
Documento di Design del Sistema

per
EC-18

Versione 1.0

Autori:

L. Maddaluno

R. Meo

M. Vitiello

27/06/2019

blank page

Indice

2	Design del Sistema	5
2.1	Analisi dell'architettura	5
2.1.1	Architettura fisica del sistema	5
2.1.2	Architettura logica del sistema	6
2.1.3	Tecnologie utilizzate	7
2.1.3.1	Amazon Web Services	7
2.1.3.1.1	AWS Cognito	8
2.1.3.1.2	AWS EC2	8
2.1.3.1.3	AWS Elastic Beanstalk	9
2.1.3.1.4	AWS RDS	9
2.1.3.1.5	AWS S3	9
2.1.3.2	Apache Tomcat	10
2.1.3.3	HTML + CSS + JavaScript	10
2.1.3.4	Java	11
2.1.3.5	JSON	11
2.1.3.6	PayPal	11
2.1.3.7	PostgreSQL	12
2.1.3.8	Retrofit	12
2.1.3.9	Braintree	13
2.2	Diagramma delle classi di design	14
2.3	Diagrammi di sequenza di design	18
2.3.1	Sequence Diagram – Registrazione – WEB	18
2.3.2	Sequence Diagram – Login – WEB	19
2.3.2.1	Sequence Diagram – Password dimenticata – WEB	20
2.3.3	Sequence Diagram – Ricerca prodotto – WEB	21
2.3.4	Sequence Diagram – Acquista prodotto – WEB	22
2.3.5	Sequence Diagram – Lascia feedback – WEB	24
2.3.6	Sequence Diagram – Aggiungi prodotto – WEB	24
2.3.7	Sequence Diagram – Modifica prodotto – WEB	25
2.3.8	Sequence Diagram – Elimina prodotto – WEB	25
2.3.9	Sequence Diagram – Visualizzazione statistiche – WEB	26
2.3.10	Sequence Diagram – Registrazione – APP	29
2.3.11	Sequence Diagram – Login – APP	30
2.3.12	Sequence Diagram – Ricerca prodotto – APP	31
2.3.13	Sequence Diagram – Acquista prodotto – APP	32
2.3.14	Sequence Diagram – Lascia feedback – APP	32

Data	Versione	Autore	Descrizione
27/06/2019	1.0	L. Maddaluno, R. Meo, M. Vitiello	Release iniziale

2 Design del Sistema

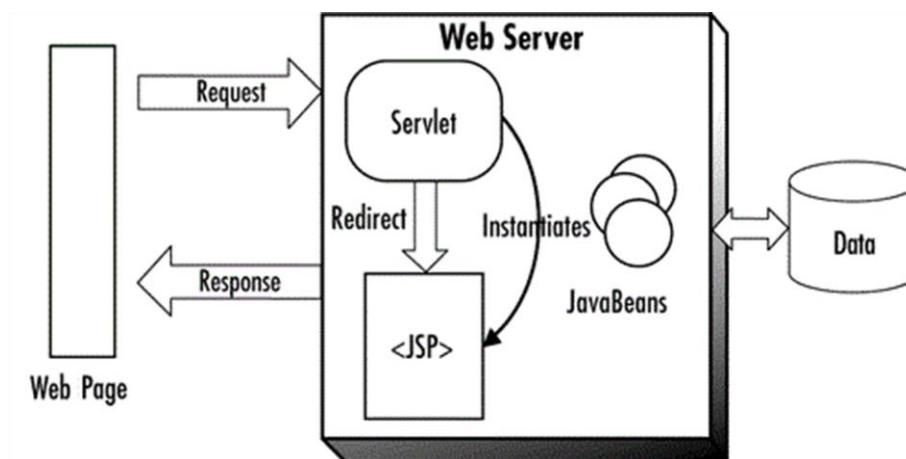
In questo documento vengono affrontate le questioni riguardanti l'implementazione delle funzioni specificate nel Documento dei Requisiti Software. Durante il processo di design vengono prese delle decisioni per soddisfare i requisiti raccolti: in particolare vengono descritte le scelte implementative seguite per quanto riguarda sia l'aspetto logico che quello fisico. Ogni tecnologia utilizzata viene descritta con cura e sono spiegate le sue principali caratteristiche.

2.1 Analisi dell'architettura

La scelta e l'utilizzo di una tecnologia è un processo di decisione indirizzato ad ottenere risultati ottimali per un progetto, in quanto una decisione sbagliata o imprudente può compromettere il soddisfacimento degli obiettivi preposti. È anche importante descrivere correttamente il design del sistema, per esempio indentificare ed applicare le strategie software che possono risolvere i problemi esistenti nel progetto.

2.1.1 Architettura fisica del sistema

Il design del sistema segue un'architettura client-server divisa in tre livelli: il livello dei dati corrisponde ad un database relazionale all'interno del quale sono salvate le informazioni necessarie al corretto funzionamento del sistema; il livello di applicazione è rappresentato dal web server Apache Tomcat, che come vedremo successivamente opera una divisione tra modello di presentazione e modello logico implementando le specifiche Java Server Pages e servlet; infine il modello di presentazione è rappresentato da pagine HTML e JSP, grazie alle quali è possibile ricevere e inviare dati al server.



2.1.2 Architettura logica del sistema

L'architettura logica del sistema è basata sul paradigma MVC (Model-View-Controller), il quale è ampiamente utilizzato nello sviluppo di applicazioni web per la sua facilità di utilizzo e la sua scalabilità. L'utilizzo dell'MVC è quasi naturale date le tecnologie utilizzate:

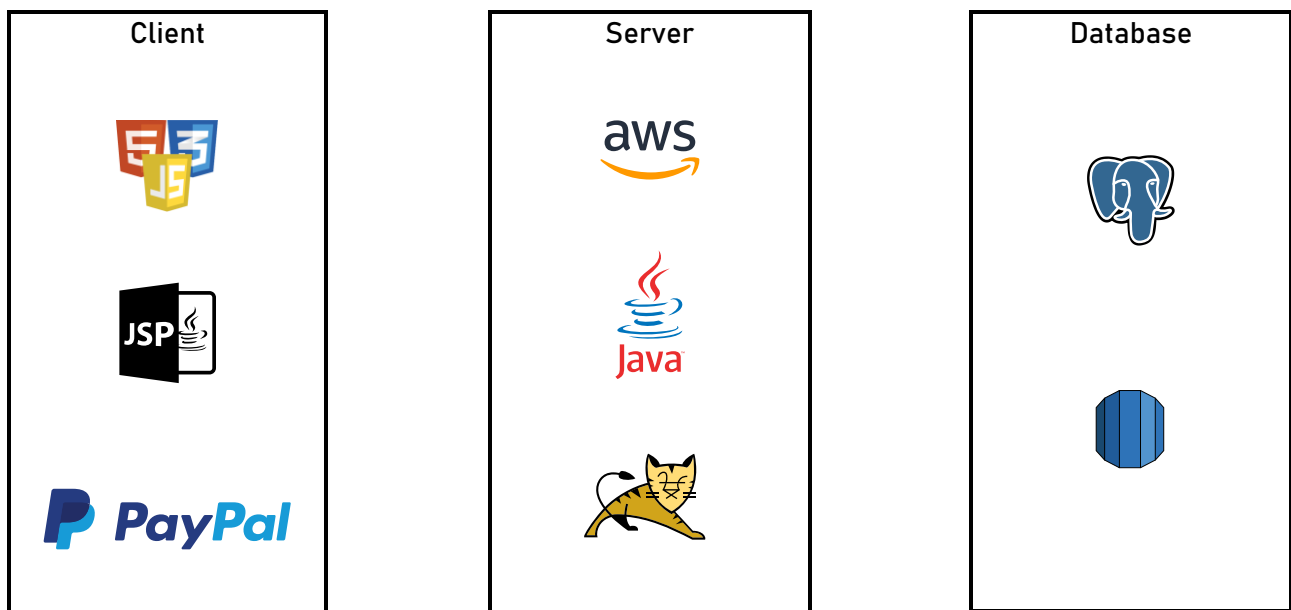
- Il sottosistema Model, che implementa la struttura dati centrale, è implementato completamente in Java, che ha il vantaggio di essere un linguaggio con caratteristiche di sicurezza che lo rendono particolarmente adatto alle applicazioni web, quali ad esempio la verifica del Bytecode prima dell'esecuzione o la gestione automatica della memoria grazie al Garbage Collector.
- Il sottosistema Controller è il responsabile del flusso di controllo: ottiene gli input dall'utente e manda messaggi al modello, comunicando tramite richieste HTTP. Grazie alle componenti presenti nel web server Tomcat, è possibile utilizzare il linguaggio Java anche per questo sottosistema.
- Il Viewer è il responsabile della presentazione del modello, ed elabora i dati ricevuti dal controller per mostrarli all'utente nella finestra del browser.

Quando l'utente effettua una richiesta, questa viene raccolta ed analizzata dal server, che avvia una particolare operazione del corrispondente controller. In seguito, il controller elabora i dati ricevuti e applica le modifiche necessarie al model. Una volta finita l'elaborazione della richiesta, il controller seleziona il formato più appropriato e invia tutte le informazioni riguardanti la richiesta corrente alla view: con queste informazioni la view genera un documento HTML che viene inviato al client tramite una risposta HTTP.

L'utilizzo di Apache Tomcat come web server permette l'esecuzione di codice Java nativamente: infatti mentre il controller è rappresentato dalle servlet, classi java che riescono a gestire richieste HTTP e inviare dati tramite risposte HTTP, la view è costituita da pagine JSP, ovvero un insieme di HTML e Java: è possibile infatti introdurre codice all'interno del template statico di formattazione, rendendo possibile un certo livello di logica anche lato client. Le pagine JSP vengono infatti compilate e tradotte in servlet dal servlet engine presente in Tomcat, per poi generare un template HTML da inviare al client tramite HTTP.

2.1.3 Tecnologie utilizzate

Il software EC-18 implementa alcune tecnologie che sono dettate dai requisiti, a partire dall'utilizzo di un database relazionale come strumento di archiviazione dati, fino ad arrivare al mezzo di pagamento di PayPal per effettuare acquisti in massima sicurezza. Altre tecnologie client-side e server-side invece necessitano di essere scelte con cura - considerando anche il fatto che far guadagnare esperienza agli sviluppatori è uno dei requisiti più importanti del progetto- per cui il sistema ha bisogno di strumenti e tecnologie che permettano di organizzare e semplificare i compiti particolarmente complessi dettati dall'architettura logica. La figura seguente mostra una panoramica delle tecnologie utilizzate nelle diverse componenti dell'architettura.



2.1.3.1 Amazon Web Services



Amazon Web Services è una piattaforma che offre una vasta gamma di servizi cloud-based, come ad esempio elaborazione, storage, database, reti, analisi, machine learning, intelligenza artificiale, Internet of Things, sicurezza e sviluppo di applicazioni, distribuzione e gestione. Questa grande selezione di servizi e funzionalità rendono più semplice, veloce ed economica la migrazione di app esistenti al cloud e permettono la progettazione di qualsiasi cosa.

