

Progetto F2A Laundry Service

Fabio Colombo

Andre Giovanni Padovano

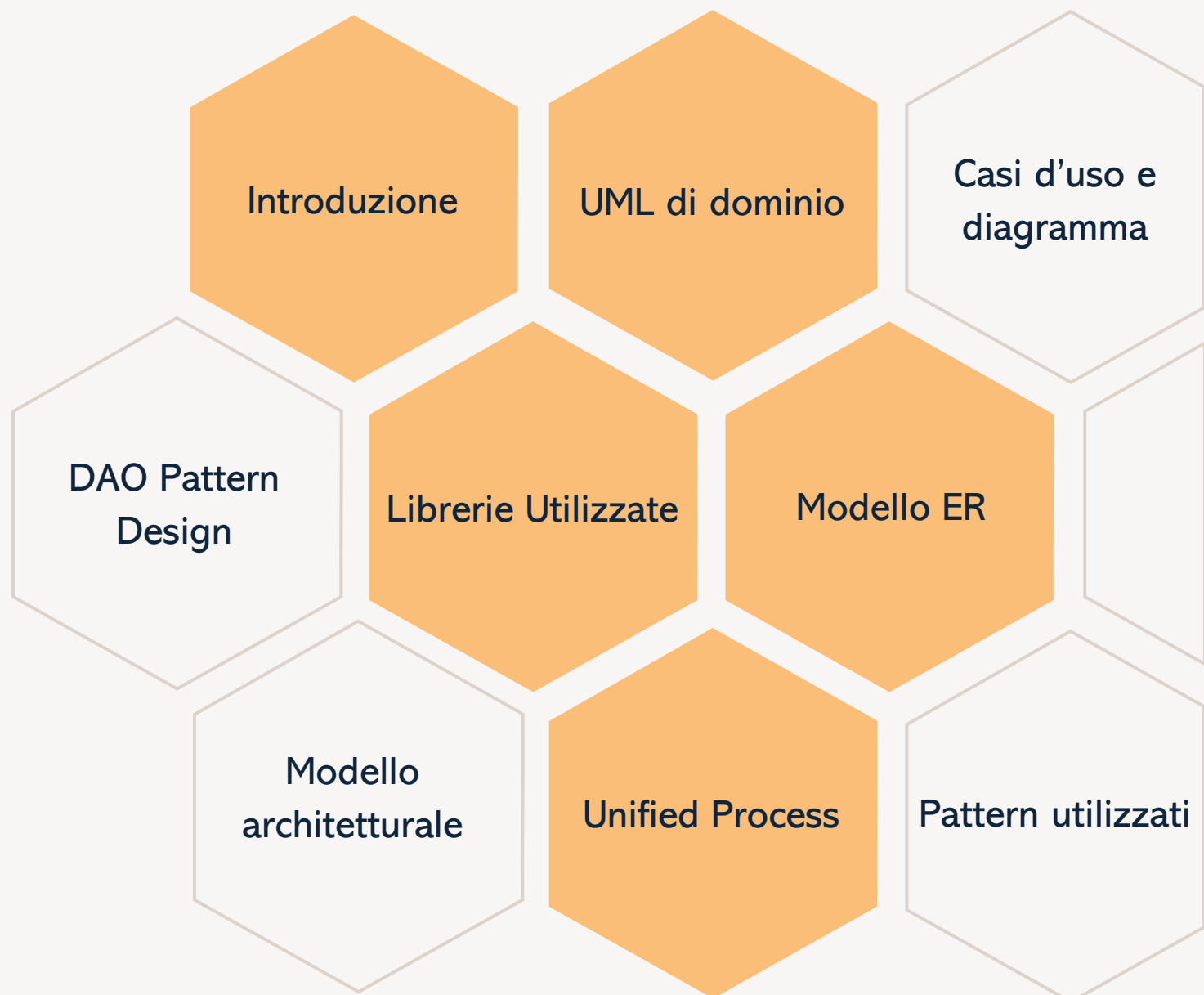
Andrea Daglio

F2A
LAUNDRY
SERVICE





Agenda



Introduzione

Cos'è F2A Laundry Service?

F2A Laundry Service è una soluzione innovativa nata per trasformare la vecchia esperienza della classica lavanderia a gettoni in qualcosa di più comodo e user-friendly

