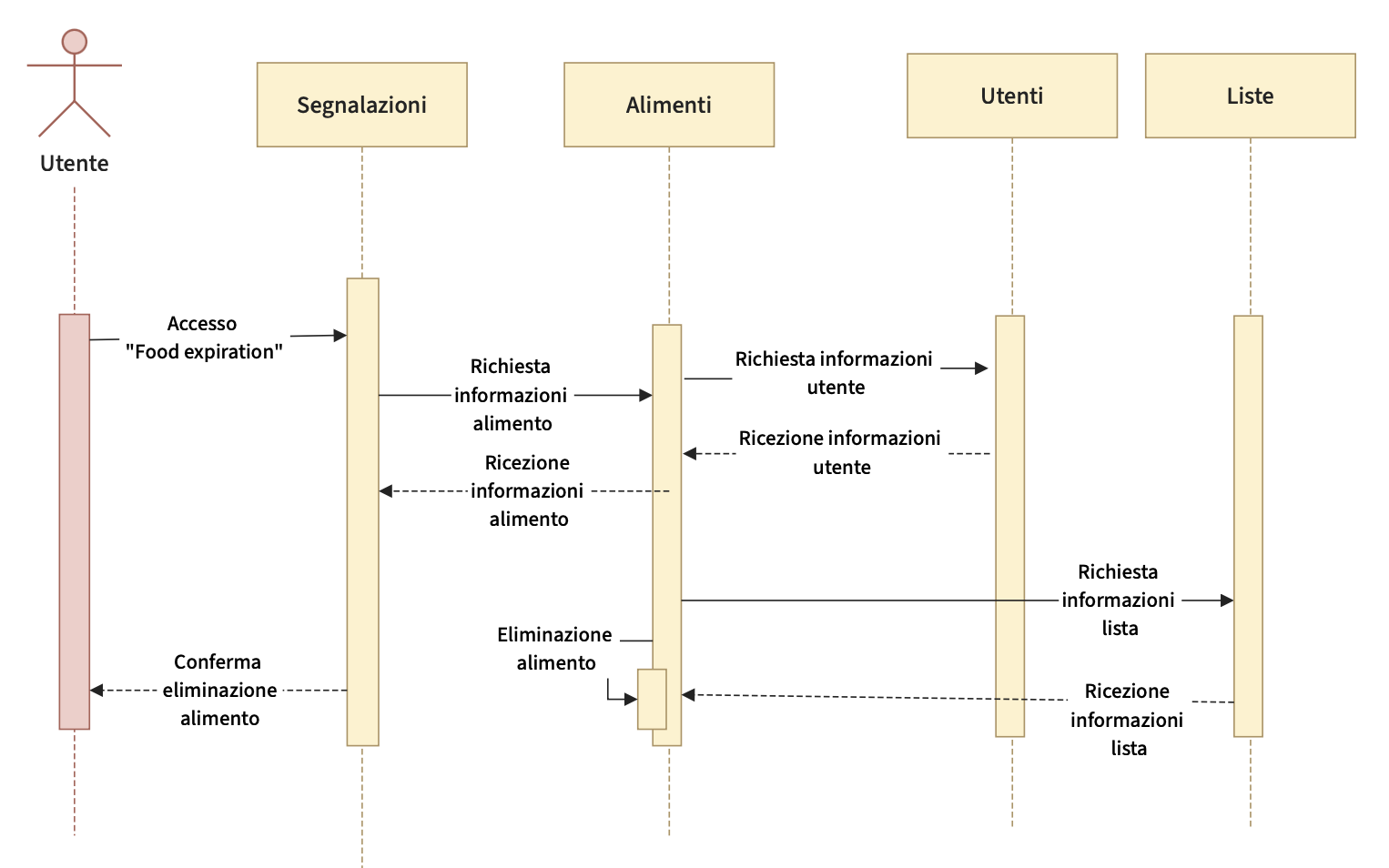
### CREAZIONE LISTA



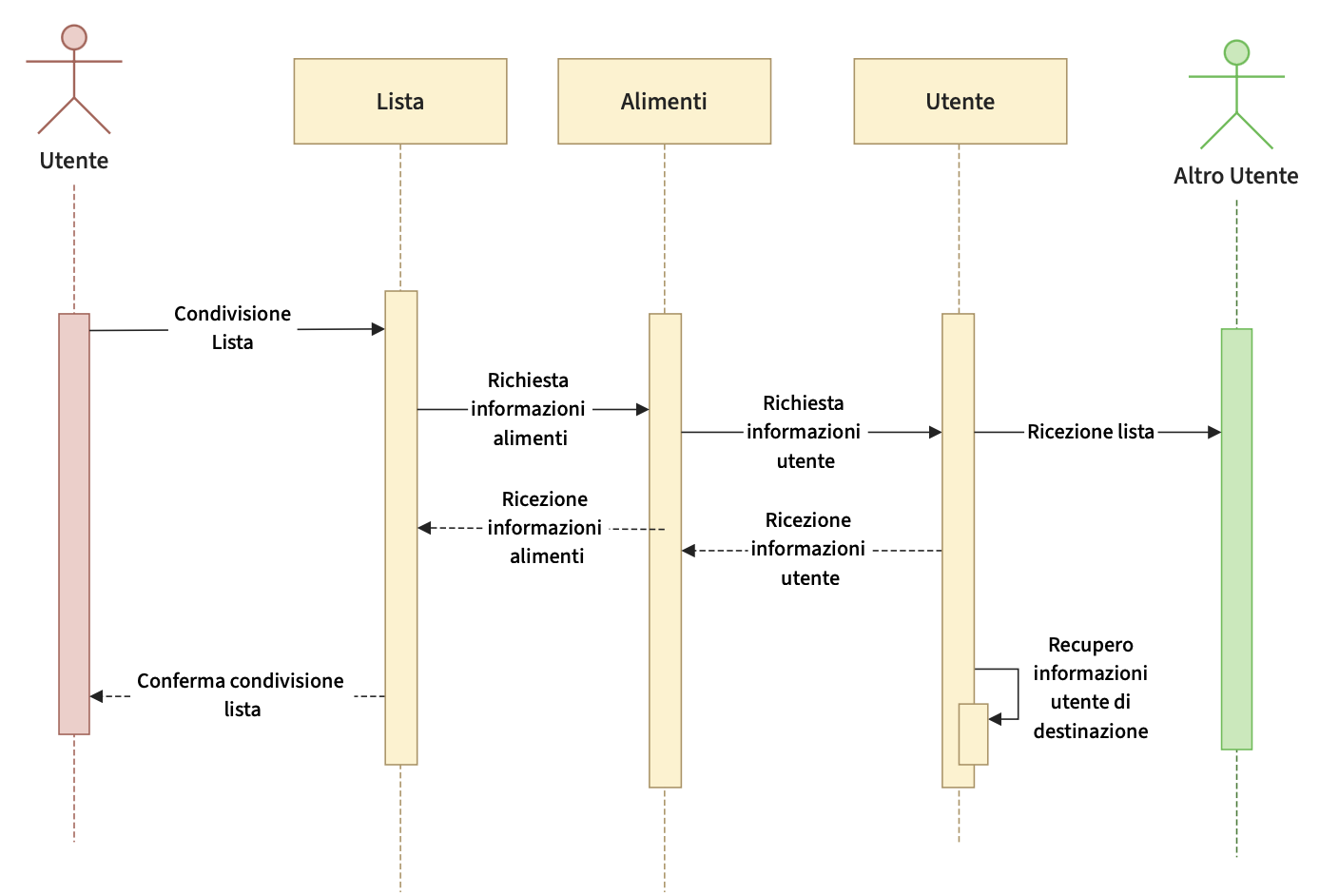
### AGGIUNTA ALIMENTO



### ELIMINAZIONE ALIMENTO SCADUTO



### CONDIVISIONE LISTA



# Progettazione dell’interfaccia Utente

In questo capitolo verranno mostrate a scopo informativo delle possibili schermate da cui sarà composta l’applicazione con una breve descrizione di ciascuna di esse.

Le schermate che compongono la nostra applicazione sono le seguenti:

* schermata di caricamento iniziale
* schermata iniziale / accesso alle liste
* vista degli ingredienti
* accesso utente
* registrazione utente
* pagina del profilo

## Schermata di caricamento iniziale

Questa schermata apparirà all’avvio dell’applicazione. Contiene il simbolo dell’applicazione, il titolo ed una descrizione dell’app.

## Schermata iniziale / accesso alle liste

Questa è la schermata principale dell’app. Contiene alcune liste speciali, ovvero ‘alimenti in scadenza’ e ‘alimenti scaduti’. A queste due liste di default, l’utente può aggiungere altre liste che permetteranno poi di aggiungere i vari alimenti presenti nella propria dispensa.

## Vista degli ingredienti

Vista degli ingredienti presenti in una determinata lista.

Ci sarà una lista dedicata agli ingredienti a breve scadenza ed una per quelli scaduti.

