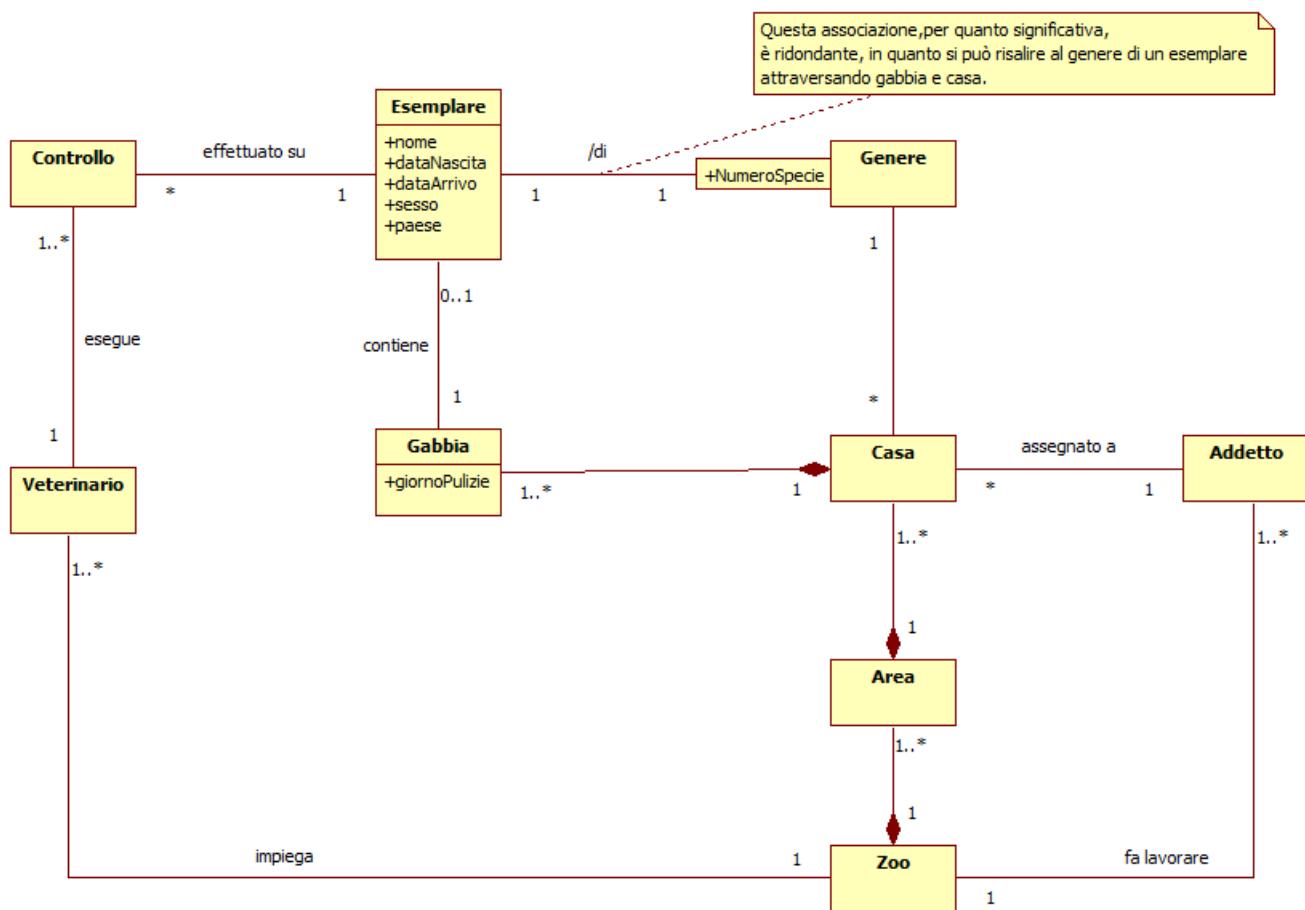


Diagrammi dei casi d'uso e delle classi

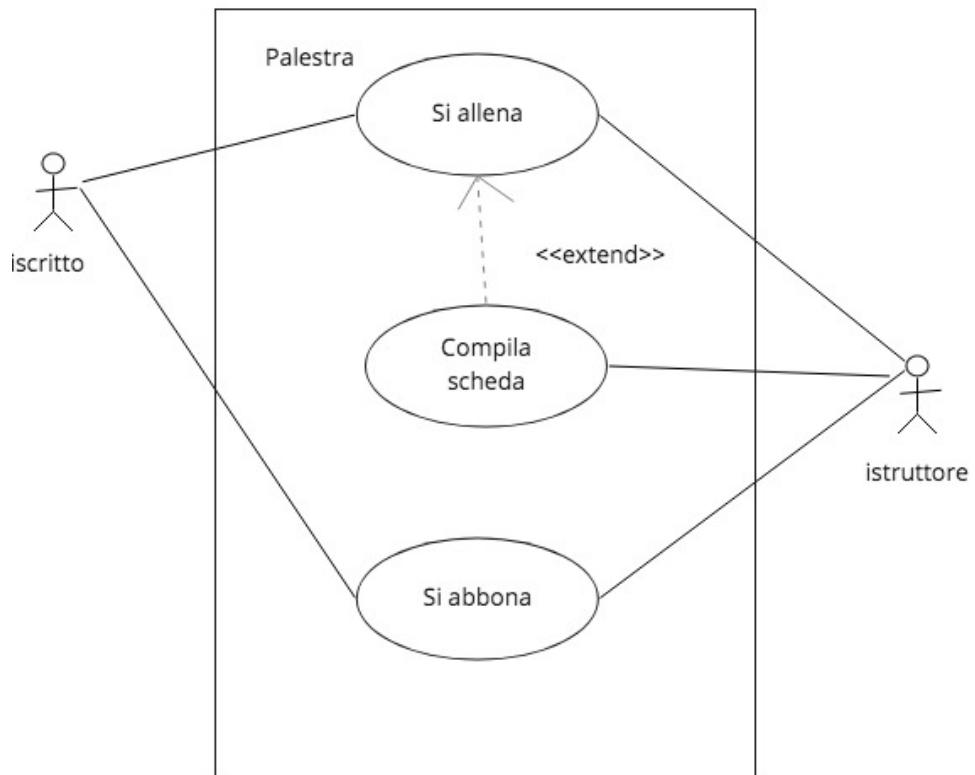
Si vuole automatizzare il sistema di gestione degli animali in uno zoo. Ogni esemplare di animale ospitato è identificato dal suo genere (es. zebra) e da un codice unico all'interno del genere di appartenenza. Per ogni esemplare si memorizzano la data di arrivo nello zoo, il nome proprio, il sesso, il paese di provenienza e la data di nascita. Lo zoo è diviso in aree; in ogni area c'è un insieme di case, ognuna destinata a un diverso genere di animali. Ogni casa contiene un insieme di gabbie, ognuna contenente un solo esemplare. Ogni casa ha un solo addetto che pulisce ciascuna gabbia in un determinato giorno della settimana. Gli animali sono sottoposti periodicamente a controllo veterinario; in un controllo, un veterinario rileva il peso di un esemplare, diagnostica un'eventuale malattia e prescrive il tipo di dieta da seguire.

Utilizzando il formalismo UML, si disegni il diagramma delle classi per il dominio applicativo descritto.

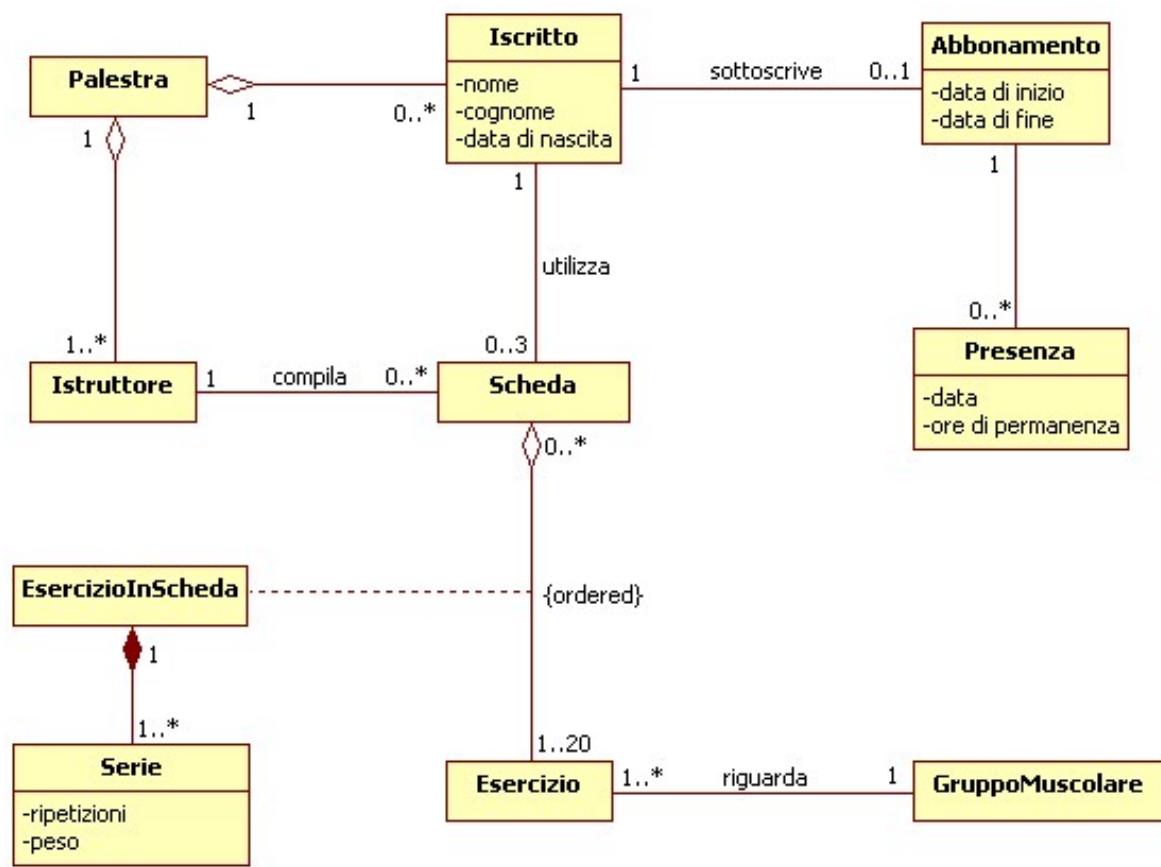


Si vuole automatizzare il sistema di gestione di una palestra. Per ogni iscritto si vogliono memorizzare, oltre ai dati anagrafici, le date di inizio e di fine abbonamento, le presenze effettive in palestra (ogni presenza è descritta dalla data e dal numero di ore di permanenza; le presenze vengono cancellate allo scadere dell'abbonamento) e le schede di allenamento da lui attualmente utilizzate. Una scheda descrive la sequenza di esercizi da effettuare durante un allenamento; un iscritto può usare contemporaneamente fino a tre schede (a rotazione). Una scheda, compilata da un istruttore, contiene fino a un massimo di 20 esercizi; ciascun esercizio deve essere svolto in un certo numero di serie, ciascuna caratterizzata da un certo numero di ripetizioni ed, eventualmente, da un peso utilizzato (esempi: "distensioni panca piana": 3 serie di cui la prima da 12 ripetizioni con peso 10 Kg, la seconda da 10 con 12 Kg, la terza da 8 con 14 Kg; "panca addominali": 2 serie da 20 ripetizioni). Ogni esercizio riguarda un gruppo specifico di muscoli.

Utilizzando il formalismo UML, si disegni il diagramma delle classi per il dominio applicativo descritto.

