UNIVERSITÀ
di VERONA
Dipartimento
di INFORMATICA

Documentazione Progetto Ingegneria del Software

A.A. 2019/2020

COVID Market

Filippo Ziche - VR429647
Anna Dalla Vecchia - VR42961

Contents

1 Specifiche del progetto

Si vuole progettare un sistema informatico per gestire il servizio di spesa on-line di un supermercato.

Per accedere al sistema, gli utenti devono essere registrati. Per gli utenti si memorizzano nome, cognome, indirizzo, CAP, città, numero di telefono ed email. Ogni utente registrato accede con email e password ed ha eventualmente associata una tessera fedeltà per la raccolta punti. Ogni tessera fedeltà ha un numero identificativo, una data di emissione e il totale dei punti raccolti. Gli utenti possono specificare un metodo di pagamento preferito (con carta di credito, con paypal o alla consegna).

Quando l'utente vuole effettuare una spesa, accede al sistema con le sue credenziali ed inizia a comporre il carrello della spesa. I prodotti disponibili sono visualizzati per reparto (Carne, Pesce, Frutta e Verdura, Alimentari, ...). Per ogni prodotto viene visualizzato il nome, la marca, la quantità contenuta nella confezione, il prezzo e una immagine che lo rappresenta. Per ogni prodotto si visualizza anche se è disponibile o meno. Potranno essere inseriti nel carrello solo prodotti disponibili al momento della spesa. Se un utente inserisce un prodotto che al momento della conferma della spesa non risulta più disponibile, il sistema segnala la cosa al cliente ed elimina il prodotto dal carrello. Dopo aver confermato la spesa, l'utente sceglie data e orario della consegna visualizzando le opzioni possibili.

L'utente può visualizzare il carrello per modificare la quantità dei prodotti inseriti o rimuovere qualche prodotto. L'utente può ricercare i prodotti per tipo (uova, biscotti, pasta), per marca o per eventuali caratteristiche (senza glutine, bio, senza latte, ...). I prodotti possono essere visualizzati in ordine di prezzo crescente o decrescente o in ordine alfabetico rispetto alla marca.

Per ogni spesa si memorizzano il codice (univoco), la data prevista per la consegna, insieme all'intervallo di tempo in cui la spesa potrà essere consegnata, i prodotti che la compongono e in quale quantità con prezzo unitario e totale di ogni prodotto, l'utente che l'ha effettuata, il costo totale, il tipo di pagamento (carta di credito, paypal o alla consegna) e il saldo punti. Ad ogni spesa vengono accreditati sulla tessera fedeltà un numero di punti pari agli euro spesi nella spesa considerata.

Il sistema deve permettere agli utenti di accedere al loro profilo, modificare i dati anagrafici, verificare il saldo punti e lo stato delle loro spese. La spesa può essere confermata, in preparazione o consegnata. Ogni utente può vedere tutte le spese che ha effettuato nel tempo con il dettaglio dei prodotti acquistati.

I responsabili del reparto spesa on-line devono autenticarsi per poter accedere al sistema e devono poter verificare lo stato delle spese e provvedere all'inserimento delle informazioni relative ai prodotti. Il sistema memorizza la quantità di prodotti disponibili in magazzino. Per i responsabili di reparto si memorizzano la matricola, i dati anagrafici, il ruolo e le credenziali di accesso (login e password).

2 Casi d'uso

Dalla specifica abbiamo individuato i casi d'uso, ovvero le modalità con cui un attore può interagire con il sistema generando un risultato osservabile. Rappresentano un requisito funzionale del sistema in cui specifica cosa ci si aspetta dal sistema trascurando il suo specifico comportamento.

Per la rappresentazione dei casi d'uso abbiamo preferito la semplicità per una maggiore chiarezza rispetto ad un'elencazione più particolare che a nostro parere risulta superflua.

Il sistema informatico proposto permette l'utilizzo da parte di un utente previa registrazione oppure di un responsabile preventivamente inserito nel database. Abbiamo individuato tre attori del sistema:

2.1 Utente registrato

L'utente registrato può accedere al sistema e visualizzare il catalogo filtrandolo a suo piacimento, aggiungere e/o modificare i prodotti presenti nel suo carrello, può confermare e acquistare il carrello scegliendo una data di consegna, una fascia oraria e un metodo di pagamento. Inoltre può visualizzare il suo profilo e nel caso modificarlo, visualizzare la sua carta fedeltà e, nel caso non fosse presente, richiederne una nuova. Infine può visualizzare lo storico delle spese.

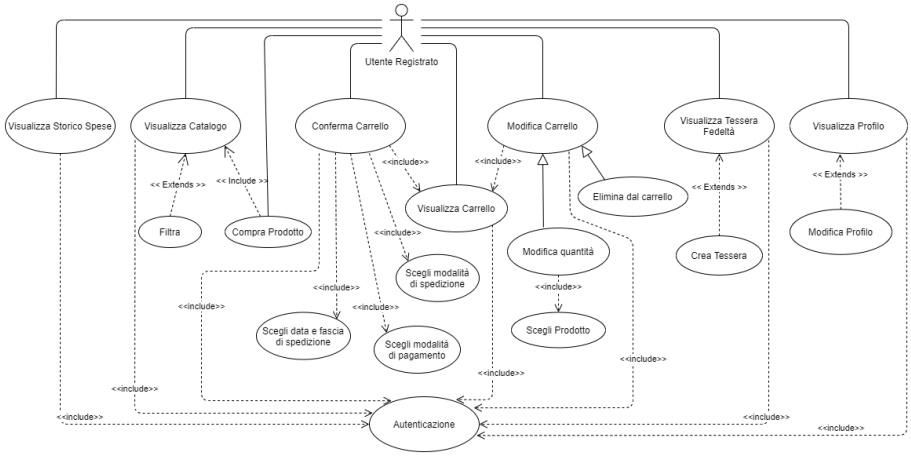

Figure 1: Casi d'uso utente registrato

2.2 Responsabile di reparto

Il responsabile di reparto può visualizzare e modificare solamente i prodotti dei reparti a lui assegnati, può visualizzare e modificare lo stato delle spese di tutto il sistema filtrandole per stato e/o per utente. Infine può visualizzare anche i suoi dati personali.

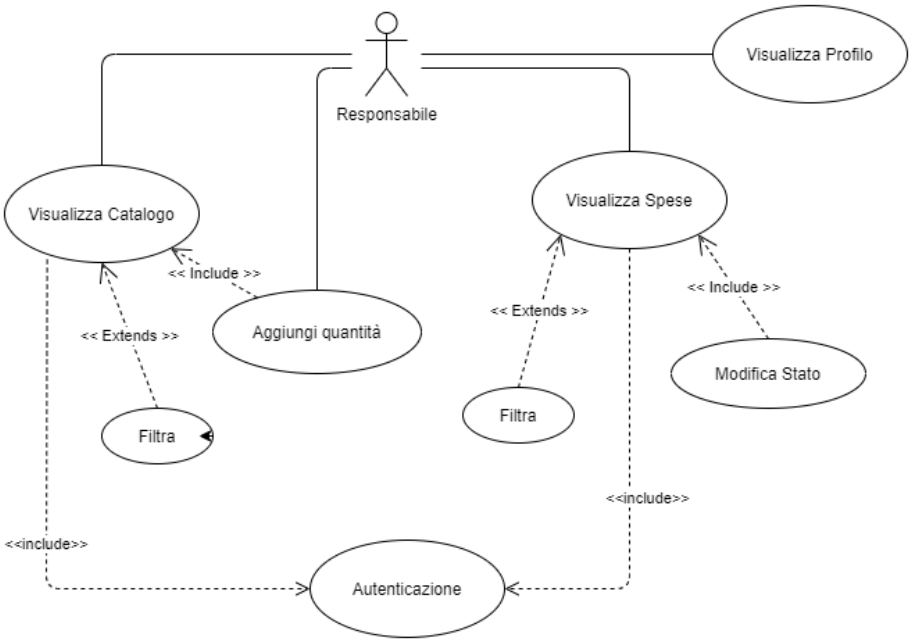

Figure 2: Casi d'uso responsabile reparto

2.3 Utente non registrato

L'utente non registrato può solamente registrarsi, non ha modo di accedere al sistema.

