

Brew Day!  
Ingengeria del Software  
aa 2021-2022

Tasca Alessandro, Lenzini Giacomo, Bonfanti Samuele, Dedo' Shana

31 Gennaio 2022

# Indice

<b>1</b>	<b>Organizzazione</b>	<b>3</b>
1.1	Diagramma di Gantt . . . . .	3
1.1.1	Iniziale . . . . .	3
1.1.2	Finale . . . . .	3
<b>2</b>	<b>Analisi</b>	<b>4</b>
2.1	Requisiti Funzionali . . . . .	4
2.2	Attori . . . . .	4
2.3	Casi d'uso . . . . .	5
2.3.1	Diagramma dei casi d'uso . . . . .	6
2.4	Modello di Dominio . . . . .	7
2.5	Diagramma di sequenza di sistema . . . . .	8
<b>3</b>	<b>Progettazione</b>	<b>9</b>
3.1	Modello E-R database . . . . .	9
3.2	Diagramma delle classi . . . . .	10
3.3	Diagramma di sequenza . . . . .	10
3.4	Diagramma macchina a stati . . . . .	11
3.5	Diagramma di attività . . . . .	12
3.6	Diagramma dell'architettura software . . . . .	13
<b>4</b>	<b>Implementazione</b>	<b>14</b>
4.1	Design Pattern . . . . .	14
4.1.1	Observer . . . . .	14
4.1.2	Singleton . . . . .	14
4.2	Pattern Architeturali . . . . .	14
4.2.1	Model View Viewmodel . . . . .	14
4.3	Design Principle . . . . .	14
4.3.1	Acyclic Dependencies Principle (ADP) . . . . .	14
4.4	Antipattern strutturali . . . . .	14
4.5	Mockup . . . . .	15
4.6	Logo . . . . .	15
4.7	Palette . . . . .	15
<b>5</b>	<b>Tecnologie utilizzate</b>	<b>17</b>
5.1	Room . . . . .	17
5.2	Espresso . . . . .	17
5.3	Github . . . . .	17
5.4	Sonarcloud . . . . .	18
<b>6</b>	<b>Istruzioni</b>	<b>19</b>
6.1	Installazione . . . . .	19
6.2	Utilizzo . . . . .	19

# 1 Organizzazione

## 1.1 Diagramma di Gantt

### 1.1.1 Iniziale

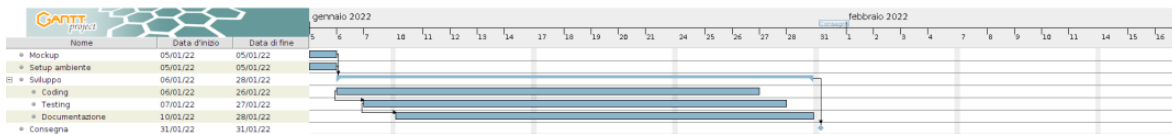


Figura 1: Diagramma di Gantt a inizio progetto

### 1.1.2 Finale

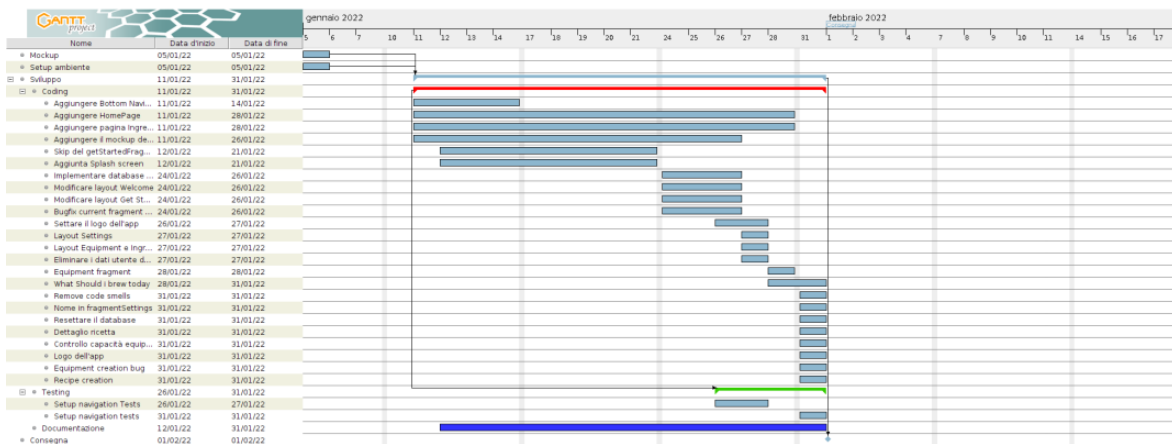


Figura 2: Diagramma di Gantt a fine progetto

## 2 Analisi

### 2.1 Requisiti Funzionali

Tipo	Descrizione
Funzionale	Mantenimento lista ingredienti
Funzionale	Aggiornamento lista ingredienti in base agli acquisti effettuati dall'utente
Funzionale	Aggiornamento lista in base ai consumi effettuati dall'utente per produrre una determinata ricetta
Funzionale	Mantenimento di una lista di ricette
Funzionale	Supporto creazione, modifica e cancellazione di ricette
Funzionale	Supporto creazione di note su ricette
Funzionale	Implementazione feature "What should I brew today?"
Funzionale	Le ricette devono essere soltanto di tipo "all grain"
Funzionale	L'equipaggiamento ha una specifica "batch size" (il max. numero di litri per singola run)
Funzionale	Le ricette possono contenere solo acqua, malto, luppolo, lievito, zucchero, additivi
Funzionale	Mantenimento cronologia delle ricette svolte con relative note di produzione
Funzionale	Mantenimento dati sulla capacità della strumentazione dell'utente

Tabella 1: Tabella riassuntiva dei requisiti identificati

### 2.2 Attori

Attore	Obiettivo
Utente	CURD ricette aggiungere ingredienti aggiungere equipaggiamento

