Università degli studi di Salerno

*Corso di Laurea in Informatica*

**Requirement Analysis Document**

**“fasTrain”**



**Autori:**

**Nome**

Acierno Erminio

Avallone Daniele

Paolillo Domenico

**Matricola**

0512104934

0512104814

0512104826

**Sommario**

[**1. Introduction** 4](#_Toc24392982)

[**2. Current System** 4](#_Toc24392983)

[**3. Proposed System** 5](#_Toc24392984)

[**3.1 Overview** 5](#_Toc24392985)

[**3.2 Functional Requirements** 5](#_Toc24392986)

[**3.3 Non-Functional Requirement** 6](#_Toc24392987)

[**3.4 Constraints** 6](#_Toc24392988)

[**3.5** **System Models** 7](#_Toc24392989)

[**3.5.1** **Scenarious** 7](#_Toc24392990)

[**3.5.2 Use Case Model** 11](#_Toc24392991)

[**3.5.3 Object Model** 26](#_Toc24392992)

[**3.5.4 Dynamic Model** 26](#_Toc24392993)

[**3.5.5 Screen Mock-ups** 27](#_Toc24392994)

# **1. Introduction**

Il progetto “fasTrain” è finalizzato alla realizzazione di un software per l’acquisto on-line di biglietti ferroviari. I biglietti che potranno essere acquistati sulla piattaforma consentono di viaggiare su treni della società “fasTrain”, società che tramite i suoi servizi, permette di viaggiare comodamente tra le città più importanti d’Italia.

Il software consentirà all’utente di conoscere le varie tratte ferroviarie coperte da “fasTrain”. Sarà inoltre possibile visualizzare tutte le informazioni relative ad una singola tratta, i posti disponibili all’interno del treno, il prezzo del singolo biglietto, ed eventualmente, effettuare un acquisto.

# **2. Current System**

L'agenzia ferroviaria “fasTrain” non è provvista di un sito web e ciò causa molto disagio ai viaggiatori che attendono tempi troppo elevati per poter acquistare i loro biglietti.

Il non avere un software per l’acquisto on-line di biglietti causa non solo ritardi nei servizi ferroviari e una lunga attesa per richiedere informazioni, ma anche gravi perdite economiche alla società.

L’agenzia “fasTrain” si è infine resa conto che ciò causa disagi ai suoi clienti, che optano per altre società di trasporto. “fasTrain” ha deciso pertanto di aprire una piattaforma che possa offrire tutti i servizi on-line utili per un viaggiatore.

# **3. Proposed System**

## **3.1 Overview**

Il sistema consentirà di vedere tutte le tratte che hanno posti disponibili, le tratte in offerta con un prezzo ridotto, ricercare una specifica corsa, effettuare registrazione e autenticazione ed, ovviamente, acquistare un biglietto per una tratta, servizio quest’ultimo riservato solo ad utenti registrati al sito.

## **3.2 Functional Requirements**

Il software dovrà fornire tutte le funzionalità riportate di seguito:

1. **Registrazione utente**: un utente può registrarsi al sito. Non obbligatorio, ma necessario all’acquisto di un biglietto.
2. **Login utente**: un utente che ha provveduto a creare un account può accedere al sito.
3. **Logout utente**: un utente loggato può disconnettersi dal sito.
4. **Prenotazione biglietto**: un utente registrato al sito che ha effettuato il login può prenotare un biglietto.
5. **Possibilità di vedere la lista dei biglietti acquistati**: un utente loggato ha la possibilità di controllare i biglietti precedentemente acquistati.
6. **Visionare offerte**: un utente ha la possibilità di controllare le offerte disponibili.
7. **Scegliere e visionare le diverse tariffe**: un utente può cercare e visionare le tariffe delle corse.
8. **Modificare le credenziali**: un utente registrato e loggato può modificare i suoi dati personali.
9. **Selezionare la classe con cui viaggiare**: un utente registrato e loggato, durante l’acquisto di un biglietto, può selezionare la classe con cui vuole viaggiare.

## **3.3 Non-Functional Requirement**

1. **Tempo di risposta**: dopo ogni interazione con l’utente, il sistema deve rispondere entro e non oltre 30 secondi.
2. **Disponibilità del sito**: il sistema deve essere disponibile 24 ore al giorno, tutti i giorni.
3. **Controllo sulla registrazione**: il sistema avrà un controllo per il corretto inserimento dei dati nel form di registrazione.
4. **Controllo sugli accessi**: il sistema nega l’accesso ad utenti non registrati.
5. **Feedback errore**: in caso di problemi (connessione, errori del server) all’utente verrà mostrata una pagina di errore.
6. **Legal requirement**: il sistema non ha nessun diritto d’autore e nessun costo.