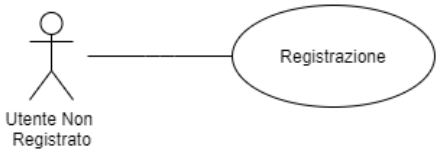

Figure 3: Casi d'uso utente non registrato

3 Schede di specifica dei casi d'uso

Seguono ora una serie di schede di specifiche più dettagliate rispetto agli schemi riportati sopra sui casi d'uso più importanti:

ID: Registrazione utente
ATTORI: Utente non registrato
PRECONDIZIONI:

1. L'utente non deve essere già presente all'interno del database.

SEQUENZA:

1. L'utente nella pagina di login clicca sulla scritta "Registrati"
2. Inserisce i suoi dati anagrafici
3. Sceglie un metodo di pagamento predefinito
4. Sceglie se creare una nuova Carta Fedeltà
5. Clicca sul bottone "Salva"
    a. Se i dati sono validi il processo è andato a buon fine
    b. Se i dati non sono validi allora torna al punto 2

POSTCONDIZIONI:

1. L'utente viene inserito nel database
2. L'utente accede al sistema e visualizza il catalogo

ID: Login Utente
ATTORI: Utente registrato, Responsabile
PRECONDIZIONI:

1. L'utente deve essere registrato nel database.

SEQUENZA:

1. L'attore inserisce le proprie credenziali
2. Preme il tasto "Login"
    a. Se le credenziali sono valide il login è andato a buon fine
    b. Se le credenziali non sono valide torna al punto 1

POSTCONDIZIONI:

1. L'attore accede al sistema e visualizza il catalogo

ID: Filtra
ATTORI: Utente registrato, Responsabile
PRECONDIZIONI:

1. L'attore deve aver eseguito l'accesso.
2. L'attore deve trovarsi in una schermata che contiene un filtro.

SEQUENZA:

1. L'attore inserisce i campi di filtraggio desiderati
2. Preme il tasto "Filtra"

POSTCONDIZIONI:

1. La schermata viene aggiornata per rispettare il filtro.

ID: Aggiungi prodotto al carrello utente
ATTORI: Utente registrato
PRECONDIZIONI:

1. L'attore deve aver eseguito l'accesso.
2. L'attore deve trovarsi nella schermata del catalogo

SEQUENZA:

1. L'attore seleziona un prodotto dalla lista
2. Modifica la quantità da comprare a piacimento
3. Clicca il bottone "Compra"

POSTCONDIZIONI:

1. Il prodotto viene inserito nel carrello dell'utente.

ID: Modifica il carrello dell'utente
ATTORI: Utente registrato
PRECONDIZIONI:

1. L'attore deve aver eseguito l'accesso.
2. L'attore deve trovarsi nella schermata del carrello

SEQUENZA:

1. L'attore seleziona un prodotto dalla lista
2. L'attore modifica la quantità del prodotto tramite lo spinner o lo elimina cliccando il bottone "Elimina dal carrello"

POSTCONDIZIONI:

1. Il carrello viene aggiornato.

ID: Conferma la spesa dell'utente
ATTORI: Utente registrato
PRECONDIZIONI:

1. L'attore deve aver eseguito l'accesso.
2. L'attore deve trovarsi nella schermata del carrello

SEQUENZA:

1. L'attore clicca "Conferma ordine"
2. L'attore inserisce la data di consegna, la fascia oraria
3. (opz.) Modifica il metodo di pagamento
4. L'attore clicca "Conferma"

POSTCONDIZIONI:

1. Viene creata la nuova spesa ed inserita nel database.
2. Aumentano i punti della carta fedeltà se questa è presente.
3. Il carrello dell'utente viene resettato.

ID: Modifica il profilo utente
ATTORI: Utente registrato
PRECONDIZIONI:

1. L'attore deve aver eseguito l'accesso.
2. L'attore deve trovarsi nella schermata del profilo

SEQUENZA:

1. L'attore clicca "Modifica"
2. L'attore aggiorna i dati del suo profilo
3. L'attore clicca "Salva"
    a. Se i dati sono validi il processo è andato a buon fine
    b. Se i dati non sono validi allora torna al punto 2

POSTCONDIZIONI:

1. Viene aggiornato il profilo utente.
2. Viene visualizzata la schermata profilo.

ID: Crea carta fedeltà
ATTORI: Utente registrato
PRECONDIZIONI:

1. L'attore deve aver eseguito l'accesso.
2. L'attore deve trovarsi nella schermata della tessera
3. L'attore non deve essere in possesso di una tessera

SEQUENZA:

1. L'attore clicca "Crea tessera"

POSTCONDIZIONI:

1. Viene creata una nuova tessera e assegnata all'utente.

ID: Inserisci prodotto al catalogo
ATTORI: Responsabile di reparto
PRECONDIZIONI:

1. L'attore deve aver eseguito l'accesso.
2. L'attore deve trovarsi nella schermata del catalogo

SEQUENZA:

1. L'attore clicca "Inserisci"
2. L'attore inserisce i dati relativi al nuovo prodotto
3. L'attore clicca "Aggiungi"
    a. Se i dati sono validi il processo è andato a buon fine
    b. Se i dati non sono validi allora torna al punto 2

POSTCONDIZIONI:

1. Viene creato il prodotto e inserito nel catalogo.

ID: Modifica stato di una spesa
ATTORI: Responsabile di reparto
PRECONDIZIONI:

1. L'attore deve aver eseguito l'accesso.
2. L'attore deve trovarsi nella schermata delle spese

SEQUENZA:

1. L'attore seleziona una spesa dalla lista
2. L'attore clicca "Prepara" o "Consegna"

POSTCONDIZIONI:

1. Viene aggiornato lo stato della spesa.

4 Diagramma delle sequenze

Precisiamo che i seguenti diagrammi sono stati semplificati rimuovendo chiamate secondarie che sono superflue per la comprensione della sequenza, ad esempio lambda espressioni e getters successivi al prima invocazione. Inoltre non visualizziamo tutti i sequence diagram in quanto molti di essi sono banali.

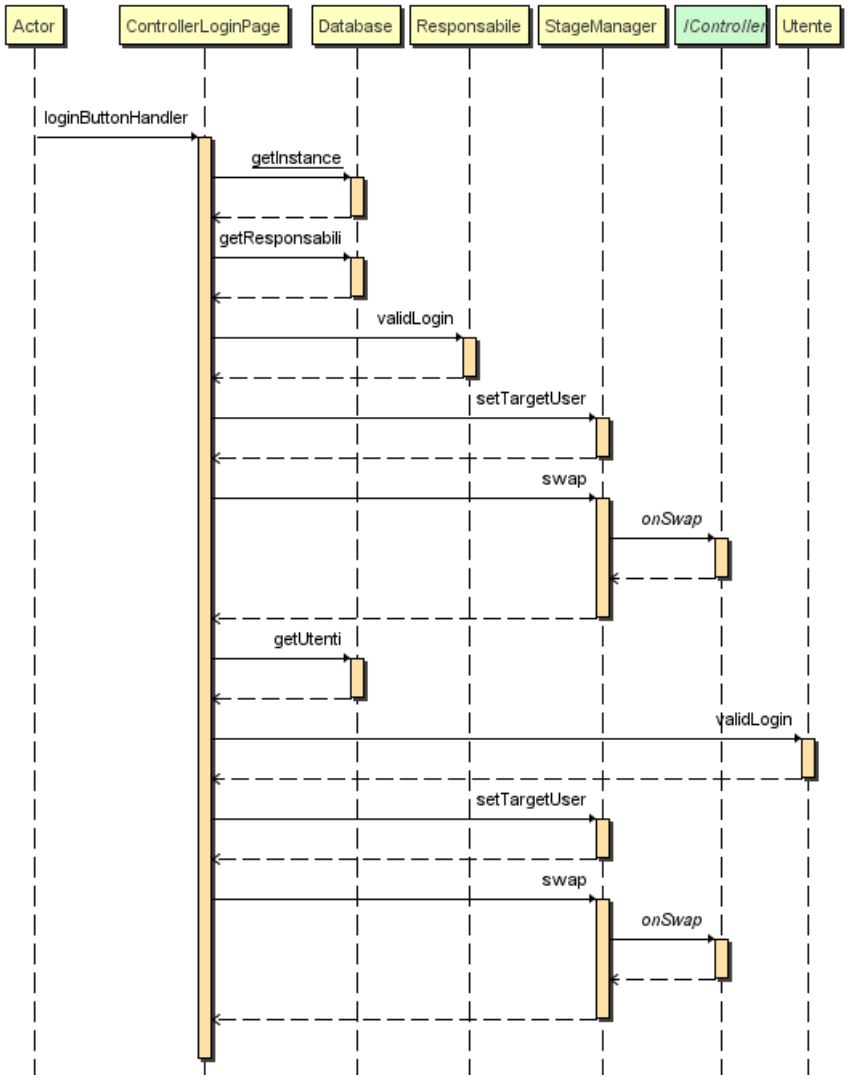

Figure 4: Sequenza del login, sia per utenti che per responsabili

Visualizziamo alcune delle sequenze relative all'utente registrato:

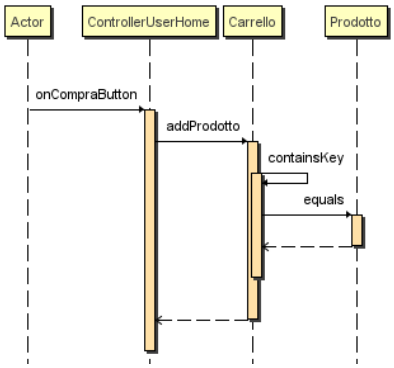

Figure 5: Sequenza acquisto dalla home

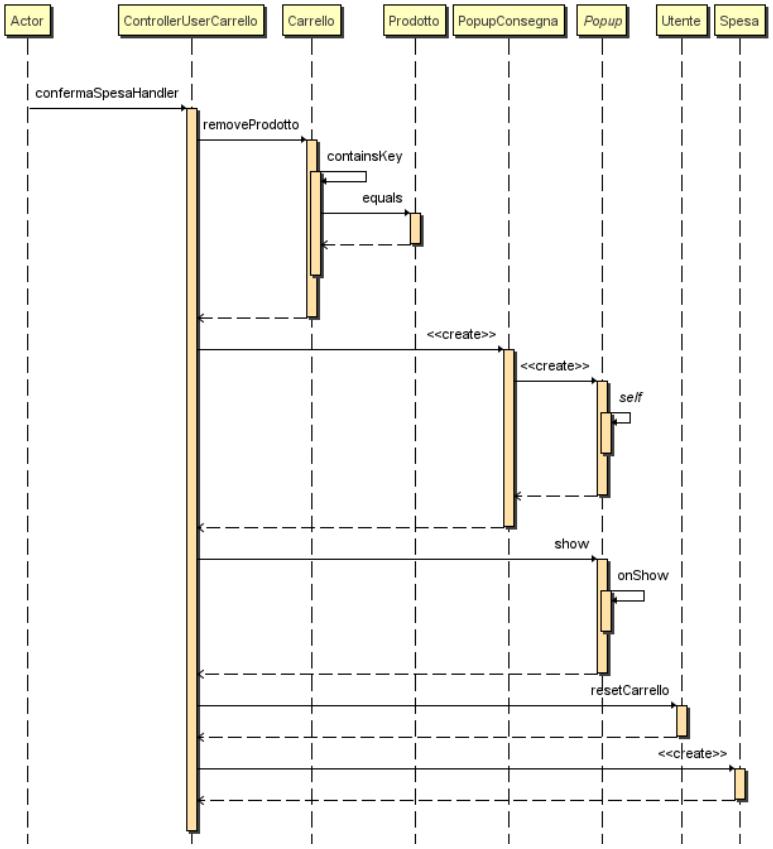

Figure 6: Sequenza di conferma carrello dell'utente

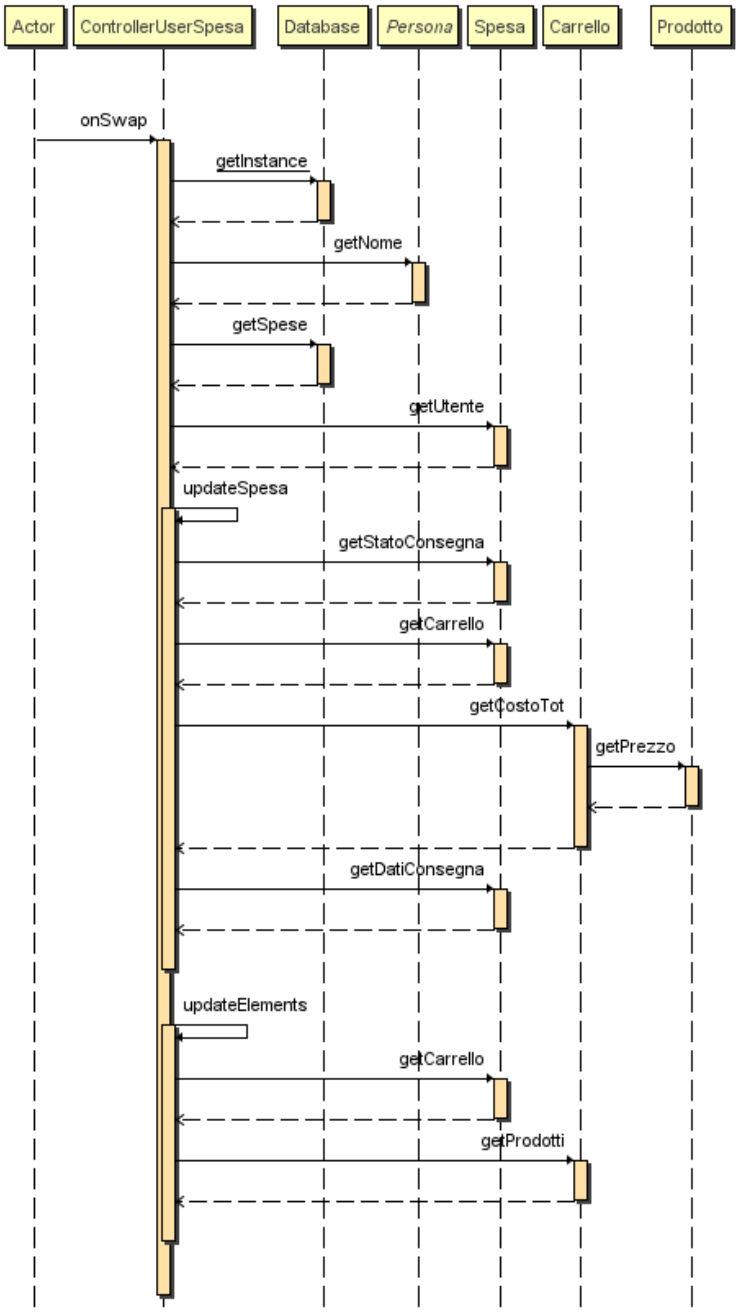

Figure 7: Sequenza di visualizzazione dello storico spese

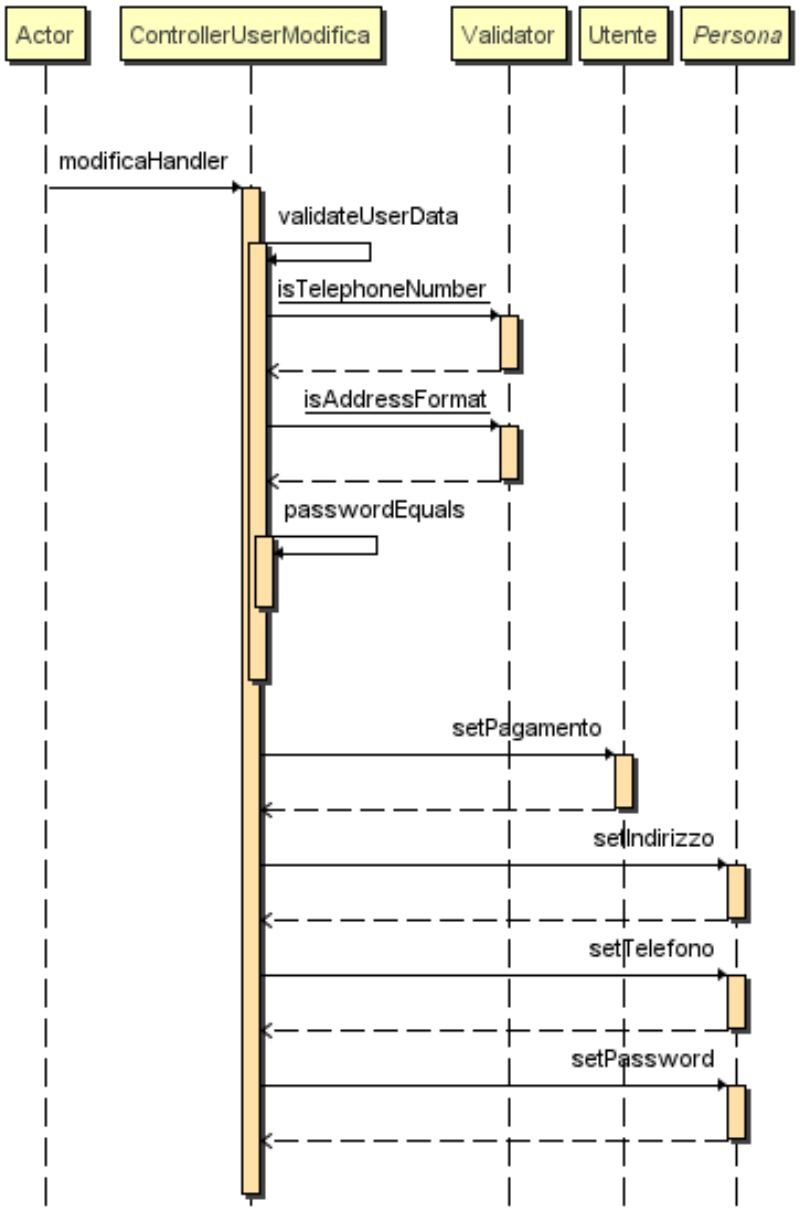

Figure 8: Sequenza di modifica profilo utente

Visualizziamo alcune delle sequenze relative al responsabile di reparto:

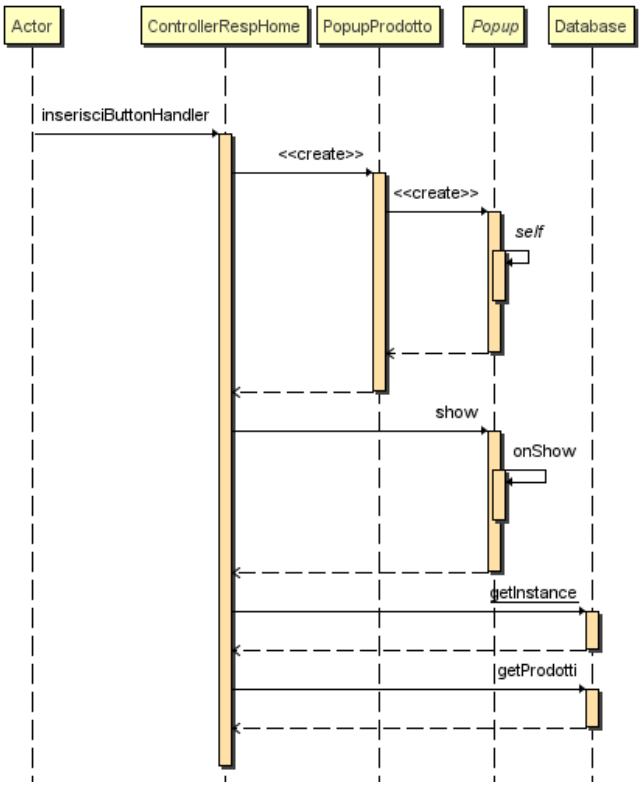

Figure 9: Sequenza di creazione prodotto nel catalogo

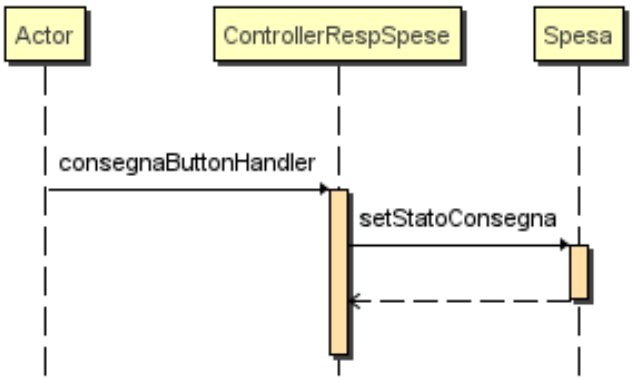

