

# Analisi dei Requisiti

## Indice generale

1. Introduzione.....	2
1.1 Scopo del Sistema.....	2
1.2 Ambito del Sistema.....	2
1.3 Obiettivi e Criteri del Sistema.....	2
1.4 Riferimenti.....	2
1.5 Sommario.....	2
2. Sistema Corrente.....	3
3. Sistema Proposto.....	3
3.1 Sommario.....	3
3.2 Requisiti Funzionali.....	3
3.3 Requisiti Non Funzionali.....	3
3.3.1 Usabilità.....	3
3.3.2 Affidabilità.....	3
3.3.3 Performance.....	3
3.3.4 Supportabilità.....	3
3.3.5 Implementazione.....	4
3.3.6 Distribuzione.....	4
3.4 Modelli di Sistema.....	4
3.4.1 Scenari.....	4
S-1. Actors: visitatore:Alice.....	4
S-2. Actors: visitatore:Bob, banca:ISP.....	4
S-3. Actors: curatore:Carlo.....	5
3.4.2 Use case model.....	6
Attori.....	6
UC-1: VisualizzaMostre.....	6
UC-2: VisualizzaMostraSingola.....	7
UC-3: AcquistaBiglietto.....	7
UC-4: LoginCuratore.....	9
UC-5: CreaMostra.....	9
UC-6: ModificaMostra.....	10
UC-7: CancellaMostra.....	10
UC-8: ImpostaDisponibilita.....	11
3.4.3 Object model.....	12
3.4.3.1 BoundaryObjects.....	12
3.4.3.2 ControlObjects.....	13
3.4.3.3 EntityObjects.....	13
3.4.4 Dynamic model.....	13
3.4.5 Interfaccia Utente: Cammini di navigazione e Mockup.....	13
4. Glossario.....	13

# **1. Introduzione**

## **1.1 Scopo del Sistema**

Lo scopo del sistema è di rendere disponibile e gestire un sito perlopiù informativo dedicato al Museo Belpaese.

Il cliente ha richiesto che il sito sia attraente e invitante al pubblico, in modo da fungere anche scopo pubblicitario.

I dati resi disponibili sul sito sono perlopiù riguardanti le mostre del Museo, che variano nel tempo. Si deve quindi provvedere anche l'abilità ai curatori di modificare i dati delle mostre.

Altre informazioni sono la storia del museo e i suoi curatori.

Oltre la visualizzazione di informazioni, il sito deve permettere a visitatori di acquistare biglietti per il museo. Questi esistono in numero limitato, quindi il sistema deve anche verificare la loro disponibilità.

## **1.2 Ambito del Sistema**

Il sito sarà modesto, di poche (circa 4-5) pagine. Avrà un database associato con pochi (1-3) schemi. La maggior parte del peso del filesystem sarà composto da immagini illustrative del museo e delle mostre. Vi sono due tipi di utenti: Visitatori e Curatori.

## **1.3 Obiettivi e Criteri del Sistema**

Il sistema deve presentare le informazioni relative al museo in modo chiaro e gradevole, e permettere l'acquisto dei biglietti. L'acquisto deve essere sicuro e non deve mai fallire dopo il pagamento, pena il rimborso del costo.

Deve essere eseguito con successo sull'hardware del museo, ed essere di facile usabilità da parte del suo staff.

Inoltre, l'aspetto del sito deve risultare gradevole al direttore del museo.

## **1.4 Riferimenti**

Vedere i documenti allegati.

- Problem Statement
- System Design

## **1.5 Sommario**

Il sistema è un semplice sito gestito dal museo e facilmente modificabile dal suo staff, che può gestire la vendita di biglietti.

## **2. Sistema Corrente**

Finora, il Museo Belpaese ha lasciato la pubblicità a volantini, manifesti e giornalisti locali. L'acquisto di biglietti finora è stato solo possibile all'entrata. La produzione dei volantini e manifesti è stata fatta via Microsoft Word, e sono stati distribuiti in vari punti del comune di Salerno.

