

Università degli studi di Napoli
Parthenope

Progetto Ingegneria del Software e
Interazione Uomo Macchina

Modello di analisi

Track Me



D'ANGELO SIMONE 0124/002601
VETRANO ALESSIO 0124/002326
GALLUCCIO LUIGI 0124/002293

Prof.ssa Barra
Prof.ssa Staffa

Sommario

1. Introduzione	3
1.1 Traccia del Progetto	3
1.2 Dominio del Progetto	3
2. Sistema	3
2.1 Panoramica (obiettivi del sistema)	3
2.2 Requisiti Funzionali	4
2.2.1 Visualizzare lo storico (FR1):	4
2.2.2 Aggiornamento dello stato (FR2):	4
2.2.3 Registrazione e Log-in (FR3):	4
2.2.4 Aggiunta Mittente/Destinatario (FR4)	4
2.2.5 Invio di un collo (FR5):	4
2.2.6 Tracciamento del collo (FR6):	4
2.3 Requisiti non Funzionali	4
2.3.1 Usabilità (NFR1)	4
2.3.2 Sicurezza (NFR2)	4
2.3.3 Manutenibilità (NFR3)	4
2.4 Vincoli ("Pseudo-Requisiti")	4
2.5 Modello del sistema	5
2.5.1 Utenti del sistema (descrizione degli utenti, dei ruoli e del contesto di utilizzo)	5
2.5.2 Scenari (con descrizione degli utenti, dei ruoli e del contesto di utilizzo)	5
2.5.2.1 Scenario Corriere-ViewHistory	5
2.5.2.2 Scenario Corriere-UpdatePack	5
2.5.2.3 Scenario Cliente-SendPack	6
2.5.2.4 Scenario Cliente-TrackPack	6
2.5.3 Modello caso d'uso	7
2.5.3.1 Modello Corriere-ViewHistory	7
2.5.3.2 Modello Corriere-UpdatePack	7
2.5.3.3 Modello Cliente-SendPack	8
2.5.3.4 Modello Cliente-TrackPack	9
2.5.3.5 Diagramma casi d'uso	10
2.5.4 Modello ad oggetti	10
2.5.4.1 Identificazione degli oggetti con la tecnica di Abbott	10
2.5.4.2 Diagrammi delle classi	12
2.5.5 Modelli dinamici	17
2.5.5.1 Tracking	17
2.5.5.2 Aggiunta di un indirizzo	18
2.5.5.3 Send Pack	19
2.5.5.4 SignUp	20
2.5.5.5 Log-in	21
2.5.5.6 History	22

2.5.5.7 ChangeSite	23
2.6 Interfaccia Utente	23
2.6.1 Realizzazione del prototipo.....	23
2.6.1.1 Interfaccia – Home View.....	24
2.6.1.2 Interfaccia – Client View Choise	24
2.6.1.3 Interfaccia – Client View Send	25
2.6.1.4 Interfaccia – Client View Track	25
2.6.1.5 Interfaccia – Courier View First Choise	26
2.6.1.6 Interfaccia– Courier View SignUp	26
2.6.1.7 Interfaccia– Courier View Log-in.....	27
2.6.1.8 Interfaccia – Courier View Choise.....	27
2.6.1.9 Interfaccia– Courier View UpdateStatus	28
2.6.1.10 Interfaccia – Courier View History	28
2.6.2 Test Di Usabilità	29
2.6.2.1 Obiettivi del test	29
2.6.2.2 Metodologia usata.....	29
2.6.2.3 Sintesi delle misure.....	29
2.6.2.4 Analisi dei risultati e sintesi delle interviste	31
2.6.2 Valutazione dell’Usabilità.....	32

1. Introduzione

1.1 Traccia del Progetto

L'applicazione che si vuole sviluppare gestisce il carico e il rintracciamento di merci nel campo della logistica, utilizzando un algoritmo genetico per ottimizzare il carico dei veicoli. L'algoritmo genetico è un metodo di ottimizzazione basato su principi ispirati alla biologia evolutiva, come la selezione naturale e la riproduzione con mutazione. L'applicazione sarà utilizzata da un'azienda di trasporto che consegna merci in diverse sedi in Italia, e che ha a disposizione un numero di veicoli identificati da un codice, tipo di veicolo e capienza container. Il collo è identificato da un codice, mittente, destinatario e peso. L'applicazione deve gestire il carico dei colli nei container scegliendo il veicolo più adatto. Il corriere, inoltre, aggiorna lo stato del collo ad ogni centro di smistamento, il quale deve essere rintracciato dal destinatario mediante il suo codice

1.2 Dominio del Progetto

Il dominio del progetto descritto è la logistica, in particolare la gestione del carico e del rintracciamento delle merci utilizzando un algoritmo genetico per ottimizzare il carico dei veicoli. L'applicazione sarà utilizzata da un'azienda di trasporto per consegnare merci in diverse sedi in Italia, gestendo il carico dei colli nei container scegliendo il veicolo più adatto e tenendo traccia dello stato di ogni collo durante il trasporto.

L'applicazione è utilizzabile da due attori protagonisti, il primo è il *Corriere* che gestisce la cronologia dei pacchi spediti dall'azienda e la posizione dei vari pacchi; ed il *Cliente* che invia i pacchi a utenti inseriti direttamente da lui con, inoltre, la capacità di tracciare quest'ultimi.

2. Sistema

2.1 Panoramica (obiettivi del sistema)

L'applicazione **che si vuole sviluppare gestisce il carico e il rintracciamento di merci nel campo della logistica.**

Gli obiettivi principali raggiunti sono:

- **Ottimizzazione delle prestazioni:** utilizzare tecniche di ottimizzazione delle prestazioni per migliorare la velocità e l'accuratezza del modello di IA.
- **Affidabilità e sicurezza:** garantire che il modello di IA sia affidabile e sicuro, utilizzando generazioni di cromosomi che permettono di ottenere il risultato che si avvicina a quello ottimale con maggiore accuratezza.
- **Uniformità di codice:** Il sistema utilizzato è sviluppato interamente in Java.
- **Usabilità:** La dashboard deve avere un'interfaccia chiara e semplice da utilizzare.
- **Disponibilità:** L'interfaccia è adattata a seconda dell'utente specifico tra *Client* e *Courier*.

2.2 Requisiti Funzionali

2.2.1 Visualizzare lo storico (FR1):

Il corriere deve poter leggere lo storico di tutti i pacchi gestiti dall'azienda.

2.2.2 Aggiornamento dello stato (FR2):

Il corriere deve poter modificare lo stato di un determinato pacco.

2.2.3 Registrazione e Log-in (FR3):

Nella view dedicata al corriere è possibile registrarsi in modo da permettere al corriere appena registrato di poter effettuare l'accesso mediante l'utilizzo delle credenziali.

2.2.4 Aggiunta Mittente/Destinatario (FR4)

Nella view del cliente è possibile inserire i dati di nuovi mittenti/destinatari.

2.2.5 Invio di un collo (FR5):

Nella view del cliente è possibile inviare un collo, scegliendo il mittente, destinatario e il peso del collo.

2.2.6 Tracciamento del collo (FR6)

Nella view del cliente è possibile visualizzare l'ultima sede in cui si trova il collo cercato.

2.3 Requisiti non Funzionali

2.3.1 Usabilità (NFR1)

Le varie schermate devono avere un'interfaccia chiara e facilmente utilizzabile.

2.3.2 Sicurezza (NFR2)

Sicurezza sul Register del corriere (Password di almeno otto caratteri).

2.3.3 Manutenibilità (NFR3)

Il programma è gestito in modo da avere possibilità di implementazioni future mediante l'utilizzo di Design Pattern e rispettando i principi SOLID.

2.4 Vincoli ("Pseudo-Requisiti")

I vincoli, anche chiamati pseudo-requisiti, in ingegneria del software sono restrizioni imposte al progetto che non riguardano direttamente le funzionalità richieste dall'utente, ma influiscono sulla progettazione e sullo sviluppo del software.

Il vincolo principale del nostro progetto consiste nella sua esclusiva esecuzione su sistemi operativi MacOS e Linux. In quanto all'esecuzione verrà utilizzato uno *bashScript* che su Windows non è disponibile.

Un vincolo ulteriore in fase di progettazione è dato dall'utilizzo di un singolo linguaggio di programmazione che nel nostro caso risulta essere Java associato all'utilizzo di JavaFx e SceneBuilder per l'implementazione dell'interfaccia.

2.5 Modello del sistema

2.5.1 Utenti del sistema (descrizione degli utenti, dei ruoli e del contesto di utilizzo)

Gli utenti principali del sistema sono Cliente e Corriere.

Cliente può inviare il pacco inserendo gli specifici mittenti e destinatari, inoltre, può tracciare il pacco conoscendo ed inserendo il codice identificativo di quest'ultimo.

Corriere può aggiornare la posizione di uno specifico pacco o impostarlo come "Consegnato", inoltre, può visualizzare la cronologia di tutti i pacchi spediti mediante il software.

2.5.2 Scenari (con descrizione degli utenti, dei ruoli e del contesto di utilizzo)

Gli scenari in ingegneria del software sono un insieme di passi descrittivi che descrivono come gli utenti interagiranno con il sistema. Essi descrivono le azioni che l'utente compie, le risposte del sistema e le condizioni sottostanti. Gli scenari sono spesso utilizzati per descrivere i casi d'uso, ovvero le funzionalità del sistema che sono interessanti per gli utenti.

2.5.2.1 Scenario Corriere-ViewHistory

ATTORI PRINCIPALI: Ernesto Esposito: Corriere

Flusso di eventi:

1. Ernesto Esposito per prima cosa definisce il suo ruolo di Corriere cliccando il bottone "*Courier*" che esprime tale ruolo.
2. Ernesto Esposito se è la prima volta che utilizza l'app Stand-alone effettua il Register cliccando "*Sign up*" e successivamente nella nuova schermata aperta riempirà i campi necessari per la registrazione.
3. Dopo aver effettuato la registrazione Ernesto Esposito si troverà davanti alla schermata di Login dove inserirà le proprie E-mail e Password.
4. Ernesto Esposito incaricato di veder lo storico dei pacchi cliccherà su "History". Successivamente a tale azione verrà presentata un'interfaccia dove risulterà presente la cronologia dei vari pacchi con specifiche su veicolo in cui sono contenute peso, posizione attuale e l'orario di avvenuto aggiornamento.

2.5.2.2 Scenario Corriere-UpdatePack

ATTORI PRINCIPALI: Angelo Ciaramella: Corriere

Flusso di eventi:

1. Angelo Ciaramella per prima cosa definisce il suo ruolo di Corriere cliccando il bottone "*Courier*" che esprime tale ruolo.
2. Angelo Ciaramella se è la prima volta che utilizza l'app StandAlone effettua il Register cliccando "*Sign up*" e successivamente nella nuova schermata aperta riempirà i campi necessari per la registrazione.