Figure 10: Sequenza di conferma/preparazione spesa

5 Diagramma delle attività

5.1 Registrazione utente

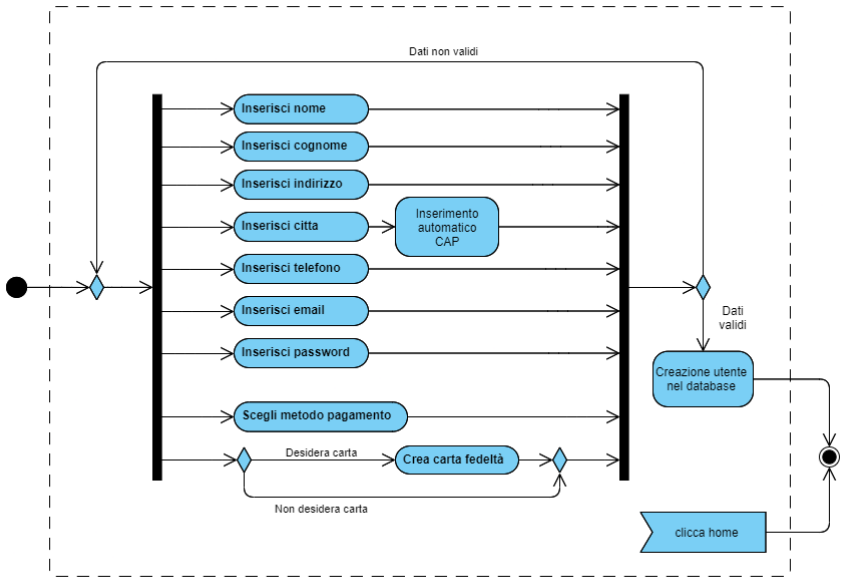

Figure 11: Diagramma di attività per la registrazione di un utente

5.2 Login

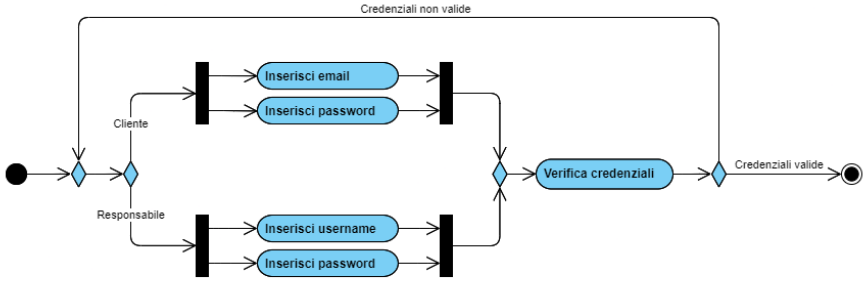

Figure 12: Diagramma di attività per il login di un utente o responsabile al sistema

5.3 Filtra catalogo

La seguente figura è un esempio di filtraggio che avviene nel catalogo utente e del responsabile, una procedura analoga ma con campi di filtraggio diversi avviene anche nella schermata stato spese del responsabile.