Tabella 2: Tabella riassuntiva attori

## 2.3 Casi d'uso

Caso d'uso	Descrizione
RegistrazioneUtente	L'attore Utente deve registrarsi con il nome in modo da poter utilizzare l'app
ModificaUtente	L'attore Utente può modificare i propri dati(nome)
EliminazioneUtente	L'attore Utente può eliminare i propri dati dall'app
AggiungiRicetta	L'attore Utente può creare una ricetta nel sistema
ModificaRicetta	L'attore Utente può modificare i dettagli relativi a una ricetta presente nel sistema
EliminaRicetta	L'attore Utente può eliminare una ricetta dal sistema
VisualizzaCronologia	L'attore Utente può visualizzare l'elenco della cronologia di utilizzo di una singola ricetta e delle sue note
VisualizzaRicetta	L'attore Utente può visualizzare le ricette presenti del sistema
AggiungiIngrediente	L'attore Utente può aggiungere un ingrediente nel sistema
AggiungiEquipaggiamento	L'attore Utente può aggiungere un equipaggiamento nel sistema
WhatShouldIbrewToday?	L'attore Utente può chiedere al sistema di proportgli una ricetta da realizzare massimizzando gli ingredienti disponibili

Tabella 3: Tabella riassuntiva casi d'uso

### 2.3.1 Diagramma dei casi d'uso

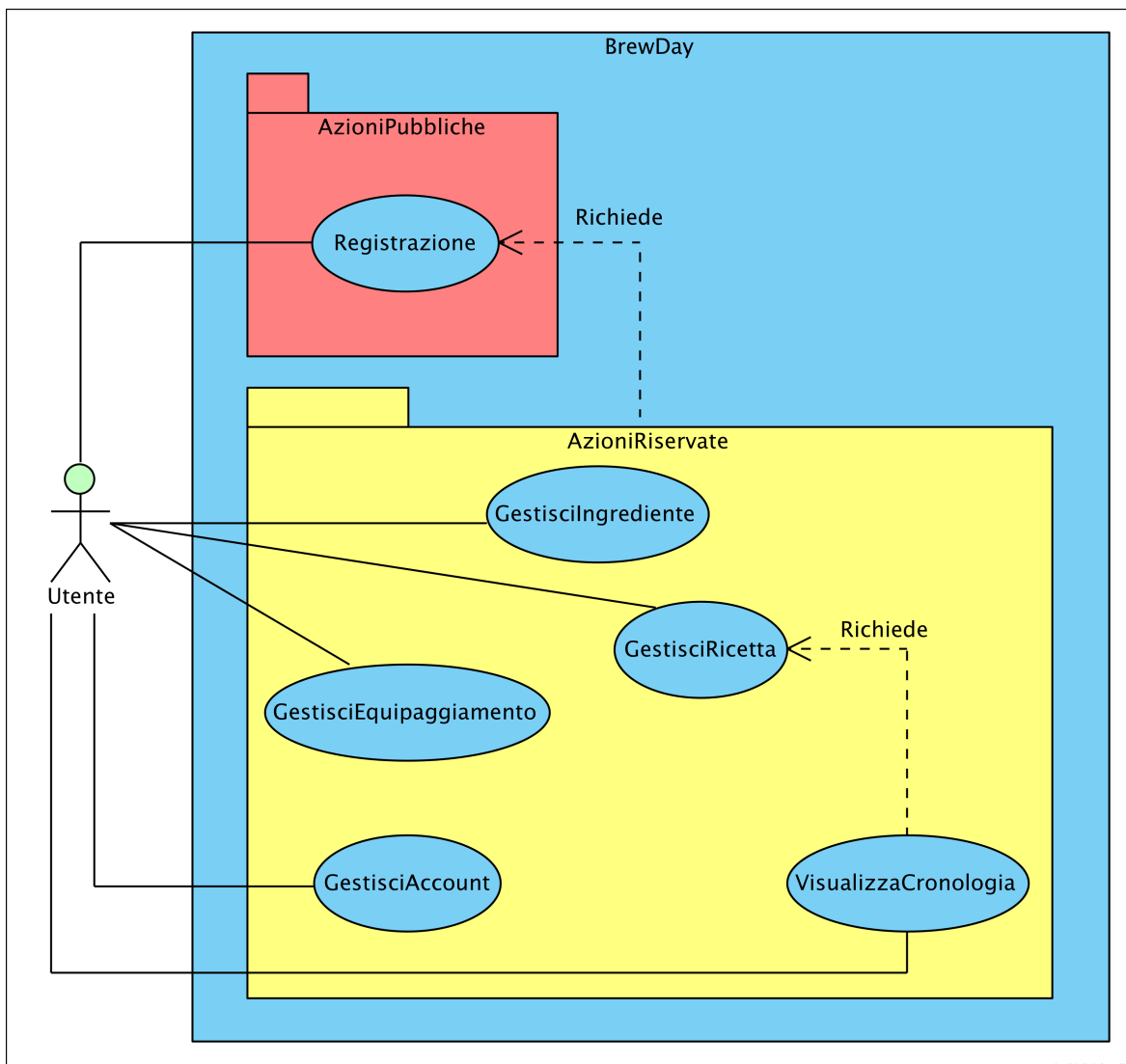


