Progetto Ingegneria del Software UNICT 2018/19

Puglisi Manlio
Spallina Alessandro

1 - FASE DI IDEAZIONE

1.1 - Visione

Copiga è un servizio di ottimizzazione per le copisterie, questo offre alle stesse la possibilità di registrarsi alla piattaforma ed essere raggiungibile da utenti che possono acquistare del credito presso una specifica copisteria e ordinare la stampa di un file piuttosto che l'acquisto di materiale di cartoleria. In una fase successiva l'utente può essere avvisato quando il materiale da egli richiesto è pronto e andare in copisteria a ritirare il tutto.

Il beneficio dell'utilizzo della piattaforma è condiviso tra le due parti che la utilizzano:

- La copisteria è in grado di accettare ordini telematici e riesce così a ridurre code di persone presenti in negozio al fine di ordinare una stampa.
- Il cliente della copisteria è in grado di ordinare stampe (e articoli di cartoleria) senza attese e la ricezione di un avviso per il ritiro.

L'acquisto di credito è specifico per ogni copisteria; l'utente può:

- Andare in copisteria a ricaricare il proprio conto direttamente in contanti.
- Ricaricare il proprio conto online via PayPal o carta di credito/debito.

Copiga permette alle copisterie di aggiungere in ogni momento elementi disponibili per essere stampati e articoli di cartoleria.

Inoltre, permette ai clienti della copisteria di inviare materiale stampabile, quali file testuali (word, pdf) o file immagini (jpg e png). Il pagamento viene effettuato nel momento in cui viene confermato l'ordine da parte della copisteria.

I costi del servizio di Copiga non sono riversati sugli utenti, ma sulle copisterie.
Il credito consumato dagli utenti per l'utilizzo dei servizi della copisteria verrà conteggiato e le fee del servizio Copiga saranno calcolate mediante diverse soglie di prezzo. Ad esempio, fino ad un guadagno (tramite la piattaforma) di 100 euro mensili il servizio è sempre gratuito, superando questa soglia la copisteria a fine mese dovrà versare una quota che varia in funzione dei volumi di credito utilizzati dagli utenti.
Le copisterie *early bird* avranno un credito bonus presso la piattaforma.

1.2 - Casi d’uso (formato breve)

Si considerino i seguenti casi d’uso, di cui è di interesse solo lo scenario principale di successo.

I casi d’uso CRUD per i tipi di file, tipi di carta e tipi di rilegatura sono stati volontariamente omessi perché ritenuti di analisi e progettazione semplice dato un linguaggio di programmazione.

Caso d’uso UC1: Registrazione nuova copisteria

Attore primario: proprietario della copisteria.

1) Il proprietario completa il form di registrazione.
2) Il sistema registra l’avvenuta compilazione del form.
3) Il sistema invia l’e-mail di conferma.
4) Il proprietario conferma la registrazione tramite link ricevuto via e-mail.

Caso d’uso UC2: Compilazione tariffario e specifiche di stampa della copisteria

Attore primario: proprietario della copisteria.

1) Il proprietario compila il tariffario specificando quali formati cartacei sono disponibili per la stampa e quali estensioni di file sono consentite.
2) Per ogni tipo di file vengono selezionate le varianti di stampa e i relativi prezzi:
   * Stampa a colori/in bianco e nero;
   * Formato stampa;
   * Tipo di rilegatura;
   * Frontespizio;
3) Il sistema registra le informazioni.

Caso d’uso UC3: Registrazione cliente

Attore primario: cliente della copisteria.

1) Il cliente completa il form di registrazione.
2) Il sistema registra l’avvenuta compilazione del form.
3) Il sistema invia l’e-mail di conferma.
4) Il cliente conferma la registrazione tramite link ricevuto via e-mail

Caso d’uso UC4: Acquisto credito copisteria in contanti

La registrazione comporta la possibilità di acquistare credito presso una specifica copisteria ed ottenerne i servizi.

Attore primario: addetto della copisteria.

Attore finale: cliente della copisteria.

1) Il cliente si reca presso la specifica copisteria.
2) Il cliente manifesta la volontà di volere acquistare credito presso la piattaforma.
3) L’addetto chiede l’identificativo univoco del sistema e la quantità di denaro che desidera venga convertita in credito.
4) Il cliente comunica l’identificativo e paga la somma desiderata.
5) L’addetto della copisteria seleziona il cliente sulla piattaforma tramite l’identificativo fornito e associa un credito pari alla somma pattuita.
6) Il cliente riceve una mail di avvenuta ricarica credito.
7) Il cliente va via con la fattura.

Caso d’uso UC5: Acquisto credito copisteria online

Attore primario: cliente della copisteria.

Attore di supporto: engine di pagamento online.

1) Il cliente seleziona la copisteria presso la quale vuole ricaricare il proprio credito.
2) Il cliente visita la sezione del portale relativa al proprio credito di una specifica copisteria.
3) Il cliente seleziona il metodo di pagamento che preferisce.
4) Il cliente inserisce l’ammontare di denaro che desidera venga convertito in credito.
5) L’Engine processa il pagamento.
6) Il sistema registra l’avvenuta azione con successo e notifica al cliente l’avvenuta ricarica.

Caso d’uso UC6: Invio file di stampa alla copisteria

Attore primario: cliente della copisteria.

Attore di supporto: addetto della copisteria

1) Il cliente seleziona una copisteria presso la quale ha un credito attivo.
2) Il cliente carica sulla piattaforma un file di tipo stampabile.
3) Il cliente seleziona le specifiche di stampa.
4) Il sistema calcola il costo dell’operazione.
5) Il cliente conferma l’ordine.
6) Il sistema detrae il credito dal cliente.
7) Fino a quando il task di stampa inviato dall’utente non viene approvato, l’utente ha la possibilità di modificare l’ordine o annullarlo.

Caso d’uso UC7: Accettazione task e invio richiesta di ritiro

Attore primario: addetto della copisteria.

Attore finale: cliente della copisteria.

1) L’addetto della copisteria accetta il task e invia alla coda di stampa il file.
2) L’addetto invia una notifica tramite la piattaforma al cliente quando il file è stato stampato ed è pronto per il ritiro.
3) Il cliente si reca in copisteria, ritira il materiale e va via.

1.3 - Glossario

- Proprietario/addetto/esercente della copisteria (sinonimi): svolgono lo stesso ruolo in quanto rappresentano il lato copisteria.
- Cliente/utente (sinonimi): utilizzatori della piattaforma lato consumer, coloro che richiedono i servizi alla copisteria.
- Engine di pagamento: sistema informatico capace di processare una transazione monetaria da parte di imprese quali Visa e PayPal. È possibile interfacciarsi con questo tramite le opportune API.
- Job/Task/Task di Stampa (sinonimi): indica la richiesta di stampa inviata dal cliente alla copisteria.

1.4 - Casi d’uso: formato esteso

I casi d’uso di seguito selezionati per essere descritti in maniera dettagliata rappresentano le attività chiave del servizio offerto da Copiga.

Caso d’uso dettagliato UC4: Acquisto credito copisteria in contanti

Portata: applicazione copiga lato copisteria;

Livello: obiettivo utente

Attore Primario: addetto della copisteria

Attore Finale: cliente della copisteria

Attore di Supporto: sistema di cassa

Parti interessate e Interessi:

- *Addetto della copisteria*: Vuole che venga acquistato credito al fine di ottenere anticipatamente i soldi che vengono versati preventivamente dai clienti. Vuole avere un sistema che gli consenta di incassare e caricare credito nell’account dell’utente in maniera semplice e veloce. Vuole un sistema che garantisca continuità di servizio, seppur in maniera limitata, anche in caso di assenza (momentanea) o malfunzionamento di rete internet.
- *Cliente della copisteria*: Vuole caricare il proprio credito sulla piattaforma in maniera veloce e sicura. Vuole avere prova dell’avvenuta ricarica. Vuole ricevere fattura.

Pre-condizioni: La copisteria è registrata presso Copiga ed ha effettuato le fasi di inizializzazione dell’attività commerciale presso la piattaforma. L’utente è registrato presso Copiga.

Garanzia di successo: La copisteria incassa e fattura sul momento. L’utente vede accreditata la somma versata.

Scenario principale di Successo:

1) Il cliente si reca presso la specifica copisteria.
2) Il cliente manifesta la volontà di volere acquistare credito presso la piattaforma.
3) L’addetto chiede l’identificativo univoco dell’utente e la quantità di denaro che desidera convertire in credito.
4) Il cliente comunica l’identificativo e paga la somma desiderata.
5) L’addetto seleziona il cliente sulla piattaforma tramite l’identificativo fornito.
6) L’addetto associa sul sistema un credito pari alla somma pattuita.
7) L’addetto incassa la somma ed emette la fattura.
8) Il cliente ritira la fattura.
9) Il cliente riceve una mail di avvenuta ricarica credito.

Flussi alternativi:

5ª) L’utente selezionato non è presente sul sistema.

1a- L’addetto richiede all’utente di registrarsi alla piattaforma col proprio dispositivo

1b- L’addetto concede al cliente l’utilizzo di un dispositivo della copisteria per effettuare la registrazione

2- Proseguire dal punto 6 dello scenario di successo.

6ª) Il servizio internet della copisteria è assente.

1- Il sistema mostra su schermo un avviso che notifica la mancanza di connessione

2- L’addetto continua con la registrazione del credito in modalità offline.

3- L’addetto comunica al cliente che a causa del disservizio l’email di confermata ricarica gli verrà recapitata non appena il servizio internet verrà ristabilito.

4- Proseguire dal punto 7 dello scenario di successo.

Requisiti speciali:

- Dall’interfaccia lato copisteria, la sezione di ricarica credito deve essere raggiungibile in massimo due click. La ricerca dell’identificativo utente deve essere coadiuvata da un meccanismo di autocompletamento del campo di ricerca.
- Il più delle volte la convalida dell’accredito e la ricevuta via email devono avvenire entro 5 minuti. In caso di servizio internet non disponibile, le informazioni vengono memorizzate in locale e inviate al server centrale non appena il servizio internet viene ristabilito, attraverso un sistema che si aggiorna automaticamente.

Frequenza di ripetizione: Potenzialmente ininterrotta.

Problemi aperti: Esaminare in maniera dettagliata le questioni burocratico-fiscali.

Caso d’uso dettagliato UC6: Invio file di stampa alla copisteria

Portata: applicazione Copiga lato cliente

Livello: obiettivo utente

Attore Primario: cliente della copisteria

Attore di Supporto: addetto della copisteria

Parti Interessate e Interessi:

- *Cliente della copisteria*: Vuole evitare le code della copisteria per risparmiare tempo inviando da remoto un job di stampa.
- *Addetto della copisteria*: Vuole ottimizzare i tempi per poter mandare in stampa quanti più documenti possibile. Vuole aumentare la propria richiesta. Vuole essere sicuro di non scaricare malware mascherati da documenti stampa.

Pre-condizioni: La copisteria è registrata presso Copiga ed ha effettuato le fasi di inizializzazione dell’attività commerciale presso la piattaforma. L’utente è registrato presso Copiga. L’utente deve essere in possesso di credito presso la copisteria scelta.

Garanzia di successo: Fino a quando il task di stampa inviato dall’utente non viene approvato, l’utente ha la possibilità di modificare l’ordine o annullarlo. Il cliente vede consegnati immediatamente i propri documenti da stampare alla copisteria. Il cliente può andare a ritirare le stampe senza portare con sé il denaro. Tempi di attesa ridotti.

Scenario principale di Successo:

1) Il cliente visualizza le copisterie presso le quali ha un credito attivo.
2) Il cliente seleziona una specifica copisteria.
3) Il cliente carica sulla piattaforma un file di tipo stampabile.
4) Il sistema scansiona il file caricato per scovare eventuali malware e ne controlla la genuinità.
5) Il cliente seleziona le specifiche di stampa:
   * Fronte-retro / solo fronte
   * Formato di foglio
   * N pagine (del file originale) per pagina (stampata)
   * Colore / bianco e nero
   * Tipo rilegatura
6) Il sistema calcola il costo dell’operazione.
7) Il cliente conferma l’ordine.
8) Il sistema detrae il credito dal cliente.

Flussi alternativi:

1ª) Il cliente ha selezionato una copisteria presso la quale non possiede credito attivo.
   1- Il cliente viene rimandato ad una schermata che lo invita a ricaricare il suo credito online.
   2a- Il cliente viene rimandato alla pagina della copisteria precedentemente selezionata.

2b- Il cliente sceglie di non ricaricare il suo credito.
3a- Proseguire dal punto 2 dello scenario di successo.
3b- Proseguire dal punto 1 dello scenario di successo.

2ª) La copisteria selezionata è momentaneamente fuori servizio.
1- Al cliente viene notificato che il servizio della copisteria è momentaneamente interrotto.
2a- Il cliente sceglie comunque di caricare i documenti che vuole siano stampati.
2b- Il cliente cambia copisteria.
3a- Proseguire dal punto 3 dello scenario di successo.
3b- Proseguire dal punto 1 dello scenario di successo.

3ª) Il sistema scopre impurità nel documento caricato.
1- Al cliente viene notificato che il documento caricato non risulta essere valido.
2- Proseguire dal punto 2 dello scenario di successo.

Requisiti speciali:

- Il file deve poter essere caricato sia tramite apposita procedura di selezione file su file system, sia tramite sistema drag and drop.

Frequenza di ripetizione: Potenzialmente ininterrotta.

Problemi aperti: Trovare un sistema di antivirus. Trovare un metodo per consentire al cliente di annullare l’ordine o modificarlo prima che questo venga accettato dalla copisteria.

Caso d’uso dettagliato UC7: Accettazione task e invio richiesta di ritiro

Portata: applicazione Copiga lato copisteria;

Livello: obiettivo utente

Attore Primario: addetto della copisteria

Attore Finale: cliente della copisteria

Attore di Supporto: sistema di stampa

Parti interessate e Interessi:

- *Addetto della copisteria*: Vuole un sistema che gli notifichi l’avvenuta ricezione di una nuova richiesta di job di stampa. Vuole ricevere in maniera sicura il materiale da stampare e vuole poterlo visionare per verificare che la richiesta sia consona. Vuole poter accettare il task e tenere traccia facilmente dei task completati e da completare, del materiale già ritirato e in ritiro. Vuole un sistema in grado di tenere traccia del consumo del credito utente derivante dall’utilizzo dei servizi della copisteria. Vuole un sistema che garantisca continuità di servizio, seppur in maniera limitata, anche in caso di assenza (momentanea) o malfunzionamento di rete internet.
- *Cliente della copisteria*: Vuole essere avvisato alla conferma del task inviato. Vuole essere avvisato non appena il materiale richiesto è pronto per essere ritirato.

Pre-condizioni: La copisteria è registrata presso Copiga ed ha effettuato le fasi di inizializzazione dell’attività commerciale presso la piattaforma. Sul sistema risultano presenti utenti che hanno credito attivo presso la copisteria designata.

Garanzia di successo: Il task è stato accettato e il materiale stampato è pronto per il ritiro da parte dell’utente.

Scenario principale di Successo:

1) L’addetto della copisteria riceve notifica di avvenuta ricezione di un nuovo task.
2) L’addetto visiona il file.
3) L’addetto accetta il task.
4) Il sistema registra il consumo del credito utente presso la copisteria.
5) L’addetto invia il materiale in stampa.
6) L’addetto prepara il materiale stampato per la consegna.
7) L’addetto conferma sulla piattaforma la possibilità di ritiro del materiale.
8) Il sistema invia una notifica al cliente designato.

Flussi alternativi:

3ª) Il materiale ricevuto per il task non è consono (es. contiene materiale coperto da copyright).
    1- L’addetto rifiuta il task allegando un messaggio di motivazione del task rifiutato.

Requisiti speciali:

- Dall’interfaccia lato copisteria, la sezione di gestione dei task deve essere raggiungibile in massimo due click. La ricezione di nuovi task di stampa deve essere notificata in maniera da richiamare l’attenzione dell’addetto (effetti audio-visivi). Il task deve essere accettato in massimo due click. La notifica di possibilità di ritiro deve avvenire in massimo due click.
- La notifica di possibilità di ritiro lato utente deve richiamare l’attenzione del cliente (effetti audio-visivi) se connesso alla piattaforma. La notifica deve essere inviata anche via e-mail.

Frequenza di ripetizione: Potenzialmente ininterrotta.

1.5 – Regole di dominio

Formato di carta, stampa in bianco e nero o a colori, scelta del numero di pagine per foglio e il tipo di rilegatura influiscono sul calcolo del prezzo.

Prezzo Totale = {[(n° pagine file)/(coeff)*(pagine per lato)]*(prezzo singola pagina)} + prezzo rilegatura

Dove:

coeff = 2 se la stampa è “fronte-retro”, 1 se “solo fronte”

prezzo singola pagina = formato carta + prezzo "BN" o prezzo "a Colori" + n° di pagine per foglio

2 – PRIMA ITERAZIONE

2.1 – Piano di Architettura Software

A questo punto del processo di sviluppo abbiamo già un idea dell'architettura da utilizzare.
Copiga ammette due "lati": lato cliente e lato esercente.
Volendo garantire una maggiore accessibilità per gli utenti, il lato cliente viene sviluppato come una web app; il lato esercente potrebbe anch'esso risiedere all'interno di una web app, ma per questione di robustezza alla mancanza di connessione internet e questioni relative a futuri servizi aggiuntivi per gli esercenti si opta di sviluppare un software Java che verrà eseguito su un computer all'interno della copisteria.
La web app sarà sviluppata in PHP utilizzando il framework Laravel.

La prima azione con il nostro servizio, compiuta sia da utenti che esercenti, dovrà essere una registrazione, questa in entrambi i casi verrà effettuata sulla web app. Poiché la compilazione del tariffario (UC2) è un'azione compita dall'esercente una tantum, riteniamo di offrire questa possibilità sulla web app, stesse considerazioni valgono per la ricarica del credito ai clienti che intendono farlo in contanti. Gli altri casi d'uso con attore primario l'esercente, invece, verranno implementati sul client Java in esecuzione sul computer della copisteria. L'interazione tra la web app che farà uso di un database MySQL e il client Java avverrà tramite REST API.

Nella fase di elaborazione di questa iterazione, in parallelo all'attività di analisi verso la progettazione, si verificherà la validità dell'architettura appena proposta e verrà abbozzata la struttura software del progetto, esempi di ciò sono rappresentati dai seguenti interrogativi:

- Il framework Laravel consente una gestione agevole della doppia registrazione/login di utente ed esercente?
- Il framework Laravel consente una gestione agevole di REST API autenticate tali da interfacciarsi con i dati relativi agli esercenti? Java è la scelta giusta per il client lato esercente che deve implementare REST API?

