

Brew Day

Requisiti funzionali

Ricette

R1: L'applicazione HomeBrew permetterà di creare una ricetta in qualunque momento, indipendentemente dalla presenza in magazzino degli ingredienti di cui necessita.

R2: L'applicazione HomeBrew permetterà di creare una ricetta caratterizzata dal nome della birra che permette di produrre, dal tempo necessario per poterla eseguire, dal procedimento da seguire, dagli ingredienti che servono per eseguirla, dalla percentuale, che indica la proporzione in termini assoluti di ogni ingrediente rispetto agli altri ingredienti della stessa ricetta e infine, dalle attrezzature che servono per poterla realizzare. Il nome della birra di una ricetta la identifica univocamente. Questo significa che non possono esistere più ricette che abbiano lo stesso nome.

R3 L'applicazione HomeBrew permetterà di modificare una ricetta in qualsiasi momento.

R4: L'applicazione HomeBrew permetterà di eliminare una ricetta salvata nel database in qualsiasi momento, specificandone il nome.

R5: L'applicazione HomeBrew permetterà di visualizzare dettagli di una specifica ricetta, dato il suo nome, in qualsiasi momento.

R6: L'applicazione HomeBrew permetterà di aggiungere una nota nel momento della creazione di una ricetta se il birraio ne ha necessità.

R7: L'applicazione HomeBrew permetterà di aggiungere una nota ad una ricetta già esistente e già salvata nel database.

R8: L'applicazione HomeBrew permetterà di modificare una ricetta, ovvero modificarne gli ingredienti associati, piuttosto che il procedimento, eliminando la ricetta in questione e creandone una nuova avente lo stesso nome. Questo processo è automaticamente gestito dall'amministratore che si occupa di controllare il database nel quale le ricette sono memorizzate.

R9: L'applicazione HomeBrew fornirà al birraio una feature chiamata "what should i brew today?" che permette di ottenere la ricetta (se esiste) che massimizza la quantità di birra che si può produrre, compatibilmente con gli ingredienti e le capacità delle attrezzature di cui il birraio dispone.

Ingredienti

R10: L'applicazione HomeBrew permetterà di aggiungere ingredienti nel database in qualunque momento;

R11: L'applicazione HomeBrew permetterà di aggiungere un ingrediente, caratterizzato dal nome, dalla quantità, dal tipo (Un ingrediente può essere di uno dei seguenti 5 tipi: Malto, luppoli, acqua, zucchero e lievito). Inoltre, un ingrediente può essere bloccato o meno. Un ingrediente bloccato è un ingrediente non

Gruppo: Birra 1

Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

disponibile nella scorta del birraio. Il nome di ogni ingrediente lo identifica univocamente. Questo significa che non possono essere memorizzati più ingredienti aventi lo stesso nome.

R12: L'applicazione HomeBrew permetterà di modificare uno specifico ingrediente, precedentemente salvato nel database, in qualsiasi momento;

R13: L'applicazione HomeBrew permetterà di eliminare un ingrediente dal database, specificandone il nome, solo se è stato inserito in precedenza;

R14: L'applicazione HomeBrew permetterà di mostrare gli ingredienti aggiunti nel sistema, dato il loro nome.

Attrezzatura

R15: L'applicazione HomeBrew permetterà di definire l'attrezzatura in possesso del birraio e che è necessaria per eseguire le ricette.

R16: L'applicazione HomeBrew permetterà di registrare nel database tre tipi di attrezzatura: tubi, fermentatori e cisterne. Ogni attrezzatura è caratterizzata dalla portata massima e da un nome che la identifica univocamente.

R17: L'applicazione HomeBrew permetterà al birraio di associare a ogni ricetta una o più attrezzature necessarie per poterla eseguire.

Note

R18: L'applicazione HomeBrew permetterà di associare ad ogni ricetta al più una nota, ovvero un breve documento che specifica delle caratteristiche della ricetta a cui è associata. Attraverso le note si possono specificare eventuali difetti, piuttosto che criticità relativi ad una ricetta;

R19: L'applicazione HomeBrew permetterà di associare ogni nota ad una sola ricetta e ogni ricetta ha al più una sola nota;

R20: L'applicazione HomeBrew permetterà di aggiungere una nota ad una ricetta già salvata nel sistema.

R21: L'applicazione HomeBrew permetterà di aggiungere una nota ad una ricetta nel momento stesso della creazione della ricetta stessa.

R22: L'applicazione HomeBrew permetterà di visualizzare il titolo di una nota, oltre che la sua descrizione specificando il nome della ricetta a cui la nota è associata.

R23: Ogni nota è caratterizzata da un titolo e da una descrizione, che rappresenta il corpo della nota.

R24: L'applicazione homeBrew permetterà di modificare la nota associata ad una ricetta.

Requisiti non funzionali

R25: L'applicazione dovrà essere disponibile 24 ore al giorno e 365 giorni all'anno.

R26: Il database dell'applicazione HomeBrew dovrà essere installato dal birraio al momento dell'installazione dell'applicazione. Tale procedimento sarà descritto in un apposito file che comporrà la documentazione.

R27: L'applicazione dovrà eseguire la connessione al database entro 5 secondi.

R30: L'applicazione HomeBrew sarà scritta in Java e MySQL per funzionare su più sistemi operativi.

R31: L'applicazione HomeBrew dovrà avere il database sullo stesso hardware dell'applicazione.

R32: L'applicazione deve essere pronta e disponibile per il 31 Gennaio 2020.

R33: L'applicazione deve essere facilmente mantenibile.

R34: L'applicazione non deve presentare troppi bug o code smell.

R35: L'applicazione non deve presentare troppe duplicazioni.

R36: L'applicazione deve gestire gli errori senza andare in crash.

R37 : Ogni membro del gruppo deve avere comprensione di ogni parte del codice.

R38: L'applicazione deve rispondere alle richieste in tempi accettabili

Scelte implementative

- Abbiamo supposto che l'attrezzatura di cui dispone il birraio casalingo non può cambiare nel corso del tempo.
- Abbiamo deciso di non permettere la creazione, la modifica e l'eliminazione delle attrezzature di cui dispone il birraio. Infatti, abbiamo immaginato il seguente scenario: Il birraio casalingo commissiona la creazione dell'applicazione HomeBrew al gruppo birra1. Come descritto nella prima scelta implementativa, sappiamo che la sua attrezzatura non potrà cambiare nel corso del tempo, di conseguenza possiamo fornire al birraio il database contenente tutti gli strumenti di cui lui effettivamente dispone.
- Abbiamo supposto che un ingrediente non può essere eliminato se esiste almeno una ricetta che lo contiene. Il birraio, tuttavia, potrà decidere di rendere non disponibile tale ingrediente semplicemente tramite un'operazione di modifica (funzionalità supportata dall'applicazione).
- Abbiamo supposto che una ricetta può contenere al più una nota.

Casi d'uso in formato breve

Attori

- Birraio → Ovvero colui che commissiona l'applicazione al nostro gruppo di lavoro. È colui che vuole tenere traccia delle ricette, degli ingredienti e delle attrezzature. Il birraio inoltre, vuole avere la possibilità di sapere quale è (se esiste) la ricetta, che conformemente con la disponibilità degli

ingredienti e delle capacità delle sue attrezzature, permette di massimizzare la quantità di birra producibile.

- Amministratore → è l'elemento del sistema che si occupa della gestione del database.

Aggiungi ricetta

Il birraio dopo aver avviato l'applicazione può decidere di aggiungere al database una nuova ricetta indicando i vari ingredienti e le attrezzature da utilizzare tra quelle memorizzate nel database. Può decidere di aggiungere una nota per eventuali annotazioni riguardanti la ricetta. Dopo averla aggiunta, l'applicazione mostra un messaggio che conferma che la ricetta è stata inserita correttamente nel database.

Modifica ricetta

Il birraio dopo aver avviato l'applicazione può decidere di modificare una ricetta precedentemente salvata nel sistema. Per fare questo inserirà il nome della ricetta che vuole modificare, gli ingredienti che le sono associati, la strumentazione che serve per poterla eseguire, il tempo, il procedimento ed eventualmente il titolo e la descrizione della nota che il birraio vuole associare alla ricetta. Se l'aggiornamento va a buon fine, allora verrà mostrato un messaggio che conferma l'avvenuta modifica.

Elimina ricetta

Il birraio vuole eliminare una delle ricette inserite nel database. Il birraio inserisce il nome della ricetta che vuole eliminare e clicca sul bottone denominato "elimina ricetta". Il sistema chiede conferma che il birraio voglia effettivamente eliminare quella ricetta, il birraio clicca sul pulsante conferma. Il sistema, se il nome inserito dall'utente coincide con il nome di una ricetta memorizzata nel database, la elimina, altrimenti non esegue alcuna operazione.

Proponi ricetta

Il birraio una volta avviata l'applicazione clicca sul bottone denominato "consigliami una ricetta". Il sistema controlla quale sia la ricetta che massimizzi la quantità di birra che permette di produrre (definita come la somma degli ingredienti associati alla ricetta), compatibilmente con la disponibilità degli ingredienti e la capacità delle attrezzature che ha a disposizione il birraio. Se tale ricetta esiste allora ne vengono mostrati i dettagli, compresa la quantità di birra che permette di produrre, altrimenti viene mostrato un messaggio che indica il fatto che non c'è nessuna ricetta disponibile dati gli ingredienti e la capacità delle attrezzature disponibili.

Aggiungi ingrediente

Il birraio una volta avviata l'applicazione, vuole aggiungere nel database nuovi ingredienti. Tramite l'interfaccia relativa agli ingredienti inserisce il nome, specifica la quantità, sceglie il tipo tra malto, zucchero, luppoli, acqua e lievito, indica se l'ingrediente è bloccato o meno e clicca sul bottone apposito per aggiungere un ingrediente. Se i campi sono stati compilati correttamente dal birraio, esso riceverà un messaggio di

Gruppo: Birra 1

Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

conferma dell' avvenuto inserimento dell'ingrediente nel database, altrimenti riceverà un messaggio di errore.

Aggiorna ingredienti

Il birraio, dopo aver avviato l'applicazione, può decidere di aggiornare gli ingredienti salvati nel database. Per fare questo utilizzerà l'apposita funzionalità dell'applicazione che gli permetterà di modificarli.

Modifica ingrediente

Il birraio dopo aver avviato l'applicazione può decidere di modificare gli ingredienti precedentemente inseriti nel database. Per fare questo inserisce il nome dell'ingrediente che vuole modificare, poi ne specifica il tipo, indica se è bloccato o meno e la quantità, dopo di che clicca sull'apposito bottone che gli permette di modificare l'ingrediente. Se l'utente inserisce correttamente i dati dell'ingrediente allora riceverà un messaggio che conferma la modifica dell'ingrediente, altrimenti riceverà un messaggio di errore.

Elimina ingrediente

Il birraio dopo aver avviato l'applicazione, decide di eliminare un ingrediente. Indica nell'interfaccia dedicata il nome dell'ingrediente e lo elimina cliccando l'apposito bottone. Per eseguire questa azione però è necessario che tale ingrediente sia già presente nel database e che non sia associato ad alcuna ricetta salvata. Se l'ingrediente specificato è già presente nel database e nessuna ricetta gli è associata, allora verrà eliminato, altrimenti non verrà eseguita alcuna azione.

Aggiungi nota

Il birraio vorrebbe aggiungere una nota relativa ad una particolare ricetta memorizzata nel sistema. Il birraio indica il nome della ricetta nell'apposita casella di testo e scrive il titolo e la descrizione della nota, clicca sull'apposito bottone per aggiungere una nuova nota e se tutto è andato a buon fine riceve un messaggio di riscontro positivo.

Aggiungi attrezzatura

Gli sviluppatori dell'applicazione, al momento della creazione dell'applicazione, memorizzano nel database le attrezzature di cui dispone il birraio.

Casi d'uso in formato dettagliato

Aggiungi nota

Portata	Applicazione HomeBrew
Livello	Obiettivo utente
Attore primario	Birraio
Parti interessate e interessi	<p>Birraio: vorrebbe aggiungere una nota relativa ad una certa ricetta, che ne specifica eventuali problematiche.</p> <p>Amministratore: aggiorna la ricetta memorizzata nel database, associandole la nota in questione.</p>
Precondizioni	<ul style="list-style-type: none"> La ricetta di riferimento deve essere già memorizzata nel database.
Garanzia di successo	Il sistema provvede alla creazione di una istanza della nota, che verrà automaticamente inserita nel database.
Scenario principale di successo	<ol style="list-style-type: none"> 1. Il birraio avvia l'applicazione. 2. Il birraio inserisce il nome della ricetta a cui vuole associare una nota; 3. Il birraio inserisce negli appositi campi di testo titolo e descrizione della nota; 4. Il sistema verifica che sia stato inserito un titolo e il testo della nota; 5. Il sistema esegue una connessione al database; 6. Il sistema verifica che la ricetta identificata dal nome specificato dal birraio sia presente nel database; 7. Il sistema crea una istanza della nota e la associa alla ricetta, modificando i campi titoloNota e descrizioneNota della tabella Ricetta del database; 8. Il sistema mostra un messaggio, che indica che l'operazione è andata a buon fine.
Estensioni	<ol style="list-style-type: none"> 1 Il birraio avvia l'applicazione. 2 Il birraio inserisce il nome della ricetta a cui vuole associare una nota; 3 Il birraio inserisce negli appositi campi di testo titolo e descrizione della nota; 4 Il sistema esegue una connessione al database; 5 Il sistema verifica che la ricetta identificata dal nome specificato dal birraio sia presente nel database;

	6 La ricetta non è presente nel database; 7 Viene mostrato un messaggio di errore.
Requisiti speciali	L'aggiunta della nota dovrà essere comunicata entro 5 secondi ogni volta che viene richiesta.
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizione	Ogni volta che un birraio vuole creare una nuova nota
Varie	

Elimina ricetta

Portata	Applicazione HomeBrew
Livello	Obiettivo utente
Attore principale	Birraio
Parti interessate e interessi	<ul style="list-style-type: none"> Birraio: vorrebbe eliminare una ricetta dal database Amministratore: deve eseguire la connessione al database ed eliminare la ricetta inserita.
Precondizioni	<ul style="list-style-type: none"> La ricetta deve essere memorizzata nel database.
Garanzia di successo	Il sistema ha provveduto all'eliminazione della ricetta. Il sistema rimane in uno stato consistente.
Scenario principale di successo	<ol style="list-style-type: none"> Il birraio avvia l'applicazione. Il birraio digita il nome della ricetta che vuole venga eliminata; Il birraio clicca sul bottone elimina ricetta; Il sistema chiede conferma che il birraio sia sicuro di voler eliminare la ricetta; Il birraio clicca sul bottone si; Il sistema esegue una connessione al database; Il sistema verifica che la ricetta in questione sia salvata nel database; Il sistema elimina la ricetta;
Estensioni	<ol style="list-style-type: none"> Il birraio avvia l'applicazione; Clicca sul pulsante elimina ricetta senza però scriverne il nome, oppure scrive il nome di una ricetta che non esiste; Il sistema non elimina alcuna ricetta;
Requisiti speciali	L'applicazione permette di eliminare una ricetta alla volta.