2.1.3.1.1 AWS Cognito



Amazon Cognito fornisce autenticazione, autorizzazione e gestione degli utenti per le applicazioni Web e mobili. Gli utenti possono accedere direttamente con un nome utente e una password, oppure tramite terze parti, ad esempio Facebook, Amazon o Google. I due componenti principali di Amazon Cognito sono i pool di utenti e i pool di identità. I pool di utenti sono directory utente che forniscono opzioni di registrazione e di accesso agli utenti delle tue app. I pool di identità consentono di concedere agli utenti l'accesso ad altri servizi AWS. È possibile usare i pool di identità e i pool di utenti separatamente o insieme.

2.1.3.1.2 AWS EC2



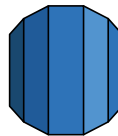
Amazon Compute Cloud dà la possibilità avviare un numero di server virtuali necessari, configurare la sicurezza e i servizi di rete e di gestire lo storage. Inoltre ti permette di aumentare o ridurre le risorse per gestire le variazioni a livello di requisiti o picchi di popolarità, riducendo la necessità di elaborare previsioni relative al traffico.

2.1.3.1.3 AWS Elastic Beanstalk



Elastic Beanstalk è un servizio che permette di distribuire e gestire rapidamente applicazioni nel cloud AWS senza necessità di apprendere l'infrastruttura che ne esegue tali applicazioni. Inoltre è possibile caricare semplicemente l'applicazione e questo servizio gestirà automaticamente i dettagli di provisioning delle capacità, bilanciamento del carico, dimensionamento e monitoraggio dello stato dell'applicazione.

2.1.3.1.4 AWS RDS



Amazon Relational Database Service è un servizio Web che semplifica la configurazione, l'uso e il dimensionamento di un database relazionale nel cloud AWS. Grazie a RDS è possibile gestire i backup, il rilevamento automatico dei guasti e l'eventuale ripristino del database.

2.1.3.1.5 AWS S3



Amazon S3 possiede varie caratteristiche utilizzabili per organizzare e gestire i dati in modi che supportano casi d'uso specifici, aumentano l'efficienza in termini di costo, rafforzano la sicurezza. I dati sono archiviati come oggetti all'interno di risorse chiamate "bucket. Le caratteristiche di S3 includono funzionalità per aggiungere tag di metadati agli oggetti, spostare e archiviare i dati tra le classi di storage S3, configurare e

rinforzare i controlli di accesso ai dati, proteggere i dati contro utenti non autorizzati, eseguire analisi dei Big Data e monitorare i dati a livello di oggetto e di bucket.

2.1.3.2 Apache Tomcat



Apache Tomcat è un web server open source che implementa le specifiche JavaServerPages (JSP) e servlet, fornendo quindi una piattaforma software per l'esecuzione di applicazioni Web sviluppate in linguaggio Java. La sua distribuzione standard include anche le funzionalità di web server tradizionale.

2.1.3.3 HTML + CSS + JavaScript



HTML, CSS e JavaScript sono le componenti fondamentali per lo sviluppo di applicazioni web per l'interfaccia grafica e la logica lato client.

- Html ovvero l'HyperText Markup Language è un linguaggio di markup. Nato per la formattazione e impaginazione di documenti ipertestuali.
- CSS ovvero Cascading Style Sheets è un linguaggio usato per definire la formattazione di documenti HTML. CSS si è necessario per separare i contenuti delle pagine HTML dalla loro formattazione o layout e permettere una programmazione più chiara e facile da utilizzare.
- JavaScript è un linguaggio di programmazione, interpretato, orientato agli oggetti. Viene eseguito direttamente lato "client-side" della pagina web e può essere utilizzato per dare un design e stabilire un comportamento delle pagine web.

2.1.3.4 Java



Java è un linguaggio di programmazione ad alto livello, orientato agli oggetti e a tipizzazione statica. I programmi scritti in linguaggio Java, dopo una fase iniziale di compilazione con ottenimento del cosiddetto bytecode, sono destinati all'esecuzione sulla piattaforma Java attraverso una fase di interpretazione (per questo motivo il linguaggio Java è detto anche semi-interpretato) ad opera di una [Java Virtual Machine](#) e in esecuzione hanno accesso alle API della libreria standard. Questi due passi forniscono un livello di astrazione che permette alle applicazioni di essere interamente indipendenti dal sistema hardware su cui esse saranno eseguite.

2.1.3.5 JSON



Json (Javascript Object notation), è un tipo di formato molto utilizzato per lo scambio dati in applicazioni client server come API. È basato su JavaScript ma il suo sviluppo è specifico per lo scambio di dati ed è indipendente dallo sviluppo del linguaggio di scripting dal quale nasce e con il quale è perfettamente integrato e semplice da utilizzare.

2.1.3.6 PayPal



PayPal è il servizio che ti consente di pagare, inviare denaro e accettare pagamenti in modo più rapido, semplice e sicuro, senza dover immettere ogni volta i tuoi dati finanziari.

La Sandbox di PayPal è un ambiente virtuale di test che simula delle transazioni di denaro esattamente come succederebbe con un account “reale” ma senza generare veri pagamenti.

2.1.3.7 PostgreSQL



PostgreSQL è un completo DBMS ad oggetti rilasciato con licenza libera. PostgreSQL usa il linguaggio SQL per eseguire delle query sui dati. Questi sono conservati come una serie di tabelle con chiavi esterne che servono a collegare i dati correlati.

2.1.3.8 Retrofit



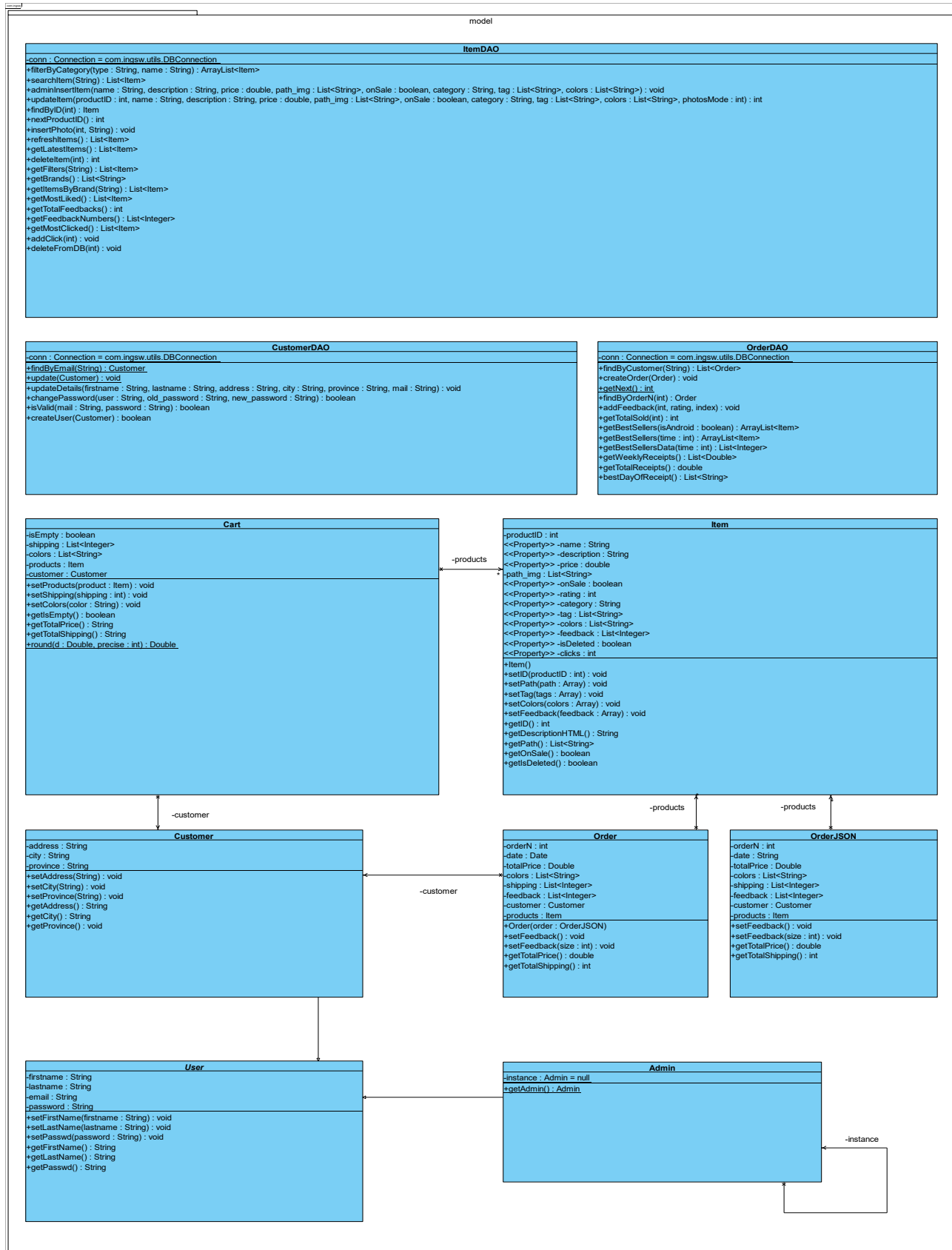
Retrofit è una libreria Android che consente di utilizzare come protocollo di comunicazione i servizi REST, un tipo di architettura software che utilizza il protocollo HTTP per effettuare richieste al server tramite le operazioni *CRUD*.

