

Progetto TravelON

Montrasio Filippo 875551
Volpato Mattia 866316
Battaglia Walter 845502

13 Giugno 2022

Indice

1	Casi d'Uso	3
1.1	Diagramma Casi d'Uso	3
1.2	Attori	4
1.3	Descrizione Casi d'uso e SSD	5
1.3.1	Casi d'uso non dettagliati	5
1.3.2	Casi d'uso dettagliati e relativi SSD	6
1.4	Modello di Dominio	11
2	Contratti, SD e Diagramma delle classi software	13
2.1	Modello delle classi software	13
2.2	Contratti	16
2.3	SD	17
2.4	Pattern Utilizzati	20

1 Casi d'Uso

1.1 Diagramma Casi d'Uso

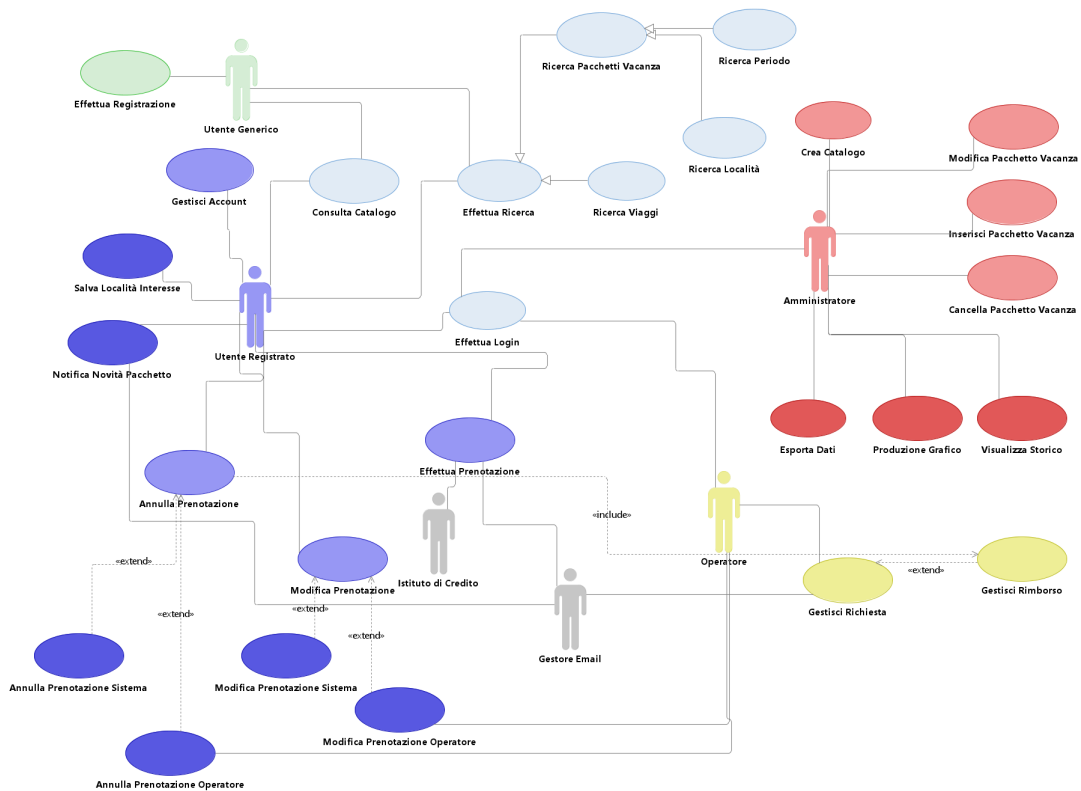


Figura 1: Sviluppato da: Gruppo

Partendo dal modello iniziale sono stati inseriti i seguenti casi d'uso:

- Salva Località Interesse
- Notifica Novità Pacchetto
- Esporta Dati
- Produzione Grafico
- Visualizza Storico
- Annulla Prenotazione Sistema
- Modifica Prenotazione Sistema
- Annulla Prenotazione Operatore

- Modifica Prenotazione Operatore

Il caso d'uso presente nel diagramma iniziale *Modifica Prenotazione* è stato modificato, aggiungendo, attraverso una *extend* i casi d'uso *Modifica Prenotazione Sistema* e *Modifica Prenotazione Operatore* in quanto rispetto alla vecchia versione di **Travel On**, la nuova versione consente all'UtenteRegistrato di modificare una prenotazione direttamente da sistema, oltre che poter richiedere la modifica della prenotazione ad un operatore.

Il caso d'uso presente nel diagramma iniziale *Annulla Prenotazione* è stato modificato, aggiungendo, attraverso una *extend* i casi d'uso *Annulla Prenotazione Sistema* e *Annulla Prenotazione Operatore* in quanto rispetto alla vecchia versione di **Travel On**, la nuova versione consente all'UtenteRegistrato di annullare una prenotazione direttamente da sistema, oltre che poter richiedere l'annullamento della prenotazione ad un Operatore.

In entrambi i casi è stata utilizzata una *extend* per creare una relazione tra i casi d'uso in quanto consente di collegare un caso d'uso esteso ad un caso d'uso base e ci permette di aggiungere delle varianti (Modifica Prenotazione Sistema, Modifica Prenotazione Operatore, Annulla Prenotazione Sistema, Annulla Prenotazione Operatore) al caso d'uso base (Modifica Prenotazione, Annulla Prenotazione).

I casi d'uso *Modifica Prenotazione Operatore* e *Annulla Prenotazione Operatore* sono stati di conseguenza associati all'attore Operatore.

