

Museo 2.0

Giuseppe Parrotta

Elaborato per il corso di
Ingegneria del Software



UNIVERSITÀ
DEGLI STUDI
FIRENZE

Dipartimento di Ingegneria Informatica
Università degli Studi di Firenze
07/04/2021



1 Introduzione

L'elaborato consiste nella realizzazione, tramite linguaggio java, di un modello di dominio auto-assegnato.

L'idea è quella di realizzare un software che permetta all'amministratore di un museo di creare delle mostre di opere d'arte, allestendo alcune delle tante sale a disposizione del museo, e gestendo la vendita dei ticket di ingresso.

Questa idea esplode in vari sotto-casi, uno tra questi è permettere alle mostre di essere svolte anche online e di potere essere fruibili a una sola parte di visitatori, cioè quelli che sono registrati.

Un altro sotto-caso è quello di permettere a un museo di realizzare mostre anche usando opere di altri musei, prendendole in affitto dopo aver pagato una certa somma di denaro.

Infine, ho pensato che la creazione di una Mostra possa essere anche fatta in modo automatico, prevedendo che l'amministratore possa essere automatico del sistema e quindi che elabori strategie in base allo stato del museo.

Per la progettazione del software sono stati impiegati i seguenti strumenti: Class Diagrams, Sequence diagrams, Use Case Diagram e Activity diagrams, Mockups. Invece, per quanto riguarda l'implementazione, sono stati fondamentali i design patterns Observer, Strategy e Singleton.

Per testare il codice ho usato la suite JUnit.

Part I

Progettazione

2 Definizione del problema

Una descrizione più dettagliata:

Si vuole realizzare una applicazione che permetta all'Amministratore di un Museo di stanziare delle somme di denaro, per poter organizzare una Mostra di Opere d'Arte.

Volendo, l'Amministratore che sta preparando una Mostra può decidere di noleggiare alcune delle Opere d'arte da altri musei, ammesso che quelle opere non risultino già affittate in qualche altra Mostra. Una volta che le opere sono state scelte ed eventualmente noleggiate, chiederà a un Impiegato Organizzatore di organizzare la Mostra. La Mostra può essere Fisica, quindi fatta in alcune Sale del Museo, o Virtuale e quindi fruibile ANCHE dal portale del Museo. Il numero di posti in questo caso aumenta, avendo a disposizione un certo numero di posti virtuali, proporzionali alla potenza dei server del Museo. L'Organizzatore avrà il compito di selezionare le Sale necessarie per creare la Mostra, e di selezionare un equipe di Impiegati che dovranno svolgere i seguenti compiti: pulizia delle Sale Fisiche, spostare l'Opera d'Arte, creare Pubblicità e nel caso in cui la Mostra sia anche Virtuale, anche l'allestimento del Portale Web.

La Mostra deve essere dichiarata conclusa se richiesto dall'Amministratore, oppure quando raggiunge la quota di incassi prevista dall'Organizzatore. Anche l'Amministratore può chiedere la chiusura di una Mostra: in questo caso, l'Organizzatore farà uno storno dei soldi dei Ticket venduti per quella Mostra meno una trattenuta in percentuale. Alla chiusura delle Mostre, le eventuali opere noleggiate andranno restituite al legittimo proprietario. L'Amministratore chiede la chiusura di una Mostra quando vede che le sale sono quasi tutte occupate.

Gli eventuali Visitatori delle Mostre possono essere di due tipi: Visitatori o Visitatori Registrati. I Visitatori (non Registrati) possono visitare il Museo, oppure possono partecipare a Mostre che quel Museo ha allestito solo se sono rimasti posti dentro le Sale Fisiche dedicate a quella Mostra. I Visitatori Registrati hanno tutte queste possibilità con in più la possibilità di Occupare i posti previsti dalle Sale Virtuali. Inoltre, ogni Visitatore può registrarsi al Museo, diventando un Visitatore Registrato.

Per Impiegati, Organizzatori o Visitatori, è possibile consultare il catalogo delle Opere e lasciare suggerimenti su quali opere portare per la prossima Mostra.