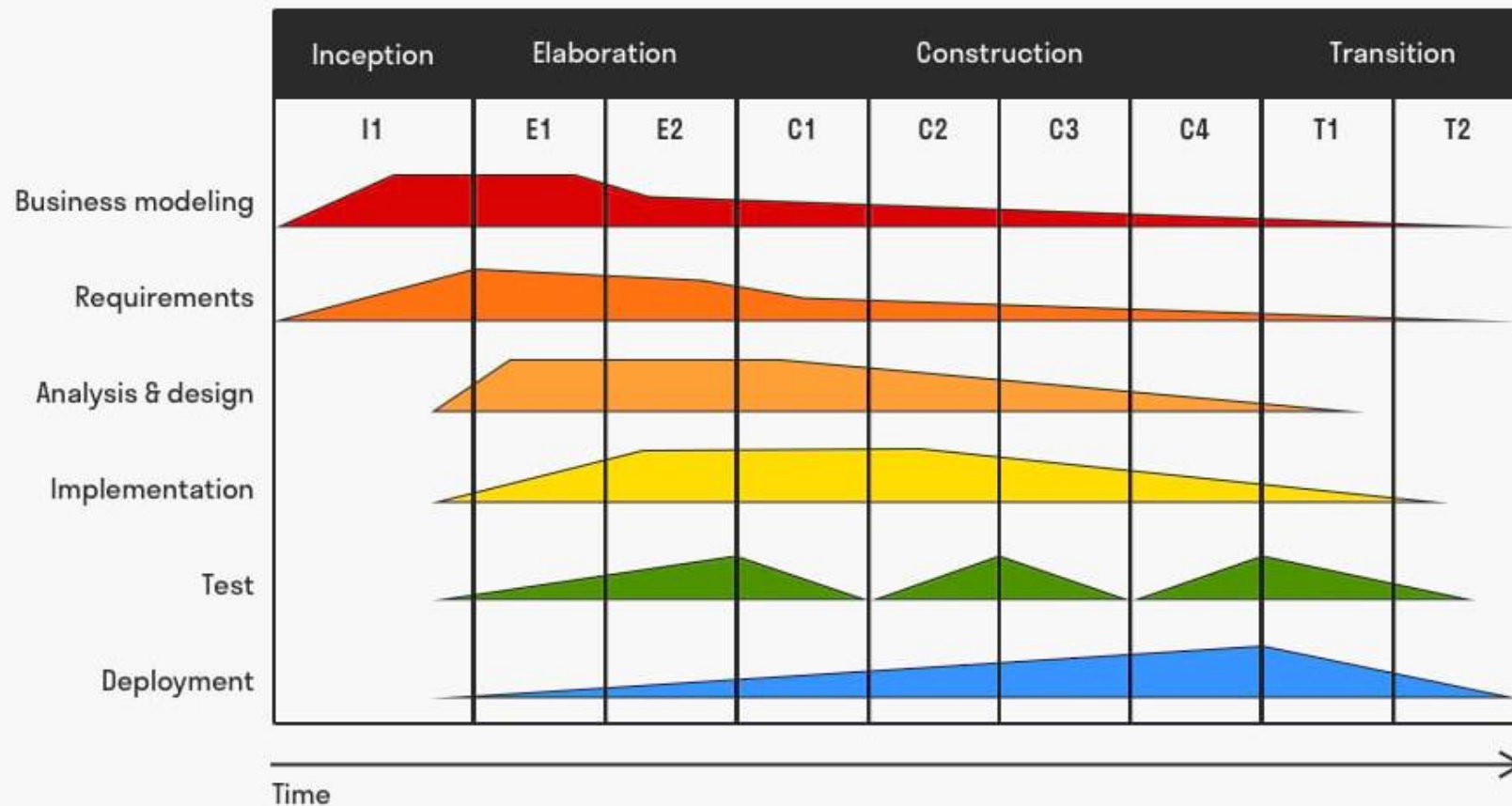
Idea e obiettivi

La piattaforma di F2A è stata pensata per garantire ai futuri clienti un servizio presso cui prenotare, depositare e ritirare i propri capi.

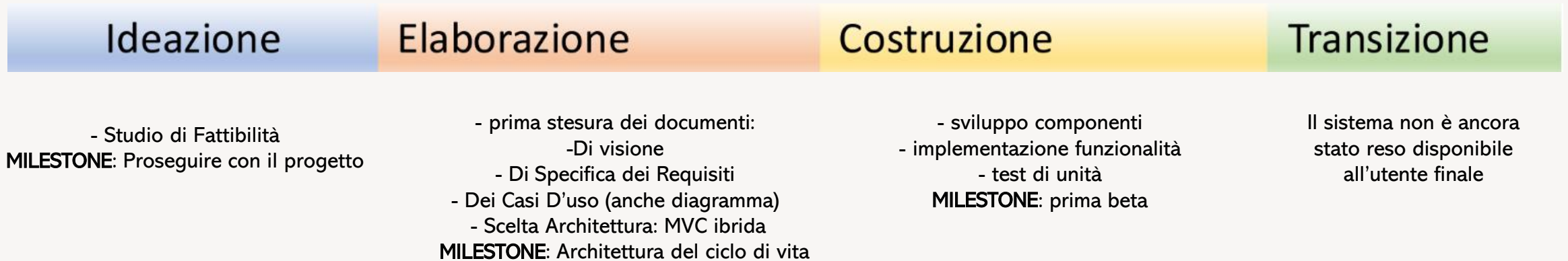


Progettazione: Unified Process

Abbiamo sfruttato il modello di progettazione UP avendo la necessità di affrontare le fasi di ciclo di vita del prodotto finale in modo iterativo e incrementale



Progettazione: Unified Process



Linguaggi e librerie utilizzate

Java SE 22 : per la stesura del codice

Javax.swing e javax.awt: per la creazione dell'interfaccia grafica

JUnit 4.1 : per fase di testing del sw

mysql-connector-j 8.0.19: per garantire la persistenza dei dati attraverso una connessione locale a database

Casi d'uso



Registrazione/Login

Registrazione/Login
presso un totem
all'interno di un negozio
F2AService



Deposito e ritiro capi

Fase di prelevamento di
un capo e redazione
prenotazione per la
lavorazione dello stesso
+ ritiro da parte del
cliente del capo presso
un negozio