3. Dopo aver effettuato la registrazione Angelo Ciaramella si troverà davanti alla schermata di Login dove inserirà le proprie E-mail e Password.
4. Angelo Ciaramella incaricato di aggiornare la posizione attuale del pacco in base alla sede della propria compagnia cliccherà sul bottone *"Update Status"*.
5. Angelo Ciaramella inserirà il codice del pacco nello specifico TextField, dove avrà due possibilità:
 - a. Scegliere la sede specifica dove il pacco risiede attualmente utilizzando la ChoiseBox e inserendo l'opzione effettiva, associando al pacco la posizione specifica.
 - b. Cliccando sul bottone *"Consegnato"* al pacco verrà associata la via del destinatario come posizione associato alla data di quel momento.

2.5.2.3 Scenario Cliente-SendPack

ATTORI PRINCIPALI: Giulio Giunta: Cliente

Flusso di eventi:

1. Giulio Giunta per prima cosa definisce il suo ruolo di Cliente cliccando il bottone *"Client"* che esprime tale ruolo.
2. Giulio Giunta successivamente per spedire un pacco cliccherà sul bottone *"Send"*
3. Al nostro cliente verrà presentata una nuova schermata dove saranno presenti due ChoiseBox, nel caso in cui nessun vecchio mittente o destinatario precedentemente inserito è utile per la spedizione che Giulio Giunta deve effettuare verrà presentata la possibilità di aggiungere i vari dati del mittente e del destinatario attraverso la voce: *" + Aggiungi Indirizzo"*.
4. Verrà presentata una nuova schermata dove Giulio Giunta sarà incaricato di compilare i vari capi per aggiungere un utente da utilizzare come mittente o destinatario.
5. Giulio Giunta dopo che saranno stati inseriti almeno 2 utenti da utilizzare come mittenti e destinatari dovrà indicare nell'opportuno TextField il peso in kg del pacco da inviare.
6. Dopo aver completato i campi necessari descritti precedentemente Giulio Giunta per inviare ufficialmente il pacco cliccherà *"Spedisci collo"*. Verrà presentata in una DialogView la conferma di invio del pacco con relativo codice identificativo di quest'ultimo.

2.5.2.4 Scenario Cliente-TrackPack

ATTORI PRINCIPALI: Vittorio Luperto: Cliente

Flusso di eventi:

1. Vittorio Luperto per prima cosa definisce il suo ruolo di Cliente cliccando il bottone *"Client"* che esprime tale ruolo.
2. Vittorio Luperto successivamente per tracciare un pacco cliccherà sul bottone *"Track"*.
3. Il cliente inserirà il codice identificativo del pacco che vuole tracciare nello specifico box.
4. Cliccando *"Track"* se il pacco è stato effettivamente spedito verrà visualizzato a schermo la posizione di quel pacco specifico.

2.5.3 Modello caso d'uso

I modelli casi d'uso sono una rappresentazione grafica dei comportamenti e delle interazioni tra sistemi, utenti e altri elementi esterni in un sistema software. Descrivono come gli utenti interagiscono con il sistema per ottenere un determinato risultato o per risolvere un problema. Aiutano a identificare i requisiti e a pianificare il design del software.

Gli attori nel nostro progetto sono rappresentati dal **Cliente**, dai **Corrieri** e dal **Filesystem**. Il cliente utilizza l'applicazione per spedire e tracciare un pacco. Il corriere, invece, si occupa di tenere traccia di tutte le spedizioni dell'azienda e, inoltre, ha la possibilità di aggiornare lo stato specifico di un collo. Infine, il filesystem si occupa della gestione dei file.

2.5.3.1 Modello Corriere-ViewHistory

Nome caso d'uso:	Corriere-ViewHistory
Attori partecipanti	Iniziato dal Corriere Comunica con FileSystem
Flusso eventi	<ol style="list-style-type: none">1. Il corriere incaricato di veder lo storico dei pacchi cliccherà su "History". Successivamente a tale azione verrà presentata un'interfaccia ,attraverso la comunicazione con il FileSystem leggendo nel file packages, dove risulterà presente la cronologia dei vari pacchi con specifiche su veicolo in cui sono contenute peso, posizione attuale e l'orario di avvenuto aggiornamento..
Condizioni di entrata	Il corriere è loggato.
Condizioni di uscita	Il corriere riceverà dal FileSystem le informazioni relative allo storico.
Requisiti di qualità	Il login del Corriere è accettato entro 15 secondi. Lo storico richiesto arriva non più di 15 secondi dopo che è stata inviata dal FileSystem.

2.5.3.2 Modello Corriere-UpdatePack

Nome caso d'uso:	Corriere-UpdatePack
Attori partecipanti	Iniziato dal Corriere Comunica con FileSystem
Flusso di eventi	<ol style="list-style-type: none">1. Il corriere incaricato di aggiornare la posizione attuale del pacco in base alla sede della propria compagnia cliccherà sul bottone "Update Status".2. Il corriere inserirà il codice del pacco nello specifico TextField, dove avrà due possibilità:<ol style="list-style-type: none">a. Scegliere la sede specifica dalla lista delle sedi presenti nei file preesistenti attraverso la

	<p>comunicazione con il Filesystem. Utilizzando la ChoiseBox inserirà l'opzione effettiva, associando al pacco la posizione specifica.</p> <p>b. Cliccando sul bottone "Consegnato" al pacco verrà associata la via del destinatario come posizione associato alla data di quel momento comunicando con il filesystem modificando il contenuto del file packages.</p>
Condizioni di entrata	Il corriere è loggato.
Condizioni di uscita	Il corriere comunicherà con il filesystem e aggiornerà la sede, OR il corriere comunicherà con il filesystem e imposterà lo stato come consegnato.
Requisiti di qualità	La posizione del pacco deve essere aggiornato entro 15 secondi.

2.5.3.3 Modello Cliente-SendPack

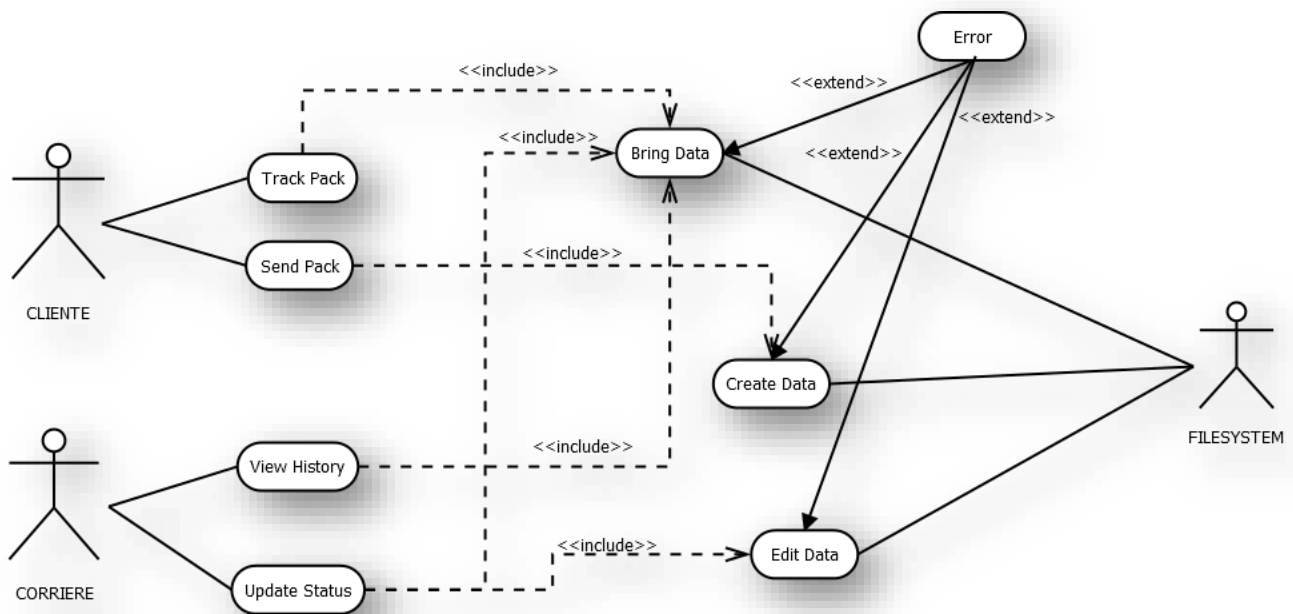
Nome caso d'uso:	Cliente-SendPack
Attori partecipanti	Iniziato dal Cliente Comunica con il FileSystem
Flusso di eventi	<ol style="list-style-type: none"> 1. Il cliente per prima cosa definisce il suo ruolo di Cliente cliccando il bottone "<i>Client</i>" che esprime tale ruolo. 2. Il cliente successivamente per spedire un pacco cliccherà sul bottone "<i>Send</i>" 3. Al nostro cliente verrà presentata una nuova schermata dove saranno presenti due ChoiseBox, comunicanti con il FileSystem in modo da capire se esistono dei clienti preesistenti. Nel caso in cui nessun vecchio mittente o destinatario precedentemente inserito è utile per la spedizione che il cliente deve effettuare verrà presentata la possibilità di aggiungere i vari dati del mittente e del destinatario attraverso la voce: "<i>+ Aggiungi Indirizzo</i>". 4. Verrà presentata una nuova schermata dove il cliente sarà incaricato di compilare i vari capi per aggiungere un utente da utilizzare come mittente o destinatario. 5. Il cliente dopo che saranno stati inseriti almeno 2 utenti da utilizzare come mittenti e destinatari dati dal FileSystem, dovrà indicare nell'opportuno TextField il peso in kg del pacco da inviare. 6. Dopo aver completato i campi necessari descritti precedentemente dal cliente per inviare ufficialmente il pacco cliccherà "<i>Spedisci collo</i>". Verrà presentata in una DialogView la conferma di invio del pacco con relativo codice identificativo di quest'ultimo.
Condizioni di entrata	Client deve possedere credenziali di mittente e destinatario a cui vuol inviare il pacco

Condizioni di uscita	Il Client ha inviato il pacco tra il mittente e il destinatario selezionato OR Il Client ha ricevuto una spiegazione indicante perché l'invio del pacco non è stata eseguito.
Requisiti di qualità	L'inserimento di mittente/destinatario deve avvenire in massimo 15 secondi. L'invio del pacco deve avvenire in massimo 15 secondi.

2.5.3.4 Modello Cliente-TrackPack

Nome caso d'uso:	Cliente-TrackPack
Attori partecipanti	Iniziato dal Cliente Comunica con il FileSystem
Flusso di eventi	<ol style="list-style-type: none"> 1. Il cliente per prima cosa definisce il suo ruolo di Cliente cliccando il bottone "<i>Client</i>" che esprime tale ruolo. 2. Il cliente successivamente per tracciare un pacco cliccherà sul bottone "<i>Track</i>". 3. Il cliente inserirà il codice identificativo del pacco che vuole tracciare nello specifico box. 4. Cliccando "<i>Track</i>" se il pacco è stato effettivamente spedito verrà visualizzato a schermo la posizione di quel pacco specifico tramite la comunicazione con il FileSystem e la lettura del file packages.
Condizioni di entrata	Client deve possedere il codice identificativo del pacco che si vuol tracciare.
Condizioni di uscita	Il Client ottiene localizzazione del pacco specifico OR Client avvisato che il codice identificativo del pacco inserito non è corretto
Requisiti di qualità	La posizione del pacco deve esser notificata al Cliente in massimo 15 secondi.