3 Manufatti

3.1 Class Diagram concettuale del Modello di Dominio

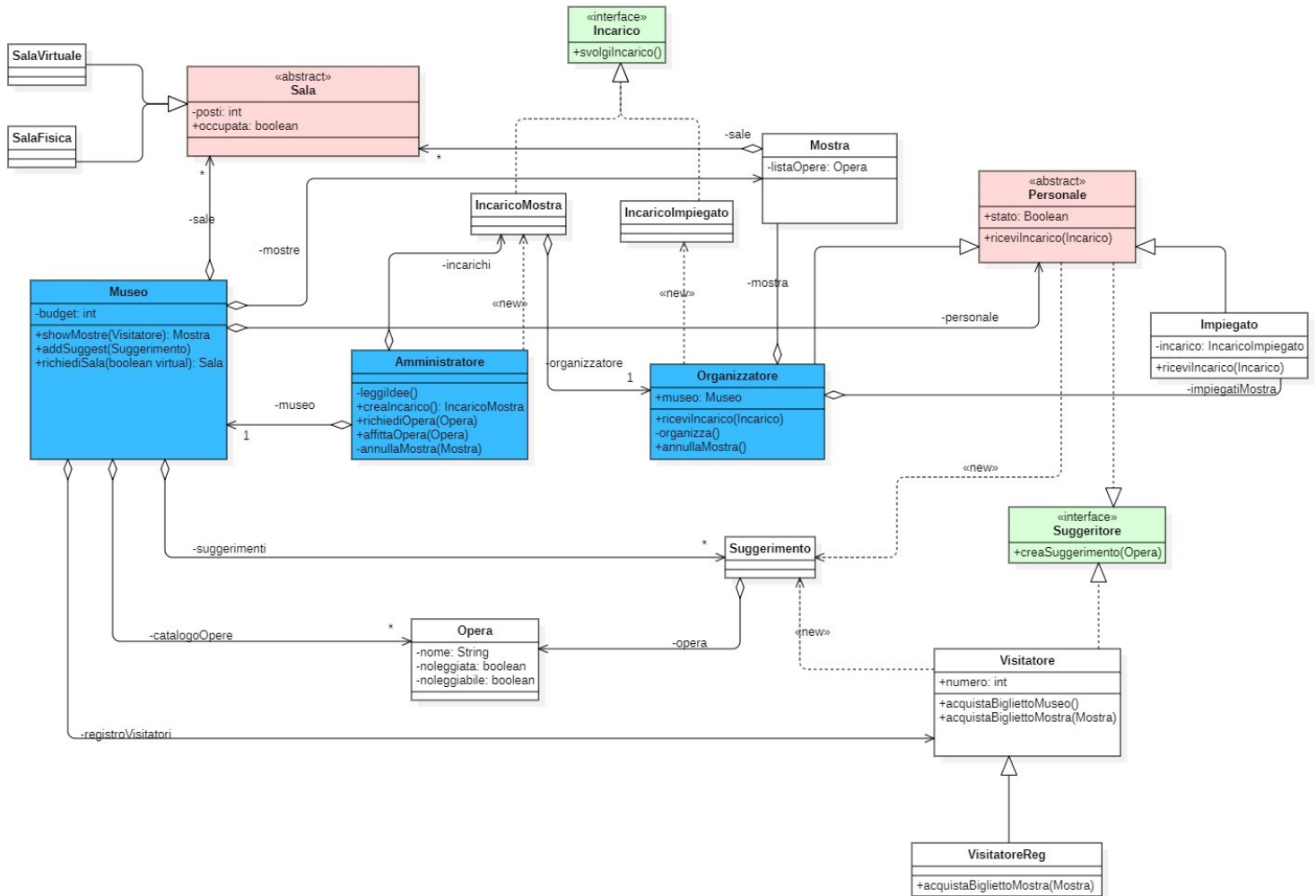


Figure 1: Modello di dominio

3.2 Package Diagram