Gruppo: Birra 1

Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizione	Ogni volta che un birraio vuole eliminare una ricetta dal database.
Varie	

Aggiungi attrezzatura

Portata	Applicazione HomeBrew
Livello	Obiettivo utente
Attore principale	Amministratore
Parti interessate e interessi	<ul style="list-style-type: none">• Amministratore: deve inserire le attrezzature nel database
Precondizioni	Bisogna avere i permessi per aggiornare il database.
Garanzia di successo	L'amministratore ha memorizzato nel database le attrezzature in possesso del birraio.
Scenario principale di successo	<ol style="list-style-type: none">1. L'amministratore inserisce al momento della creazione del database, le attrezzature di cui dispone il birraio.
Estensioni	Nessuna
Requisiti speciali	Nessuno
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizione	Una volta sola, durante l'installazione dell'applicazione.
Varie	

Elimina Ingrediente

Portata	Applicazione HomeBrew
Livello	Obiettivo utente
Attore principale	Birraio

Parti interessate e interessi	<ul style="list-style-type: none"> • Birraio: vuole eliminare un ingrediente precedentemente inserito nel database. • Amministratore: deve connettersi al database per eliminare l'ingrediente specificato dal birraio.
Precondizioni	<ul style="list-style-type: none"> • L'ingrediente deve essere già stato memorizzato nel database in precedenza.
Garanzia di successo	<ul style="list-style-type: none"> • L'ingrediente è stato correttamente rimosso dal database • Il sistema rimane in uno stato consistente.
Scenario principale di successo	<ol style="list-style-type: none"> 1. Il birraio avvia l'applicazione; 2. Il birraio decide di eliminare un ingrediente; 3. Il birraio indica il nome dell'ingrediente che vuole eliminare; 4. Il sistema esegue una connessione con il database; 5. Il sistema verifica la presenza dell'ingrediente nel database; 6. Il sistema verifica che l'ingrediente non sia associato ad alcuna ricetta memorizzata nel database; 7. Il sistema elimina l'ingrediente.
Estensioni	<ol style="list-style-type: none"> 1. Il birraio entra nell'applicazione; 2. Il birraio decide di eliminare un ingrediente; 3. Il birraio non scrive il nome dell'ingrediente da eliminare, oppure scrive il nome di un ingrediente che non è stato precedentemente salvato nel database; 4. Il birraio clicca il bottone elimina ingrediente. 5. Non viene eliminato nessun ingrediente dal database.
Requisiti speciali	L'applicazione permette di eliminare un solo ingrediente per volta.
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizione	Ogni qualvolta il birraio decide di eliminare un ingrediente dal database.
Varie	

Proponi ricetta

Portata	Brew day
----------------	----------

Livello	Obiettivo utente
Attore primario	Birraio
Parti interessate e Interessi	<ul style="list-style-type: none"> Birraio: vuole conoscere la ricetta, se esiste, che massimizza la quantità di birra producibile, compatibilmente con gli ingredienti disponibili e con la capacità delle attrezzature del sistema.
Pre-condizioni	Sono memorizzati ingredienti disponibili e attrezzature di cui dispone il birraio nel database.
Garanzia di successo	I dettagli della ricetta che massimizza la quantità di birra (definita come somma delle quantità degli ingredienti che la compongono) sono mostrati al birraio.
Scenario principale di successo	<ol style="list-style-type: none"> 1. Il birraio avvia l'applicazione; 2. Il birraio clicca sul bottone denominato "consigliami una ricetta"; 3. Il sistema esegue una connessione al database; 4. Il sistema controlla che ci sia almeno una ricetta che abbia tutti gli ingredienti associati disponibili e che sia compatibile con l'attrezzatura a disposizione del birraio; 5. Il sistema ricerca la ricetta che massimizza la quantità di birra producibile; 6. Il sistema mostra i dettagli di tale ricetta;
Estensioni	Non esiste alcuna ricetta compatibile con gli ingredienti disponibili e/o con la capacità degli strumenti a disposizione del birraio. Viene mostrato un messaggio che indica che non ci sono ricette disponibili.
Requisiti speciali	Risposta alla feature entro 30 secondi il 90% delle volte.
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizione	Ogni qualvolta il birraio vuole accedere alla feature "What should i brew today?"
Varie	

Aggiungi ingrediente

Portata	Applicazione Brew day
Livello	Obiettivo utente
Attore primario	Birraio
Parti interessate e Interessi	<ul style="list-style-type: none"> Birraio: Vuole inserire un nuovo ingrediente nel database. Amministratore: si connette al database per aggiungere l'ingrediente.
Pre-condizioni	

Garanzia di successo (o post condizioni)	L'aggiunta dell'ingrediente è andata a buon fine e il database è stato aggiornato correttamente, rimanendo in uno stato consistente.
Scenario principale di successo	<ol style="list-style-type: none"> 1. Il birraio avvia l'applicazione; 2. Il birraio vuole aggiungere un nuovo ingrediente; 3. Il birraio indica il nome dell'ingrediente; 4. Il birraio indica la quantità; 5. Viene scelto il tipo di ingrediente fra malto, luppolo, acqua, zucchero e lievito; 6. Il birraio specifica se l'ingrediente è bloccato o meno; 7. Il birraio clicca sull'apposito bottone dedicato all'inserimento di un ingrediente; 8. Il sistema si connette con il database e aggiunge una riga alla tabella ingrediente. 9. Il sistema mostra un messaggio che conferma l'avvenuto inserimento dell'ingrediente.
Estensioni	<ol style="list-style-type: none"> 1. Il birraio avvia l'applicazione; 2. Il birraio vuole aggiungere un ingrediente; 3. Il birraio indica il nome dell'ingrediente; 4. Il birraio inserisce un input diverso da un numero nel campo quantità; 5. Il sistema mostra un messaggio di errore; 6. Il sistema non aggiunge l'ingrediente nel database.
Requisiti speciali	L'applicazione permette di aggiungere un ingrediente alla volta
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizione	Ogni qual volta si vuole aggiungere un ingrediente nuovo.
Varie	

Aggiorna ingrediente

Portata	Applicazione Brew today
Livello	Sotto-funzione
Attore primario	Birraio
Parti interessate e Interessi	<ul style="list-style-type: none"> • Birraio: vuole aggiornare la lista degli ingredienti disponibili. • Amministratore: gestisce l'aggiornamento dello stato del database
Pre-condizioni	L'ingrediente che si vuole aggiornare è salvato nel database.
Garanzia di successo (o post condizioni)	Il database è stato aggiornato correttamente, rimanendo in uno stato consistente.

Scenario principale di successo	<ol style="list-style-type: none"> 1. Il birraio avvia l'applicazione. 2. Il birraio inserisce i dati dell'ingrediente e clicca sul bottone modifica ingrediente. 3. Il sistema si connette al database. 4. Il sistema modifica correttamente l'ingrediente.
Estensioni	<ol style="list-style-type: none"> 1. Il birraio entra nell'applicazione. 2. Il birraio vuole modificare un ingrediente. 3. Il birraio non compila il campo nome dell'ingrediente. 4. L'applicazione mostra un messaggio di errore.
Requisiti speciali	L'applicazione permette di aggiornare un ingrediente alla volta.
Elenco delle varianti tecnologiche e dei dati	Nessuna
Frequenza di ripetizione	Si esegue questo caso d'uso ogni qual volta si vuole modificare le caratteristiche di un ingrediente precedentemente salvato nel database.
Varie	

Aggiungi ricetta

Portata	Sistema di gestione dell'applicazione
Livello	Obiettivo utente
Attore principale	Birraio
Parti interessate e interessi	<ul style="list-style-type: none"> • Birraio: vuole aggiungere una nuova ricetta per produrre una particolare birra. • Amministratore: deve aggiornare il database, eseguendo un inserimento nella tabella ricetta, ricettaingrediente e ricettaattrezzatura.
Precondizioni	<ul style="list-style-type: none"> • Nel database devono essere presenti le attrezzature di cui la ricetta ha bisogno
Garanzia di successo	<p>È stata aggiunta nel database una nuova ricetta associata a degli ingredienti e a delle attrezzature con una eventuale nota.</p> <p>Il sistema rimane in uno stato consistente.</p>

Scenario principale di successo	<ol style="list-style-type: none"> 1. Il birraio avvia l'applicazione; 2. Il birraio vuole aggiungere una ricetta; 3. Il birraio compila i campi necessari per creare una ricetta, compresi titolo e descrizione della nota; 4. Il sistema esegue una connessione con il database; 5. L'amministratore aggiunge la ricetta nel database eseguendo una operazione di inserimento, operando sulle tabelle ricetta, ricettaingrediente, ricettaattrezzatura.
Estensioni	<ol style="list-style-type: none"> 1. Il birraio avvia l'applicazione; 2. Il birraio vuole aggiungere una ricetta; 3. Il birraio compila tutti i campi necessari per creare una nuova ricetta, ma non specifica né il nome né il titolo della nota; 4. Il sistema esegue una connessione al database; 5. L'applicazione aggiunge al database la nuova ricetta, senza associarne alcuna nota.
Requisiti speciali	L'applicazione può aggiungere una ricetta alla volta
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizione	Ogni qualvolta il birraio vuole aggiungere una nuova ricetta al database.
Varie	

Modifica ricetta

Portata	Sistema di gestione dell'applicazione
Livello	Obiettivo utente
Attore principale	Birraio
Parti interessate e interessi	<ul style="list-style-type: none"> • Birraio: vuole modificare una ricetta precedentemente salvata nel database; • Amministratore: deve aggiornare il contenuto del database, modificando la ricetta, come richiesto dal birraio.
Precondizioni	Nel database deve essere presente la ricetta da modificare.
Garanzia di successo	È stata modificata la ricetta come richiesto dal birraio, lasciando il sistema in uno stato consistente.

Scenario principale di successo	<ol style="list-style-type: none"> 1. Il birraio avvia l'applicazione; 2. Il birraio decide di modificare una ricetta precedentemente salvata nel database; 3. Il birraio scrive nel campo apposito il nome della ricetta da modificare; 4. Il birraio inserisce i nuovi dati della ricetta; 5. Il birraio clicca il pulsante dedicato alla modifica della ricetta; 6. Il sistema esegue una connessione con il database; 7. L'applicazione memorizza nel database la ricetta modificata; 8. Viene mostrato un messaggio di conferma del fatto che la ricetta è stata modificata correttamente.
Estensioni	<ul style="list-style-type: none"> • Il birraio tenta di modificare una ricetta, inserendo il nome di una ricetta, che non è presente nel database; • Viene mostrato un messaggio di errore.
Requisiti speciali	L'applicazione può modificare una ricetta alla volta
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizione	Ogni qualvolta il birraio vuole modificare una nuova ricetta al database.
Varie	

Modifica ingrediente

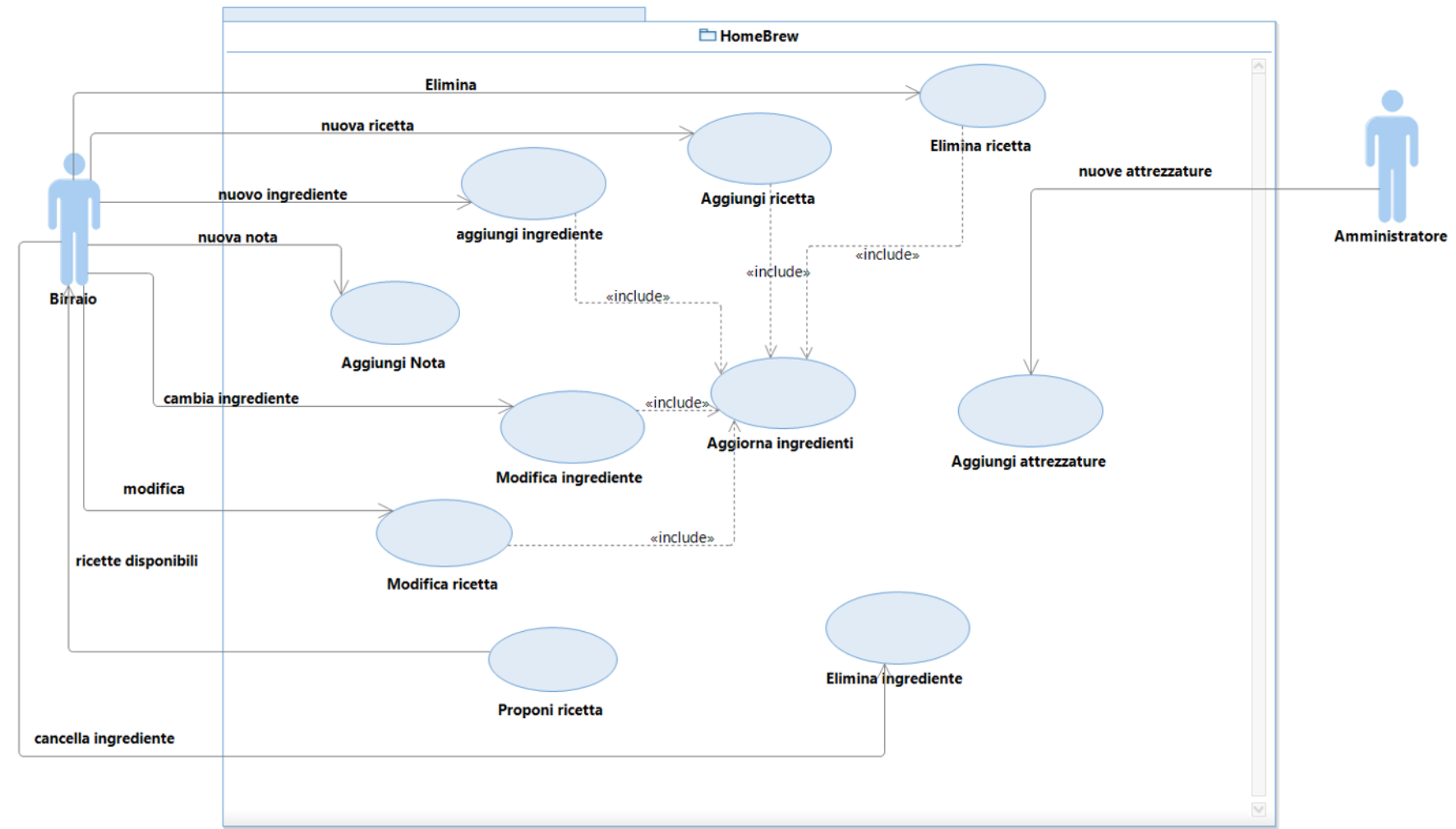
Portata	Sistema di gestione dell'applicazione
Livello	Obiettivo utente
Attore principale	Birraio
Parti interessate e interessi	<ul style="list-style-type: none"> • Birraio: vuole modificare un ingrediente esistente; • Amministratore: deve modificare i campi della tabella ingrediente in base all'input ricevuto dal birraio.
Precondizioni	Nel database deve essere memorizzato l'ingrediente da modificare.
Garanzia di successo	L'ingrediente è stato modificato correttamente, lasciando il sistema in uno stato consistente.