## **3. Sistema Proposto**

### **3.1 Sommario**

Un sito web responsive e dinamico con una homepage attraente, una lista delle mostre in corso modificabile dai curatori del museo, e la possibilità di acquistare biglietti digitali.

### **3.2 Requisiti Funzionali**

I visitatori possono visualizzare la pagina principale, la pagina delle mostre e la pagina di ogni mostra. Inoltre possono acquistare biglietti.

I curatori devono poter accedere al sito con email e password, aggiungere, cancellare, modificare e pubblicare mostre, e gestire la disponibilità dei biglietti.

### **3.3 Requisiti Non Funzionali**

#### **3.3.1 Usabilità**

Le interfacce utente devono essere chiare e facili da usare. Sia visitatori che staff sono poco esperti con sistemi informatici, e in particolare i visitatori possono essere anche completamente inesperti con interfacce web. Occorrerà provvedere istruzioni d'uso anche allo staff del museo, abbastanza inesperto.

#### **3.3.2 Affidabilità**

L'affidabilità è un'elevata priorità nella vendita di biglietti. In particolare, la più grande criticità è la possibilità che il cliente paghi il biglietto e poi non lo riceva.

Il Museo ha garantito la disponibilità di una decina di biglietti oltre il limite di vendita in ogni giorno per coprire la possibilità del loro acquisto nel caso di un leggero malfunzionamento. Ciò permette al sistema un po' di flessibilità con la disponibilità dei biglietti.

Resta cruciale garantire che il pagamento del biglietto venga effettuato se e solo se il biglietto è effettivamente inviato.

#### **3.3.3 Performance**

Non vi sono elevati requisiti di performance. Il sito deve rispondere a tutte le richieste web entro tre secondi e supportare fino a una decina di utenti simultaneamente.

#### **3.3.4 Supportabilità**

In futuro, il museo potrebbe estendere le funzionalità del sito in vari modi, permettendo

l'aggiunta di pagine per singoli reperti, più tipi di staff, e una sezione notizie. Potrebbe aumentare anche il numero di utenti concorrenti. Bisognerà garantire la possibilità di espandere ogni parte del sistema, e di trasferirlo su macchine più potenti.

La manutenzione verrà gestita su due livelli: da un membro dello staff del museo dalle qualifiche medie-basse (un tirocinante universitario che ha seguito un semplice corso di programmazione in Java e Python) e dal nostro team di supporto, contattabile da remoto.

### **3.3.5 Implementazione**

Il museo ha hardware limitato. La macchina disposta per l'installazione e operazione del sistema è un PC con prestazioni medio-basse.

Il nostro team di testing è molto limitato, quindi si desidera minimizzare la quantità di 'parti mobili' capaci di fallire.

### **3.3.6 Distribuzione**

Il Sistema verrà installato una sola volta dal cliente. Verrà installato dal cliente stesso con il supporto di un ingegnere consultato in maniera telematica al momento.

## **3.4 Modelli di Sistema**

### **3.4.1 Scenari**

#### **S-1. Actors: visitatore:Alice**

1. Alice passa davanti al museo durante la giornata, e considera di visitarlo in settimana.
2. Accede al sito del museo. Vede immediatamente la homepage.
3. Incuriosita, passa alla sezione delle mostre, e vede la mostra 'Cucina nell'Antica Roma'.
4. Va alla pagina della mostra, e legge la sua descrizione. Interessata da alcuni dei reperti descritti, decide di vederli dal vivo. Vede che la mostra è temporanea, e chiuderà entro fine mese.
5. Sapendo la data della mostra, Alice decide di fare una gita in museo il prossimo sabato, e abbandona il sito. Compra il biglietto all'entrata.