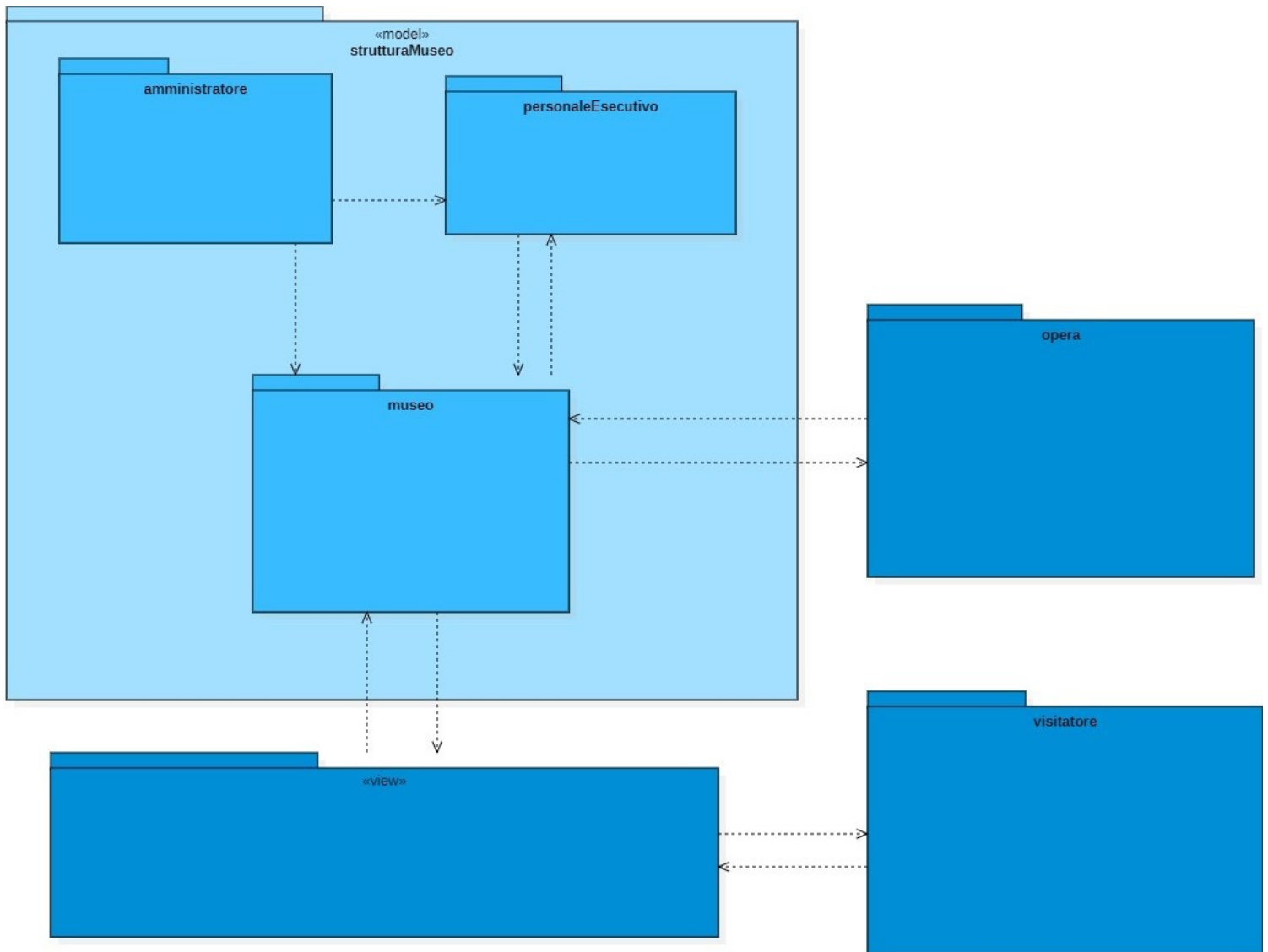


Figure 2: Package diagram

3.3 Use Case Diagram

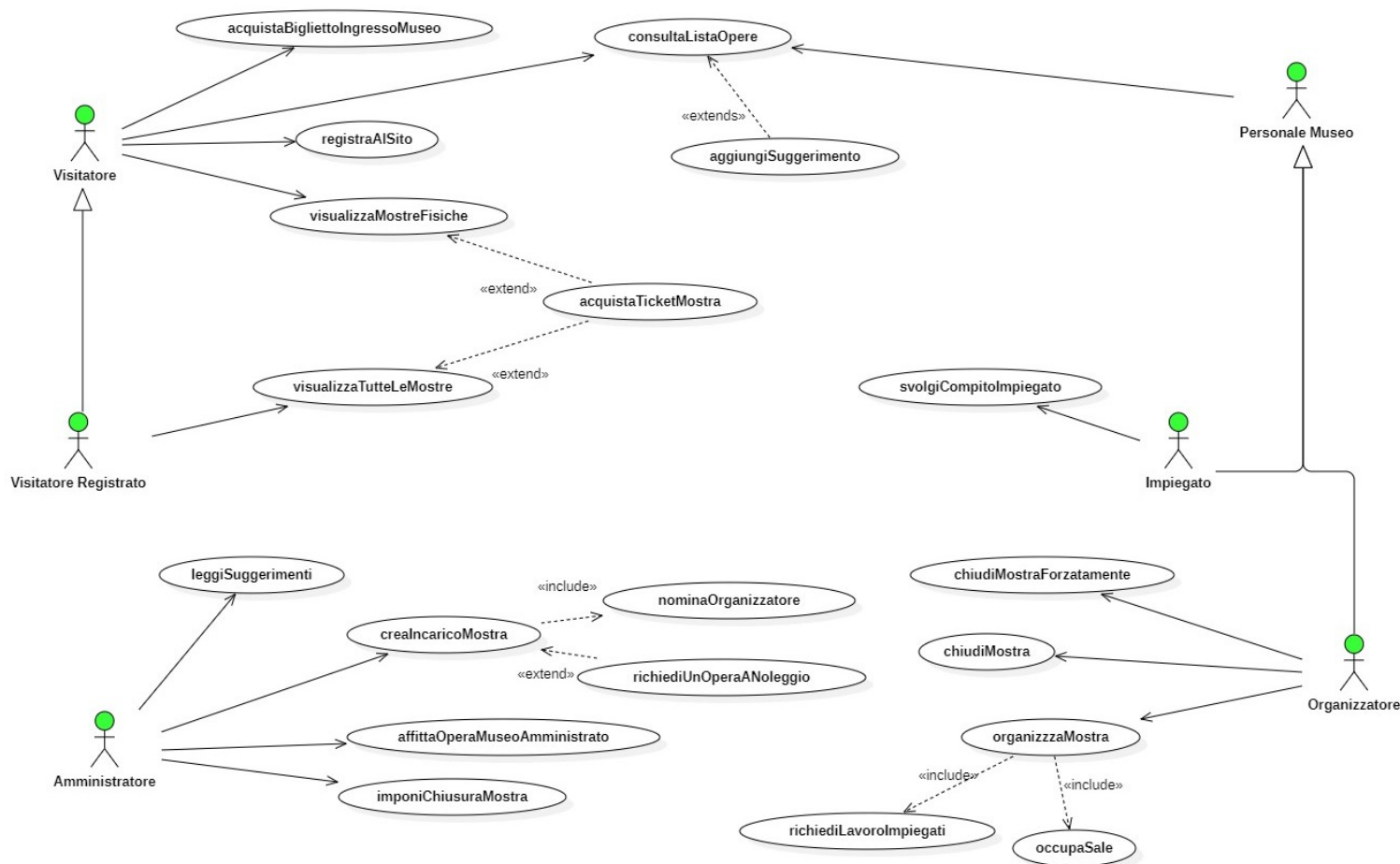


Figure 3: Use Case Diagram

Questi sono i casi d'uso identificati. Ho ritenuto di particolare interesse una più specifica descrizione dei casi d'uso più particolari, sfruttando gli Use Case Templates riportati nella sezione successiva.