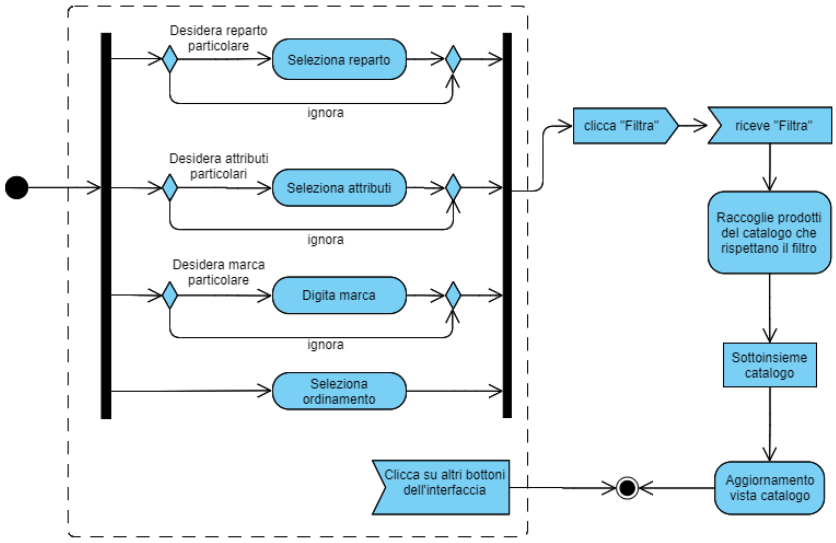

Figure 13: Diagramma di attività del settaggio del filtro del catalogo

5.4 Aggiunta di un prodotto al carrello

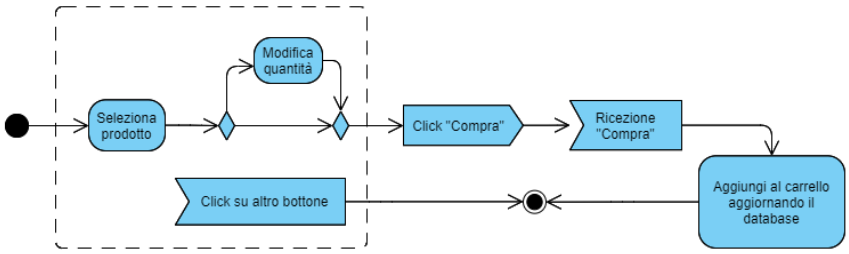

Figure 14: Diagramma di attività per l'inserimento di un prodotto al carrello dell'utente

5.5 Conferma acquisto del carrello

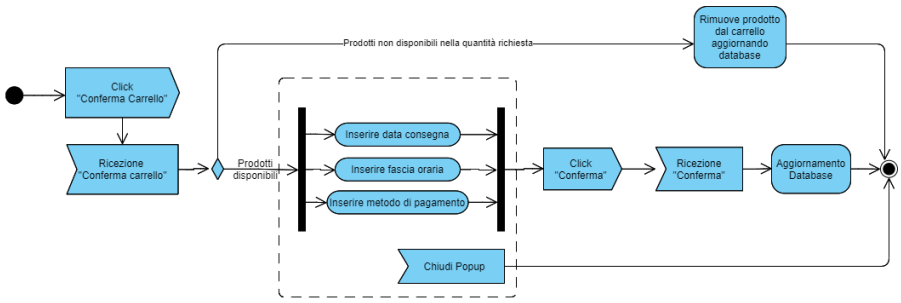

Figure 15: Diagramma di attività per la conferma del carrello e quindi la creazione di una nuova spesa per l'utente tramite popup

5.6 Modifica credenziali utente

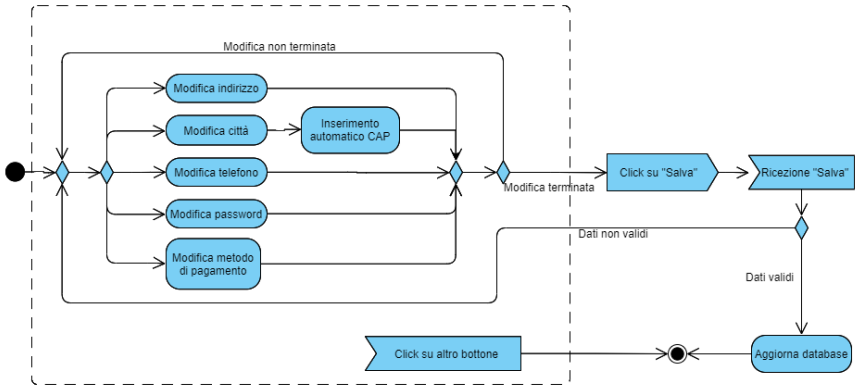

Figure 16: Diagramma di attività per la modifica del profilo di un utente

5.7 Aggiunta di un prodotto al catalogo

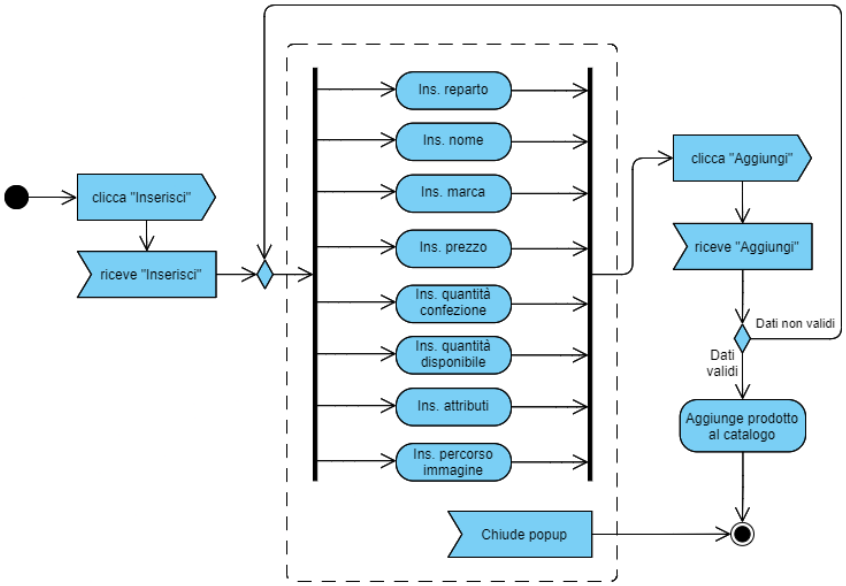

Figure 17: Diagramma di attività per l'aggiunta di un nuovo prodotto al catalogo tramite popup

5.8 Modifica stato spesa

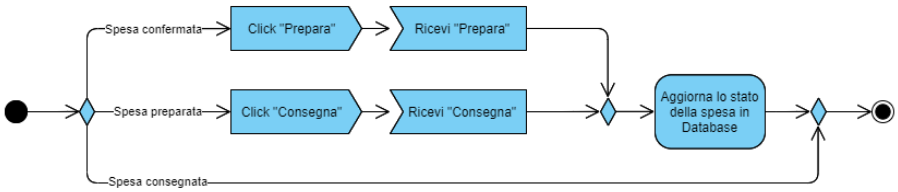

Figure 18: Diagramma di attività per la modifica di una spesa da parte del responsabile, che dunque sceglie se prepararla e/o confermarla

6 Diagramma delle classi

Mostriamo ora il diagramma delle classi dei gruppi principali che abbiamo individuato nel sistema.

6.1 Database

Il database contiene le liste dei responsabili, utenti, prodotti e spese che sono stati creati nel sistema.