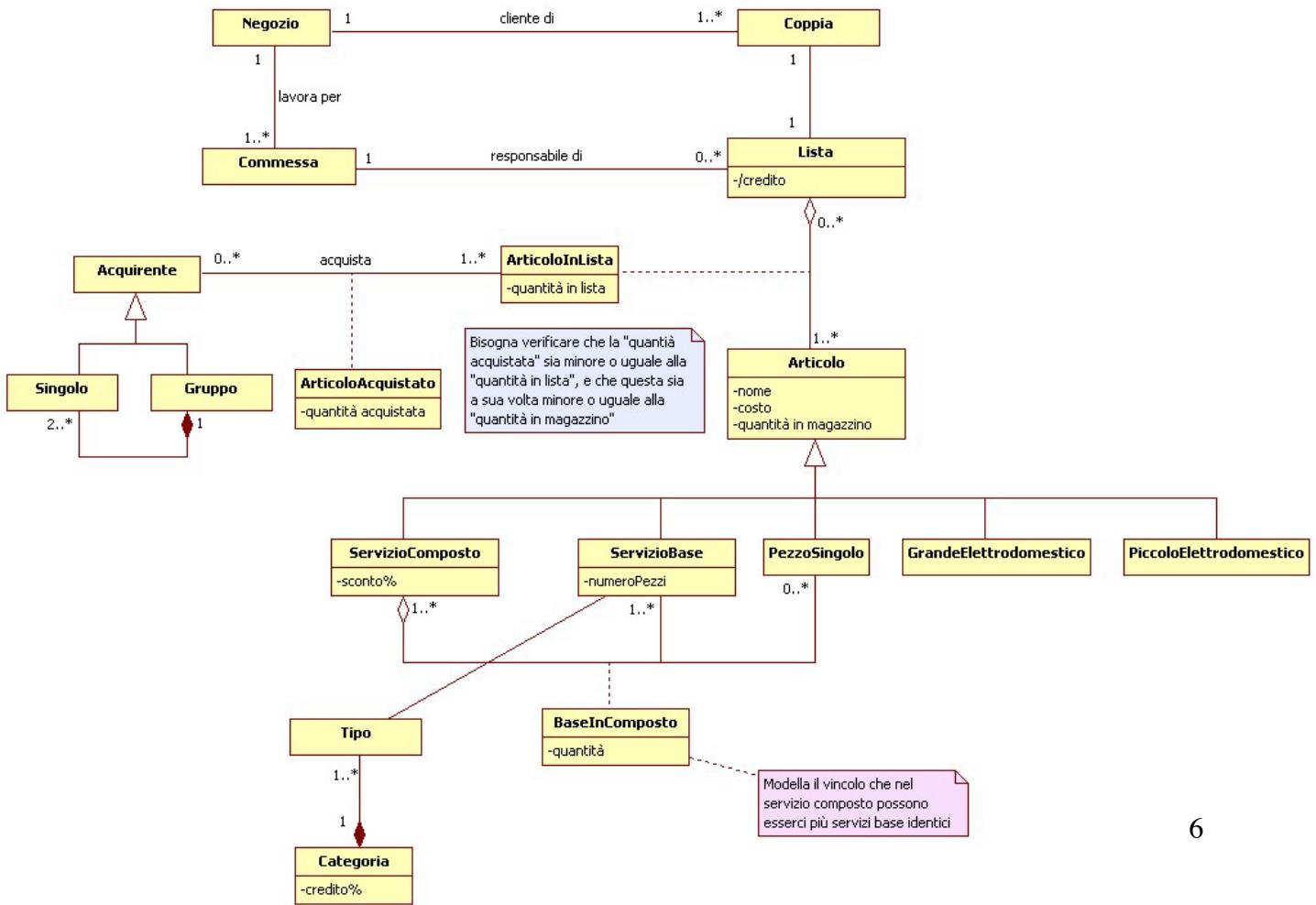
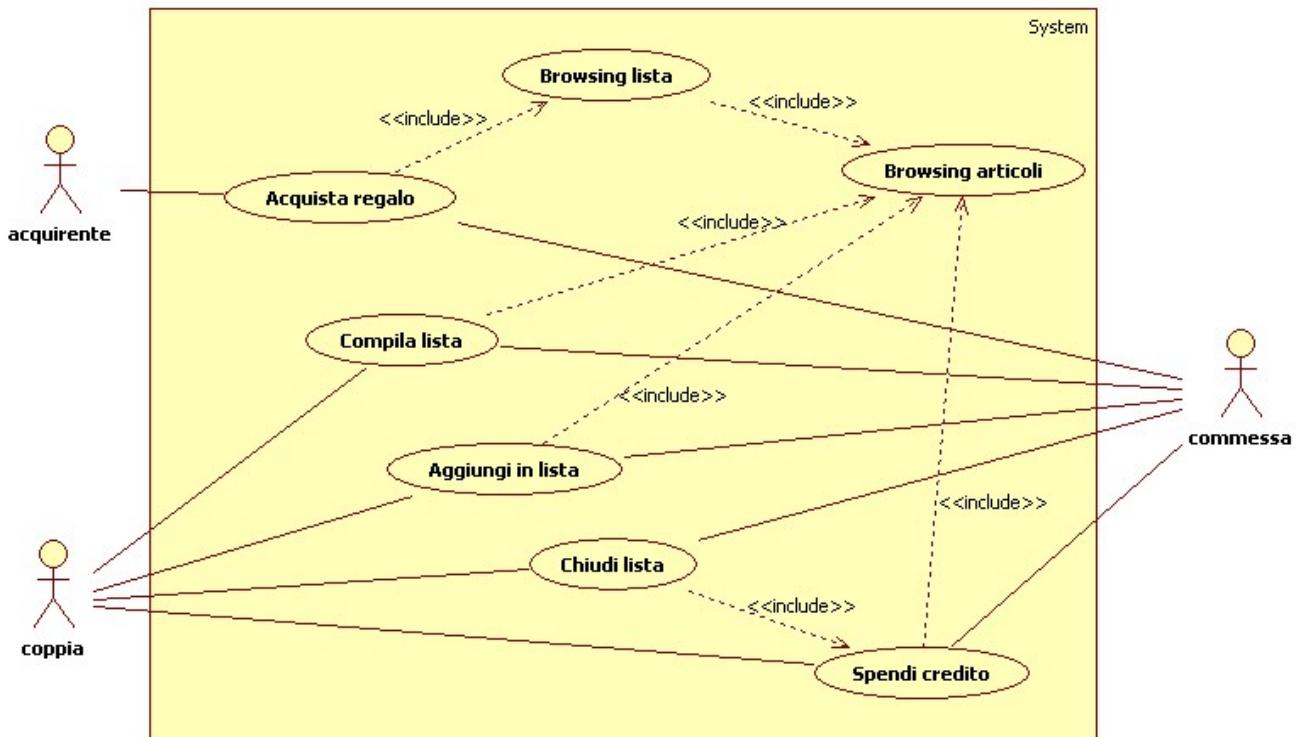


miro



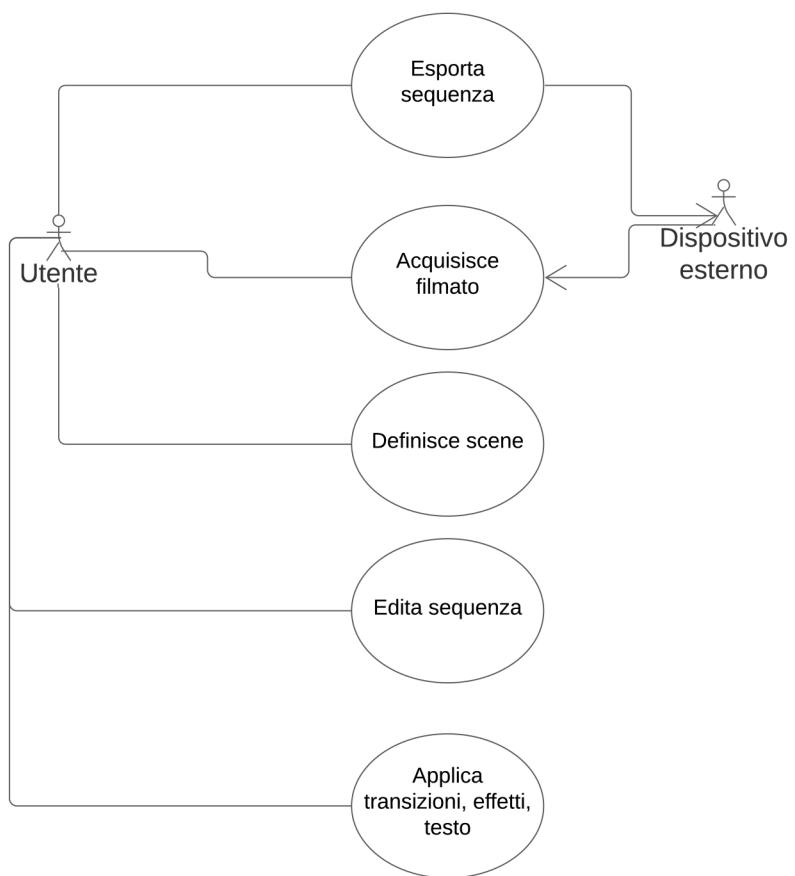
Un negozio di articoli per la casa gestisce elettronicamente le “liste di nozze”. Ogni coppia di futuri sposi si reca inizialmente presso il negozio e compila insieme a una commessa (che diventa la responsabile della lista) un elenco di articoli che desidera in regalo. Lo stesso articolo può esser inserito anche più volte nella stessa lista. Ogni articolo è caratterizzato da un nome, un costo e una quantità residua di magazzino. Gli articoli sono suddivisi in 5 tipologie: servizi composti, servizi base, pezzi singoli, piccoli elettrodomestici e grandi elettrodomestici. I servizi composti (ad esempio “collezione profilo oro”) sono costituiti da uno o più servizi base ed eventualmente uno o più pezzi singoli (ad esempio 2×6 piatti fondi serie classic+2×6 piatti serie classic+1 zuppiera). Ogni servizio base (ad esempio “6 piatti fondi serie classic”) è a sua volta costituito da uno più oggetti identici e risulta (contrariamente ai servizi composti) “non spezzabile”, ovvero può essere regalato solo completo; il “tipo” di un servizio base classifica gli oggetti che ne fanno parte (ad esempio bicchieri da acqua, forchette, piatti), i tipi sono ulteriormente raggruppati in categorie (ad esempio bicchieri, posate). Alle collezioni composte è associato uno sconto che viene applicato solo nel caso in cui la collezione venga regalata completa. Coloro che desiderano fare regali alla coppia si recano presso il negozio, contattano la commessa responsabile della lista e scelgono dalla lista uno o più articoli (anche più articoli dello stesso tipo). Un regalo può essere fatto da un singolo o da un gruppo costituito da 2 o più persone. Ogni coppia di sposi accumula un credito pari a una percentuale (diversa per ogni categoria di articoli) del costo degli articoli ricevuti in regalo. Se la lista viene esaurita, gli sposi possono aggiungere altri articoli. Una volta chiusa la lista, la coppia può spendere il credito accumulato acquistando ulteriori articoli anche non presenti in lista. Una lista chiusa viene mantenuta in memoria per 1 anno.

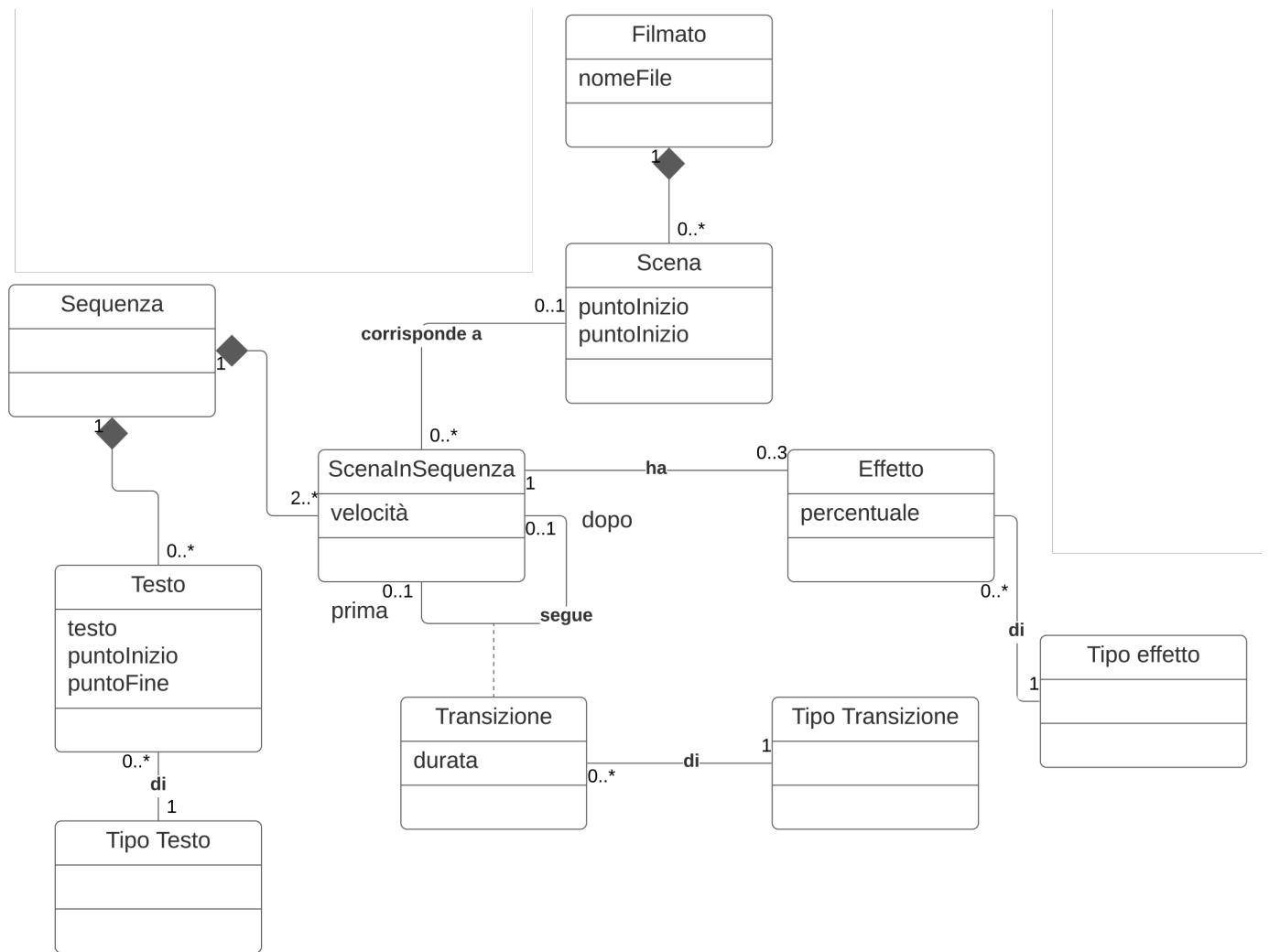
Utilizzando il formalismo UML, si disegnino il diagramma dei casi d'uso e delle classi per il dominio applicativo descritto.