#### **S-2. Actors: visitatore:Bob, banca:ISP**

1. Bob è un appassionato di musei. Un po' annoiato, decide di controllare il sito del museo per vedere se vi sono mostre interessanti.
2. Entra nel sito, e immediatamente vede che la mostra 'Gladiatori' arriva al museo la settimana prossima.
3. Interessato, seleziona la mostra e vede la sua pagina.
4. Soddisfatto, decide di acquistare un biglietto. Va alla pagina di acquisto, inserisce la sua email e i dati della sua carta di credito, seleziona la data in cui vuole visitare il museo e acquista il suo biglietto.
5. Il sito richiede il pagamento dalla banca di Bob, usando tali dati. Se il pagamento va a buon fine, invia il biglietto virtuale a Bob via email.
6. Giorni dopo, quando visita il museo, Bob accede alla sua email e visualizza il suo

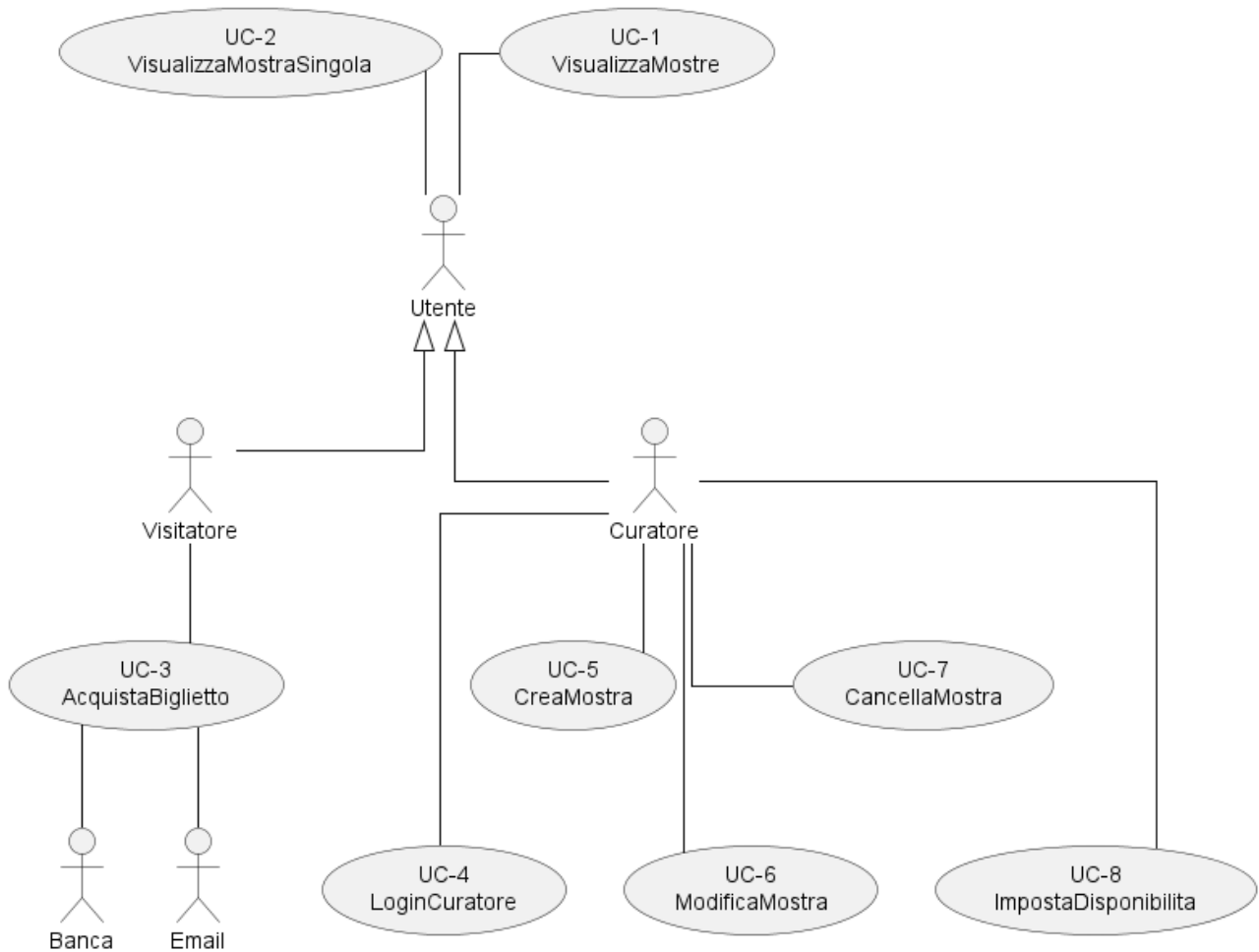
biglietto virtuale. Lo presenta all'entrata, e entra nel museo.