2.5.3.5 Diagramma casi d'uso



2.5.4 Modello ad oggetti

I modelli ad oggetti sono una rappresentazione grafica dei concetti e delle relazioni tra gli oggetti in un sistema software. Gli oggetti sono elementi che rappresentano oggetti reali o concetti astratti e le loro proprietà e comportamenti. I modelli ad oggetti vengono utilizzati per descrivere la struttura e il comportamento del sistema e aiutano a identificare i requisiti e a pianificare il design del software.

2.5.4.1 Identificazione degli oggetti con la tecnica di Abbott

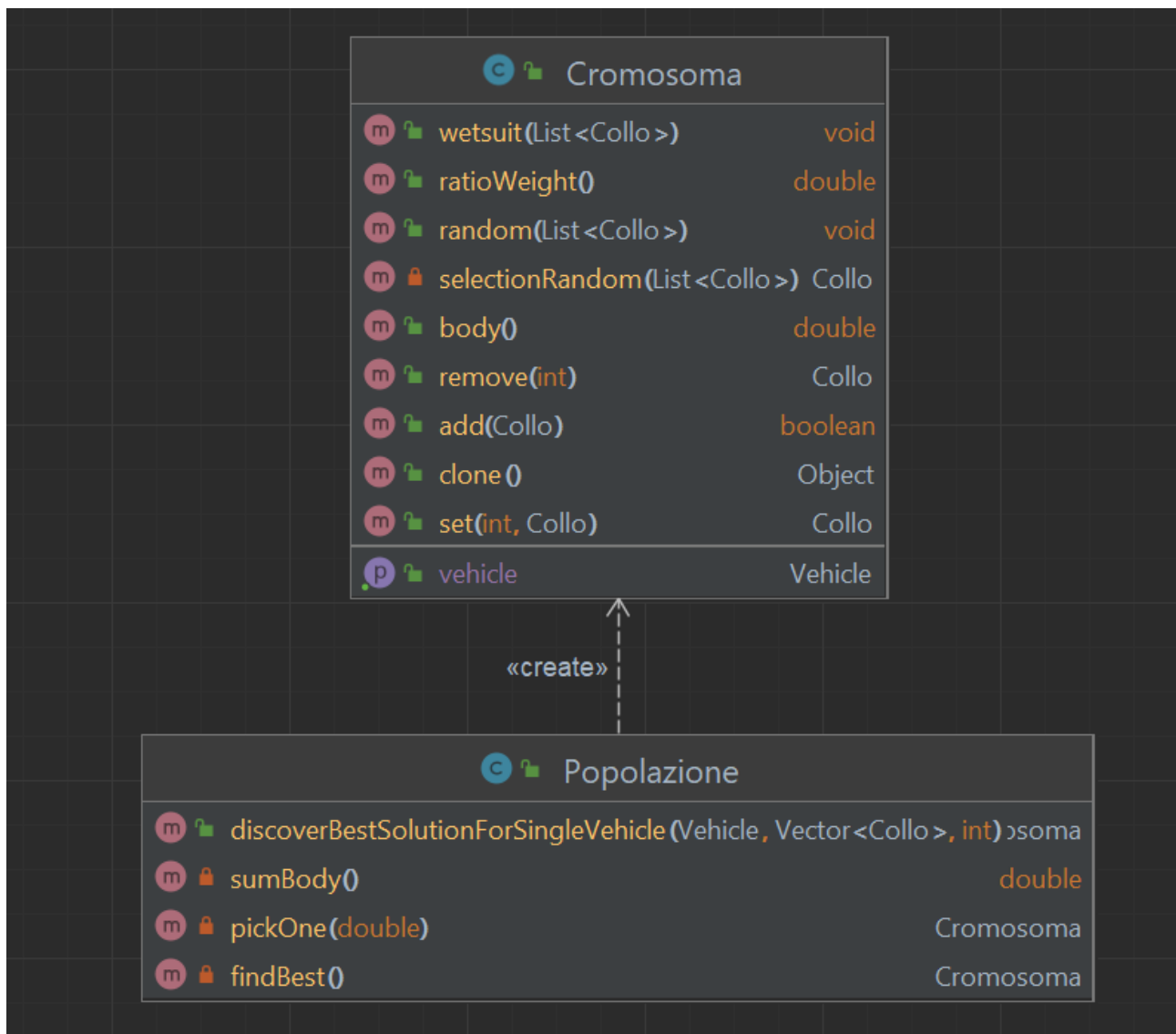
ENTITY	
Collo	Pacco inviato tra mittente e destinatario.
Client	Invio e tracciamento di un collo.
Courier	Cambio Stato e visione dello storico.
Vehicle	Veicolo da riempire in base all'algoritmo genetico.
Site	Sede assegnata dal corriere in base agli spostamenti.

CONTROLLER	
ChoiseClientController	Scelta tra la spedizione di un collo o il tracking di un collo.
ChoiseController	Scelta tra l'utente cliente o l'utente corriere.
ChoiseCourierController	Scelta tra l'aggiornamento dello stato di una spedizione o visualizzare lo storico aziendale.
ClientController	Gestione dell'aggiunta di un nuovo mittente/destinatario.
CourierController	Controller che si occupa dell'aggiornamento dello stato di una spedizione con possibilità di avvenuta consegna.
CourierMainController	Scelta tra effettuare il log-in o la registrazione.
DetailController	Controller che definisce il contenuto dell'history.
HistoryController	Controller che mostra lo storico dei pacchi.
LoginController	Controller che si occupa del login di un corriere precedentemente registrato.
MainController	Controller tasto 'help'.
SendController	Controller che si occupa della spedizione di un collo.
SignupController	Controller che si occupa della registrazione di un corriere.
TrackingController	Controller che gestisce il tracking del collo.

SERVICE	
PackageService	Intermediario tra classe Collo e file packages.csv, gestendo i dati relativi ai colli.
CourierService	Intermediario tra classe Corriere e file courier.csv, gestendo i dati relativi ai corrieri.
ClientService	Intermediario tra classe Client e file client.csv, gestendo i dati relativi ai clienti (mittenti/destinatari passati).
VehicleService	Intermediario tra classe Vehicle e file vehicles.csv, gestendo i dati relativi ai veicoli (presenti già all'interno del sistema).
SedeService	Intermediario tra classe Sede e file sedi.csv, gestendo i dati relativi alle sedi (presenti già all'interno del sistema).

2.5.4.2 Diagrammi delle classi

Il diagramma delle classi rappresentato seguentemente descrive la gestione dell'algoritmo genetico dato dalle classi Popolazione e Cromosoma.



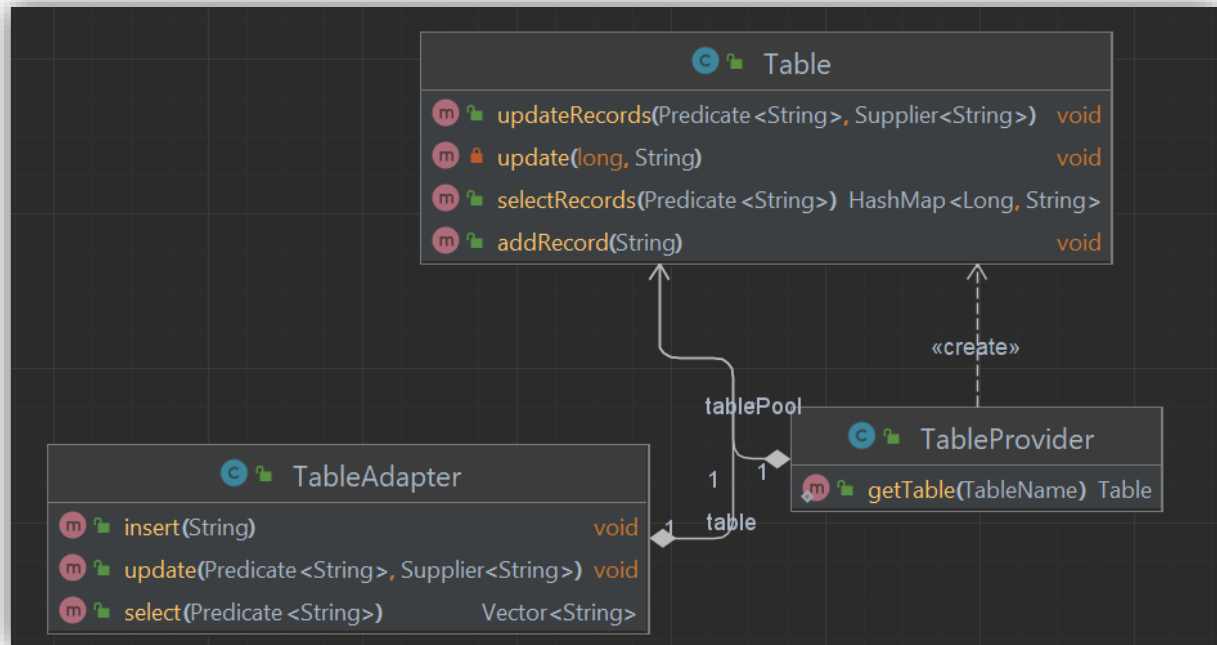
I file CSV (Comma Separated Value) sono utilizzati per conservare i dati dell'applicazione in modo permanente nel PC. I file sono associati a delle classi che contengono delle istanze corrispondenti ai valori dei campi del file. La classe TABLE si occupa di gestire la scrittura sul file utilizzando i metodi per inserire, aggiornare o eliminare elementi nel file. Utilizzando questa tipologia di classe facciamo riferimento al primo Design Pattern che abbiamo utilizzato nel nostro progetto di cui tratteremo seguentemente. Attraverso il metodo update della classe TABLE riusciamo a scrivere nei file a cui facciamo riferimento.