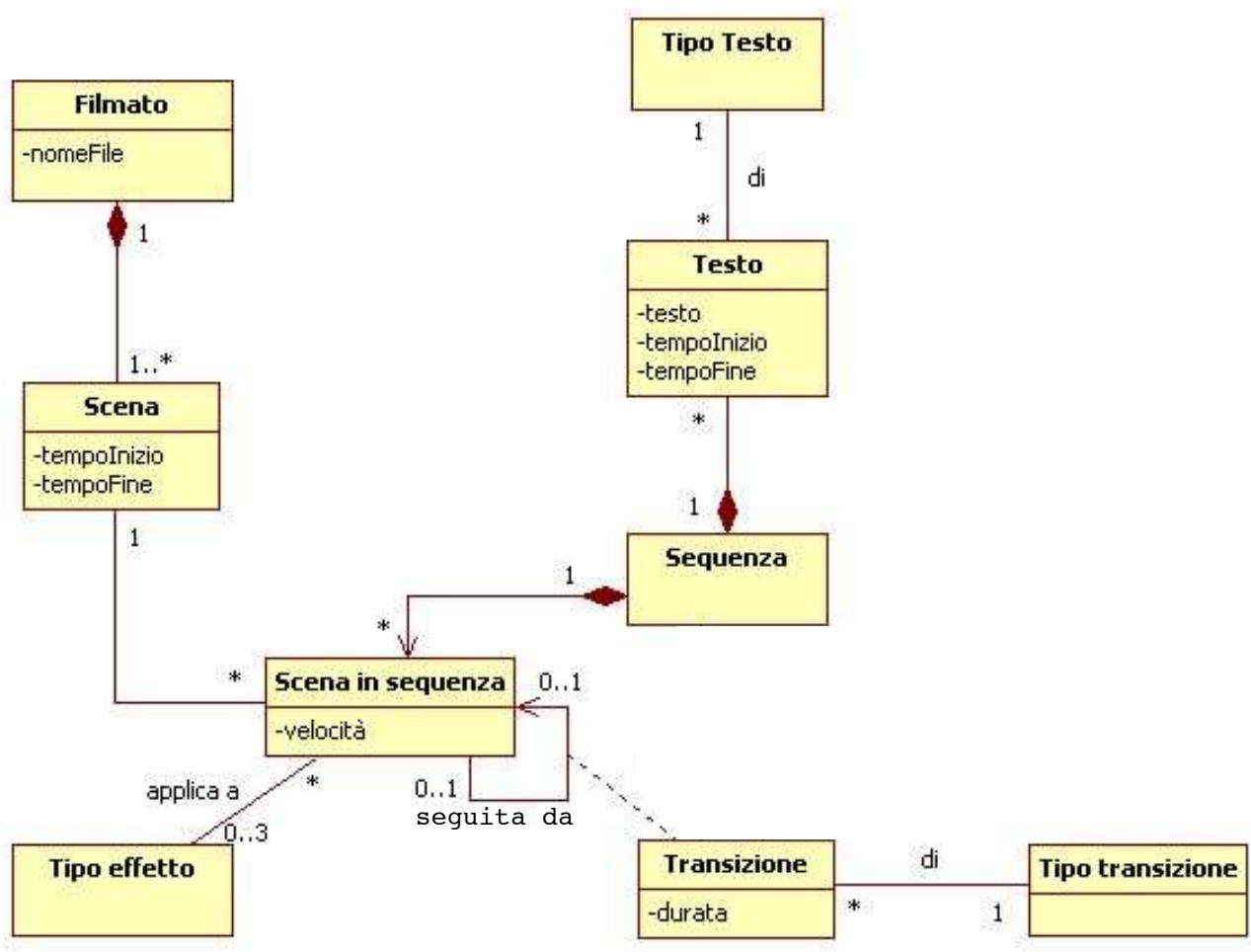


Si vuole modellare un'applicazione per il montaggio professionale di sequenze video. L'applicazione permette di acquisire diversi filmati, ciascuno memorizzato sotto forma di file. Da ogni file è possibile estrarre una o più scene, ciascuna definita da un punto di inizio e un punto di fine. Una sequenza video comprende un sequenza di scene, unite tra loro tramite transizioni di tipi diversi (ad es. fade incrociato). All'interno di una sequenza, a una scena è possibile applicare fino a 3 effetti digitali (ad es. solarizzazione); inoltre, la scena può essere a velocità normale, accelerata o rallentata. In qualunque punto della sequenza è possibile sovrapporre un testo, secondo diverse modalità (ad es. scorrimento dall'alto).

Utilizzando il formalismo UML, si disegnino il diagramma dei casi d'uso e delle classi per il dominio applicativo descritto.

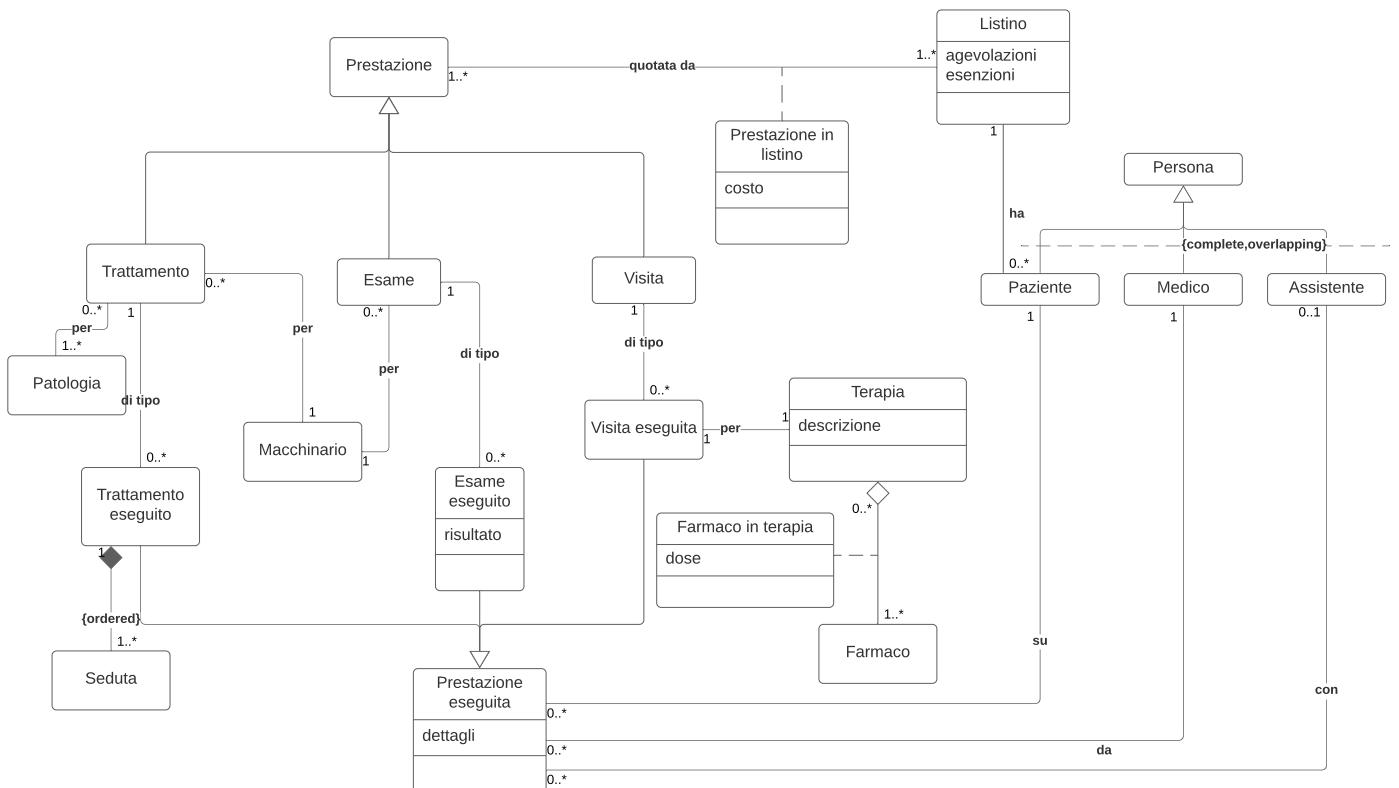






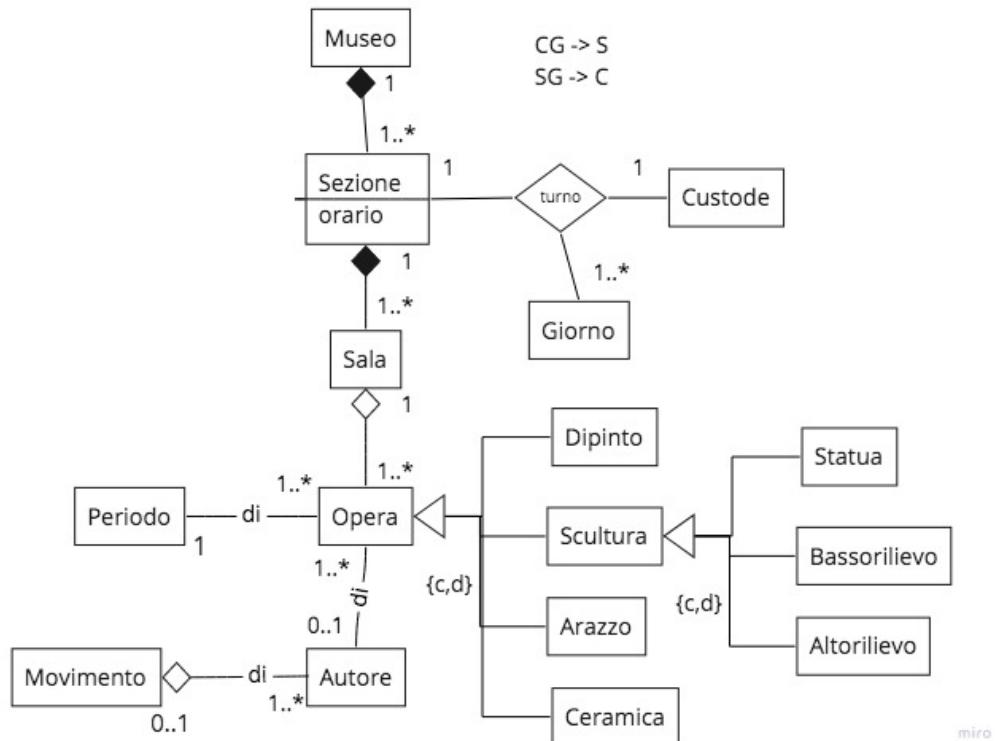
Uno studio medico vuole automatizzare il proprio sistema di memorizzazione delle prestazioni eseguite. Lo studio è specializzato nel trattamento continuativo di particolari patologie mediante l'uso di appositi macchinari; un particolare trattamento va eseguito sempre con lo stesso macchinario ed è solitamente composto da diverse sedute. Altre tipologie di prestazioni sono gli esami e le visite specialistiche. Anche gli esami, come i trattamenti, prevedono l'utilizzo di un macchinario appropriato al tipo d'esame. Per ogni seduta di trattamento interessa l'ordine in cui viene eseguita rispetto alle altre, per ogni esame interessa il risultato mentre per ogni visita interessa la terapia medica da attuare. La terapia consiste in una breve descrizione e in un elenco di quali farmaci assumere e secondo quali dosi. Ogni prestazione eseguita è corredata da un testo che ne riporta i dettagli, mentre ogni tipologia di prestazione ha una semplice descrizione. Ogni prestazione su un paziente è eseguita da un medico che può avvalersi, in alcuni casi, dell'aiuto di un assistente (non medico). I pazienti, i medici e gli assistenti sono caratterizzati da una breve anagrafica. Inoltre, a ogni paziente è associato un particolare listino che ne riassume le esenzioni e qualsiasi altra agevolazione (da rappresentare con una semplice descrizione). Per ogni listino e per ogni tipologia di prestazione è riportato il prezzo da applicare che, nel caso dei trattamenti, va associato all'intero trattamento e non alle singole sedute.

Utilizzando il formalismo UML, si disegni il diagramma delle classi per il dominio applicativo descritto.



Un museo si compone di diverse sezioni, ciascuna comprendente un certo numero di sale. Ogni sezione ha un orario di visita ed è custodita giornalmente da un solo custode, nel rispetto di un turno settimanale che resta invariato per tutto l'anno; il turno definisce per ciascun custode la sezione di cui si deve occupare in ciascun giorno della settimana. Ciascuna sala comprende diverse opere d'arte; le opere d'arte si dividono in dipinti, sculture (a loro volta distinte in statue, bassorilievi e altorilievi), arazzi e ceramiche. Ogni opera ha un autore, ma di alcune non se ne conosce il nome. Ogni opera appartiene a un periodo storico (ad es. Rinascimento, Ottocento, ecc.); alcuni autori appartengono a un movimento artistico (ad es. Impressionismo, Puntoinismo).

