**2. Progettazione e Sviluppo Iterazione 1**

In questa iterazione, sono state definite le entità principali del sistema: Cliente, Corriere e Magazzino. Inoltre, è stato progettato il caso d'uso "Registra Vendita" che permette al cliente di spedire un pacco indicando le sue caratteristiche, ottenendo il prezzo della spedizione e concludendo l'acquisto. Per quanto riguarda la progettazione delle Spedizioni, abbiamo adottato i design pattern Factory Method non astratti e Strategy. Questi pattern ci hanno permesso di gestire due tipi di spedizione: veloce e normale. Il costo e la data di consegna di ciascuna spedizione vengono calcolati in base ai parametri specificati dal cliente. I costi e le date di consegna vengono calcolati come segue:

* costo spedizione veloce = peso pacco \* 0,8
* costo spedizione normale = peso pacco \* 0,3
* data di consegna spedizione veloce = data di spedizione + valore assoluto della differenza di cap tra mittente e deposito
* data di consegna spedizione normale = data di spedizione + valore assoluto della differenza di cap tra mittente e deposito + 5 giorni

Infine la generazione del codice QR di ogni spedizione sarà simulato da un encrypt della concatenazione di codice fiscale del mittente, del destinatario, della data di spedizione e dello stato di spedizione, abbiamo dunque introdotto la classe Encryption. Questa classe contiene i metodi encrypt e decrypt, che consentono di criptare e decriptare una stringa. Nel contesto delle spedizioni, utilizziamo questi metodi per creare una stringa univoca che rappresenta il codice QR della spedizione.

**2.1. Analisi Orientata agli Oggetti**

Le entità principali del sistema di gestione delle spedizioni sono Cliente, Corriere e Magazzino. Ogni entità ha attributi e comportamenti specifici che le caratterizzano.

* Utente:
  + Attributi: username (String), password (String)
  + Comportamenti:
    - Registra vendita: L'utente può registrare una nuova vendita per conto di un cliente, inserendo le informazioni relative al pacco, come grandezza, peso, tipologia di spedizione e indirizzo di destinazione.
* Cliente:
  + Attributi: id (int), codiceFiscale (String), nome (String), cognome (String), indirizzo (String), telefono (String), comune (String), cap (int)
  + Comportamenti: Nessun comportamento specifico. Il cliente è considerato un attore che interagisce con il sistema tramite l'utente per richiedere la registrazione di una nuova vendita.
* Corriere:
  + Attributi: cap (int), nome (String), cognome (String), id (int)
  + Comportamenti:
    - Aggiorna stato: Il corriere può aggiornare lo stato di una spedizione scansionando il **QRCode associato al pacco. Questo** comportamento influenzerà il percorso della spedizione e l'avanzamento nella consegna.
* Magazzino:
  + Attributi: id (int), comune (String), cap (int), indirizzo (String) capienzaPacchi (int), pacchiInGiacenza (int)
  + Comportamenti:
    - Aggiorna giacenza: Il magazzino può aggiornare il numero di pacchi in giacenza in base alle operazioni di arrivo e partenza dei pacchi.

L'entità Spedizione rappresenta una spedizione nel sistema di gestione delle spedizioni.

* Attributi:
  + trackingCode (int): identificatore univoco della spedizione.
  + codiceQR (String): codice QR associato alla spedizione, generato criptografando una stringa che comprende il codice fiscale del mittente, il codice fiscale del destinatario, la data di spedizione e lo stato.
  + mittente (Cliente): oggetto Cliente che rappresenta il mittente della spedizione.
  + destinatario (Cliente): oggetto Cliente che rappresenta il destinatario della spedizione.
  + deposito (Magazzino): oggetto Magazzino che rappresenta il magazzino di deposito della spedizione.
  + corriere (Corriere): oggetto Corriere associato alla spedizione.
  + pesoPacco (double): peso del pacco da spedire.
  + prezzo (double): prezzo della spedizione.
  + tipoSpedizione (String): tipo di spedizione ("Veloce" o "Normale").
  + stato (String): stato corrente della spedizione.
  + dataSpedizione (Date): data di spedizione della spedizione.
  + dataConsegna (Date): data prevista di consegna della spedizione.
* Metodi:
  + creaSpedizione: metodo che crea una nuova spedizione utilizzando il Factory Method SpedizioneFactory in base al tipo di spedizione specificato. Imposta le informazioni della spedizione, come lo stato, il tracking code, il mittente, il destinatario, il deposito, il corriere, il peso, la data di spedizione, il prezzo e la data di consegna.
  + setPrezzo: metodo che imposta il prezzo della spedizione utilizzando la strategia di spedizione specificata.
  + setDataConsegna: metodo che imposta la data prevista di consegna della spedizione utilizzando la strategia di spedizione specificata.
  + updateCodiceQR: metodo che aggiorna il codice QR associato alla spedizione criptografando una stringa che comprende il codice fiscale del mittente, il codice fiscale del destinatario, la data di spedizione e lo stato.
  + updateStato: metodo che aggiorna lo stato della spedizione, aggiornando il codice QR, aggiornando la giacenza nel magazzino in base allo stato
  + setCorriere e setDeposito verranno implementati in seguito

A screenshot of a computer

Description automatically generated

**2.2. Diagramma di Sequenza registra vendita**

Nel diagramma di sequenza, l'attore principale è l'utente del sistema, che richiede la creazione di una spedizione. L'oggetto Spedizione riceve la richiesta di creazione della spedizione e avvia il processo di creazione. Viene utilizzato il metodo creaSpedizione per creare l'oggetto Spedizione e impostare le informazioni necessarie come il tracking code, il mittente, il destinatario, il peso e la data di spedizione. Il prezzo e la data di consegna verranno calcolati tramite l’interfaccia StrategiaSpedizione in cui sono definiti i metodi per calcolare il prezzo e la data di consegna in base alla strategia di spedizione specifica. Ecco una descrizione dei metodi dell'interfaccia:

* double calcolaPrezzo(Spedizione spedizione): Questo metodo riceve un oggetto di tipo Spedizione come parametro e restituisce il prezzo calcolato per la spedizione in base alla strategia di spedizione specifica. Il prezzo viene restituito come un valore di tipo double.
* Date calcolaDataConsegna(Spedizione spedizione): Questo metodo riceve un oggetto di tipo Spedizione come parametro e restituisce la data di consegna calcolata per la spedizione in base alla strategia di spedizione specifica. La data di consegna viene restituita come un oggetto di tipo Date.

L'interfaccia StrategiaSpedizione fornisce un'astrazione per le diverse strategie di spedizione implementate nella classe SpedizioneNormale e SpedizioneVeloce.