Scenario principale di successo	<ol style="list-style-type: none"> 1. Il birraio avvia l'applicazione; 2. Il birraio decide di modificare un ingrediente precedentemente salvato nel database; 3. Il birraio scrive i dettagli dell'ingrediente da modificare compilando tutti i campi necessari; 4. Il birraio clicca il pulsante per modificare l'ingrediente; 5. Il sistema esegue una connessione con il database; 6. Il sistema verifica che l'ingrediente effettivamente è contenuto nel database; 7. L'applicazione modifica l'ingrediente memorizzato nel database; 8. Viene mostrato un messaggio che conferma la corretta modifica dell'ingrediente.
Estensioni	<ol style="list-style-type: none"> 1. Il birraio avvia l'applicazione. 2. Il birraio decide di modificare un ingrediente precedentemente salvato nel database; 3. Il birraio modifica l'ingrediente non specificandone il nome; 4. L'applicazione mostra un messaggio di errore al birraio; 5. Nessun ingrediente viene modificato.
Requisiti speciali	L'applicazione può modificare un ingrediente alla volta.
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizione	Ogni qualvolta il birraio vuole modificare un ingrediente.
Varie	

Gruppo: Birra 1

Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

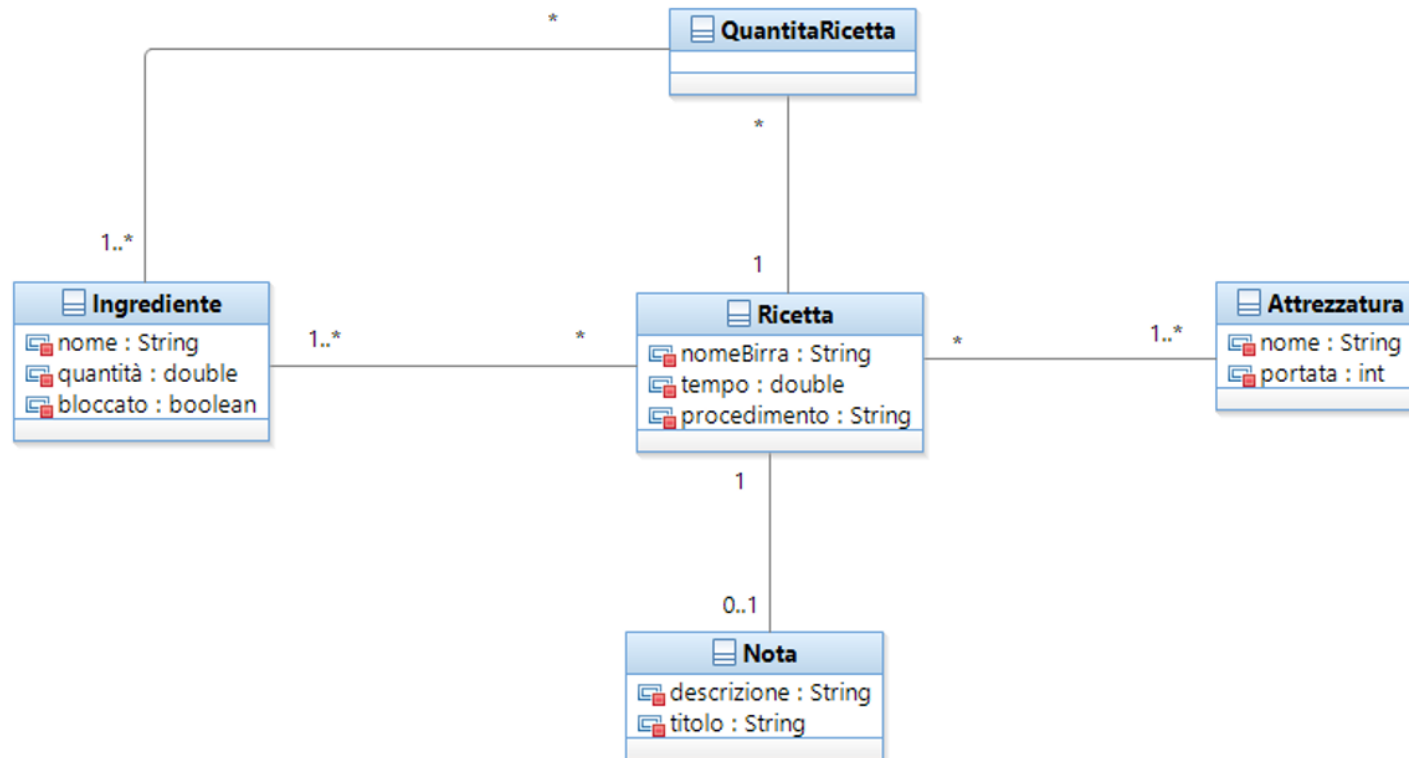
Diagramma dei casi d'uso



Gruppo: Birra 1

Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

Modello di dominio



Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

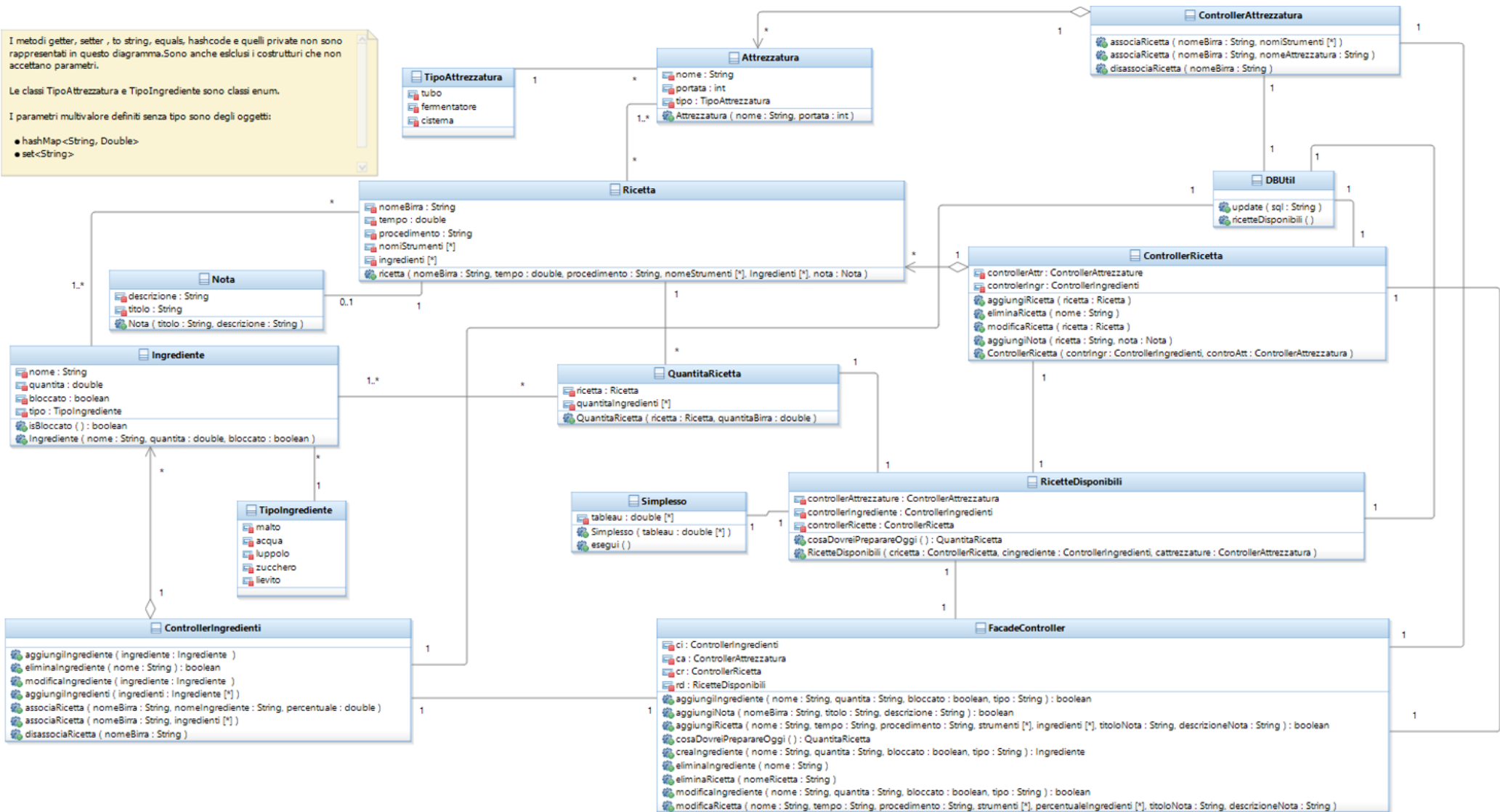
Diagramma delle classi di progetto

I metodi `getter`, `setter`, `toString`, `equals`, `hashCode` e quelli `private` non sono rappresentati in questo diagramma. Sono anche esclusi i costruttori che non accettano parametri.

Le classi `TipoAttrezzatura` e `TipoIngrediente` sono classi enum.

I parametri multivalore definiti senza tipo sono degli oggetti:

- `hashMap<String, Double>`
- `set<String>`

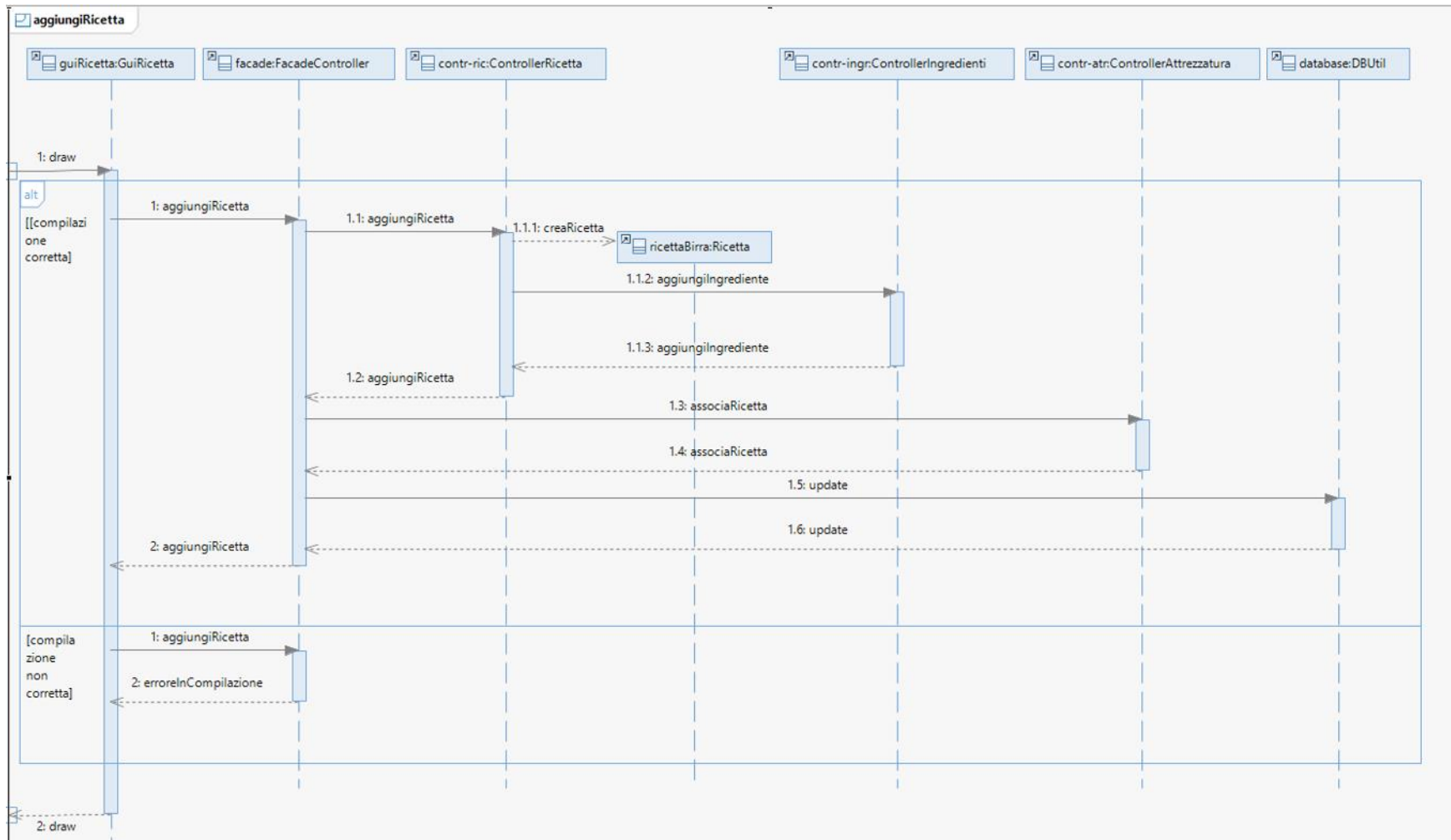


Gruppo: Birra 1

Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

Diagrammi di sequenza

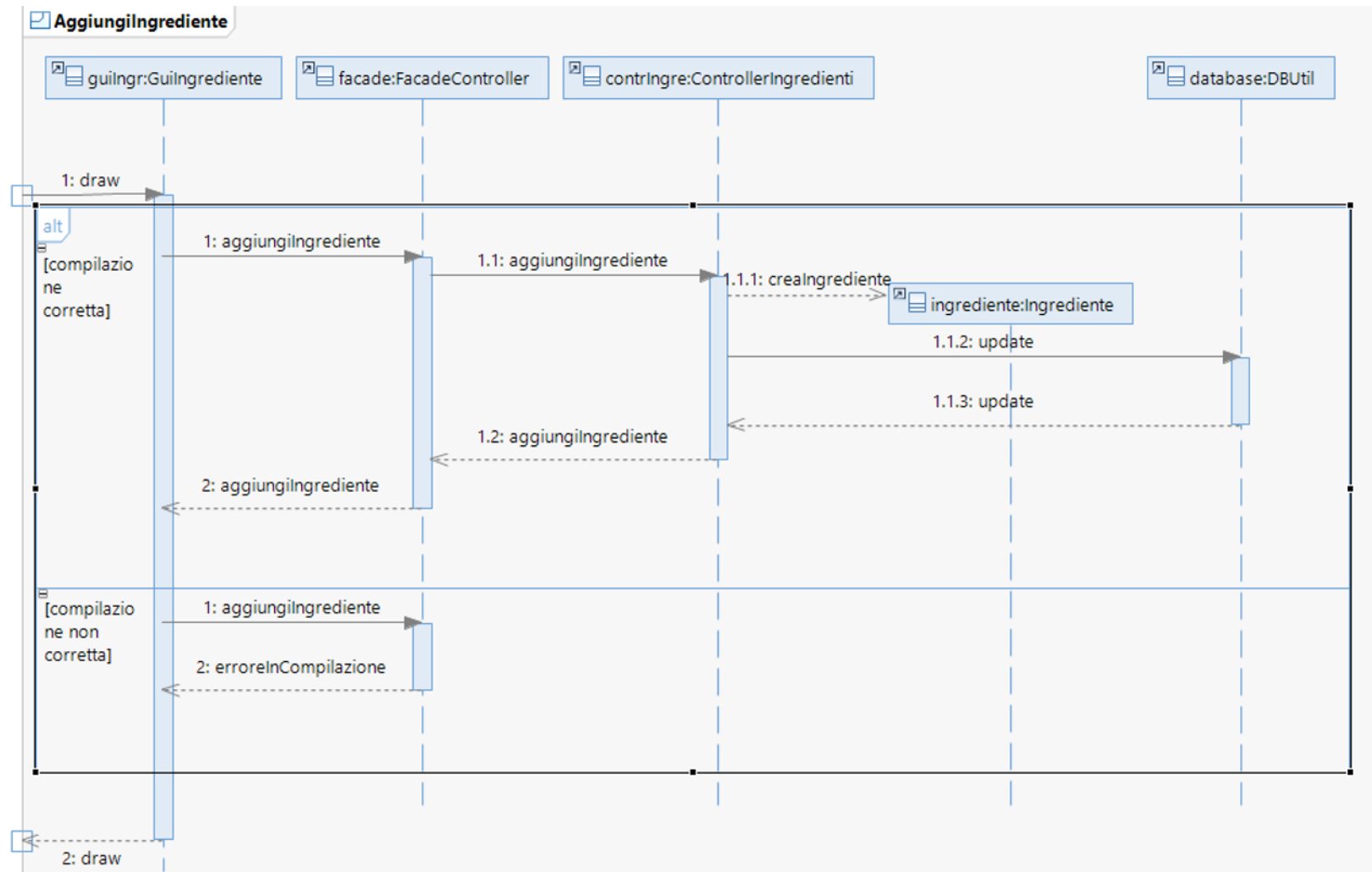
Aggiungi ricetta



Gruppo: Birra 1

Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

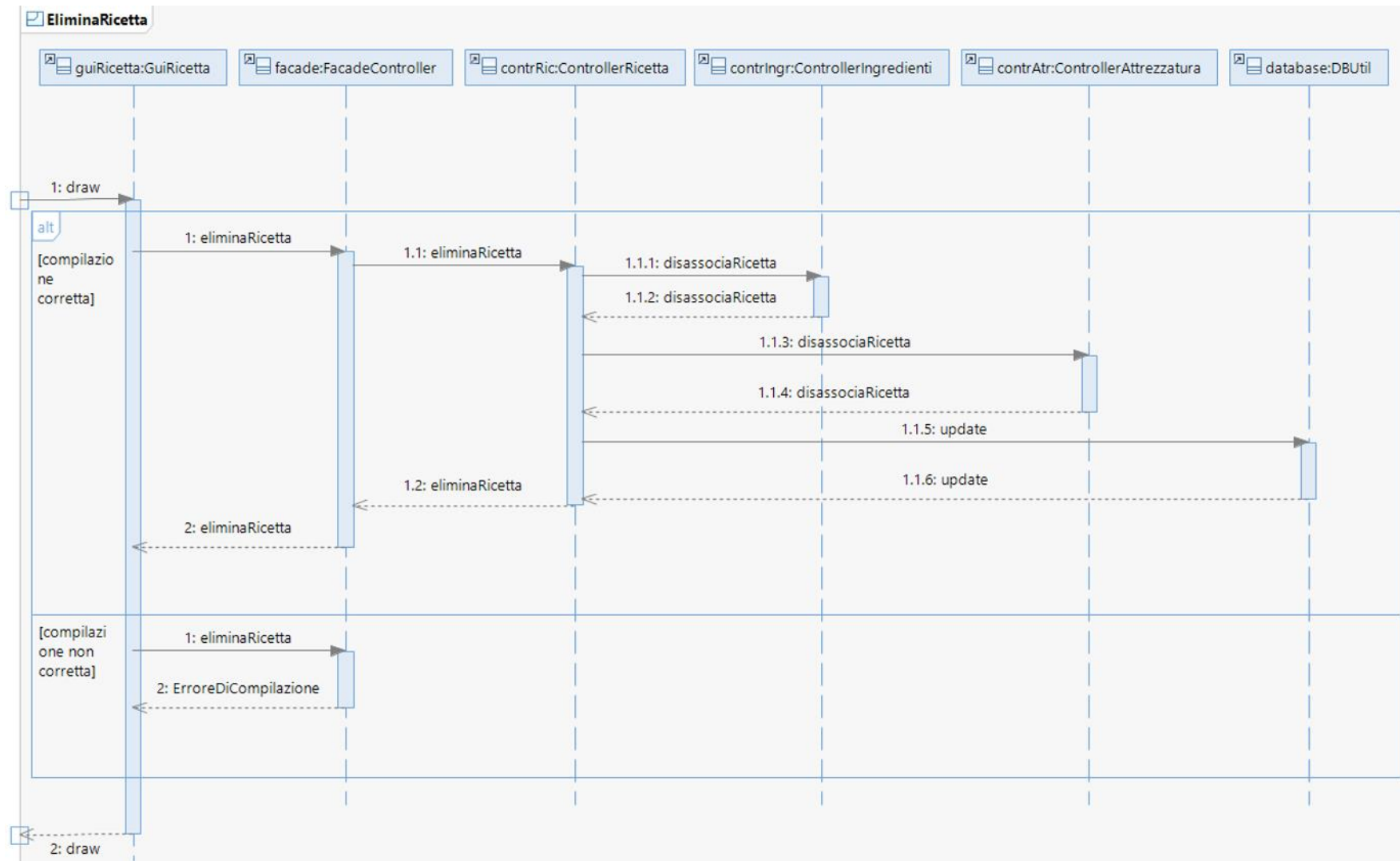
Aggiungi Ingrediente



Gruppo: Birra 1

Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

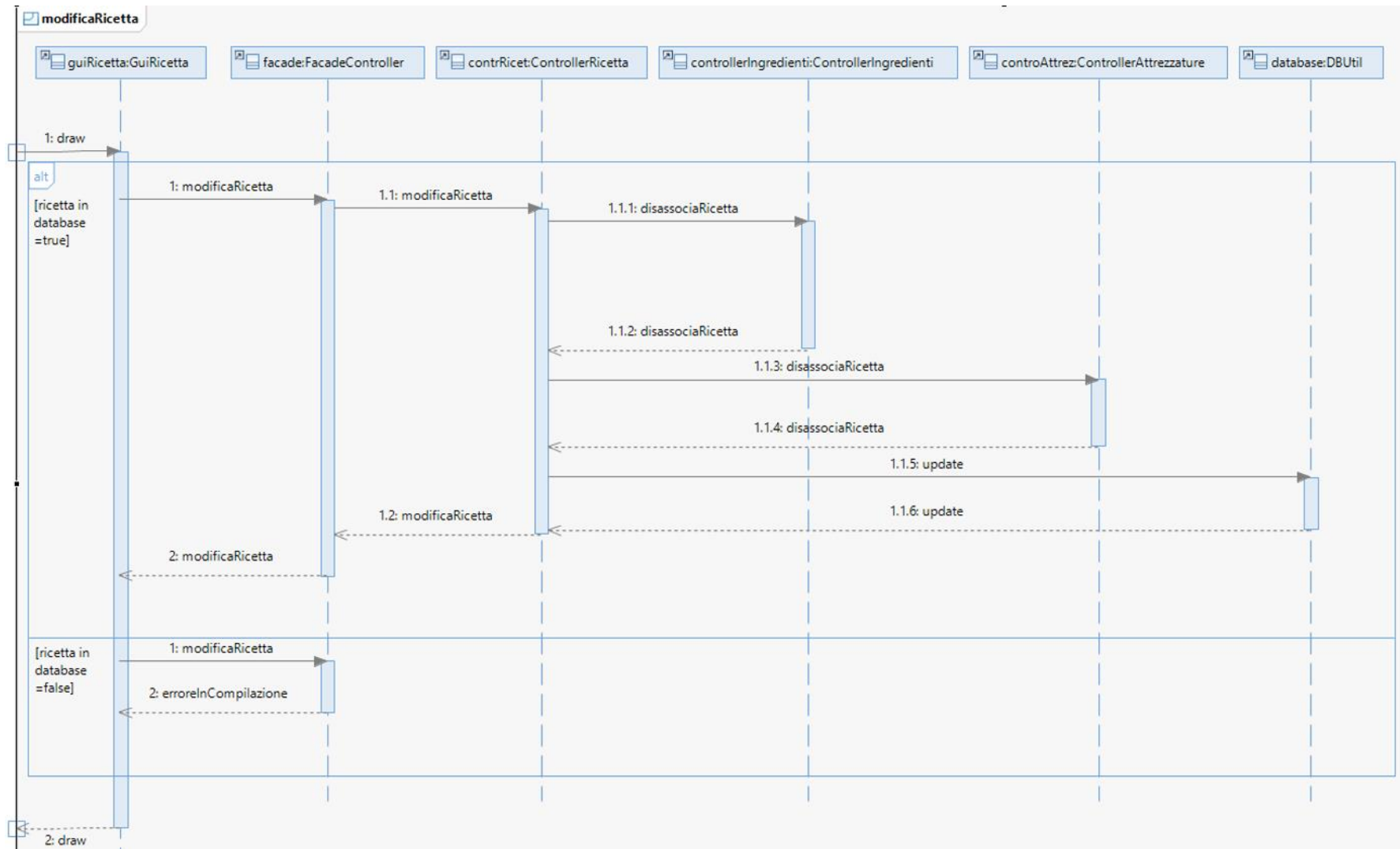
Elimina ricetta



Gruppo: Birra 1

Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

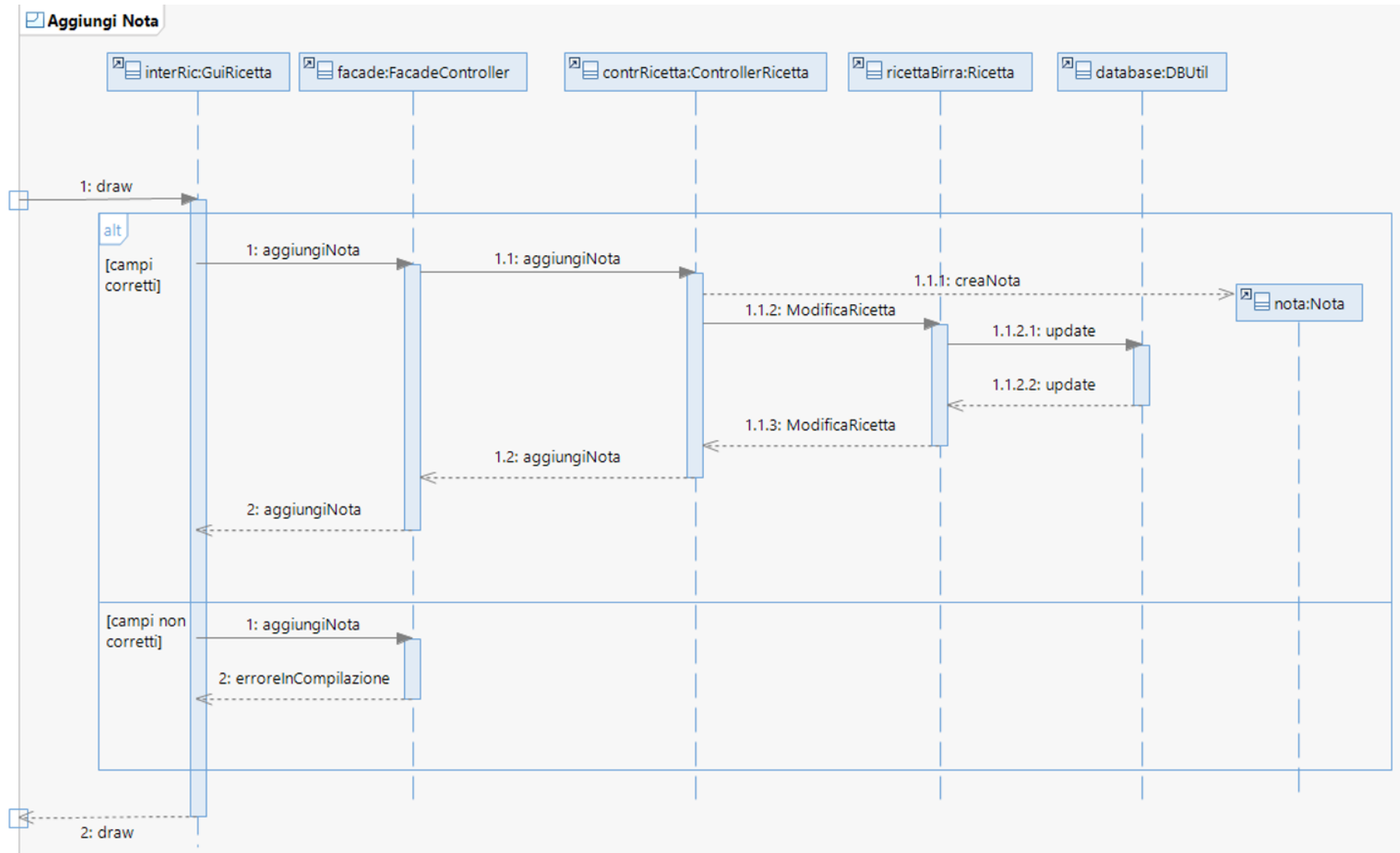
Modifica ricetta



Gruppo: Birra 1

Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

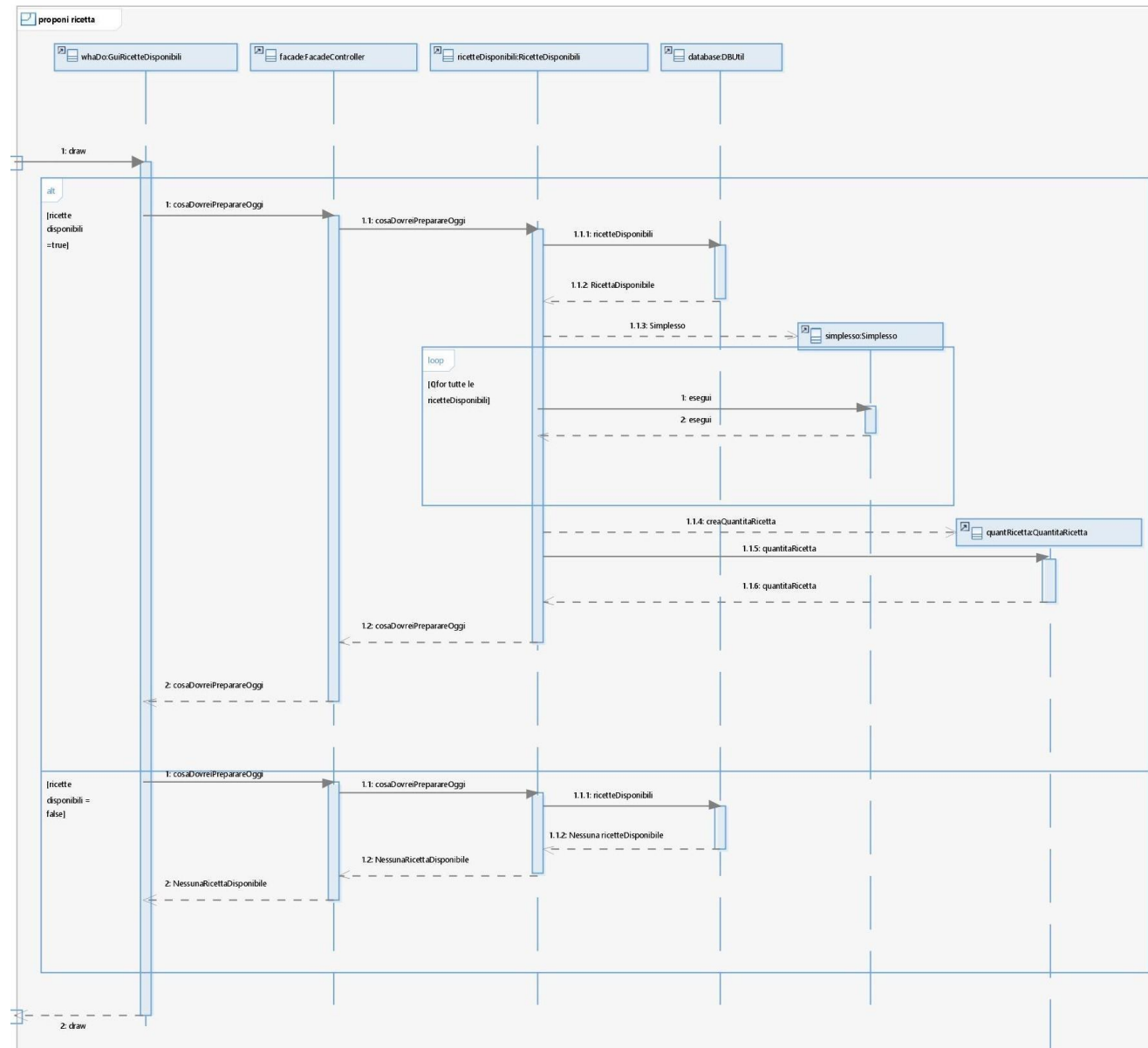
Aggiungi nota



Gruppo: Birra 1

Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

Proponi ricetta



Contratti

Operazione:	cosaDovreiPreparareOggi()
Riferimenti	Caso d'uso: Proponi ricetta
Pre-condizioni	<ul style="list-style-type: none"> • Sono presenti nel database degli ingredienti. • Sono presenti nel database delle attrezzature.
Post-condizioni	<ul style="list-style-type: none"> • È stata proposta una ricetta al birraio, che massimizza la quantità di birra che permette di produrre.

Operazione:	aggiungiIngrediente (ingrediente Ingrediente)
Riferimenti	Caso d'uso: aggiungi Ingrediente
Pre-condizioni	
Post-condizioni	<ul style="list-style-type: none"> • È stato aggiunto un ingrediente al database.

Operazione:	modificaIngrediente (ingrediente Ingrediente)
Riferimenti	Caso d'uso: modifica Ingrediente
Pre-condizioni	<ul style="list-style-type: none"> • È presente nel database l'ingrediente richiesto.