I casi d'uso *Salva Località Interesse* e *Notifica Novità Pacchetto* sono stati associati all'attore UtenteRegistrato in quanto è il cliente (UtenteRegistrato) a poter salvare le località di interesse e ricevere notifiche riguardo i nuovi pacchetti da parte del sistema.

1.2 Attori

Gli attori coinvolti nel progetto sono:

- Attori Primari
 - Amministratore
 - Operatore
- Attori Finali
 - UtenteGenerico
 - UtenteRegistrato
- Attori di Supporto
 - Istituto di credito
 - Gestore Mail

1.3 Descrizione Casi d'uso e SSD

I casi d'uso descritti sono i seguenti:

- Salva Località Interesse
- Annulla Prenotazione Sistema
- Esporta Dati

1.3.1 Casi d'uso non dettagliati

UC001 -Salva Località Interesse- L'Utente Registrato può salvare nel proprio account le località di interesse. (Sviluppato da: Montrasio Filippo)

UC002 -Annulla Prenotazione Sistema- Il Cliente può annullare la propria prenotazione direttamente da sistema (Sviluppato da: Battaglia Walter)

UC003 -Esporta Dati- L'Amministratore può esportare i dati (Sviluppato da: Volpato Mattia)

1.3.2 Casi d'uso dettagliati e relativi SSD

<i>UC001 -Salva Località Interesse-</i>	
Nome Caso d'uso	Salva Località Interesse
Portata	Sistema <i>TravelOn</i>
Livello	Obiettivo Utente
Attore Primario	Utente Registrato
Parti interessate e Interessi	<ul style="list-style-type: none"> • UtenteRegistrato (AttorePrimario): Vuole poter salvare le località interessate per poter successivamente ricevere notifiche relative a nuovi pacchetti • Sistema: vuole inviare a UtentiRegistrato notifiche automatiche riguardo le località di interesse degli stessi utenti.
PreCondizioni	Utente si è autenticato a UtenteRegistrato
PostCondizioni	<ul style="list-style-type: none"> • Il Sistema ha salvato le località d'interesse selezionate da UtenteRegistrato • Il Sistema potrà mandare informazioni relative a nuovi pacchetti a Utente Registrato
Scenario Principale di successo	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando Utente Registrato decide di voler ricevere automaticamente notizie riguardo l'aggiunta di nuovi pacchetti 2. Il sistema mostra le località al quale sono associati i pacchetti 3. Utente Registrato inserisce le località di interesse <ul style="list-style-type: none"> • Utente Registrato esegue il passo 3 fino a quando non ha inserito tutte le località di interesse 4. Utente Registrato ha inserito le località di interesse 5. Utente Registrato conferma le località selezionate 6. Il sistema salva le località selezionate da Utente Registrato 7. Il sistema conferma la registrazione di Utente Registrato
Estensione	<ol style="list-style-type: none"> 3.a il formato della località inserita da Utente Registrato è errato <ol style="list-style-type: none"> 3.a.a Sistema avvisa Utente Registrato di inserire nuovamente la località 3.a.b Utente Registrato corregge la località inserita 3.b Utente Registrato non vuole inserire nessuna città <ol style="list-style-type: none"> 3.b.a Sistema avvisa Utente Registrato che deve inserire almeno una località 3.b.b Utente Registrato inserisce una località
Extension Point	-
Requisiti Speciali	-
Frequenza di ripetizione	L'esecuzione del caso d'uso avviene ogni volta che Utente Registrato vuole salvare una nuova località. La frequenza potrebbe essere ininterrotta