Figura 3: Diagramma dei casi d'uso

## 2.4 Modello di Dominio

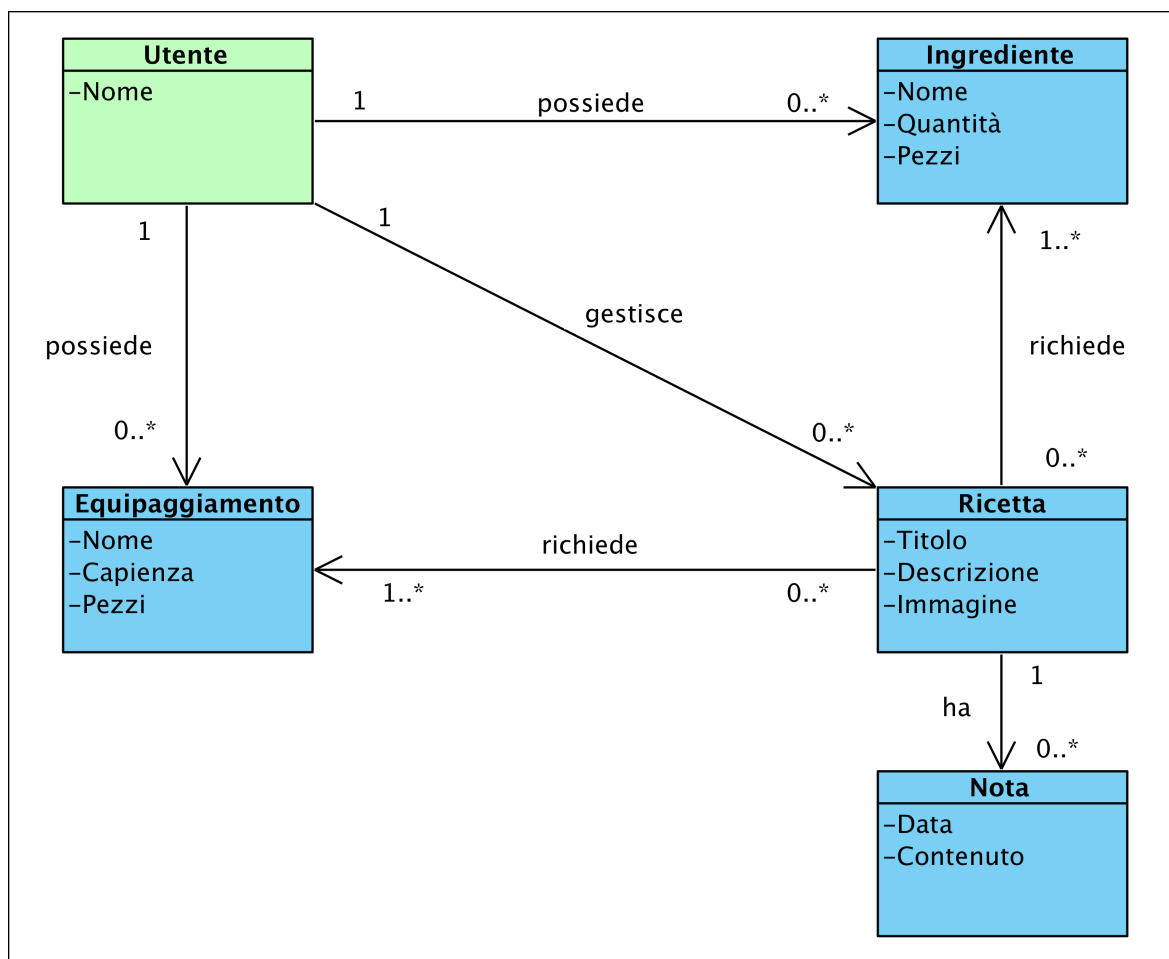


Figura 4: Modello di Dominio

## 2.5 Diagramma di sequenza di sistema

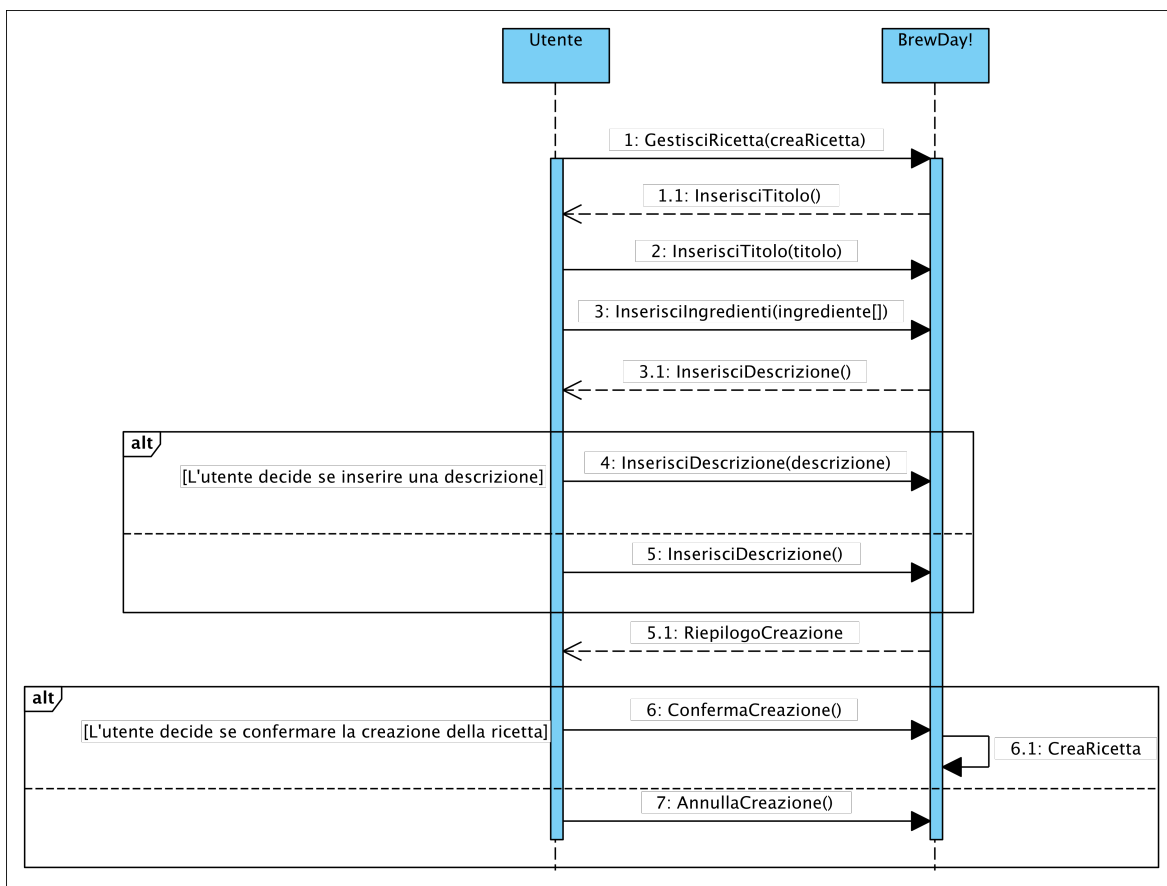


Figura 5: Diagramma di sequenza di sistema per addRecipe



### 3 Progettazione

#### 3.1 Modello E-R database

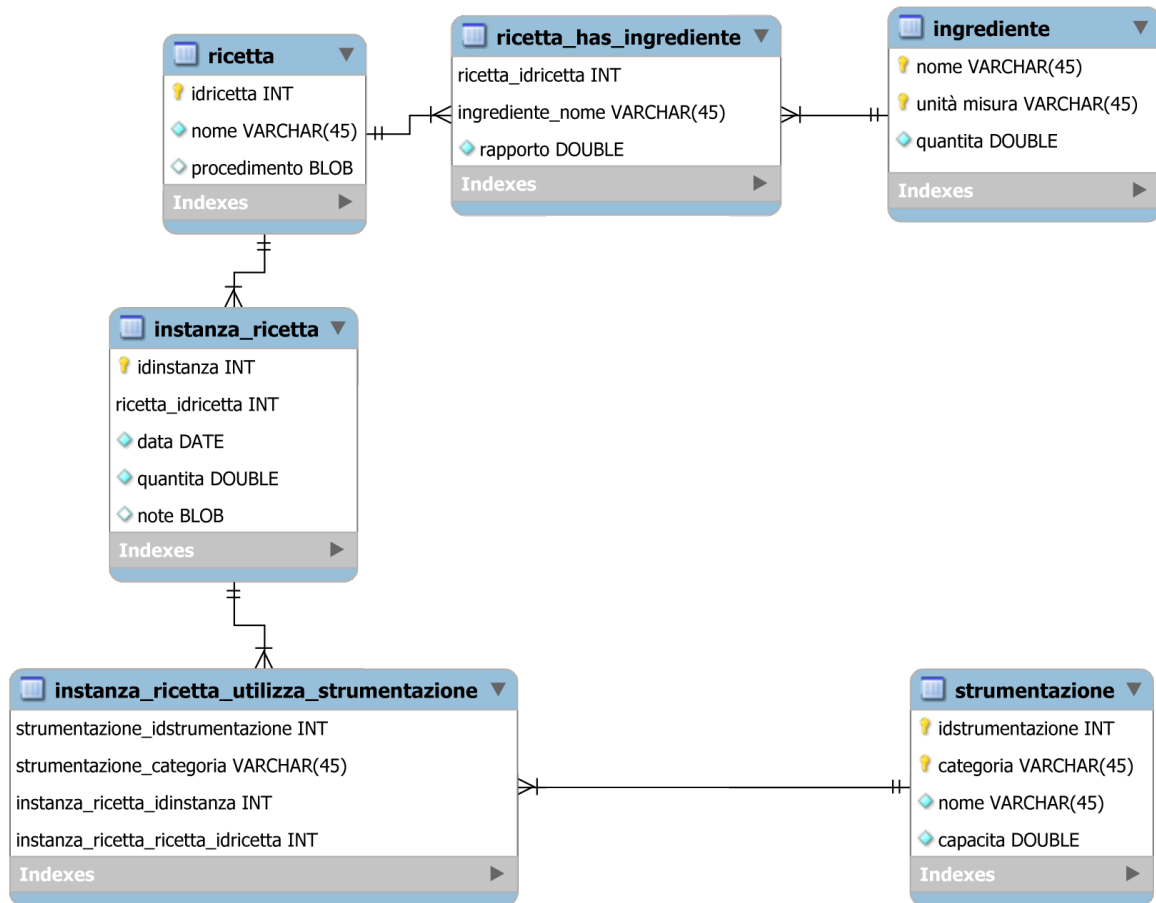


Figura 6: Modello E-R raffigurante la struttura del database

### 3.2 Diagramma delle classi

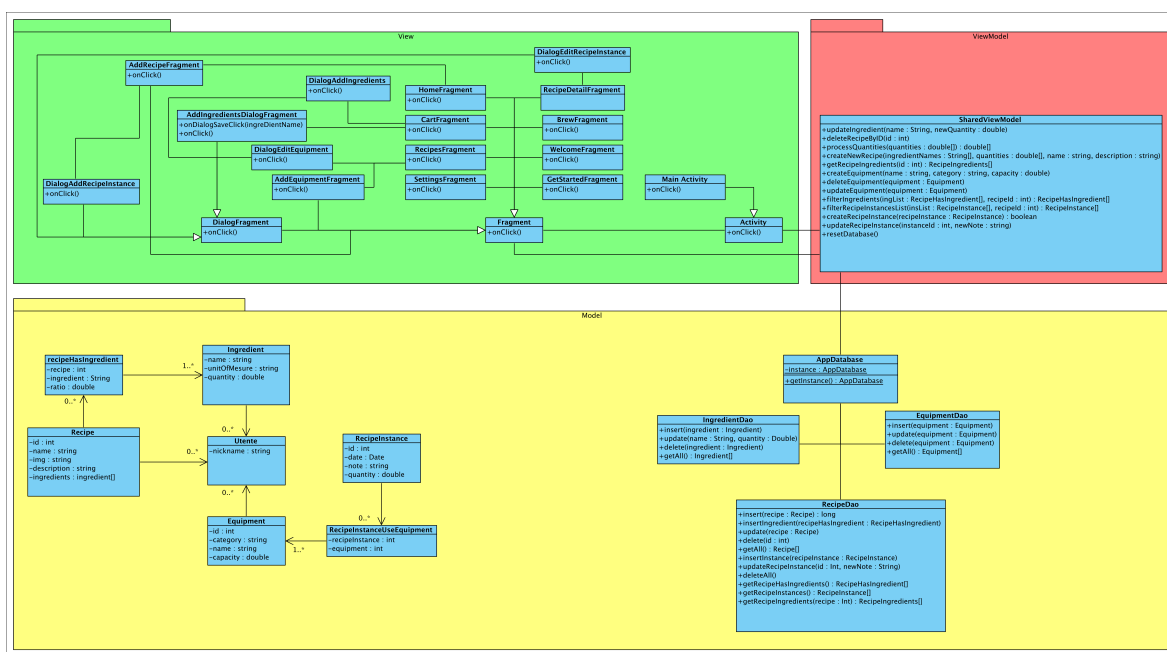


Figura 7: Diagramma delle classi

### 3.3 Diagramma di sequenza

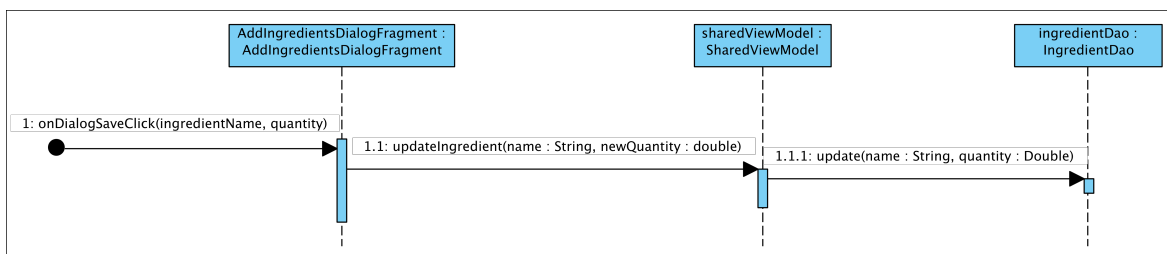