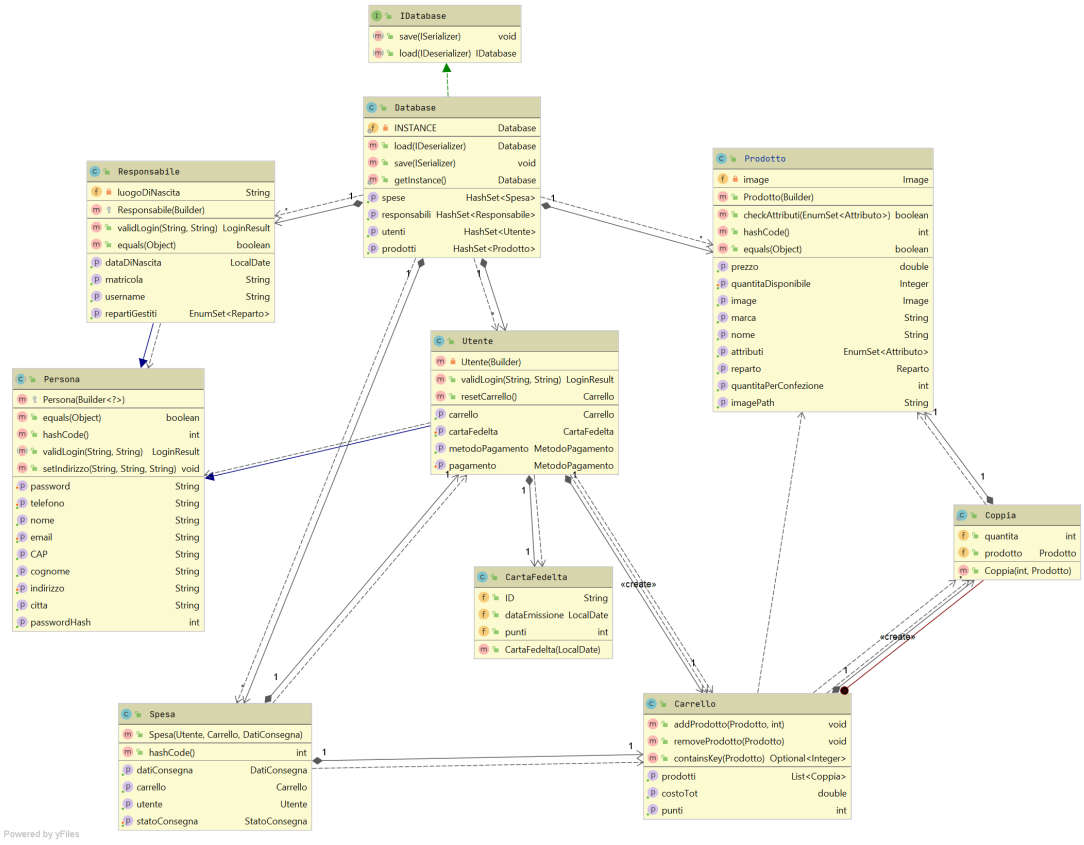

Figure 19: Diagramma delle classi che compongono il database

6.2 Serializer e Deserializer

Le due interfaccie ISerializer e IDeserializer permettono al database di essere salvato in formati diversi in base alla necessità, ad esempio, nel nostro caso, viene salvato in un file binario tramite BinarySerializer e ricaricato tramite BinaryDeserializer ma nulla vieta la possibilità di implementare nuove classi per salvare su file di testo o trasferire via web i dati su un dispositivo remoto.

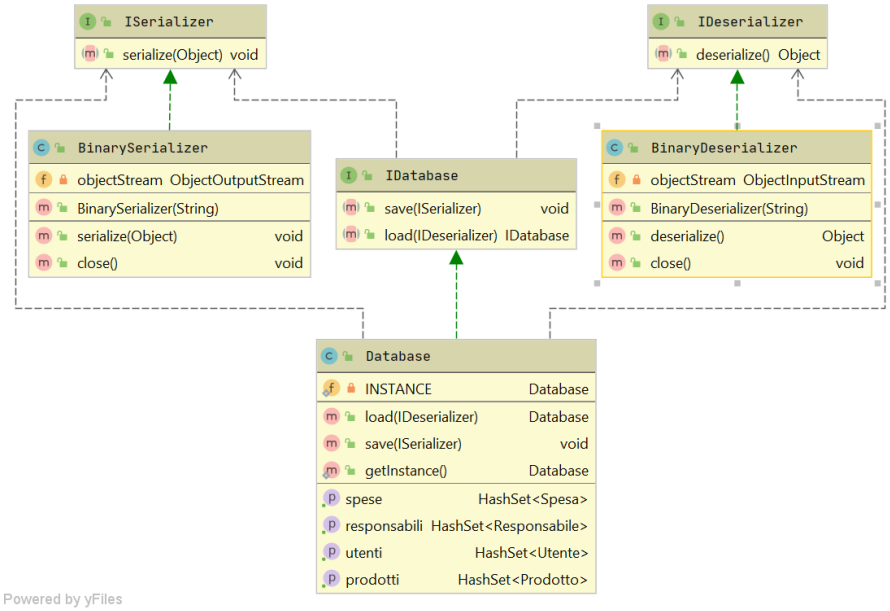

Figure 20: Diagrammi delle classi relative al salvataggio e caricamento del database

6.3 Controllers

Le seguenti classi sono addette al controllo dell'interfaccia utente, popolando i vari widget e rispondendo agli eventi, come il click di bottoni o l'inserimento di testo. Ne esiste una per ogni schermata dell'applicazione.

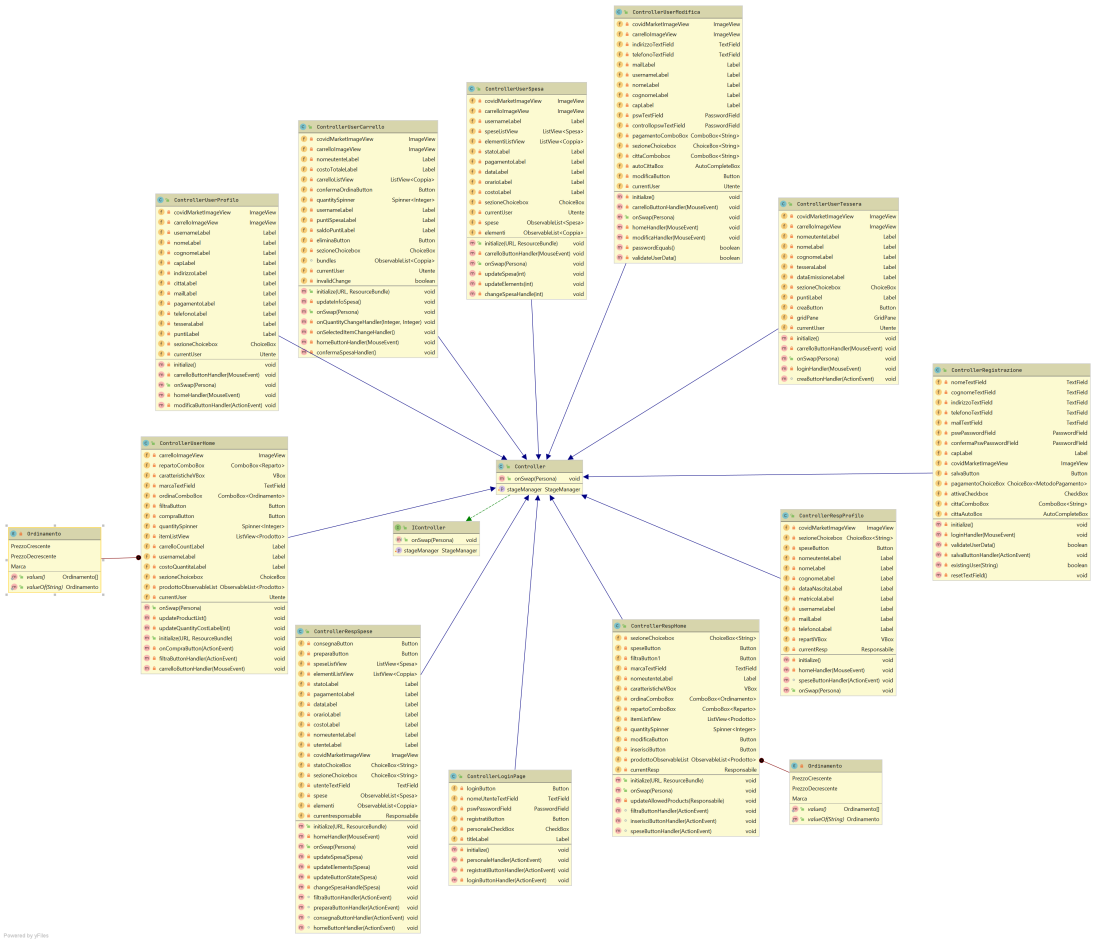

Figure 21: Diagrammi delle classi dei controllers

6.4 Popup

Queste classi speciali sono dei controllers per i popup dell'applicazione, ovvero schermate temporanee che servono solamente per ottenere dati dall'utente. PopupProdotto serve ad ottenere i dati di un nuovo prodotto del catalogo, mentre PopupConsegna per i dati di consegna di una spesa appena confermata.