2.2 - Casi d’uso: formato esteso

Adesso riteniamo che i casi d’uso non ancora descritti in maniera estesa necessitino di essere analizzati in maniera più approfondita.

Caso d’uso esteso UC1: Registrazione nuova copisteria

Portata: applicazione copiga lato copisteria;

Livello: obiettivo utente

Attore Primario: proprietario della copisteria

Parti interessate e Interessi:

- *Proprietario della copisteria*: Vuole accedere al servizio offerto dalla piattaforma.

Pre-condizioni: Nessuna

Garanzia di successo: La copisteria è registrata alla piattaforma, prima di essere raggiungibile da utenti deve compilare è richiesta la compilazione del tariffario come descritto in UC2.

Scenario principale di Successo:

1) Il proprietario della copisteria accede all’apposita sezione.
2) Il proprietario inserisce:
   * Nome della copisteria
   * Indirizzo della copisteria
   * Partita Iva
   * E-mail
   * Password
   * Recapito telefonico
3) Il sistema verifica l’univocità dell’e-mail.
4) Il sistema registra l’avvenuta compilazione del form.
5) Il sistema invia l’e-mail di conferma.
6) Il proprietario conferma la registrazione tramite link rivevuto via e-mail.

Flussi alternativi:

3ª) L’indirizzo e-mail è già registrato
    1a- Il sistema chiede se si vuole recuperare la password
    1b- Il sistema chiede di cambiare e-mail per la registrazione corrente
    2- Proseguire dal punto 2 dello scenario di successo.

Problemi aperti: Sviluppare un proof of concept per l’invio di e-mail con la tecnologia scelta.

Caso d’uso dettagliato UC2: Compilazione tariffario e specifiche di stampa della copisteria

Nota: per lo scenario principale di successo al punto 2 di questo caso d’uso stiamo in questa sede assumendo come unici file accettati formati di tipo immagine e testuale, questa sezione potrebbe apparire inutile, ma si ritiene di prevederla poiché in futuro potrebbero essere aggiunte anche altre voci (quali file progetto di stampe 3d o file plot) che la copisteria potrebbe non trattare.

Portata: applicazione Copiga lato copisteria

Livello: obiettivo utente

Attore Primario: proprietario della copisteria

Parti Interessate e Interessi:

- *Proprietario della copisteria*: Vuole inizializzare il proprio account di tipo copisteria per essere raggiungibile dagli utenti.

Pre-condizioni: La copisteria è registrata presso Copiga.

Garanzia di successo: La copisteria è raggiungibile dagli utenti.

Scenario principale di Successo:

1) Il proprietario della copisteria accede all’apposita sezione.
2) Il proprietario seleziona le tipologia di file che tratta:
   * Immagini
   * File di testo
3) Il proprietario definisce le specifiche di stampa offerte;
   Per ogni formato di carta:
   a) Seleziona il nome formato
   b) Inserisce il prezzo copia in bianco e nero
   c) Inserisce il prezzo copia a colori.
4) Il proprietario inserisce le tipologie di rilegatura;
   Per ogni rilegatura inserisce:
   a) Tipo di rilegatura
   b) Prezzo.

5) Il proprietario salva la configurazione.
6) Il sistema restituisce il messaggio di confermata registrazione del nuovo tariffario che mostra il riepilogo del tariffario inserito.

Requisiti speciali:

- L'interfaccia di selezione specifiche deve essere user friendly.

Frequenza di ripetizione: Una tantum.

Caso d'uso UC3: Registrazione cliente

Portata: applicazione Copiga a livello cliente

Livello: obiettivo utente

Attore primario: cliente della copisteria

Parti interessate e interessi:

- *Cliente della copisteria*: Vuole accedere al servizio offerto dalla piattaforma.

Pre-condizioni: nessuna.

Garanzia di successo: il cliente risulta registrato alla piattaforma.

Scenario principale di successo:

1) Il cliente accede alla sezione apposita per la registrazione.
2) Il cliente compila il form di registrazione in cui inserisce:
   * Indirizzo email
   * Nome
   * Cognome
   * Password
   * Recapito telefonico
3) Il sistema registra l'avvenuta compilazione del form.
4) Il sistema invia l'email di conferma.
5) Il cliente conferma la registrazione tramite link ricevuto via email.

Flussi alternativi:

2ª) Indirizzo email già presente nel sistema.

1- Il sistema notifica al cliente che l'email inserita è già presente all'interno del sistema.
2- Il sistema chiede se il cliente ha dimenticato la password.
3a- Il cliente torna alla sezione principale della piattaforma.
3b- Il cliente compila il form per il recupero della password.

Frequenza di ripetizione: potenzialmente ininterrotta.

2.3 – Analisi

Per questa iterazione sono stati scelti i requisiti:

- Scenario principale di successo di UC2
- Scenario principale di successo di UC6 limitato al caricamento di soli file pdf
- Scenario principale di successo di UC7
- I dati sono già presenti su database.

Caso d’uso UC2 (Compilazione tariffario copisteria): Modello di Dominio

In questa fase è di interesse lo scenario principale di successo, riscontriamo le seguenti classi concettuali:

- *ProprietarioCopisteria*: attore primario, interagisce direttamente con la piattaforma;
- *Copiga*: la piattaforma in esame;
- *SpecificheDiStampa*: l’insieme delle caratteristiche di stampa accettate;
- *TipoFile*: è un tipo di file che è possibile scegliere di accettare di stampare, es. file testuale;
- *Carta*: indica il formato, la scala colori e il prezzo per foglio;
- *Rilegatura*: indica la tipologia di rilegatura e il prezzo, es. ad anelli o a caldo

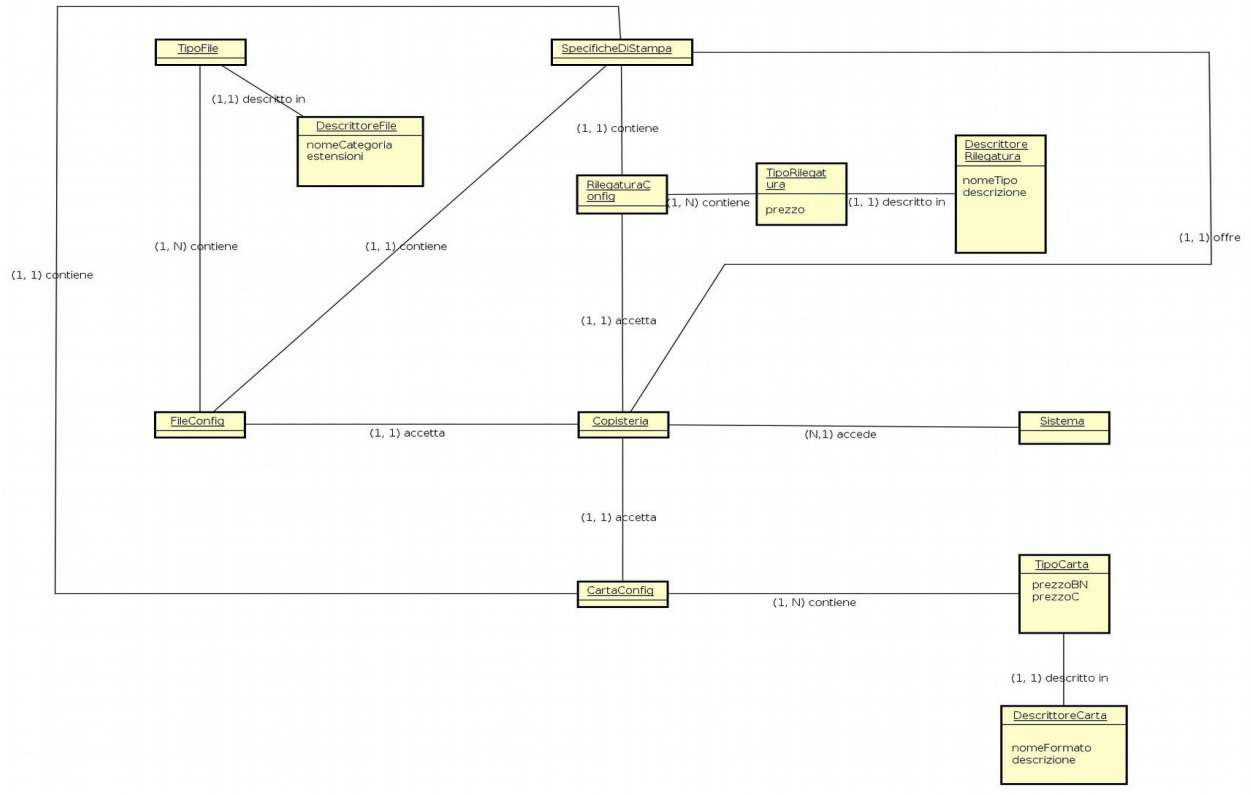


Figura 1: Modello di Dominio di UC2 – Compilazione tariffario copisteria

Caso d’uso UC6 (Invio file di stampa alla copisteria): Modello di Dominio

In questa fase è d’interesse lo scenario principale di successo limitatamente al solo invio di file pdf, riscontriamo le seguenti classi concettuali (di seguito un elenco da integrare con le classi concettuali trovate nel precedente modello di dominio):

- *Cliente*: attore primario, interagisce direttamente con il sistema;
- *Credito*: rappresenta il credito dell’utente, questo possiede un credito che è specifico per ogni copisteria, non può quindi essere un attributo poiché non è un tipo di dato primitivo;
- *StampaFile*: rappresenta la specifica ordinazione effettuata dal cliente;

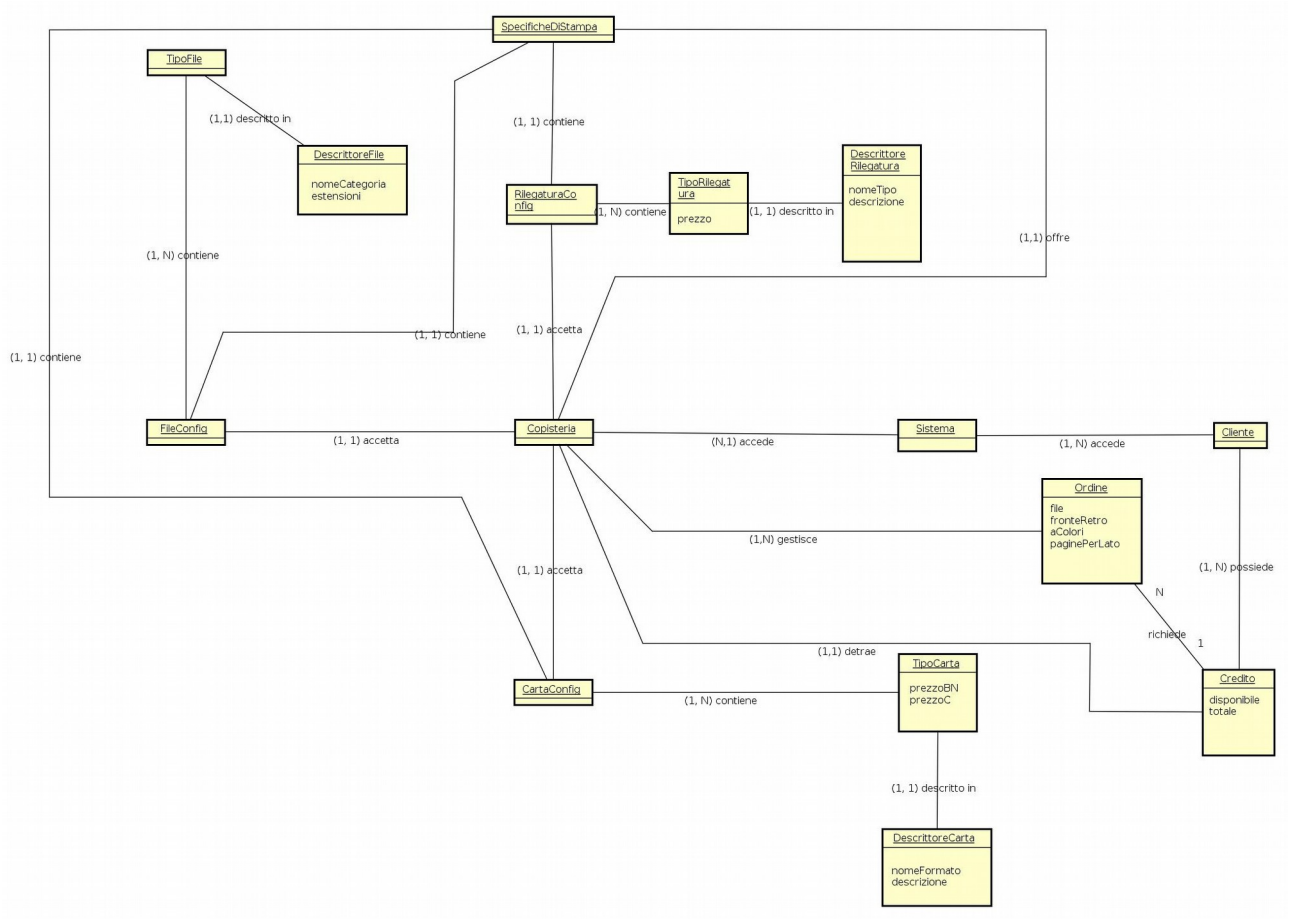


Figura 2: Modello di Dominio di UC6 – Invio file di stampa alla copisteria

Caso d’uso UC7 (Accettazione task e invio richiesta di ritiro): Modello di Dominio

In questa fase è d’interesse lo scenario principale di successo, riscontriamo le seguenti classi concettuali (di seguito un elenco da integrare con le classi concettuali trovate nei precedenti modelli di dominio):

- *Notifica*: si occupa di notificare l’accettazione del job di stampa al cliente e di notificare la copisteria della ricezione di un nuovo job di stampa.

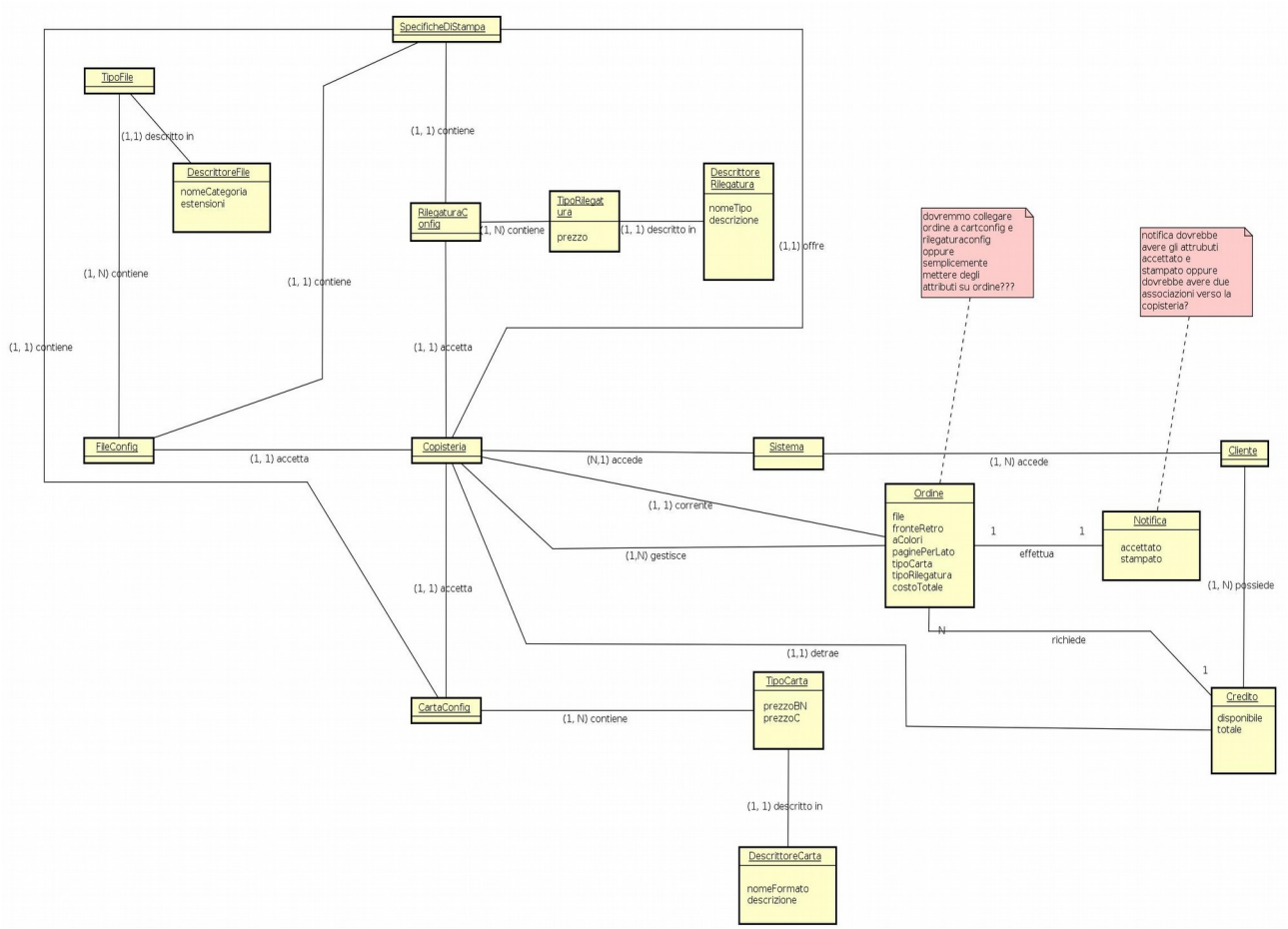
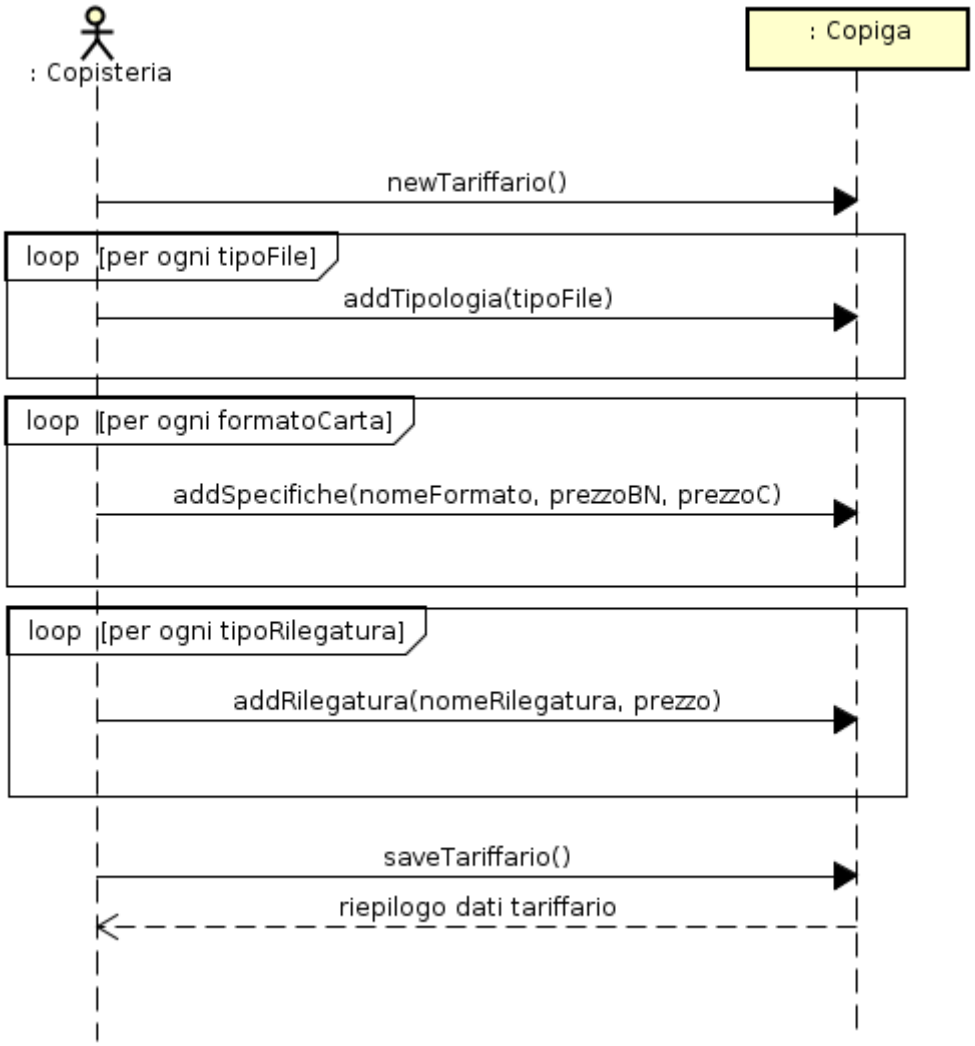