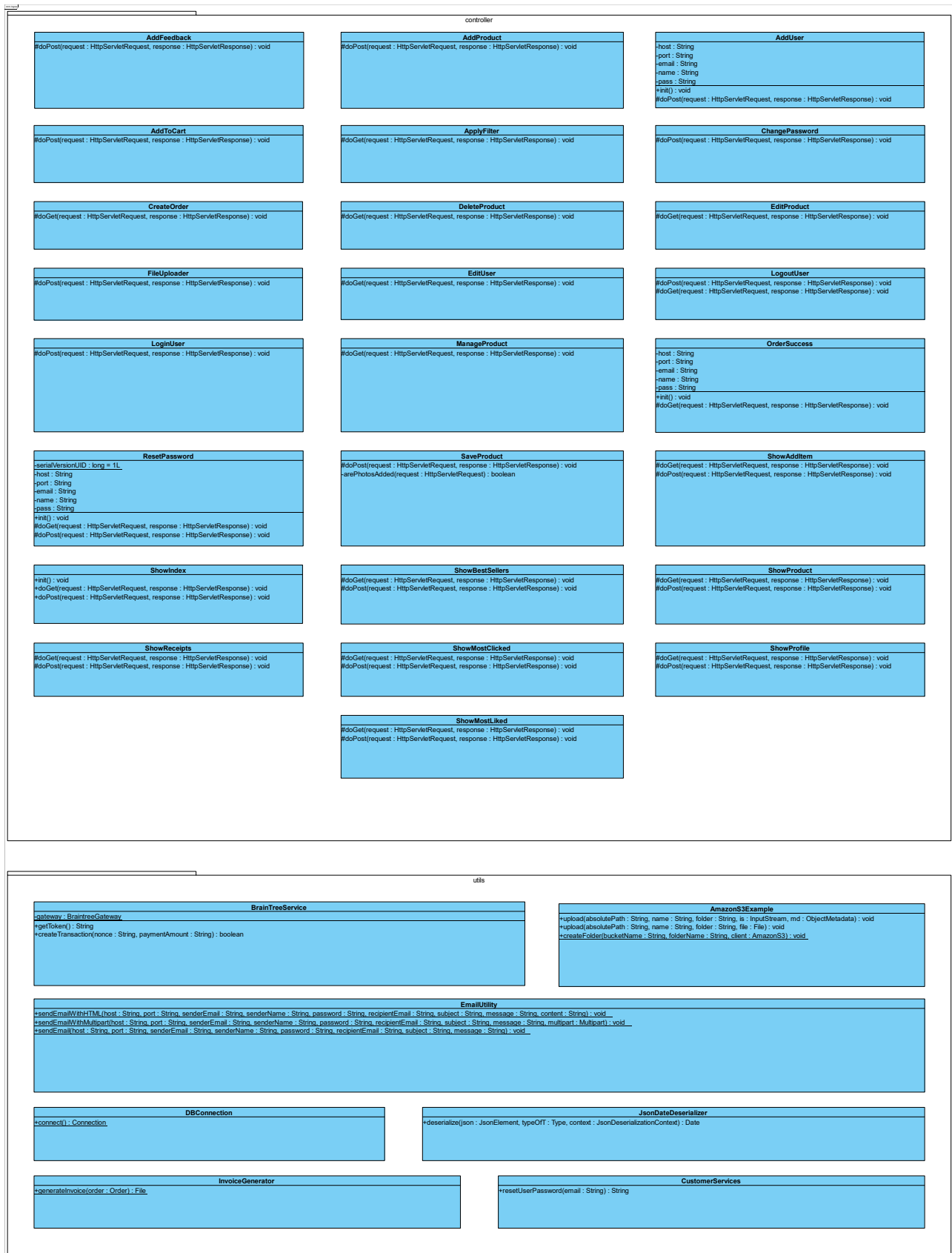
1. GET per accedere ad una risorsa;
2. PUT per creare una nuova risorsa;
3. POST/PATCH per modificare i valori di risorse preesistenti;
4. DELETE per cancellare dei dati.

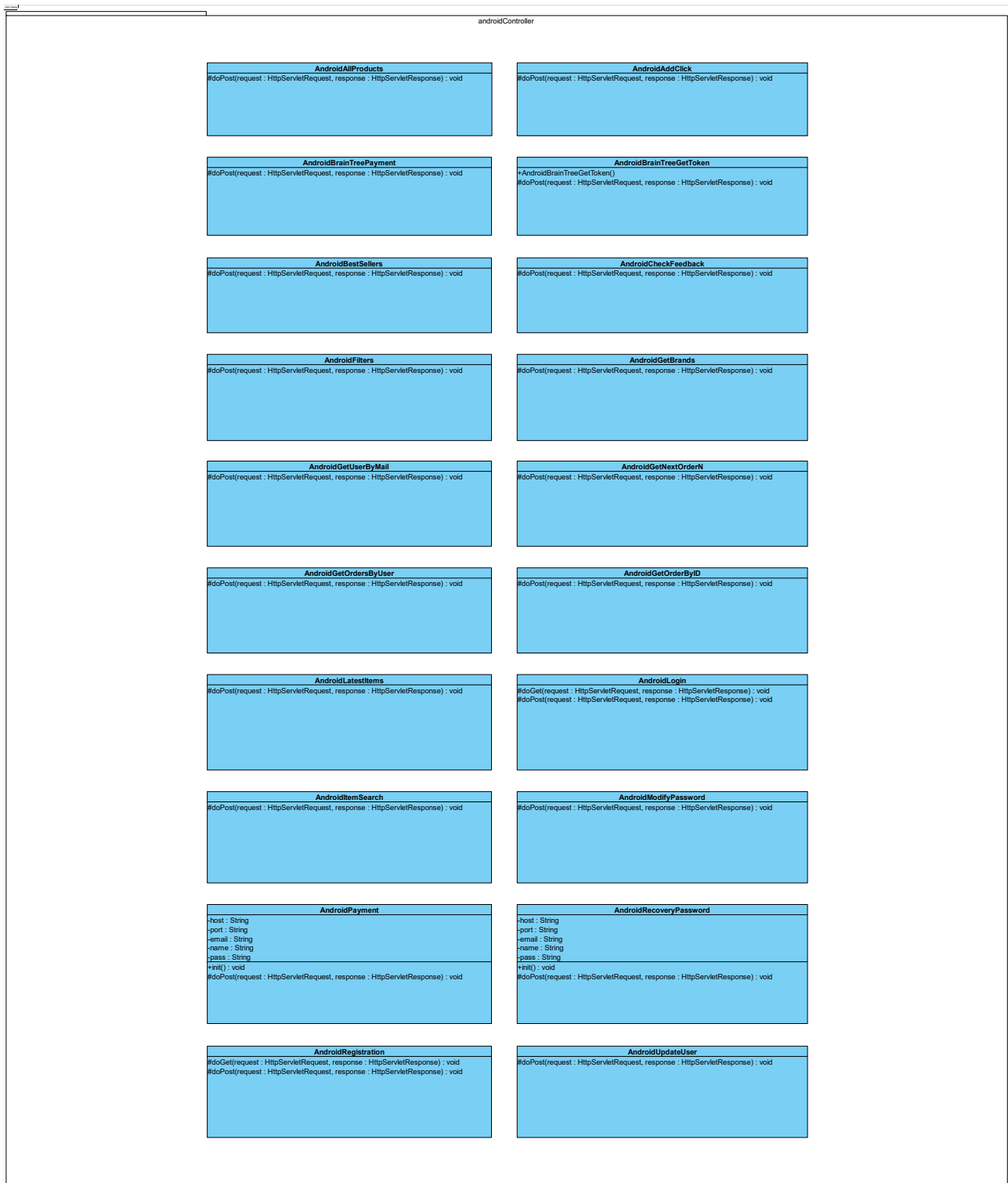
2.1.3.9 Braintree

Braintree è una divisione di PayPal specializzata nello sviluppo e nella pubblicazione di API per sistemi di pagamento per dispositivi mobile. L'integrazione con PayPal permette di effettuare pagamenti utilizzando misure di sicurezza sempre aggiornate e supporta anche le carte di credito. È diventato il metodo di pagamento predefinito di PayPal sui dispositivi mobile.

