

Brew Day documentation

Martino Andrian, Gioele De Piano, Anselmo Folcio

January 2021

Contents

1	Introduzione	4
2	Analisi dei requisiti	4
3	Diagramma Gantt e organizzazione del lavoro	6
3.1	Descrizione delle attività	6
3.2	Calendario delle attività	6
4	Diagramma dei casi D'uso	7
5	Diagrammi di dominio	8
5.1	Diagramma delle classi di dominio	8
5.2	Diagramma EER	9
6	Diagrammi SSD	10
6.1	SSD aggiunta ingrediente	10
6.2	SSD aggiunta ricetta	11
7	Diagrammi di stato	12
7.1	Diagramma di stato Aggiungi ricetta	12
7.2	Diagramma di stato Aggiungi lotto	13
8	Diagrammi di sequenza di progettazione	14
8.1	Diagramma di sequenza aggiunta ingredienti	14
8.2	Diagramma di sequenza aggiunta ricette	15
9	Diagrammi di attività	16
9.1	Diagramma di attività Birra del Giorno	16
10	Diagramma dell'architettura software	17
11	Design principles	18
11.1	Liskov Substitution Principle	18
12	Architectural patterns	19
12.1	MVC	19
12.2	Data mapper	19
12.3	Dependent Mapping	19

13 Tecnologie utilizzate	21
13.1 SQLite	21
13.2 install4j	21
13.3 SWT	21

1 Introduzione

L'applicazione Brew Day offre un servizio di organizzazione delle attività di homebrewing permettendo al cliente di memorizzare le proprie ricette, tenere nota dei risultati ottenuti per ogni lotto prodotto e gestire il magazzino degli ingredienti

2 Analisi dei requisiti

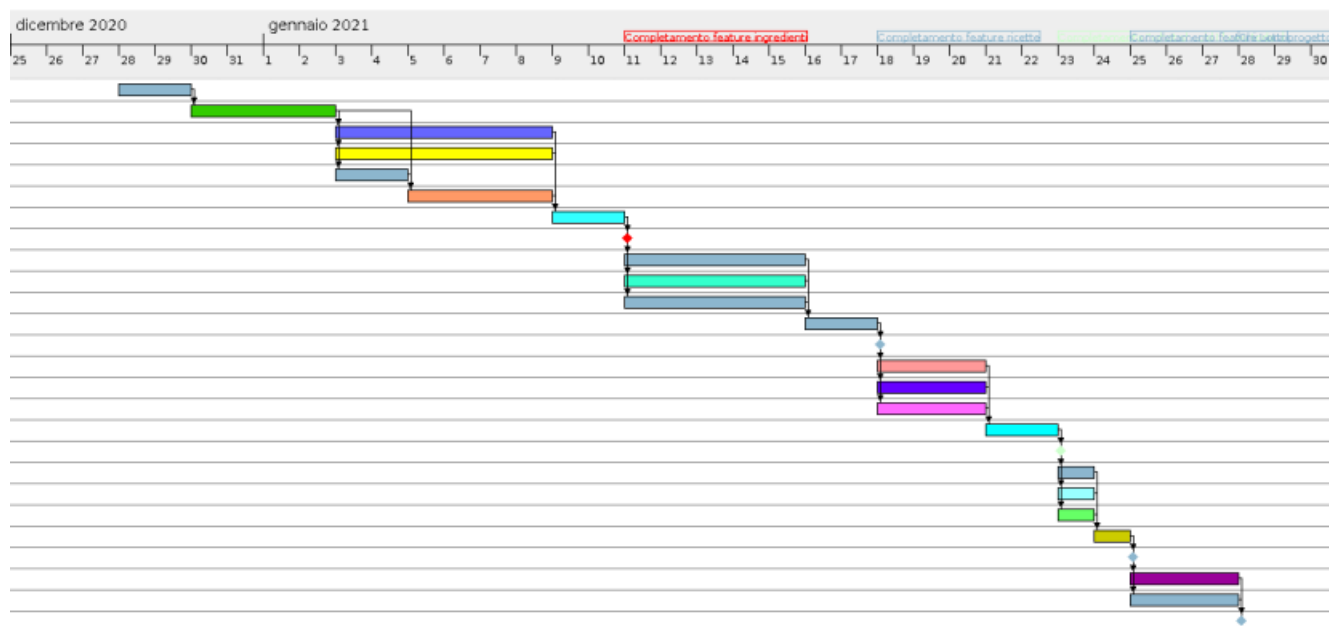
ID	Descrizione	Tipo	Priorità
1	L'utente deve poter registrare una propria password personale al primo avvio dell'applicazione	Autenticazione Funzionale	M
2	L'utente deve poter effettuare il login specificando la propria password ad ogni avvio dell'applicazione successivo al primo	Autenticazione Funzionale	M
3	Il sistema dovrà permettere di visualizzare una lista degli ingredienti disponibili inseriti dall'utente, dovrà inoltre permettergli di eliminare o modificare tali ingredienti	Ingredienti Funzionale	M
4	Il sistema Brew Day dovrà permettere all'utente di visualizzare nella home la lista di ricette inserite dall'utente permettendogli di eliminarle o modificarle, dovrà inoltre permettergli di inserire nuove ricette attraverso un'apposita schermata.	Ricette Funzionale	M
5	All'avvio dell'applicazione per la prima volta, dopo la registrazione, il sistema deve obbligatoriamente permettere all'utente di inserire il proprio equipaggiamento prima di compiere ulteriori azioni	Equipaggiamento Funzionale	M
6	Il sistema Brew Day deve notificare la scarsità degli ingredienti attraverso un'apposita vista	Lista della Spesa Funzionale	S
7	Il sistema Brew Day deve permettere all'utente di modificare il proprio equipaggiamento, specificando la sua nuova capacità produttiva	Equipaggiamento Funzionale	M
8	Ad ogni ciclo produttivo il sistema Brew Day deve aggiornare le disponibilità degli ingredienti	Lista della Spesa Funzionale	M
9	Ogni volta che viene prodotta una ricetta il sistema Brew Day deve permettere all'utente di aggiungere una nota sul lotto prodotto, eventualmente dandone anche una valutazione	Nota Funzionale	S