3.4 Use Case Templates

<u>Name</u>	UCAdmin#2 - Crea incarico Mostra
<u>History</u>	
<u>Source</u>	
<u>Level</u>	Users goal
<u>Description</u>	L'amministratore del Museo crea un incarico per una mostra
<u>Scope</u>	Creazione Mostra
<u>Actors</u>	Amministratore
<u>Pre - Conditions</u>	L'utente ha fatto l'accesso al sistema come amministratore
<u>Post- Conditions</u>	Viene emesso un incarico e fornito a un Organizzatore
<u>Normal flow</u>	0 L'utente accede al pannello amministratore 1 L'utente legge nel pannello la quantità di denaro disponibile 2 Extends: l'utente può scegliere se consultare i suggerimenti 3 L'utente sceglie di avviare una creazione 4 Il sistema mostra il pannello di creazione 5 L'utente seleziona opere e organizzatore e conferma 6 Extends: Il sistema chiede conferma per il noleggio di opere 7 Il sistema manda notifica all'organizzatore nominato
<u>Alternative flows</u>	6b L'utente cambia opere
<u>References</u>	UCAdmin#3 - Nomina Organizzatore
<u>Non functional requirements</u>	

<u>Name</u>	UCOrganiz#3 - Organizza Mostra
<u>History</u>	
<u>Source</u>	
<u>Level</u>	Users goal
<u>Description</u>	L'organizzatore sviluppa l'incarico ricevuto
<u>Scope</u>	Creazione Mostra
<u>Actors</u>	Organizzatore (primary), Museo(secondary)
<u>Pre - Conditions</u>	L'amministratore ha creato l'incarico
<u>Post- Conditions</u>	Il Museo dispone di una mostra in più
<u>Normal flow</u>	0 L'organizzatore accede al pannello organizzatore 1 Il sistema notifica l'incarico 2 L'utente legge la notifica 3 L'utente apre il pannello per la creazione 4 Il sistema mostra quali sono le sale disponibili del museo 5 Il sistema mostra quali sono gli impiegati disponibili 6 L'utente seleziona impiegati e imposta loro un ruolo 7 L'utente sceglie le sale del museo 8 Il sistema chiede conferma 9 Il sistema crea la mostra
<u>Alternative flows</u>	8b L'utente annulla 9b Torna al punto 4
<u>References</u>	UCOrganizz#4 -Richiedi Impiegati UCOrganizz#5 - Occupa Sale
<u>Non functional requirements</u>	