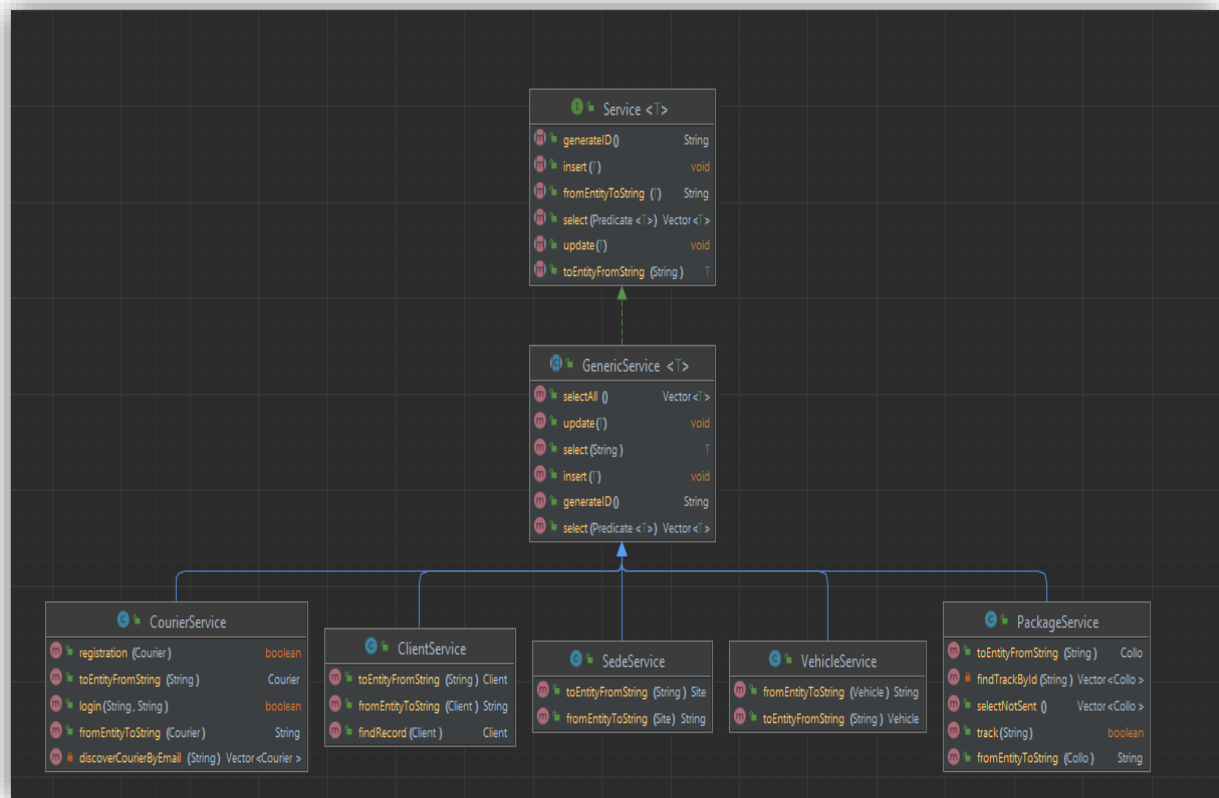
Oltre l'utilizzo di TABLE, la classe SERVICE rappresenta un'interfaccia di un servizio generico chiamato "Service" in Java. L'interfaccia estende l'entità generica T che estende la classe ENTITY. L'interfaccia definisce alcuni metodi che devono essere implementati dalla classe che implementa

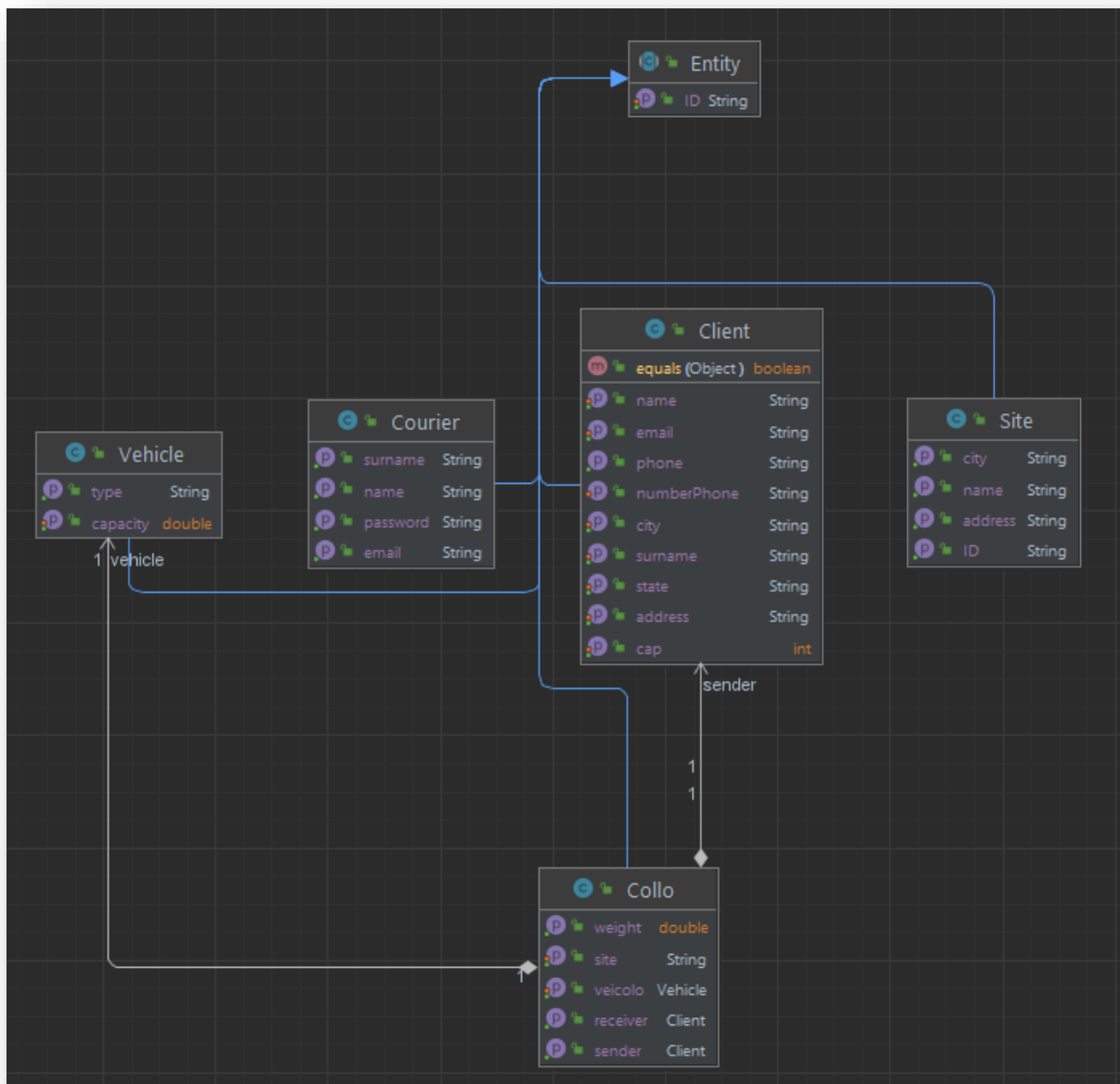
l'interfaccia.

L'utilizzo di una interfaccia generica consente di creare una classe di servizio per ogni tipo di entità senza dover ripetere il codice per ogni tipo di classe di entità.

Di seguito i diagrammi UML per la gestione dei file, delle entità e dei servizi:

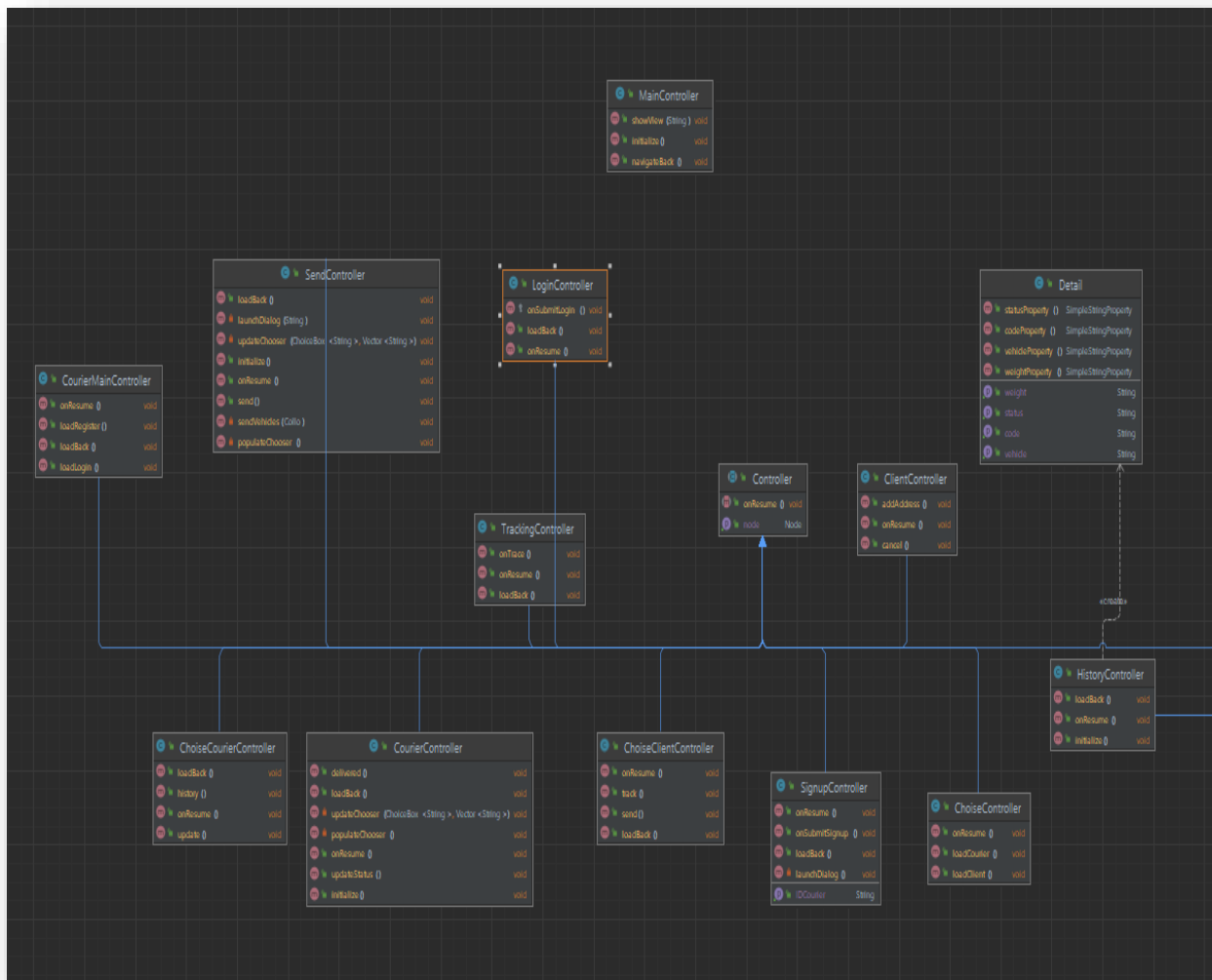






Tramite l'utilizzo di Scene Builder è possibile gestire, attraverso l'interfaccia grafica del tool, i file *.xml che abbiamo utilizzato per costruire l'intera interfaccia grafica. Ogni file comunicherà attraverso il corrispettivo Controller che gestirà le azioni e le funzioni scaturite dall'interazione con l'utente.