SSDSalvaLocalitaInteresse

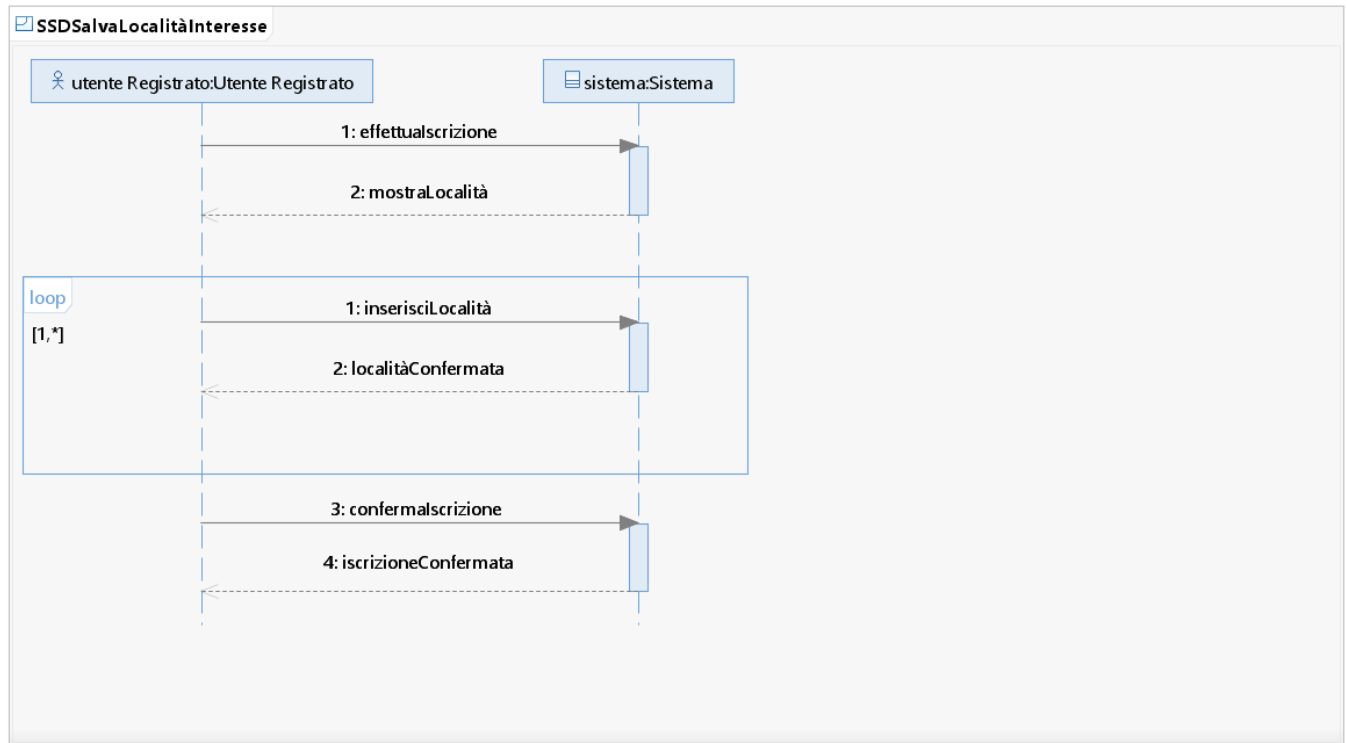


Figura 2: Sviluppato da: Montrasio Filippo

<i>UC002 -Annulla Prenotazione Sistema-</i>	
Nome Caso d'uso	Annulla Prenotazione Sistema
Portata	Sistema <i>TravelOn</i>
Livello	Obiettivo Utente
Attore Primario	Utente Registrato
Parti interessate e Interessi	UtenteRegistrato (AttorePrimario): vuole annullare una prenotazione effettuata
PreCondizioni	Utente Registrato ha già effettuato il login ed una prenotazione, ed effettuato il pagamento della caparra
PostCondizioni	La prenotazione viene annullata e gestito il rimborso
Scenario Principale di successo	<ol style="list-style-type: none"> 1. L'utente Registrato specifica di voler eliminare una prenotazione 2. Il sistema mostra le prenotazioni effettuate dall'Utente Registrato 3. L'utente Registrato seleziona la prenotazione da annullare 4. Il sistema annulla la prenotazione 5. Viene gestito il rimborso: Includi GestisciRimborso 6. il sistema notifica via mail l'annullamento della prenotazione
Estensione	<ol style="list-style-type: none"> 3.a L'utente non seleziona alcuna prenotazione <ol style="list-style-type: none"> 3.a.a il sistema comunica che non 'e stata selezionata alcuna scelta e annulla l'operazione
Extension Point	-
Requisiti Speciali	-
Frequenza di ripetizione	il caso d'uso viene ripetuto ogni qual volta che un Utente Registrato vuole annullare una prenotazione