Lavorazione Capi

Capo attraversa
diverse fasi di
lavorazione
all'interno della
sede centrale



Manutenzione macchinari

Manutenzione in
seguito a guasti
imprevisti sulle
stazioni



Consultazione + svolgimento ticket

Attività di
ritiro/consegna dei capi
prima e dopo la
lavorazione degli stessi

Diagramma dei casi d'uso

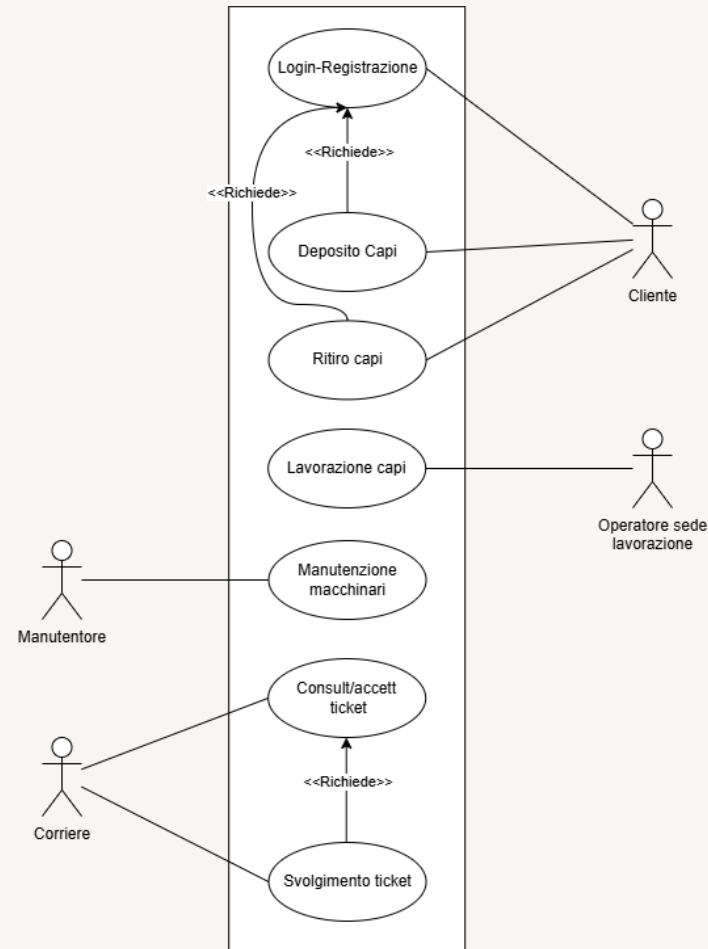


Diagramma di dominio

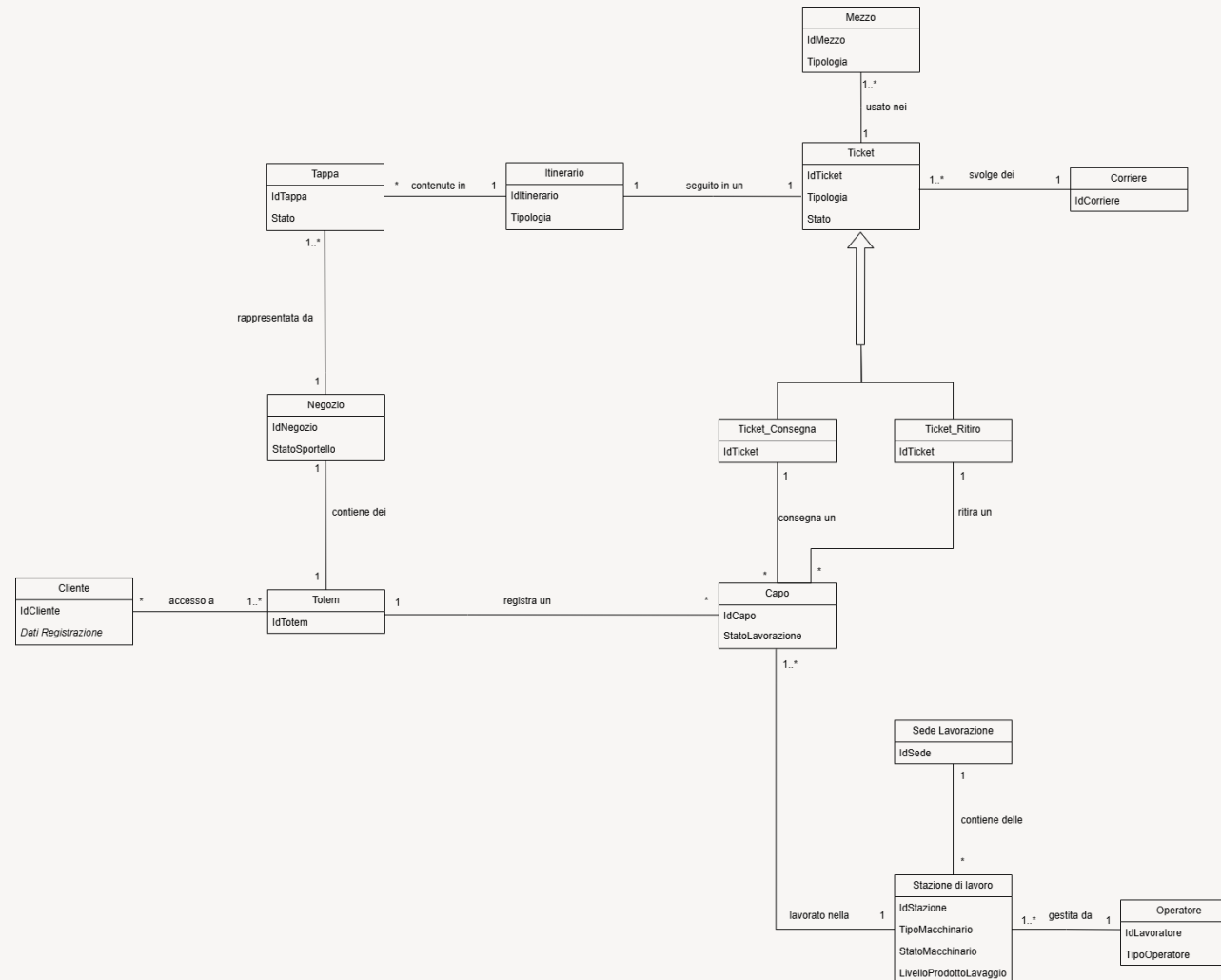
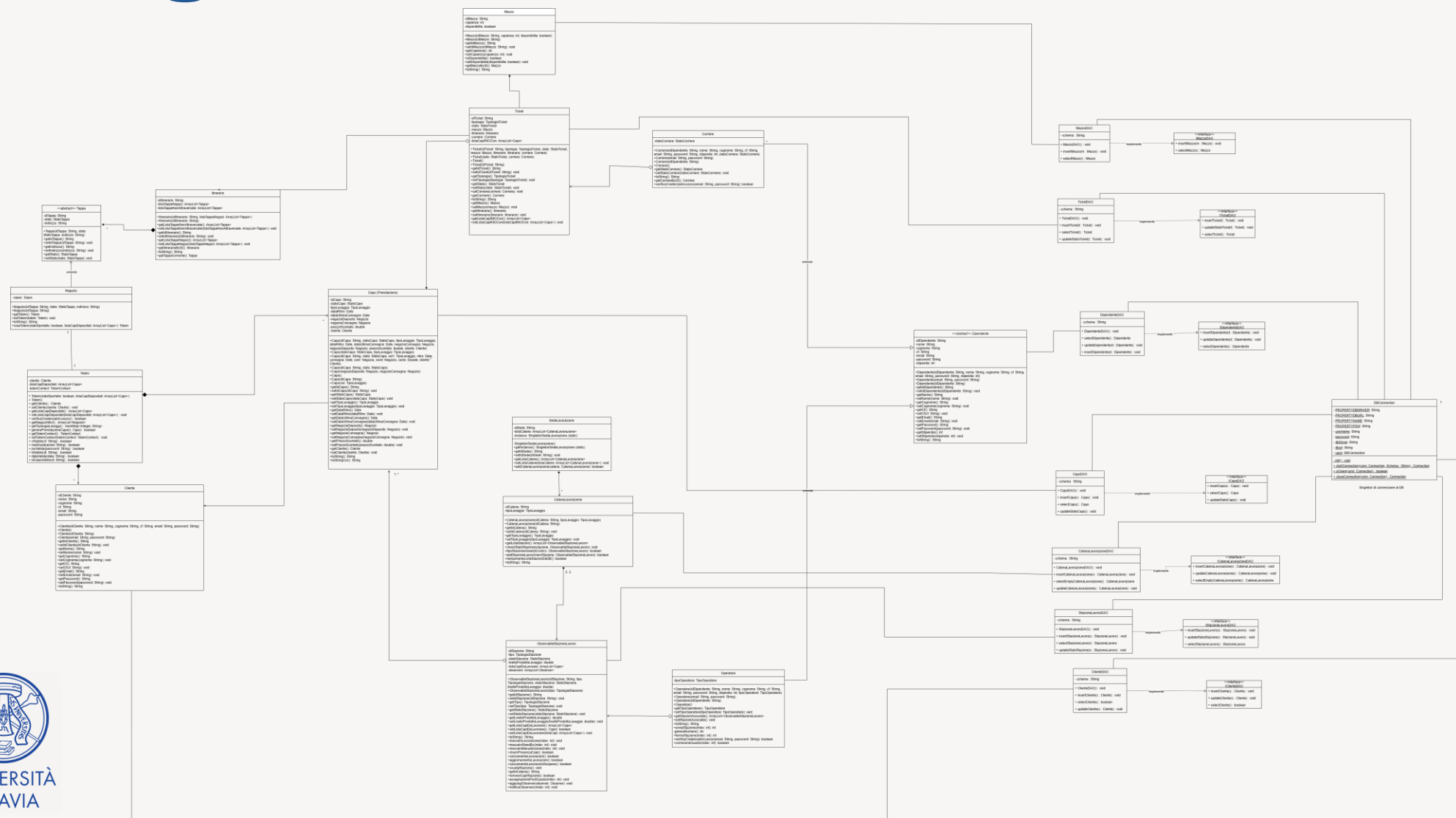