3 Diagramma Gantt e organizzazione del lavoro

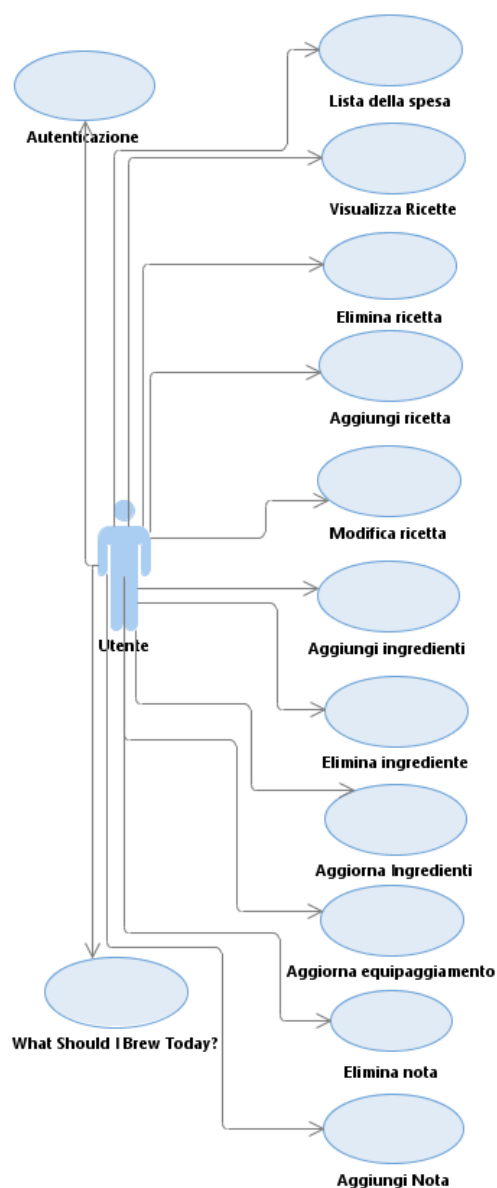
3.1 Descrizione delle attività

Nome	Data d'inizio	Data di fine
• Inizio progetto e analisi tempi	28/12/20	29/12/20
• Analisi e progettazione	30/12/20	02/01/21
• Analisi tecnologia DB	03/01/21	04/01/21
• Implementazione interfaccia grafica per feature ingredienti	03/01/21	08/01/21
• Implementazione backend feature ingredienti	03/01/21	08/01/21
• Implementazione Database ingredienti	05/01/21	08/01/21
• Documentazione analisi sonarqube e miglioramento codice	09/01/21	10/01/21
• Completamento feature ingredienti	11/01/21	11/01/21
• Implementazione interfaccia grafica Ricette	11/01/21	15/01/21
• Implementazione backend feature ricette	11/01/21	15/01/21
• Database feature ricette	11/01/21	15/01/21
• Documentazione, analisi sonarqube e miglioramento codice	16/01/21	17/01/21
• Completamento feature ricette	18/01/21	18/01/21
• Implementazione interfaccia grafica feature "what should I brew today?" + lista della spesa	18/01/21	20/01/21
• Implementazione backend feature "what should I brew today?" + lista della spesa	18/01/21	20/01/21
• Database "what should I brew today?" + lista della spesa	18/01/21	20/01/21
• Documentazione e supporto feature "what should I brew today?", spesa e analisi sonarqube e miglioramento codice	21/01/21	22/01/21
• Completamento "what should I brew today?" + lista spesa	23/01/21	23/01/21
• Implementazione interfaccia grafica feature Lotto	23/01/21	23/01/21
• implementazione backend feature Lotto	23/01/21	23/01/21
• implementazione database feature Lotto	23/01/21	23/01/21
• Documentazione, supporto, analisi sonarqube	24/01/21	24/01/21
• Completamento feature Lotto	25/01/21	25/01/21
• Completamento documentazione	25/01/21	27/01/21
• Packaging	25/01/21	27/01/21
• Chiusura progetto	28/01/21	28/01/21