Figura 3: Modello di Dominio di UC7 – Accettazione task e invio richiesta di ritiro

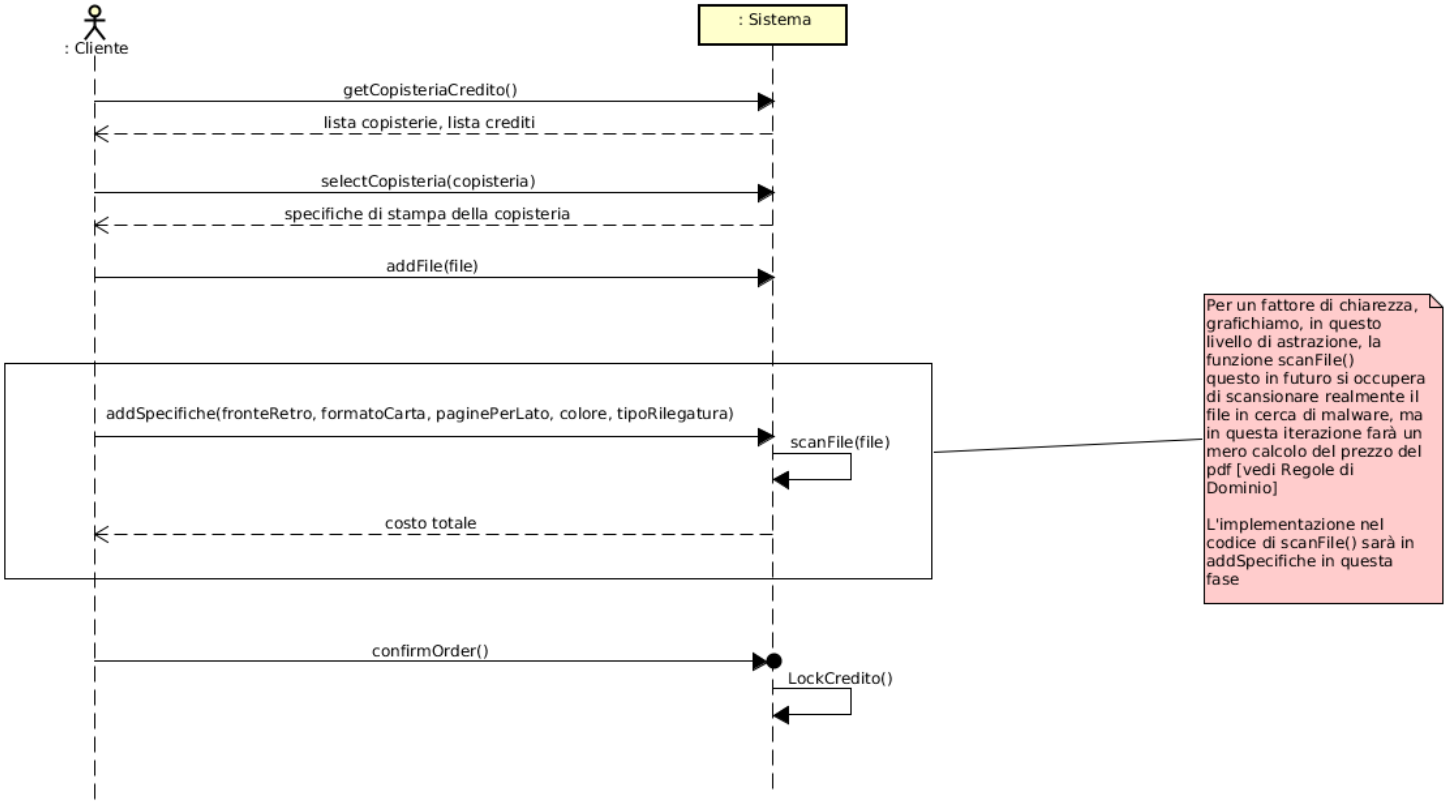
Caso d’uso UC2 (Compilazione tariffario copisteria): Diagramma di Sequenza di Sistema

Il diagramma di sequenza di sistema per UC2 è il seguente:



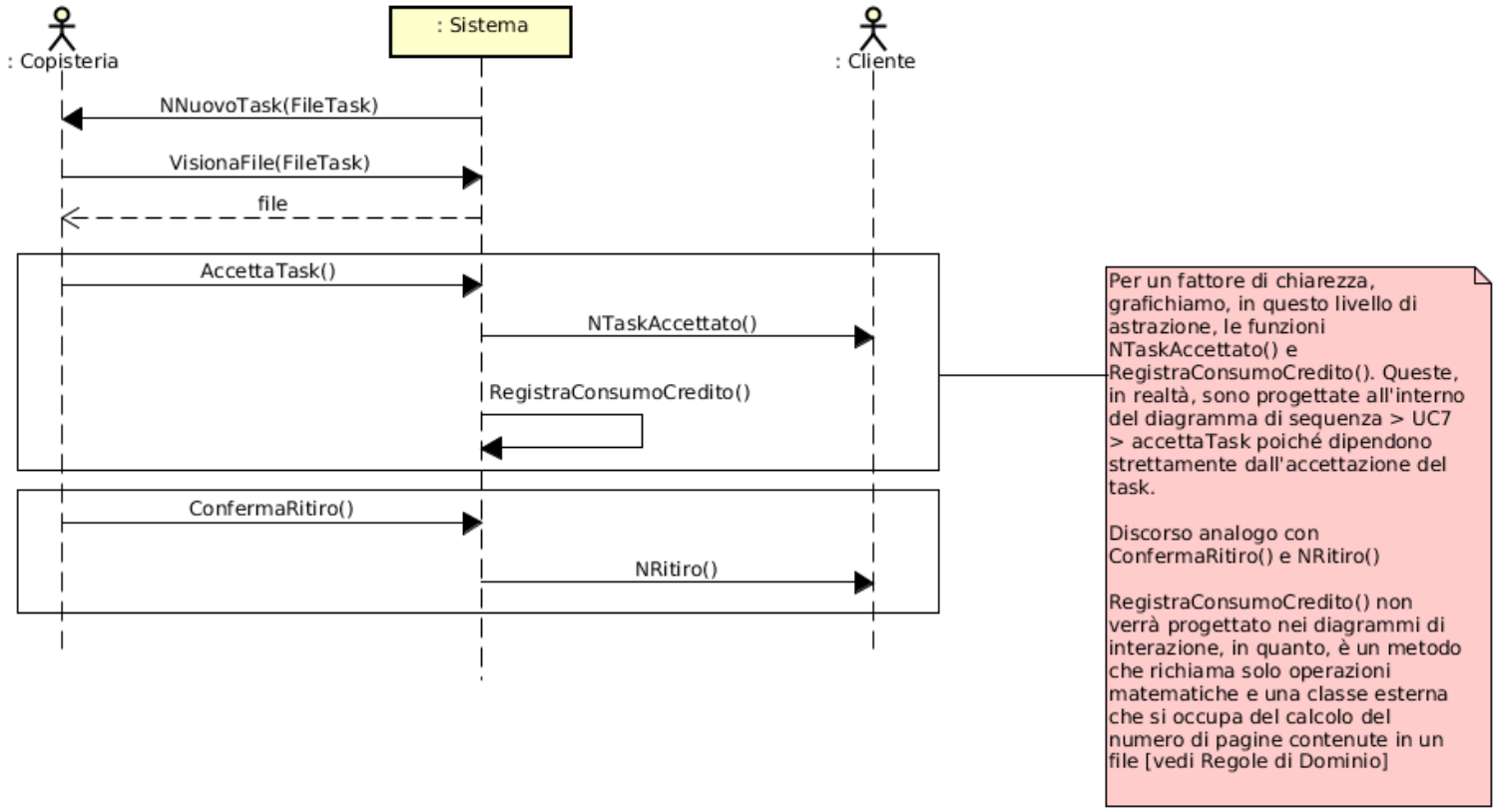
Caso d’uso UC6 (Invio file di stampa alla copisteria): Diagramma di Sequenza di Sistema

Il diagramma di sequenza di sistema per UC6 è il seguente:



Caso d’uso UC7 (Accettazione task e invio richiesta di ritiro): Diagramma di Sequenza di Sistema

Il diagramma di sequenza di sistema per UC7 è il seguente:



Caso d’uso UC2 (Compilazione tariffario e specifiche di stampa della copisteria): Diagramma di Sequenza di Sistema

Il diagramma di sequenza di sistema per UC2 è il seguente:

Operazione: newTariffario()
Riferimenti: Caso d’uso UC2
Pre-condizioni: -
Post-condizioni: - Sono state istanziate CartaConfig c*c*, RilegaturaConfig rc, FileConfig *fc* e collegate a copisteria tramite “accetta”

Operazione: addTipologia(tipoFile:String)
Riferimenti: Caso d’uso UC2
Pre-condizioni: - Sono presenti le istanze *cc*, *rc* e *fc*
Post-condizioni: - È stata istanziata TipoFile *tf* e connessa a *fc* tramite “contiene”
- *tf* è collegata a DescrittoreFile tramite “descritto in”

Operazione: addSpecifiche(nomeFormato:String, prezzoBN:float, prezzoC: float)
Riferimenti: Caso d’uso UC2
Pre-condizioni: - Sono presenti le istanze *cc*, *rc* e *fc*
Post-condizioni: - È stata istanziata TipoCarta *tc* e connessa a *cc* tramite “contiene”
- *tc* è collegata a DescrittoreCarta tramite “descritto in”
- Tutti gli attributi di *tc* sono stati inizializzati

Operazione: addRilegatura(nomeRilegatura:String, prezzo:float)
Riferimenti: Caso d’uso UC2
Pre-condizioni: - Sono presenti le istanze *cc*, *rc* e *fc*
Post-condizioni: - È stata istanziata TipoRilegatura *tr* e connessa a r*c* tramite “contiene”
- *tr* è collegata a DescrittoreRilegatura tramite “descritto in”
- Tutti gli attributi di *tr* sono stati inizializzati

Operazione: saveTariffario()
Riferimenti: Caso d’uso UC2
Pre-condizioni: - Sono presenti le istanze *cc*, *rc* e *fc*
Post-condizioni: - Le istanze *cc*, *rc* e *fc* sono state associate a SpecificheDiStampa tramite “contiene”

Caso d’uso UC6 (Invio file di stampa alla copisteria): Contratti delle operazioni

Operazione: getCopisteriaCredito()
Riferimenti: Caso d’uso UC6
Pre-condizioni: -
Post-condizioni: - Cliente *cli* è stato associato ai crediti di tipo Credito attraverso “possiede”

Operazione: selectCopisteria(copisteria:String)
Riferimenti: Caso d’uso UC6
Pre-condizioni: - L’utente ha credito presso almeno una copisteria
Post-condizioni: - È stata creata una nuova istanza *o* di Ordine
- *o* è stato associato a Credito *cre* attraverso “richiede”
- *o* è stato associato a Copisteria *c* attraverso “corrente”
- *c* è stato associato a CartaConfig attraverso “accetta”
- *c* è stato associato a RilegaturaConfig attraverso “accetta”

Operazione: addFile(file:String)
Riferimenti: Caso d’uso UC6
Pre-condizioni: - È in corso l’inserimento di un Ordine *o*
Post-condizioni: - (È stato inviato un file candidabile per la stampa)

Operazione: scanFile(file:String)
Riferimenti: Caso d’uso UC6
Pre-condizioni: - È in corso l’inserimento di un Ordine *o*
Post-condizioni: - (Il file è stato ritenuto consono)
- *o.file* è stato inizializzato
- *c* è stato associato a FileConfig attraverso “accetta”

Operazione: addSpecifiche(fronteRetro:Bool, formatoCarta:String, paginePerLato:Int, colore:Bool, tipoRilegatura:String, tipoCarta:String, tipoRilegatura:String)
Riferimenti: Caso d’uso UC6
Pre-condizioni: - È in corso l’inserimento di un Ordine *o*
- *o.file* è inizializzato
Post-condizioni: - *o*.paginePerLato, *o*.fronteRetro, *o*.aColori, *o*.tipoCarta, *o*.tipoRilegatura sono stati inizializzati
- *o*.costoTotale è stato inizializzato

Operazione: confirmOrder()
Riferimenti: Caso d’uso UC6

Pre-condizioni: - *o* presenta tutti gli attributi inizializzati
Post-condizioni: - *o* è stato associato a *c* tramite “gestisce”

Operazione: lockCredito()
Riferimenti: Caso d’uso UC6
Pre-condizioni: - *o* è associato a *c* tramite “gestisce”
Post-condizioni: - *o* è stato associato a Notifica *n* tramite “effettua”
- *cre*.disponibile viene aggiornato

Caso d’uso UC7 (Accettazione task e invio richiesta di ritiro): Contratti delle operazioni

Operazione: NnuovoTask(fileTask:String)
Riferimenti: Caso d’uso UC7
Pre-condizioni: - Ordine *o* è associato a Notifica *n* tramite “effettua”
Post-condizioni: -

Operazione: VisionaFile(fileTask:String)
Riferimenti: Caso d’uso UC7
Pre-condizioni: - Esiste la relazione tra *c* e *o* tramite “gestisce”
Post-condizioni: -

Operazione: AccettaTask()
Riferimenti: Caso d’uso UC7
Pre-condizioni: - Esiste la relazione tra *c* e *os*
Post-condizioni: - *n*.accettato è stato inizializzato

Operazione: NTaskAccettato()
Riferimenti: Caso d’uso UC7
Pre-condizioni: - *n*.accettato è inizializzato
Post-condizioni: -

Operazione: RegistraConsumoCredito()
Riferimenti: Caso d’uso UC7
Pre-condizioni: - *n*.accettato è inizializzato
Post-condizioni: - Copisteria *c* e Credito *cre* sono stati associati tramite “detrae”
- *cre*.totale viene aggiornato

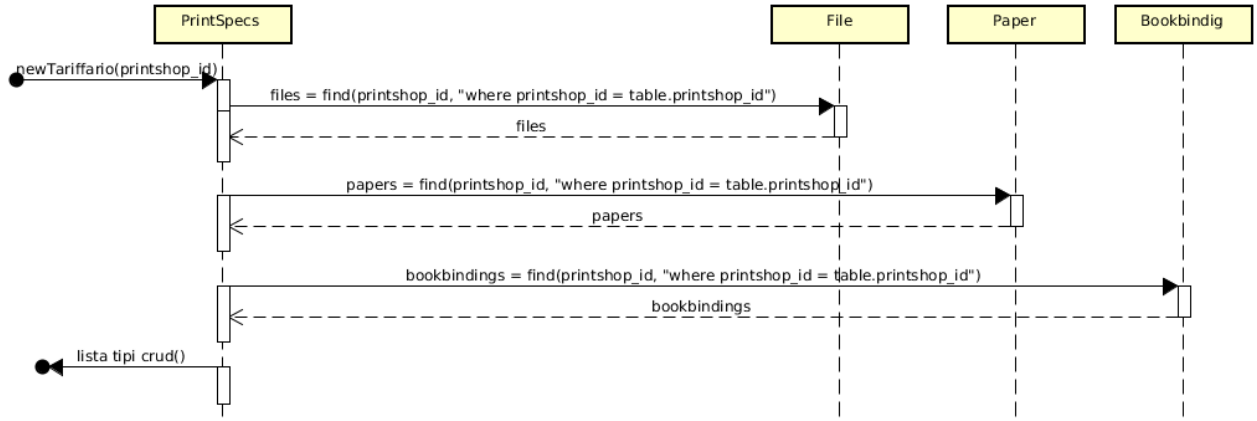
Operazione: ConfermaRitiro()
Riferimenti: Caso d’uso UC7
Pre-condizioni: - *n*.accettato è inizializzato
Post-condizioni: - *n*.stampato è stato inizializzato

Operazione: NRitiro()
Riferimenti: Caso d’uso UC7
Pre-condizioni: - *n*.stampato è inizializzato
Post-condizioni: -

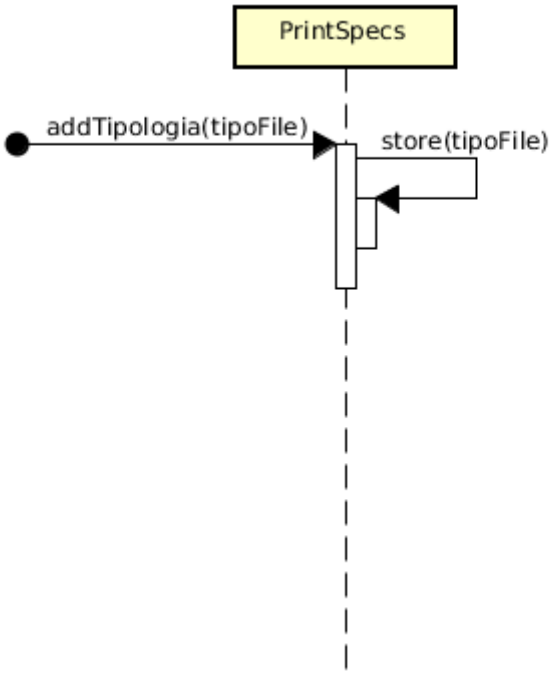
2.4 – Progettazione

2.4.1 - Caso d’uso UC2 (Compilazione tariffario copisteria): Diagrammi di interazione