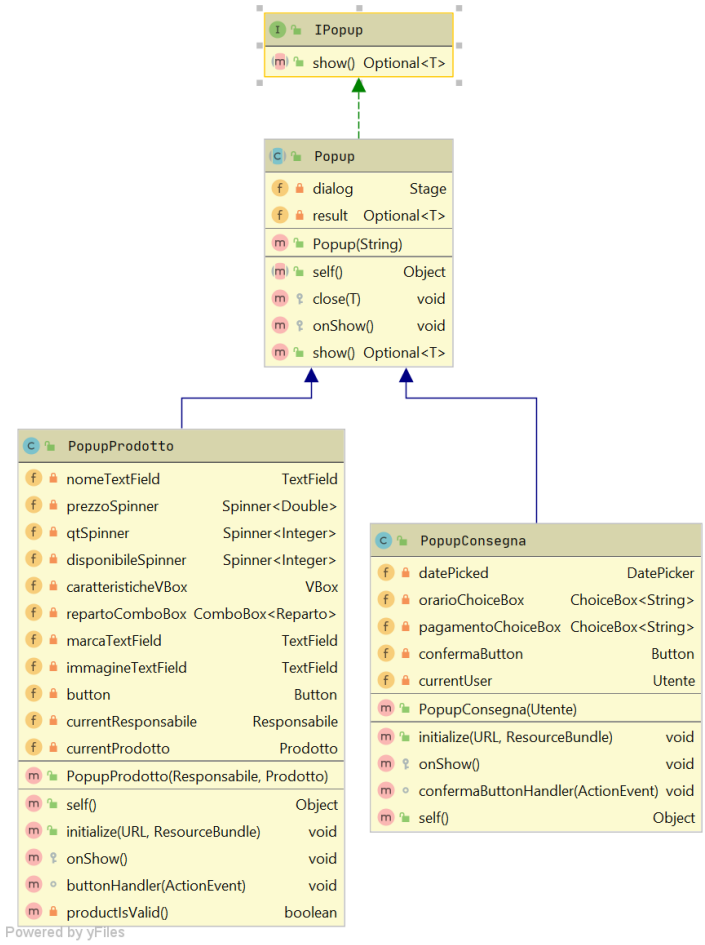

Figure 22: Diagrammi delle classi dei popup

6.5 StageManager

Tutti i controller sono gestiti dallo StageManager che ha il compito di caricarli, semplificare il cambio di schermata e il passaggio dei parametri fra di esse. Inoltre invoca l'evento onSwap che è utile per popolare i campi delle schermate in base, ad esempio, all'utente corrente.

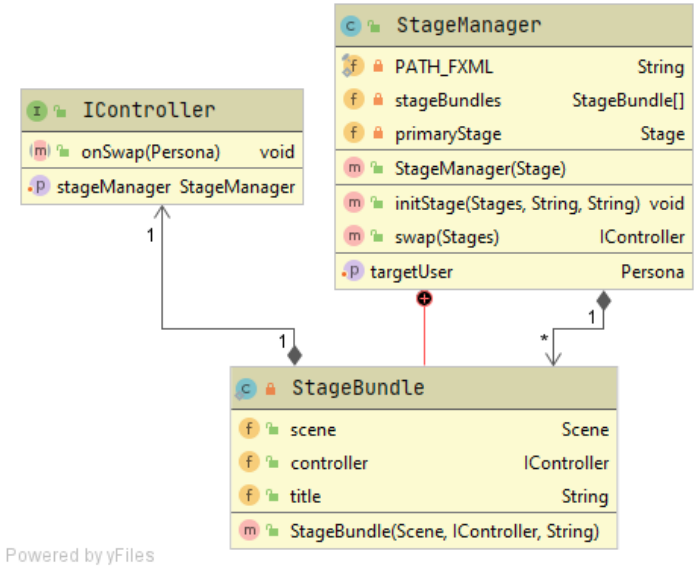

Figure 23: Diagrammi delle classi gestite dallo StageManager

6.6 Attori

In questo diagramma vengono visualizzate le relazioni fra attori e le altre classi del sistema che li supportano, come la carta fedeltà per l'utente.

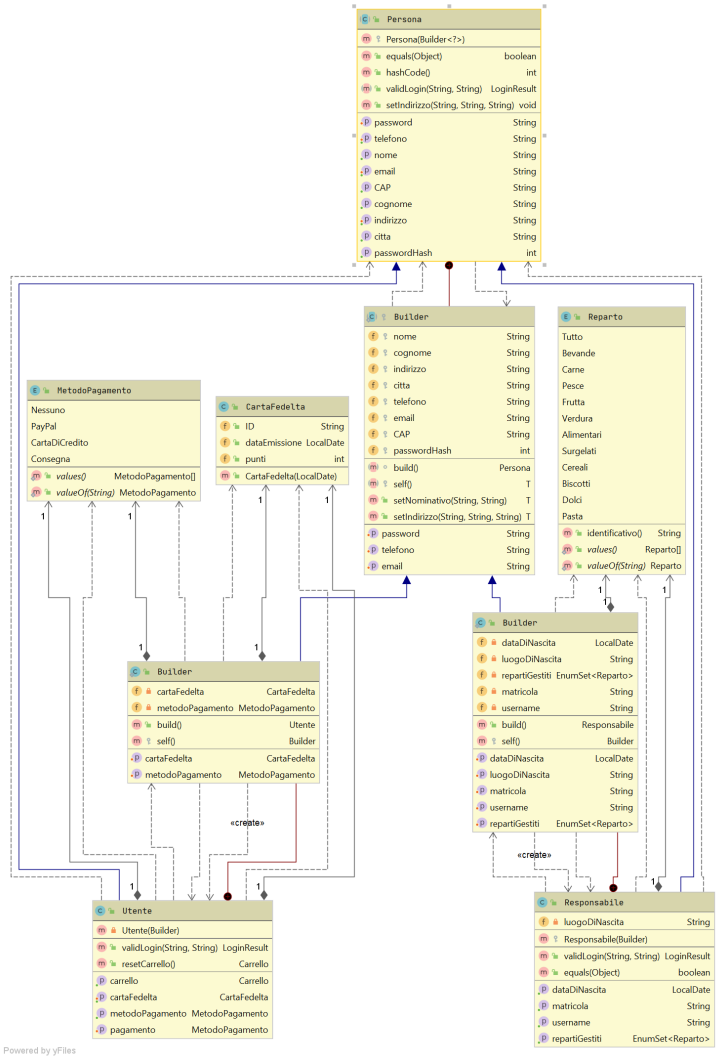

Figure 24: Diagrammi delle classi relative agli attori

6.7 Prodotto

In questo diagramma vengono visualizzate le relazione fra le classi che permettono l'acquisto. La Spesa, che deriva da un Carrello con associato l'orario e la data di consegna e un metodo di pagamento. Il Carrello, che è un insieme di prodotti con associata una quantità.

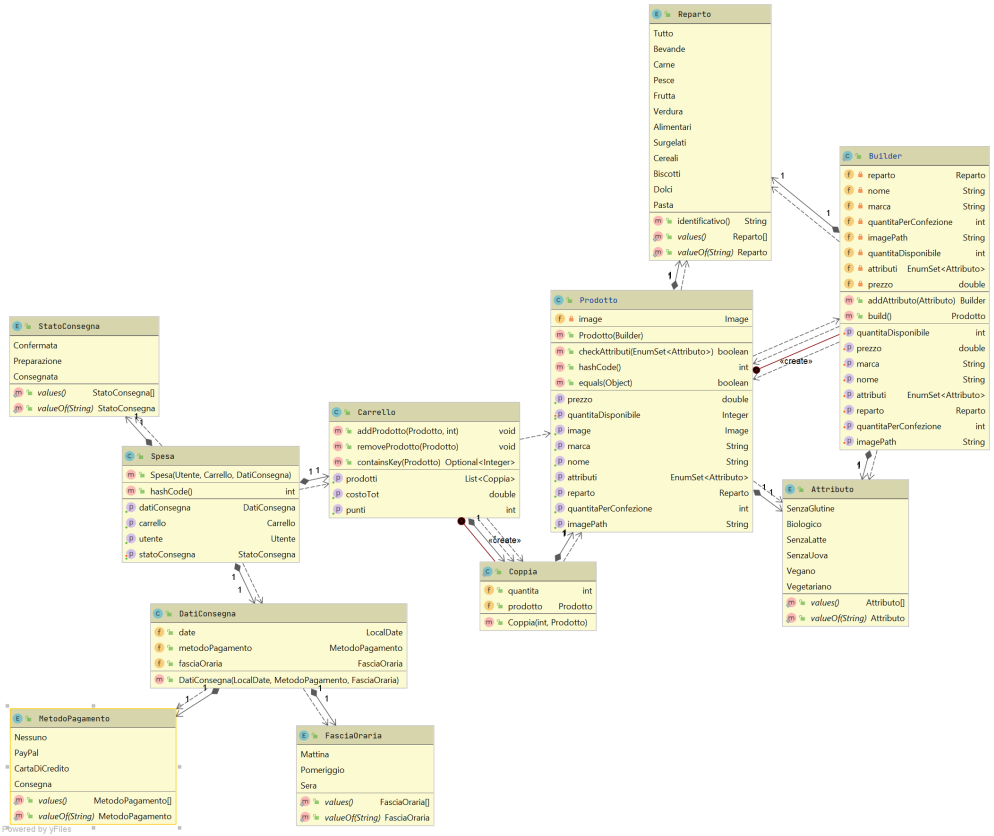

Figure 25: Diagrammi delle classi relative al commercio

7 Scelte progettuali

In questo capitolo presentiamo le scelte progettuali e i principali design pattern usati per la realizzazione del sistema informativo richiesto.