SSDAnnullaPrenotazioneSistema

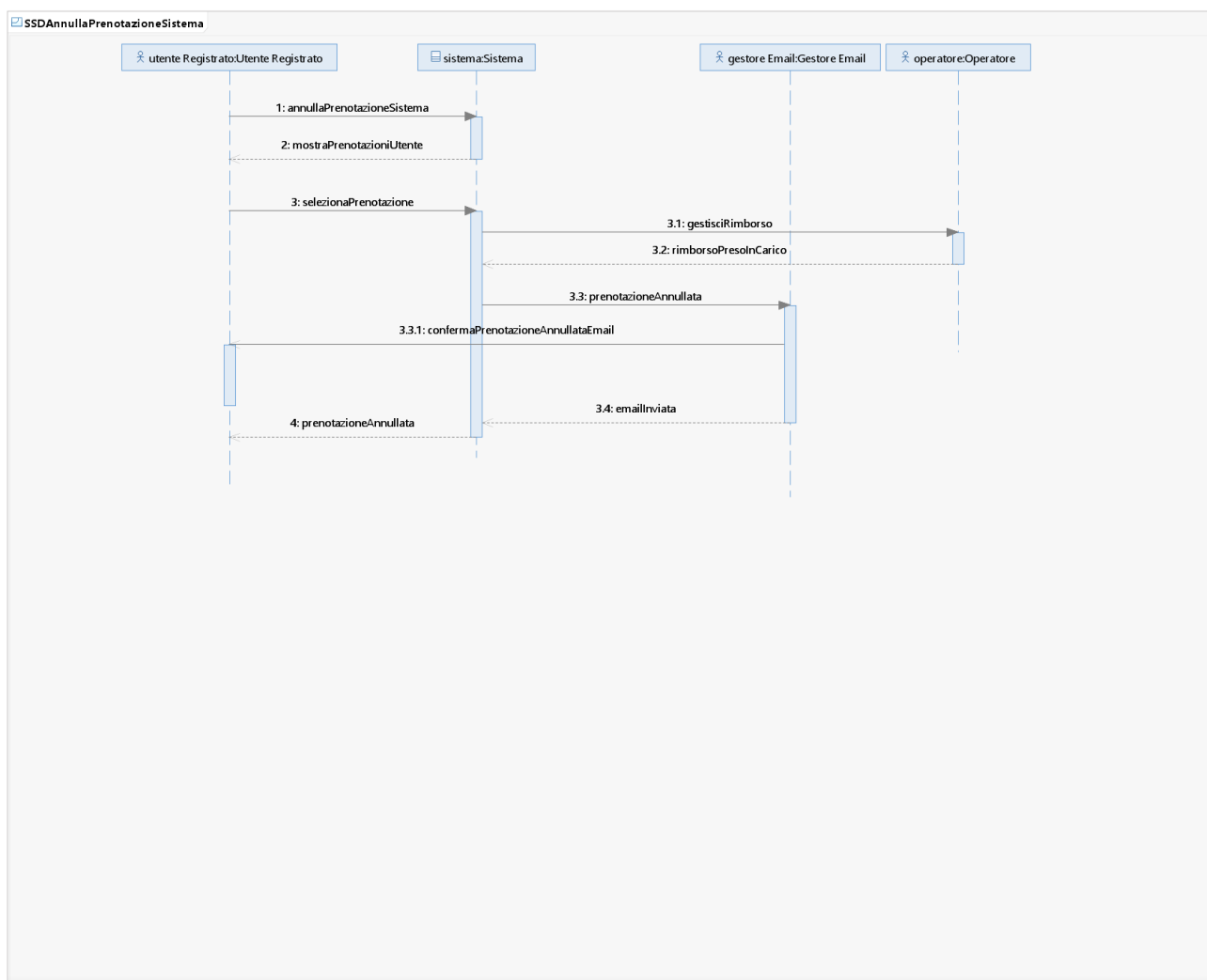


Figura 3: Sviluppato da: Battaglia Walter

<i>UC003 -Esporta Dati-</i>	
Nome Caso d'uso	Esporta Dati
Portata	Sistema <i>TravelOn</i>
Livello	Obiettivo Utente
Attore Primario	Amministratore
Parti interessate e Interessi	Amministratore (AttorePrimario): L'amministratore vuole esportare dei dati relativi alle prenotazioni in un formato selezionato pacchetti
PreCondizioni	L'amministratore si è autenticato correttamente nel sistema
PostCondizioni	I dati selezionati sono stati esportati nel formato selezionato con successo
Scenario Principale di successo	<ol style="list-style-type: none"> 1. L'amministratore comunica di voler effettuare un'operazione di esportazione dei dati 2. Il sistema mostra quali dati è possibile esportare 3. L'amministratore seleziona i dati desiderati 4. Il sistema mostra i formati in cui 'e possibile esportare i dati 5. L'amministratore seleziona il formato desiderato <ul style="list-style-type: none"> • L'amministratore ripete il passo 5 finch'è non ha selezionato tutti i formati desiderati 6. Il Sistema esporta con successo i dati
Estensione	<ol style="list-style-type: none"> 2.a Nel sistema non sono presenti dati <ol style="list-style-type: none"> 2.a.a Il sistema comunica all'amministratore che non 'e presente alcun dato 5.a L'amministratore non seleziona alcun formato <ol style="list-style-type: none"> 5.a.a Il sistema esporta i dati in un formato predefinito
Extension Point	-
Requisiti Speciali	-
Frequenza di ripetizione	Potenzialmente ininterrotto

SSDEsportaDati

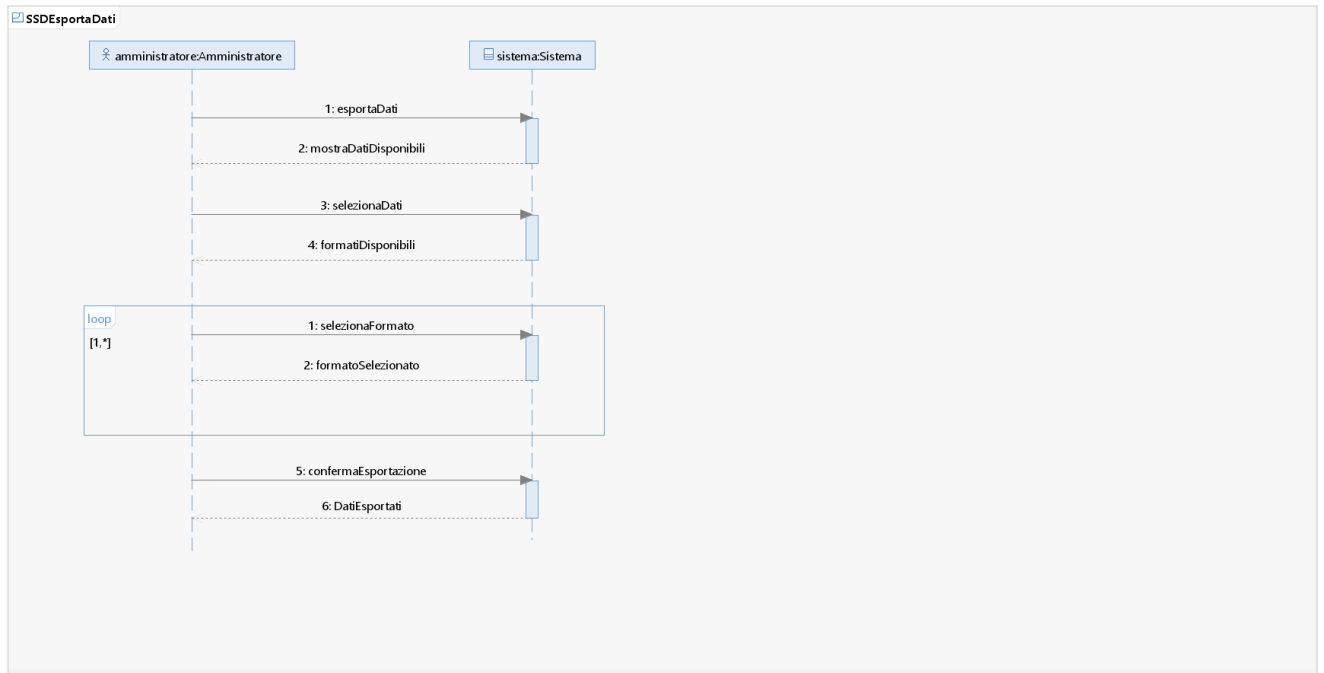


Figura 4: Sviluppato da: Volpato Mattia

1.4 Modello di Dominio

Partendo dal modello di dominio fornito è stato sviluppato un modello di dominio aggiornato nel quale sono state implementate 2 classi necessarie e ricavate dai casi d'uso e dalla consegna:

- Classe: **Località**
- Classe: **StoricoPrenotazione** La classe **StoricoPrenotazione** è stata aggiunta al modello di dominio come conseguenza della consegna *"L'Amministratore può visualizzare lo storico delle prenotazioni"*. La classe è stata messa in relazione con la classe **Prenotazioni** attraverso una aggregazione che mostra:
 - Aggregato: **StoricoPrenotazione**
 - Parte: **Prenotazione**

La classe **Località** è stata aggiunta in quanto UtenteRegistrato può scegliere le località di interesse, per ricevere notifiche su nuovi pacchetti. La classe **Località** è associata sia con UtenteRegistrato (per la scelta delle località di interesse) sia con PacchettoVacanza.

Sono stati aggiunti alla classe **Località** gli attributi:

- nome
- nazione

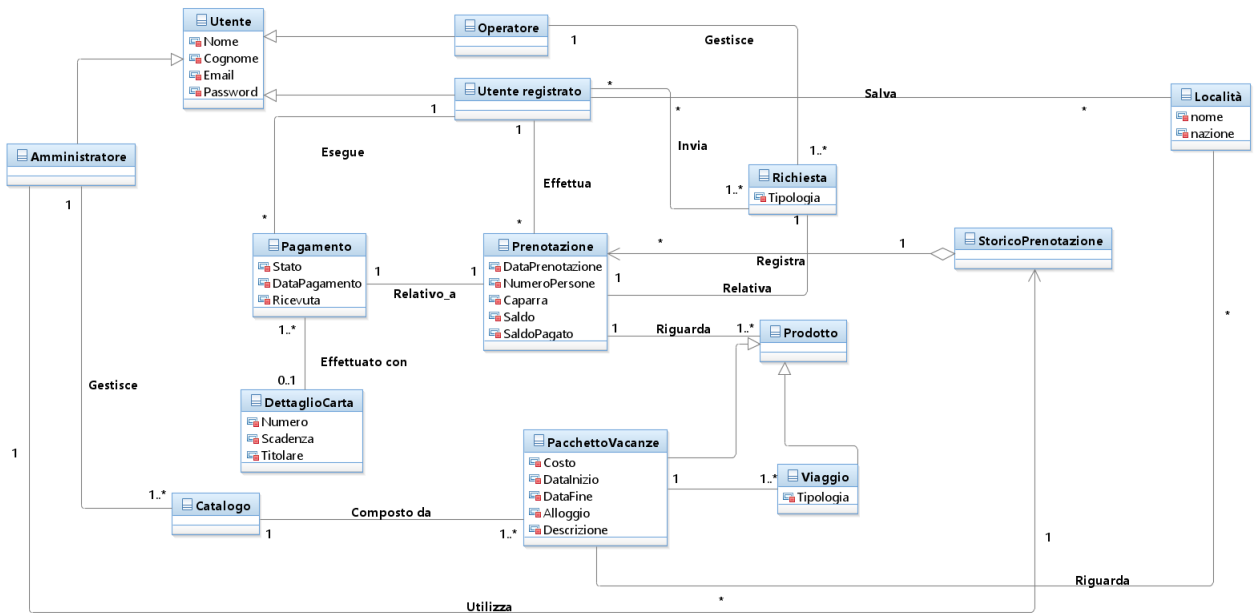


Figura 5: Sviluppato da: Gruppo

2.1 Modello delle classi software

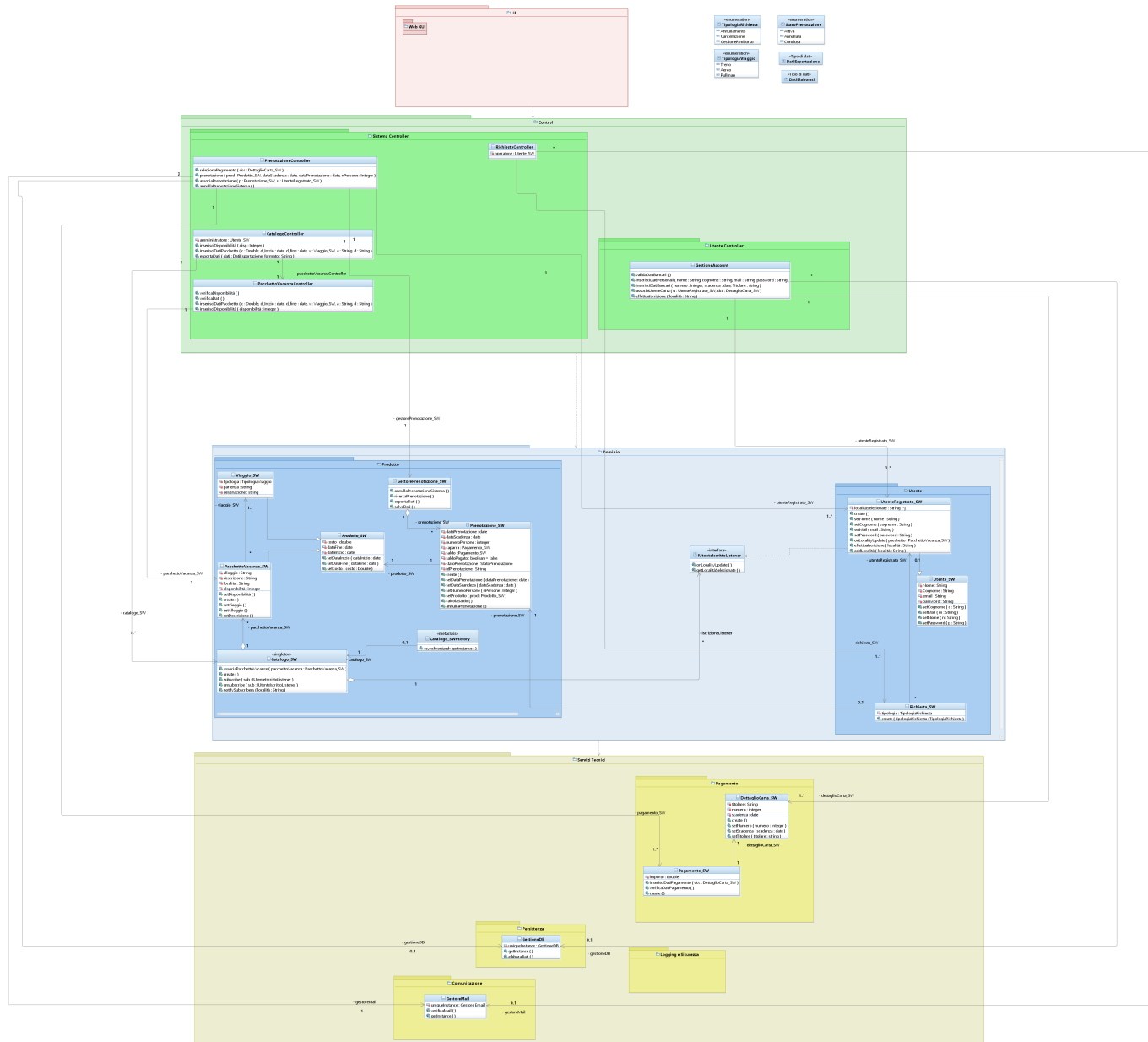


Figura 6: Sviluppato da: Volpato Mattia

L'aggiornamento del *modello delle classi software* è il risultato dell'implementazione di nuove operazioni sotto forma di Sequence Diagram. Vengono riportate solo le operazioni della nuova estensione.

Partendo dalla classe **PrenotazioneController** è stato definito il metodo `annullaOperazioneSistema()`, descritto attraverso il corrispondente SD.

Nella classe **CatalogoController** è stato definito il metodo, descritto nel SD, `esportaDati(DatiEsportazione, formato)`. L'attributo *DatiEsportazione* è definito dallo specifico datatype "tipo di dati", presente sempre nel modello delle classi software.

Nella classe **UtenteController** è stato definito il metodo `effettuaIscrizione(località)`, descritta dal SD. L'operazione è stata introdotta in quanto consente a `UtenteRegistrato` di salvare le località di interesse per poter ricevere notifiche da parte del Sistema riguardo l'aggiunta di nuovi pacchetti relativi alle località selezionate.