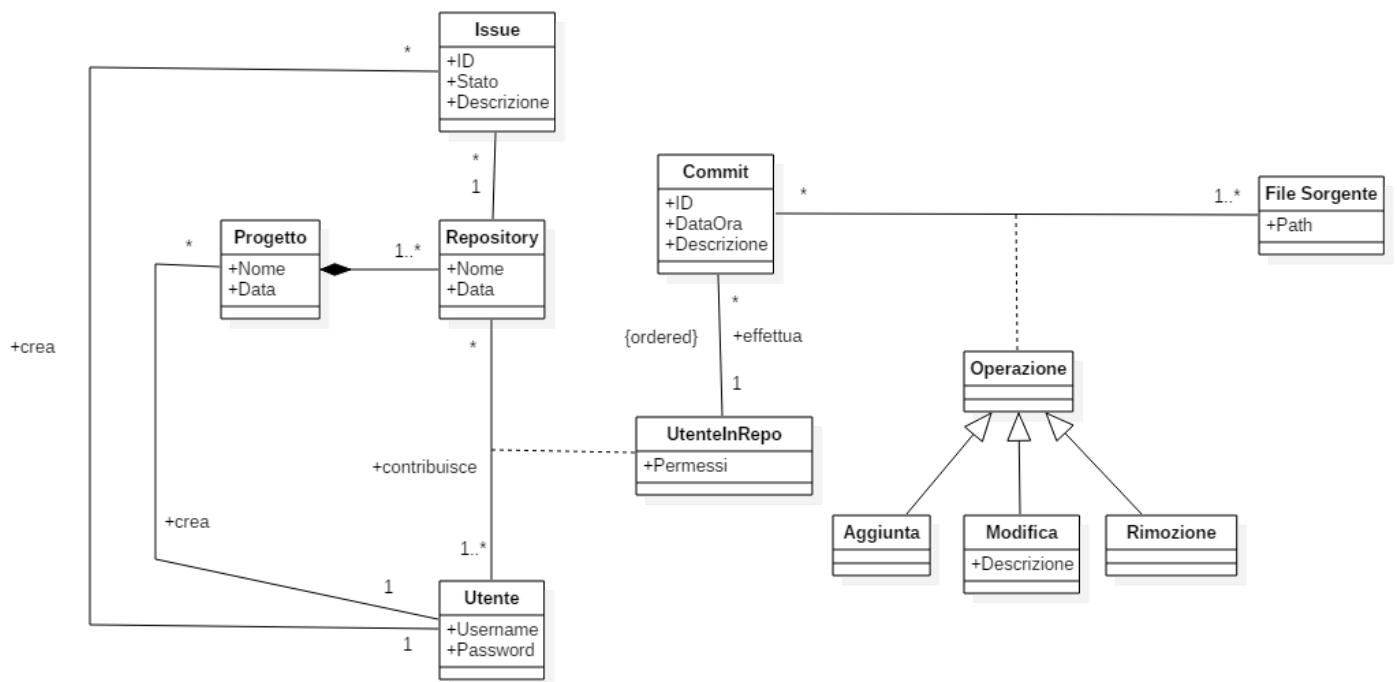
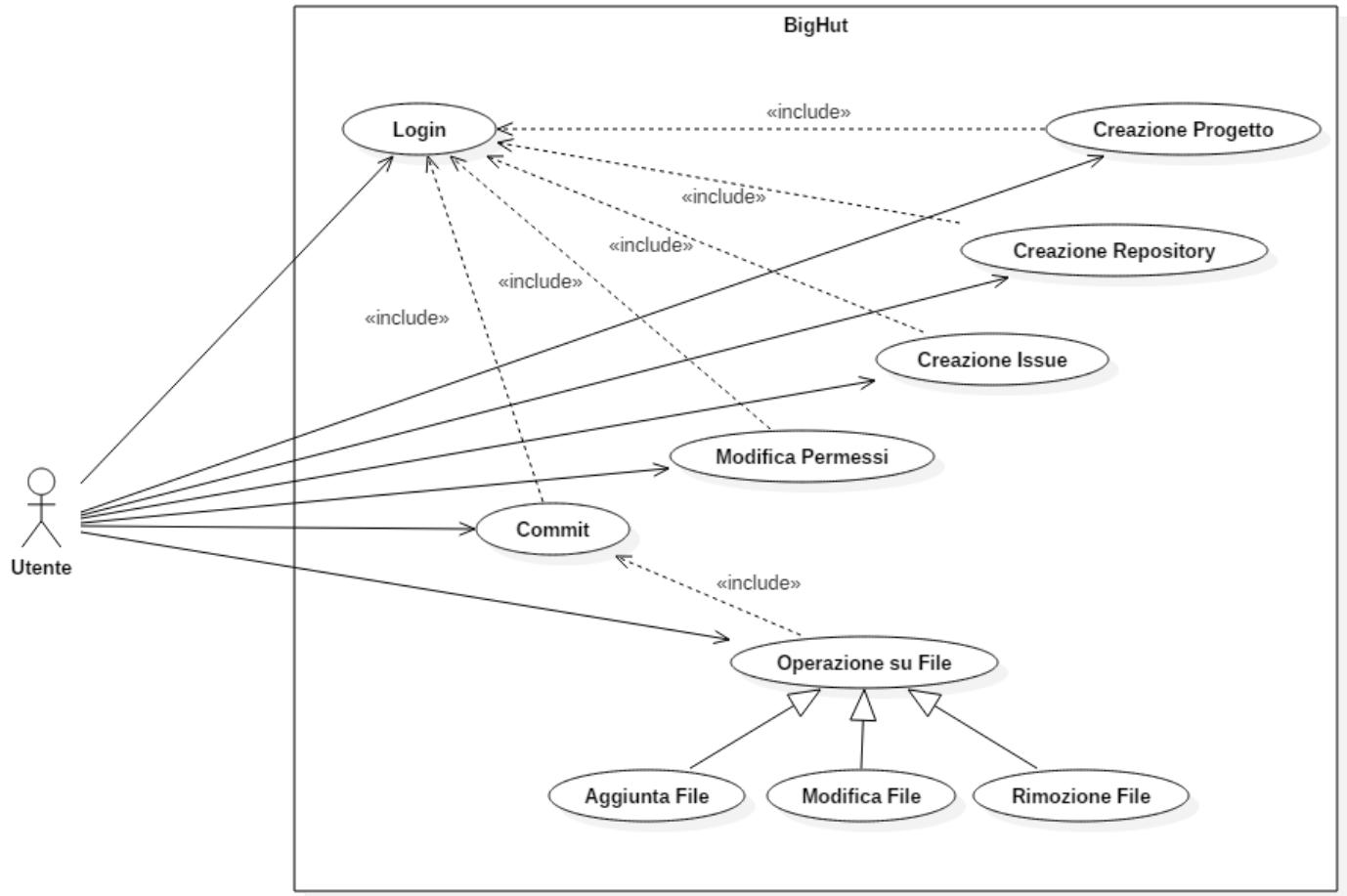
Utilizzando il formalismo UML, si disegni il diagramma delle classi per il dominio applicativo descritto.



Il sistema web-based BigHut per il versioning del codice mette a disposizione dei suoi utenti un servizio di hosting e gestione di repository di codice. Per accedere ai vari servizi, un utente deve effettuare il login tramite username e password. Dopo il login, un utente può creare un nuovo progetto e, al suo interno, inserire più repository; sia progetti che repository sono definiti da nome e data di creazione. Per ciascun repository, l'utente creatore del progetto corrispondente può concedere ad altri utenti dei permessi, per esempio lettura e modifica (utilizzare un generico campo testuale per descriverli tutti). Ogni repository è formato da una sequenza di commit (ognuno definito da un ID e una data) che coinvolge uno o più file sorgente (ognuno definito da un path). Ogni commit è effettuato da un utente (tra quelli che contribuiscono a quel repository) e può esprimere diverse operazioni, una per ogni file sorgente coinvolto. Un'operazione può essere di tre tipi: aggiunta, modifica (descritta da un generico campo testuale) e rimozione. Esempio: con un singolo commit è possibile aggiungere un nuovo file sorgente e modificarne un altro. Infine, un utente può aprire una issue (definita da un ID e una descrizione) relativa a uno specifico repository per segnalare eventuali bug, proposte di miglioramento, ecc. Lo stato iniziale di una issue è "aperto", ma può in seguito diventare "chiuso" (per esempio, perché il bug viene risolto).

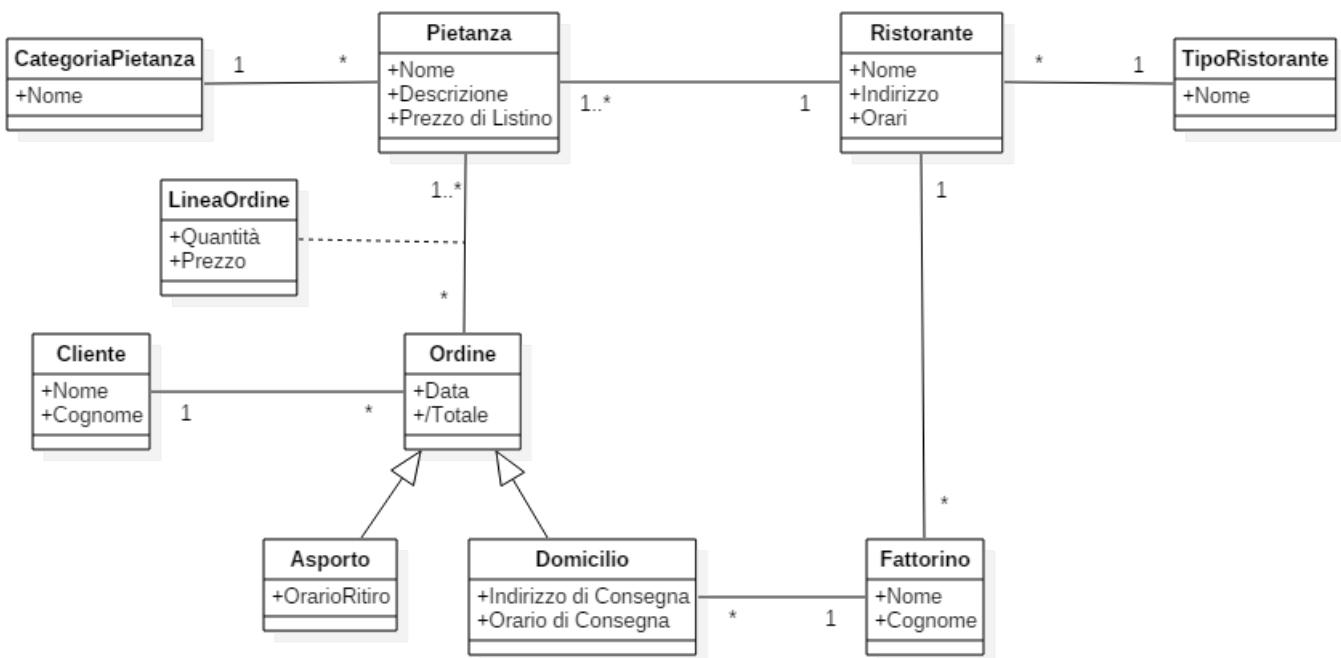
Si modellino le specifiche sopra riportate in UML attraverso:

- 1) un diagramma dei casi d'uso
- 2) un diagramma delle classi



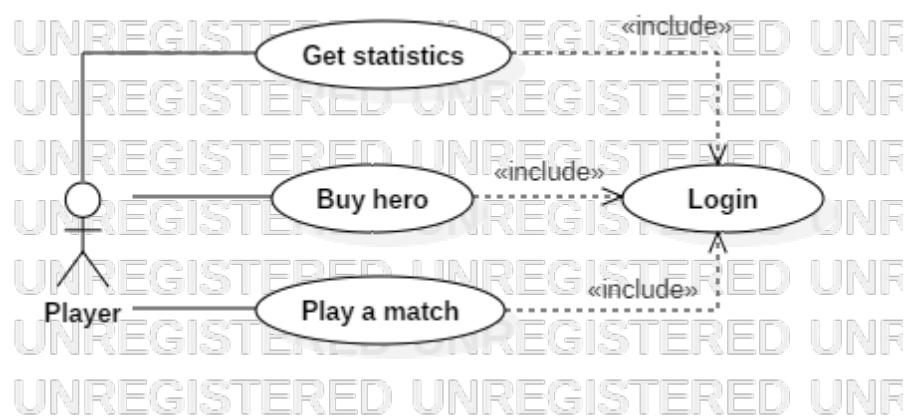
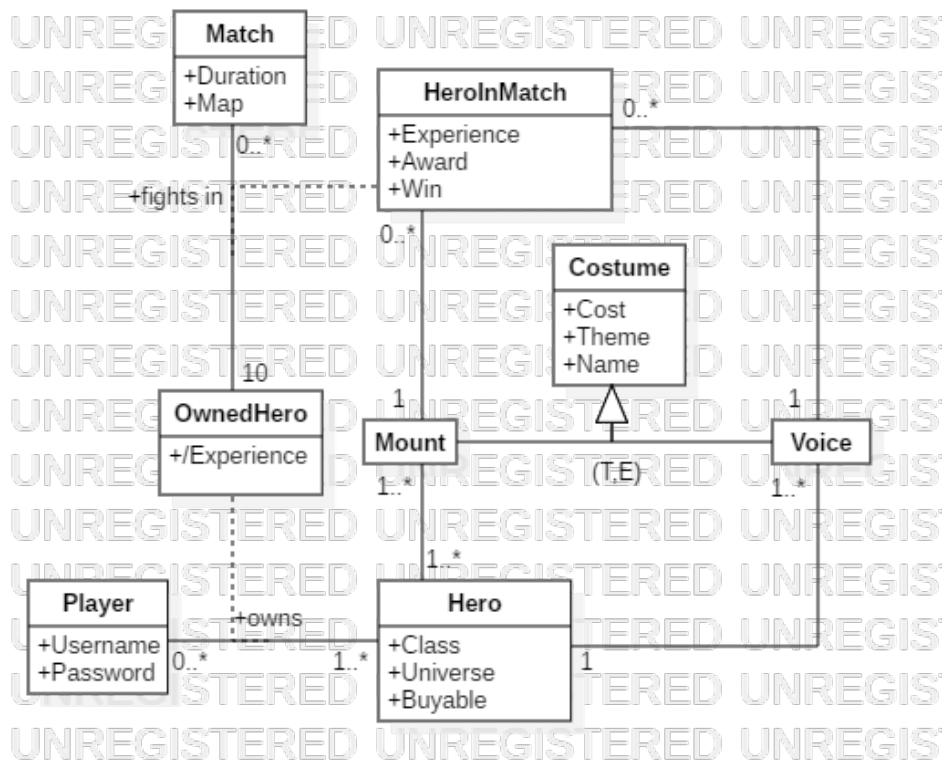
Tramite l'applicazione online GiastIt è possibile ordinare pietanze dai numerosi ristoranti convenzionati, ciascuno appartenente a una categoria (per esempio, ristorante cinese). Per ogni ristorante è possibile visualizzare informazioni quali nome, indirizzo e orari di apertura. Ogni ristorante mette inoltre a disposizione un elenco delle pietanze (nome, descrizione e prezzo unitario) che è possibile ordinare. Per facilitarne la consultazione, tutte le pietanze vengono organizzate in categorie (es. antipasti, carne, etc.); una pietanza fa riferimento ad una ed una sola categoria. I clienti registrati possono effettuare ordini direttamente online e possono scegliere tra l'opzione asporto o a domicilio. Ad ogni ordine è associato l'orario per cui dovrà essere pronto; per ognuna delle pietanze richieste è specificata la quantità. Nel caso di ordini a domicilio si vuole inoltre conoscere l'indirizzo di consegna e il fattorino incaricato (ogni fattorino fa riferimento ad uno ed un solo ristorante).