Diagramma delle classi



Diagrammi di sequenza

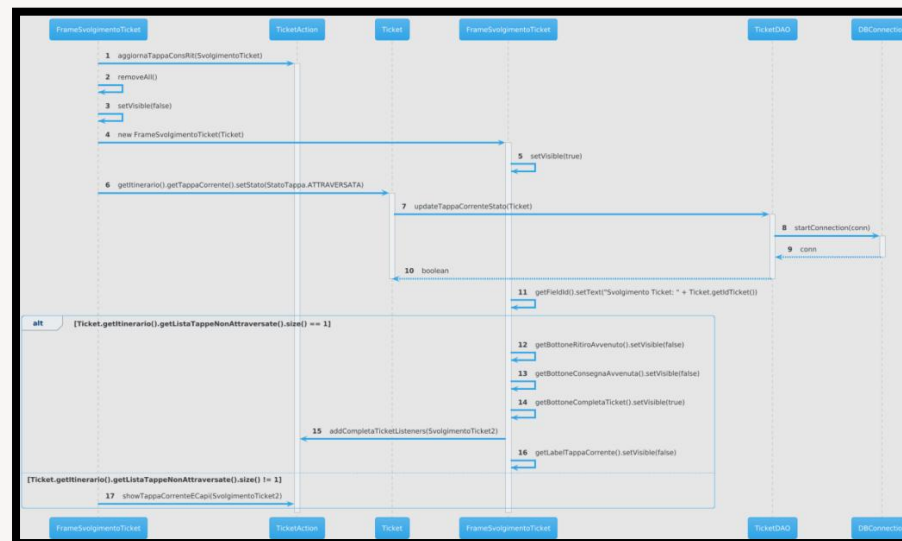
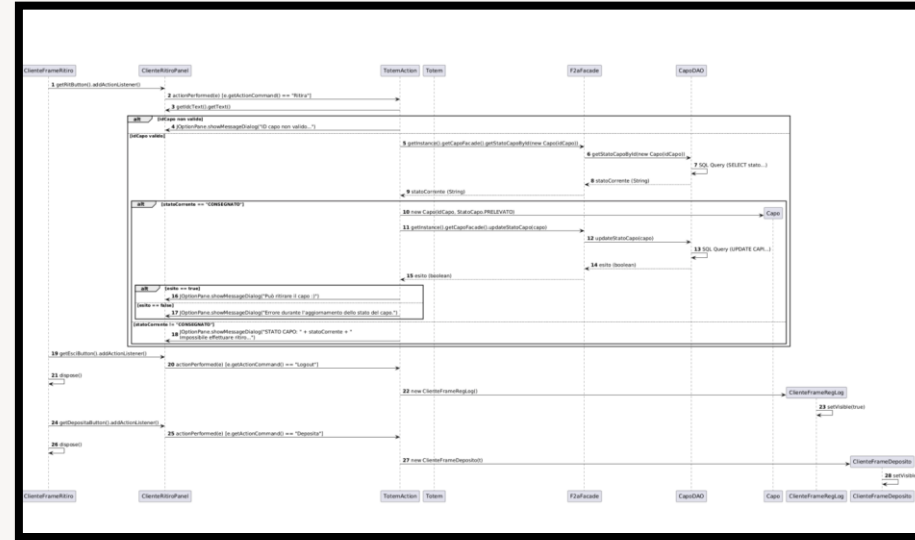
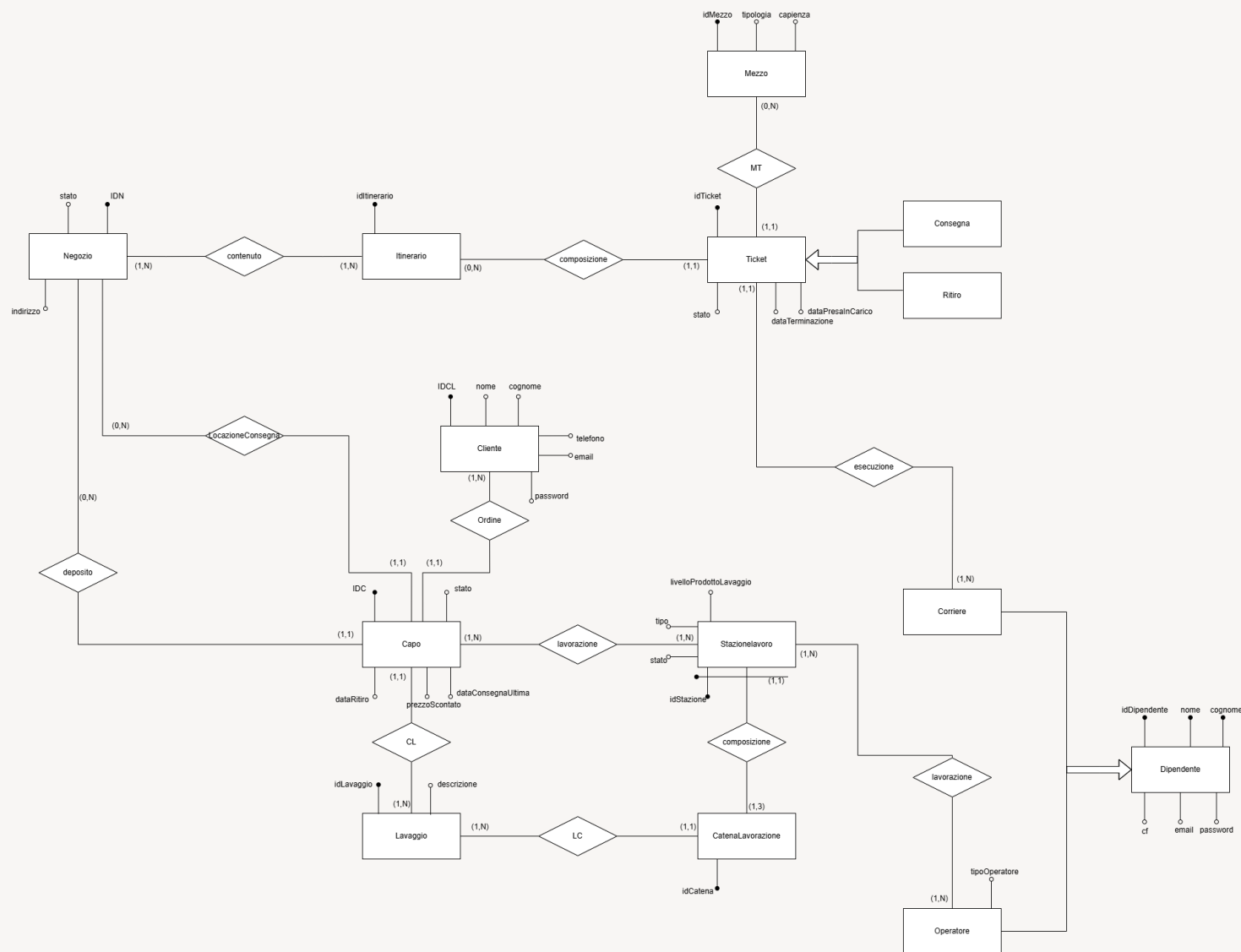