Post-condizioni	<ul style="list-style-type: none"> • Sono state modificate le colonne delle tabelle che contengono riferimenti all'ingrediente in questione.

Operazione:	aggiungiRicetta (Ricetta nome)
Riferimenti	Caso d'uso: aggiungi Ricetta
Pre-condizioni	<ul style="list-style-type: none"> • Sono presenti nel database le attrezzature a cui fa riferimento la ricetta.
Post-condizioni	<ul style="list-style-type: none"> • È stata memorizzata una nuova ricetta nel database. • Sono stati associati ingredienti alla nuova ricetta. • Sono state associate attrezzature alla nuova ricetta.

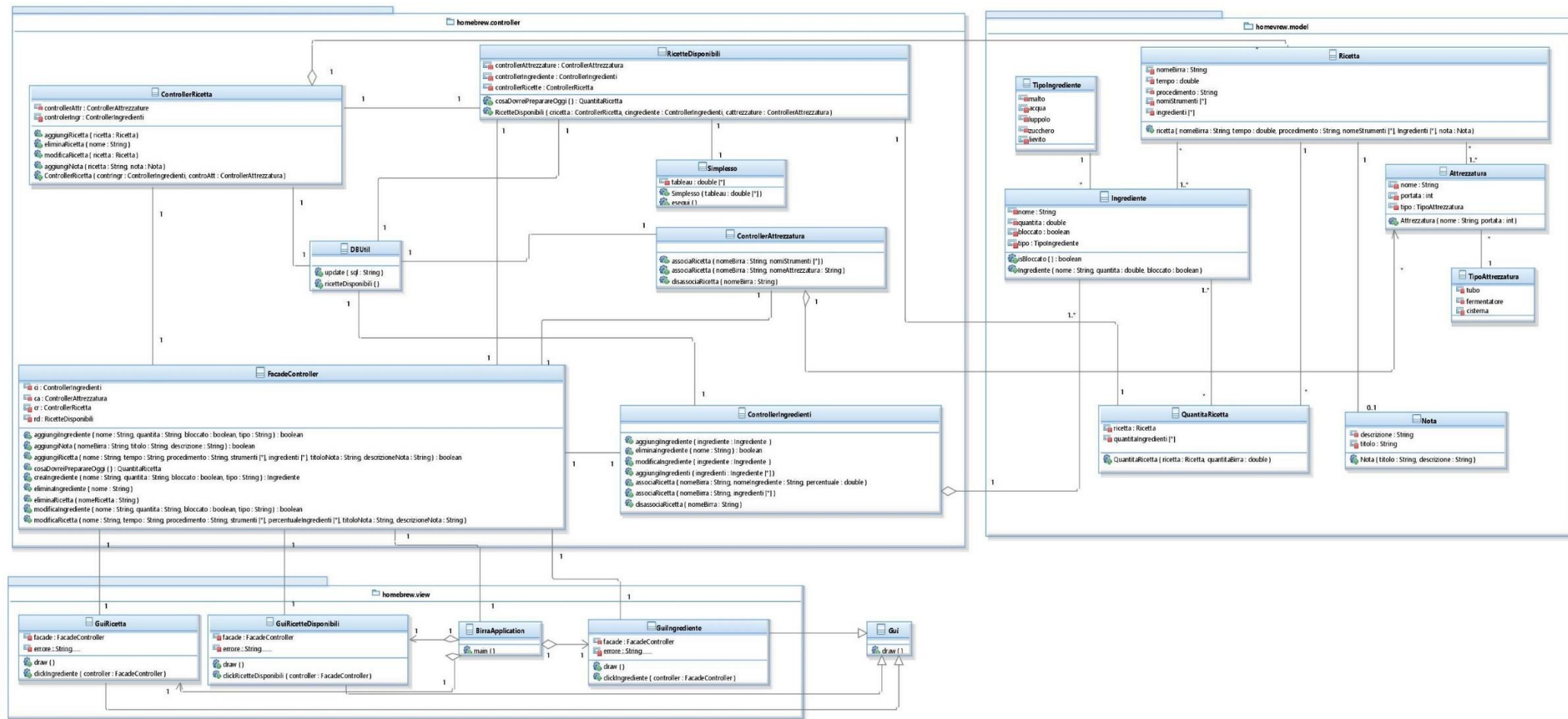
Operazione:	eliminaRicetta (Ricetta nome)
Riferimenti	Caso d'uso: eliminaRicetta
Pre-condizioni	<ul style="list-style-type: none"> • La ricetta è salvata nel database • Gli ingredienti ad essa associata sono salvati nel database • Le attrezzature ad essa associate sono salvate nel database
Post-condizioni	<ul style="list-style-type: none"> • È stata eliminata la ricetta indicata. • Gli ingredienti sono stati dissociati da tale ricetta. • Le attrezzature sono state dissociate da tale ricetta.

Operazione:	modificaRicetta (Ricetta nome)
Riferimenti	Caso d'uso: modifica Ricetta
Pre-condizioni	<ul style="list-style-type: none"> • È presente nel database la ricetta richiesta • Sono presenti nel database gli ingredienti a cui fa riferimento la ricetta. • Sono presenti nel database le attrezzature a cui fa riferimento la ricetta.
Post-condizioni	<ul style="list-style-type: none"> • È stata memorizzata la ricetta modificata nel database. • Sono stati associati ingredienti alla ricetta modificata. • Sono state associate attrezzature alla ricetta modificata.

Gruppo: Birra 1

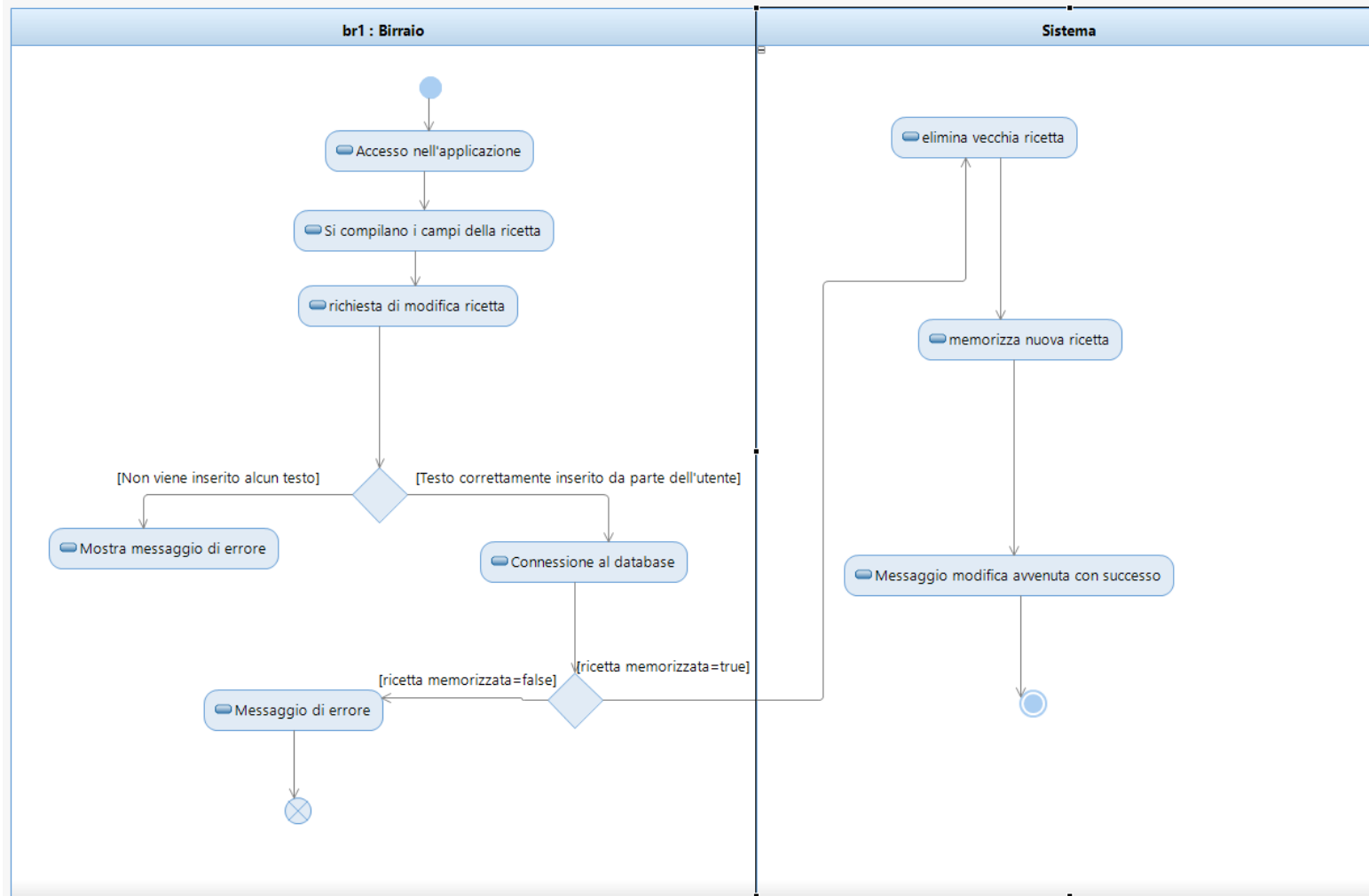
Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