## **3.4 Constraints**

1. L’applicazione utilizza la logica MVC: sarà quindi usato come linguaggio di programmazione Java.
2. Tutti i documenti prodotti dal team di sviluppo dovranno essere in formato .docx o .pdf.
3. Il sistema non avrà bisogno di alcuna installazione sui dispositivi sulla quale verrà utilizzato, in quanto è una piattaforma on-line.

## **System Models**

### **Scenarious**

#### **3.5.1.1 Acquista Biglietto**

|  |  |
| --- | --- |
| **Attori partecipanti** | Utente registrato, Antonio |
| **Flusso di eventi** | * Antonio ha avuto un importante offerta di lavoro e ha la necessità di prenotare un treno che gli permetta di andare da Salerno a Roma. * Antonio intenzionato ad acquistare un biglietto sul sito “fasTrain” effettua l’accesso inserendo la mail e la password scelte quando si è registrato. * Effettuato con successo il login, Antonio nota che sulla home page del sito ci sono delle offerte ma sfortunatamente non ci sono offerte relative alla data in cui vuole partire quindi decide di effettuare una ricerca sul catalogo di FasTrain. * Antonio completa la form di ricerca nel catalogo con i dati richiesti: stazione di partenza🡪Salerno, stazione di destinazione🡪Roma, data di partenza🡪05/03/2019. * Cliccando sul bottone di ricerca appare una lista di tutte le corse disponibili da Salerno a Roma per il giorno selezionato. * Trovata una corsa che parte alle 7 di mattina Antonio ne verifica la disponibilità cliccando sul pulsante che gli consente di vedere tutte le informazioni di quella corsa. * La pagina contente le informazioni sulla corsa scelta ha i seguenti campi: stazione di partenza🡪Salerno, stazione di destinazione🡪Roma Termini, codice del treno🡪4526, orario di partenza🡪07:03, ora di arrivo🡪08:58, durata🡪1h50m, prezzo🡪19,50€. * Una volta scelta la prima classe dalla spunta apposita, il numero di biglietti da acquistare e il nominativo del passeggero da apporre sul bigleitto, Antonio decide di procedere con l’acquisto. * Antonio sarà reindirizzato alla pagina di pagamento che mostra tutti i suoi dati e una form da compilare: numero carta di credito🡪0123 4567 8901 2345, compagnia carta🡪Visa, intestatario carta🡪Nome Cognome, data scadenza🡪05/22, CVV🡪123. * Una volta completata la form ed inviata, se i dati sono corretti l'acquisto sarà completato e notificato. |

**3.5.1.2 Aggiungi nuova corsa**

|  |  |
| --- | --- |
| **Attori partecipanti** | Amministratore del catalogo, Giovanni |
| **Flusso eventi** | * Giovanni gestore del catalogo del sito fasTrain ha ricevuto l’ordine dalla compagnia di inserire la nuova corsa Firenze-Torino delle ore 15:00 istituita di recente a causa della frequente richiesta dei pendolari. * L’amministratore effettua il login al gestionale. * Andando sul catalogo delle corse Giovanni ha la possibilità di inserire nuove offerte in merito alle corse già inserite o aggiungere una nuova corsa. * L’amministratore clicca sul pulsante per inserire una nuova corsa ed è reindirizzato alla pagina per compilare una form: stazione di partenza🡪Firenze, stazione di destinazione🡪Torino, data🡪05/10/2019, orario di partenza🡪15:00, ora di arrivo🡪17:55, durata corsa🡪2h55m, codice treno🡪4218, numero posti economy🡪470, costo biglietto economy🡪20,50€, numero posti business🡪400, costo biglietto business🡪31,80€, numero posti premium🡪240, costo biglietto premium🡪42,50€. * Una volta aver confermato e aver avuto la conferma dal sistema dell’avvenuto inserimento della nuova corsa, Giovanni si accorge di aver commesso un errore in fase di compilazione della form. * L’amministratore ha la facoltà di eliminare del tutto le corse o di modificarne le informazioni già inserite. * L’amministratore effettua una ricerca per trovare la corsa appena inserita compilando i campi: stazione di partenza🡪Firenze, stazione di destinazione🡪Torino, data🡪05/10/2019. * Trovata la suddetta corsa Giovanni può modificarne i dati inseriti in precedenza cliccando sull’apposito pulsante. * Il campo errato è numero posti economy🡪770. * Una volta effettuata la correzione numero posti economy 🡪770, Giovanni può salvare la modifica apportata dall’apposito pulsante. * Il sistema restituisce un messaggio dell’avvenuta modifica. |