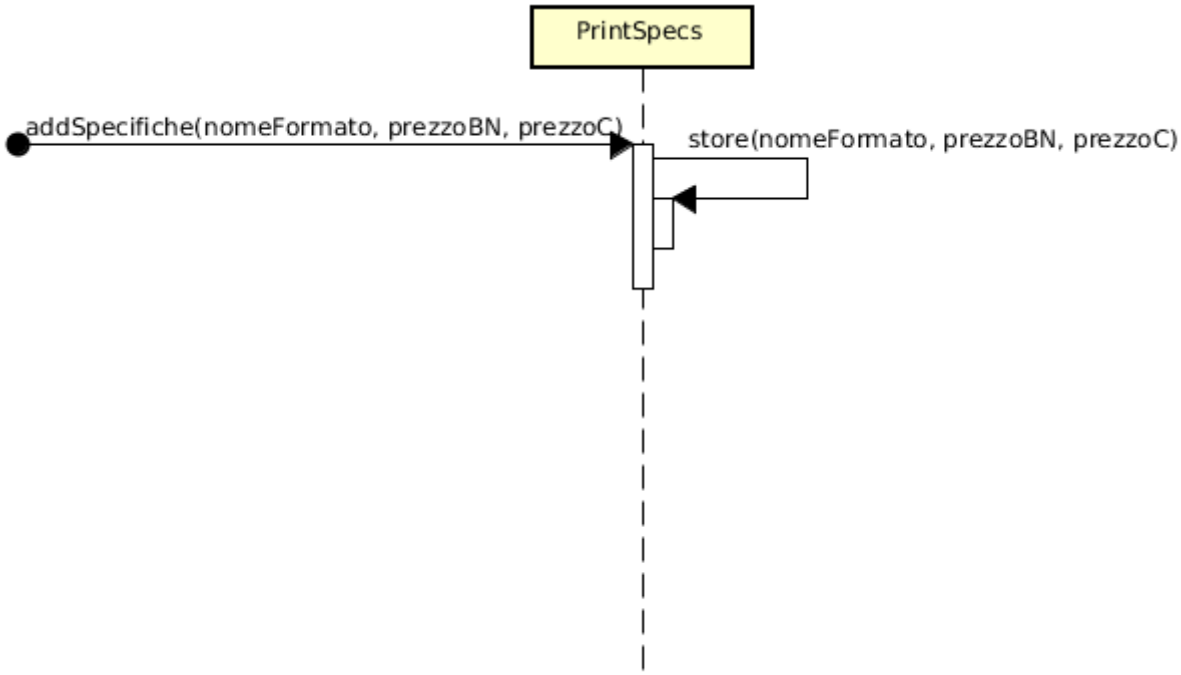
2.4.1.1 – newTariffario(printshop_id : longInt)



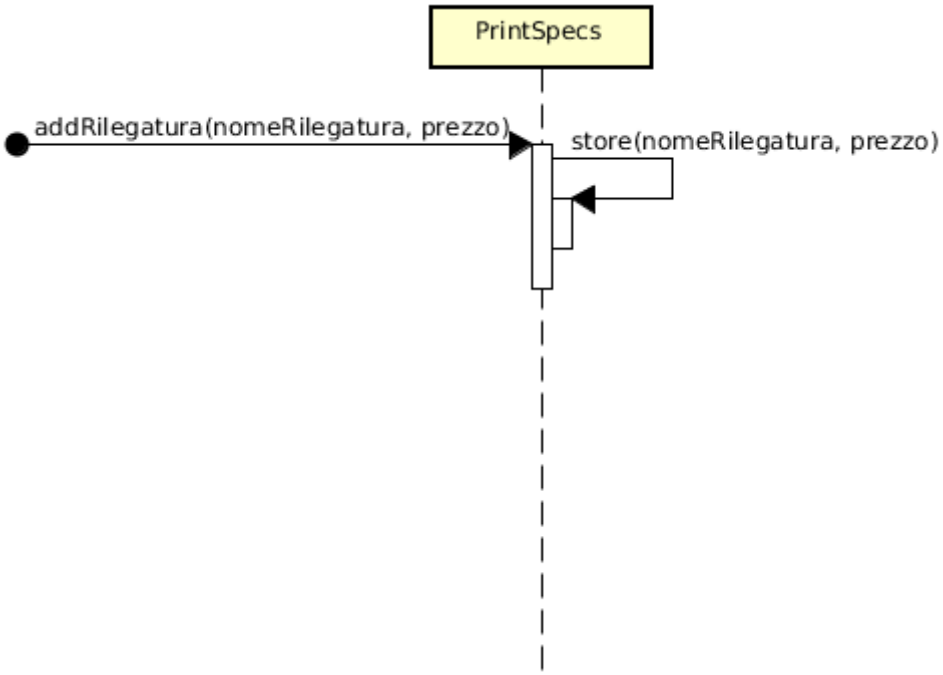
2.4.1.2 – addTipologia(tipoFile : String)



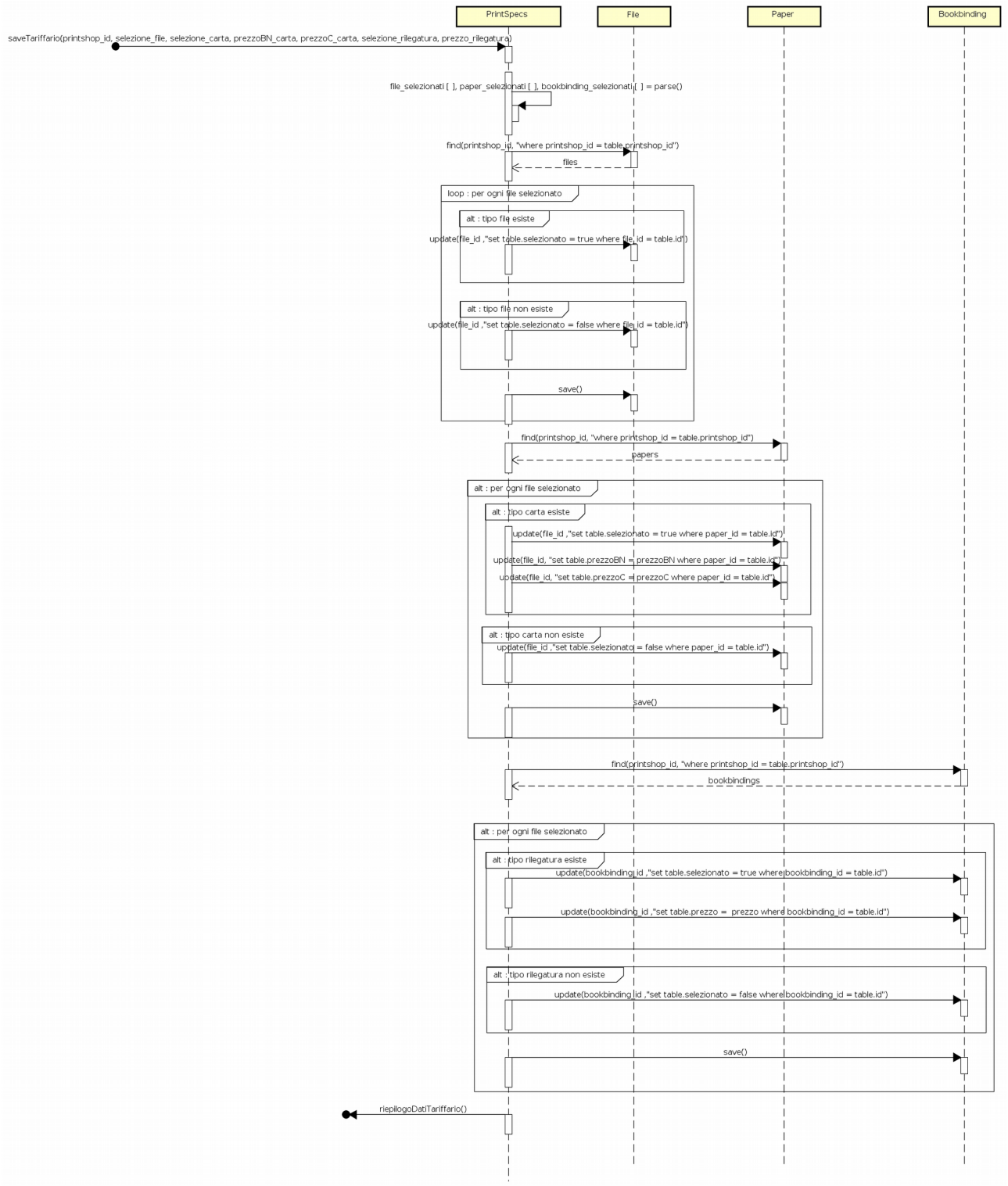
2.4.1.3 – addSpecifiche(nomeFormato : String, prezzoBN : float, prezzoC : float)



2.4.1.4 – addRilegatura(nomeRilegatura : String, prezzo : float)

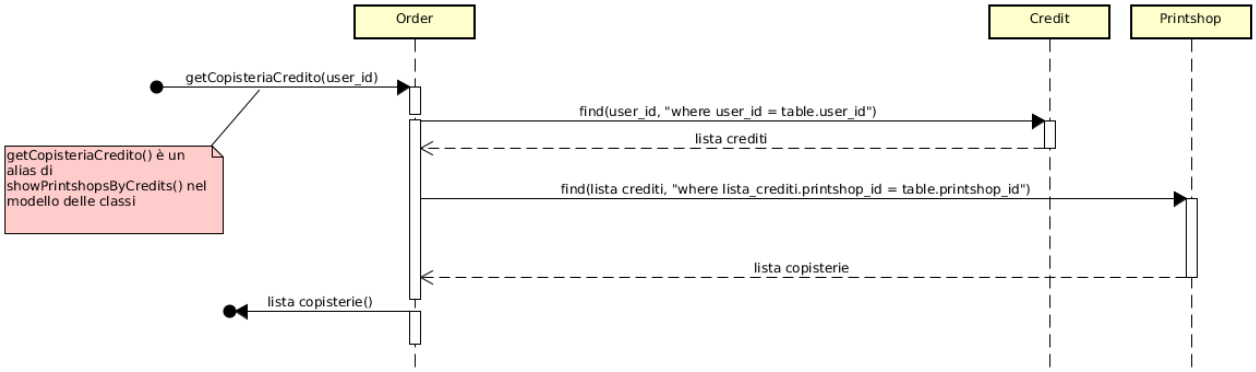


2.4.1.5 - saveTariffario()

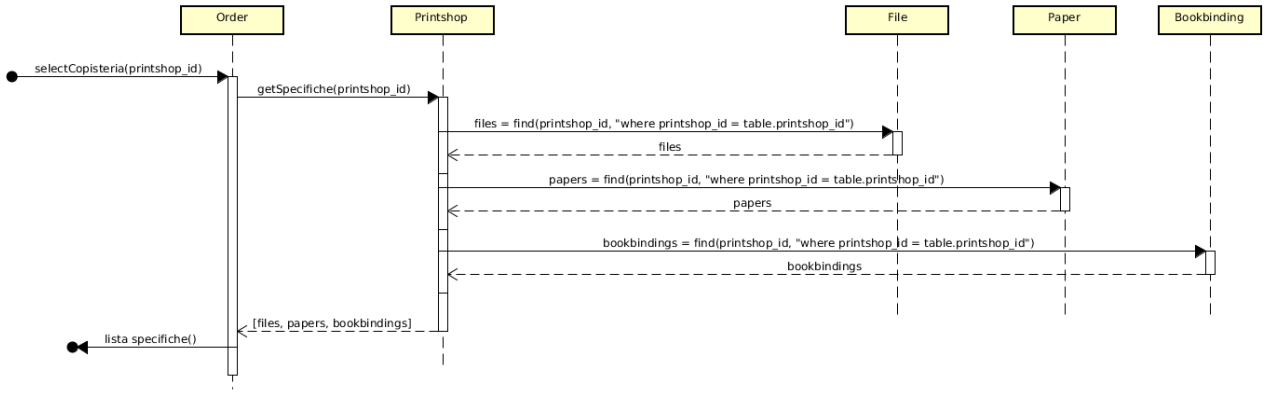

PrintSpecs
File
Paper
Bookbinding
saveTariffario(printshop_id, selezione_file, selezione_carta, prezzoBN_carta, prezzoC_carta, selezione_rilegatura, prezzo_rilegatura)
file_selezionati [ ], paper_selezionati [ ], bookbinding_selezionati [ ] = parse()
find(printshop_id, "where printshop_id = table.printshop_id")
files
loop : per ogni file selezionato
alt : tipo file esiste
update(file_id ,"set table.selezionato = true where file_id = table.id")
alt : tipo file non esiste
update(file_id ,"set table.selezionato = false where file_id = table.id")
save()
find(printshop_id, "where printshop_id = table.printshop_id")
papers
alt : per ogni file selezionato
alt : tipo carta esiste
update(file_id ,"set table.selezionato = true where paper_id = table.id")
update(file_id, "set table.prezzoBN = prezzoBN where paper_id = table.id")
update(file_id, "set table.prezzoC = prezzoC where paper_id = table.id")
alt : tipo carta non esiste
update(file_id ,"set table.selezionato = false where paper_id = table.id")
save()
find(printshop_id, "where printshop_id = table.printshop_id")
bookbindings
alt : per ogni file selezionato
alt : tipo rilegatura esiste
update(bookbinding_id ,"set table.selezionato = true where bookbinding_id = table.id")
update(bookbinding_id ,"set table.prezzo = _prezzo where bookbinding_id = table.id")
alt : tipo rilegatura non esiste
update(bookbinding_id ,"set table.selezionato = false where bookbinding_id = table.id")
save()
riepilogoDatiTariffario()

2.4.2 - Caso d’uso UC6 (Invio file di stampa alla copisteria): Diagrammi di interazione

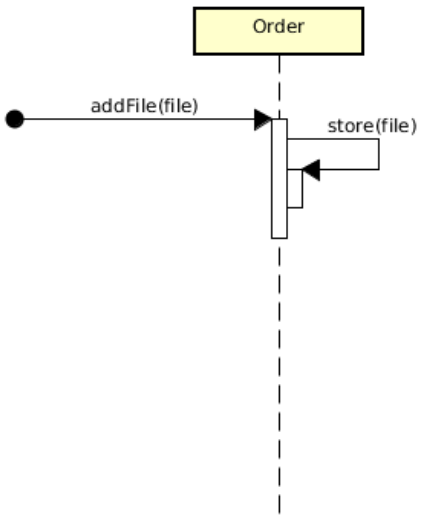
2.4.2.1 – getCopisteriaCredito()



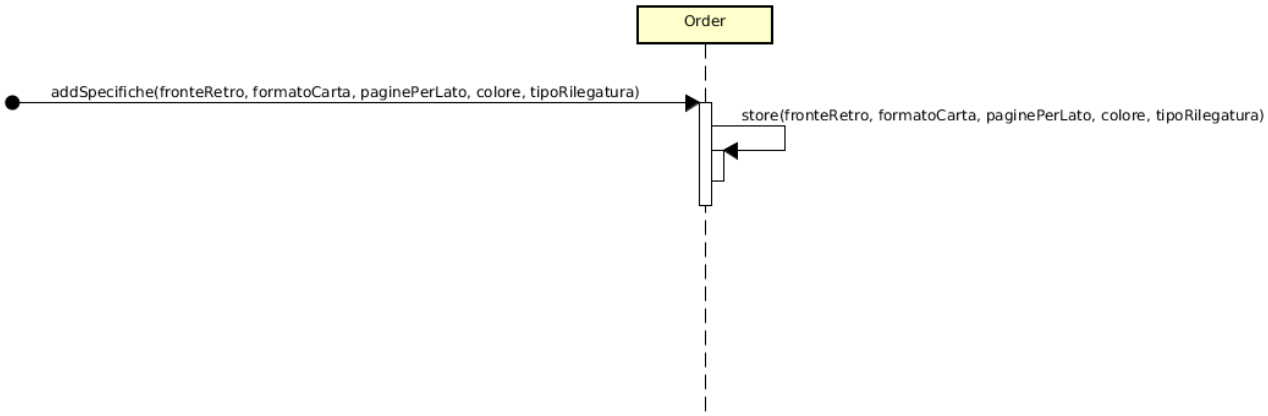
2.4.2.2 – selectCopisteria(printshop_id : longInt)



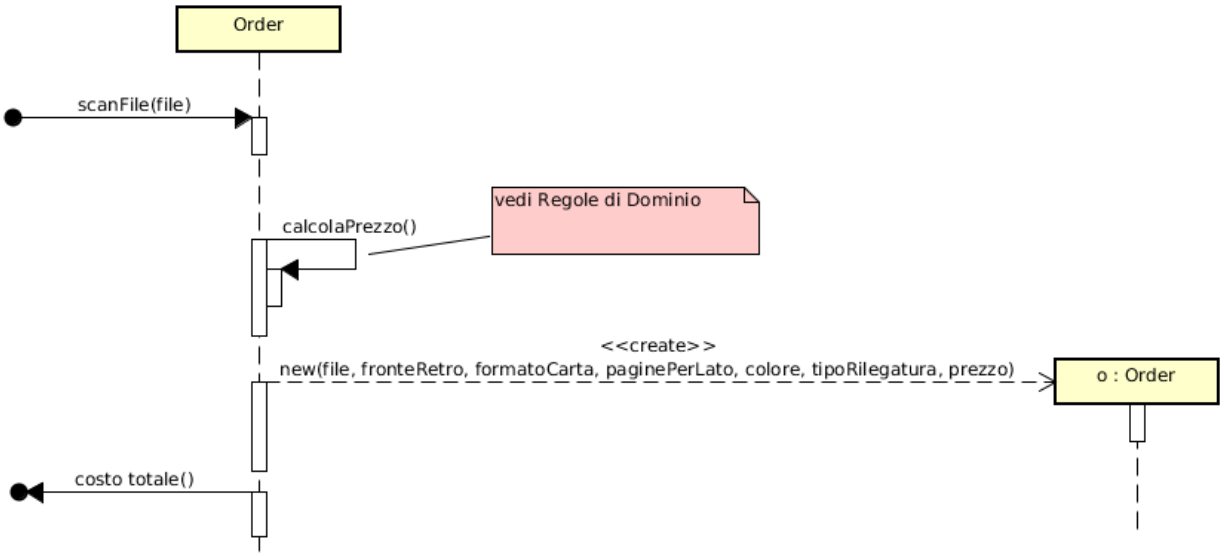
2.4.2.3 – addFile(file : String)