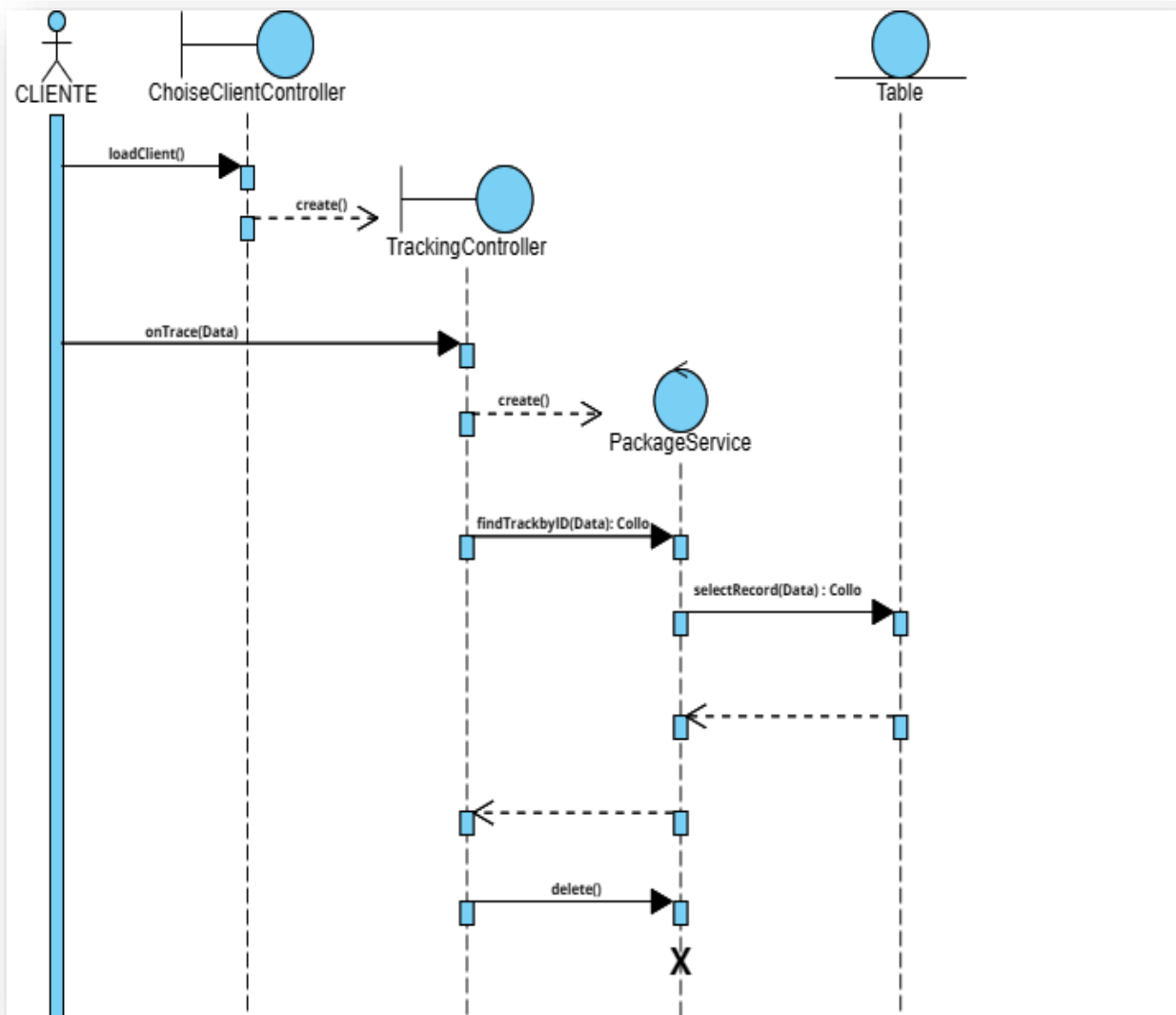
Di seguito il diagramma UML delle classi Controller:



2.5.5 Modelli dinamici

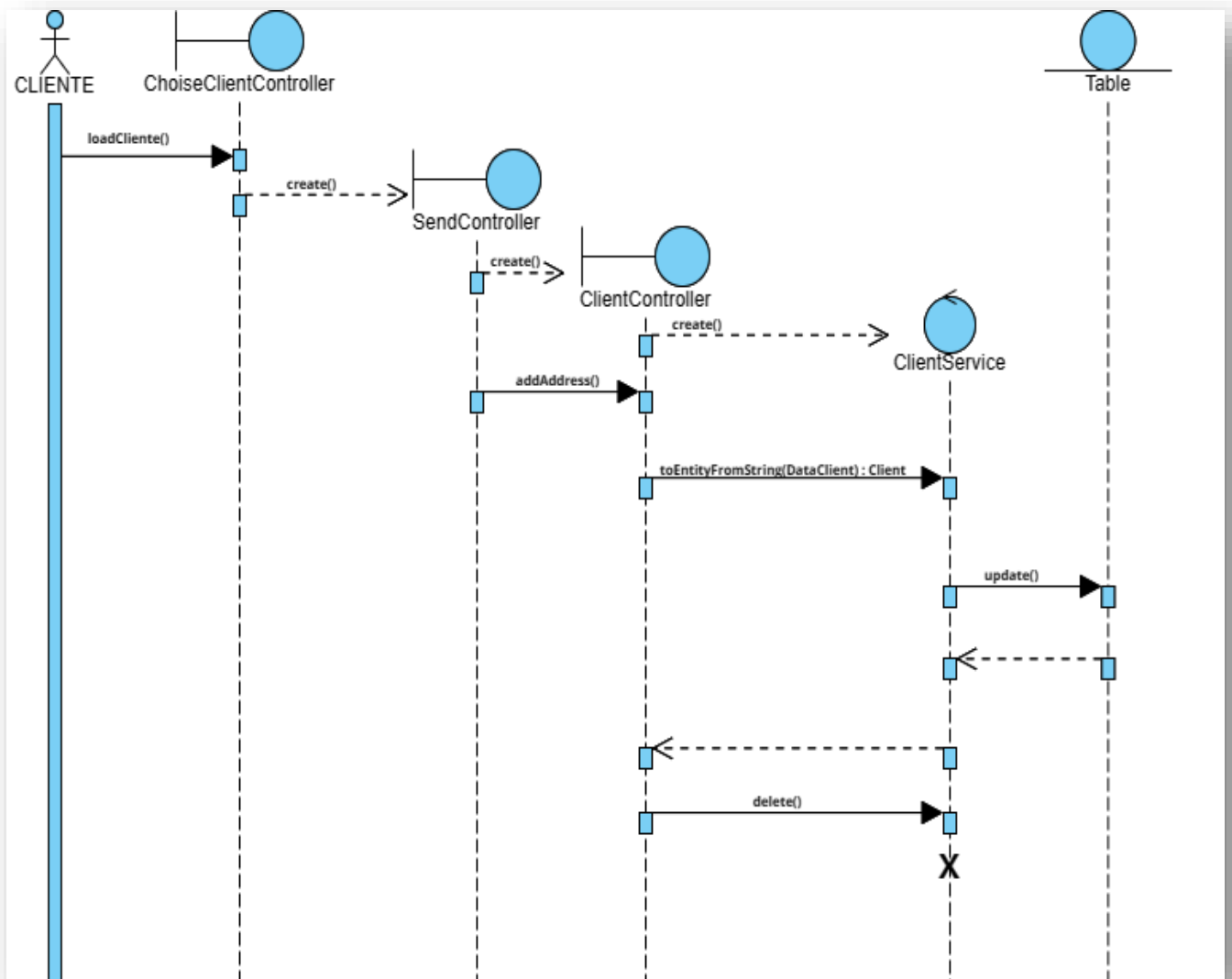
2.5.5.1 Tracking

Diagramma per tracciamento delle spedizioni.