### **3.5.2 Use Case Model**



#### **3.5.2.1 Actors**

Nel sistema, saranno disponibili i seguenti profili:

**Utente**: sarà offerto all’utente le seguenti operazioni: registrazione, login, acquisto biglietto, visualizzazione biglietti prenotati, modifica prenotazione, modifica dati personali, logout, ricerca corsa, visione offerte.

**Gestore:** sarà compito del gestore curare alcune operazioni come: aggiungere una nuova corsa, rimuovere una corsa, modificare una corsa, ricercare una corsa.

**3.5.2.2 Casi d’uso**

**3.5.2.2.1 Registrazione**

|  |  |
| --- | --- |
| **Attori partecipanti** | Utente |
| **Entry condition** | L’utente è un visitatore e vuole registrarsi. |
| **Flusso eventi** | * L’utente clicca sul bottone “accedi” presente nella home page nella sezione in alto a destra. * Il sistema mostra all’utente una schermata per effettuare il login o registrarsi. L’utente clicca sul bottone “registrazione”. * L’utente è reindirizzato alla pagina di registrazione dove deve compilare una form con i campi: nome, cognome, email, età, numero di telefono, indirizzo, città, provincia, password, conferma password. Una volta compilati i campi l’utente clicca il pulsante “registrati”. * Se i dati inseriti non sono corretti il sistema risponde con un messaggio di errore facendo intervenire il caso d’uso “registrazione errata”. |
| **Exit condition** | * Il sistema risponde indirizzando l’utente alla pagina iniziale e mostrando un feedback “registrazione effettuata”. |
| **Condition requirements** | Il completamento della procedura deve essere effettuato in un massimo di 5 minuti. |
| **Eccezione** | Registrazione errata |

**3.5.2.2.2 Login**

|  |  |
| --- | --- |
| **Attori partecipanti** | Utente |
| **Entry condition** | Utente già registrato che si trova nella homepage |
| **Flusso eventi** | * L’utente clicca sul bottone “Accedi” * Il sistema mostra un form che prevede i campi: email, password. * L’utente dopo aver compilato i campi clicca sul pulsante “Accedi”. * Il sistema controlla i dati, se i dati non sono corretti viene mostrato un messaggio di errore facendo intervenire il caso d’uso “Login errato”. |
| **Exit condition** | Il sistema risponde indirizzando l’utente alla pagina iniziale. |
| **Condition requirements** | Il completamento della procedura deve essere effettuato con un tempo massimo di 2 minuti. |
| **Eccezione** | Login Errato |

**3.5.2.2.3 Registrazione Errata**

|  |  |
| --- | --- |
| **Attori partecipanti** | Utente |
| **Entry condition** | L’utente è un visitatore e vuole registrarsi. |
| **Flusso eventi** | * L’utente clicca sul bottone “accedi” presente nella home page nella sezione in alto a destra. * Il sistema mostra all’utente una schermata per effettuare il login o registrarsi. L’utente clicca sul bottone “registrazione”. * L’utente è reindirizzato alla pagina di registrazione dove deve compilare una form con i campi: nome, cognome, email, età, numero di telefono, indirizzo, città, provincia, password, conferma password. Una volta compilati i campi l’utente clicca il pulsante “registrati”. * Il sistema una volta controllati i dati inseriti rileva che in uno o più campi sono presenti errori. |
| **Exit condition** | Il sistema risponde mostrando nella pagina di registrazione una scritta “registrazione non effettuata” evidenziando in rosso i campi errati. |
| **Condition requirements** | Il completamento della procedura deve essere effettuato in un massimo di 5 minuti. |