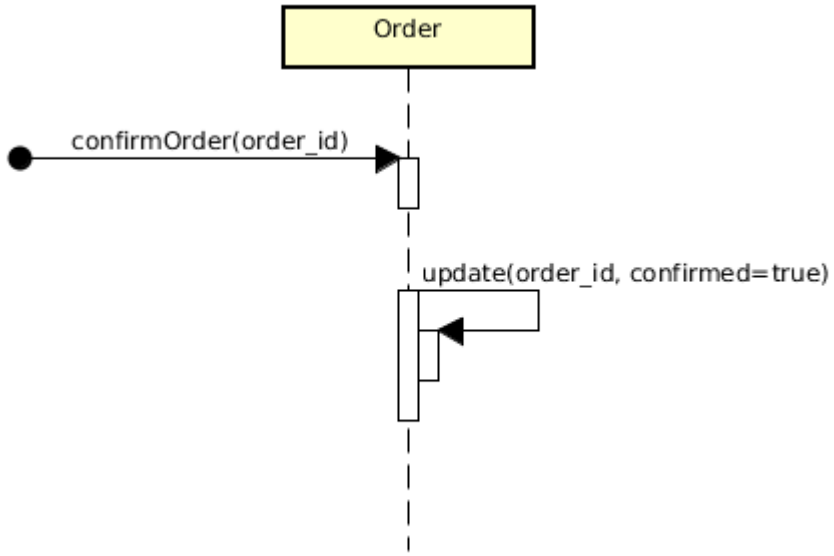
2.4.2.4 – addSpecifiche(fronteRetro : boolean, formatoCarta, paginePerLato : int, colore : boolean, tipoRilegatura : String)



2.4.2.5 – scanFile(file : String)

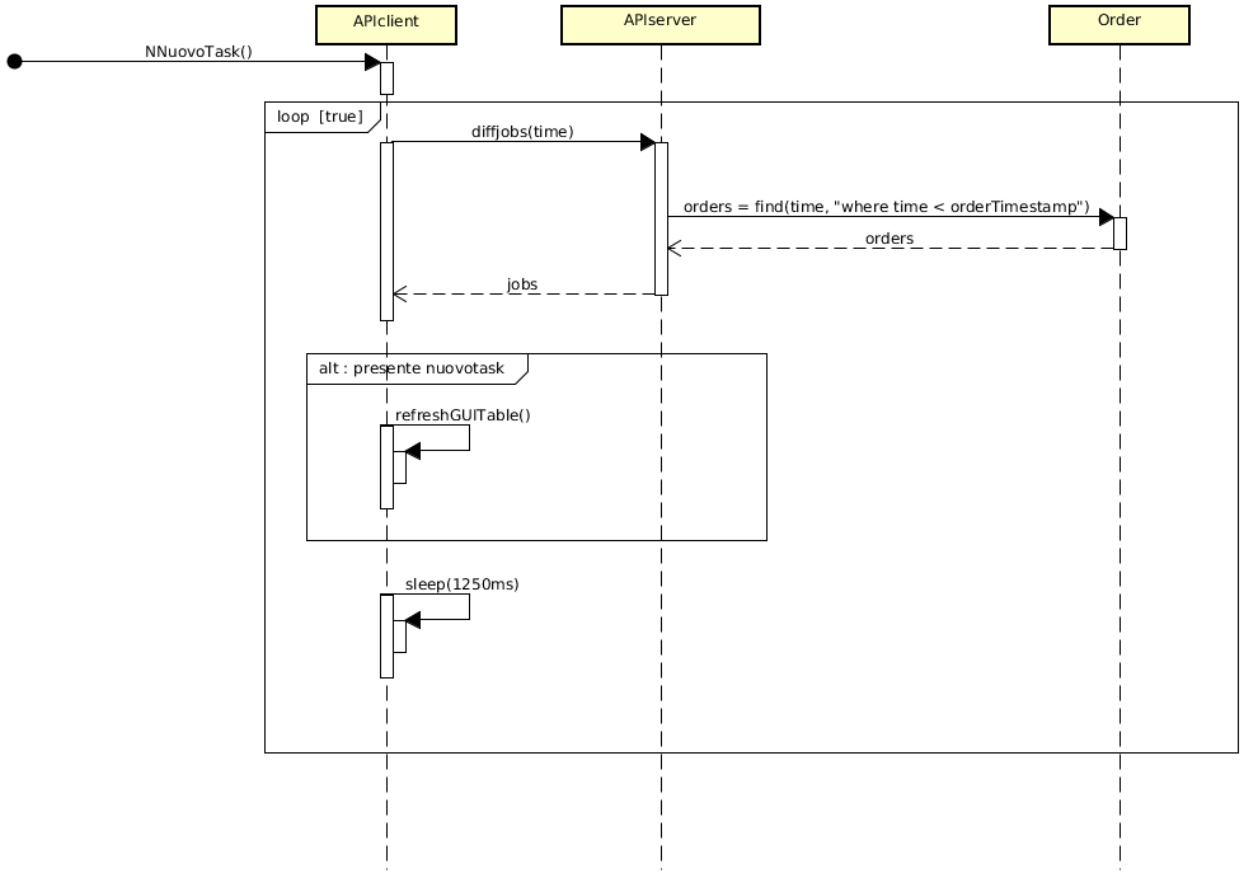


2.4.2.6 – confirmOrder(order_id : longInt)

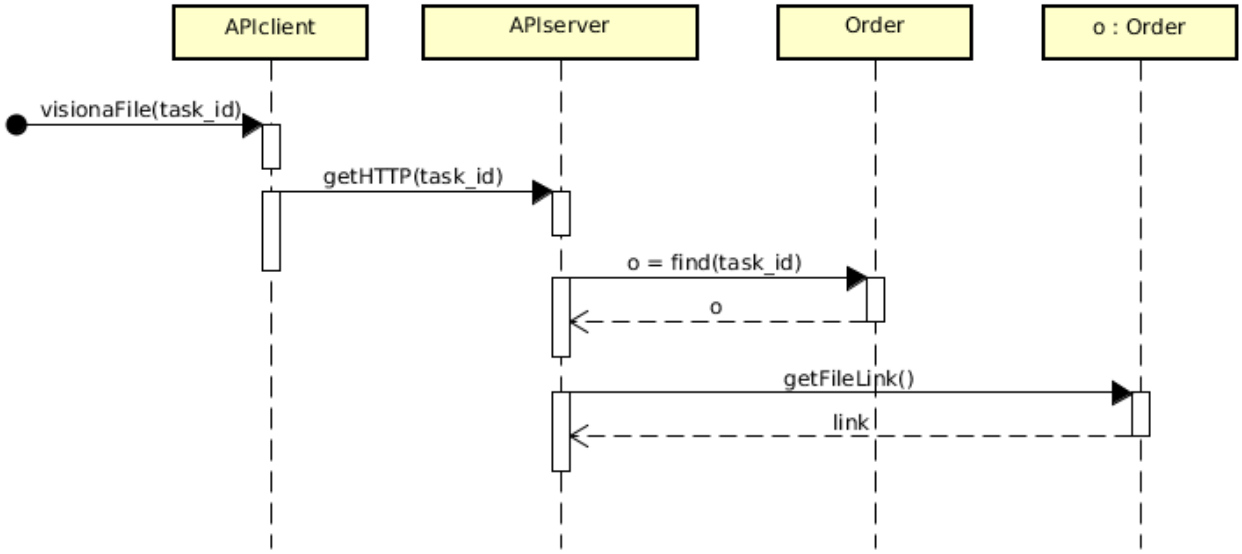


2.4.3 - Caso d’uso UC7 (Accettazione task e invio richiesta di ritiro): Diagrammi di interazione

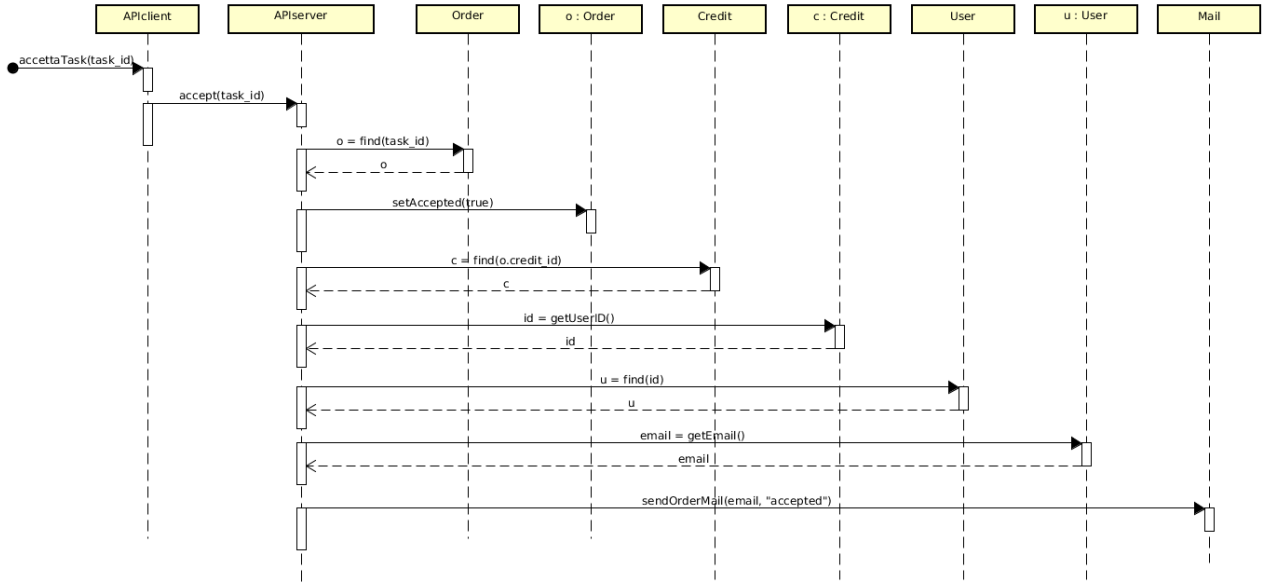
2.4.3.1 – NnuovoTask(task_id : longInt)



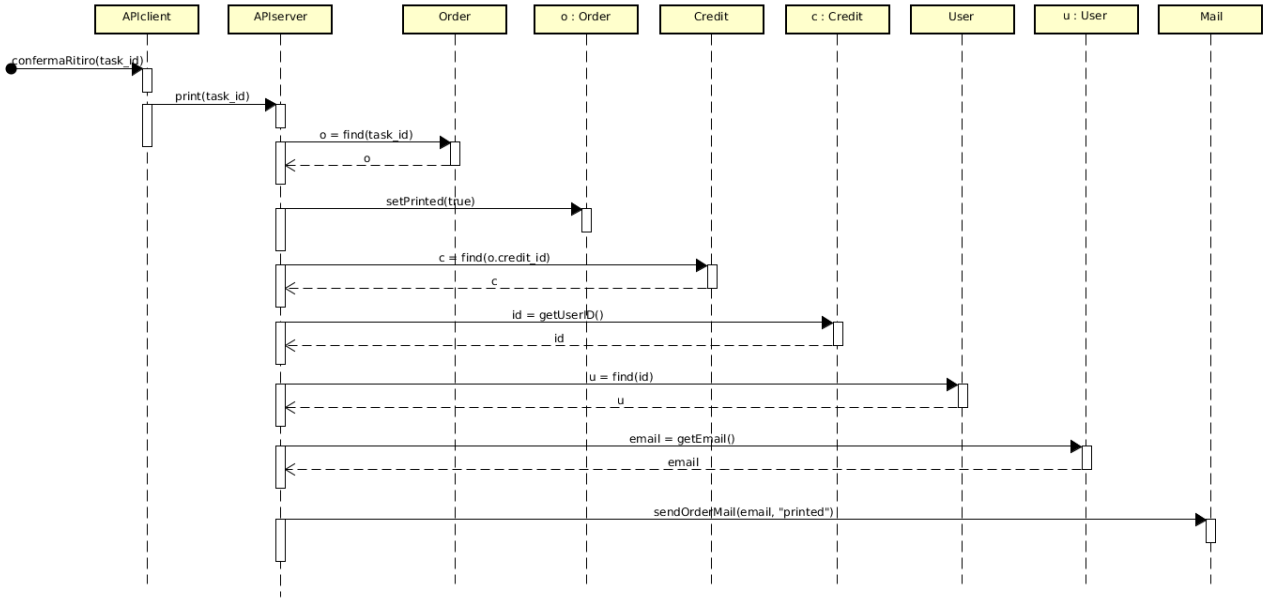
2.4.3.2 – visionaFile(task_id : longInt)



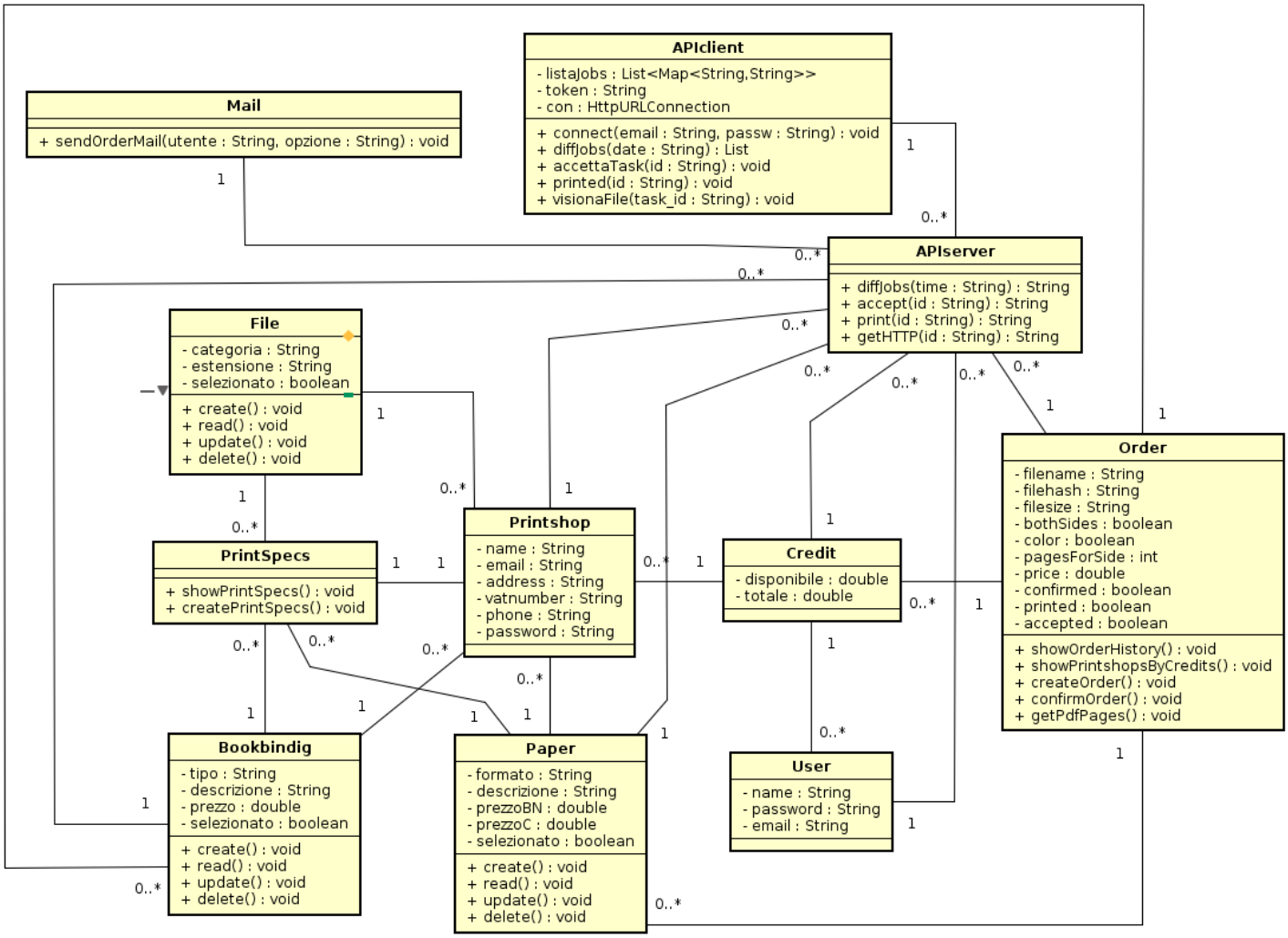
2.4.3.3 – accettaTask(task_id : longInt)



2.4.3.4 – confermaRitiro(task_id : longInt)



2.4.4 – Diagramma delle Classi



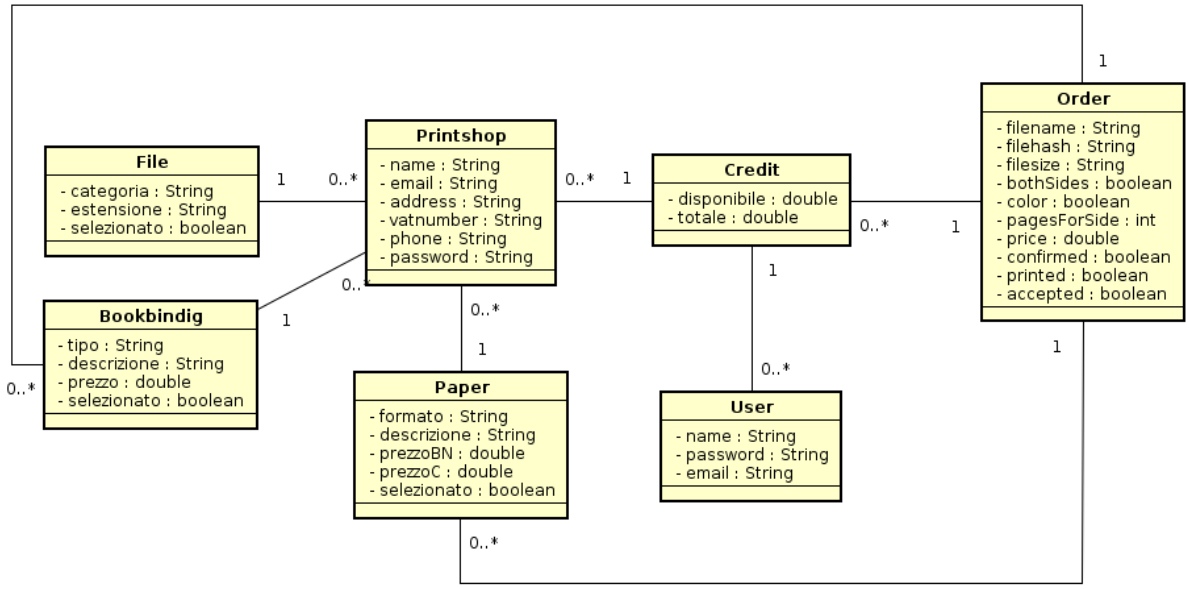
Nota: abbiamo ritenuto che questa notazione per la cardinalità fosse più chiara.