**S-3. Actors: *curatore:Carlo***

1. Carlo è un curatore del museo.
2. Accede al sito con email e password, e va alla sezione mostre.
3. Crea una mostra, e vi aggiunge un titolo e una descrizione.
4. Una volta che la mostra è pronta, accede al sito, e la pubblica.

5.

### 3.4.2 Use case model



#### Attori

- Utente
  - Curatore
  - Visitatore
- Banca
- Email

#### UC-1: VisualizzaMostre

Nome	VisualizzaMostre
Attori	Utente
Flusso di Eventi	1. L'Utente seleziona di visualizzare le Mostre. 2. Il <b>Sistema</b> reindirizza l'Utente alla pagina delle Mostre.
Condizione d'Entrata	L'Utente entra nella pagina delle Mostre

Condizione d'Uscita	L'Utente esce dalla pagina delle Mostre
---------------------	---

### **UC-2: VisualizzaMostraSingola**

Nome	VisualizzaMostraSingola
Attori	Utente
Flusso di Eventi	<ol style="list-style-type: none"> <li>1. L'Utente seleziona di visualizzare una Mostra.</li> <li>2. Il <b>Sistema</b> reindirizza l'Utente alla pagina della Mostra.</li> </ol>
Condizione d'Entrata	L'Utente entra nella pagina delle Mostra Singola
Condizione d'Uscita	L'Utente esce dalla pagina della Mostra Singola

### **UC-3: AcquistaBiglietto**

Nome	AcquistaBiglietto
Attori	Visitatore, Banca, Email
Flusso di Eventi	<ol style="list-style-type: none"> <li>1. Il Visitatore seleziona l'acquisto di un Biglietto per la Mostra.</li> <li>2. Il <b>Sistema</b> offre un modulo per l'inserimento dei dati necessari per l'acquisto, limitando la selezione di data e numero di biglietti in base alla loro Disponibilita in ogni data.</li> <li>3. Il Visitatore inserisce i dati e li invia.</li> <li>4. Il <b>Sistema</b> verifica la correttezza dei dati e richiede la conferma dell'acquisto.</li> <li>5. Il Visitatore conferma l'acquisto.</li> <li>6. Il <b>Sistema</b> crea un Biglietto e chiede il pagamento dalla Banca.</li> <li>7. La Banca conferma il pagamento.</li> <li>8. Il <b>Sistema</b>: <ol style="list-style-type: none"> <li>1. Conferma il successo dell'acquisto al Visitatore</li> <li>2. Aggiorna la Disponibilita della Mostra.</li> <li>3. Invia il Biglietto all'indirizzo Email specificato dal Visitatore.</li> </ol> </li> </ol>
Condizione d'Entrata	Il Visitatore accede alla pagina della Mostra.
Condizione d'Uscita	<p>Il Visitatore esce dalla pagina della Mostra.</p> <p>E SE è stato eseguito il passo 6 ALLORA è stato eseguito il passo 8.</p>