Le voci conterranno il nome dell’alimento, la quantità e la data di scadenza. è possibile aggiungere nuove voci premendo il pulsante ‘+’ in basso a destra. Premendo la voce sarà possibile correggere/aggiornare i campi. Sarà possibile aggiungerli alla lista degli alimenti importanti premendo la stella sulla destra. è presente un tasto cancella in alto a sinistra per eliminare direttamente la voce.

Nella barra degli strumenti in alto è presente un pulsante indietro, un pulsante per la condivisione della lista, ed un menù per le funzioni secondarie.

## Accesso e registrazione utente

Semplice schermata usata per far accedere l’utente al proprio account. Dopo che l’utente avrà fatto il Login non gli sarà più richiesto di entrare a meno che non venga effettuato il Logout.

Se non si è ancora registrati, dalla pagina di login si può accedere alla schermata che permette la registrazione.

Una volta che la registrazione è andata a buon fine, l’utente verrà reindirizzato alla schermata di homepage.

## A screenshot of a phone Description automatically generated

## 6.5. Pagina del profilo

Schermata dell’utente. Contiene il contatto e la possibilità di accedere alle notifiche delle scadenze, oltre che tornare alla pagina principale per visualizzare tutte le liste.

# Riferimenti

Per creare questo documento sono stati utilizzati come riferimento:

* ”Documento di progettazione” di alcuni gruppi degli anni passati;
* Documenti già stilati dal team (“Piano di progetto” e “Documento di analisi e specifica”)
* Template del “Documento di progettazione” messo a disposizione dal professore
* <https://moqups.com/it/templates/diagrams-flowcharts/uml-diagrams/>