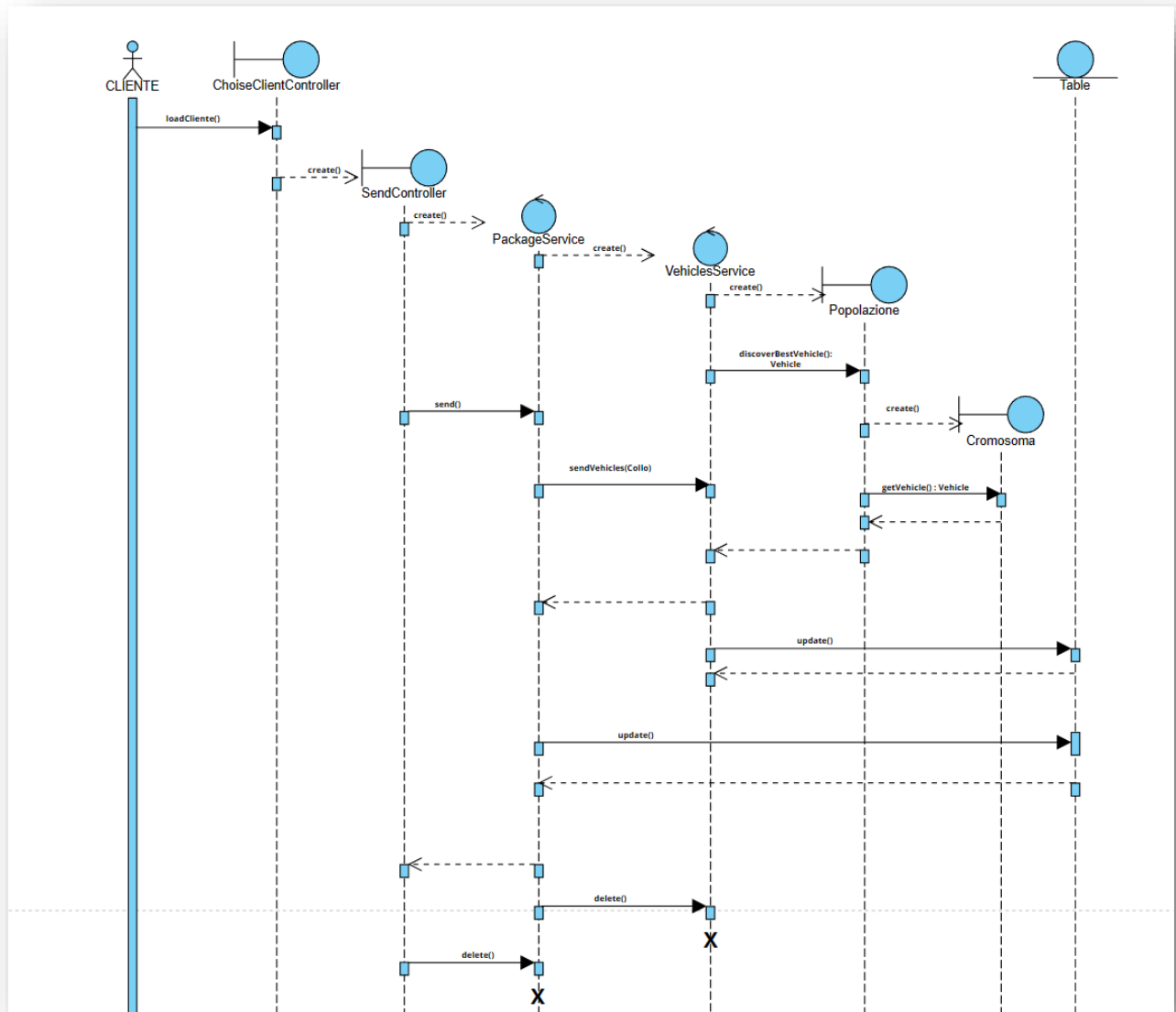
2.5.5.2 Aggiunta di un indirizzo

Diagramma per l'aggiunta di un nuovo indirizzo



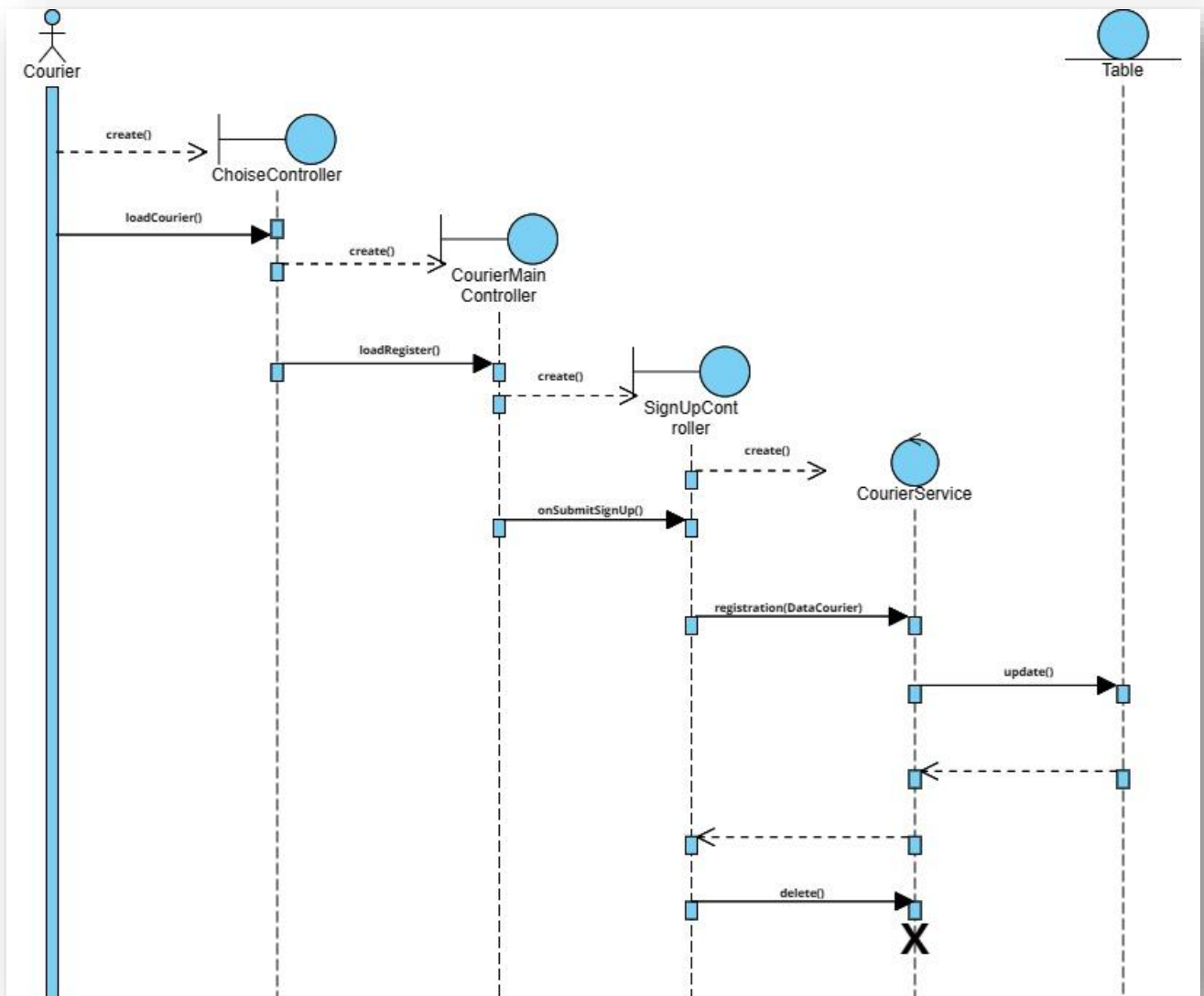
2.5.5.3 Send Pack

Diagramma per la spedizione di un nuovo collo.



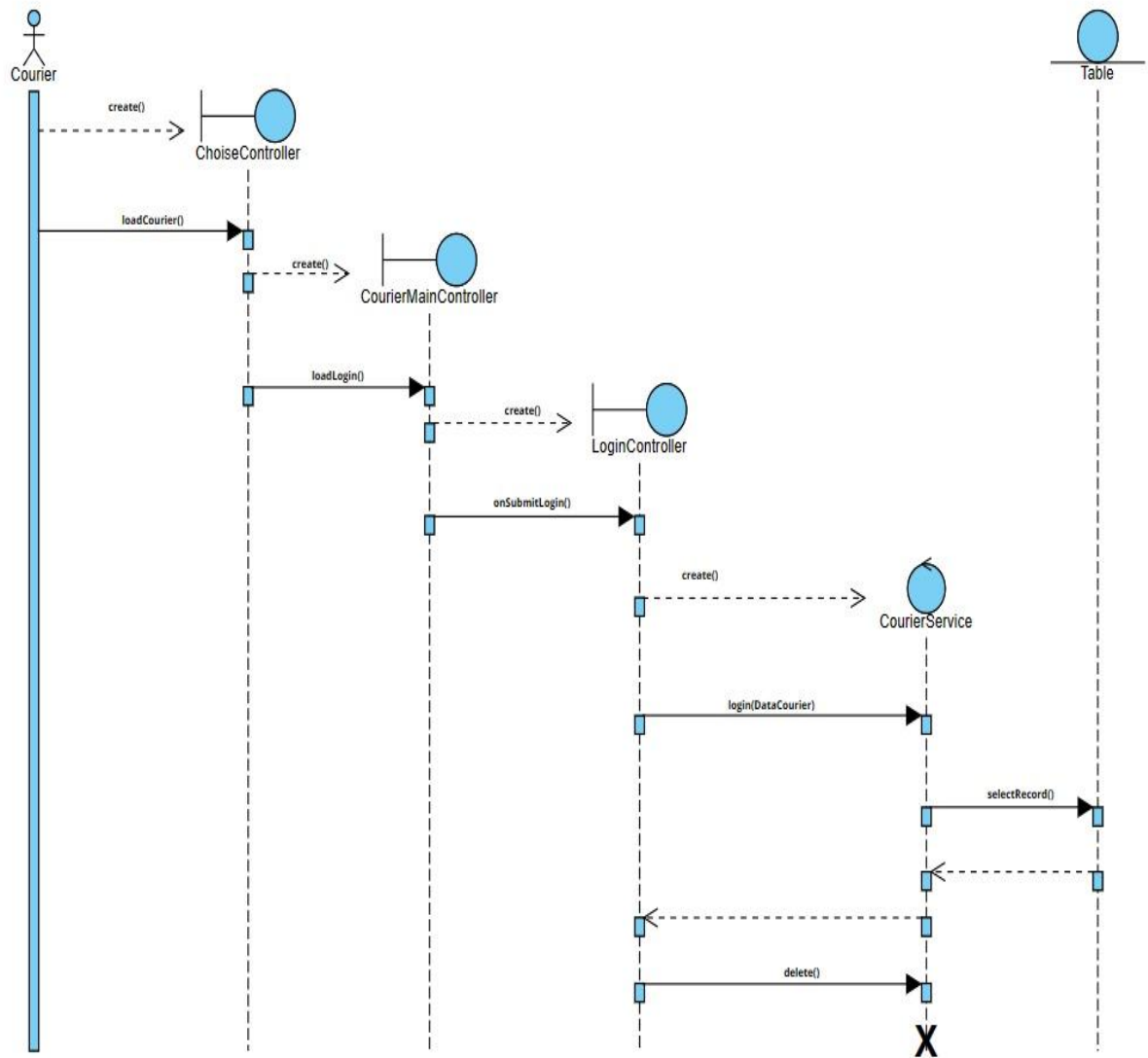
2.5.5.4 SignUp

Diagramma per la registrazione di un nuovo corriere.