Flussi Alternativi	<p>Se al passo 4 la verifica dei dati fallisce OPPURE al passo 7 la banca rifiuta la richiesta di pagamento E il Visitatore non ha lasciato la pagina:</p> <ol style="list-style-type: none"><li>1. Il <b>Sistema</b> presenta un messaggio di errore appropriato. 'Errore di Pagamento'</li><li>2. Torna al passo 3.</li></ol> <p>Se al passo 1 non vi sono Disponibilita</p> <ol style="list-style-type: none"><li>1. Il <b>Sistema</b> impedisce l'acquisto con un messaggio appropriato 'Nessun biglietto disponibile.'</li><li>2. Termina il caso d'uso.</li></ol>									
Dati Richiesti	<ul style="list-style-type: none"><li>• Nome (stringa di max 50 caratteri)</li><li>• Cognome (stringa di max 50 caratteri)</li><li>• Email (stringa in formato email (<a href="#">xxxx@yyy.zzz</a>))</li><li>• Conferma Email (stringa identica alla precedente)</li> <li>• # Carta (stringa di 16 caratteri numerici)</li><li>• Data Scadenza Carta (MM/YY)</li><li>• Codice Sicurezza Carta (stringa di 3 caratter numerici)</li> <li>• # Biglietti (intero, default 1)</li><li>• Data Biglietto (DD/MM/YY, compresa tra data inizio Mostra/data corrente (la maggiore dei due) e data fine Mostra)</li></ul>									
Requisiti di Qualità	<p>Garantire l'invio del biglietto a pagamento effettuato. Limitare la possibilità di inviare un biglietto senza ricevere pagamento.</p> <table><tr><td><b>Possibilità</b></td><td>Pagato</td><td>Non Pagato</td></tr><tr><td>Inviato</td><td>OK</td><td>NO</td></tr><tr><td>Non Inviato</td><td>NO!!!</td><td>OK</td></tr></table>	<b>Possibilità</b>	Pagato	Non Pagato	Inviato	OK	NO	Non Inviato	NO!!!	OK
<b>Possibilità</b>	Pagato	Non Pagato								
Inviato	OK	NO								
Non Inviato	NO!!!	OK								



#### UC-4: LoginCuratore

Nome	LoginCuratore
Attori	Curatore
Flusso di Eventi	<ol style="list-style-type: none"><li>1. Il Curatore inserisce i dati del suo Account e li invia.</li><li>2. Il <b>Sistema</b> verifica se esiste un Account corrispondente all'username con password associata è corretta, completa il Login e reindirizza alla pagina delle Mostre.</li></ol>
Condizione d'Entrata	Il Curatore accede alla pagina di Login per Curatori.
Condizione d'Uscita	Il Curatore abbandona la pagina di Login per Curatori.
Dati Richiesti	<ul style="list-style-type: none"><li>• Username (Stringa di massimo 20 caratteri alfanumerici)</li><li>• Password (Stringa tra 20 e 100 caratteri alfanumerici e/o speciali)</li></ul>
Flussi Alternativi	<p>Se al passo 2, la verifica fallisce:</p> <ol style="list-style-type: none"><li>1. Il <b>Sistema</b> verifica quanti tentativi di login sono stati fatti.<ol style="list-style-type: none"><li>1. Se meno di 3, torna al passo 1 con un messaggio di errore appropriato. 'Nome Utente o Password scorretti.'</li><li>2. Se almeno 3, impedisce ulteriori tentativi di login all'Utente.</li></ol></li></ol>

#### UC-5: CreaMostra

Nome	CreaMostra
Attori	Curatore
Flusso di Eventi	<ol style="list-style-type: none"><li>1. Il Curatore decide di creare una Mostra.</li><li>2. Il <b>Sistema</b> offre un Form per l'inserimento dei dati della nuova Mostra.</li><li>3. Il Curatore inserisce i dati richiesti.</li><li>4. Il <b>Sistema</b> salva la Mostra.</li></ol>
Condizione d'Entrata	Il Curatore entra nella pagina delle Mostre
Condizione d'Uscita	Il Curatore esce dalla pagina delle Mostre
Dati Richiesti	<ul style="list-style-type: none"><li>• Nome (Stringa di massimo 100 caratter alfanumerici)</li><li>• Corpo (Stringa di massimo 10.000 caratteri alfanumerici)</li><li>• Stato (In preparazione, Pubblicata, Conclusa)</li><li>• Data Inizio (DD/MM/YYYY)</li><li>• Data Fine (DD/MM/YYYY)</li></ul>