Figura 8: Diagramma di sequenza per addIngredient

### 3.4 Diagramma macchina a stati

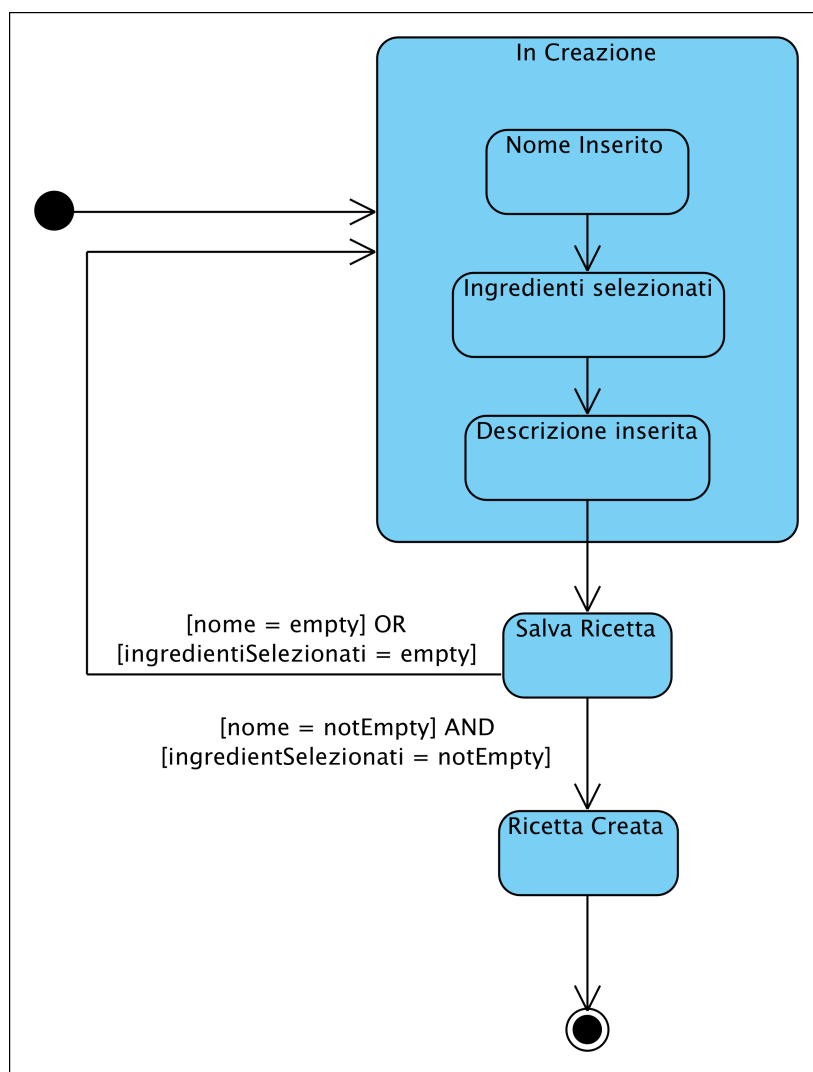


Figura 9: Diagramma macchina a stati per `addRecipe`

### 3.5 Diagramma di attività

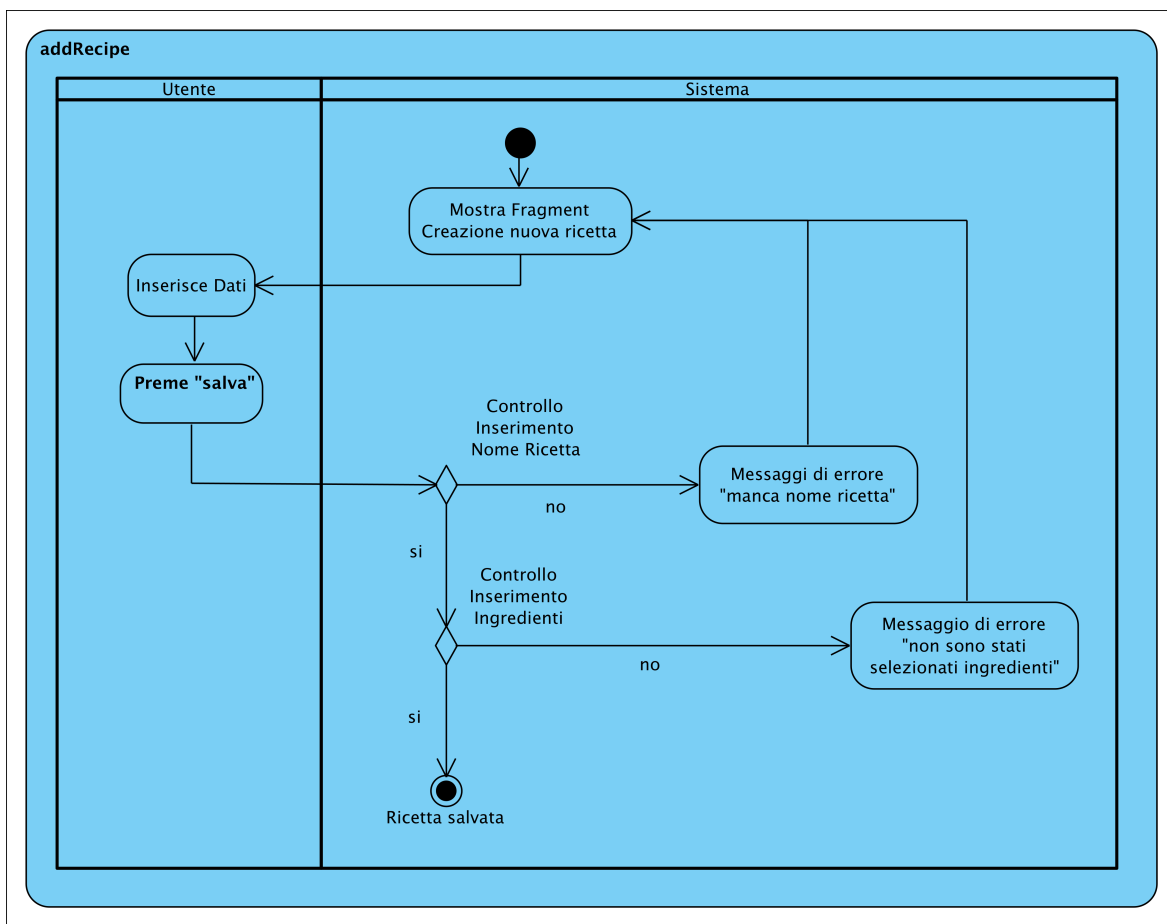


Figura 10: Diagramma di attività per addRecipe

### 3.6 Diagramma dell'architettura software

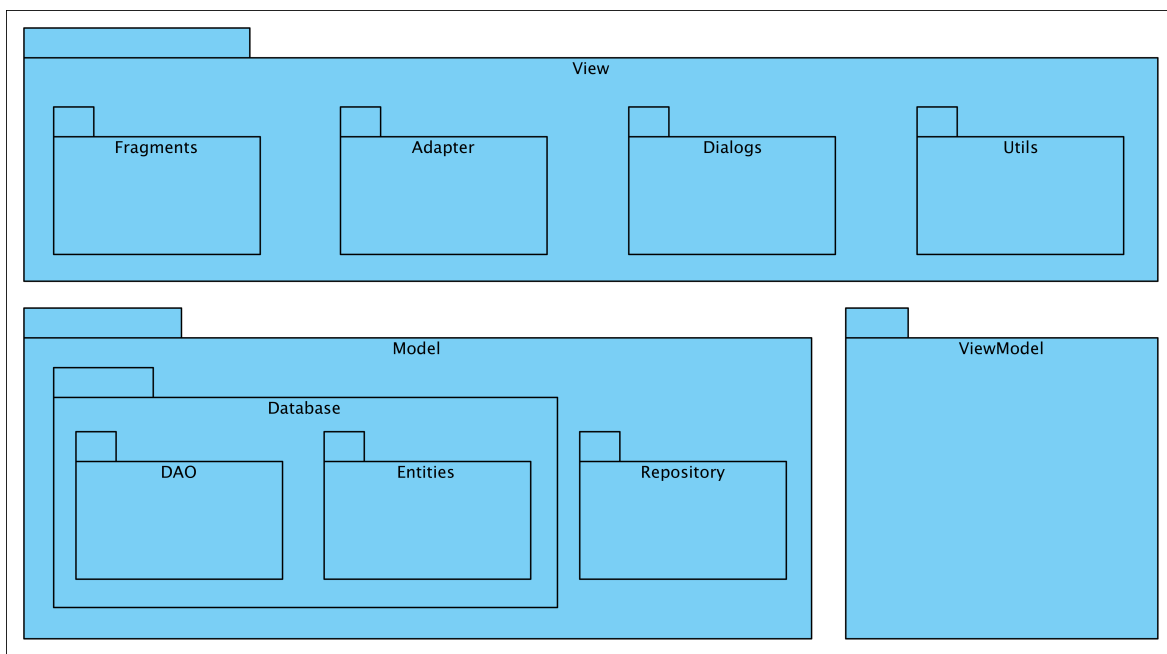


Figura 11: Diagramma dell'architettura software

## 4 Implementazione

### 4.1 Design Pattern

I design pattern utilizzati fanno riferimento a [Mar00]

#### 4.1.1 Observer

Il pattern Observer è presente tramite l'utilizzo di LiveData. Oltre a notificare le view degli aggiornamenti relativi a ricette, ingredienti ed equipaggiamento, i LiveData sono lifecycle-aware, ovvero rispettano il ciclo di vita delle varie componenti dell'applicazione (come activity o fragment) andando ad aggiornare solamente le componenti in uno stato attivo.

#### 4.1.2 Singleton

Il pattern Singleton è stato utilizzato per avere una singola istanza del database all'interno dell'applicazione.

### 4.2 Pattern Architetture

I pattern architetturali utilizzati fanno riferimento a quelli presentati da Fowler [Fow12].