Diagramma dell'architettura software



Diagrammi di attività

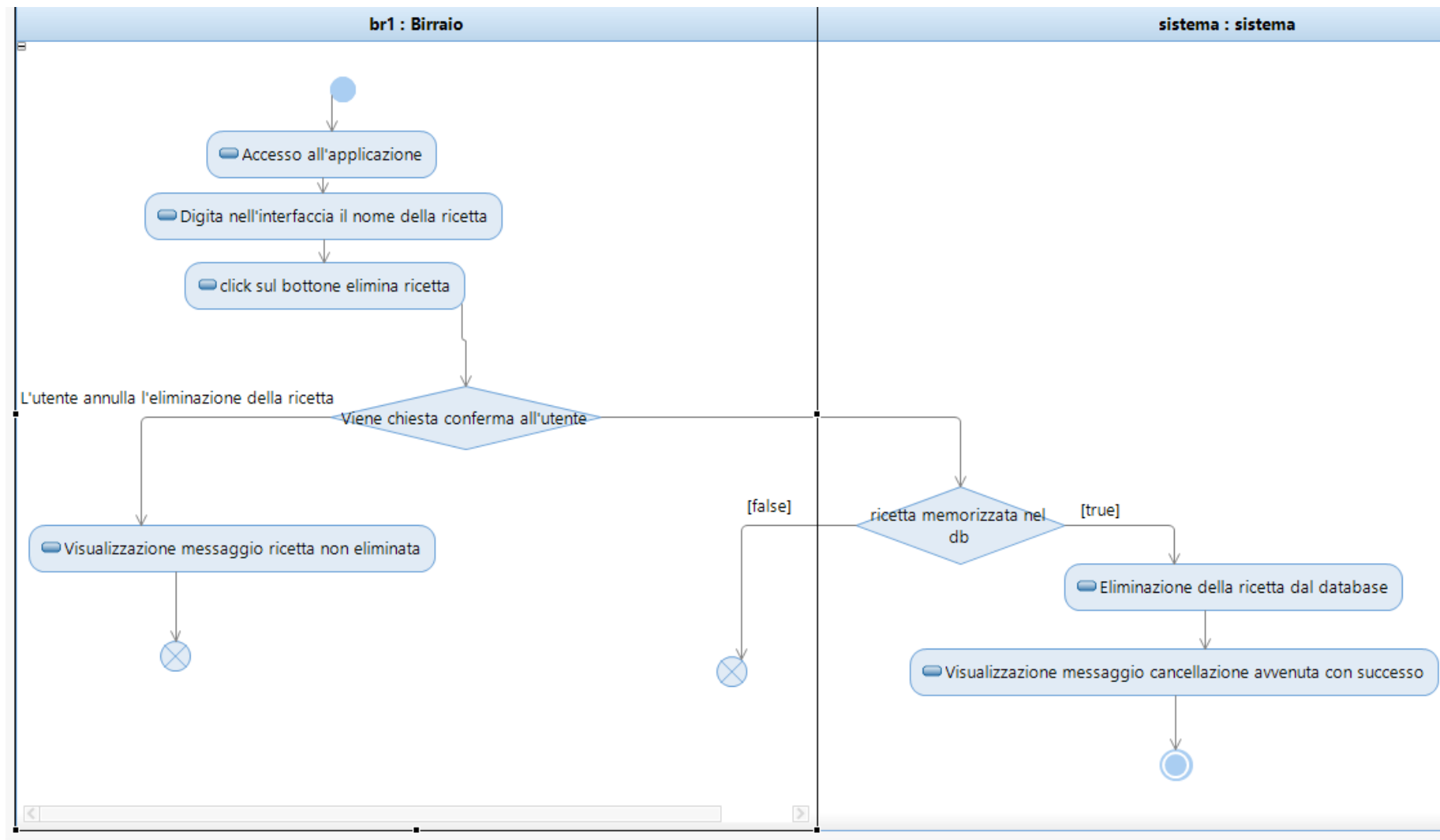
Modifica ricetta



Gruppo: Birra 1

Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

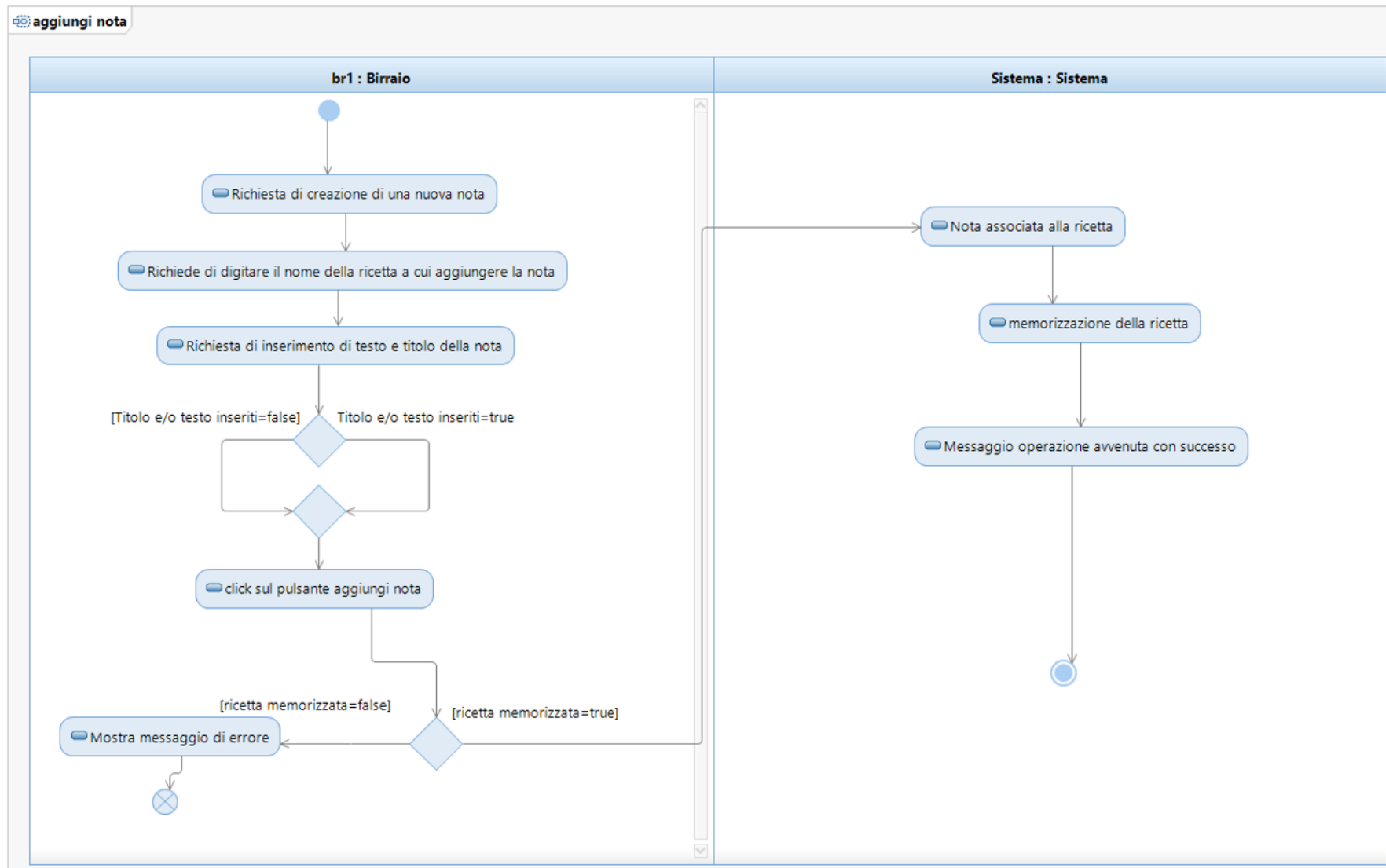
Elimina ricetta



Gruppo: Birra 1

Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

Aggiungi Nota

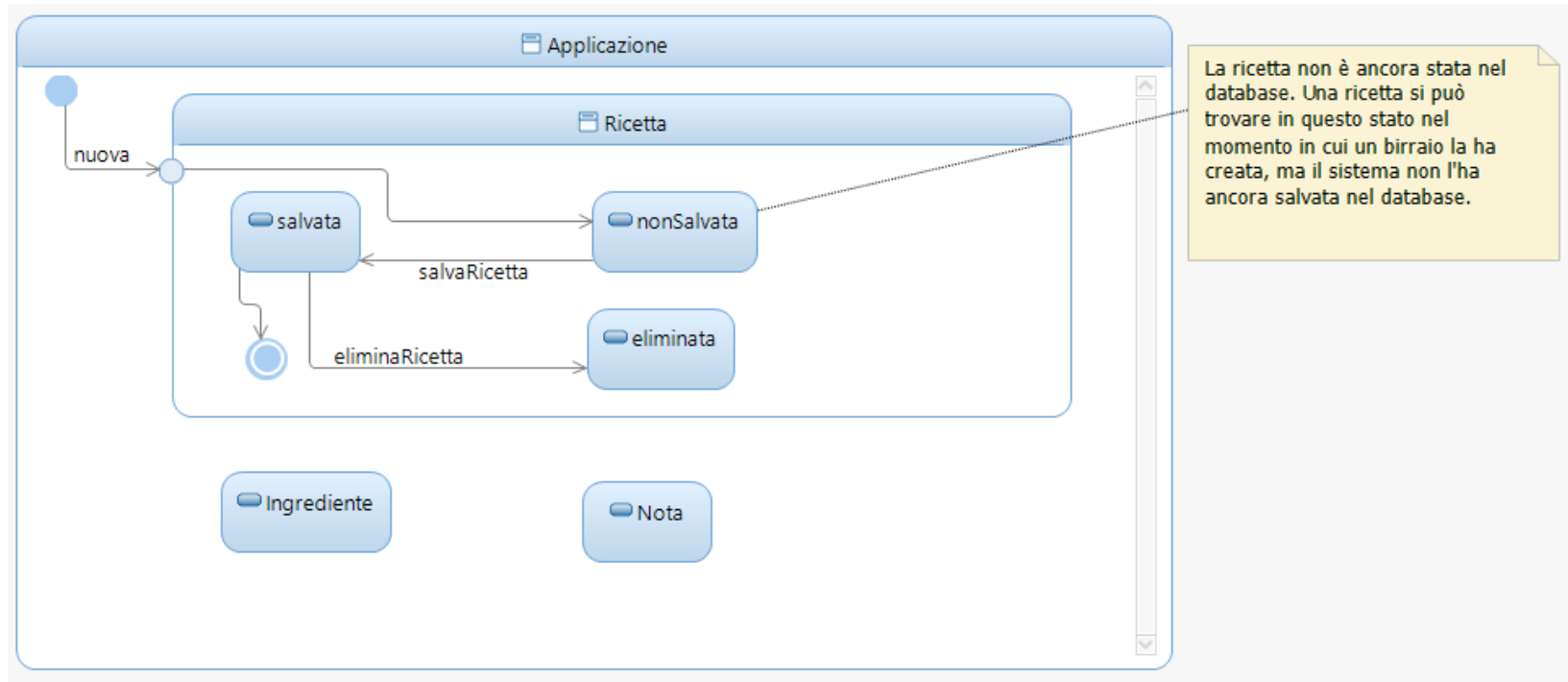


Gruppo: Birra 1

Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

Diagrammi di Stato

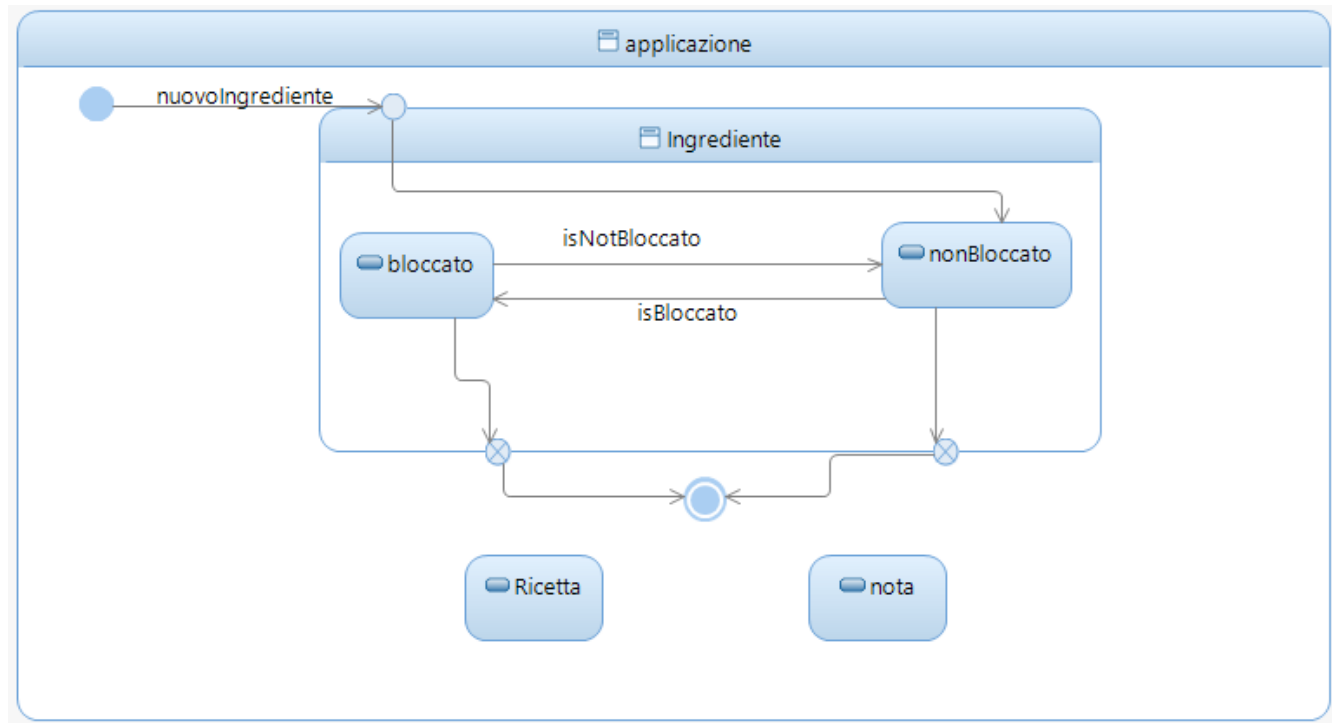
Aggiungi ricetta



Gruppo: Birra 1

Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

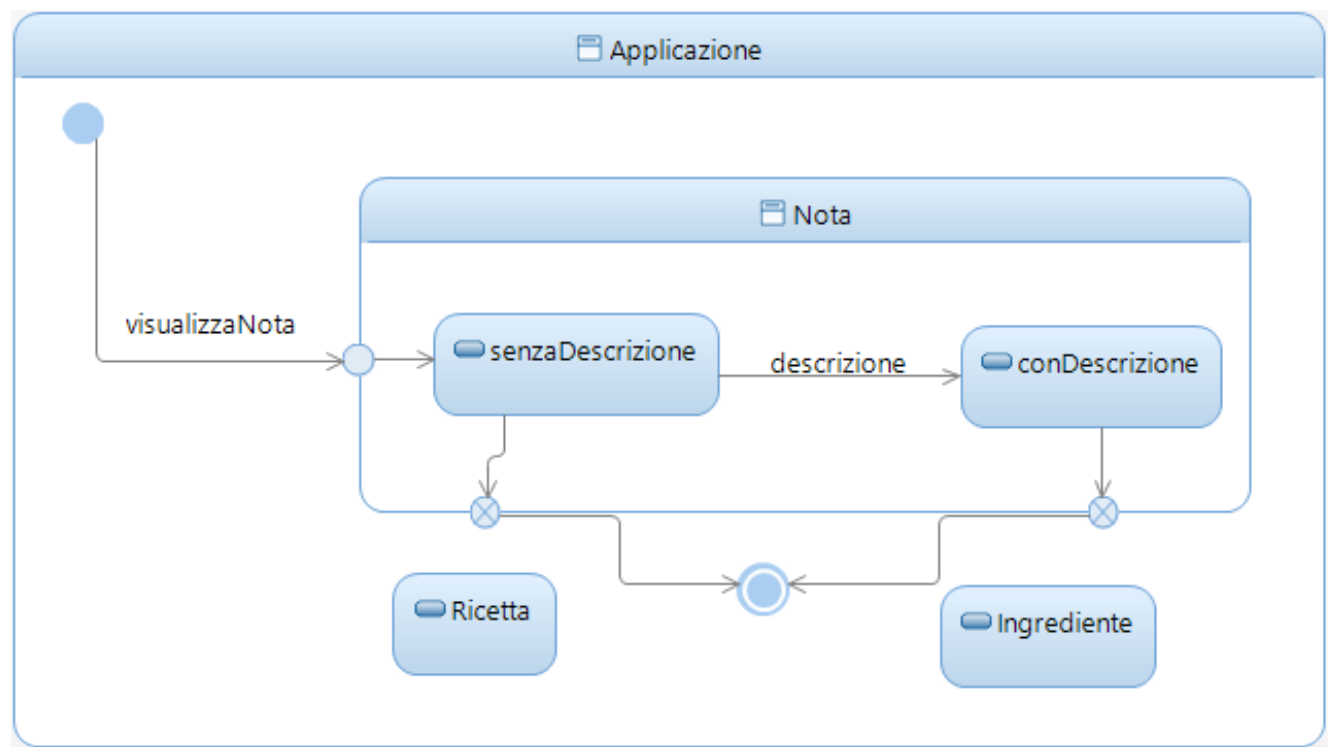
Modifica ingrediente



Gruppo: Birra 1

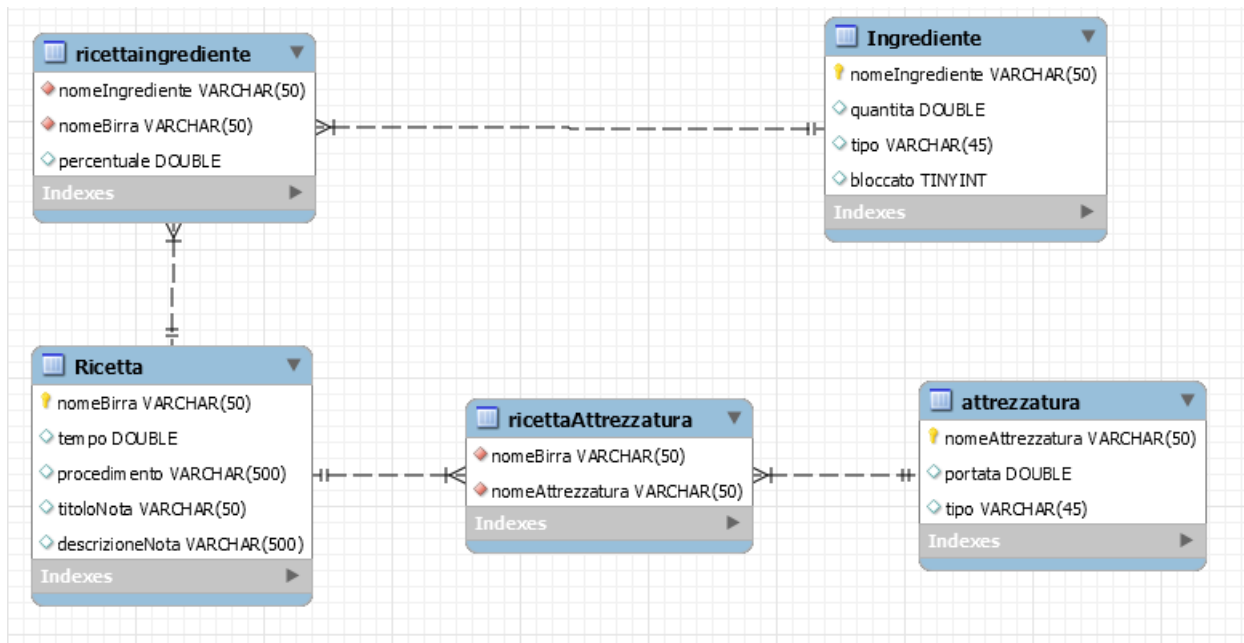
Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

[Visualizza nota](#)



Database

Per gestire la memorizzazione degli ingredienti, delle ricette, delle note e delle attrezzature abbiamo creato appositamente un database basato su MySQL. Nella figura sottostante è rappresentato il modello relazionale del database che abbiamo deciso di creare.



Come si può notare il database è composto da cinque tabelle. La prima, la tabella Ingrediente permette di tenere memoria di tutti gli ingredienti disponibili o meno (bloccato significa che l'ingrediente non è disponibile). La tabella Ricetta invece permette di tenere memoria di tutte le ricette create dal birraio. Si noti che la tabella ricetta contiene un campo chiamato titoloNota e uno chiamato descrizioneNota. Abbiamo deciso di fare questa scelta progettuale, invece di creare una tabella a parte per memorizzare le note di una ricetta, per il semplice fatto che ogni ricetta può avere associata al più una nota. La tabella ricettaIngrediente invece, permette di associare ad ogni ricetta gli ingredienti di cui il birraio necessita per poterla eseguire. È fondamentale anche per poter tener traccia del valore percentuale, che sta ad indicare la proporzione in termini assoluti di ogni ingrediente rispetto agli altri ingredienti della stessa ricetta. La tabella attrezzatura contiene tutte le attrezzature che ha a disposizione il birraio. Come già specificato in fase di esplicitazione delle scelte implementative, tutte le attrezzature del birraio sono immutabili, ovvero non subiranno cambiamenti nel corso del tempo. Infine, la tabella ricettaattrezzatura fornisce un collegamento tra la tabella ricetta e la tabella attrezzatura e permette di associare ad ogni ricetta l'attrezzatura di cui il birraio necessita per poterla eseguire.

Design Pattern usati

Facade: necessario per evitare che la GUI comunicasse con più classi controller. Abbiamo sussunto le funzionalità dei diversi controller in un'unica classe facade (che si chiama FacadeController), che ne va a richiamare i metodi. Inoltre, tale classe implementa i controlli sull'input inserito dall'utente, lanciando eventualmente delle opportune eccezioni.

Pattern Architetture usati