3.2 Calendario delle attività



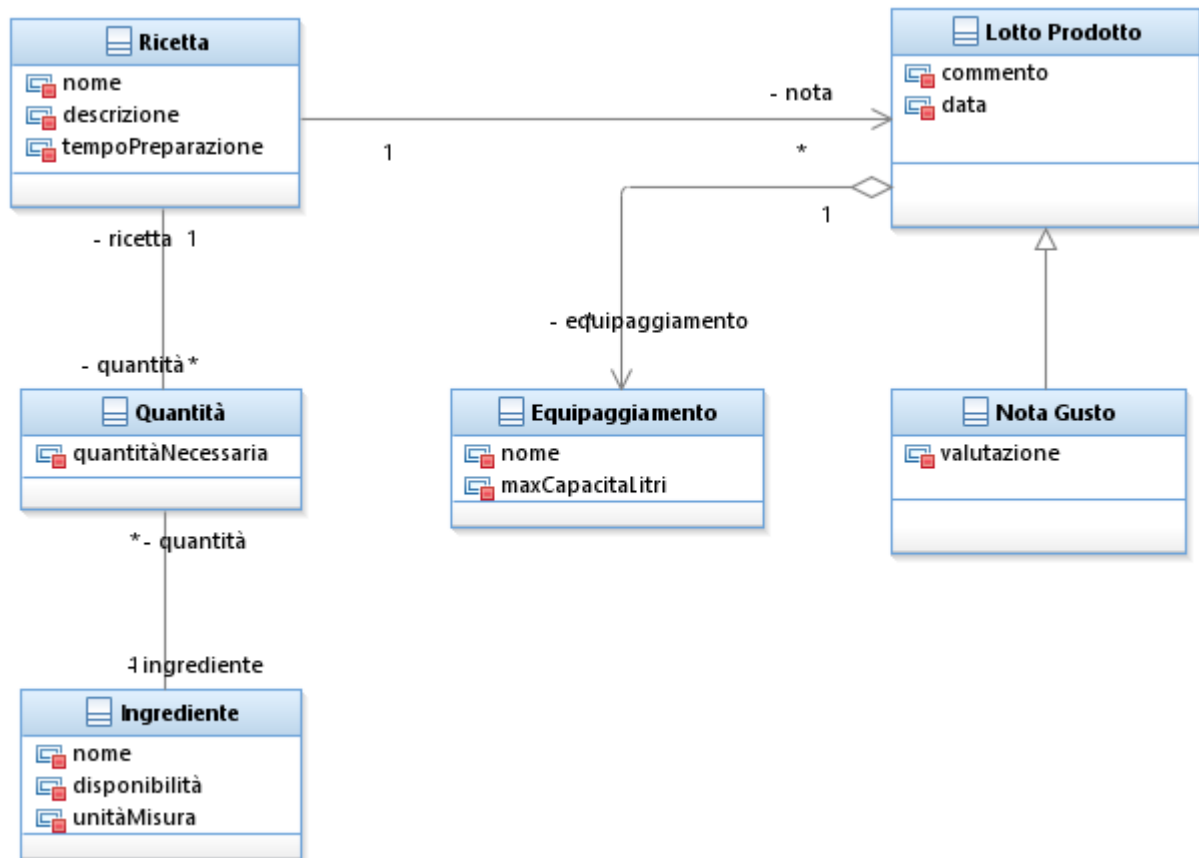
4 Diagramma dei casi D'uso



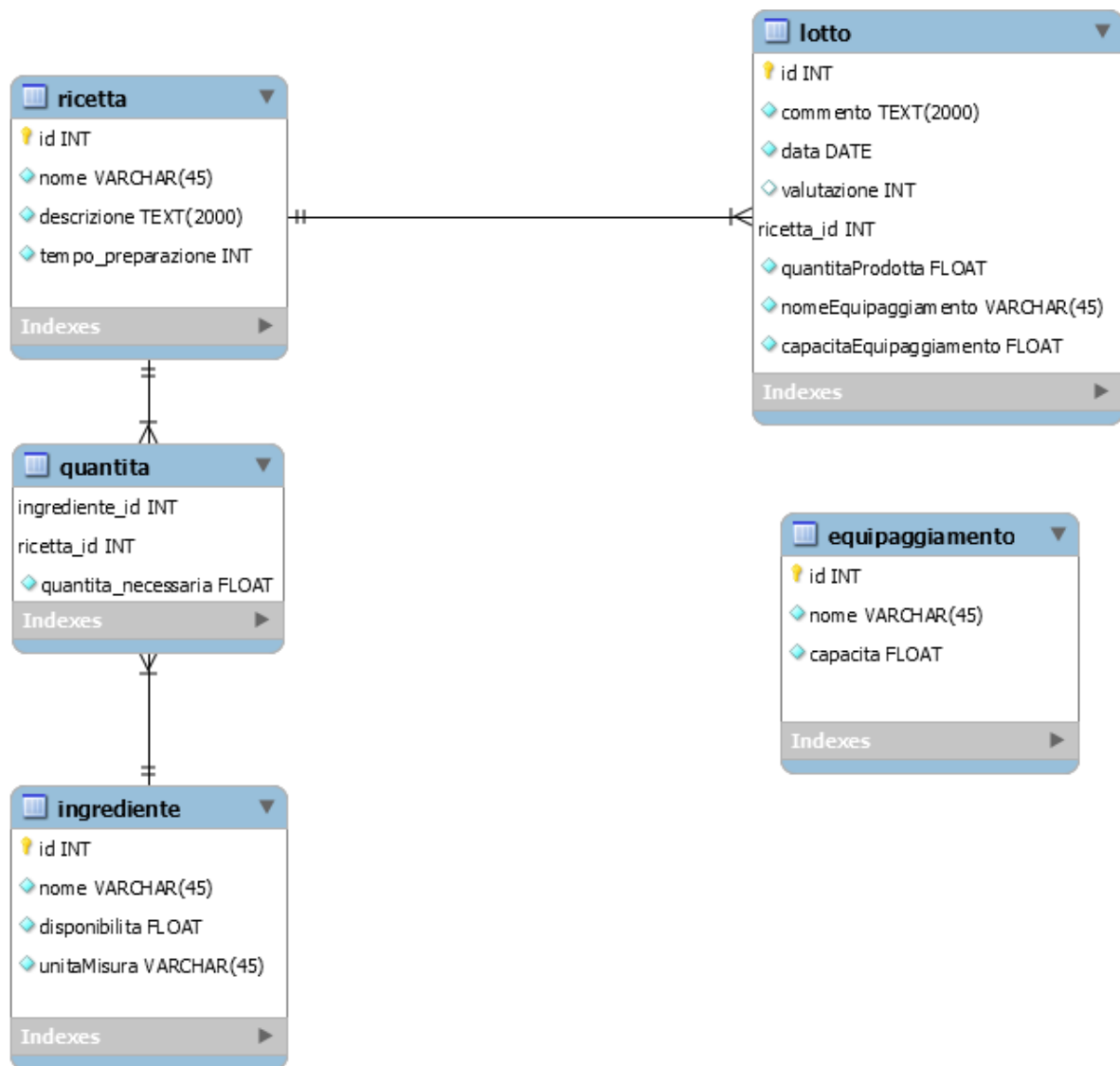
Brew Day è un'applicazione di discrete dimensioni, per semplicità e chiarezza sono stati definiti dei casi d'uso non eccessivamente grandi. Sono stati definiti in media 3 casi d'uso per ogni sfera di funzionalità del sistema (Ricetta, Ingrediente, Nota, Autenticazione, Equipaggiamento).