#### 4.2.1 Model View Viewmodel

Il pattern architetturale MVVM è una variante del pattern Presentation Model Design di Martin Fowler [Fow12]. Si compone di tre parti principali: il model (modello), la view (vista) e il view model (modello della vista) a cui si accosta il meccanismo di binding per mantenere sincronizzati questi ultimi due. Nello sviluppo dell'applicazione questo pattern è stato di fondamentale importanza per gestire la logica delle view attraverso il view model.

### 4.3 Design Principle

I design principle utilizzati fanno riferimento a [Mar00]

#### 4.3.1 Acyclic Dependencies Principle (ADP)

È stato seguito questo design principle con l'intento di mantenere i package privi di cicli.

### 4.4 Antipattern strutturali

Non sono stati riscontrati Antipattern strutturali.

## 4.5 Mockup

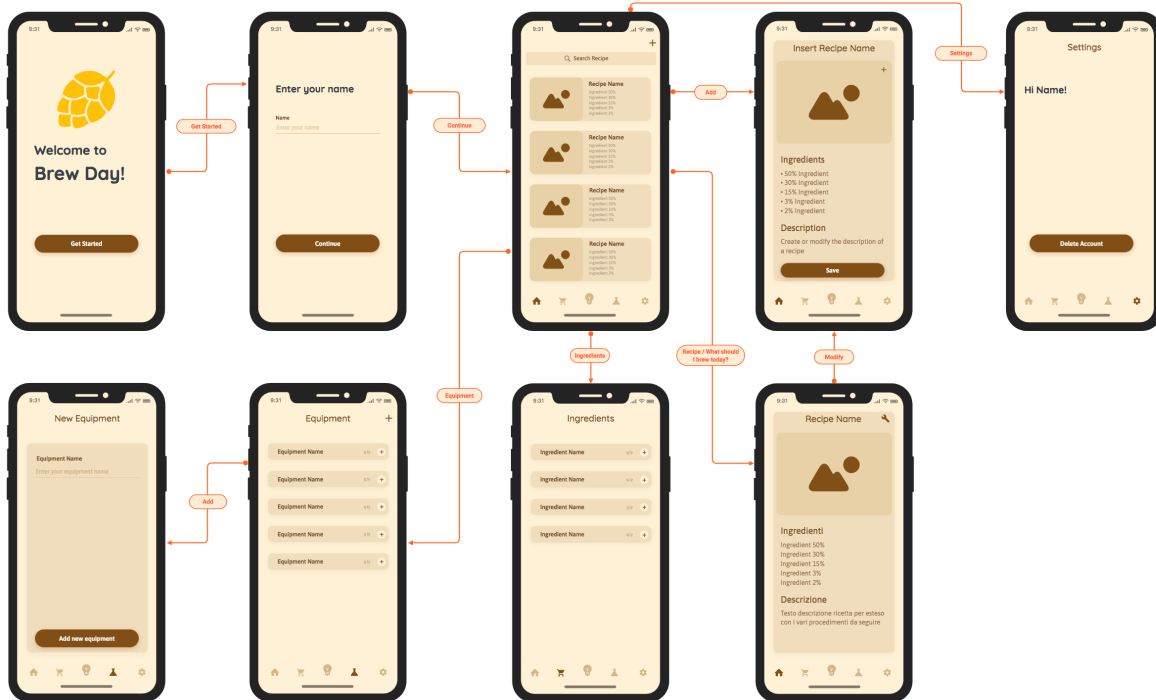


Figura 12: Mockup iniziale dell'applicazione

## 4.6 Logo

Per il logo dell'applicazione è stata scelta l'immagine di un luppolo stilizzato.



Figura 13: Logo dell'applicazione

## 4.7 Palette

A partire dal logo è stata definita la palette di colori utilizzati, prediligendo tonalità chiare per lo sfondo e tonalità scure per il testo.

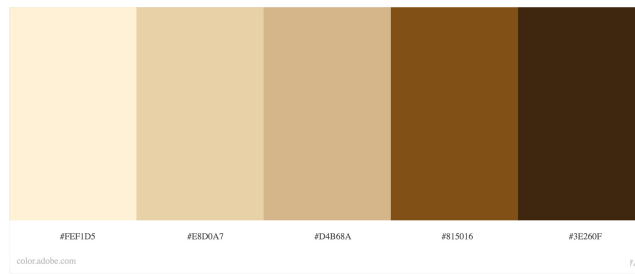


Figura 14: Palette scelta per l'applicazione



## 5 Tecnologie utilizzate

### 5.1 Room

L'utilizzo della libreria Room ha permesso un facile mantenimento del database. In particolare sono state create le diverse Entità e i vari DAO (data access object) ad esse associati.

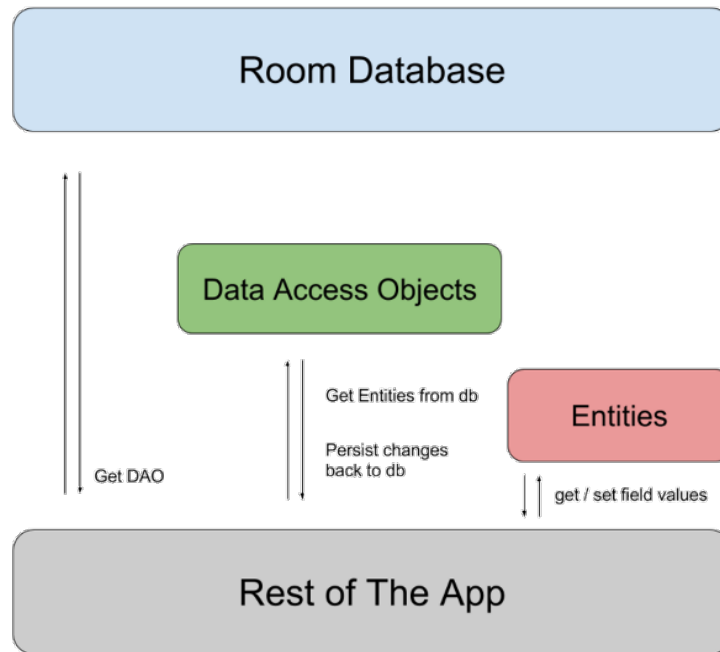


Figura 15: Schema sintetico sul funzionamento di Room

### 5.2 Espresso

Espresso è un framework usato nella creazione di test automatizzati per l'interfaccia utente (UI) di Android che fa parte dell'Android Testing Support Library.