**3.5.2.2.4 Login Errato**

|  |  |
| --- | --- |
| **Attori partecipanti** | Utente |
| **Entry condition** | È un utente registrato e si trova nella homepage |
| **Flusso eventi** | * L’utente clicca sul bottone “Accedi” * Il sistema mostra un form che prevede i campi: email, password. * L’utente dopo aver compilato i campi clicca sul pulsante “Accedi”. * Il sistema non riconosce la coppia email-password impedendo l’accesso alla piattaforma. |
| **Exit condition** | Il sistema risponde indirizzando l’utente alla pagina di login mostrando un feedback “login non riuscito”. |
| **Condition requirements** | Il completamento della procedura deve essere effettuato con un tempo massimo di 1 minuto. |

**3.5.2.2.5 Ricerca corsa**

|  |  |
| --- | --- |
| **Attori partecipanti** | Utente o gestore |
| **Entry condition** | L’utente si trova nella homepage |
| **Flusso eventi** | * L’utente completa la form di ricerca posta al centro della pagina con i dati richiesti: stazione di partenza, stazione di destinazione, data partenza. * L’utente clicca sul pulsante “cerca” ed il sistema ritornerà le corse compatibili con i dati specificati dall’utente. * L’utente visualizza una serie di corse che presentano le seguenti informazioni: stazione di partenza, stazione di destinazione, ora di partenza, ora di arrivo, data di partenza, codice treno, classi. |
| **Exit condition** | L’utente visualizza il risultato della ricerca. |
| **Condition requirements** | Il completamento della procedura deve essere effettuato con un tempo massimo di 1 minuto. |

**3.5.2.2.6 Info biglietto**

|  |  |
| --- | --- |
| **Attori partecipanti** | Utente |
| **Entry condition** | L’utente deve aver completato la ricerca di un biglietto. (Si veda caso d’uso “Ricerca Corsa”). |
| **Flusso eventi** | * L’utente clicca sul pulsante “Compra” * L’utente visualizza una serie di dati: stazione partenza, stazione arrivo, codice treno, ora di partenza, ora di arrivo, durata, prezzo classe premium, prezzo classe business, prezzo classe economy. * L’utente deve scegliere la classe di viaggio mediate tre radio button: classe premium, classe business, classe economy. * Selezionando una classe, l’utente visualizza la variazione del prezzo. * L’utente seleziona la quantità di biglietti da voler comprare e clicca il pulsante “Procedi all’acquisto”. |
| **Exit condition** | L’utente visualizza lo storico ordini dei biglietti. |
| **Condition requirements** | Il completamento della procedura deve essere effettuato con un tempo massimo di 1 minuto. |

**3.5.2.2.7 Acquisto di un biglietto**

|  |  |
| --- | --- |
| **Attori partecipanti** | Utente loggato |
| **Entry condition** | Utente loggato vuole comprare il biglietto di una corsa (si guardi caso d’uso “Info biglietto”). |
| **Flusso eventi** | * L’utente clicca sul pulsante “Procedi all’acquisto”. * Il sistema mostra una pagina con i dati anagrafici da inserire sul biglietto (nome e cognome), le informazioni sulla corsa scelta (stazione partenza, stazione arrivo, durata, ora partenza, codice treno, classe scelta, numero biglietti). * Una form per il pagamento richiederà il completamento dei seguenti campi: numero carta, intestatario carta, data scadenza, CVV, compagnia della carta. * Una volta compilato tutto l’utente può procedere con il completamento dell’acquisto premendo il tasto “Acquista”. * Il sistema risponde con un feedback di avvenuto acquisto e reindirizzerà l’utente alla lista dei biglietti acquistati. |
| **Exit condition** | L’utente ha acquistato un biglietto. |
| **Condition requirements** | Il completamento della procedura deve essere effettuato con un tempo massimo di 1 minuto. |

**3.5.2.2.8 Inserisci corsa**

|  |  |
| --- | --- |
| **Attori partecipanti** | Gestore |
| **Entry condition** | Il gestore è loggato |
| **Flusso eventi** | * Il gestore preme sul bottone “Inserimento Corsa” * Viene rimandato ad una pagina dove deve completare una form inserendo i seguenti dati: stazione di partenza, stazione di destinazione, data, ora di partenza, ora di arrivo, prezzo classe premium, prezzo classe business, prezzo classe economy, sconto, codice treno, numero posti classe premium, numero posti classe business, numero posti classe economy. * Il gestore invia la form, se i dati sono corretti le informazioni vengono salvate nel sistema. |
| **Exit condition** | Il sistema risponde mostrando il feedback di avvenuto inserimento della corsa. |
| **Condition requirements** | Il completamento della procedura deve essere effettuato con un tempo massimo di 1 minuto. |