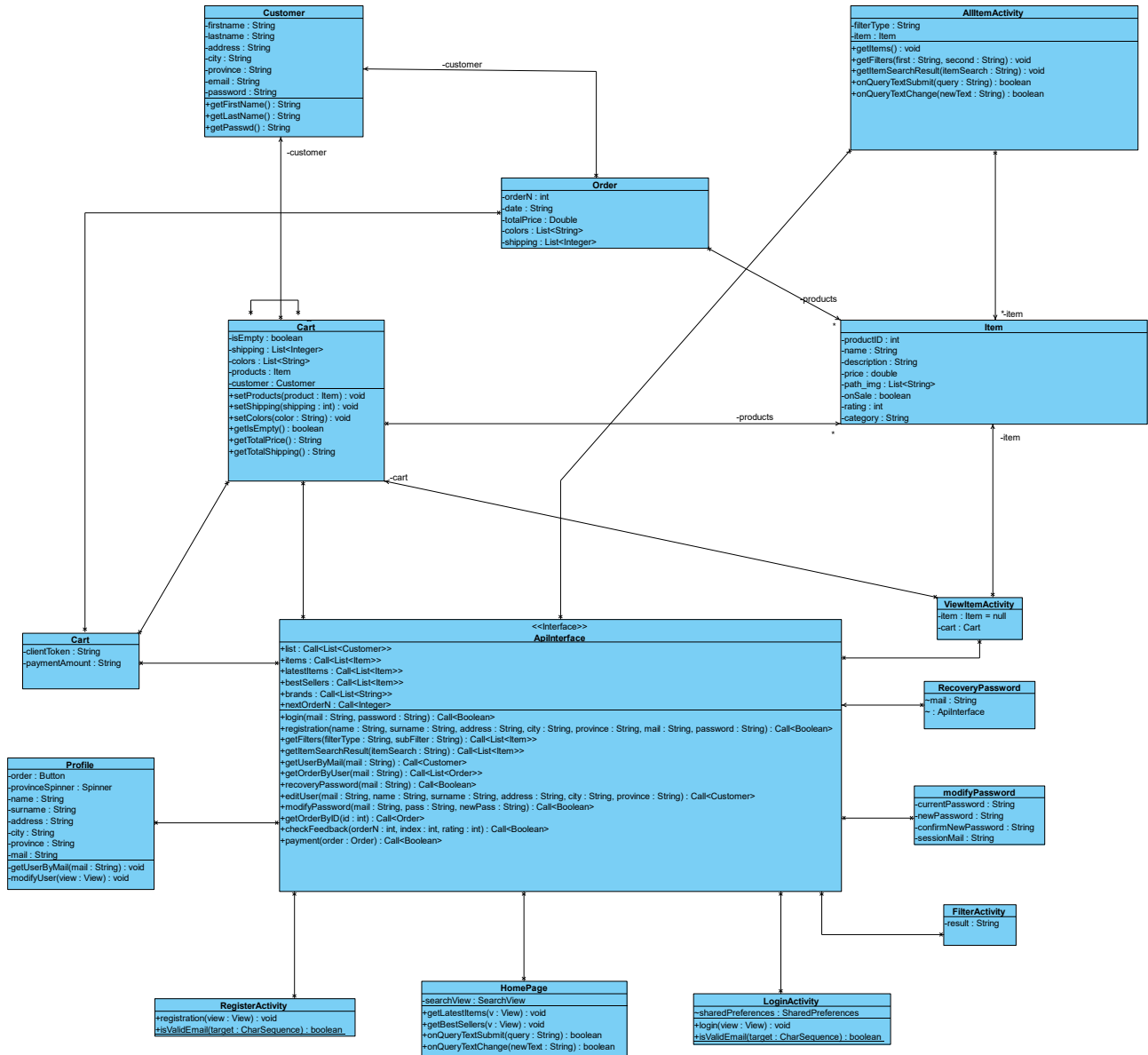
2.2 Diagramma delle classi di design







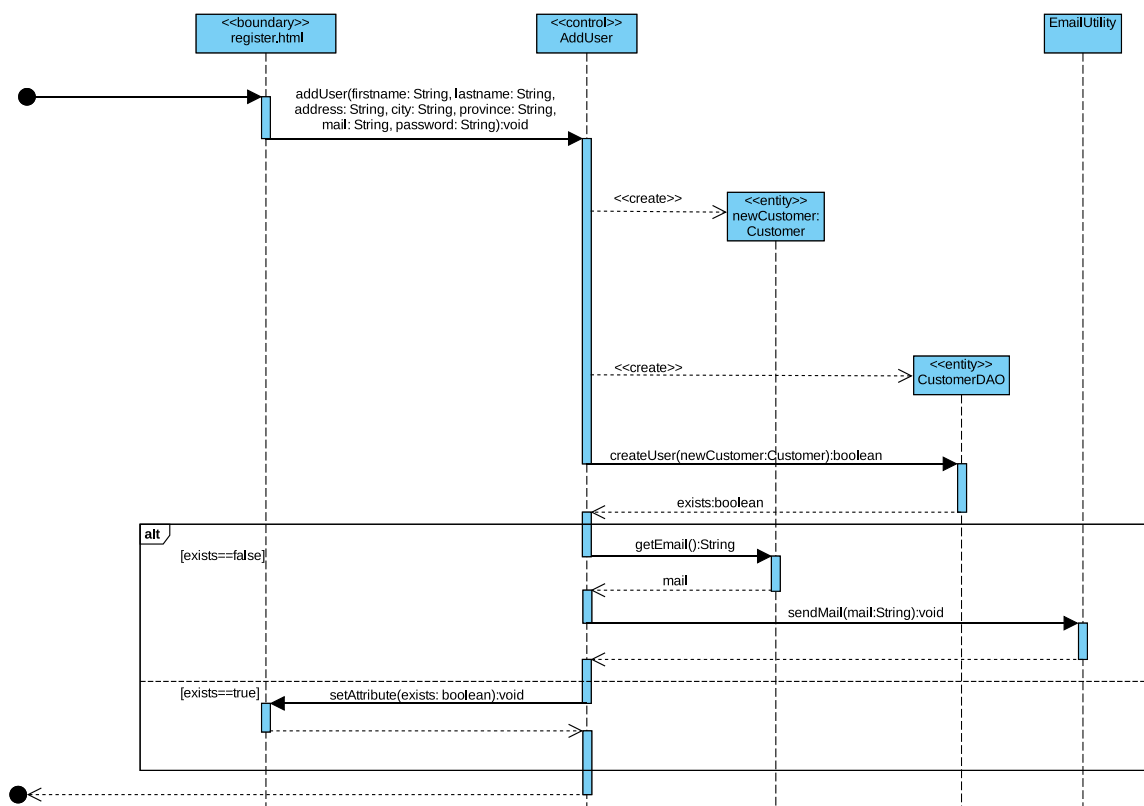
Applicazione



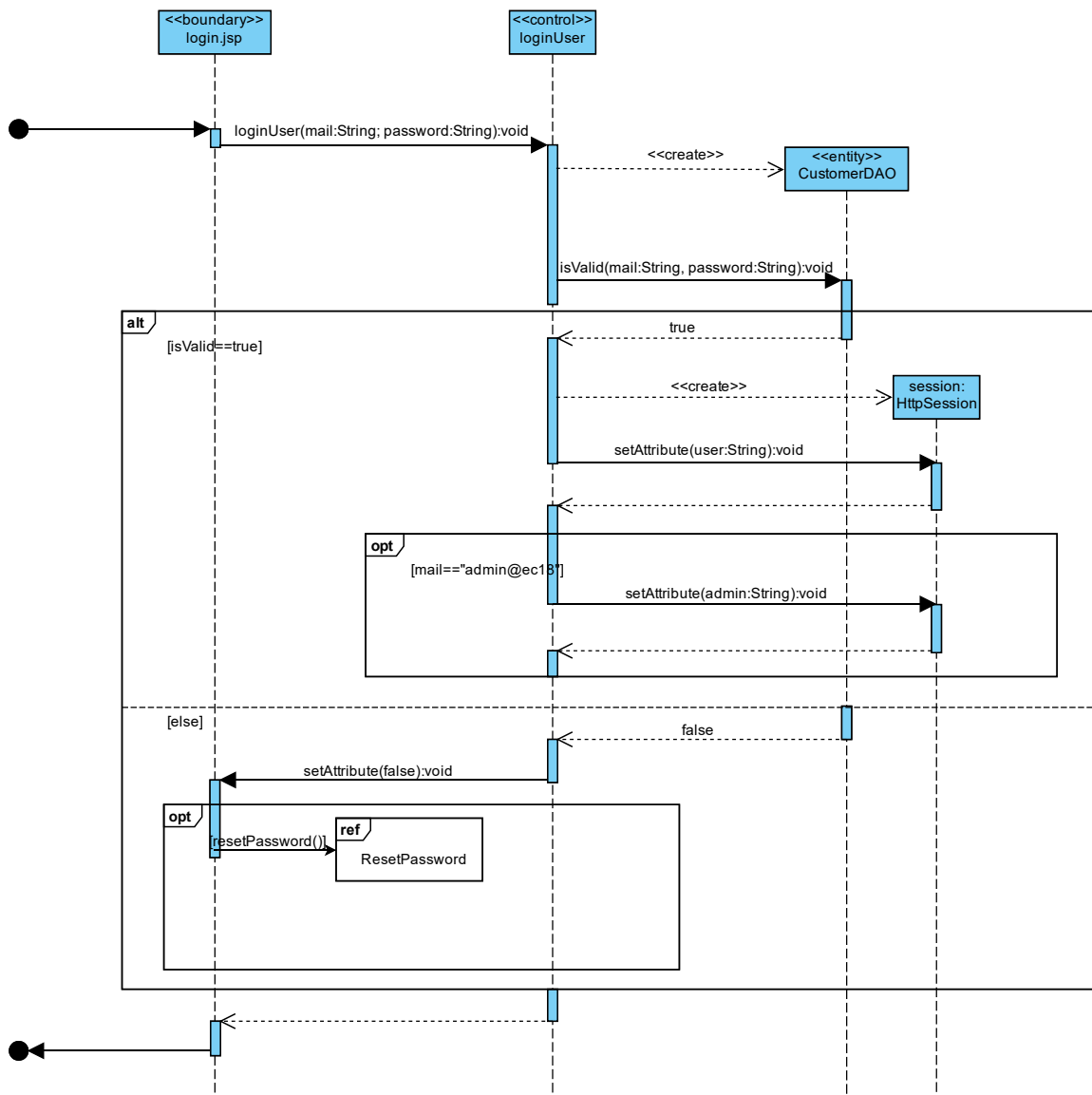
2.3 Diagrammi di sequenza di design

In questo capitolo verranno illustrati le sequenze di interazione tra i vari boundary controller e entity per definire il comportamento e il ruolo all'interno del sistema. A differenza del capitolo 1.3.2 qui vi sono passaggi più tecnici per definire il design del sistema.