5 Diagrammi di dominio

5.1 Diagramma delle classi di dominio

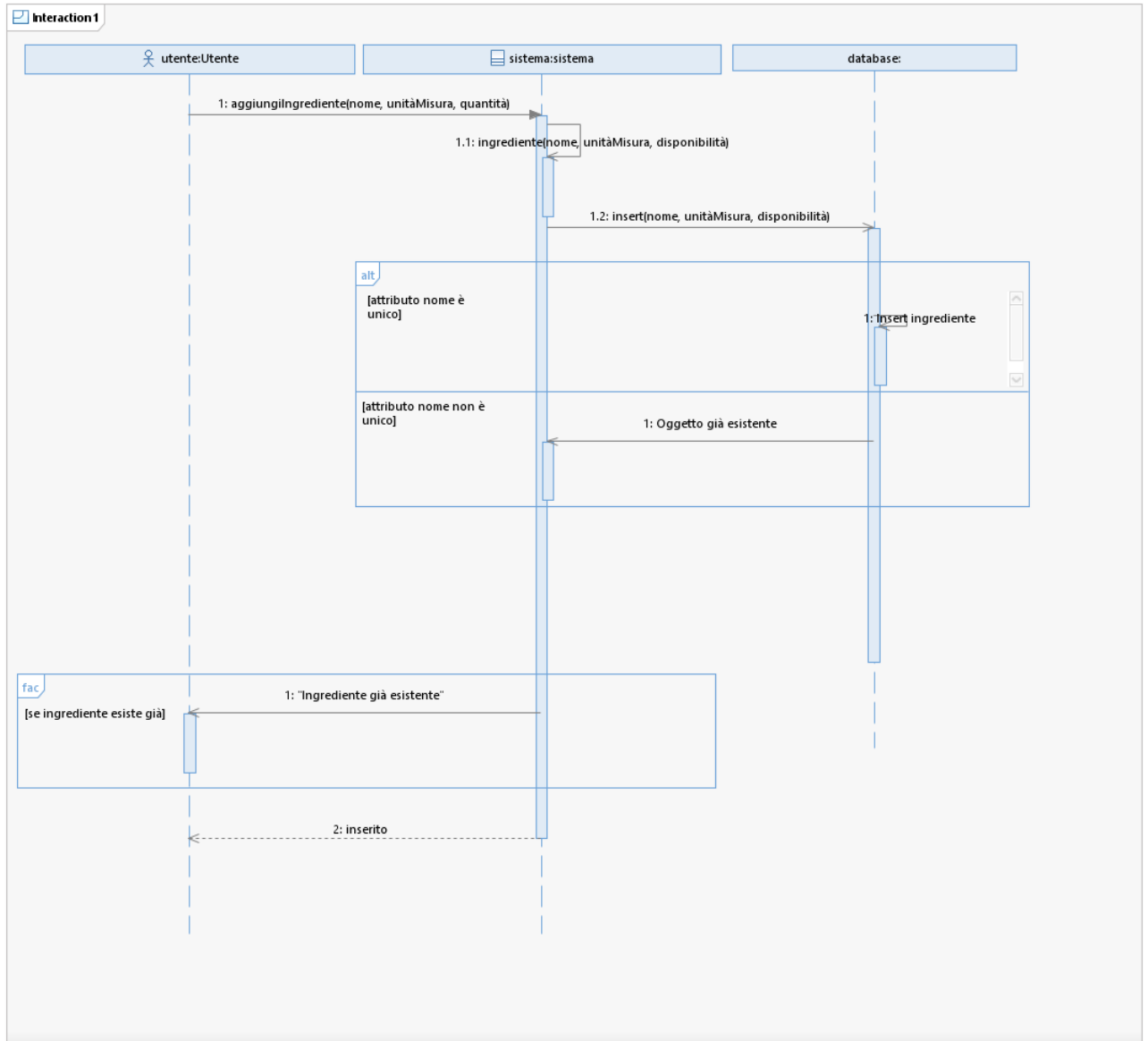


5.2 Diagramma EER

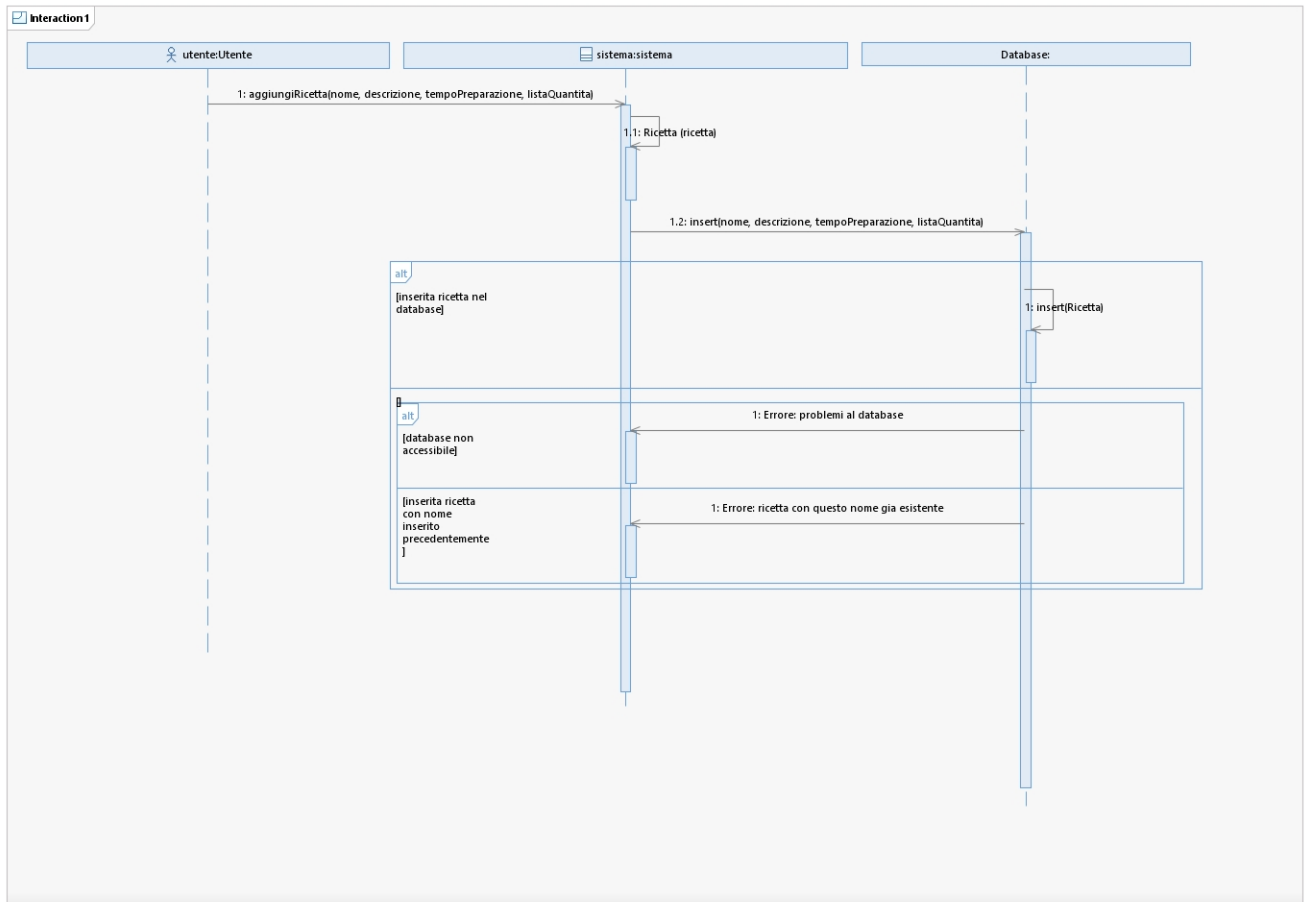


6 Diagrammi SSD

6.1 SSD aggiunta ingrediente

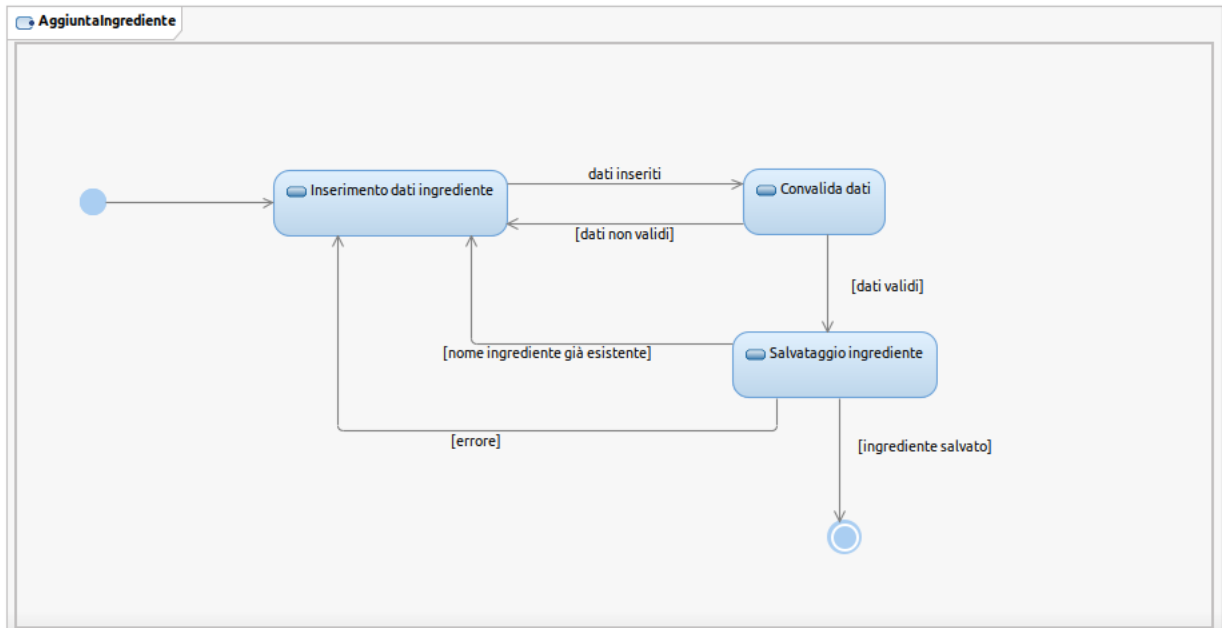


6.2 SSD aggiunta ricetta

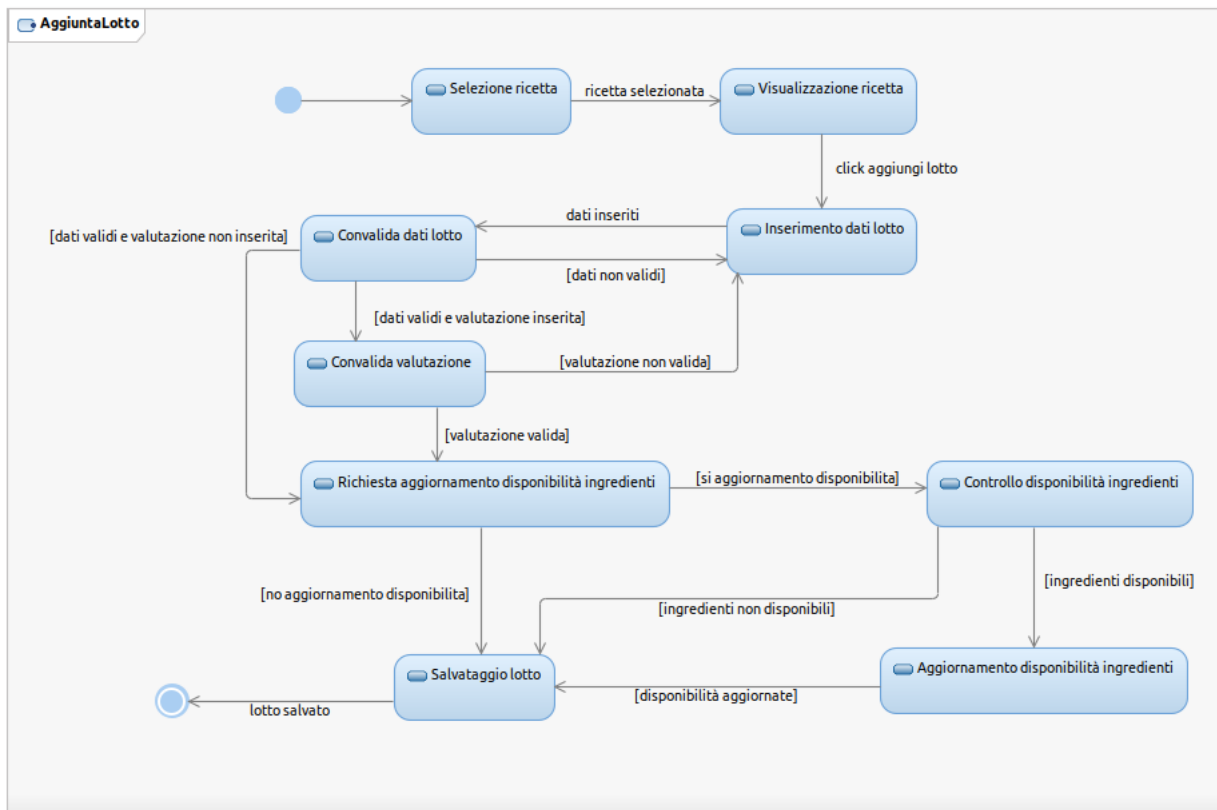


7 Diagrammi di stato

7.1 Diagramma di stato Aggiungi ricetta

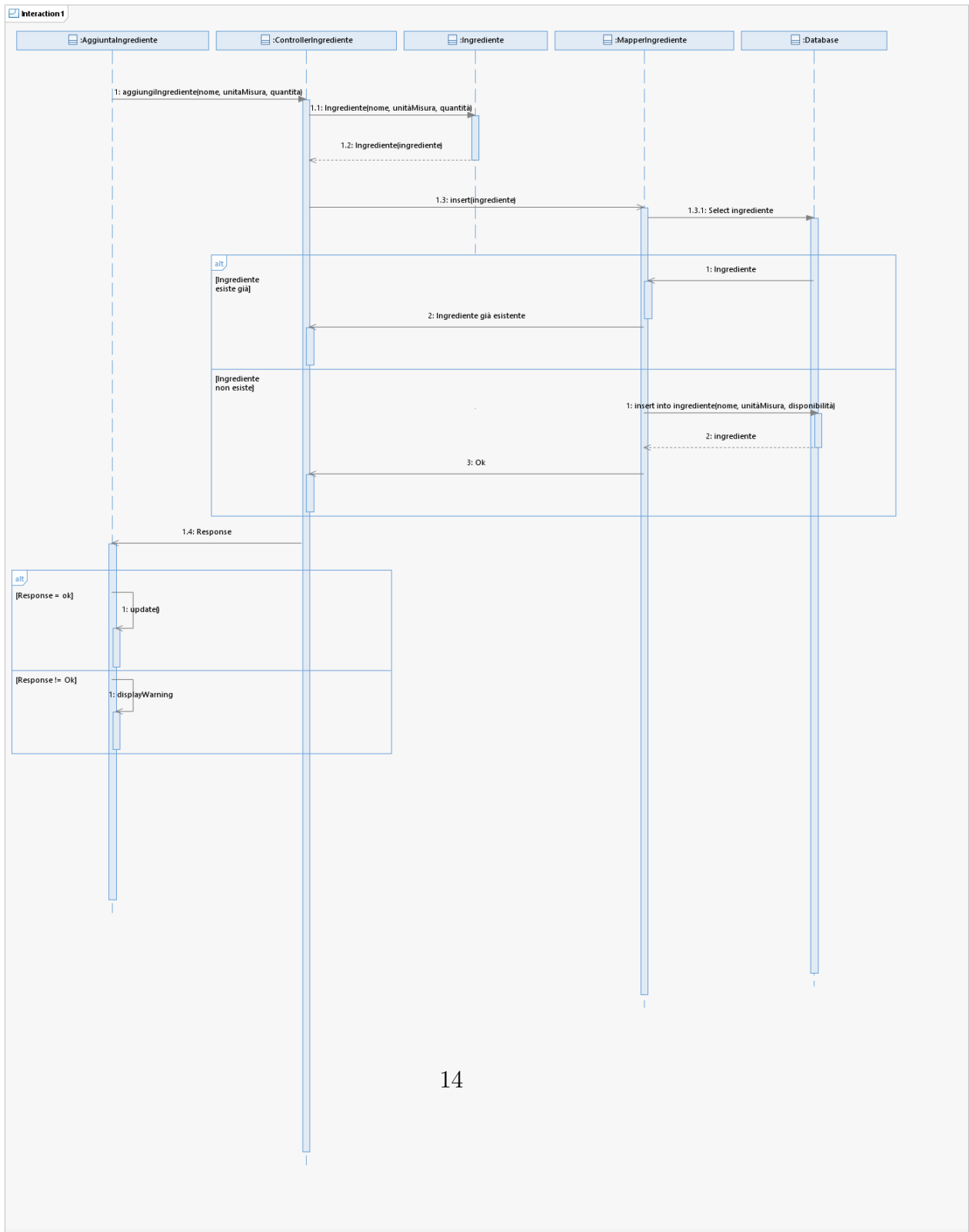


7.2 Diagramma di stato Aggiungi lotto

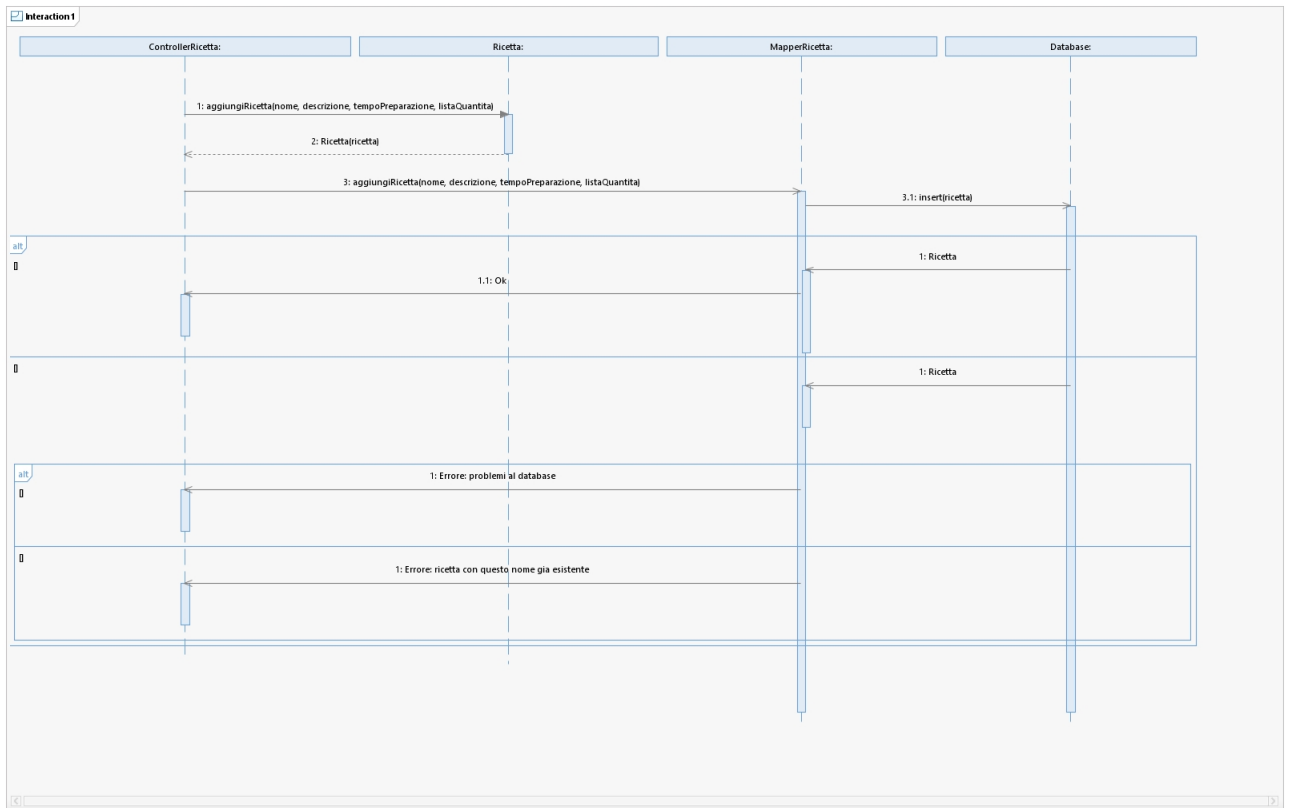


8 Diagrammi di sequenza di progettazione

8.1 Diagramma di sequenza aggiunta ingredienti

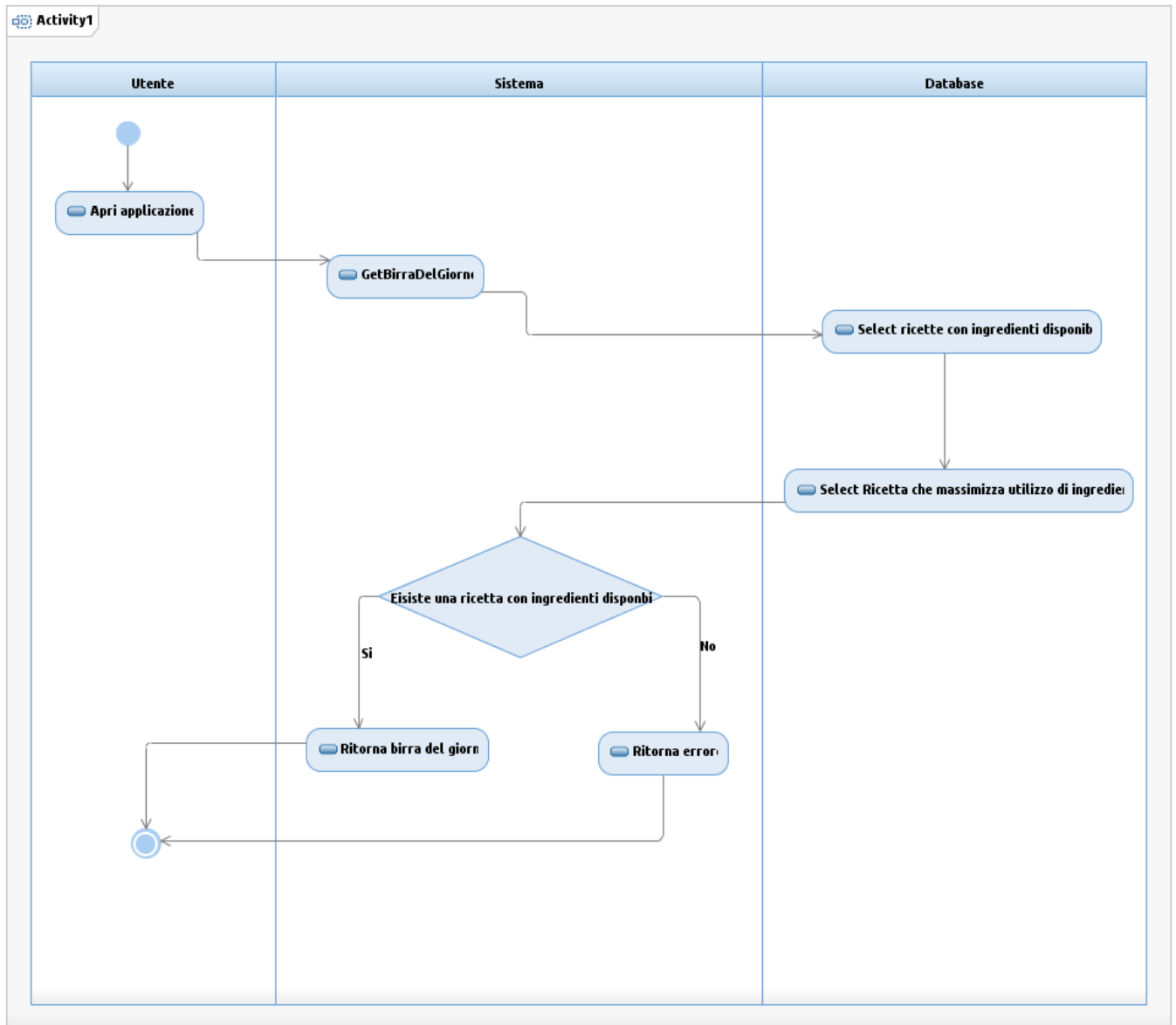


8.2 Diagramma di sequenza aggiunta ricette



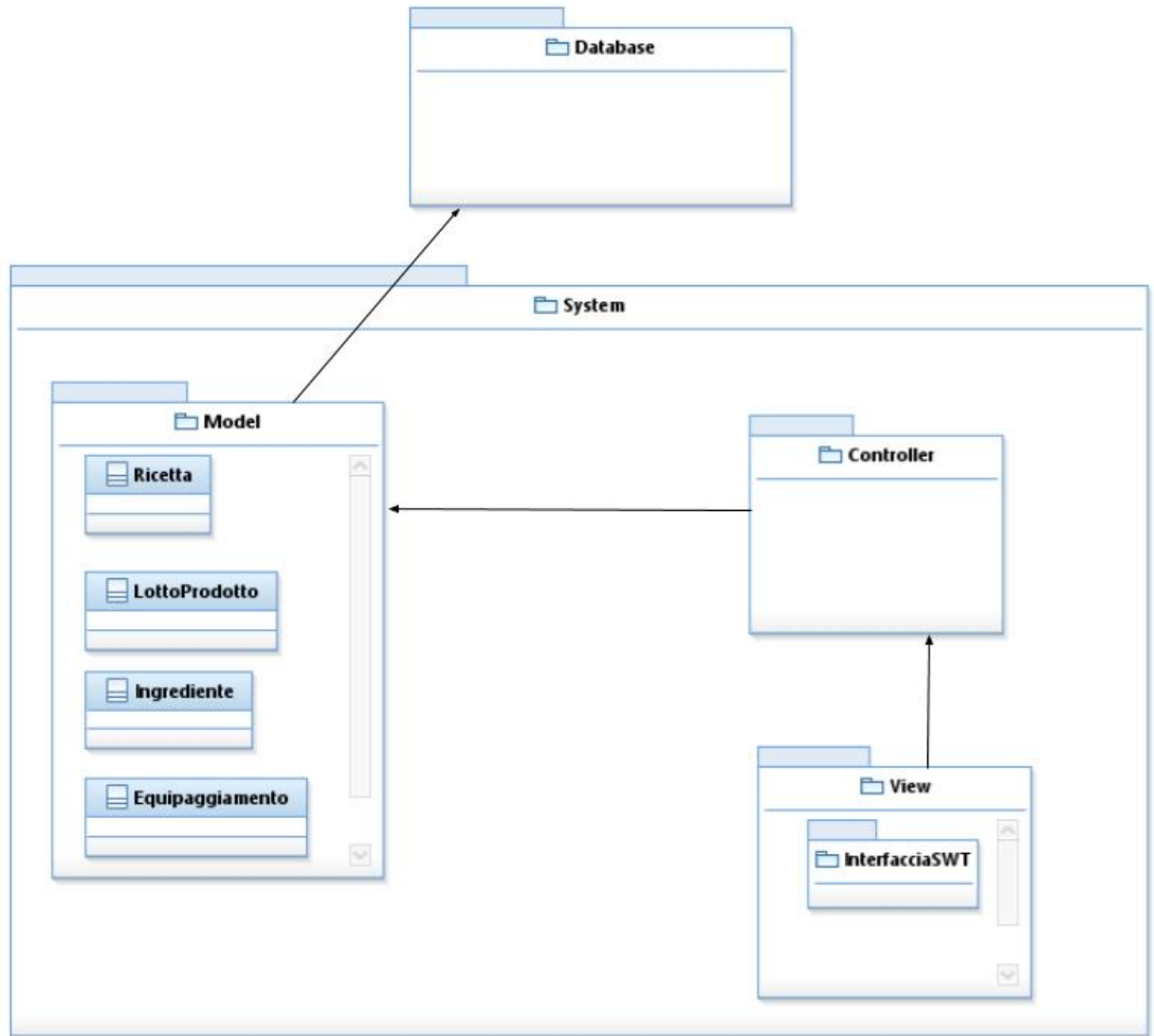
9 Diagrammi di attività

9.1 Diagramma di attività Birra del Giorno



10 Diagramma dell'architettura software

L'architettura di Brew Day è stata progettata secondo un classico MVC a cui è stato aggiunto un layer Database per la memorizzazione dei dati.