**3.5.2.2.9 Modifica dati corsa**

|  |  |
| --- | --- |
| **Attori partecipanti** | Gestore |
| **Entry condition** | Il gestore è loggato e si trova nella homepage e ha cercato la corsa che vuole modificare (Si guardi il caso d’uso “Ricerca Corse”). |
| **Flusso eventi** | * Il Gestore clicca il pulsante “modifica” presente vicino le singole corse sulla destra. * Il Gestore decide quale dato modificare tra quelli presenti: stazione di partenza, stazione di destinazione, data, ora di partenza, ora di arrivo, prezzo classe premium, prezzo classe business, prezzo classe economy, sconto, codice treno, numero posti classe premium, numero posti classe business, numero posti classe economy. * Il Gestore preme il pulsante “applica modifica” e se ha completato correttamente i campi, la modifica verrà effettuata e salvata nel sistema |
| **Exit condition** | Il sistema risponde mostrando il feedback di avvenuta modifica. |
| **Condition requirements** | Il completamento della procedura deve essere effettuato con un tempo massimo di 1 minuto. |

**3.5.2.2.10 Cancella corsa**

|  |  |
| --- | --- |
| **Attori partecipanti** | Gestore |
| **Entry condition** | Il gestore è loggato e si trova nella homepage e ha cercato la corsa che vuole eliminare (Si guardi il caso d’uso “Ricerca Corse”). |
| **Flusso eventi** | * Il Gestore clicca il pulsante “Elimina” presente vicino le singole corse sulla destra. * Il sistema richiede una conferma sulla scelta dell’eliminazione. * Il gestore clicca sul tasto “Si, elimina”. |
| **Exit condition** | Il sistema risponde mostrando un feedback di avvenuta cancellazione. |
| **Condition requirements** | Il completamento della procedura deve essere effettuato con un tempo massimo di 1 minuto. |

**3.5.2.2.11Visione offerte**

|  |  |
| --- | --- |
| **Attori partecipanti** | Utente |
| **Entry condition** | Utente in cerca di un offerta |
| **Flusso eventi** | * L’utente sulla homepage clicca sulla voce di menu “Offerte”. * Il sistema restituisce la pagina con le offerte in corso. * L’utente trova nella lista degli sconti una corsa al quale è interessato quindi clicca sul pulsante “Scopri di più” per ottenere maggiori informazioni. * Il sistema reindirizza l’utente alla pagina in cui visualizza i dati relativi alla corsa in offerta: stazione di partenza, stazione di destinazione, data e ora di partenza, classe. |
| **Exit condition** | L’utente visualizza le informazioni di un offerta (si guardi il caso d’uso “Info biglietto”. |
| **Condition requirements** | Il completamento della procedura deve essere effettuato con un tempo massimo di 1 minuto. |

**3.5.2.2.12 Modifica dati utenti**

|  |  |
| --- | --- |
| **Attori partecipanti** | Utente |
| **Entry condition** | L’utente deve essere loggato. |
| **Flusso eventi** | * L’utente tramite il menu seleziona la scritta “I miei dati”. * Il sistema reindirizza l’utente alla pagina che contiene i suoi dati personali. * L’utente nota di aver commesso un errore in fase di registrazione, cliccando sul campo in questione e ne modifica il contenuto. * L’utente preme il pulsante “Modifica”. * Il sistema restituisce un feedback di avvenuta modifica aggiornando la pagina. |
| **Exit condition** | L’utente otterrà la modifica dei propri dati. |
| **Condition requirements** | Il completamento della procedura deve essere effettuato con un tempo massimo di 1 minuto. |

**3.5.2.2.13 Visualizza Biglietti acquistati**

|  |  |
| --- | --- |
| **Attori partecipanti** | Utente |
| **Entry condition** | L’utente deve essere loggato |
| **Flusso eventi** | * L’utente tramite il menu clicca la scritta “Biglietti prenotati”. * Il sistema mostra la lista di tutti i biglietti acquistati. |
| **Exit condition** | L’utente visualizza lo storico ordini dei biglietti. |
| **Condition requirements** | Il completamento della procedura deve essere effettuato con un tempo massimo di 1 minuto. |