Diagramma ERA



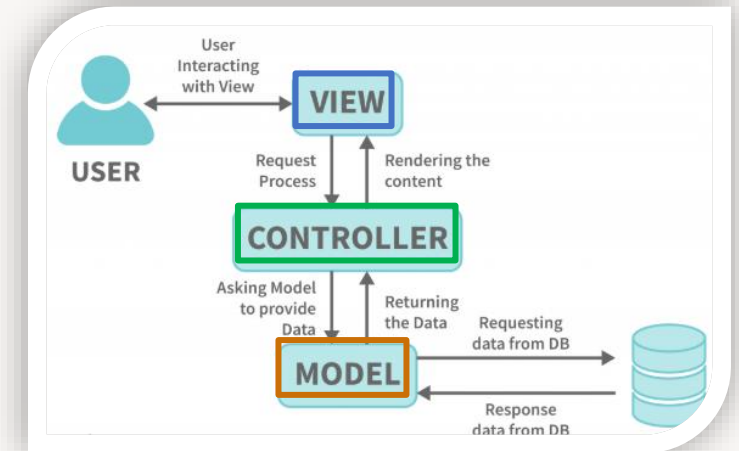
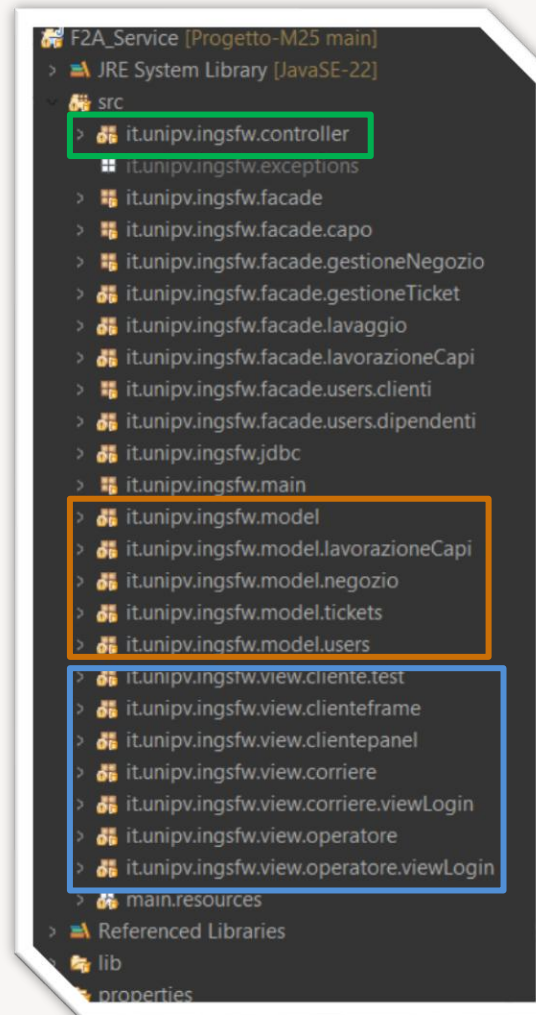
Architettura MVC

Separazione efficiente tra:

- Logica di modello
- View
- Controller

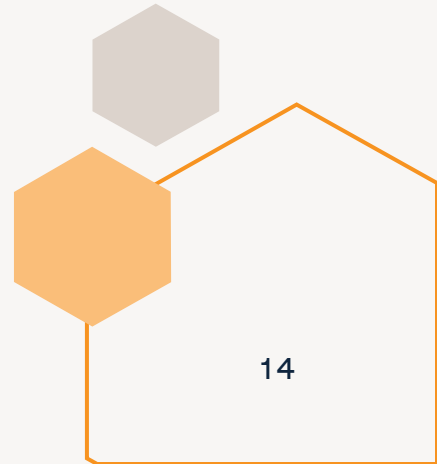
Vantaggi:

- Modularità;
- Scalabilità
- Protected Variation



Pattern GoF utilizzati

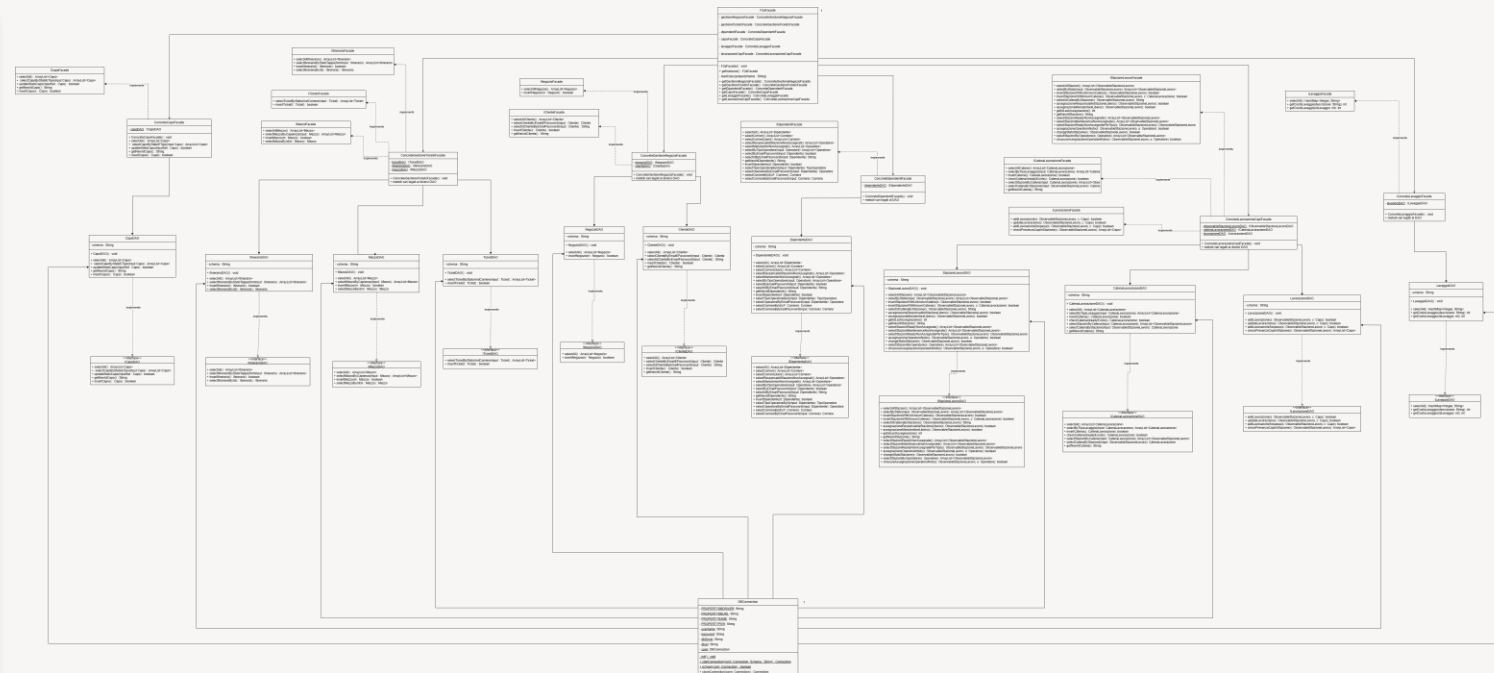
- **FACADE** : usato per proteggere gli elementi di dominio(stabili) da quelli più variabili nel tempo come le classi DAO e le view.
- **SINGLETON** : usato per rappresentare l'esistenza di un'unica sede di lavorazione, e come scelta progettuale per F2aFacade;
- **OBSERVER** : usato per notificare il cambio di stato delle stazioni nel modello alle view attraverso il controller
- **STRATEGY** : abbiamo usato i pattern Strategy per favorire una maggiore estendibilità del codice nell'eventualità si voglia cambiare/aggiungere tipi di scontistica in fase di prenotazione di una qualche lavorazione di un capo.
- **FACTORY** : utilizzato per costruzione della strategy per la scontistica e per la costruzione delle classi DAO.



Pattern FACADE

Il pattern FACADE si propone di risolvere il problema dell'accoppiamento tra singoli elementi di un sotto-sistema complesso mediante la definizione di un oggetto facciata.

In questo caso F2aFacade si pone come intermediario tra i DAO (che si affacciano sul livello di persistenza) e i vari oggetti, così facendo si nasconde a questi ultimi una logica potenzialmente complessa e variabile.



```
src
> it.unipv.ingsfw.controller
> it.unipv.ingsfw.facade
> DaoFactory.java
> F2aFacade.java
it.unipv.ingsfw.facade.capo
> ConcreteCapoFacade.java
> ICapoFacade.java
it.unipv.ingsfw.facade.gestioneNegozio
> ConcreteGestioneNegozioFacade.java
> IClientiFacade.java
> INegozioFacade.java
it.unipv.ingsfw.facade.gestioneTicket
> ConcreteGestioneTicketFacade.java
> ITinerarioFacade.java
> IMezzoFacade.java
> ITicketsFacade.java
it.unipv.ingsfw.facade.lavaggio
> ConcreteLavaggioFacade.java
> ILavaggioFacade.java
it.unipv.ingsfw.facade.lavorazioneCapi
> ConcreteLavorazioneCapiFacade.java
> ICatenaLavorazioneFacade.java
> ILavorazioneFacade.java
> IStazioneLavoroFacade.java
it.unipv.ingsfw.facade.users.clienti
> ConcreteClientiFacade.java
> IClientiFacade.java
it.unipv.ingsfw.facade.users.dipendenti
> ConcreteDipendentiFacade.java
> IDipendentiFacade.java
```