Nel caso, ad esempio, della relazione tra Bookbinding e PrintSpecs è da leggersi:

Bookbinding appartiene ad 1 *PrintSpecs*

PrintSpecs ha 0-N *Bookbindings*

2.4.3 – Diagramma di Database



3 – SECONDA ITERAZIONE

3.1 – Visione

3.1.1 – Introduzione

Il progetto in sviluppo offre un servizio di ottimizzazione per le copisterie: consente l'invio di ordini, da parte di utenti, alla copisteria più vicina senza code e con una gestione del credito sicura e trasparente.

3.1.2 – Posizionamento

3.1.2.1 – Opportunità di business

Attualmente non è presente in Italia un servizio unico multi-copisteria nel quale un utente può registrarsi e inviare in stampa alla copisteria più vicina dei file comodamente da casa.
Mediamente una copisteria locale su quattro ha un sito web, quando questo è presente nel 90% dei casi questo è un sito web statico, il restante 10% possiede un sistema di invio ordini di stampa alla specifica copisteria.

Durante la fase di ideazione si è ritenuta vantaggiosa la mancanza di un competitor diretto di Copiga.

Scopo di questa attuale versione MVP di Copiga è quello di attrarre proprietari di copisterie locali all'utilizzo della piattaforma in fase di test con il fine di ricevere feedback sulle feature ritenute vitali per l'attività di una copisteria sulla nostra piattaforma.
In una fase successiva si valuterà se vi sono i presupposti per un lancio di Copiga nel resto della penisola.

3.1.2.2 – Formulazione del problema

- **Descrizione del problema**: L'attuale fonte di reddito per una copisteria consiste nel fotocopiare documenti cartacei e nello stampare file. Quest'ultima attività è oggi lenta e ottimizzabile: è necessario andare fisicamente in copisteria, fare la fila, consegnare una memoria di massa contenente il file da scaricare all'addetto della copisteria ed aspettare che il materiale venga stampato. Mediamente il tempo stimato per quanto appena descritto è di 20 minuti.
- **Attori coinvolti**: Addetto copisteria, cliente copisteria
- **Impatto**: In particolari mesi dell'anno le copisterie sono fortemente congestionate, recarsi in copisteria per stampare file in questi periodi può scoraggiare il cliente. L'addetto della copisteria è costretto ad organizzare molteplici file all'interno del locale (fila ritiro, fila invio file stampe, fila fotopie).
- **Benefici di una soluzione di successo**: informatizzare l'invio di file alla copisteria riduce il numero di code in copisteria a tre; l'utente è avvisato quando è pronto il materiale per il ritiro: vengono eliminati i tempi di attesa in copisteria per i clienti che possiedono già il file da stampare.

3.1.2.3 – Formulazione della posizione del prodotto

- **Destinatari**: Copiga è rivolto ai proprietari di copisterie che vogliono ridurre il numero di clienti in attesa all'interno dei propri negozi e che vogliono offrire un servizio aggiuntivo capace di attrarre nuovi clienti.
- **Tipologia**: Piattaforma multi-copisteria: tecnologia web lato utente, web/desktop lato copisteria.
- **Funzione**: Copiga permette l'invio di ordini alle copisterie, il calcolo del prezzo è automatico, la gestione del credito è intuitiva e trasparente.
- **Soluzioni alternative attuali**: sono state trovate esclusivamente singole copisterie che offrono questo servizio tramite piattaforme web commissionate negli anni scorsi, spesso questi siti web presentano user experience pessima.
- **Caratteristiche del prodotto**: Se prima una copisteria avrebbe dovuto commissionare un sito web custom (dal costo di 800 - 1300 €) per questo servizio, su Copiga qualsiasi copisteria può entrare a fare parte del circuito senza spese iniziali.

3.1.3 – Descrizioni delle parti interessate

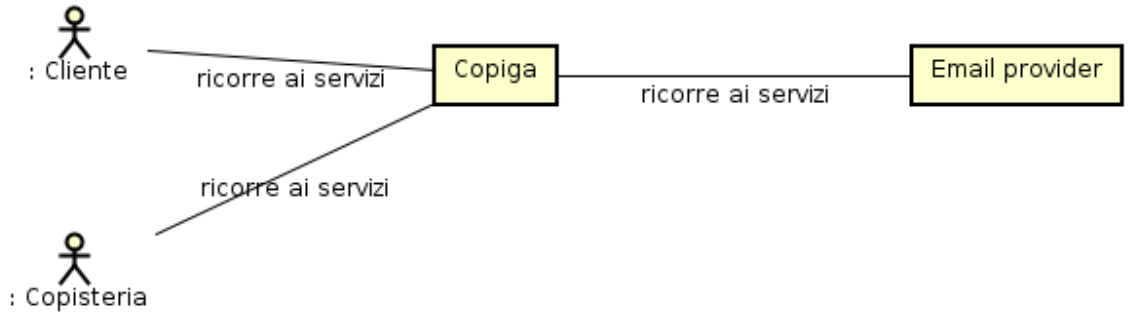
3.1.3.1 – Ambiente dell'utente

- La piattaforma sarà utilizzata da addetti copisteria e da clienti copisteria: due dashboard con comandi differenti
- L'utilizzatore del sistema deve essere in grado di effettuare un'operazione in tempi estremamente ridotti, entro il minuto.
- La comunicazione verso il cliente è gestita automaticamente tramite email.
- Il template del sito web deve essere responsive e offrire una user experience senza frizioni anche su dispositivi mobile

3.1.4 – Descrizione generale del prodotto

3.1.4.1 – Punto di vista del prodotto

È stato effettuato il deploy del server http in una vps a Toronto presso i datacenter Digital Ocean; qui risiede tutta la logica applicativa, il database e il server API REST.



Il client operativo lato copisteria sarà eseguito in locale su device della copisteria.

3.1.4.2 – Riepilogo dei vantaggi

- Riduzione della congestione in copisteria
- Tempi morti ridotti per stampare dei file in copisteria
- Procedura di pagamento veloce: eliminazione problema derivante da scarsità di denaro in moneta

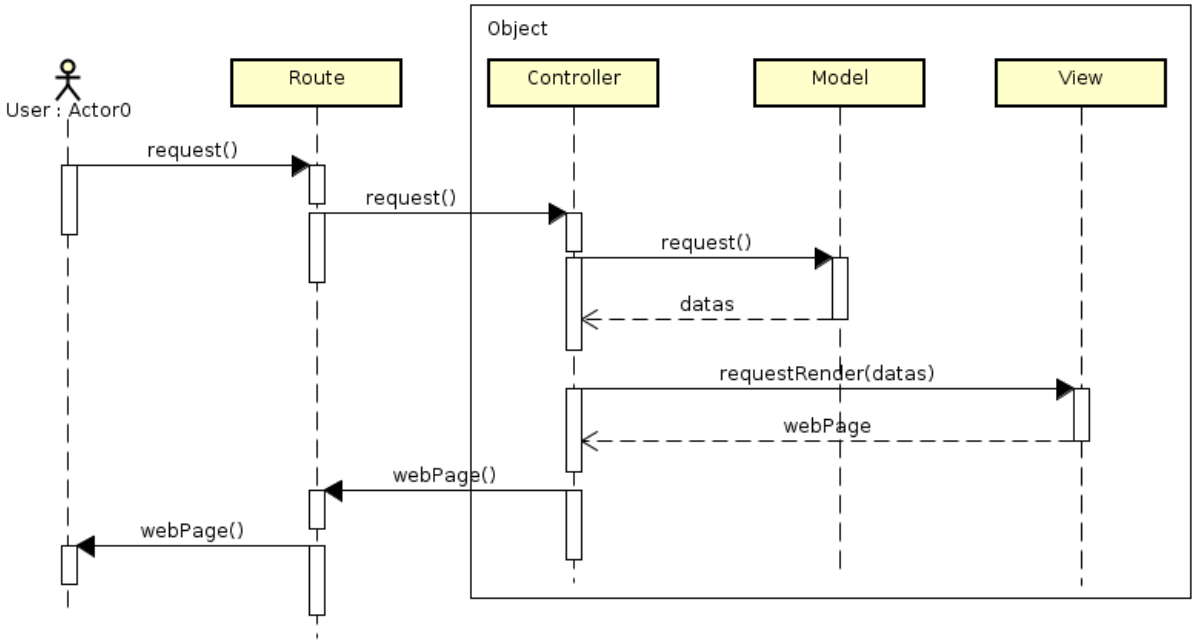
3.1.4.3 – Ipotesi e dipendenze

- Accesso alla rete internet
- Disponibilità di una vps performante

3.2 – Documento sull’Architettura Software

Nell’iterazione passata si sono verificate le criticità emerse nella precedente versione di questo documento, le tecnologie in utilizzo consentono una gestione agevole di quanto si necessita per il progetto.

Parte del progetto è implementato utilizzando il framework web lato server Laravel, questo e l’ecosistema di software di supporto ci permette di utilizzare il pattern MVC e di ottenere i vantaggi che questo comporta. Di seguito una schematizzazione di come è possibile gestire una richiesta da parte di un utente.



Nei diagrammi di interazione si considera un unico oggetto l’unione di controller, model e view.

Dovendo implementare la possibilità di poter registrarsi (e quindi autenticarsi) sia come utente che come copisteria sulla piattaforma web, nell’iterazione precedente ci domandavamo se questo fosse facilmente fattibile su un framework che si sta imparando ad utilizzare proprio in queste fasi.
La risposta a questa domanda è si: essendo molto comune l’implementazione di una registrazione ed un login in una web app, *artisan* – la principale utility di supporto allo sviluppo – consente con un solo comando l’implementazione di registrazione e login della classe utente.
La nostra visione sul progetto impone però che anche il proprietario della copisteria possa registrarsi e identificarsi sulla piattaforma come, appunto, una copisteria che vuole utilizzare Copiga al fine di offrire i suoi servizi; questo non è possibile con utility a nostra disposizione e si necessita quindi l’implementazione manuale di questa funzionalità attraverso l’utilizzo degli strumenti appositi offerti dal framework. Nonostante sia necessaria l’implementazione manuale della registrazione lato copisteria, questa è standard: si decide di escludere i casi d’uso UC1 e UC3 dalla progettazione.

Ricapitolando, Copiga presenta:

- Lato Utente: interamente su web app.
- Lato Copisteria: funzionalità critiche come l'inserimento di un nuovo tipo di carta o la compilazione del tariffario sono su web app; funzionalità di visualizzazione ordini di stampa, accettazione del job e notifiche sullo stato dell'ordine all'utente risiedono sull'app desktop sviluppata in JAVA.

Il software in JAVA necessita una comunicazione con il resto del software, questa avviene tramite REST API autenticate, si utilizza il package jwt-auth. Questo prevede una fase di login tramite richiesta HTTP (nel nostro caso specifico una POST HTTP) nel quale body sono inserite in chiaro email e password della copisteria, il messaggio di risposta include un token (una stringa 316 char) di autenticazione, questo deve essere inviato nel campo body delle API request che prevedono autenticazione al fine di essere riconosciuto dal sistema.
In fase di login il client non è vulnerabile a sniffing attack: sebbene si inviino in chiaro le credenziali, queste sono cifrate automaticamente poiché è stato installato un certificato TLS firmato dall'autorita di certificazione Let's Encrypt.
È possibile trovare il documento di riferimento per lo sviluppo di API client al seguente link: https://github.com/AlessandroSpallina/Copiga/wiki/API-Document

La gestione del database è facilitata poiché Laravel include Eloquent ORM: ogni tabella del database è mappata ad un model corrispondente; nei diagrammi di interazione è possibile visualizzare graficamente query al database tramite model, per semplicità si preferisce graficare queste interazioni con il database attraverso pseudofunzioni come "find" o "update" che non sono le funzioni effettivamente utilizzate, questa scelta è stata fatta al fine di mostrare chiaramente che quelle sono query al db avendo la certezza che lo sviluppatore che si occupa di quelle funzioni sa perfettamente come accedere ai dati tramite i model in Laravel.

L'invio delle mail per notificare al cliente della copisteria lo stato dell'ordine in UC7 è stato implementato attraverso opportune utility offerte dal framework Laravel e si è testato l'effettivo funzionamento dell'invio mail con lo strumento mailtrap.io che consente di simulare invio a mail fittizie.
Nel file .env è possibile modificare le credenziali per l'account di mailtrap.io, a questo indirizzo si dispone di una dashboard capace di visualizzare tutto il traffico mail inviato dal server.

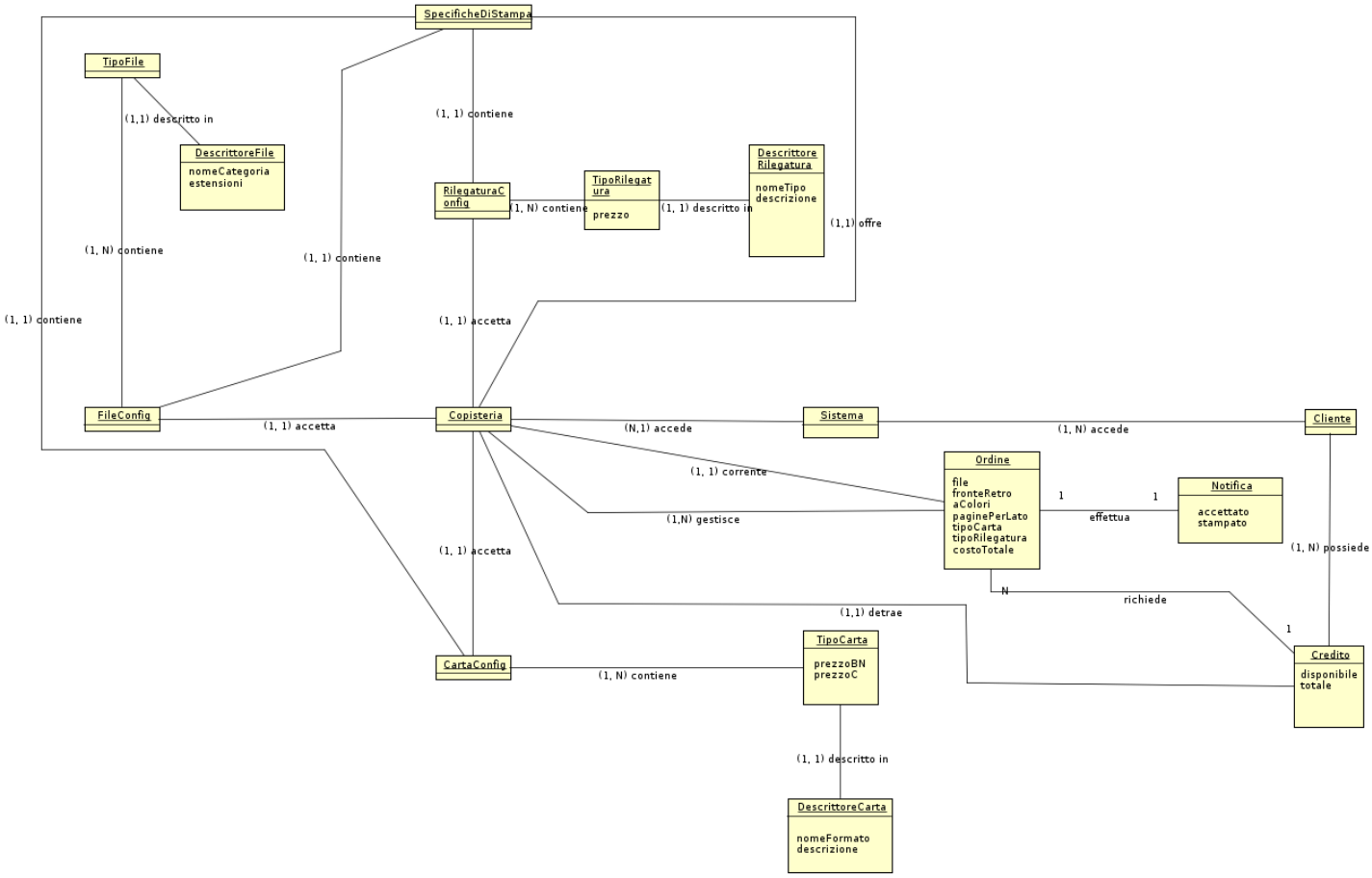
3.3 - Analisi

Per questa iterazione sono stati scelti i seguenti requisiti:

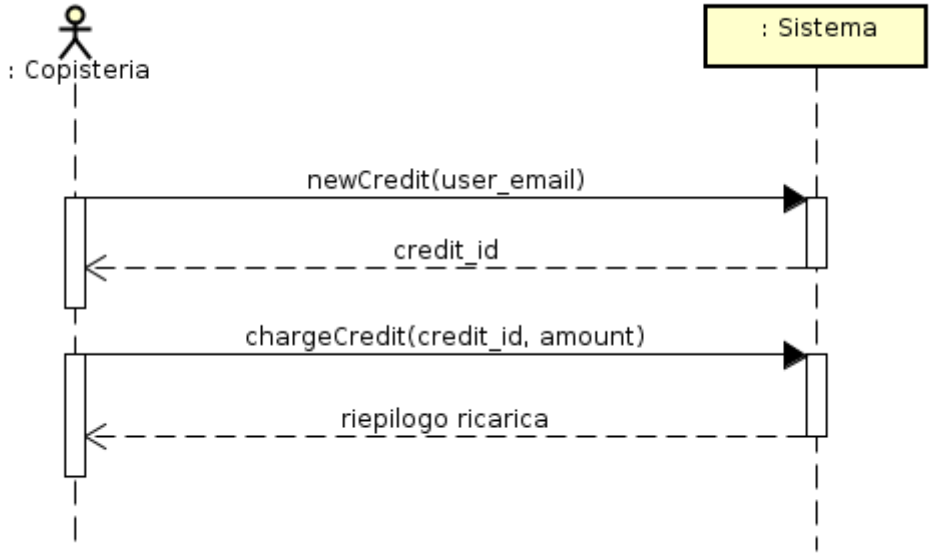
- Scenario principale di successo di UC4 e lo scenario alternativo 5a

Caso d’uso UC4 (Acquisto credito copisteria in contanti)

In questa fase è di interesse lo scenario principale di successo, non è necessario aggiornare il modello di dominio poiché comprende già la classe concettuale credito.