11 Design principles

11.1 Liskov Substitution Principle

Nell'ambito della funzionalità delle note aggiunte per ogni lotto prodotto non è stata effettuata una divisione netta tra note senza valutazione e note con valutazione, queste ultime infatti sono una sottoclasse delle prime. Secondo il principio LSP per qualsiasi operazione svolta dall'utente su nota senza valutazione essa è valida anche per oggetti di tipo nota con valutazione.

12 Architectural patterns

12.1 MVC

Per la progettazione del sistema Brew Day si è adottato il pattern architetturale MVC che separa la vista con cui l'applicazione comunica con l'utente (View) dalla gestione logica dei comandi (controller) e dai metodi per accedere ai dati(model). Al pattern MVC sono stati inoltre aggiunti due layer: Database e Security.

Il layer Database si occupa di memorizzare e salvare tutte le informazioni riguardanti l'applicazione e i dati utili all'utente (ricette, ingredienti, equipaggiamento). Poichè la quantità di informazioni gestite dal layer database non è eccessivamente grande si è deciso di evitare un approccio con server locale (es. MySQL), si è deciso invece di creare un file locale contenente tali informazioni attraverso SQLite. Si tratta di una libreria software che implementa un DBMS, ma a differenza di MySQL è molto compatta e veloce, ottima per applicazioni di questo tipo. SQLite permette la creazione di un file locale a cui è possibile accedere solo tramite l'applicazione Brew Day e che svolge il compito di Database.

Il layer security si occupa dell'autenticazione dell'utente. La prima volta che l'utente accede all'applicazione prima di svolgere qualsiasi operazione deve registrare una password personale. Una volta inserita la password il layer security si occupa di creare un file locale criptato (security.bd) contenente la password che l'utente dovrà inserire ogni volta che effettuerà l'accesso.

12.2 Data mapper