Utilizzando il formalismo UML, si disegni il diagramma delle classi per il dominio applicativo descritto.



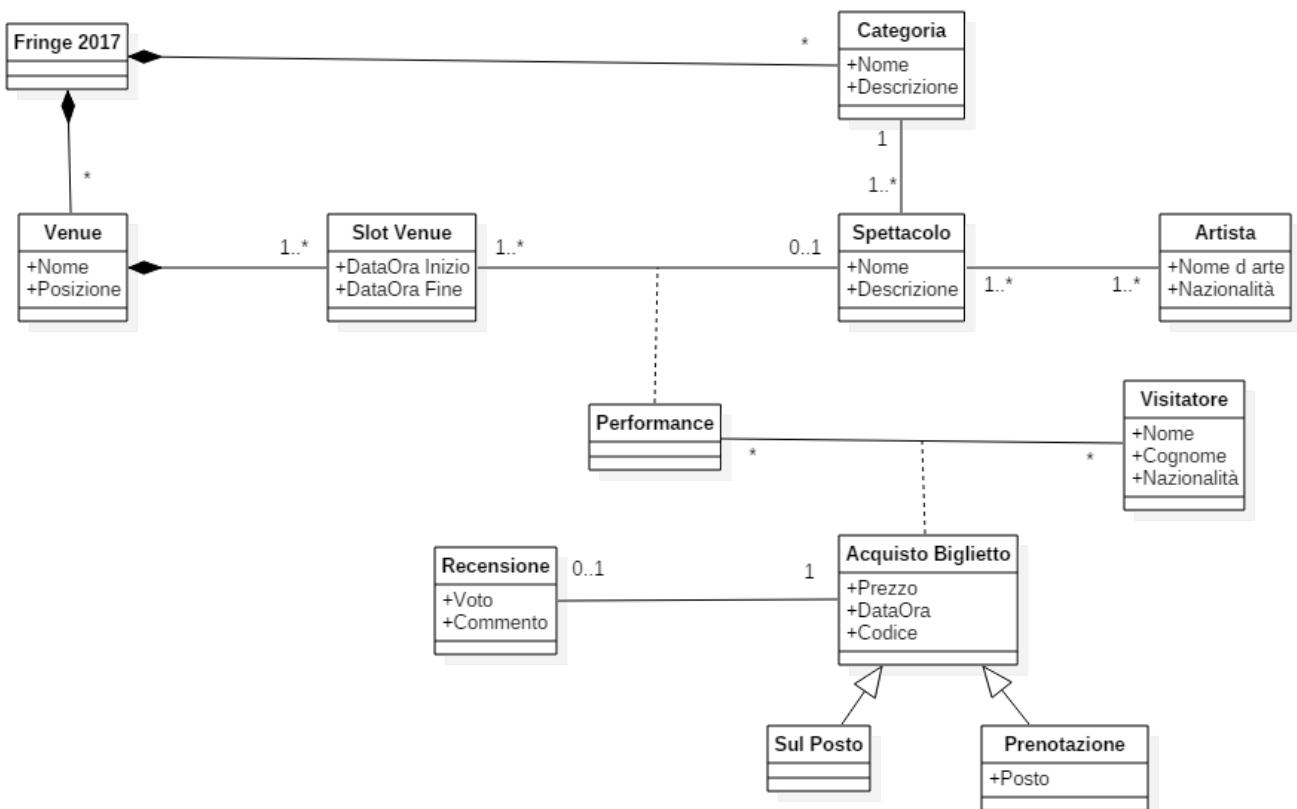
Si vuole modellare tramite diagramma delle classi il funzionamento di HOST, un famoso gioco multiplayer. Per prendere parte al gioco, ogni giocatore si autentica con username e password. Effettuato l'accesso, al giocatore appare una schermata di eroi di cui si conoscono la classe (guerriero, specialista, etc.) e l'universo di appartenenza. Il gioco è organizzato in match a cui prendono parte 10 eroi. Un giocatore può prendere parte ad un match con un singolo eroe. Tra tutti gli eroi disponibili, il giocatore seleziona un eroe in suo possesso (alcuni eroi sono disponibili a tutti, altri devono essere comprati) per prendere parte al match (di cui si conoscono la durata e la mappa in cui si svolge il match). Ad ogni match l'utente assegna due tipi di elementi costume al proprio eroe (selezionabili tra quelli disponibili per il proprio eroe): la cavalcatura e la voce. Di tali elementi si conosce il nome, il tema (halloween, natalizio, estivo, ...) e il costo. In particolare, una cavalcatura può essere assegnata a più eroi e una voce può essere assegnata a un solo eroe. Al termine di ogni match, per ogni eroe si registrano e visualizzano i punti esperienza ricevuti, l'esito del match (vittoria/sconfitta) e optionalmente un award.

Utilizzando il formalismo UML, si disegnino il diagramma delle classi e dei casi d'uso per il dominio applicativo descritto.



Ogni anno, durante il mese di agosto, a Edimburgo si svolge il festival Fringe, nel quale si radunano artisti e visitatori da tutto il mondo. Durante il Fringe, molti pub, teatri ed edifici pubblici mettono a disposizione i loro locali per ospitare vari spettacoli. Ogni venue (descritta da un nome ed una posizione geografica) mette a disposizione degli slot temporali, definiti da una data/ora iniziale ed una data/ora finale, per le performance dei vari spettacoli (ognuno definito da un nome ed una descrizione). Ovviamente, in ogni slot di una data venue può essere eseguita una sola performance. Ogni spettacolo appartiene inoltre ad una determinata categoria ed è organizzato ed eseguito da un uno o più artisti; ogni artista è descritto da un nome d'arte e dalla sua nazionalità. Per poter assistere ad una performance, un visitatore (descritto da nome, cognome e nazionalità) deve acquistare il corrispettivo biglietto. L'acquisto può essere effettuato sia sul posto, oppure online via prenotazione. Nel secondo caso viene anche assegnato un numero di posto riservato all'acquirente. Ogni biglietto, acquistato tramite prenotazione o meno, espone la data/ora di acquisto, il prezzo e un codice che il visitatore può utilizzare (accedendo al sito del festival) per rilasciare una recensione sulla performance; la recensione è costituita da una valutazione (da 1 a 5 stelle) e da un commento testuale.

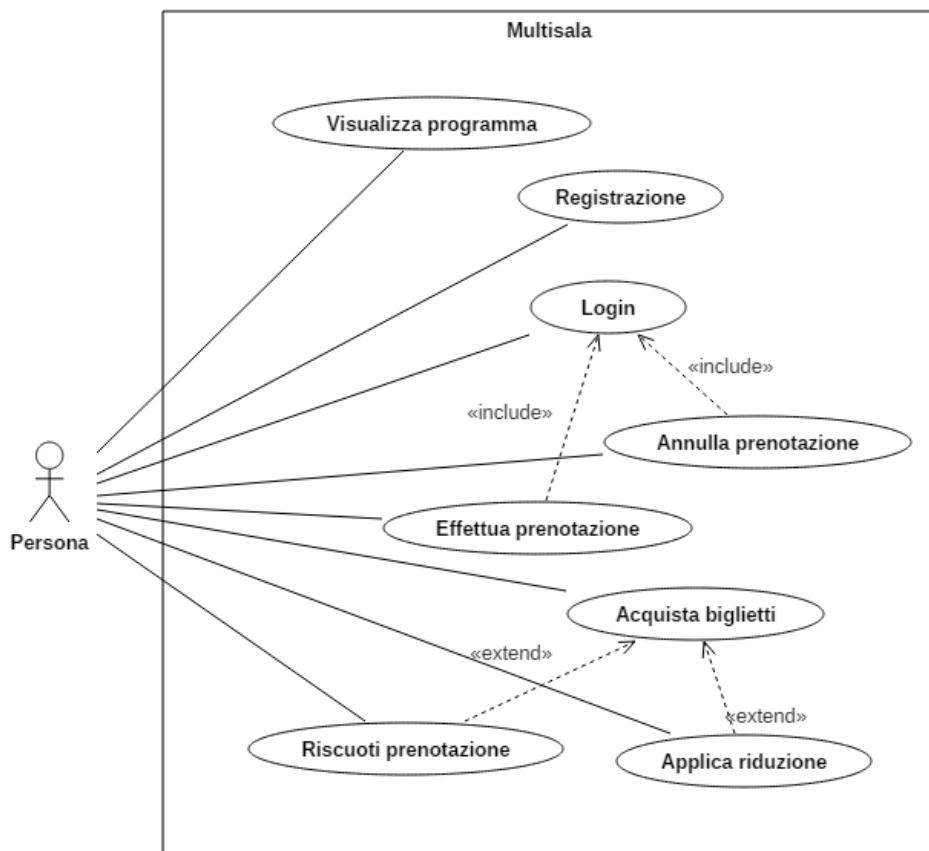
Si modellino le specifiche sopra riportate in UML attraverso un diagramma delle classi.