Dependent mapping: necessario per mappare le classi composte nel database. In particolare, ogni volta che viene eseguita un'operazione sul database relativa ad un oggetto di tipo ricetta, è necessario eseguire delle operazioni sul database anche per gli oggetti che vanno a comporlo, ovvero i suoi ingredienti e i suoi strumenti.

Model-View-Controller: necessario per separare le responsabilità, renderle distinte e coese con una separazione netta degli interessi.

Data mapper: necessario per poter mappare i dati, a cui sono stati associate più classi, per poi memorizzarli nel database così da mantenere queste due strutture indipendenti fra di loro.

Design Principles usati

Open closed Principle (OCP): abbiamo cercato di rendere l'applicazione facilmente mantenibile. Questo significa che è possibile aggiungere feature aggiuntive, lasciando completamente inalterato il resto del codice. Per implementare la funzionalità "What should i brew today?" questo principio ci è stato particolarmente utile: inizialmente abbiamo sviluppato tutte le funzionalità secondarie, come la possibilità di aggiungere una ricetta, modificarla e così via, solo successivamente abbiamo potuto implementare la feature più importante (e complessa) dell'intera applicazione senza dover modificare il codice che si occupa delle altre funzionalità.

Acyclic Dependencies Principle (ADP): Analizzando il diagramma di architettura software si può notare che le dipendenze tra i vari package non formano cicli. Questo è fondamentale per garantire una maggiore riusabilità del codice.

Single Responsibility Principle: con tale principio abbiamo fatto in modo che ogni modulo, classe e metodo fosse responsabile di una singola parte della funzionalità che offre l'applicazione. Questo significa che ogni singola responsabilità è incapsulata nelle singole classi, metodi e package. In questo modo viene garantita una forte riusabilità e leggibilità del codice.

Understand

Procedendo con l'implementazione del codice abbiamo eseguito varie analisi con Understand. Questo software ci ha permesso di capire quali sono le classi maggiormente strutturate. Inoltre, ci ha fornito informazioni utili relative alla suddivisione delle linee di codice, come possiamo vedere nell'immagine sottostante.

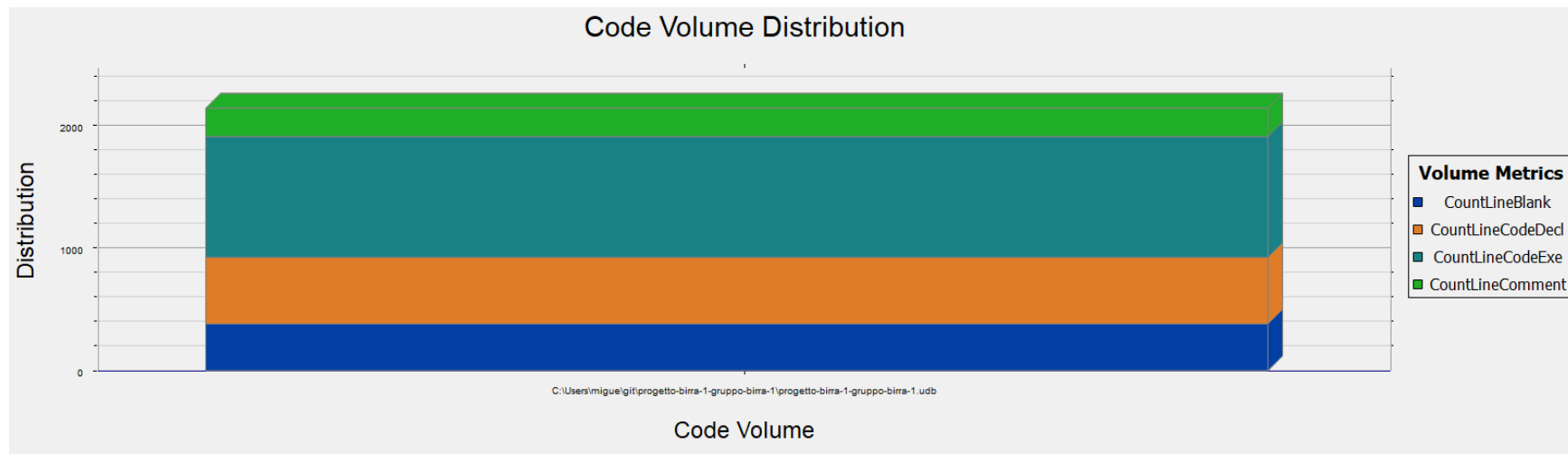
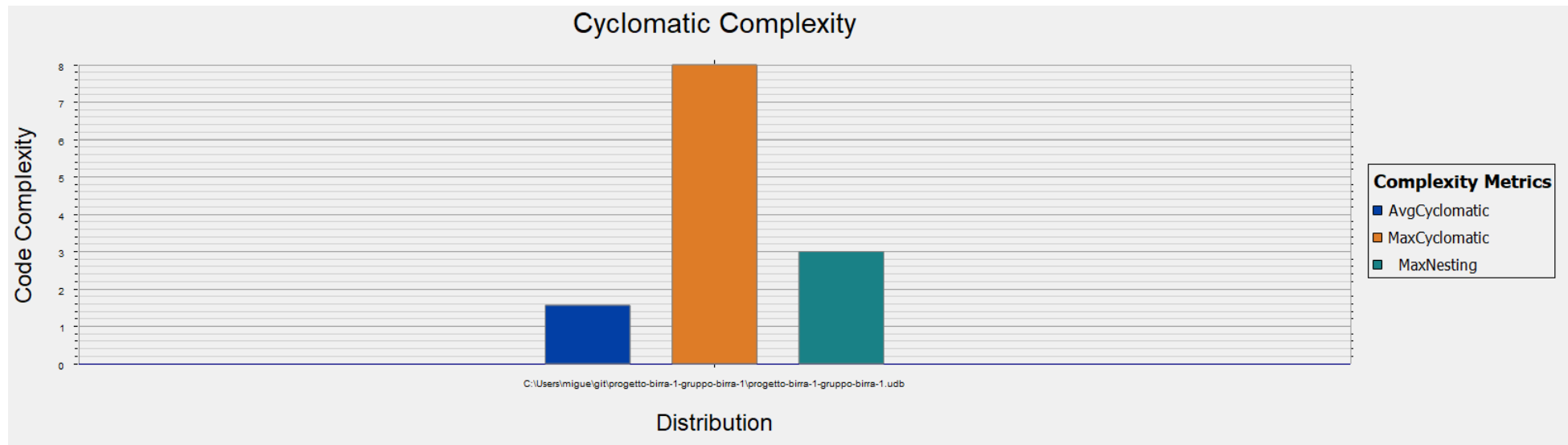


Figura 1 Distribuzione del codice

Notiamo che abbiamo un numero abbastanza elevato di righe bianche (utili per migliorare la leggibilità del codice). Viceversa, il numero di righe che rappresentano i commenti non è particolarmente elevato, anche se sufficiente per garantire una facile comprensione del codice per chi non ne conosce il funzionamento.



Con il grafico soprastante, sempre ottenuto mediante il software Understand, abbiamo considerato tre metriche differenti: AverageCyclomatic, MaxCyclomatic, che indicano rispettivamente la complessità ciclomatica media e massima, e maxNesting, che indica il massimo numero di scope innestati all'interno di un metodo.