L'applicazione Brew Day fa largo uso del Data Source Architectural Pattern Data Mapper. All'interno del layer database sono state infatti definite le classi MapperIngrediente, MapperRicetta, MapperLotto e MapperNota. Queste classe servono da layer intermedio tra gli oggetti nel model e le tabelle del database mantenendo indipendenza tra oggetti in memoria dell'applicazione e oggetti del database.

12.3 Dependent Mapping

In relazione al Data Mapper è stato sfruttato anche il pattern Dependent Mapping che con efficienza, tramite un'unica classe Mapper di una classe madre, gestisce le operazioni CRUD a livello database di una sottoclasse che contiene una collezione di oggetti della classe parent. In questo caso la classe parent è Ricetta, essa infatti per ogni oggetto

contiene una collezione di tipo Quantità. Tutte le operazioni di lettura, aggiornamento e eliminazione degli oggetti quantità vengono effettuate nella classe MapperRicetta.

13 Tecnologie utilizzate

13.1 SQLite



SQLite è una libreria software che implementa un DBMS incorporandolo nell'applicazione stessa. E' un approccio molto più leggero e semplice rispetto all'implementazione di un server con MySQL, perfetto per piccole applicazioni. La tecnologia di SQLite permette la creazione di un file locale che svolge la funzione di database. Nello specifico di Brew Day tale database è accessibile solo attraverso una password criptata contenuta in un file security, in modo tale da evitare che chiunque non sia proprietario dell'applicazione possa compiere azioni illegittime.

13.2 install4j



install4j è un software professionale che partendo da un codice sorgente permette di generare un file eseguibile del programma che illustra all'utente finale i passi per l'installazione. Essendo un software a pagamento per la costruzione dell'eseguibile di Brew Day è stata utilizzata la prova gratuita di 90 giorni.

13.3 SWT

SWT è una libreria Java per la programmazione di interfacce grafiche. Attraverso la modellazione manuale del lay out grafico SWT genera il rispettivo codice automatico.