Nella classe **Catalogo_SW** i metodi definiti sono i seguenti:

- `subscribe(sub)`
- `unsubscribe(sub)`
- `notifySubscribers(località)`

Le operazioni, e i relativi attributi, sono sempre necessari per il processo di iscrizione da parte di `UtenteRegistrato` per ricevere notifiche da parte di Sistema. La classe factory **Catalogo_SWFactory** è stata aggiunta per gestire l'iscrizione di `UtenteRegistrato`. Crea l'unica istanza di **Catalogo_SW** e passa la reference alle classi che la richiedono.

La classe **Prenotazione_SW** definisce il metodo all'annullamento della prenotazione a carico di `UtenteRegistrato` (o `Operatore`). La classe **GestorePrenotazione_SW**, parte della aggregazione con l'aggregato **Prenotazione_SW**, definisce i metodi necessari per la gestione della prenotazione; è di conseguenza associata con **ControllerPrenotazione**. I metodi definiti da **GestorePrenotazione_SW** sono:

- `annullaPrenotazioneSistema()`
- `ricercaPrenotazione()`
- `esportaDati`
- `salvaDati`

L'interfaccia **IUtenteIscrittoListener** definisce i metodi

- `getLocalitàSelezionate()`
- `onLocalityUpdate(Pacchetto_SW p)`

La classe **UtenteRegistrato** definisce i metodi

- onLocalityUpdate(*pacchetto*)
- effettuaIscrizione(*località*)
- addLocalità(*località*)

necessari per l'aggiunta di località di interesse (addLocalità(*località*)) per effettuare l'iscrizione (effettuaIscrizione(*località*)) alla ricezione di notifiche relativi a nuovi pacchetti (onLocalityUpdate(*prodotto*)).

La classe **Richiesta_SW** definisce il metodo create(*tipologiaRichiesta*) per richiedere il rimborso dopo aver annullato la Prenotazione.

Infine la classe Singleton **GestioneDB** definisce il nuovo metodo elaboraDati().
I pattern utilizzati, descritti nello specifico sono riportati in fondo al documento.