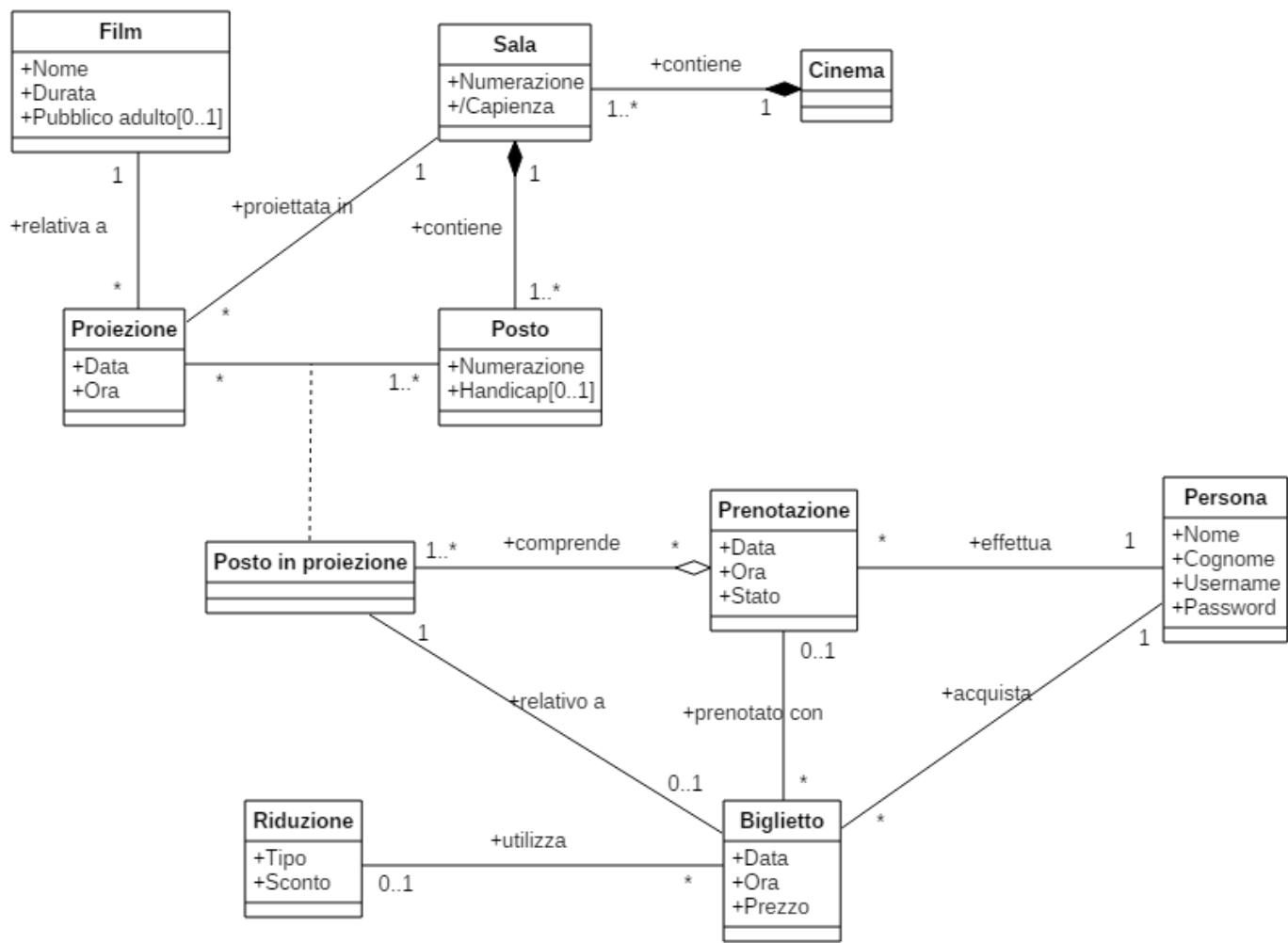


Si vuole rappresentare un sistema per la gestione dei biglietti in un cinema multisala. Il cinema è composto da diverse sale (di cui si memorizzano la numerazione e la capienza), ciascuna delle quali contiene diversi posti a sedere (per ciascuno dei quali si memorizzano la numerazione e l'eventuale esclusività per portatori di handicap). I film (contraddistinti da un nome, una durata ed un eventuale limitazione ad un pubblico adulto) possono essere proiettati in più giorni e in più sale contemporaneamente. Previa registrazione sul portale web del cinema (fornendo nome, cognome, username e password), i clienti possono effettuare prenotazioni (non vincolanti e annullabili in qualunque momento) per uno o più posti di una determinata proiezione; i biglietti possono essere poi ritirati presso casse automatiche inserendo il codice della prenotazione ed effettuando il pagamento. Si noti che un biglietto viene emesso per ogni posto. Se la prenotazione non viene riscossa entro 30 minuti dall'inizio della proiezione, i posti possono essere venduti ad altre persone tramite acquisto diretto presso le casse automatiche. Per le prenotazioni si vogliono memorizzare lo stato ("attiva", "annullata" o "scaduta"), la data e l'ora di prenotazione, mentre per i biglietti si vogliono memorizzare il prezzo, la data e l'ora di acquisto e l'eventuale prenotazione di riferimento. In fase di acquisto presso le casse è possibile selezionare una delle riduzioni disponibili, le quali danno diritto ad uno sconto sul prezzo base.

Si modellino le specifiche sopra riportate in UML attraverso:

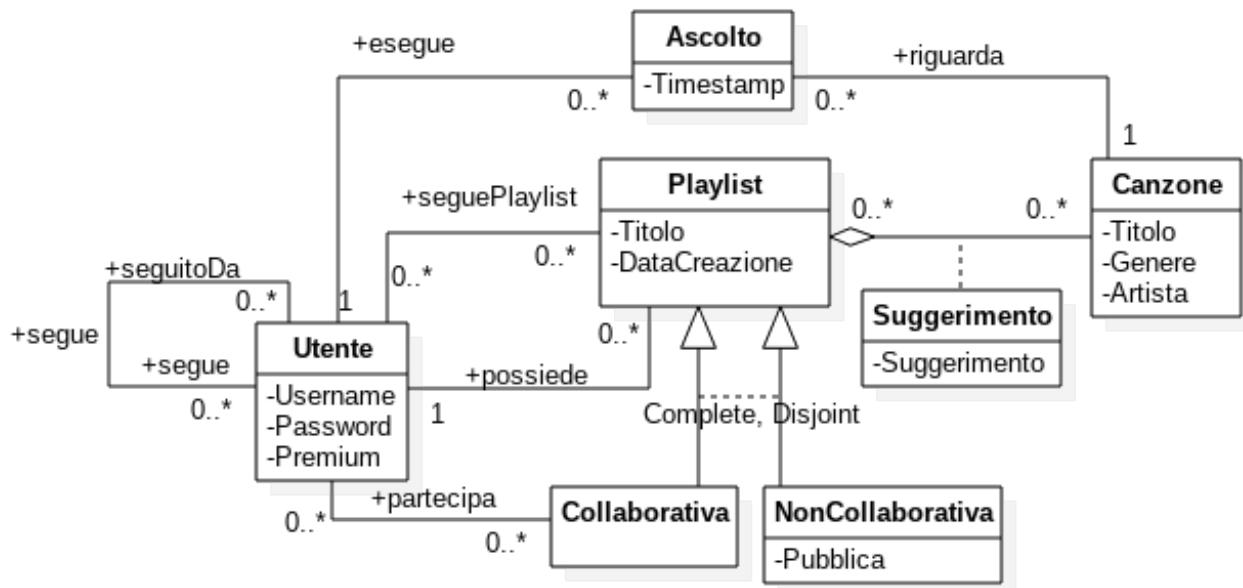
- 1) un *diagramma dei casi d'uso*
- 2) un *diagramma delle classi*





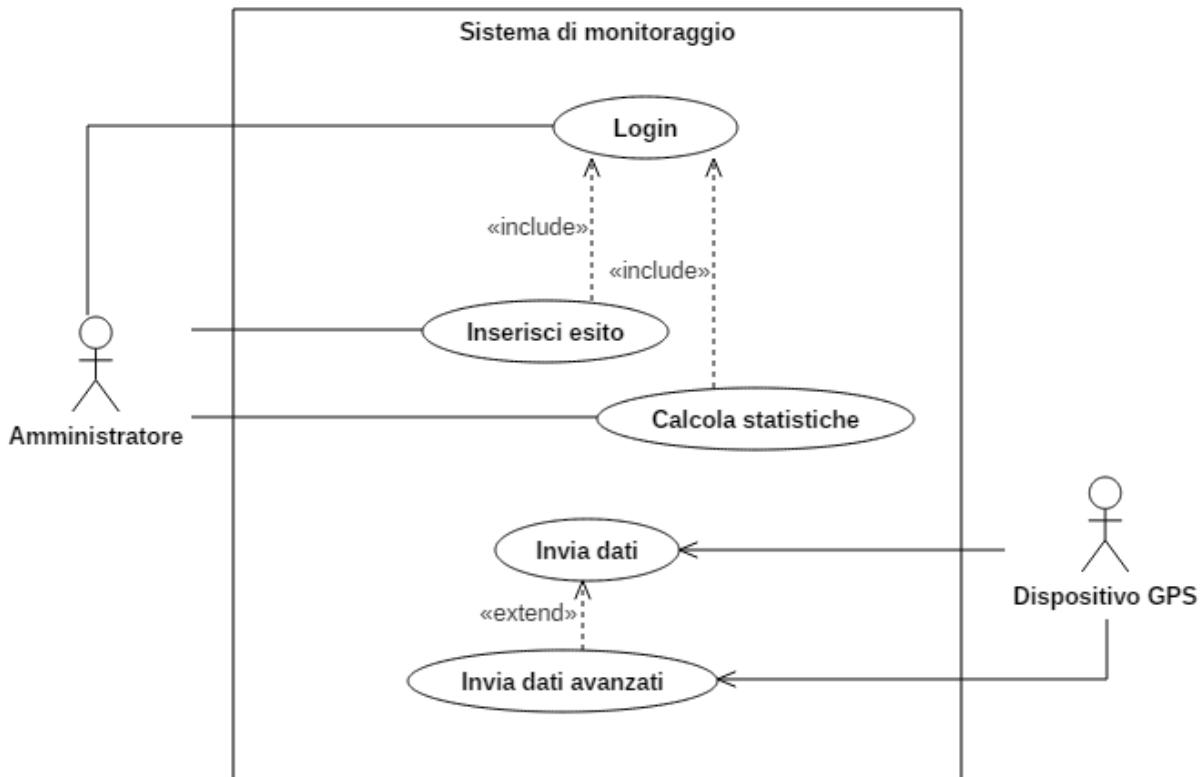
L'applicazione spoti5 consente ai suoi utenti di ascoltare musica online e di organizzare le canzoni preferite in playlist. Gli utenti (di cui si conoscono username, password) si dividono in premium e standard. Ogni utente può seguire/essere seguito da altri utenti e può possedere più playlist (di cui spoti5 registra titolo e data di creazione). Le playlist possono essere seguite da più utenti e hanno una duplice natura: collaborativa (a cui possono partecipare attivamente più utenti) e non collaborativa (pubblica o privata). Ad ogni playlist può essere aggiunto un numero arbitrario di canzoni (descritte dal titolo, dal genere e dall'artista/band). Si vuole tenere traccia del fatto che una canzone possa essere aggiunta alla playlist a seguito di un suggerimento da parte di spoti5. Infine, l'applicazione memorizza lo storico degli ascolti di ogni utente.

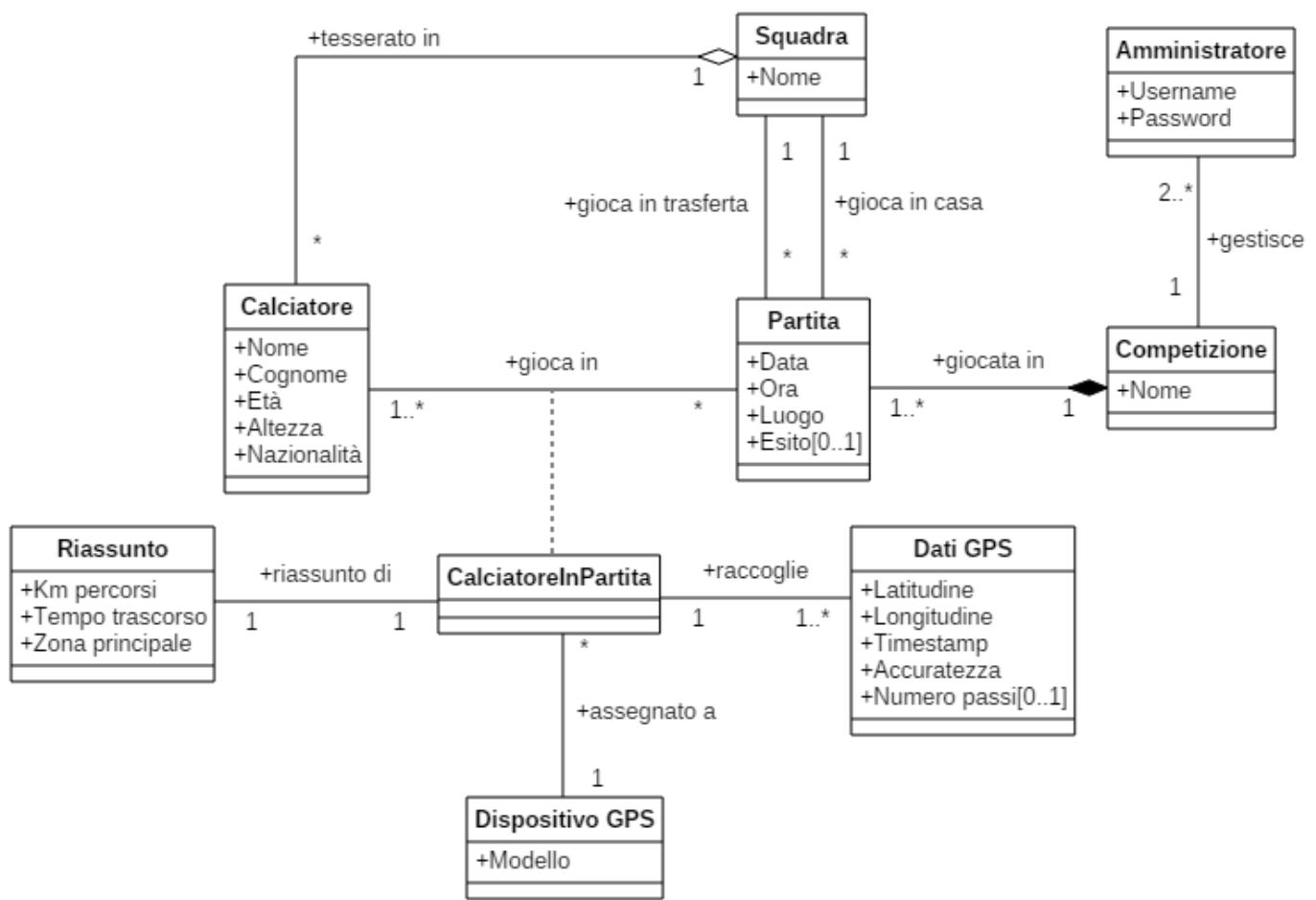
Si modellino le specifiche sopra riportate in UML attraverso un diagramma delle classi.