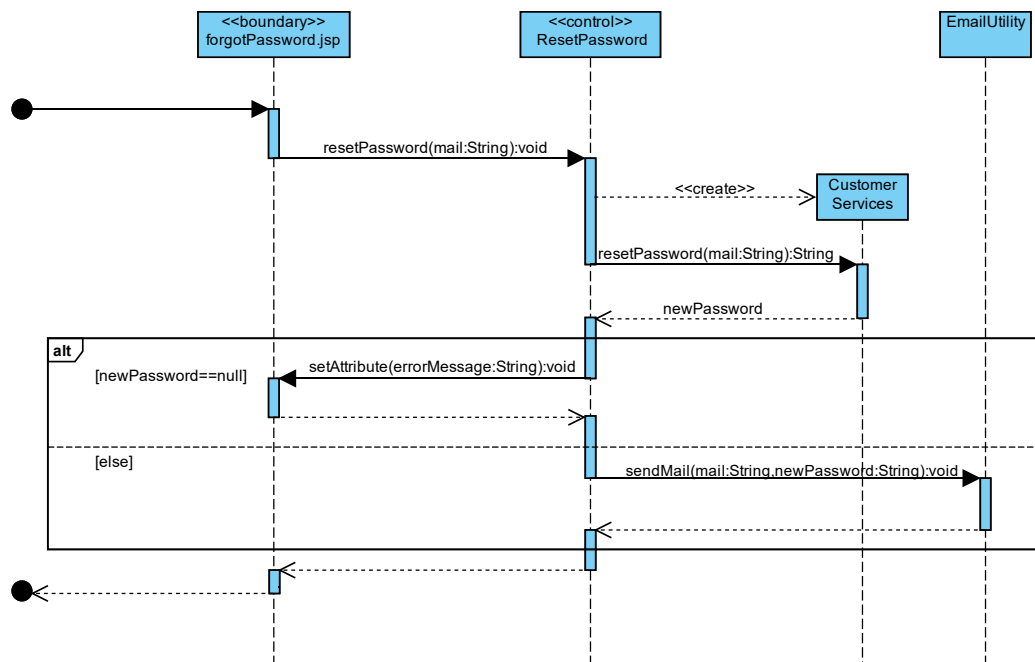
2.3.1 Sequence Diagram – Registrazione



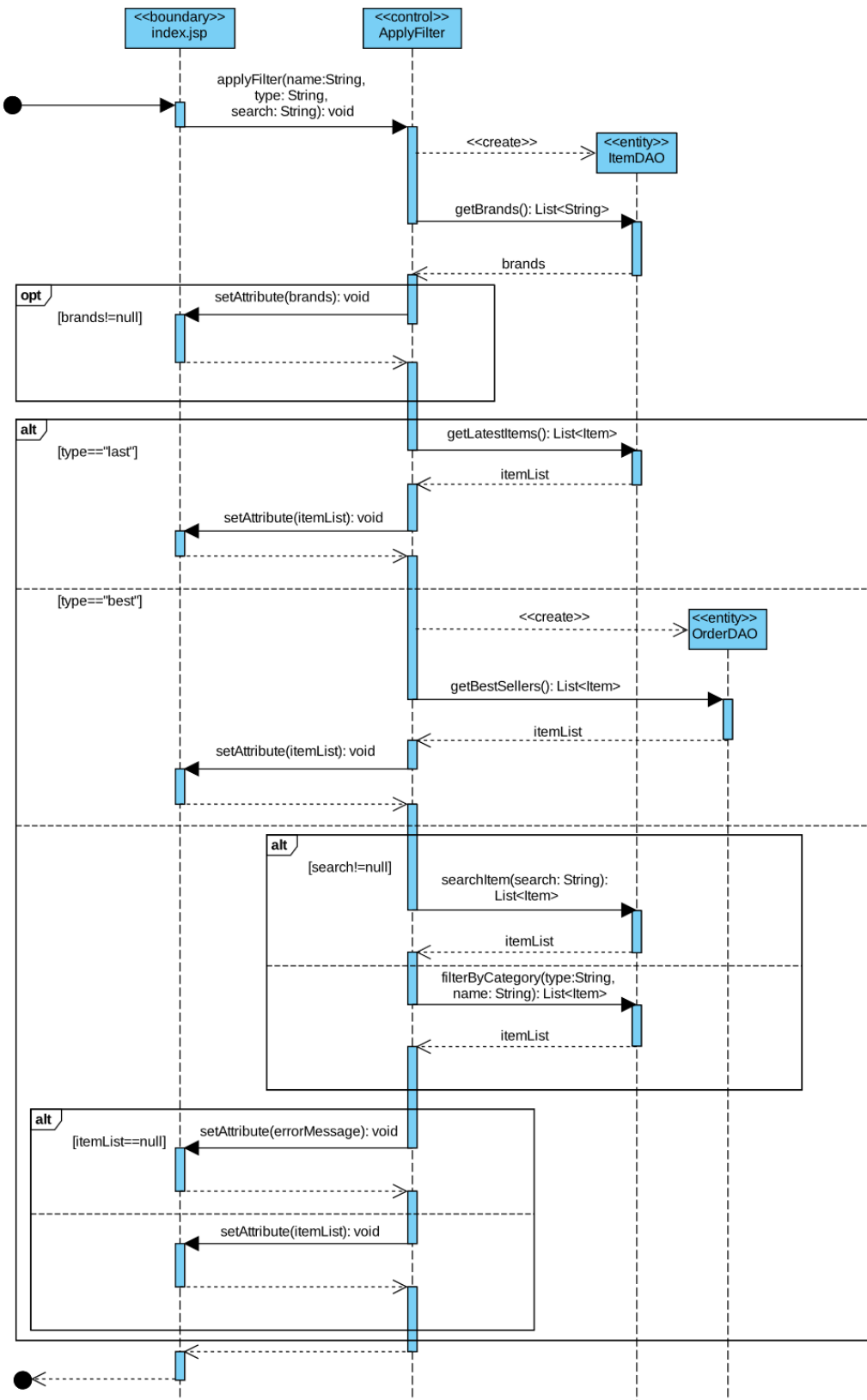
2.3.2 Sequence Diagram – Login



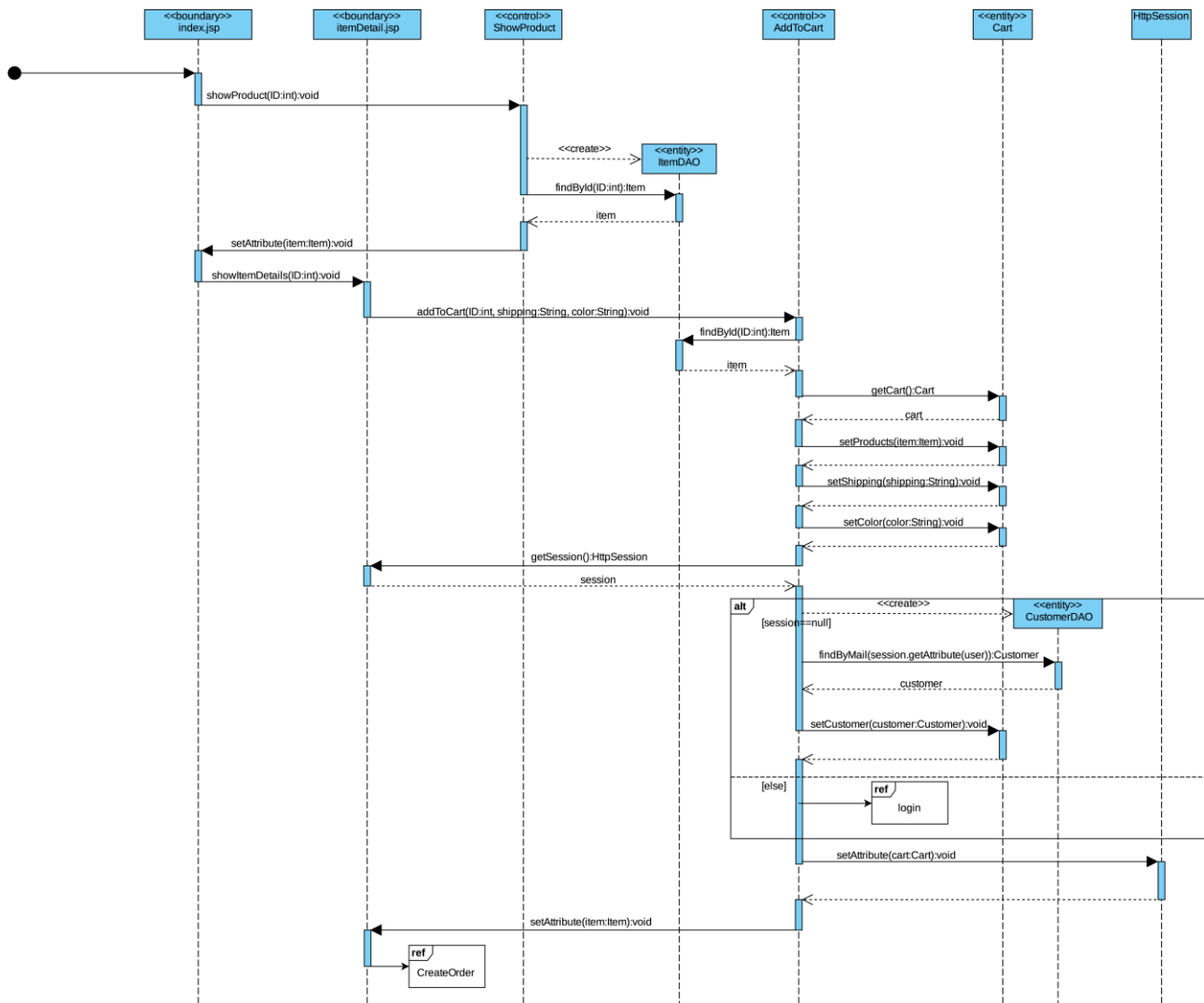
2.3.2.1 Sequence Diagram – Password dimenticata

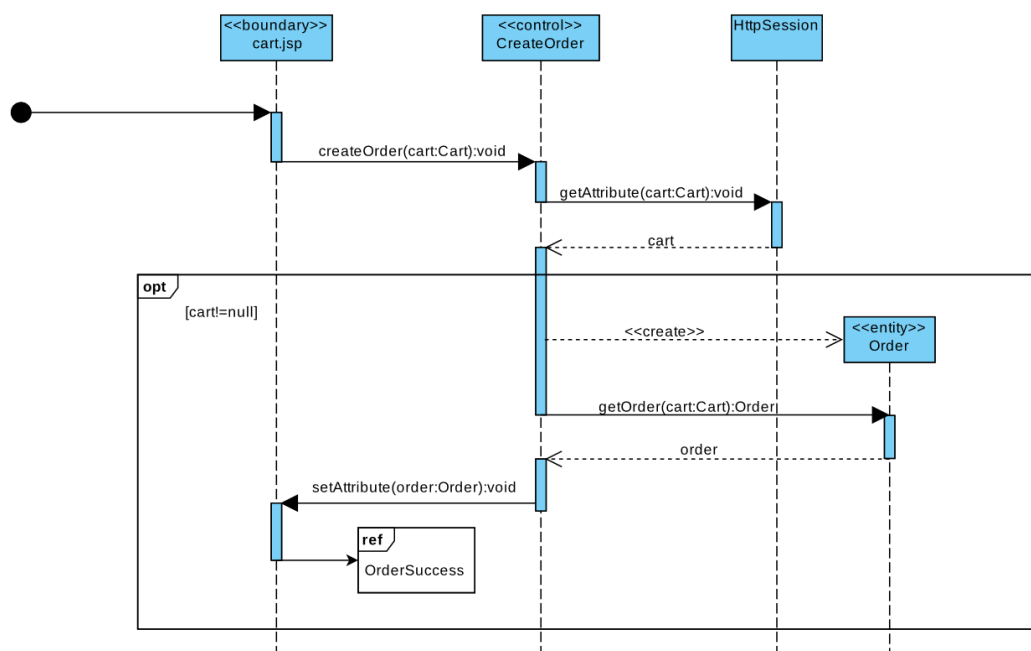
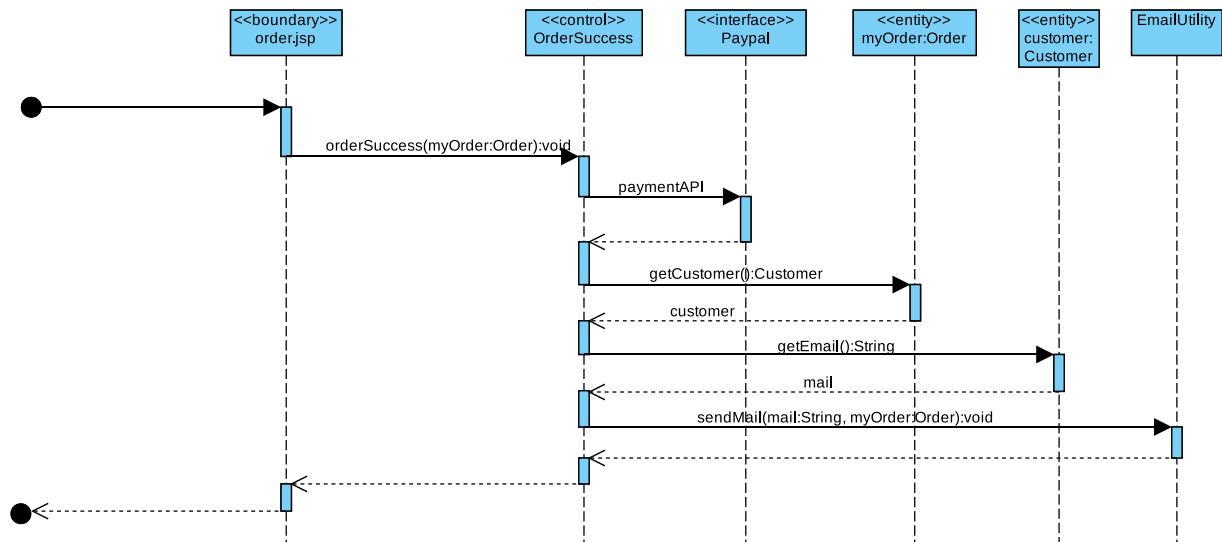