2.2 Contratti

I contratti descritti, ricavati dalle operazioni di sistema sono:

- C01 AnnullaPrenotazioneSistema (Sviluppato da: Battaglia Walter)
- C02 EsportaDati (Sviluppato da: Volpato Mattia)
- C03 EffettuaIscrizione (Sviluppato da: Montrasio Filippo)

AnnullaPrenotazioneSistema

C01 -AnnullaPrenotazioneSistema-	
Contratto:	C01 AnnullaPrenotazioneSistema
Operazione:	annullaPrenotazioneSistema(<i>prenotazione</i>)
Riferimenti Caso d'Uso:	Annulla Prenotazione Sistema
PreCondizioni:	E in corso l'annullamento di una Prenotazione p tramite il sistema (senza passare dall'operatore)
PostCondizioni:	<ul style="list-style-type: none"> • La prenotazione p è stata annullata • La gestione del rimborso 'e stata inoltrata all'operatore • È stata inviata una mail di conferma all'utente Registrato

EsportaDati

C02 -EsportaDati-	
Contratto:	C02EsportaDati
Operazione:	esportaDati(<i>dati,formato</i>)
Riferimenti Caso d'Uso:	Esporta Dati
PreCondizioni:	E in corso l'esportazione da Storico Prenotazioni
PostCondizioni:	Viene prodotto un file del formato specificato contenente i dati richiesti nell'esportazione

EffettuaIscrizione

C03 -EffettuaIscrizione-	
Contratto:	C03EffettuaIscrizione
Operazione:	EffettuaIscrizione(<i>località, idUtente</i>)
Riferimenti Caso d'Uso:	Salva Località Interesse
PreCondizioni:	È in corso l'iscrizione al sistema di notifica novità
PostCondizioni:	Vengono associate le località scelte all'utente che sta effettuando l'iscrizione