Le prime due metriche fanno riferimento al numero di cammini linearmente indipendenti attraverso il codice sorgente. Questo significa che se il codice non contiene if o for allora la complessità ciclomatica sarà pari a 1. Nel nostro specifico caso, abbiamo ottenuto un livello di complessità ciclomatica media molto basso (addirittura inferiore a 2), ma nel caso peggiore la complessità ciclomatica ha raggiunto un valore pari a 8, ovvero un valore piuttosto elevato. Tale picco è stato raggiunto nella classe semplice, ovvero quella più complessa, che permette di risolvere un problema di ottimizzazione lineare. Di conseguenza possiamo affermare con certezza che, data la complessità delle operazioni svolte da tale classe, il risultato ottenuto è del tutto giustificabile.

Per quanto riguarda la metrica MaxNesting invece, abbiamo ottenuto un valore (3) perfettamente nella media. Questo significa che il nostro codice non presenta un numero troppo elevato di scope innestati tra loro. Questo garantisce una buona leggibilità e una ottima manutenibilità del codice.

Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

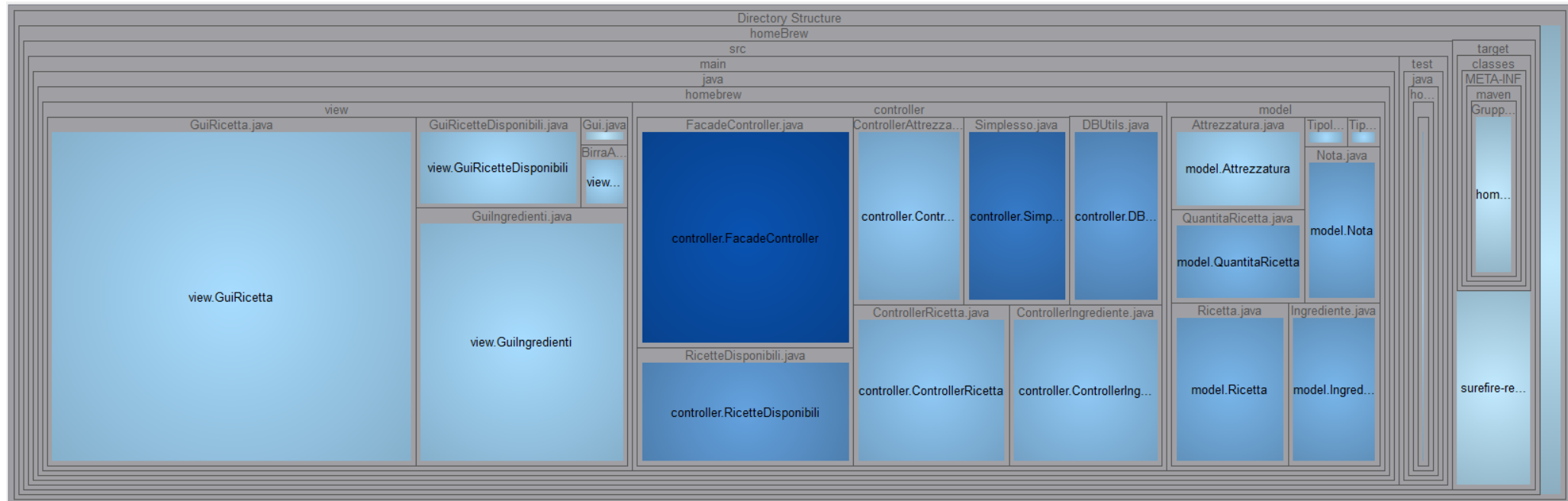


Figura 2 TreeMap dell'applicazione definita sulla base della complessità delle classi.

Un'ulteriore analisi che abbiamo effettuato per mezzo di Understand è stata possibile grazie ad un grafico che prende il nome di MetricsTreemap. Esso permette di mettere in relazione due metriche a scelta. Noi abbiamo scelto di mettere in relazione la metrica CountLine (che permette di contare il numero di righe di codice) e maxCyclomatic (che misura la complessità di ogni classe). Si può facilmente notare che le classi più corpose, ovvero quelle composte da più linee di codice e che eseguono più chiamate ad altri metodi sono le classi Controller. In particolare, è la classe FacadeController ad essere la più importante da questo punto di vista. Si tratta di una cosa del tutto prevedibile dato il fatto che FacadeController si occupa dei controlli dell'input dell'utente e che funge da collegamento tra tutte le classi controller e le classi che compongono la GUI.

Metric	Value
Project	progetto-birra-1-gruppo-birra-1.udb
Blank Lines	381
Classes	20
Code Lines	1.732
Comment Lines	231
Comment to Code Ratio	0,13
Declarative Statements	520
Executable Statements	715
Files	27
Functions	150
Inactive Lines	0
Lines	2.309
Subprograms	0

Figura 3 Sommario delle metriche del progetto homebrew

La figura soprastante invece, ci fornisce informazioni relativamente alla dimensione del progetto in termini di righe di codice, numero di righe di commento, numero di classi, funzioni usate e così via.

SonarQube

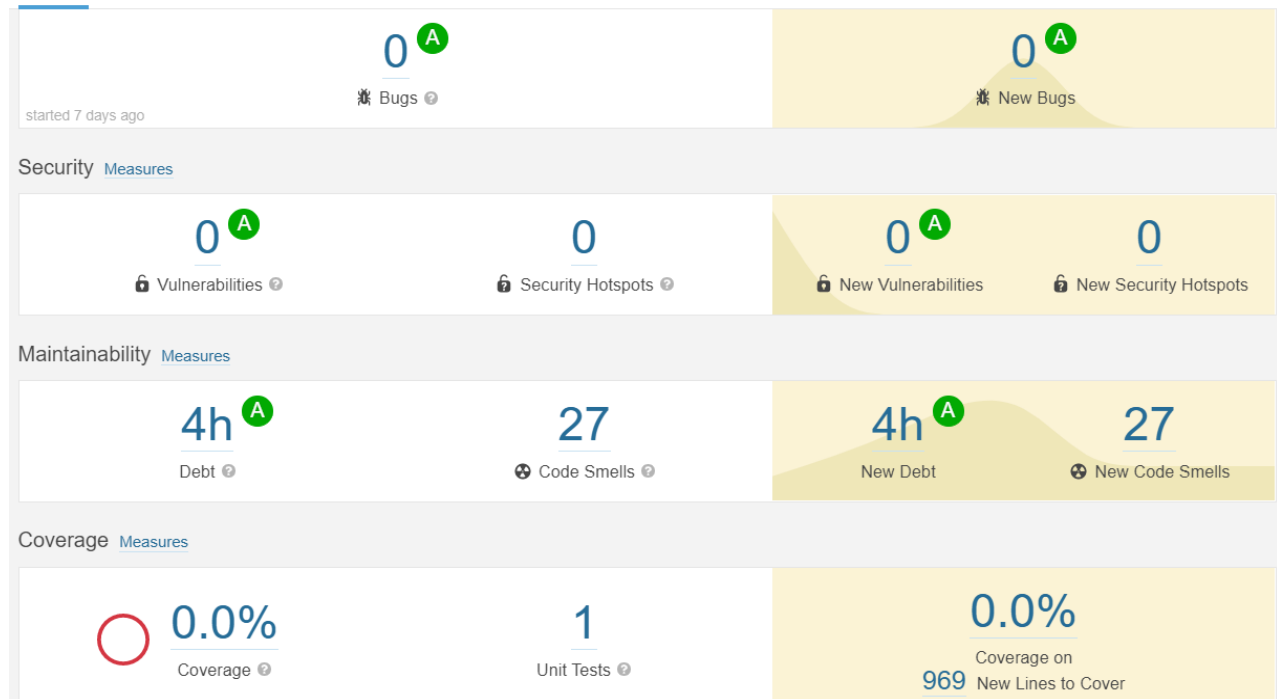
Per valutare la qualità del codice prodotto, abbiamo utilizzato costantemente, durante la fase di sviluppo del codice, il software SonarQube. Esso ci è stato particolarmente utile perché ci ha evitato di scrivere codice qualitativamente povero.

È possibile eseguire l'analisi sonarQube del nostro progetto lanciando il server sonarQube in locale e seguendo i seguenti passaggi: *aprire il progetto homeBrew con eclipse → click destro sul file pom.xml → run as → maven build... → inserire nel campo goals il comando sonar:sonar → connettersi alla pagina*

Gruppo: Birra 1

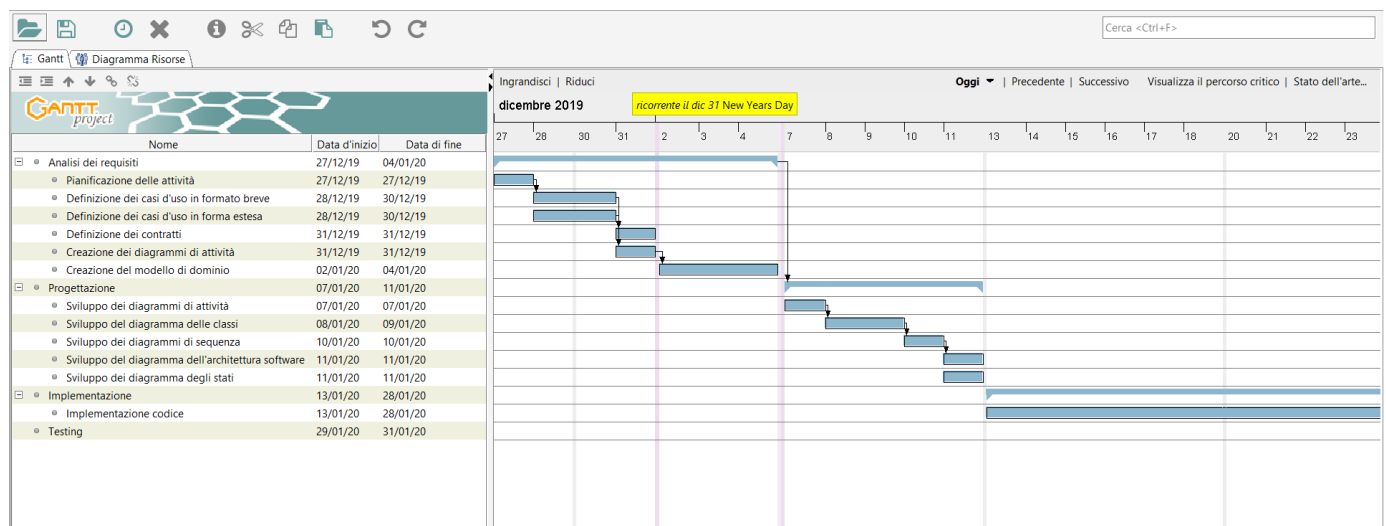
Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

localhost:9000 da un qualsiasi browser. Il risultato che si otterrà sarà quello riportato nella figura sottostante.



Come si può notare tutti gli indicatori sono ad A. Questo significa che non ci sono criticità di rilievo. Per ottenere tale risultato è stato necessario tenere costantemente controllati, procedendo con lo sviluppo del codice, gli indicatori forniti da SonarQube. Sono rimasti alcuni code smell che il software segnala, che però riteniamo essere per lo più falsi positivi. Tra l'altro, gran parte di questi vengono classificati come minor, ovvero meno rilevanti.

Tempistiche e risorse coinvolte

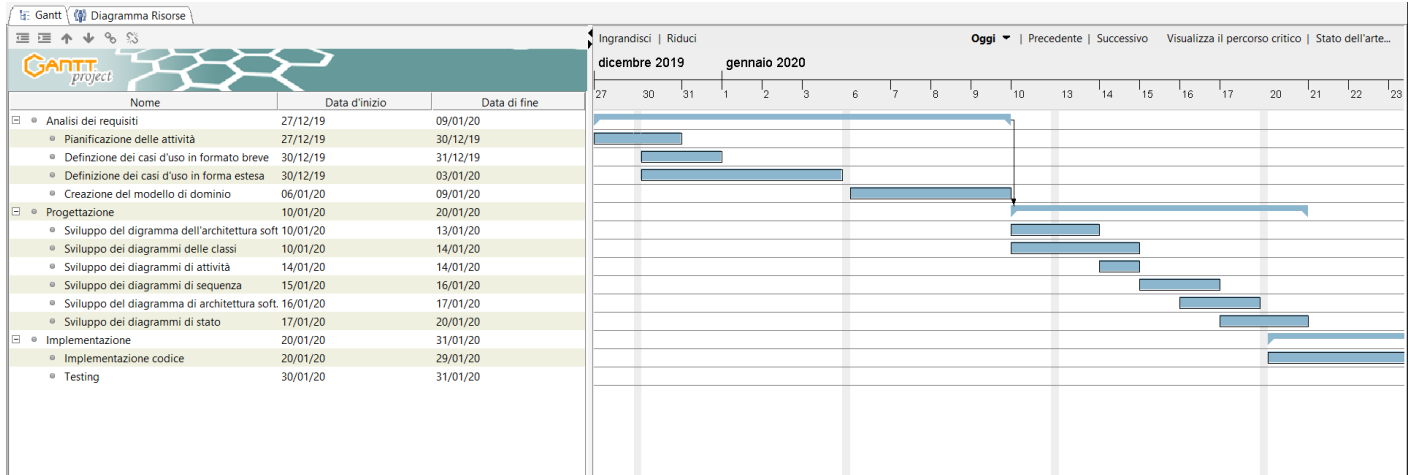


L'immagine soprastante mostra le tempistiche che, prima di iniziare il progetto, ci eravamo prefissati di seguire. Abbiamo deciso di dedicare tempo in abbondanza per ogni fase del progetto, in modo tale da sopperire ad eventuali problematiche, che avremmo incontrato durante l'esecuzione del progetto.

Gruppo: Birra 1

Membri: Romano Fabio, Stoppa Miguel, Tomasoni Lorenzo

L'immagine sottostante invece, rappresenta le tempistiche che effettivamente abbiamo seguito durante la progettazione e implementazione del progetto. Come si può notare, siamo stati leggermente in ritardo sulla tabella di marcia. Ritardo dovuto principalmente al fatto che all'incirca alla metà del mese di gennaio tutti i membri del gruppo hanno avuto altri esami da sostenere. Nonostante ciò, il progetto è stato completato in tempo, anzi sono avanzati un paio di giorni per il testing dell'applicazione.



Per quanto concerne le risorse coinvolte, ogni membro del gruppo ha lavorato attivamente ad ogni parte del progetto, sia per la parte di progettazione, sia per la parte di implementazione.