DAO Design Pattern

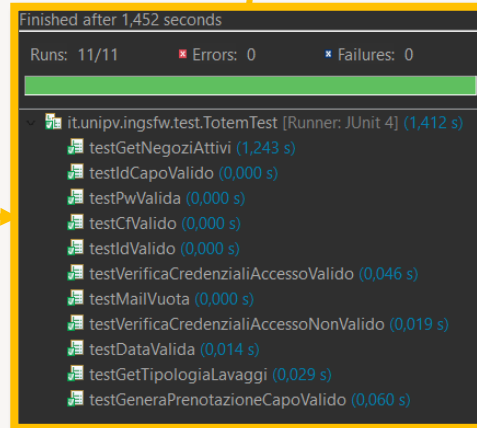
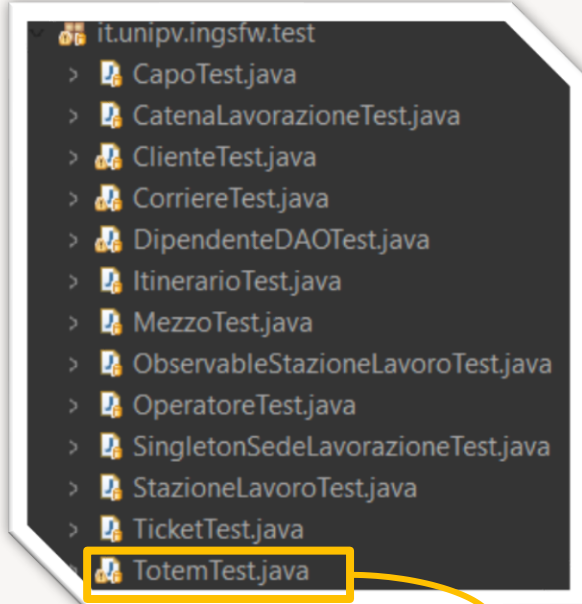
Il pattern DAO prevede un oggetto di dominio che esponga le informazioni da rendere persistenti e un interfaccia che definisca alcuni metodi, quest'ultima verrà poi implementata dalle varie classi che connettono il dominio con lo specifico database.

```
it.unipv.ingsfw.model
> Capo.java
> CapoDAO.java
> ICapoDAO.java
> ILavaggioDAO.java
> LavaggioDAO.java
> StatoCapo.java
> TipoLavaggio.java
it.unipv.ingsfw.model.lavorazioneCapi
> CatenaLavorazione.java
> CatenaLavorazioneDAO.java
> ICatenaLavorazioneDAO.java
> ILavorazioneDAO.java
> IObservableStazioneLavoroDAO.java
> LavorazioneDAO.java
> ObservableStazioneLavoro.java
> ObservableStazioneLavoroDAO.java
> SingletonSedeLavorazione.java
> StatoStazione.java
> TipologiaStazione.java
it.unipv.ingsfw.model.negozio
> INegozioDAO.java
> Negozio.java
> NegozioDAO.java
> StatoTappa.java
> Tappa.java
> Totem.java
```

```
it.unipv.ingsfw.model.tickets
> ItinerarioDAO.java
> IMezzoDAO.java
> ITicketDAO.java
> Itinerario.java
> ItinerarioDAO.java
> Mezzo.java
> MezzoDAO.java
> StatoTicket.java
> Ticket.java
> TicketDAO.java
> TipologiaTicket.java
it.unipv.ingsfw.model.users
> Cliente.java
> ClienteDAO.java
> Corriere.java
> Dipendente.java
> DipendenteDAO.java
> IClienteDAO.java
> IDipendenteDAO.java
> Operatore.java
> StatoCorriere.java
> TipoOperatore.java
```

Testing

Utilizzando la libreria Junit 4, abbiamo creato dei test sulle classi sw di dominio per assicurarci che i metodi fondamentali del programma funzionino come previsto in risposta alle azioni principali che verranno intraprese nel corso dell'esecuzione del programma stesso.



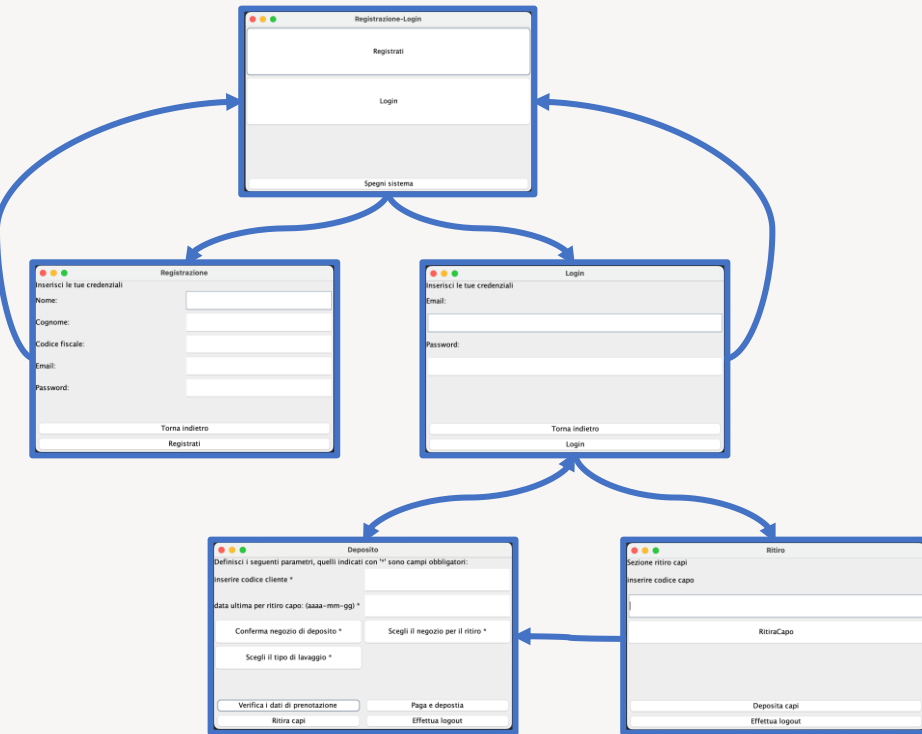
Part. di Equivalenza	Descrizione				
Partizione	Descrizione				
cf valido	String di 16 caratteri alfanumerici				
cf non valido	String vuota, null o di lunghezza diversa da 16				
mail vuota	String vuota o null				
mail non vuota	String con contenuto				
pw valida	String di almeno 8 caratteri				
pw non valida	String vuota, null o di lunghezza inferiore a 8				
id valido	String di 5 caratteri alfanumerici, con le prime due lettere e le ultime tre numeri				
id non valido	String vuota, null o formato errato				
data valida	String nel formato "yyyy-MM-dd" e almeno tre giorni nel futuro				
data non valida	String vuota, null o formato errato o data nel passato				
Part. di non correttezza	Descrizione				
input non corretto	Descrizione				
cf vuoto o null	String vuota o null				
cf lunghezza errata	String con lunghezza diversa da 16				
cf caratteri non alfanumerici	String con caratteri diversi da lettere e numeri				
mail non vuota valida	String con formato email valido				
mail non vuota non valida	String con formato email non valido				
pw vuota o null	String vuota o null				
pw lunghezza inferiore a 8	String con meno di 8 caratteri				
id vuoto o null	String vuota o null				
id lunghezza errata	String con lunghezza diversa da 5				
id formato errato	String con formato diverso da due lettere seguite da tre numeri				
data vuota o null	String vuota o null				
data formato errato	String con formato diverso da "yyyy-MM-dd"				
data nel passato	String che rappresenta una data nel passato				
data non esistente	String che rappresenta una data non esistente (es. 2024-02-30)				
idCapo vuoto o null	String vuota o null				
idCapo lunghezza errata	String con lunghezza diversa da 4				
idCapo formato errato	String con formato diverso da una lettera seguita da tre numeri				