Si vuole modellare un sistema di monitoraggio dei calciatori per la Lega Calcio, utilizzato per raccogliere statistiche di movimento attraverso dispositivi GPS. Il sistema funziona per diverse competizioni (Serie A, Serie B, ecc.), ciascuna delle quali è composta da diverse partite; di ogni partita si conoscono la data e l'ora, il luogo, l'eventuale esito (inserito dagli amministratori del sistema al termine della partita) e le due squadre che si affrontano (una delle quali gioca "in casa", l'altra "in trasferta"). Di ogni calciatore si conoscono nome, cognome, età, altezza, nazionalità e squadra di appartenenza; per semplicità, si assuma che un calciatore non possa cambiare squadra. A ciascun calciatore viene assegnato un dispositivo GPS (che può cambiare di partita in partita e di cui si conosce il modello), il quale raccoglie informazioni sulla posizione in campo del giocatore: con cadenza di un secondo, il dispositivo invia al sistema la latitudine e la longitudine della posizione, insieme ad un timestamp e ad un indice di accuratezza del segnale GPS; i modelli più recenti inviano anche il conteggio dei passi rilevati. Al termine della partita, gli amministratori (previa login) lanciano delle elaborazioni statistiche sui dati raccolti per ottenere un dato riassuntivo per ciascun giocatore, il quale contiene il numero di chilometri percorsi, il tempo trascorso correndo e la zona del campo principalmente coperta. Le credenziali degli amministratori sono memorizzate nel sistema; un amministratore copre le partite di una sola competizione, ma una competizione è gestita da almeno due amministratori.

Si modellino le specifiche sopra riportate in UML attraverso un diagramma delle classi e un diagramma dei casi d'uso.

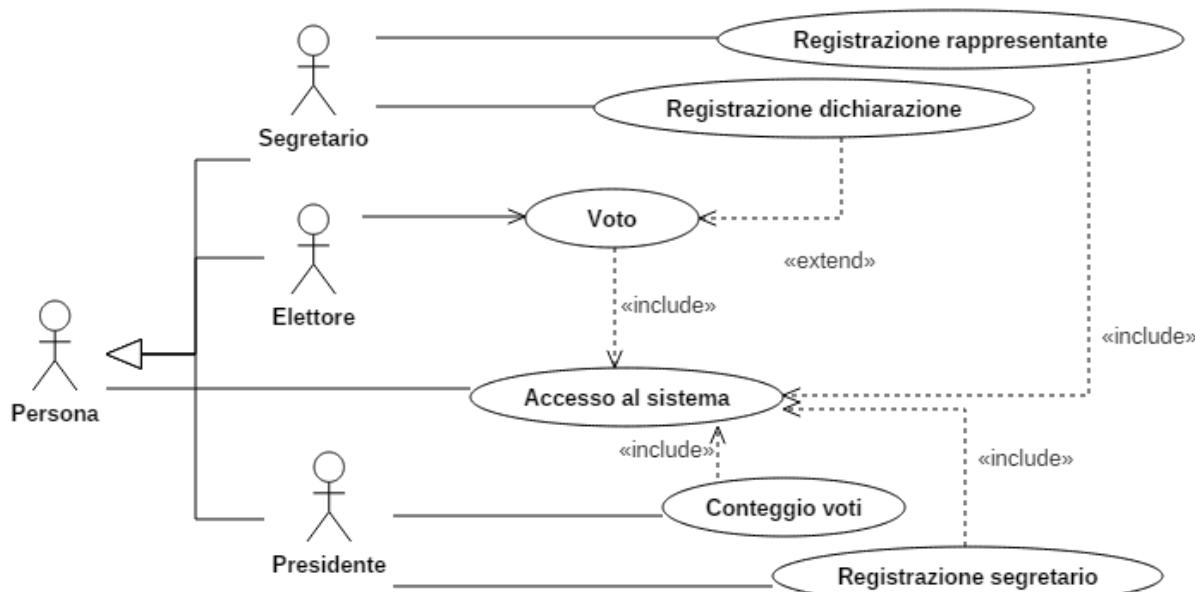


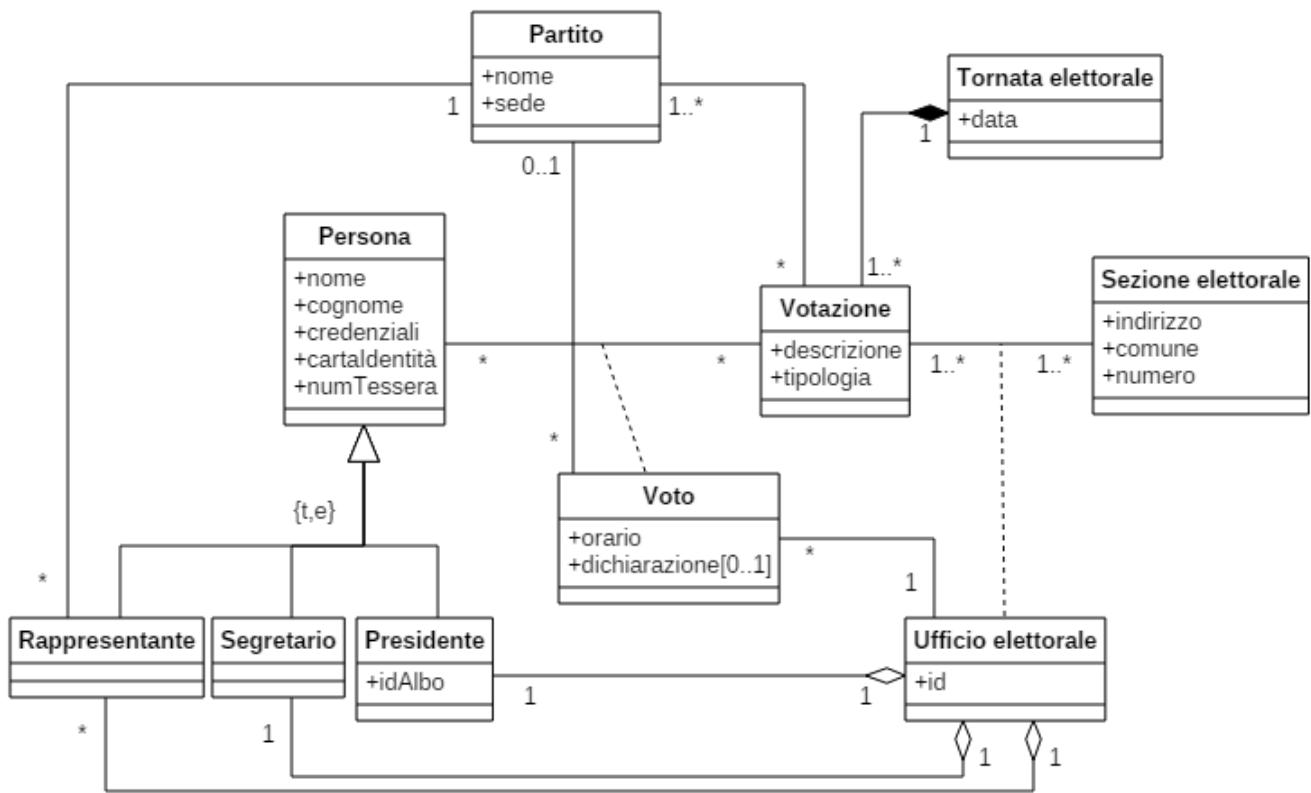


Si vuole rappresentare un sistema per la gestione elettronica delle votazioni. Le votazioni sono contraddistinte da una descrizione (es: votazione per la Camera dei Deputati, oppure per il Senato) e da una tipologia (es: politiche, amministrative) e sono raggruppate in tornate, identificate da una data. A una votazione si presentano più partiti, contraddistinti da un nome e da una sede. Esistono diverse sezioni elettorali, di cui si memorizzano l'indirizzo, il comune ed il numero; in occasione di una votazione, un ufficio elettorale si insedia presso ciascuna sezione. L'ufficio (a cui è assegnato un ID univoco) è composto da un presidente (di cui si memorizza l'ID di registrazione presso l'Albo dei Presidenti), scelto automaticamente dal sistema, e da un segretario, scelto e registrato dal presidente. Eventualmente possono prendervi parte anche dei rappresentanti di lista, i quali fanno riferimento ad un determinato partito; la registrazione dei rappresentanti di lista all'ufficio deve essere gestita dal segretario. Si consideri che una persona possa ricoprire uno solo dei ruoli sopra menzionati. Per ciascuna votazione, gli elettori (di cui si conoscono nome, cognome, credenziali, carta d'identità e numero della tessera elettorale) accedono al sistema ed esprimono un unico voto, rivolto ad uno dei partiti che si sono presentati (oppure rivolto a nessuno); i voti registrati devono tenere traccia anche dell'orario, dell'ufficio elettorale presso cui il voto è stato espresso e di eventuali dichiarazioni lasciate dall'elettore (le quali vengono registrate dal segretario). Alla fine della votazione, il conteggio dei voti dell'ufficio viene azionato e validato dal relativo presidente.

Si modellino le specifiche sopra riportate in UML attraverso:

- 1) un diagramma dei casi d'uso
- 1) un diagramma delle classi

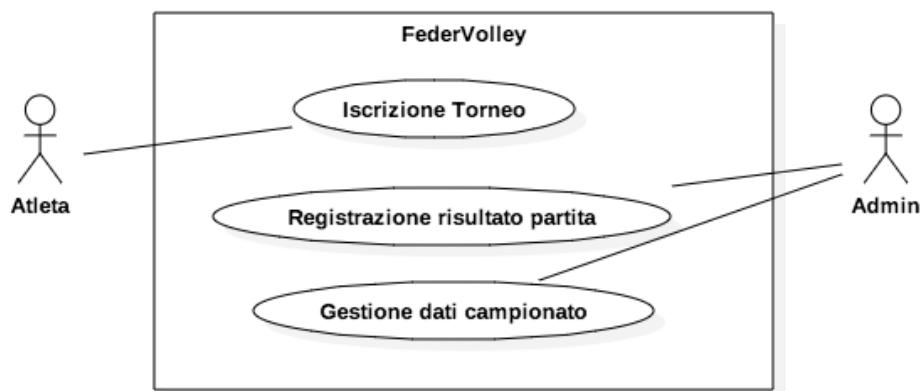
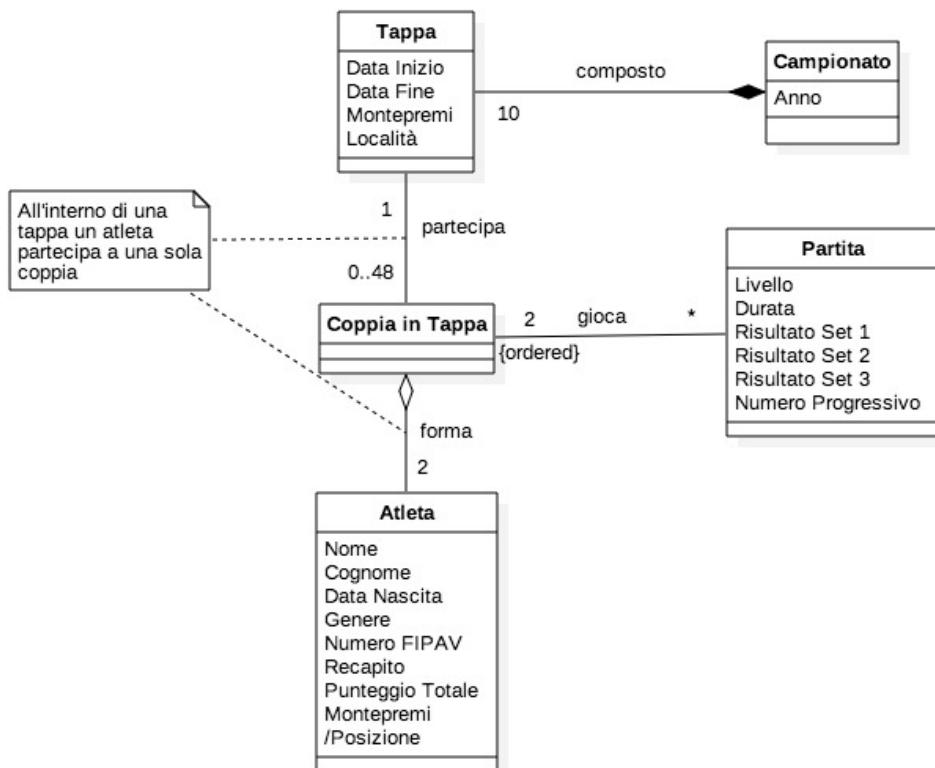




La FererVolley vuole realizzare il sistema informativo per la gestione dei campionati italiani assoluti di beach volley maschile che si svolgono con frequenza annuale.