Figure 4: Use Case Templates di due casi d'uso

3.5 Mockups

3.5.1 Pannello dell'amministratore

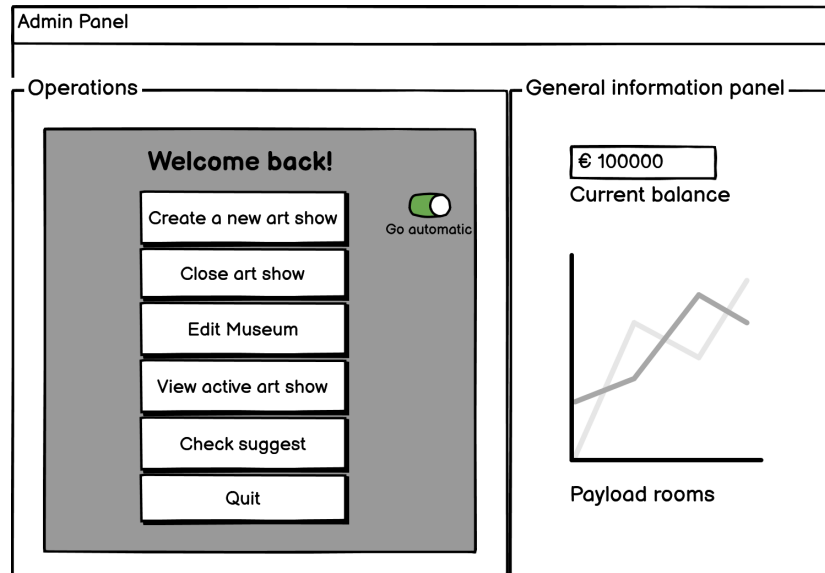


Figure 5: Admin Panel

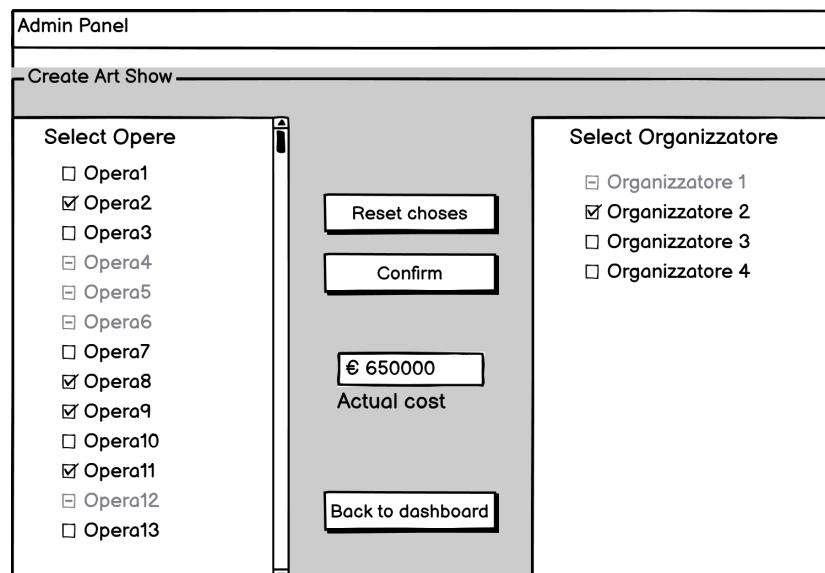


Figure 6: Creazione Mostra

3.5.2 Pannello Impiegato e Organizzatore

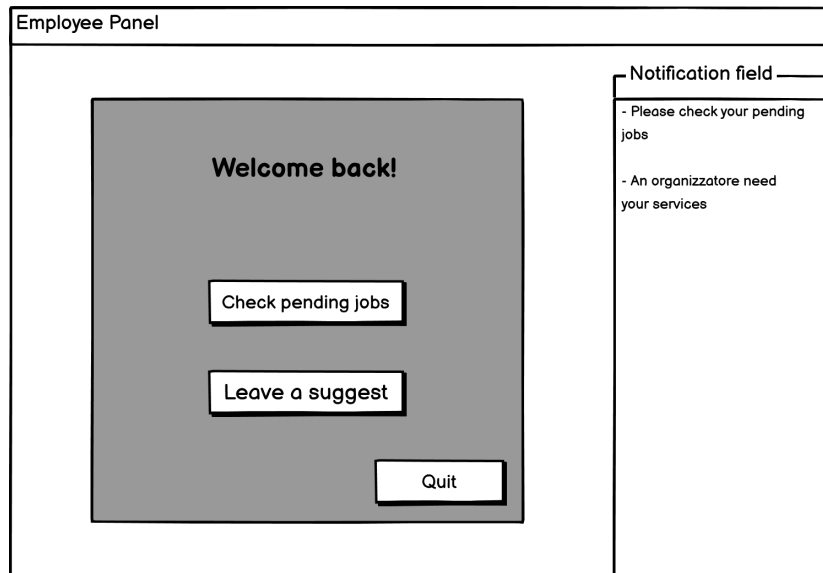


Figure 7: Dashboard Impiegato

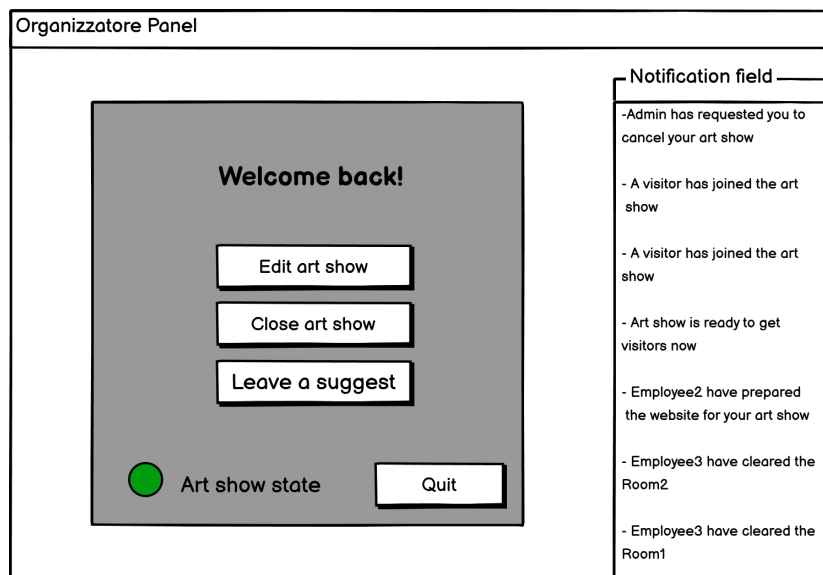


Figure 8: Dashboard Organizzatore

Organizzatore Panel

Realize Art show

Select Employees

☒ Employee 1

Select role

☒ Employee 2

Select role

☒ Employee 3

Select role

☐ Employee 4

☐ Employee 5

☒ Employee 6

Select role

☐ Employee 7

☐ Employee 8

☐ Employee 9

☐ Employee 10

☐ Employee 11

☐ Employee 12

☐ Employee 13

Reset choses

Confirm

€ 30000

Allocated funds

Back to dashboard

Select Rooms

☐ Room 1

☒ Room 2

☐ Room 3

☐ Room 4

☒ Room 3

☒ Room 4

☐ Virtual Room 1

☐ Virtual Room 2

☒ Virtual Room 3

☐ Virtual Room 4

☐ Virtual Room 5

☐ Virtual Room 6

☐ Virtual Room 7

Figure 9: Realizzazione Mostra

3.5.3 Pannello Visitatori

Visitor Panel

Welcome to Museum 2.0!

Enter as a visitor

Login

Register

Leave a suggest

Quit

Figure 10: Pannello visitatori

Visitor Unregistered Panel

Please choose an art show:

☒ Art show 1
☐ Art show 2
☐ Art show 3
☐ Art show 4

Show detail

Confirm

Back

Figure 11: Pannello scelta Mostra

Visitor Registered Panel

Please choose an art show:

☒ Art show 1
☐ Art show 2
☐ Art show 3
☐ Art show 4

Show detail

Confirm

Log out

Figure 12: Pannello scelta Mostra per visitatori registrati

3.6 Activity Diagram

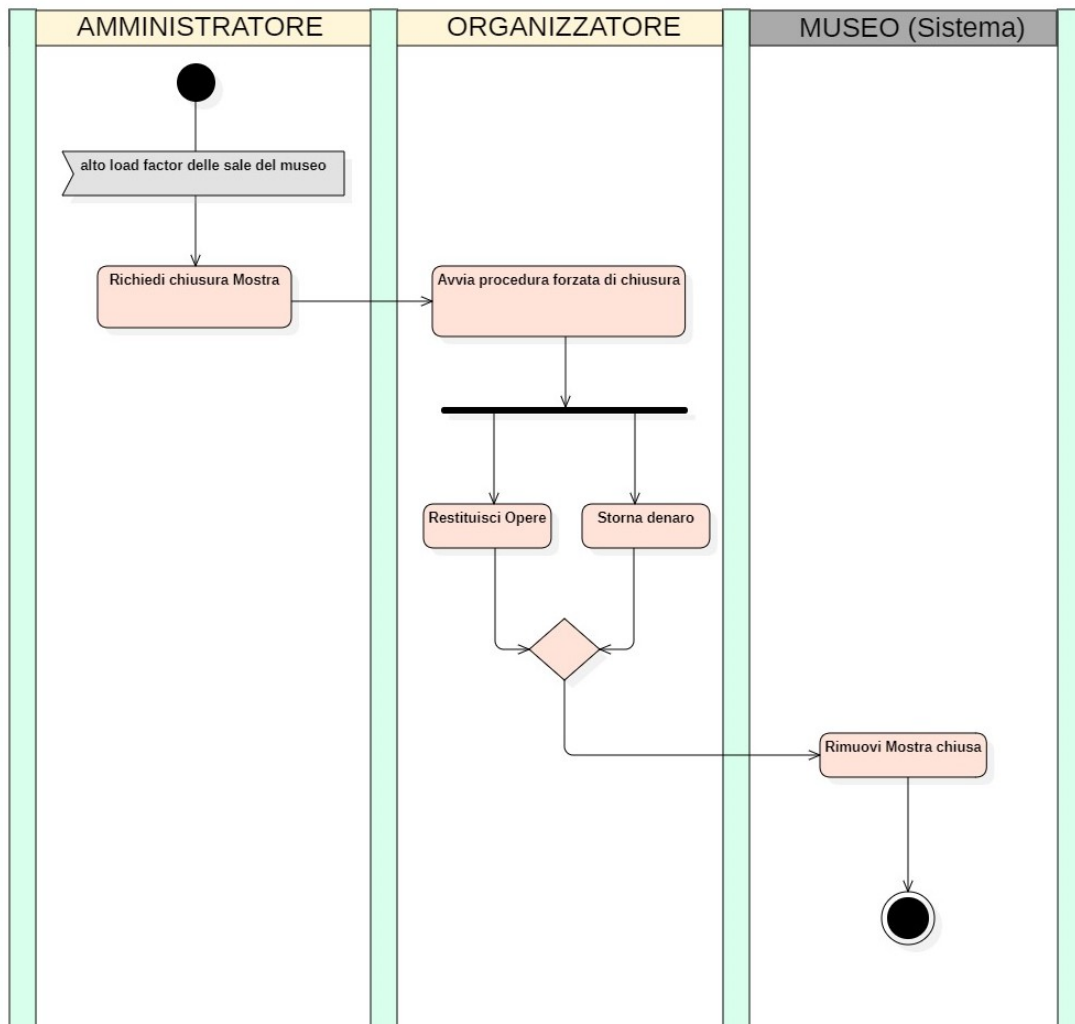


Figure 13: Activity diagram della procedura di chiusura forzata di una mostra

Part II

Implementazione

Le classi che ho scritto sono state tutte accorpate in vari package, per come visto nel package diagram in Sezione 2.2. Quello più grande, *strutturaMuseo*, è suddiviso in 3 più piccoli: *personale*, *amministratore* e *museo*. Gli altri due (tre se si considera il package view) sono invece a sè stanti. e verranno descritti dopo questi 3.

4 Package strutturaMuseo - package museo

Qui dentro sono contenute tutte quelle classi che chiamerei di struttura. Oltre alle varie classi di supporto, quale la classe Mostra e la classe Suggerimento, c'è la classe Museo che ha un ruolo centrale: tutte le classi in questo package forniscono un servizio al Museo stesso. Infatti oggetti quali Tickets, Sale, e le sopra citate Mostre e Suggerimenti, hanno tutte uno spazio più o meno di rilievo nella struttura del Museo. Inizio a definire le classi di contorno per poi concludere con la classe Museo.

4.1 Package sale

Le classi SalaVirtuale e SalaFisica sono entrambe specializzazioni della classe astratta Sala. Sono tutti oggetti che definirei inerti. Vengono create all'avvio del Sistema e inserite in un insieme immutabile di oggetti, e vengono passati i riferimenti agli organizzatori solo per cambiare lo stato da libere a occupate.

4.2 Package Amministratore - Classi e Interfacce

Questo package è composto da una classe Amministratore, e da una serie di classi a supporto, quali le classi delle varie strategie che può adottare e la classe delle IstanzeMostra.

4.2.1 Classe Amministratore

Questa classe implementa l'interfaccia Observer, essendo che deve aggiornare il suo stato ogni volta che nel museo (Observable) Di seguito una descrizione

5 Tests