2.3.3 Sequence Diagram – Ricerca Prodotto

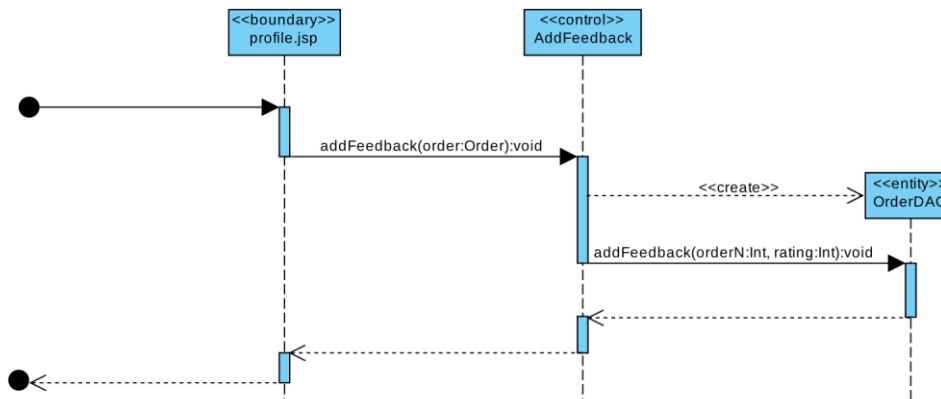


2.3.4 Sequence Diagram – Acquista Prodotto

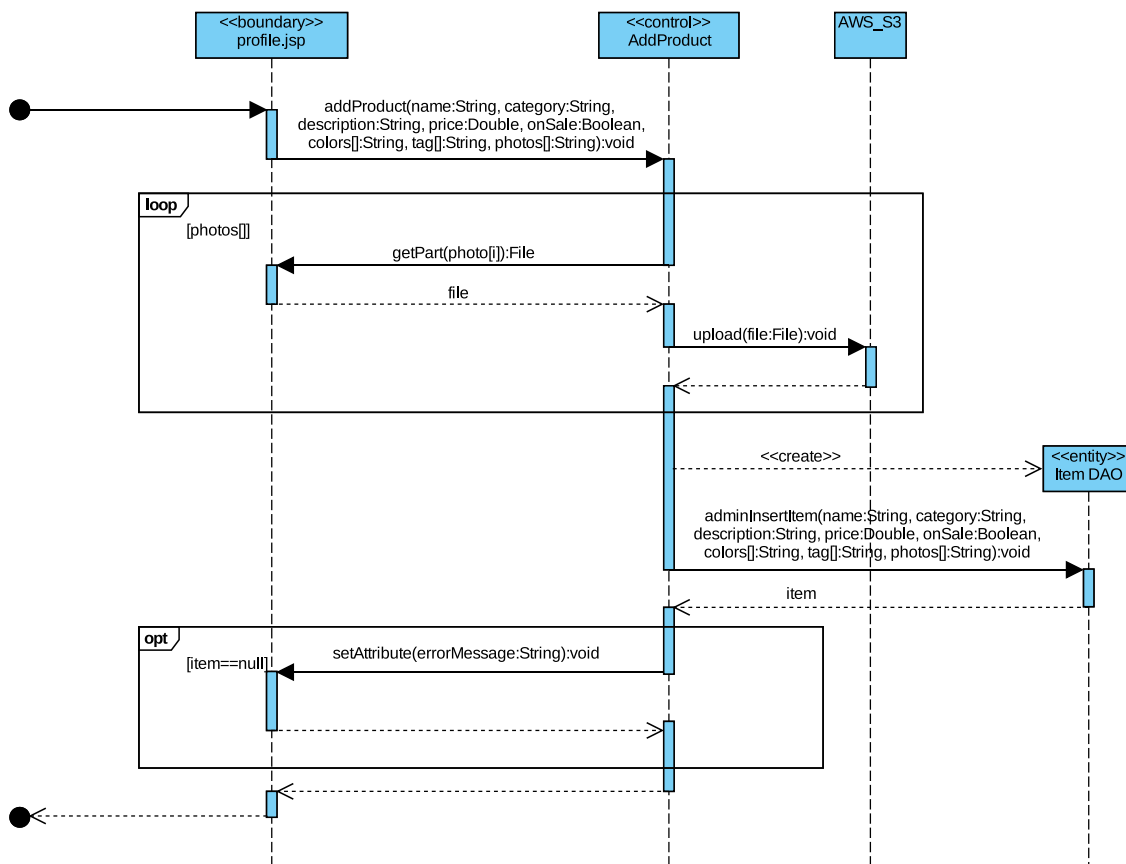




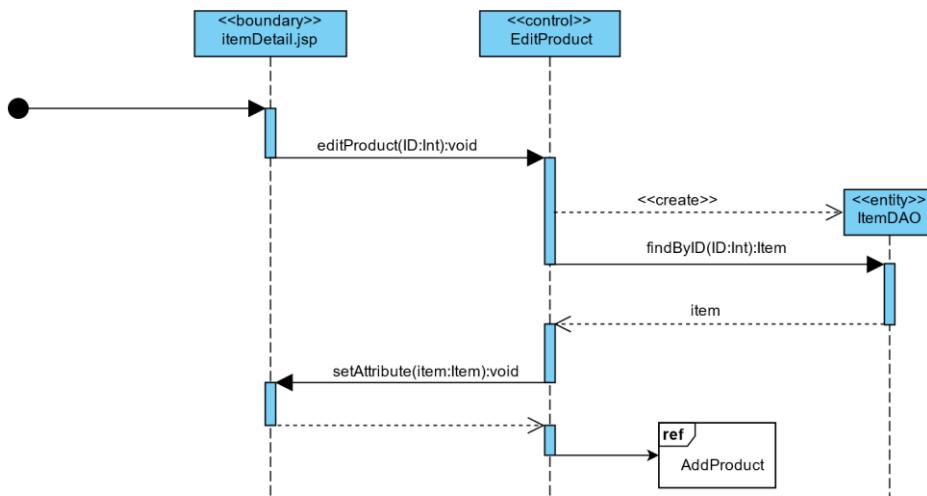
2.3.5 Sequence Diagram – Lascia Feedback



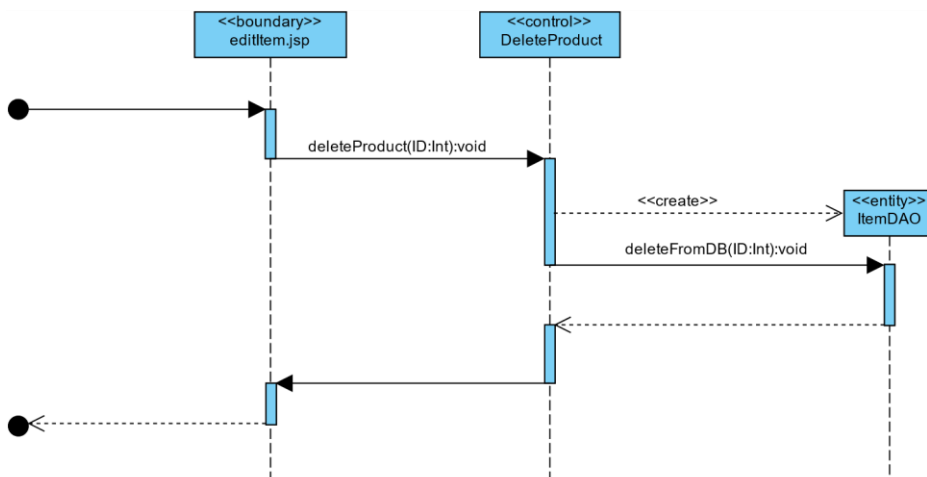
2.3.6 Sequence Diagram – Aggiungi Prodotto