I test Espresso possono essere eseguiti sul device effettivo oppure su un emulatore, simulando il reale comportamento di un utente.

### 5.3 Github

Durante lo sviluppo del progetto sono state utilizzate diverse funzionalità offerte da github nell'ambito dell'organizzazione del lavoro. In particolare è stato utilizzato il sistema delle issue e delle pull request. Il primo per tenere traccia delle nuove features (alcune di queste sono state anche discusse, sempre all'interno della sezione issue) da aggiungere e dei bug riscontrati, mentre il secondo per fare la merge dei vari branch nel master con l'intento di produrre un lavoro più "pulito". In aggiunta si è cercato di utilizzare nomi dei branch e messaggi di commit inerenti alle modifiche apportate.

## 5.4 Sonarcloud

L'utilizzo del tool SonarCloud ha permesso di controllare la qualità del codice scritto, in particolare ha permesso di individuare e risolvere alcuni code smell minori. Il controllo di SonarCloud è stato applicato unicamente al branch master.

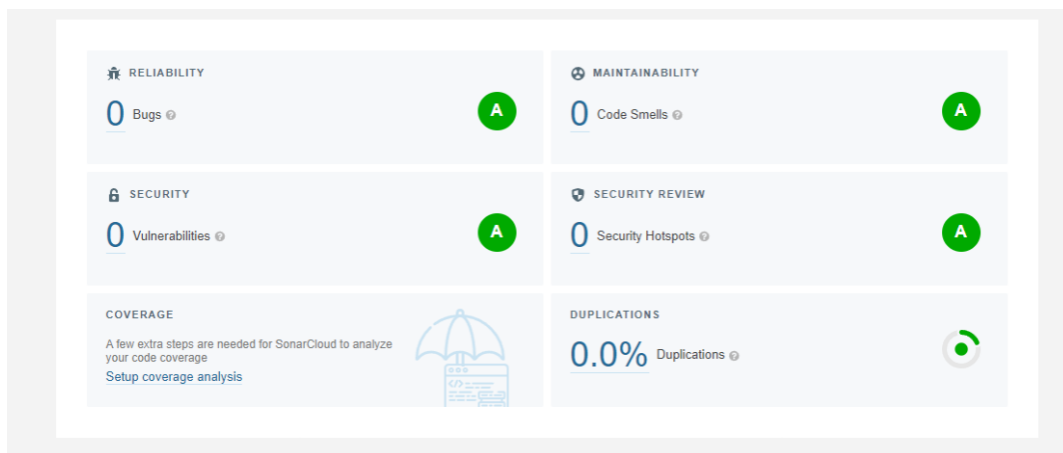


Figura 16: Analisi di SonarCloud in data 31.01.2022

## 6 Istruzioni

### 6.1 Installazione

È possibile installare il file eseguibile APK direttamente su un dispositivo Android (versione 8.0 o successive), dando in precedenza i permessi per poter installare applicazioni non provenienti dal PlayStore.

In alternativa è possibile usare direttamente l'emulatore presente all'interno dell'IDE Android Studio.

- Installare Android Studio
- Scaricare il codice dell'applicazione da git
  - Get from VCS
  - Inserire dati di accesso
  - Inserire link al repository  
`https://github.com/UnimibSoftEngCourse2022/progetto-birra-1-beer2beer.git`  
`git@github.com:UnimibSoftEngCourse2022/progetto-birra-1-beer2beer.git`
- Creare un device per l'emulatore
  - Tools > Device Manager
  - Virtual > Create device
  - Scegliere un dispositivo a scelta (scegliere una versione di android 8.0 o successive)
- Assicurarsi di aver selezionato il dispositivo appena creato e premere sulla freccia verde "play"

### 6.2 Utilizzo

L'applicazione è pronta all'uso.

Una volta installata e aperta verrà visualizzata una schermata di benvenuto. Procedendo verrà chiesto di inserire un nome, dopodiché ci si troverà nella homepage.

L'applicazione permette di aggiungere nuovi equipaggiamenti ed eventualmente modificarli (una volta creati) scegliendo per ognuno un nome, la categoria di appartenenza e la capacità in litri. Inoltre è possibile eliminare un equipaggiamento facendo swipe a sinistra.

Gli ingredienti presenti nell'applicazione sono predefiniti e inizialmente settati a 0. È possibile modificare manualmente il loro valore nell'apposita sezione.

Per quanto riguarda le ricette è possibile aggiungerne di nuove, selezionando un titolo, gli ingredienti e una descrizione facoltativa. Una volta creata una ricetta è possibile cancellarla dalla homepage facendo swipe a sinistra. Cliccando sulla ricetta, invece, è possibile visualizzare la ricetta completa di dettagli (ingredienti e descrizione) e le eventuali note aggiunte in precedenza. In particolare, ogni qual volta si aggiunge una nuova istanza della ricetta (si prepara una ricetta) è possibile scrivere una nota.

Premendo sul tasto centrale del menu verrà mostrata una ricetta che risponde alla funzionalità "What should I brew today?".

Infine, eliminando il proprio account dalle impostazioni, verranno eliminati tutti i dati presenti nel database dell'applicazione.

## Riferimenti bibliografici

- [Fow12] Martin Fowler. *Patterns of Enterprise Application Architecture: Pattern Enterpr Applica Arch.* Addison-Wesley, 2012.
- [Mar00] Robert C Martin. Design principles and design patterns. *Object Mentor*, 1(34):597, 2000.