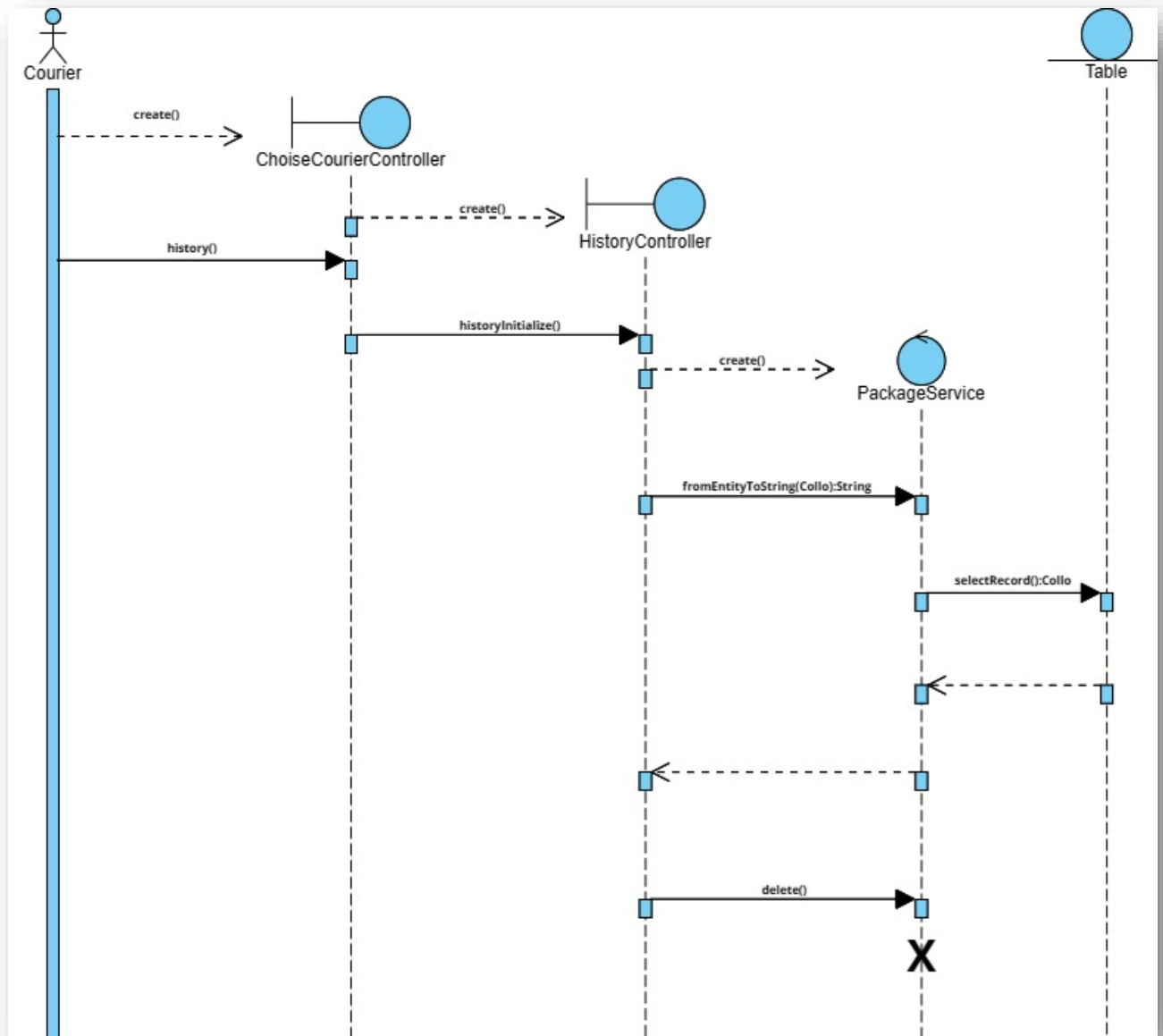
2.5.5.5 Log-in

Diagramma per effettuare il log-in

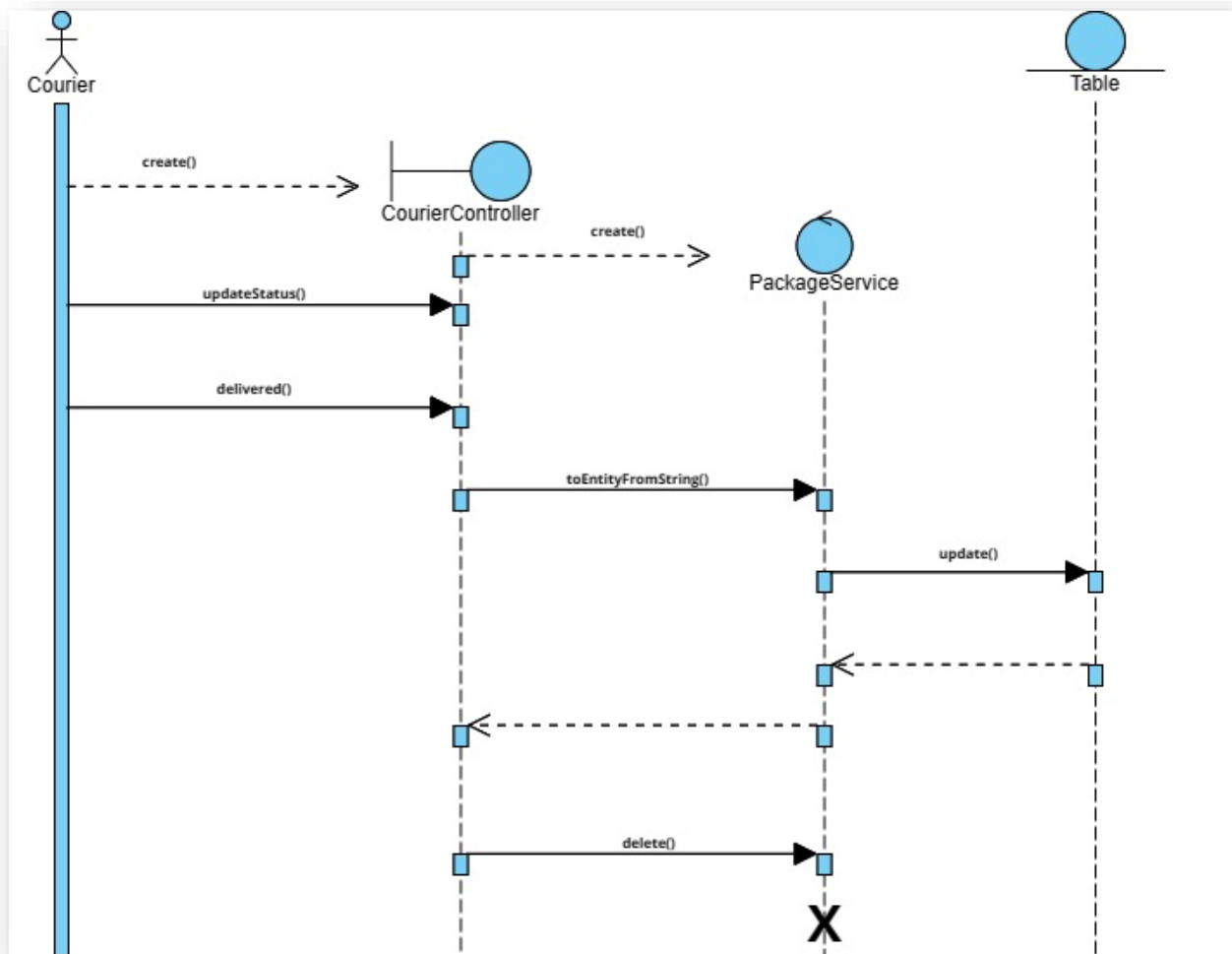


2.5.5.6 History

Diagramma per la visione dello storico



2.5.5.7 ChangeSite

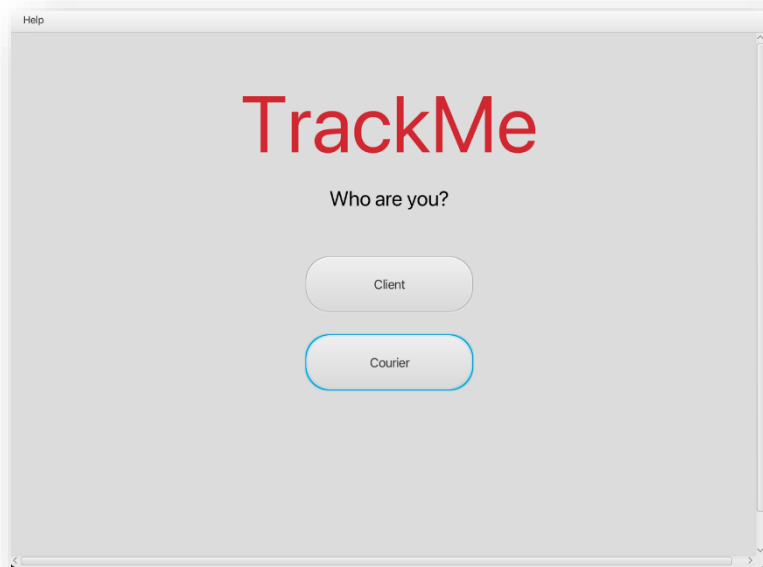


2.6 Interfaccia Utente

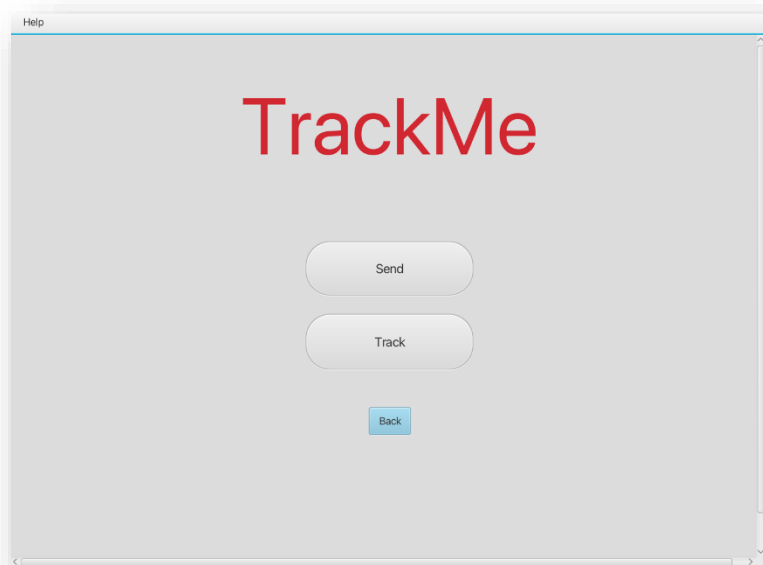
2.6.1 Realizzazione del prototipo

- **Scopo:** L'interfaccia dell'applicazione è semplice e intuitiva, in modo da poter essere utilizzata da utenti di tutte le età e di qualsiasi livello di competenza tecnologica. La semplicità è stata ottenuta attraverso la riduzione del numero di elementi visivi e la semplificazione dei flussi di navigazione. L'intuitività è stata invece ottenuta attraverso la creazione di un layout chiaro e logico, che consente all'utente di comprendere rapidamente come utilizzare il sistema.
- **Modalità d'uso:** La modalità d'uso dell'applicazione è "interattiva", infatti, per l'utente è possibile usare l'applicazione e navigare fra le schermate utilizzando i pulsanti e utilizzare i campi di testo per inserire dei dati.
- **Completezza funzionale:** L'applicazione è fortemente orizzontale, infatti esso permette all'utente di valutare l'interfaccia nella sua interezza e gli permette di comprendere l'interazione con essa.
- **Tipo di prototipo:** applicazione Standalone sviluppata in JAVA.

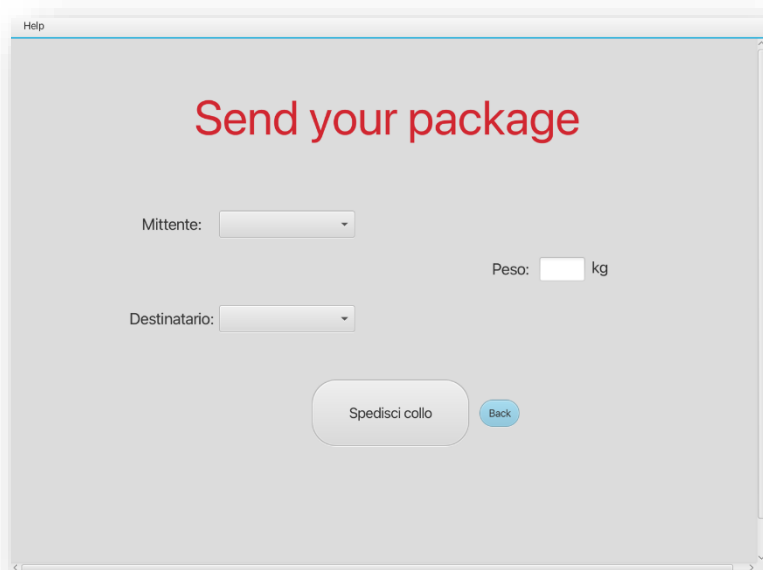
2.6.1.1 Interfaccia – Home View



2.6.1.2 Interfaccia – Client View Choise

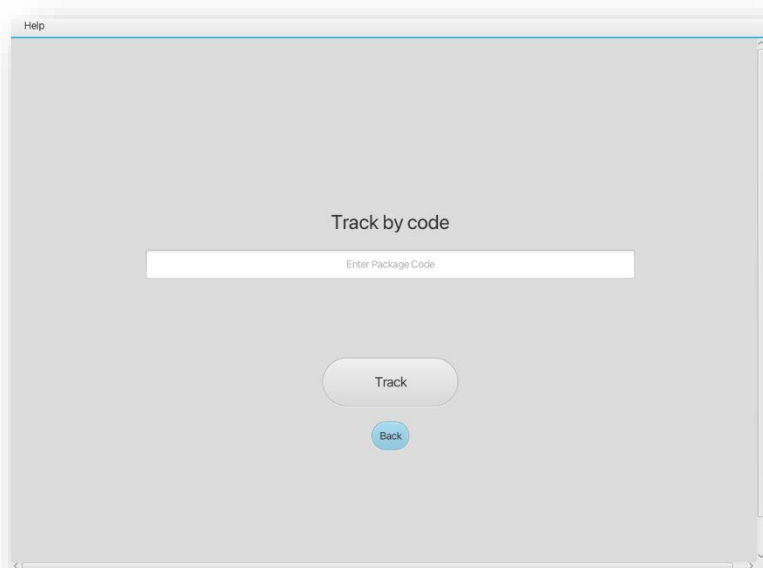


2.6.1.3 Interfaccia – Client View Send



A screenshot of a web application window titled "Send your package". The window has a light gray background and a "Help" link in the top left corner. The main heading "Send your package" is in red. Below the heading, there are two dropdown menus: "Mittente:" and "Destinatario:". To the right of the "Mittente:" dropdown is a text input field labeled "Peso:" followed by "kg". At the bottom, there are two buttons: a large, rounded "Spedisci collo" button and a smaller, circular "Back" button.