7.1 Metodologia di sviluppo

Dopo un'attenta lettura e analisi delle specifiche abbiamo suddiviso il progetto in macro aree, la priorità è stata la creazione del modello, ovvero che classi usare per descrivere i dati, come implementarle e come poterle conservare su memoria secondaria. È seguita dunque l'idea del database binario, che facilità il salvataggio e il caricamento a scapito della possibilità di modificarlo manualmente dal file. Successivamente ci siamo concentrati sull'interfaccia grafica sia per quanto riguarda i controller sia per i file stilistici fxml, abbiamo deciso di usare un singolo controller per ogni schermata, responsabile di qualsiasi azione su di essa, e di gestirli tramite uno StageManager. Questo ne facilita l'utilizzo semplificando la procedura di cambio schermata e il trasferimento di dati tra schermate.

Abbiamo deciso di sviluppare il progetto usufruendo di JavaFX, una libreria grafica basata su Java, che supportata ufficialmente cross platform. L'abbiamo preferita a Java Swing, oltre all'essere più moderna, anche per la presenza di SceneBuilder, uno strumento estremamente utile per realizzare le interfacce grafiche senza doverle creare in codice, tramite l'utilizzo di file fxml.

E' stato usato git con repository remota su gitlab per poter facilmente salvare e mantenere il progetto e per permette lo sviluppo in contemporanea di alcune sezioni di codice. Infatti le componenti più complicate e principali del progetto sono state sviluppate in duo, con un membro del gruppo che scriveva il codice mentre l'altro supportava, controllava e si accertava della correttezza di quanto veniva scritto. Mentre la correzione di bug ed eventuali modifiche secondarie per il miglioramento generale del software abbiamo suddiviso il lavoro ed ogni membro ha lavorato autonomamente.

Per la verifica dei campi, come ad esempio l'email o l'indirizzo, abbiamo usato dei Regex, ovvero delle macchine a stati finiti per controllare la correttezza dell'input. Questi sono tutti presenti nella classe statica Validator.

L'editor utilizzato è stato IntelliJ per entrambi i componenti del gruppo in quanto l'università offre una licenza premium del prodotto ma, soprattutto, si presta molto bene alla creazione della documentazione tramite plugins.

7.2 Organizzazione file progetto

I file sono così organizzati

src/main/controller	Contiene tutte le classi relative al controllo dell'interfaccia grafica
src/main/model	Contiene tutte le classi per rappresentare i dati
src/main/storage	Contiene il database e i serializer/deserializer
src/main/view	Contiene classi utilitarie per i controller, come ad esempio un ChoiceBox con autocompletamento
src/main/resources	Contiene il database.bin e il file per le città
src/main/resources/fxml	Contiene tutti i file fxml per le schermate
src/main/resources/icons	Contiene le icone delle schermate, ad esempio il logo
src/main/resources/images	Contiene le immagini dei prodotti del catalogo

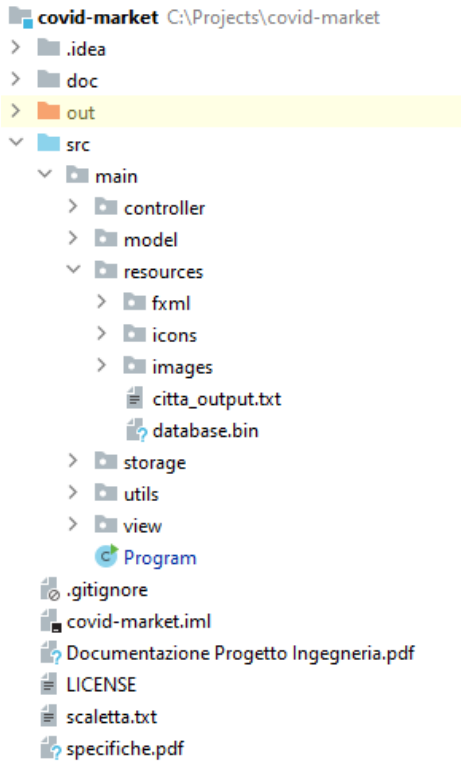

Figure 26: Organizzazione dei file nel progetto

7.3 Design Pattern utilizzati

Vengono presentati in questa sezione i principali pattern utilizzati

7.3.1 Model View Controller

Lo schema architetturale scelto per l'applicazione è quello del MVC (Model-View-Controller), è stato deciso nel momento in cui abbiamo scelto JavaFX in quanto la nostra applicazione è strutturata attorno alla libreria grafica e siamo dunque costretti ad applicare lo stesso schema.

Le tre componenti di questo pattern comunicano tra loro. La View genera eventi che vengono mandati al Controller corrispondente, il quale contiene la logica applicativa: questo decide, infatti, se interfacciarsi con il Model, per richiedere o salvare dati, oppure se comandare altre azioni alla View.

Schematicamente possiamo vedere il flusso dell'esecuzione come segue:

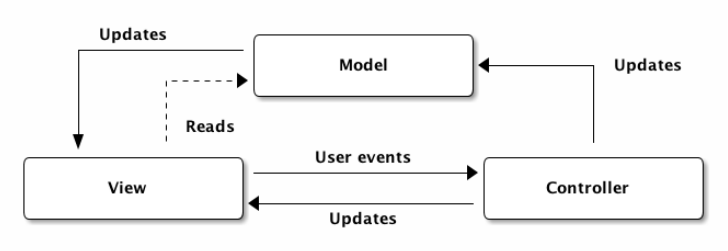

Figure 27: Schema MVC

7.3.2 Singleton

L'unico singleton della nostra applicazione è il Database, in quanto è risultato comodo garantire il suo accesso da qualsiasi controller senza doverlo inoltrare attraverso lo StageManager.

7.3.3 Observer

Abbiamo utilizzato gli observer contenuti in JavaFX per generare eventi quando. ad esempio, viene cliccato un bottone a schermo oppure si modifica il valore di un campo. Questi eventi sono intercettati dal Controller che agisce di conseguenza.

7.3.4 Builder

Il builder pattern rende più chiara alla vista la creazione di un oggetto rispetto ad un classico costruttore, soprattutto se il numero di argomenti è molto elevato, abbiamo dunque deciso di implementarlo per la maggior parte delle classi del modello: Prodotto, Persona, Utente, Responsabile.

Risulta interessante che i Builder di Utente e Responsabile estendono il Builder di Persona, questo non è stato semplice da implementare.

7.3.5 Iterator

Classico pattern utilizzato per qualsiasi iterazione di collezioni, abbiamo usato gli Iterator base presenti nelle librerie Java.

7.4 Test

Abbiamo testato l'applicazione durante tutto il suo sviluppo con semplici test ad hoc sulle nuove componenti appena realizzate, successivamente l'abbiamo affidata anche a terze parti (sia nostri colleghi che famigliari), raccogliendo la loro esperienza riguardo a bug, inconsistenze o incompletezze relative alla specifica.

Abbiamo entrambi controllato il codice scritto dall'altro ed eseguito costantemente simulazione ad ogni modifica rilevante.

Non abbiamo utilizzato test JUnit.