2.3.7 Sequence Diagram – Modifica Prodotto

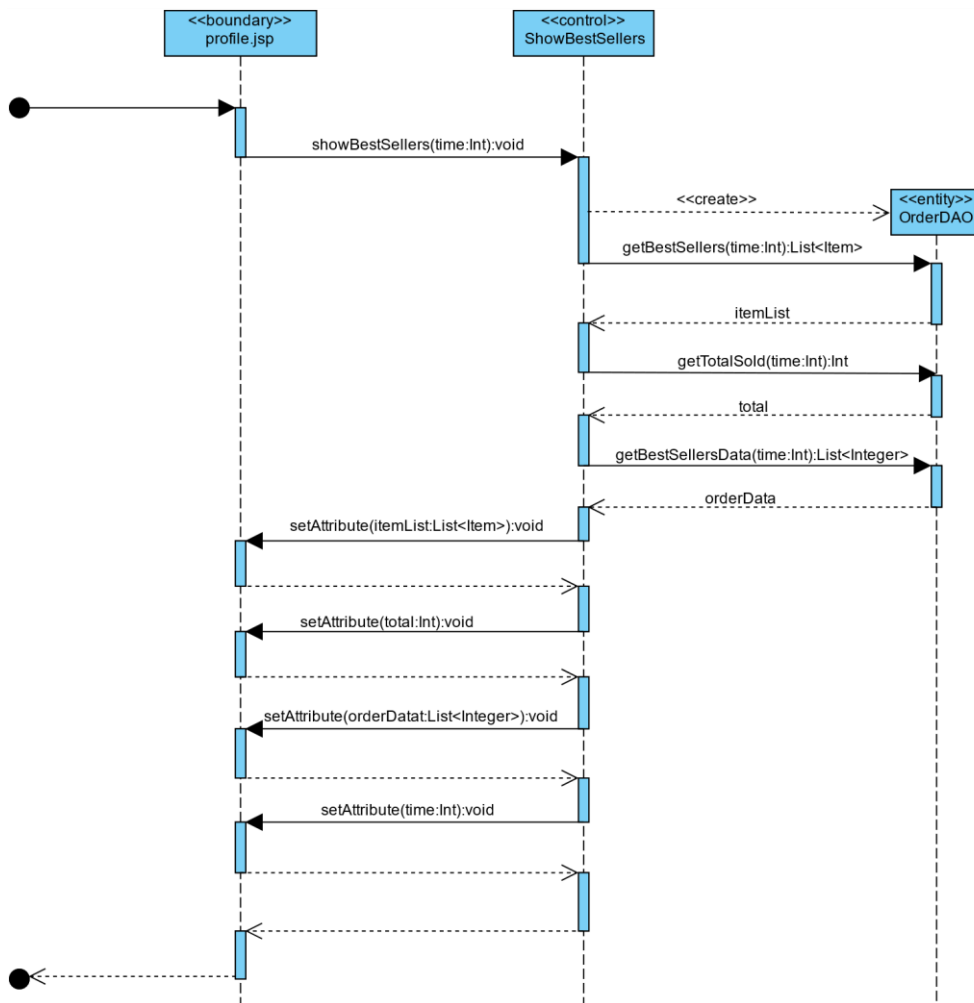


2.3.8 Sequence Diagram – Elimina Prodotto

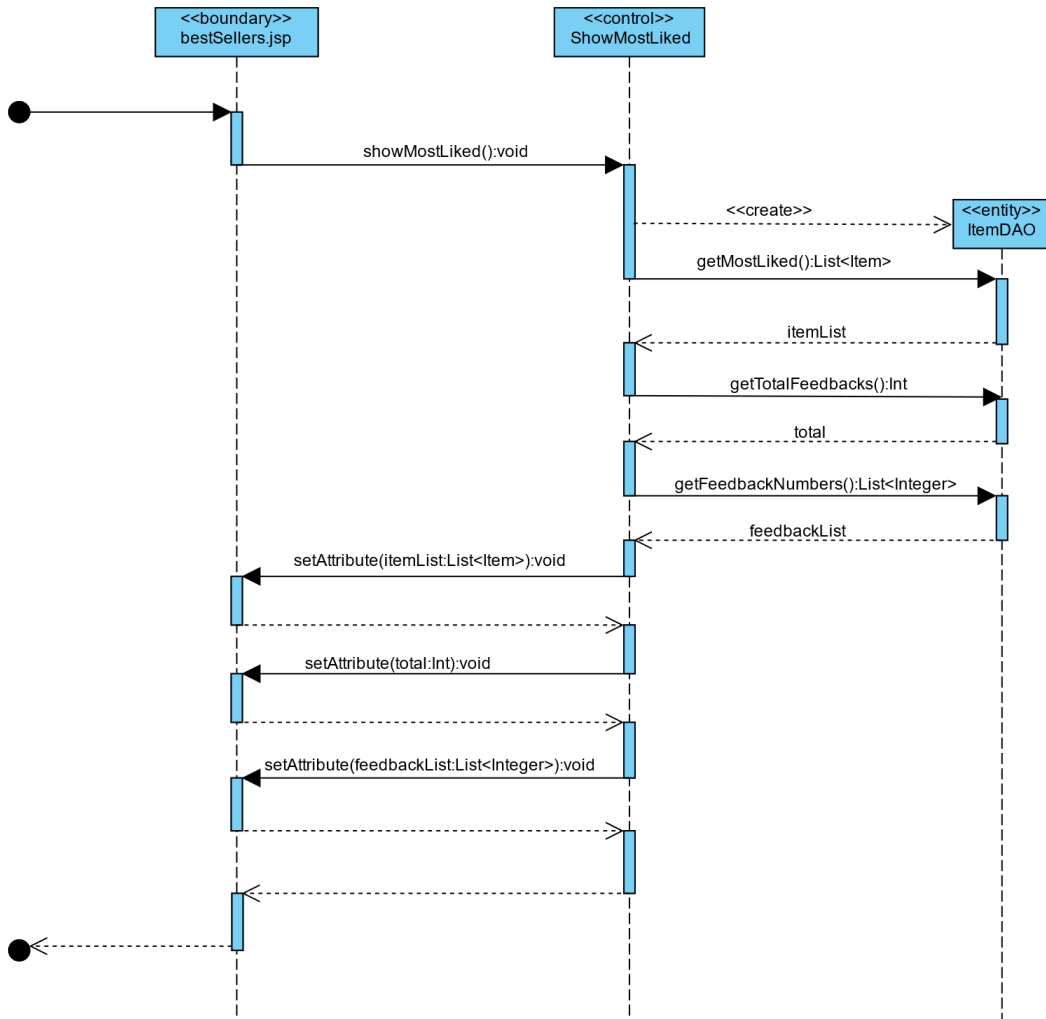


2.3.9 Sequence Diagram – Visualizzazione Statistiche

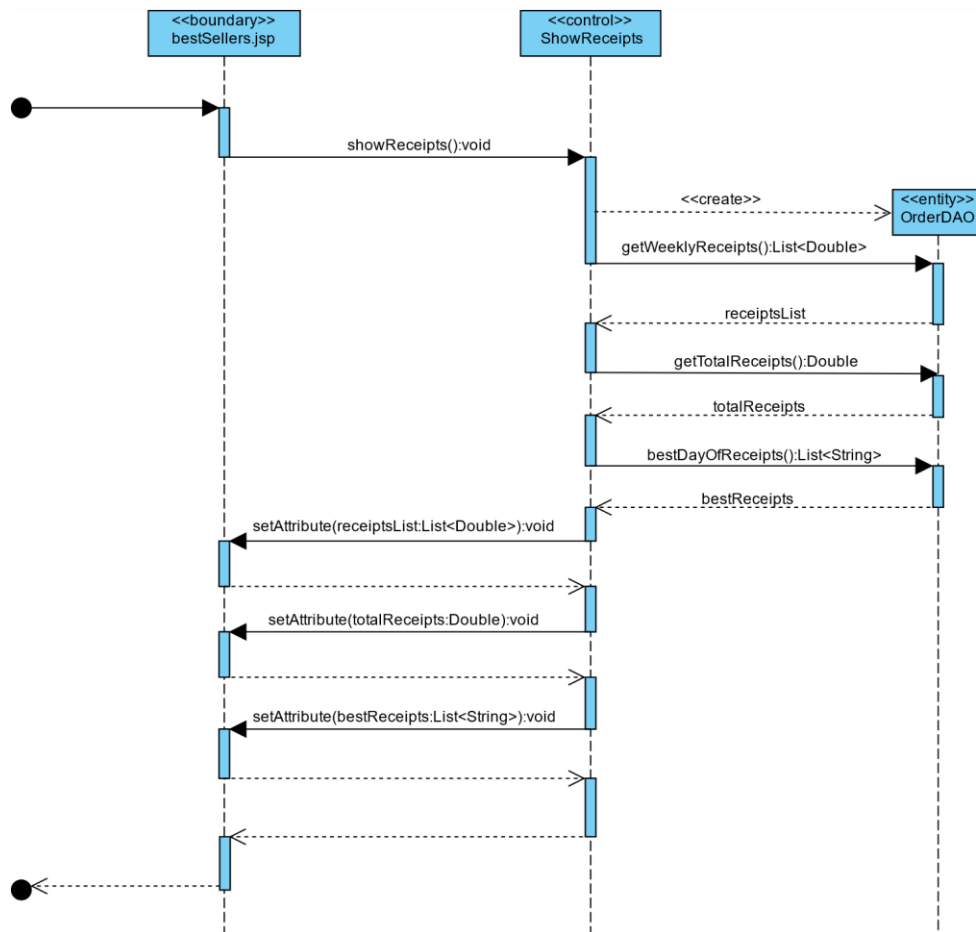
Articoli più venduti



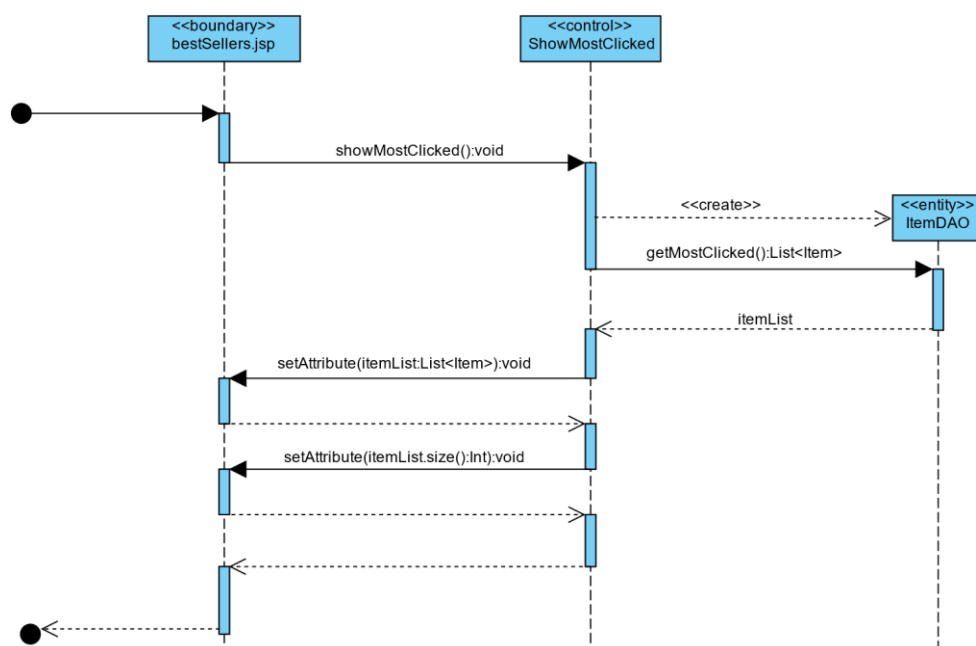
Articoli più piaciuti



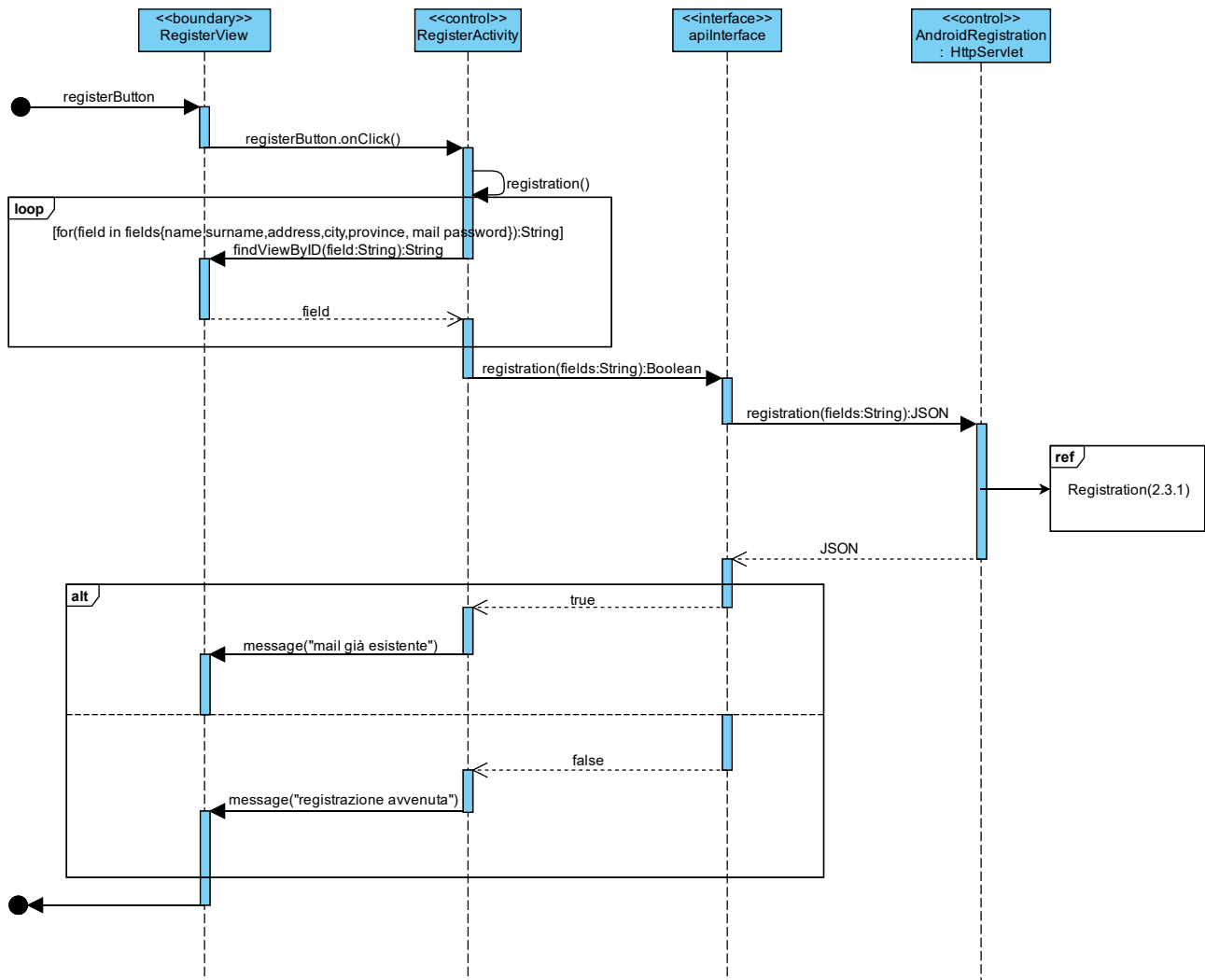