2.6.1.4 Interfaccia – Client View Track

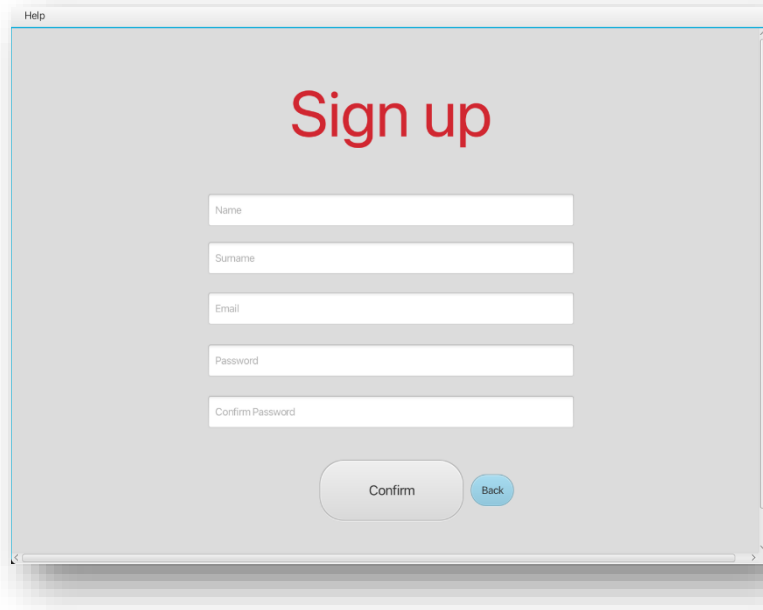


A screenshot of a web application window titled "Track by code". The window has a light gray background and a "Help" link in the top left corner. The main heading "Track by code" is in black. Below the heading is a text input field with the placeholder text "Enter Package Code". At the bottom, there are two buttons: a large, rounded "Track" button and a smaller, circular "Back" button.

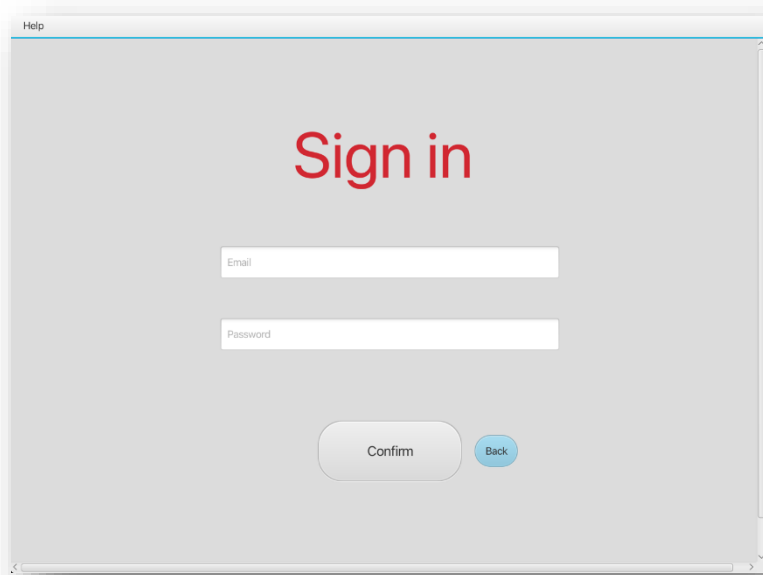
2.6.1.5 Interfaccia – Courier View First Choise



2.6.1.6 Interfaccia– Courier View SignUp

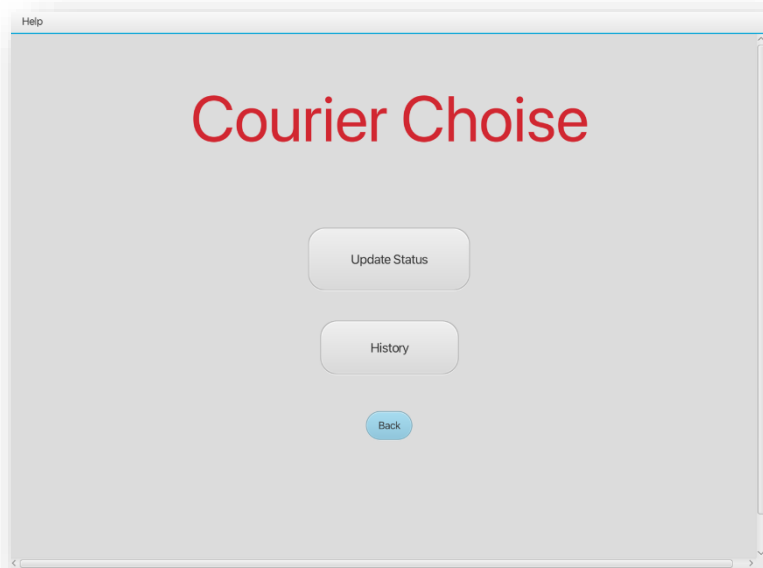


2.6.1.7 Interfaccia– Courier View Log-in



A screenshot of a web browser window displaying the 'Sign in' page. The page has a light gray background. At the top left, there is a 'Help' link. The main heading 'Sign in' is centered in a large, bold, red font. Below the heading, there are two white input fields: the first is labeled 'Email' and the second is labeled 'Password'. At the bottom, there are two buttons: a light gray 'Confirm' button and a blue 'Back' button.

2.6.1.8 Interfaccia – Courier View Choose



A screenshot of a web browser window displaying the 'Courier Choose' page. The page has a light gray background. At the top left, there is a 'Help' link. The main heading 'Courier Choose' is centered in a large, bold, red font. Below the heading, there are three buttons: a light gray 'Update Status' button, a light gray 'History' button, and a blue 'Back' button.

2.6.1.9 Interfaccia– Courier View UpdateStatus

Help

Trading unit code

Sorting center:

Consegnato

Report

Back

2.6.1.10 Interfaccia – Courier View History

[illegible]

2.6.2 Test Di Usabilità

Per i test sull'usabilità del software, sono stati usati alcuni utenti campione che hanno utilizzato il sistema sotto nostra osservazione, con conseguente raccolta di dati e analisi di questi.

2.6.2.1 Obiettivi del test

L'obiettivo del test è quello di effettuare una valutazione sull'esperienza d'uso dell'applicazione. Nello specifico riferendosi alla semplicità con cui si interfaccia con un utente.

La valutazione può avvenire mediante:

- **Valutazioni attraverso l'analisi di esperti di usabilità**, per esempio, attraverso i principi di euristica. I principi di euristica sono linee guida generali per la valutazione dell'usabilità di un prodotto. Essi comprendono criteri come la chiarezza dell'interfaccia, la completezza delle informazioni, la prevedibilità del comportamento del sistema, la flessibilità e la capacità di recuperare dagli errori.
- **Valutazioni con la partecipazione degli utenti** attraverso test di usabilità e interrogazioni.

2.6.2.2 Metodologia usata

Il nostro test si è basato sul far partecipare al test individui specializzati in diversi ambiti. Chi ha partecipato al test non era a conoscenza dell'interfaccia della nostra applicazione ma conosceva gli utilizzi di quest'ultima. Il test prevedeva un'interazione con il sistema e una conseguente domanda riguardo il grado di difficoltà riscontrato.

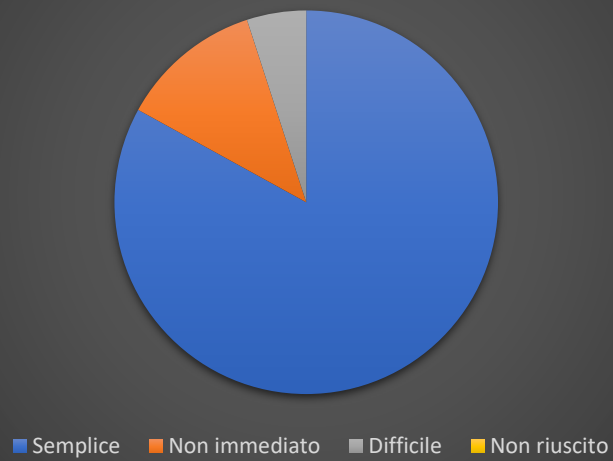
Il test è durato poco meno di 5 minuti a persona.

2.6.2.3 Sintesi delle misure

Lato Client:



Traccia pacco

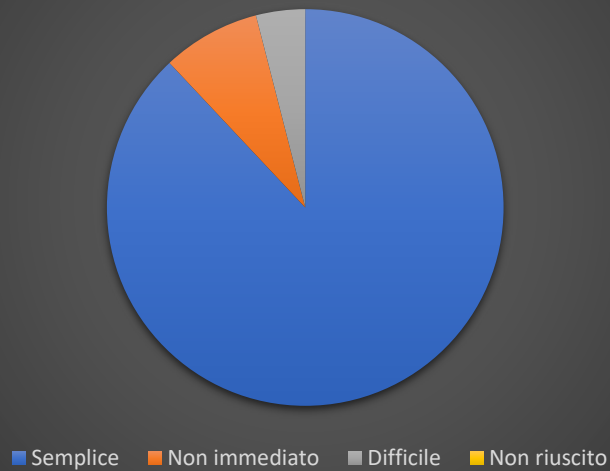


Lato Couriers

Visualizzazione storico dei pacchi



Aggiornamento del pacco da parte del corriere



2.6.2.4 Analisi dei risultati e sintesi delle interviste

Durante i vari test effettuati i problemi principali riscontrati sono fatti:

- Il **tracciamento del pacco** è vincolato dal codice sorgente di quest'ultimo che deve essere necessariamente inserito. È capitato che il tester non sapesse dove reperire il codice da inserire.
- La **visualizzazione dei pacchi** presenta un semplice tasto history che per pochissimi è risultato ambiguo.
- Il **cambiamento di sede** è vincolato dal codice sorgente di quest'ultimo che deve essere necessariamente inserito. È capitato che il tester non sapesse dove reperire il codice da inserire o il luogo da inserire per aggiornare il pacco.

2.6.2 Valutazione dell'Usabilità