Off by 1

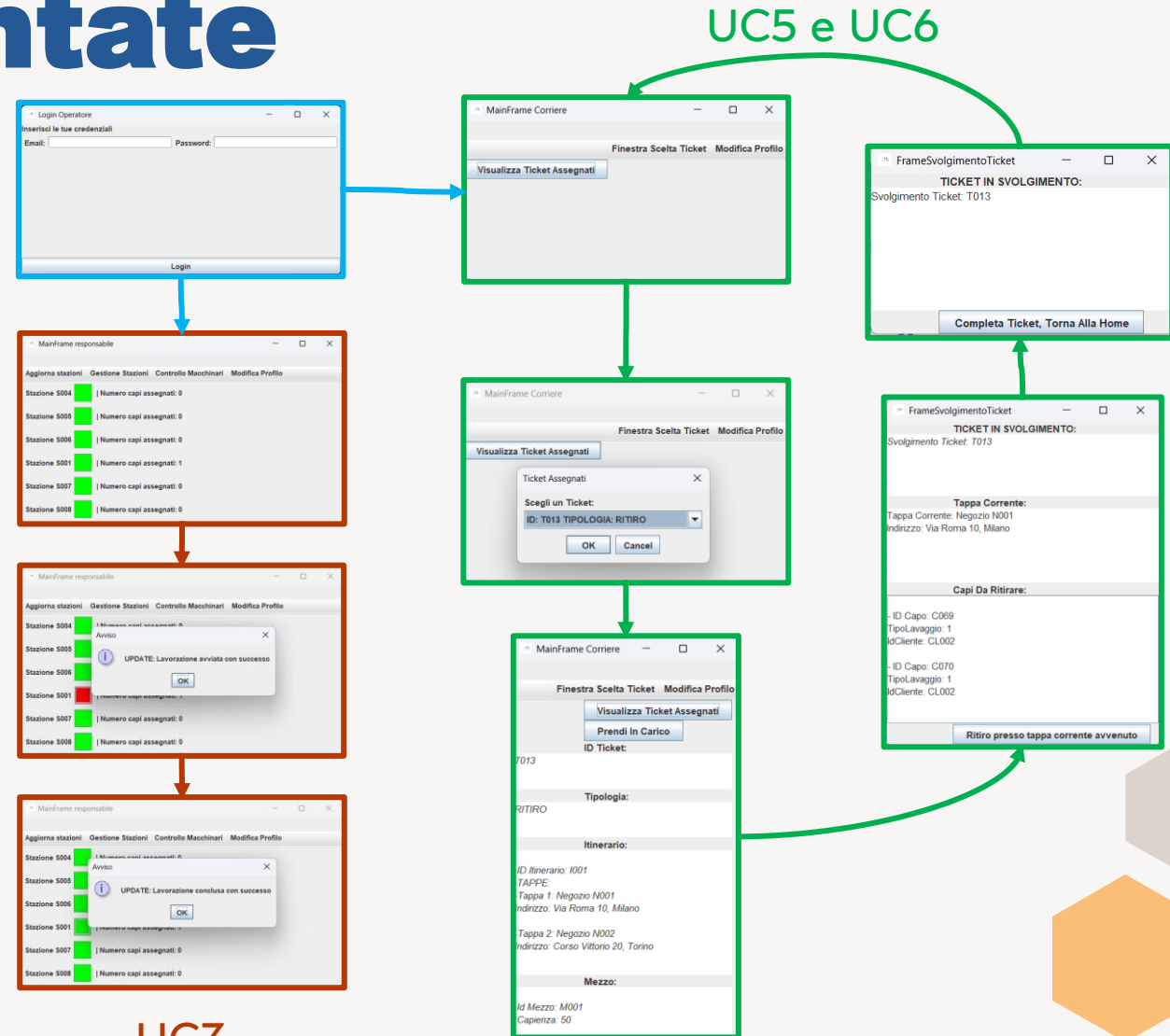
Caso di test off-by-1	Descrizione
cf lunghezza 15	String con 15 caratteri
cf lunghezza 17	String con 17 caratteri
pw lunghezza 7	String con 7 caratteri
pw lunghezza 9	String con 9 caratteri
id lunghezza 4	String con 4 caratteri
id lunghezza 6	String con 6 caratteri
data un giorno nel futuro	String con data un giorno nel futuro
data due giorni nel futuro	String con data due giorni nel futuro
data quattro giorni nel futuro	String con data quattro giorni nel futuro
idCapo lunghezza 3	String con 3 caratteri
idCapo lunghezza 5	String con 5 caratteri

GUI implementate

UC1 e UC2



UC3





Grazie

Fabio Colombo

Andrea Giovanni Padovano

Andrea Daglio