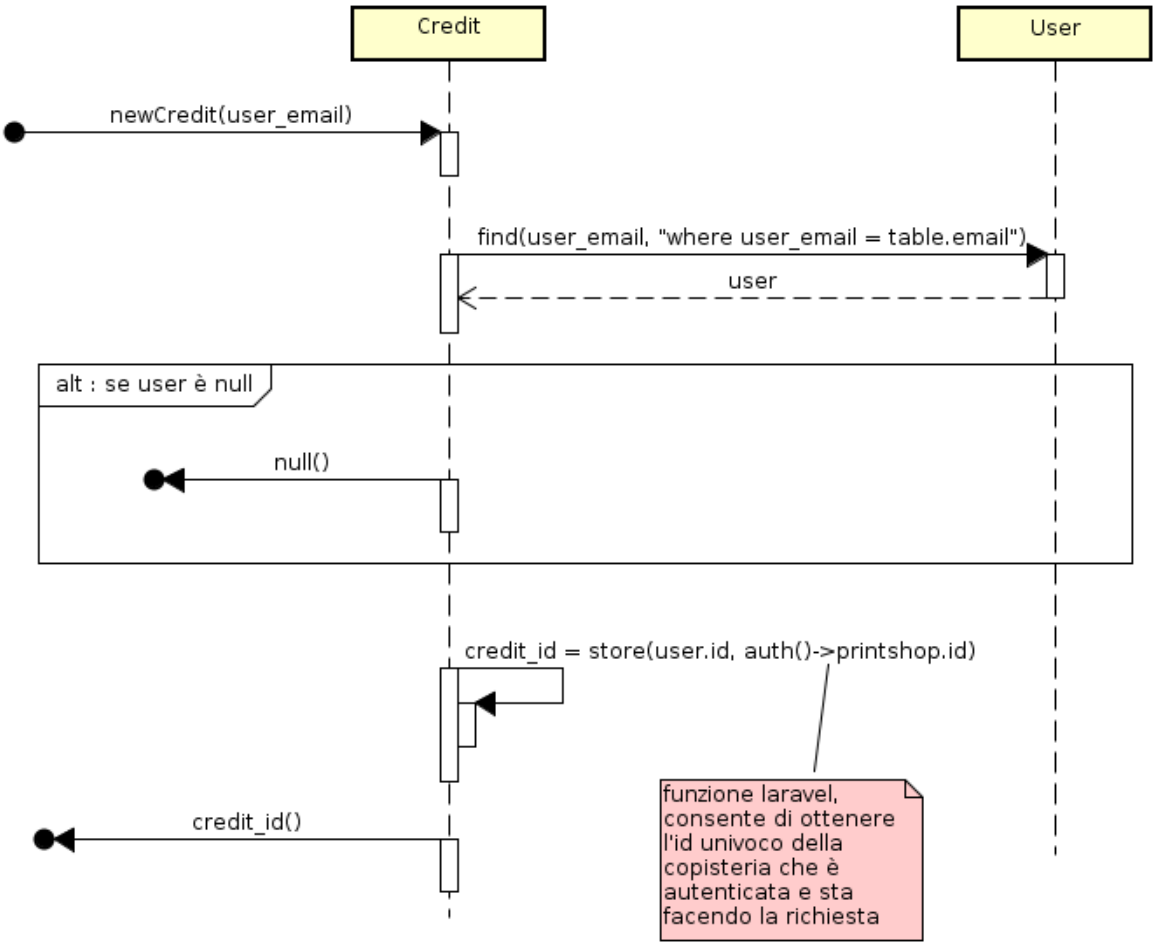
Il diagramma di sequenza di sistema per UC4 è il seguente:



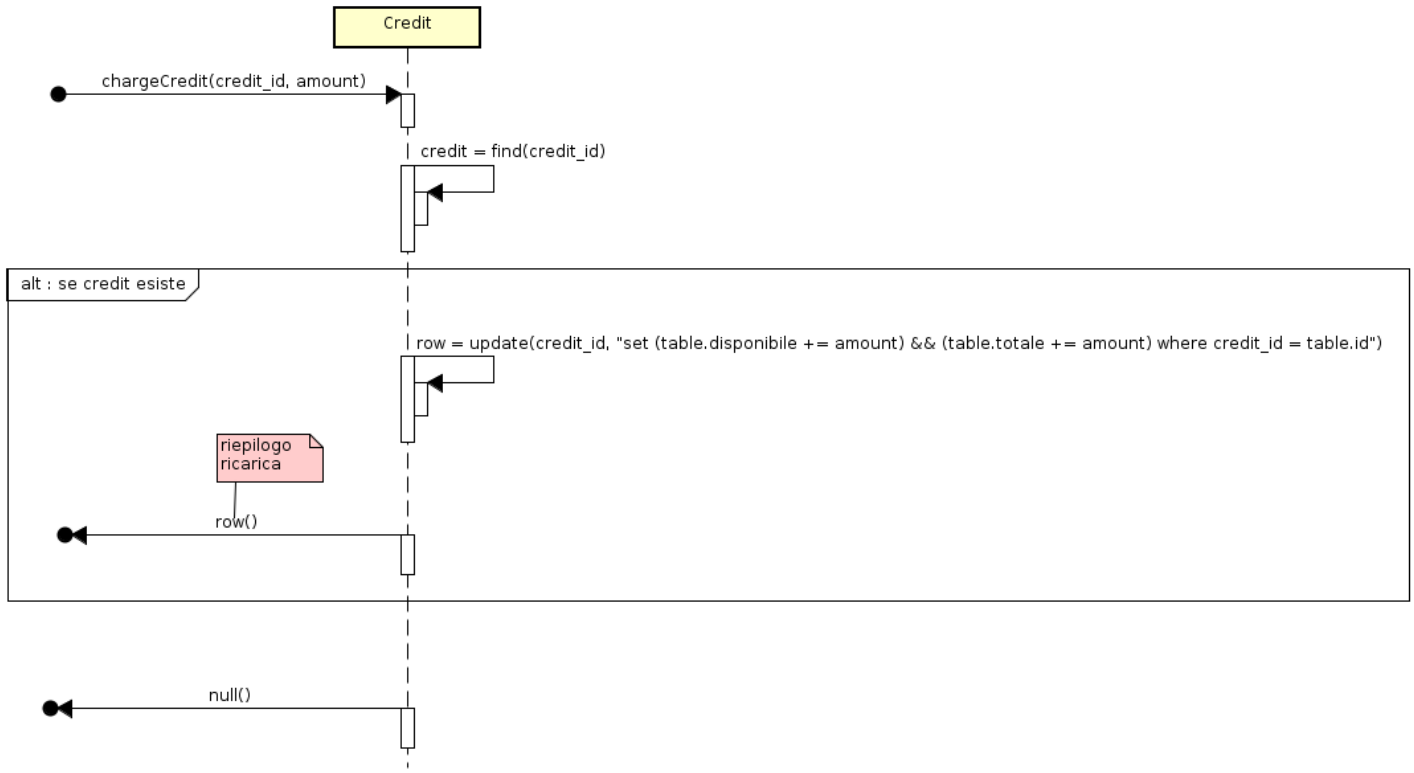
3.4 – Progettazione

Di seguito i diagrammi di interazione per le funzioni di UC4

3.4.1 – newCredit(user_email : String)

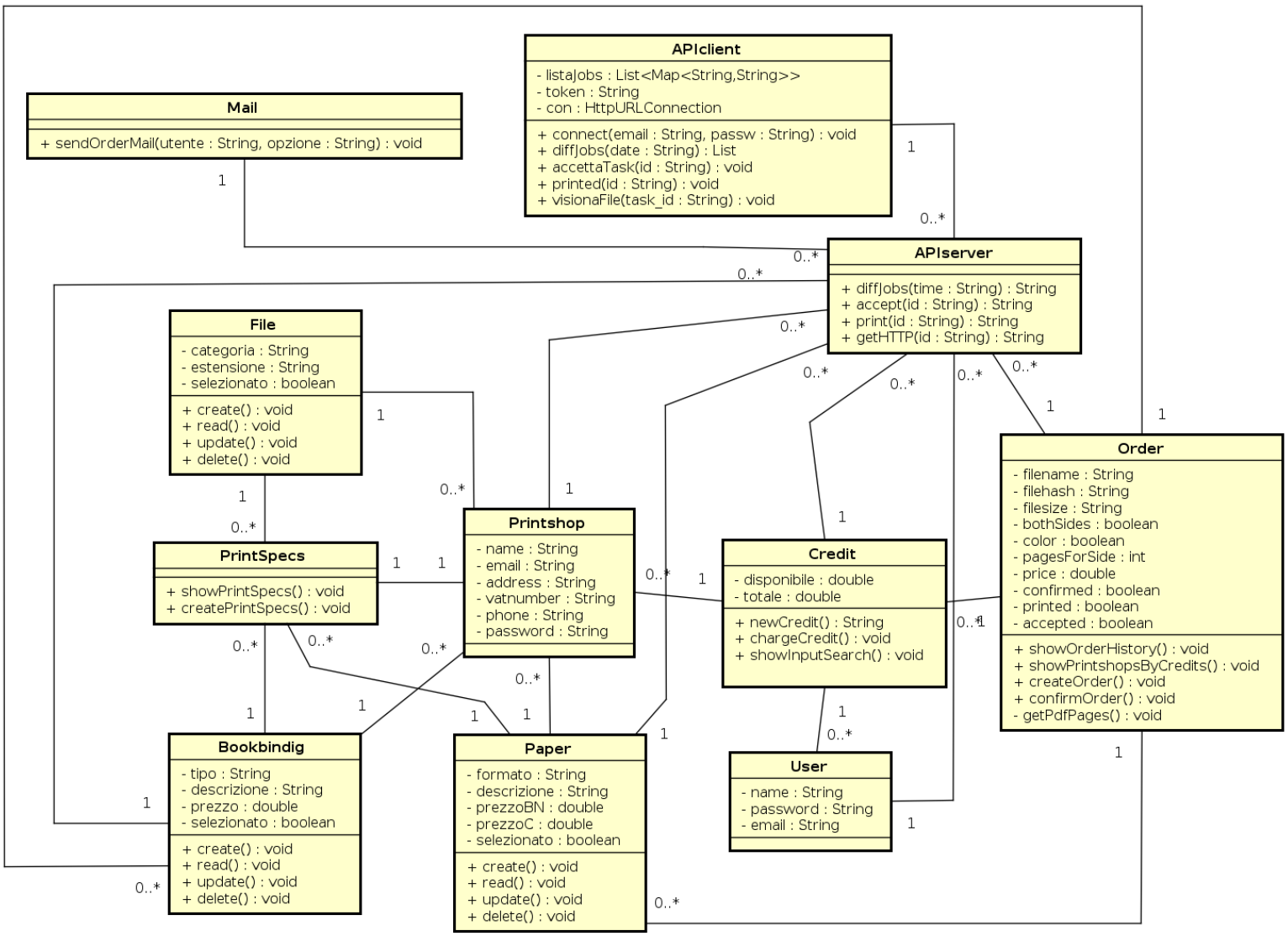

Credit
User
newCredit(user_email)
find(user_email, "where user_email = table.email")
user
alt : se user è null
null()
credit_id = store(user.id, auth()->printshop.id)
funzione laravel, consente di ottenere l'id univoco della copisteria che è autenticata e sta facendo la richiesta
credit_id()

3.4.2 – chargeCredit(credit_id : longInt, amount : float)



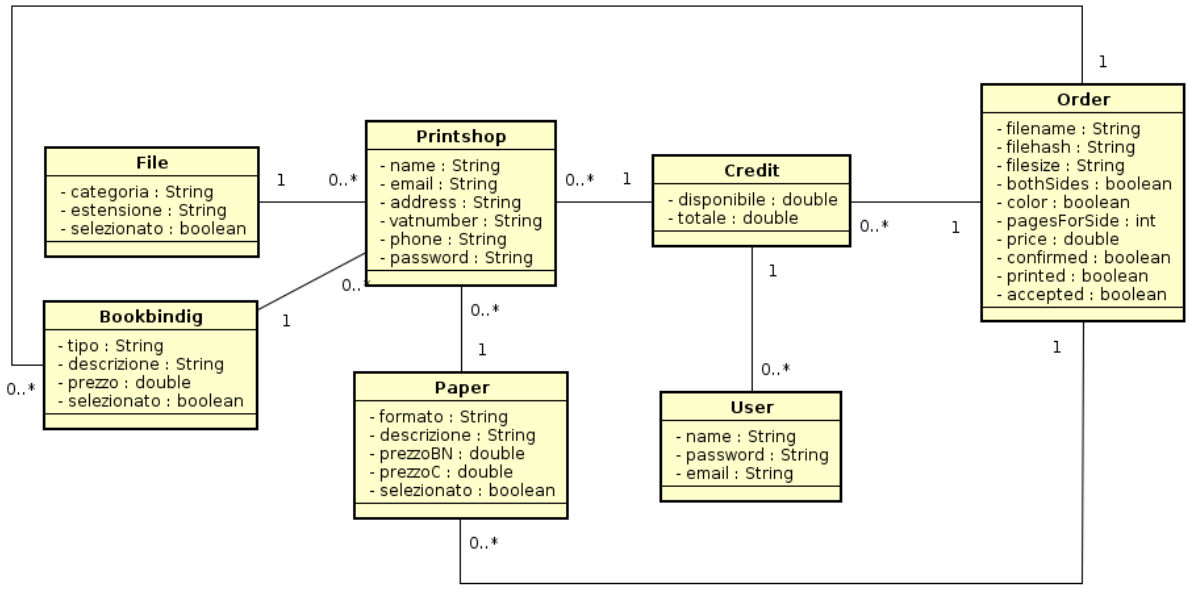
3.4.3 – Diagramma delle Classi

Sono stati inseriti i metodi di UC4 nella classe Credit.



3.4.4 – Modello dei dati

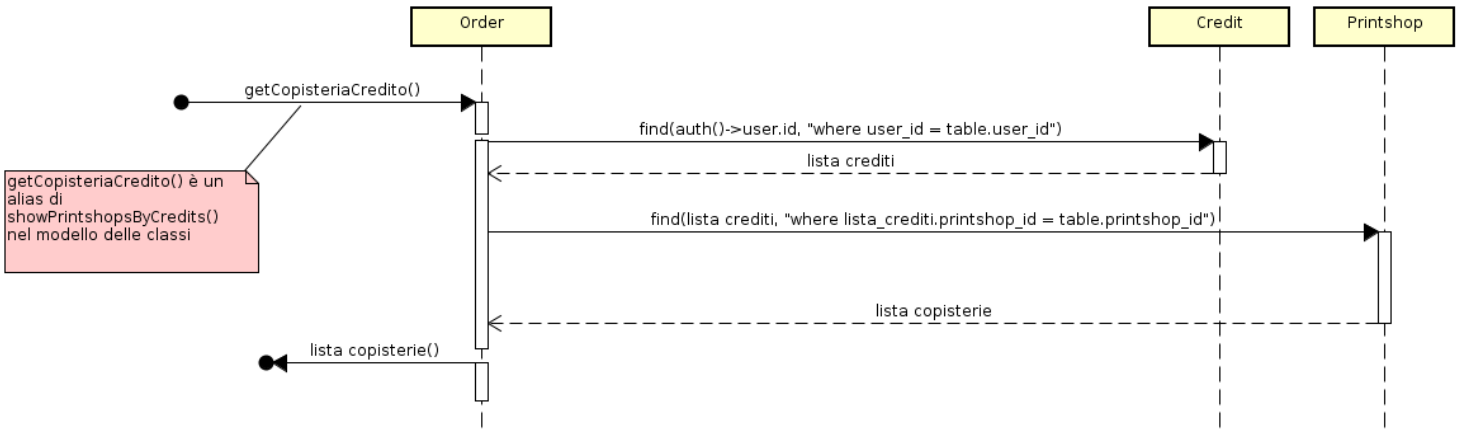
Il modello dei dati, poiché questo caso d'uso non prevede la memorizzazione dello storico delle ricariche credito, è rimasto invariato dopo la progettazione di UC4.



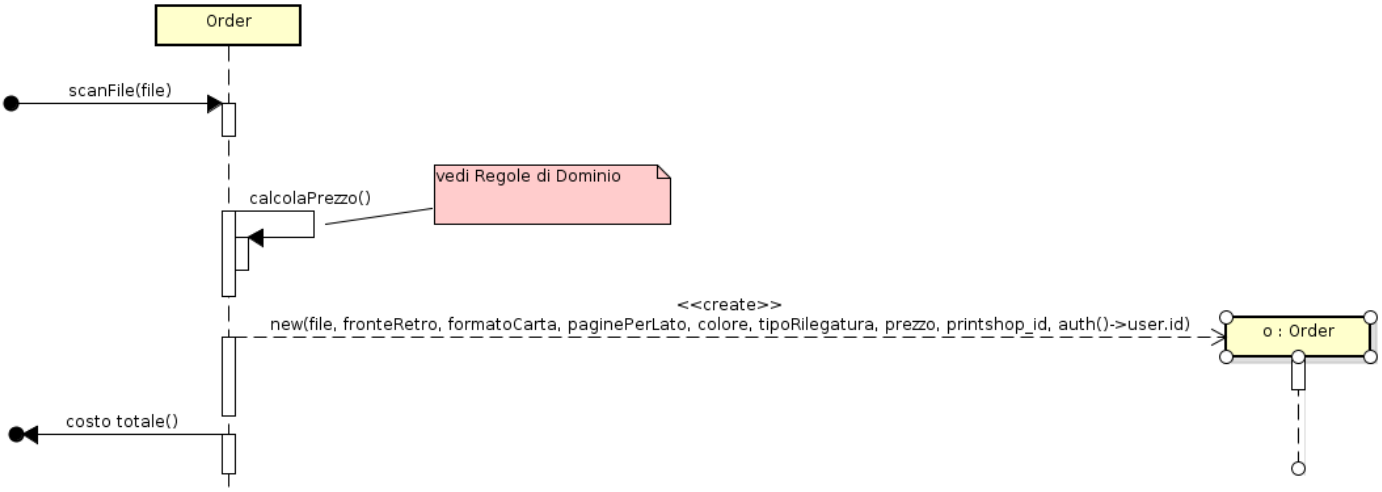
3.5 – Refactoring

Avendo precedentemente motivato la non trattazione dei casi d'uso di registrazione utente e copisteria in par 3.1, effettuiamo ora un refactoring che includa le funzionalità ottenute nel codice tramite i meccanismi discussi sopra.