**3.5.2.2.14 Modifica biglietto**

|  |  |
| --- | --- |
| **Attori partecipanti** | Utente loggato |
| **Entry condition** | L’utente ha effettuato un acquisto |
| **Flusso eventi** | * L’utente si trova nella pagina “Biglietti acquistati” * L’utente nota di aver commesso un errore nel digitare il nominativo del passeggero nell’ultima prenotazione effettuata. * L’utente preme il pulsante “Modifica”. * Il sistema reindirizza l’utente nella pagina “modifica biglietto” in cui l’utente ha la possibilità di modificare il nominativo della prenotazione fino al giorno prima della corsa scelta. Dopo aver controllato i campi nome, cognome, codice documento, l’utente clicca il pulsante “conferma modifica”. * Il sistema restituisce un feedback di avvenuta modifica reindirizzando l’utente alla pagina “Biglietti prenotati”. |
| **Exit condition** | L’utente otterrà la modifica del nominativo passeggero sul biglietto. |
| **Condition requirements** | Il completamento della procedura deve essere effettuato con un tempo massimo di 1 minuto. |

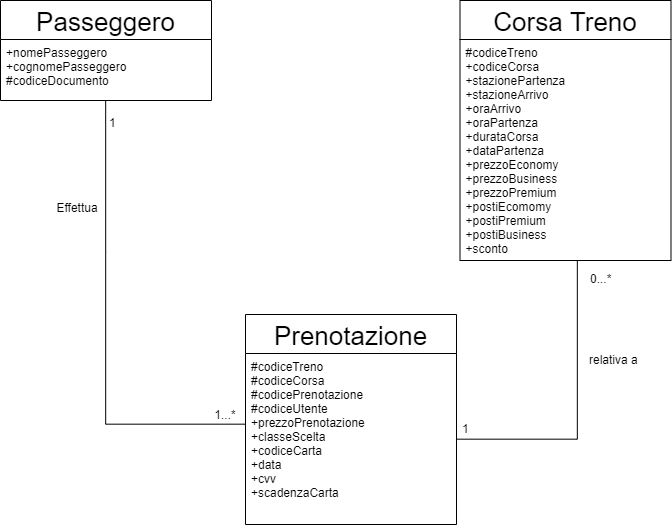
**3.5.2.2.15 Logout**

|  |  |
| --- | --- |
| **Attori partecipanti** | Utente |
| **Entry condition** | L’utente deve essere loggato |
| **Flusso eventi** | * L’utente loggato clicca la scritta “Logout” in alto a destra del menu. * Il sistema porterà a compimento la procedura reindirizzando l’utente alla homepage. |
| **Exit condition** | L’utente non è più loggato, reindirizzamento alla home page. |
| **Condition requirements** | Il completamento della procedura deve essere effettuato con un tempo massimo di 1 minuto. |

#### 

### 

### **3.5.3 Object Model**



#### 

#### **3.5.3.1 Passeggero**

La classe Passeggero è una classe che dipende dalla classe Prenotazione, essa ha come attributi i dati del passeggero, formati da: nome, cognome e codice documento. Le operazioni di questa classe sono i get e set degli attributi.

#### **3.5.3.2 Corsa Treno**

Corsa è una classe che dipende dal gestore che memorizza le informazioni. Queste informazioni definiscono la corsa: codiceCorsa, stazione di partenza, stazione di destinazione, DataPartenza, Ora di partenza, ora arrivo, prezzo economy, prezzo business, prezzo premium, codice treno, posti classe economy, posti classe business e posti classe premium. I prezzi economy, business e premium sono diversi siccome la qualità e i servizi sono differenti. Le operazioni di Corsa sono i getter e i setter degli attributi.

Ovviamente alcuni campi devono essere diversi dagli altri: ora partenza deve essere diversa da ora di arrivo, stazione di partenza deve essere diversa da stazione di arrivo.

#### **3.5.3.3 Prenotazione**

Prenotazione è una classe che dipende da Utente (che effettua la prenotazione). È caratterizzato da il codice della prenotazione che è diverso da ogni prenotazione, il codice treno, il codice corsa, il prezzo totale della prenotazione che è dato dal costo dei biglietti che dipende dalla classe designata. Poi è presente il numero della carta di pagamento con la data di scadenza e il CVV. Le operazioni di questa classe sono i get e set delle istanze.

### **3.5.4 Dynamic Model**

#### **3.5.4.1 Sequence Diagrams**

#### **3.5.4.2 State Machine Diagram**

#### **3.5.4.3 Activity Diagram**

### **3.5.5 Screen Mock-ups**