**UC-6: ModificaMostra**

Nome	ModificaMostra
Attori	Curatore
Flusso di Eventi	<ol style="list-style-type: none"><li>1. Il Curatore seleziona di modificare la Mostra.</li><li>2. Il <b>Sistema</b> offre un modulo per la modifica della Mostra.</li><li>3. Il Curatore inserisce i nuovi dati e conferma la modifica.</li><li>4. Il <b>Sistema</b> salva i dati modificati.</li></ol>
Condizione d'Entrata	Il Curatore entra nella pagina di una Mostra.
Condizione d'Uscita	Il Curatore esce dalla pagina di una Mostra.

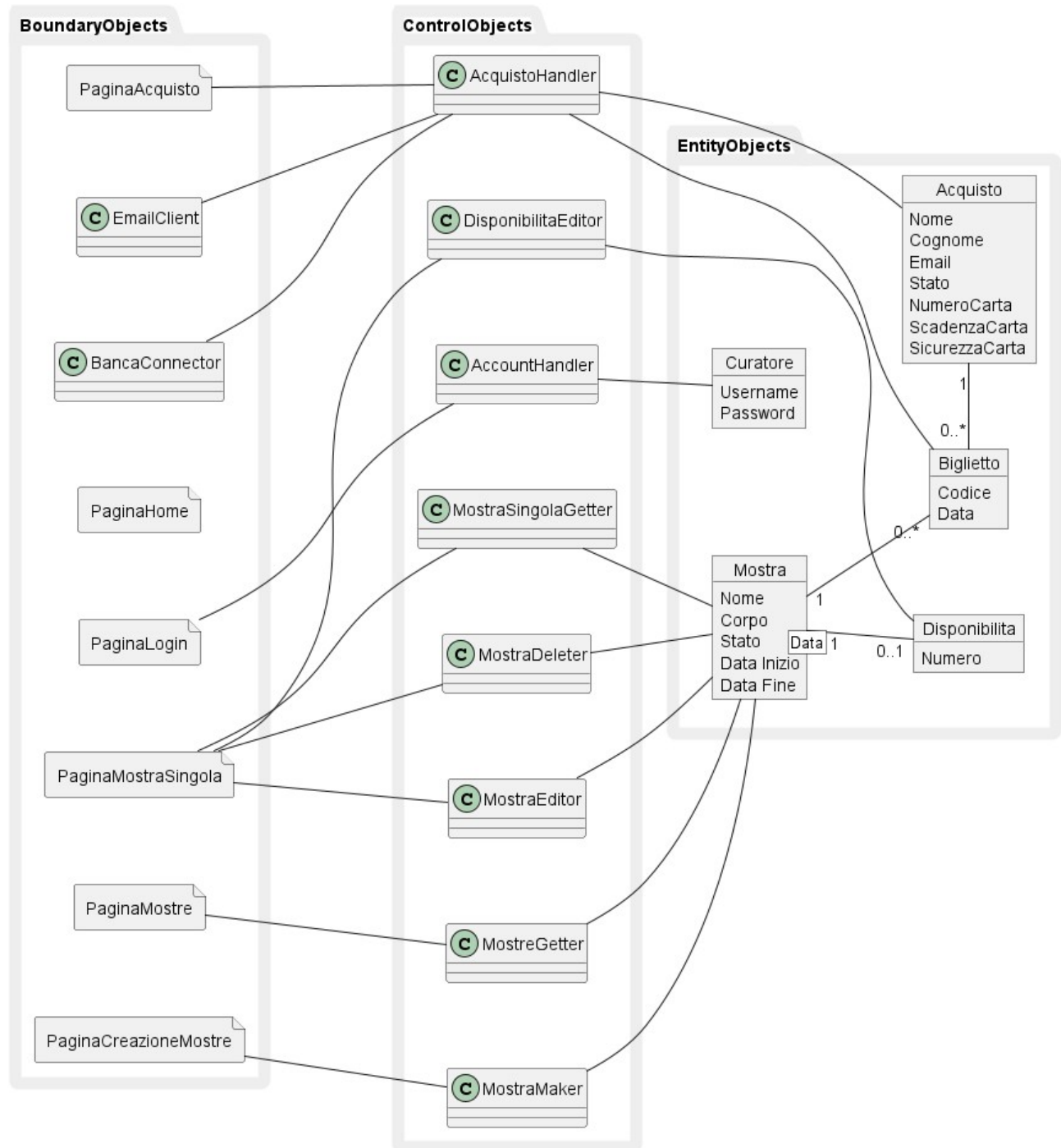
**UC-7: CancellaMostra**

Nome	CancellaMostra
Attori	Curatore
Flusso di Eventi	<ol style="list-style-type: none"><li>1. Il Curatore seleziona di cancellare la Mostra.</li><li>2. Il <b>Sistema</b> richiede la conferma dell'operazione.</li><li>3. Il Curatore conferma la pubblicazione.</li><li>4. Il <b>Sistema</b> rimuove la Mostra.</li></ol>
Condizione d'Entrata	Il Curatore entra nella pagina di una Mostra
Condizione d'Uscita	Il Curatore esce dalla pagina di una Mostra

**UC-8: ImpostaDisponibilita**

Nome	ImpostaDisponibilita
Attori	Curatore
Flusso di Eventi	<ol style="list-style-type: none"><li>1. Il Curatore decide di impostare la Disponibilità di una Mostra.</li><li>2. Il <b>Sistema</b> gli offre la lista delle date in cui è aperta la Mostra.</li><li>3. Il Curatore:<ol style="list-style-type: none"><li>1. Seleziona una data, e imposta il numero di biglietti disponibili in tale data.</li><li>2. Salva le modifiche.</li></ol></li><li>4. Il <b>Sistema</b> salva le modifiche fatte dal Curatore.</li></ol>
Condizione d'Entrata	Il Curatore entra nella pagina delle Disponibilità