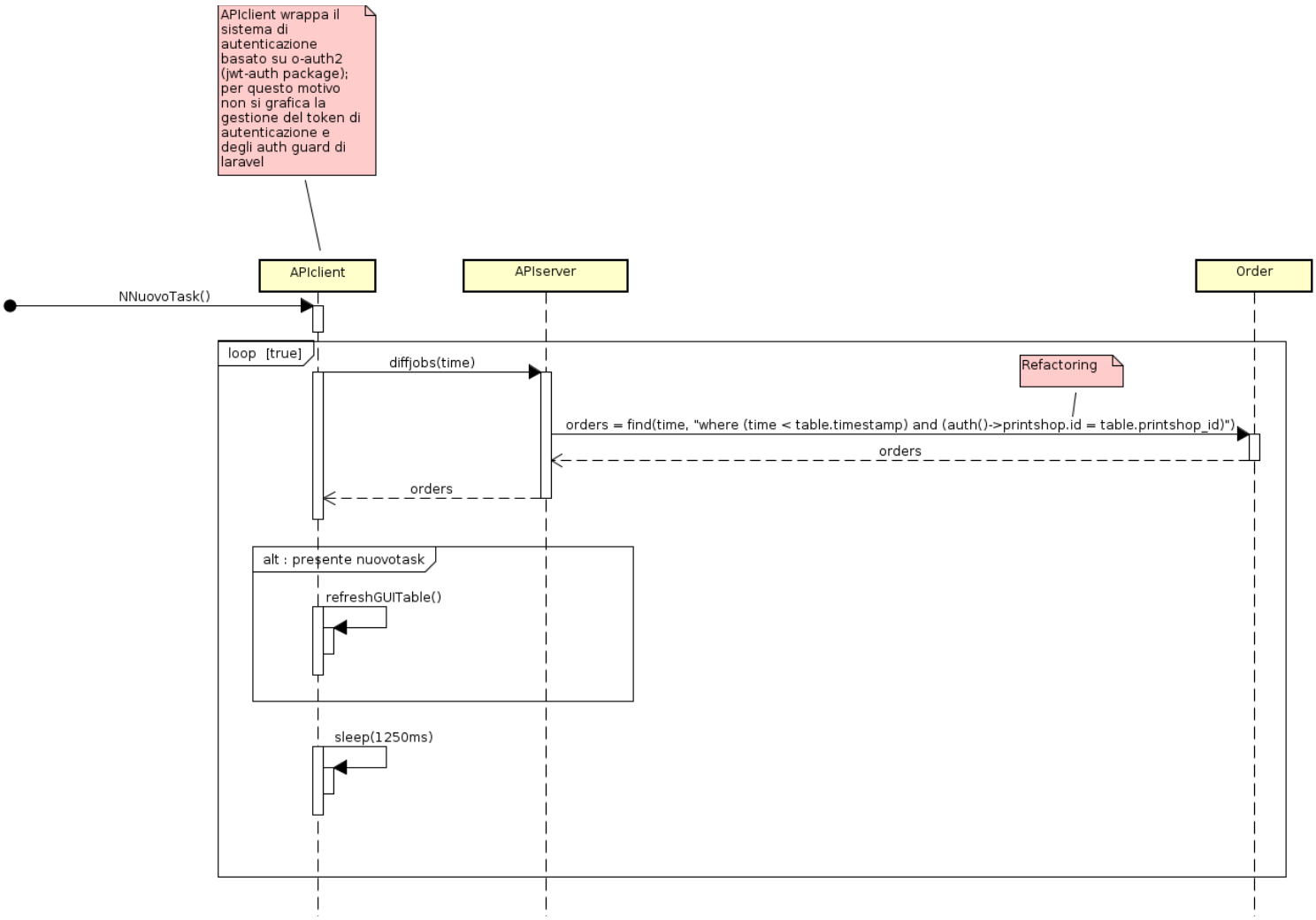
3.5.1 – getCopisteriaCredito [UC6]



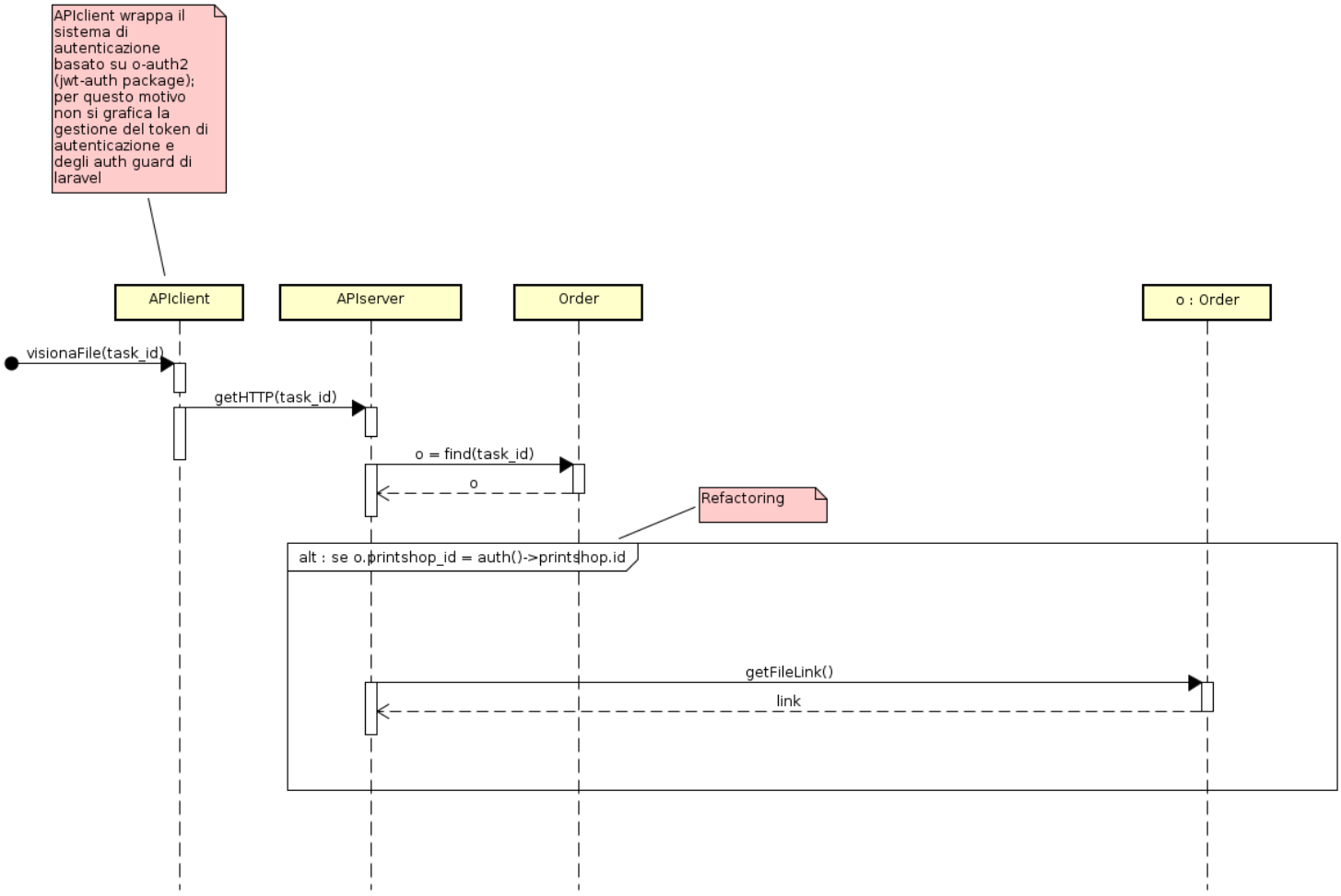
3.5.2 – scanFile [UC6]



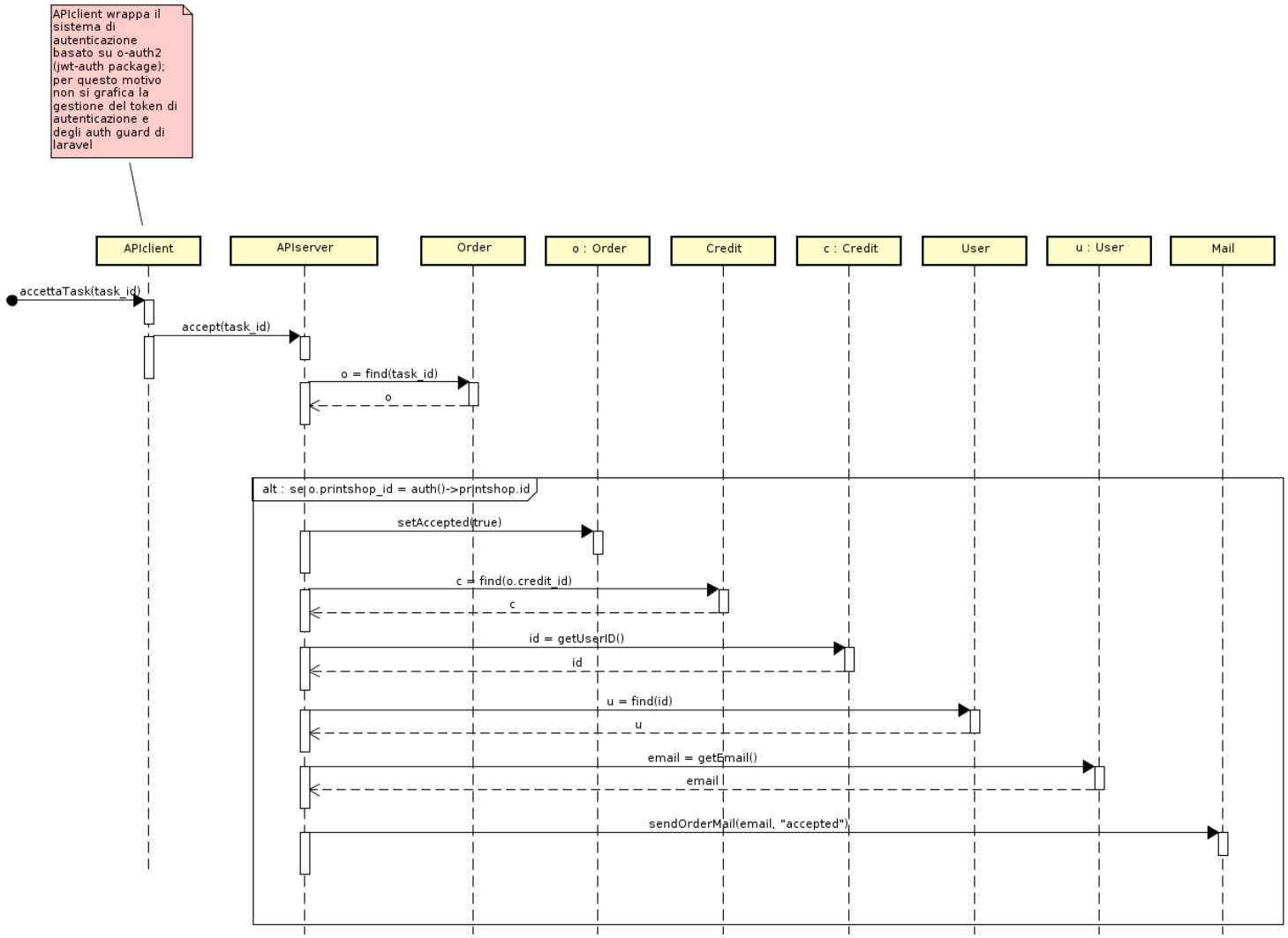
3.5.3 – NNuovoTask [UC7]



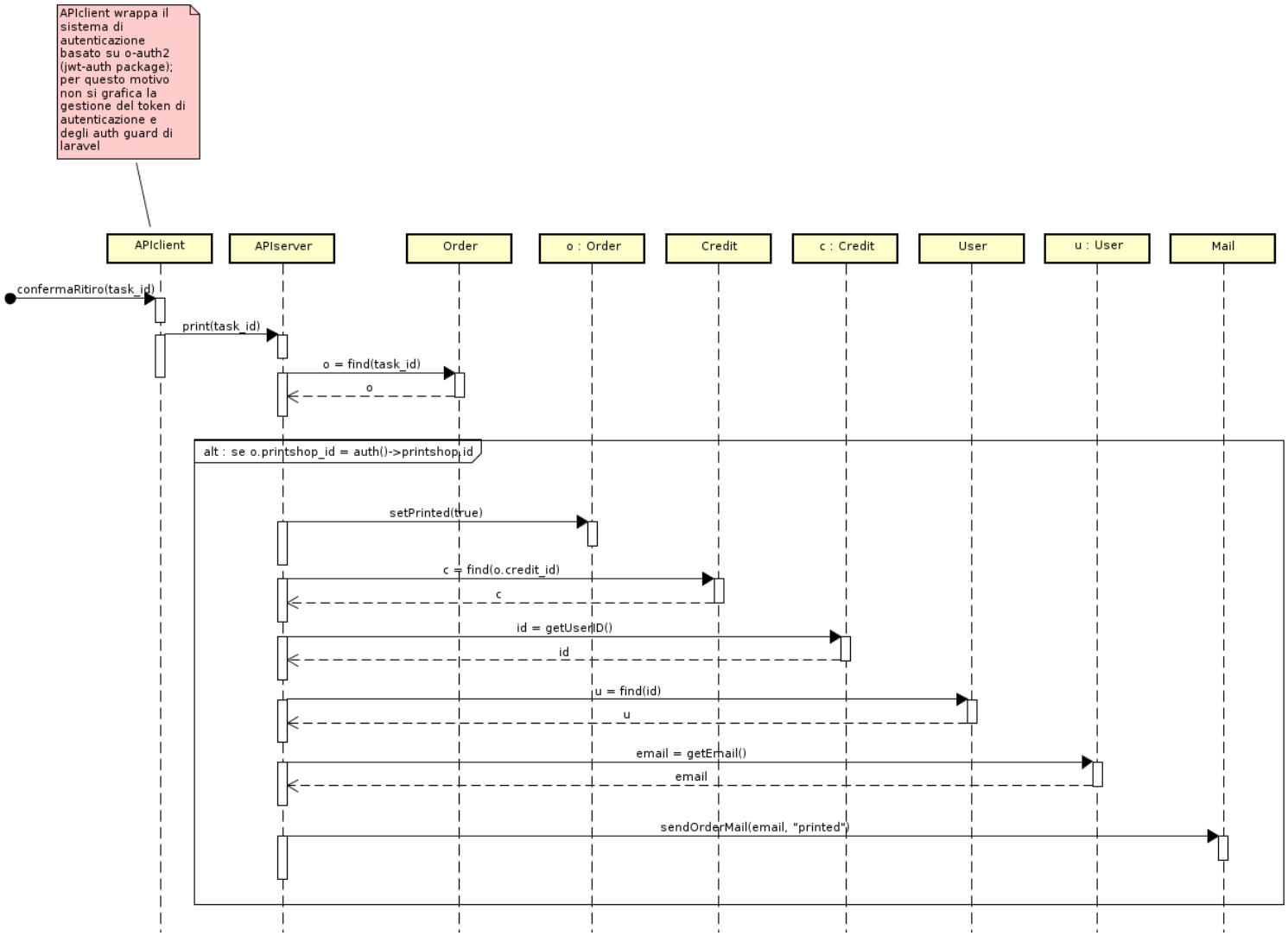
3.5.4 – visionaFile [UC7]



3.5.5 – accettaTask [UC7]



3.5.6 – confermaRitiro [UC7]



3.6 – Testing

Nonostante lo sviluppo di questo MVP di Copiga si sia effettuato tramite continui proof of concept e integrazione di questi nel codebase, si è comunque effettuata una fase di testing e analisi del codice da parte del team di sviluppo di Copiga e test da parte di colleghi esterni al progetto e familiari, ciò al fine di verificare l'intuitività del sistema e la resistenza agli errori commessi di un utente generico.

Successivamente a ciò si è passati all'implementazione di testing automatizzato al fine di ottenere il numero più grande di feedback possibile.

Il testing consiste nel trovare, attraverso metodologie sistematiche e ripetibili, le differenze tra il comportamento atteso del sistema e il suo comportamento effettivo a tempo di esecuzione.

Scopo del testing, quindi, è di massimizzare il numero di errori scoperti in modo da consentire agli sviluppatori di correggerli.

I test si dividono in due tipi: Test Funzionali e Unit Test.

I test funzionali servono per verificare l'interezza dell'applicazione e testano il software nella sua completezza, a livello generale. A questa categoria appartengono i test detti "Black Box", ossia test in cui non si conoscono i moduli e i comportamenti interni del sistema.

Gli Unit Test (o Test Unitari), eseguiti da noi, sono prove che vanno a verificare la correttezza direttamente dal codice, in ogni sua piccola parte. I nostri tool utilizzati per le nostre unit test sono JUnit per il lato esercente (JAVA) e PHPUnit per il lato web in Laravel.

I test eseguiti, sfruttano le tecniche di copertura delle funzionalità.
Eseguendo una veloce analisi del software JAVA, risulta evidente che le funzioni principali svolte dal software sono riuscire a connettere l'esercente al server principale e restituire l'elenco dei task per agevolare la gestione degli ordini di stampa. Le funzionalità rimanenti sono state tutte testate manualmente via Postman (un REST API development environment).
Lato server, invece, si è ritenuto che la classe Order rappresentasse il core dell'attività della piattaforma e che la creazione di un nuovo ordine dovesse avere un test.

3.6.1 – APIclient.connect()

E' il metodo che si occupa della connessione e autenticazione dell'esercente al server.

I test eseguiti sono tre:

1. Connessione se id e password sono corretti;
2. Connessione se id e password sono errati;
3. Connessione se prova a loggare un cliente della copisteria.

3.6.2 – APIclient.diffJobs()

E' il metodo che si occupa del fetch degli ordini di stampa inviati alla specifica copisteria non ancora flaggati come stampati.

I test eseguiti sono sei:

1. Ricerca di job di stampa se viene immessa la data di ricerca standard;
2. Ricerca di Job di stampa se viene immessa una data futura;
3. Ricerca di job di stampa con date scritte in formati differenti per controllare se il cast al tipo Date viene eseguito correttamente (2 casi);
4. Ricerca di job di stampa con la data dell'ultimo job;
5. Ricerca di job di stampa se al posto di una data viene inserita una stringa con un contenuto casuale.

Dall'ultimo test di APIclient.diffJobs() sono fuoriusciti bug lato server, in cui il cast non viene eseguito, non viene effettuato alcun controllo e viene restituito tutto l'elenco dei jobs.

3.6.2 – Order.createOrder()

È il metodo che si occupa di gestire la creazione di un ordine lato utente.

Il test eseguito verifica:

1. Autenticazione di utente con credito
2. Selezione di una copisteria presso la quale l’utente ha credito
3. Caricamento file
4. Inserimento specifiche di stampa

Durante l’esecuzione di questo test si è notato un bug di visualizzazzione (l’utente visualizzava nello storico ordini anche ordini non confermati, risolto facendo la query dal database con “where table.confirmed = true”.