2.3 SD

SDAnnullaPrenotazioneSistema

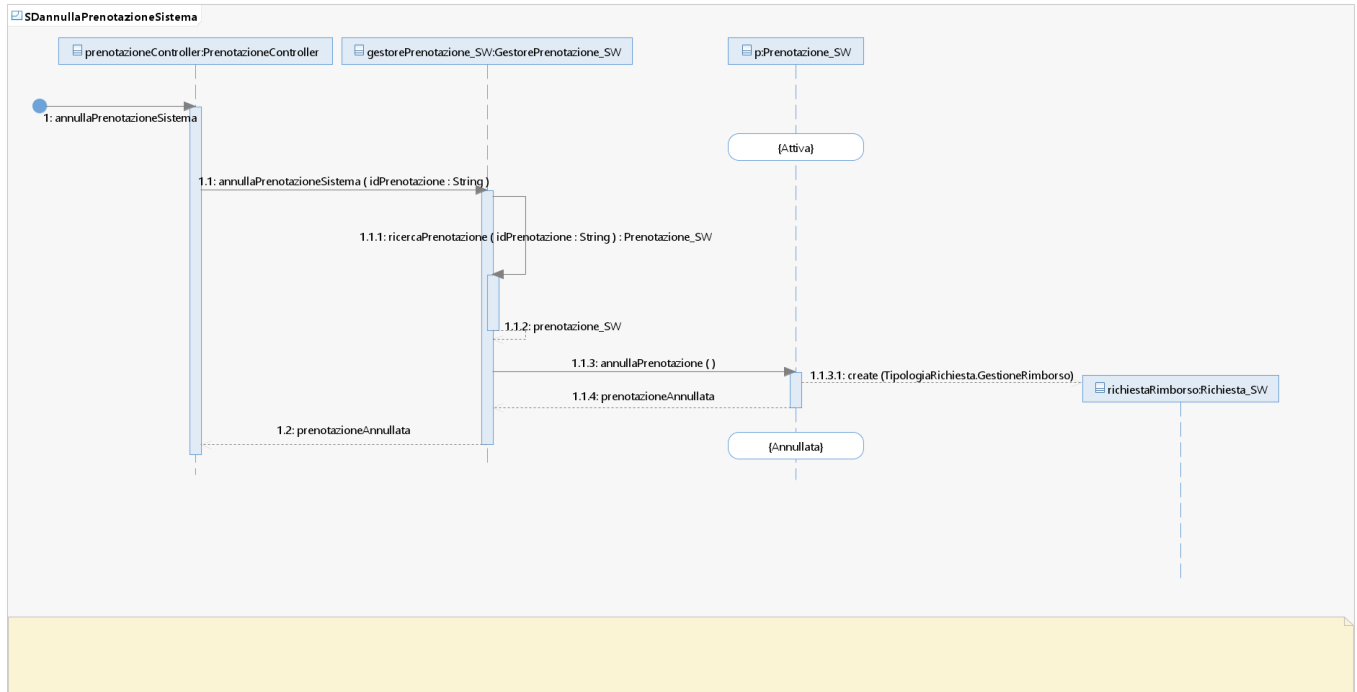


Figura 7: Sviluppato da: Battaglia Walter

SDEsportaDati

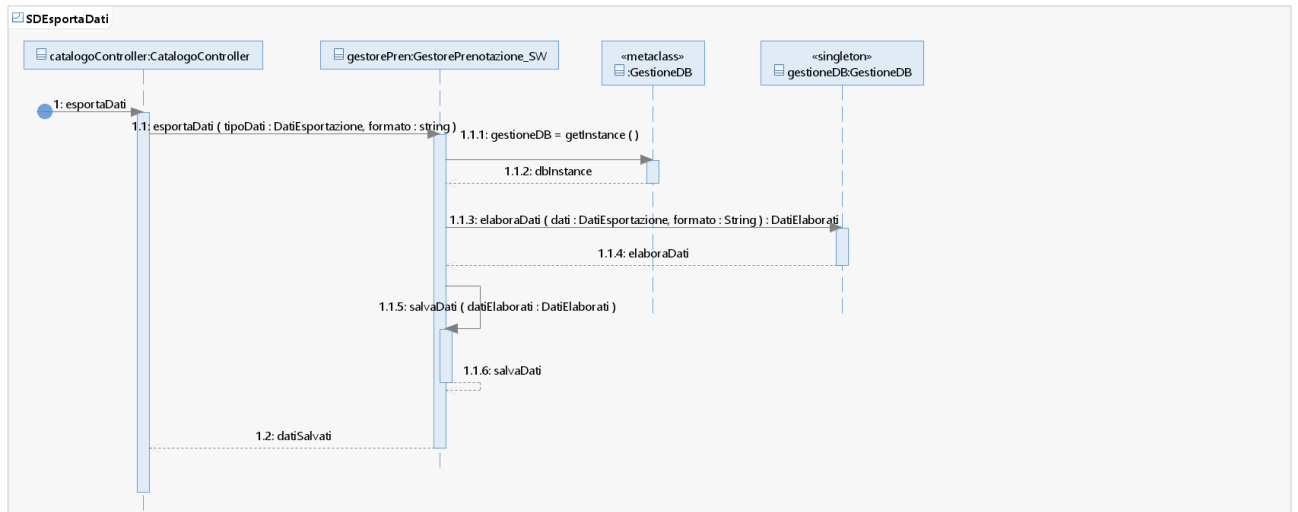


Figura 8: Sviluppato da: Volpato Mattia

SDEffettuaIscrizione

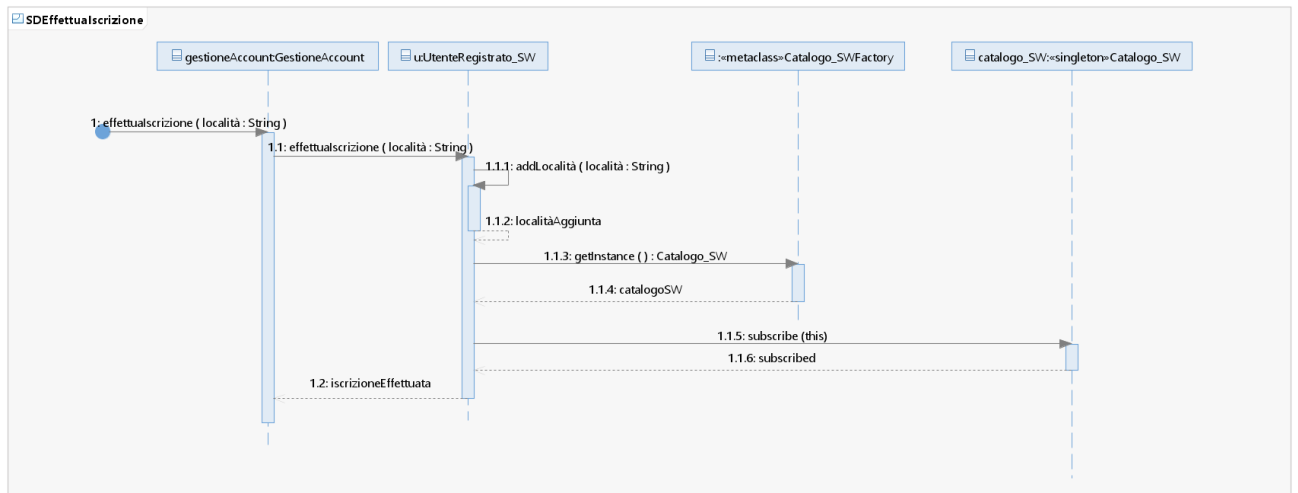


Figura 9: Sviluppato da: Montrasio Filippo

2.4 Pattern Utilizzati

PATTERN GOF

- Singleton:
 - La classe **Catalogo_SW** istanzia un solo oggetto che gestisce la raccolta dei pacchetti
 - La classe **GestioneDB** istanzia un solo oggetto per la gestione del Database di sistema;
- Abstract Factory
 - La classe **Catalogo_SWFactory** si occupa di creare l'unica istanza di **Catalogo_SW** e di passarne la reference alle classi che la richiedono
- Observer (Publish-Subscribe)
 - per gestire il processo di notificazione automatica agli utenti registrati all'atto dell'inserimento di nuove località nel catalogo, viene utilizzata la classe **Catalogo_SW** come Publisher: le classi che vorranno iscriversi per ricevere notifiche dovranno implementare l'interfaccia **IUtenteIscrittoListener**, che definisce due metodi:
 1. `getLocalitàSelezionate()`: ritorna una lista contenente le località a cui l'utente è interessato
 2. `onLocalityUpdate(Pacchetto_SW p)`: permette di gestire la notificazione all'utente dell'inserimento di una nuova località nel catalogo

Per effettuare un'iscrizione, la classe **UtenteRegistrato** ottiene la reference al **Catalogo_SW**, tramite la factory **Catalogo_SWFactory**, per poi invocare il metodo `subscribe` della classe **Catalogo_SW**.

Ogni volta che viene aggiunto un nuovo pacchetto al catalogo, **Catalogo_SW** controlla le località coinvolte nel pacchetto; successivamente, per ogni subscriber invoca il metodo `getLocalitàSelezionate()` per vedere se quest'ultimo è interessato a una o più di queste località: in caso affermativo, per ogni località aggiunta che si trova nella lista ritornata viene invocato il metodo `onLocalityUpdate(Pacchetto_SW p)`, passando come parametro il pacchetto che contiene la località di interesse, che permette di gestire la notificazione dell'evento all'utente.