Condizione d'Uscita	Il Curatore esce dalla pagina delle Disponibilità

### 3.4.3 Object model



#### 3.4.3.1 BoundaryObjects

- Pagine: Le varie pagine descritte nei Casi d'Uso.
- EmailClient: Il servizio che si occupa di mandare email.
- BancaConnector: Il servizio che si occupa di richiedere pagamenti dalla banca.

### **3.4.3.2 ControlObjects**

- MostreGetter: Si occupa del prelievo dei dati di tutte le Mostre per la visualizzazione. (UC-1)
- MostraSingolaGetter: Si occupa del prelievo dei dati di una Mostra per la visualizzazione (UC-2)
- AcquistoHandler: Si occupa delle logiche applicative relative all'acquisto di biglietti (UC-3)
- AccountHandler: Si occupa del login per i curatori (UC-4)
- MostraMaker: Si occupa della creazione di una Mostra (UC-5)
- MostraEditor: Si occupa della modifica di una Mostra (UC-6)
- MostraDeleter: Si occupa della cancellazione di una Mostra (UC-7)
- DisponibilitaEditor: Si occupa della modifica della disponibilità dei biglietti per le mostre. (UC-8)

### **3.4.3.3 EntityObjects**

- Acquisto: I dati di un acquisto. Include un insieme di Biglietti, e
- Biglietto: Un biglietto per visitare una Mostra in una specifica data. Il suo codice permette l'entrata in museo.
- Curatore: L'account associato a un Curatore.
- Disponibilita: Una coppia data-numero associata a una Mostra che indica quanti biglietti sono disponibili per tale data.
- Mostra: Una Mostra del museo. Può essere in preparazione, pubblicata, o terminata.

## **3.4.4 Dynamic model**

## **3.4.5 Interfaccia Utente: Cammini di navigazione e Mockup**

# **4. Glossario**