Incassi



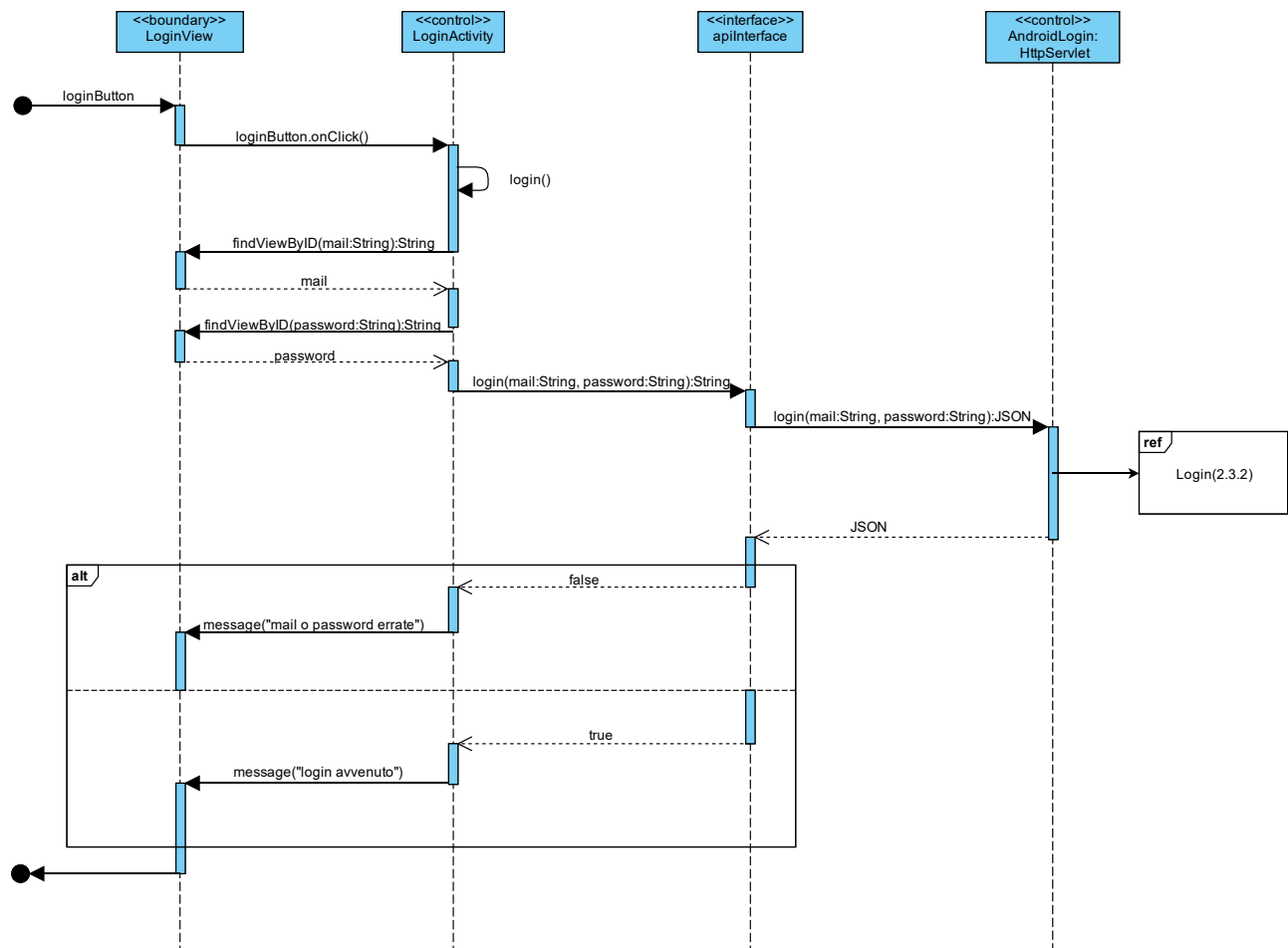
Più Cliccati



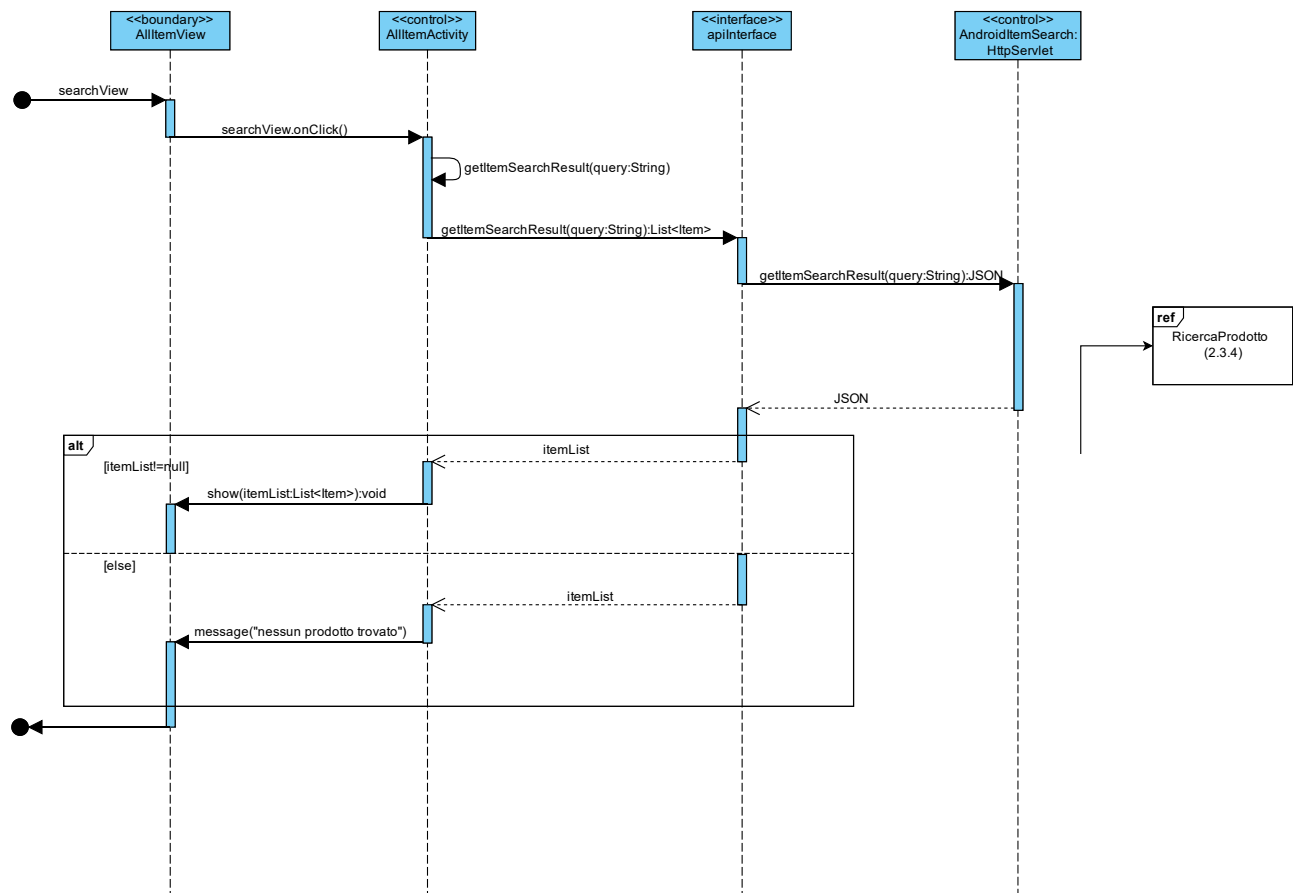
2.3.10 Sequence Diagram – Registrazione – APP



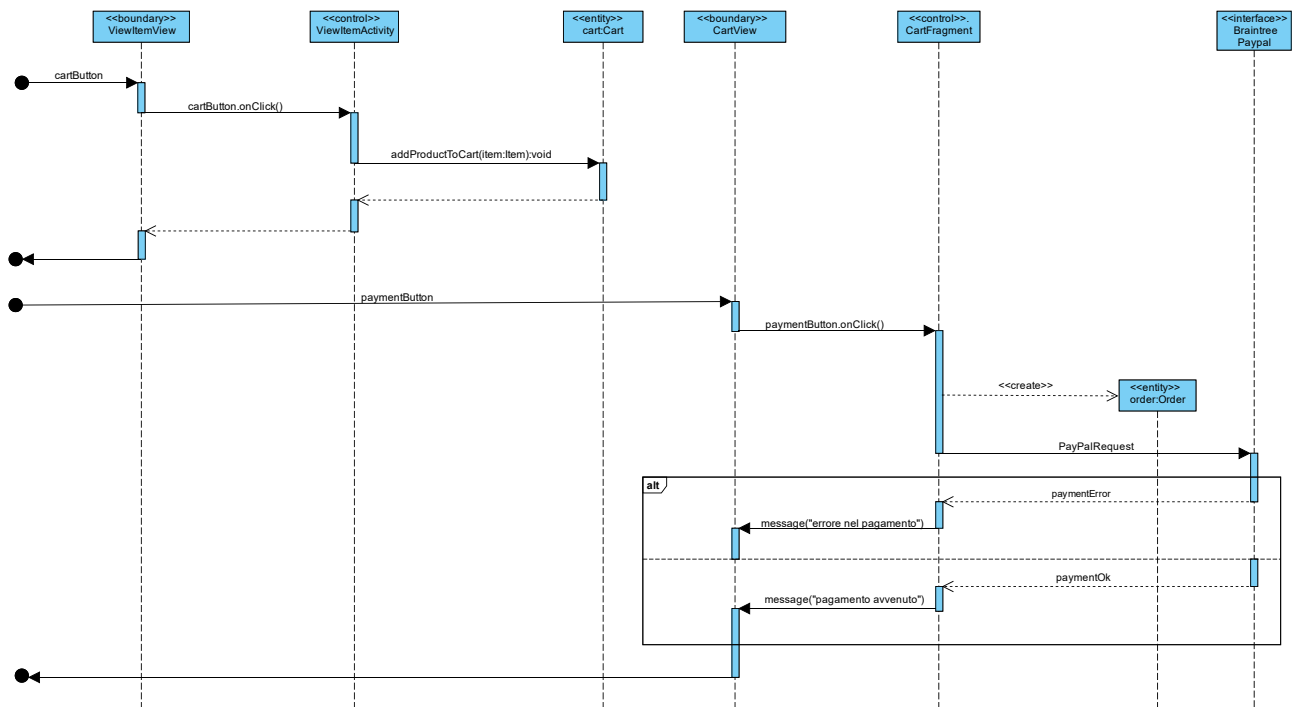
2.3.11 Sequence Diagram – Login – APP



2.3.12 Sequence Diagram – Ricerca prodotto – APP



2.3.13 Sequence Diagram – Acquista prodotto – APP



2.3.14 Sequence Diagram – Lascia feedback – APP