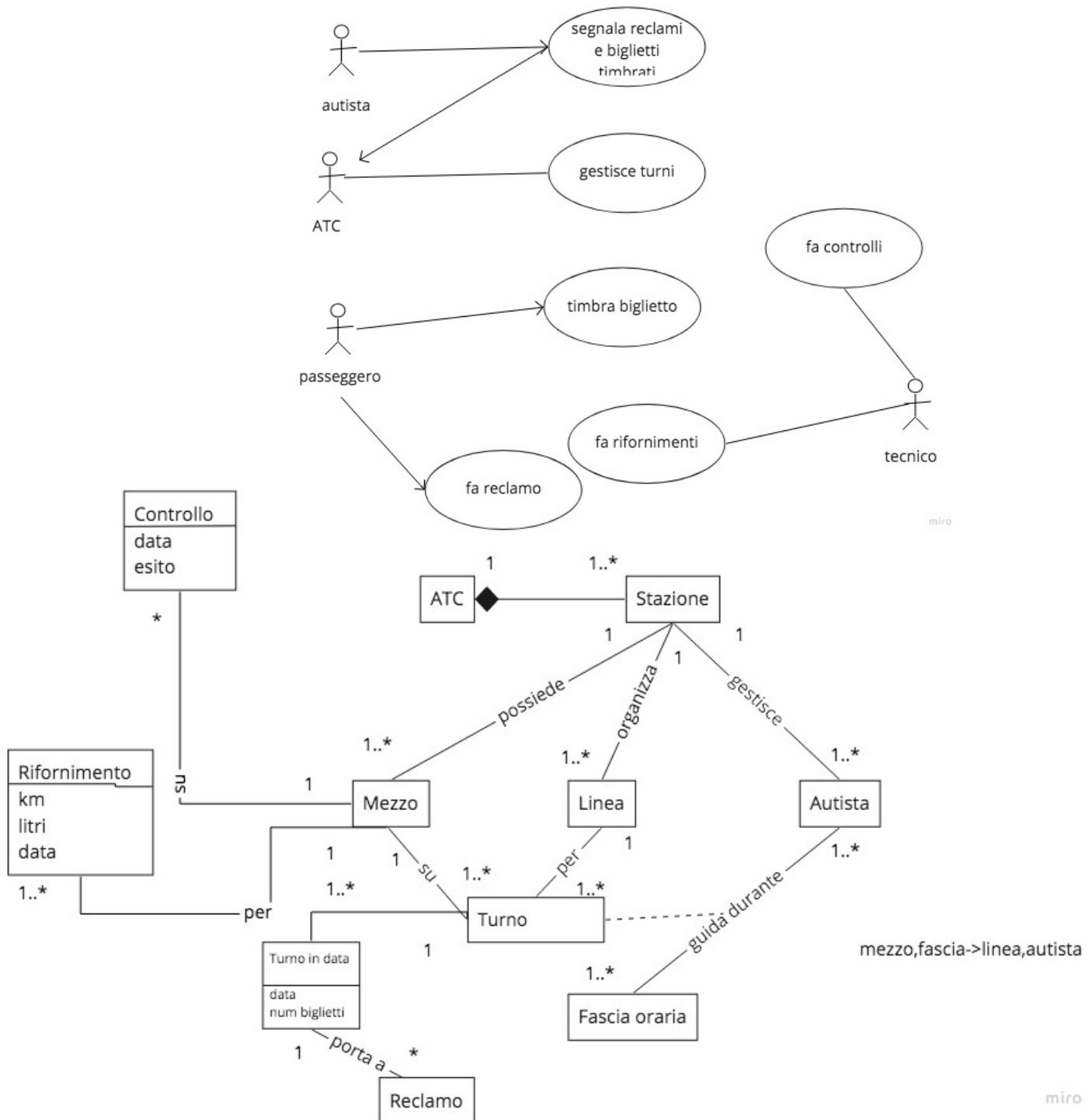
Il campionato è sempre composto da 10 tappe (ciascuna corrispondente ad un singolo torneo) che si svolgono in diverse località italiane. Ciascuna tappa è caratterizzata da una data d'inizio, data di fine, nome della località, e montepremi. A ciascuna tappa si possono iscrivere fino a 48 coppie di giocatori che entrano a far parte del torneo. Di ciascun giocatore sono memorizzati i dati anagrafici, il numero di tesserino FIPAV e il recapito; è inoltre interessante tener traccia del punteggio complessivamente acquisito nelle tappe, il montepremi vinto e la posizione in classifica. Ciascun giocatore può cambiare compagno nei diversi tornei, ma non durante le partite di uno stesso torneo. Di tutte le partite di ciascun torneo viene registrato un numero progressivo, il livello (quarti, semifinale, ...), la durata, le coppie sfidanti e il risultato dei 3 set (es. 21-14, 19-21, 15-13). A seguito di ogni torneo ovviamente la posizione di ciascun giocatore può variare in base al punteggio e al montepremi acquisito.

Utilizzando il formalismo UML, si disegnino il diagramma delle classi e dei casi d'uso per il dominio applicativo descritto.



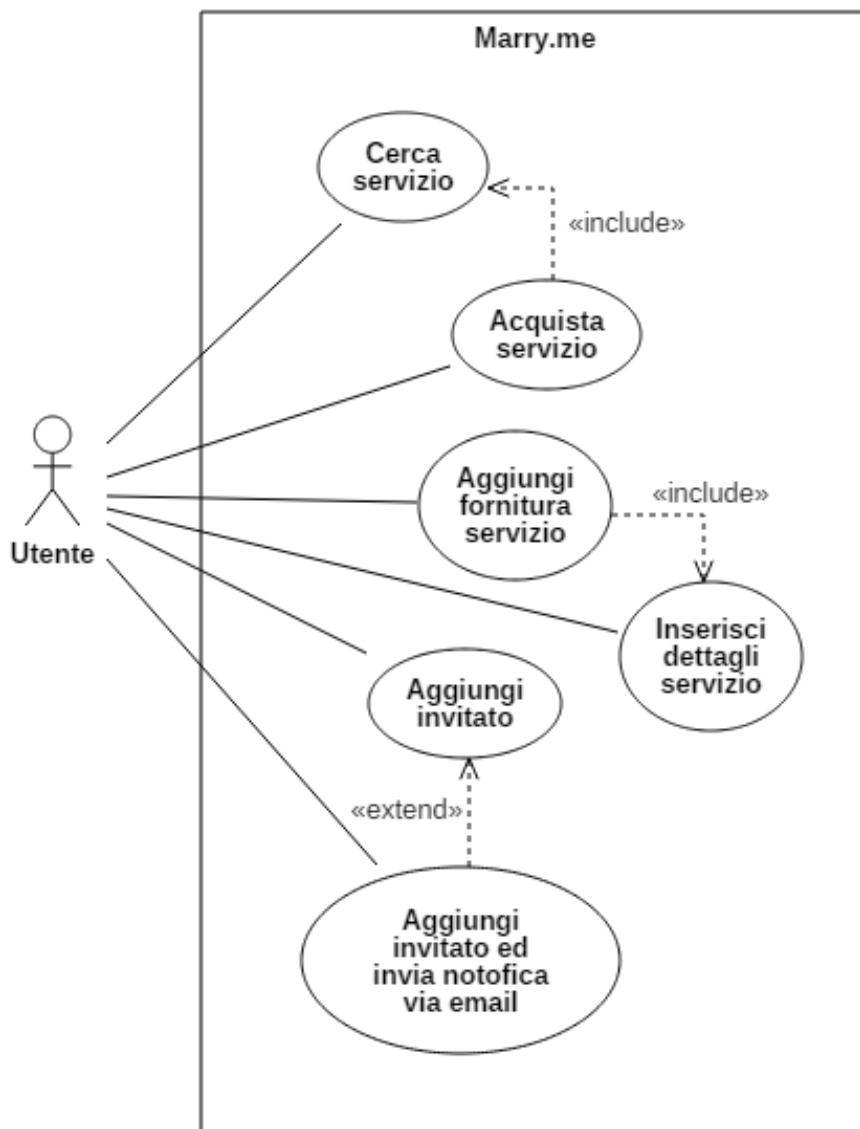
Disegnare il diagramma delle classi e il diagramma dei casi d'uso per il dominio che segue.

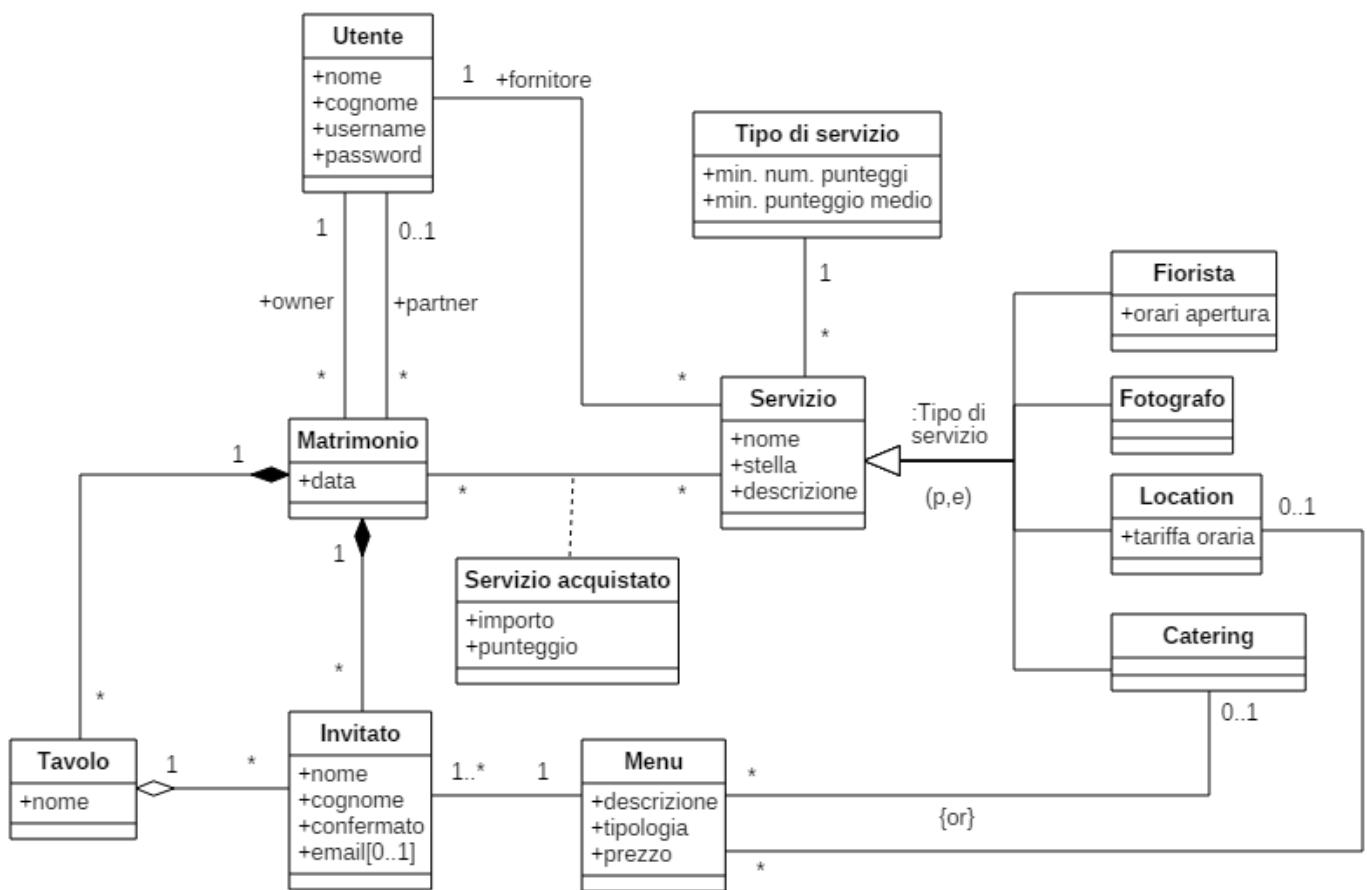
Un'azienda di trasporti comunale intende automatizzare la gestione dei propri mezzi, delle linee urbane e del personale autista. L'azienda gestisce più stazioni all'interno del proprio territorio: ogni stazione è base per un certo numero di mezzi, ha un certo numero di autisti e un certo numero di linee urbane da organizzare. Lo scheduling dei mezzi e del personale avviene nel seguente modo: nell'arco della giornata esistono tre fasce orarie, data una fascia oraria e una linea urbana esistono più mezzi in azione sulla stessa con i rispettivi conducenti. Alla fine del proprio turno, l'autista deve segnalare eventuali reclami (memorizzati poi in un campo di testo) e il numero di biglietti timbrati. Ultima specifica da modellare riguarda la gestione del controllo mezzi (ogni mezzo viene controllato periodicamente) e dei rifornimenti (mantenere il rilevamento chilometrico, i litri di carburante e la data di rifornimento)



Il sito marry.me consente agli utenti registrati di programmare matrimoni. Ogni utente può creare uno o più eventi (contraddistinti da una data) ed eventualmente associarli all'utente del partner. I due utenti degli sposi possono quindi creare la propria lista di invitati, indicando l'eventuale conferma e con la possibilità di indicare un'email per inviare un promemoria all'invitato. Gli invitati possono essere organizzati in tavoli (indicando il nome di questi ultimi) e possono essere associati ad un menù, il quale ha una descrizione, una tipologia (es: "normale", "celiaco", "artista") ed un prezzo per persona. Il sito mette inoltre a disposizione dei servizi (associati ad un utente fornitore) che gli sposi possono cercare ed acquistare; per ogni servizio acquistato si memorizzano l'importo speso ed il punteggio (da 1 a 5) eventualmente assegnato a matrimonio concluso. I servizi hanno un nome, una descrizione ed un'eventuale "stella" che evidenzia i servizi più meritevoli; la stella viene assegnata in base a due criteri (un numero minimo di punteggi assegnati ed un minimo punteggio medio) che variano in base alla tipologia di servizio. Tra le tipologie disponibili sono presenti i fioristi (di cui si memorizza l'orario di apertura), i fotografi, le location (di cui si memorizza la tariffa oraria) ed il catering. I menù possono essere associati ad un catering o ad una location.

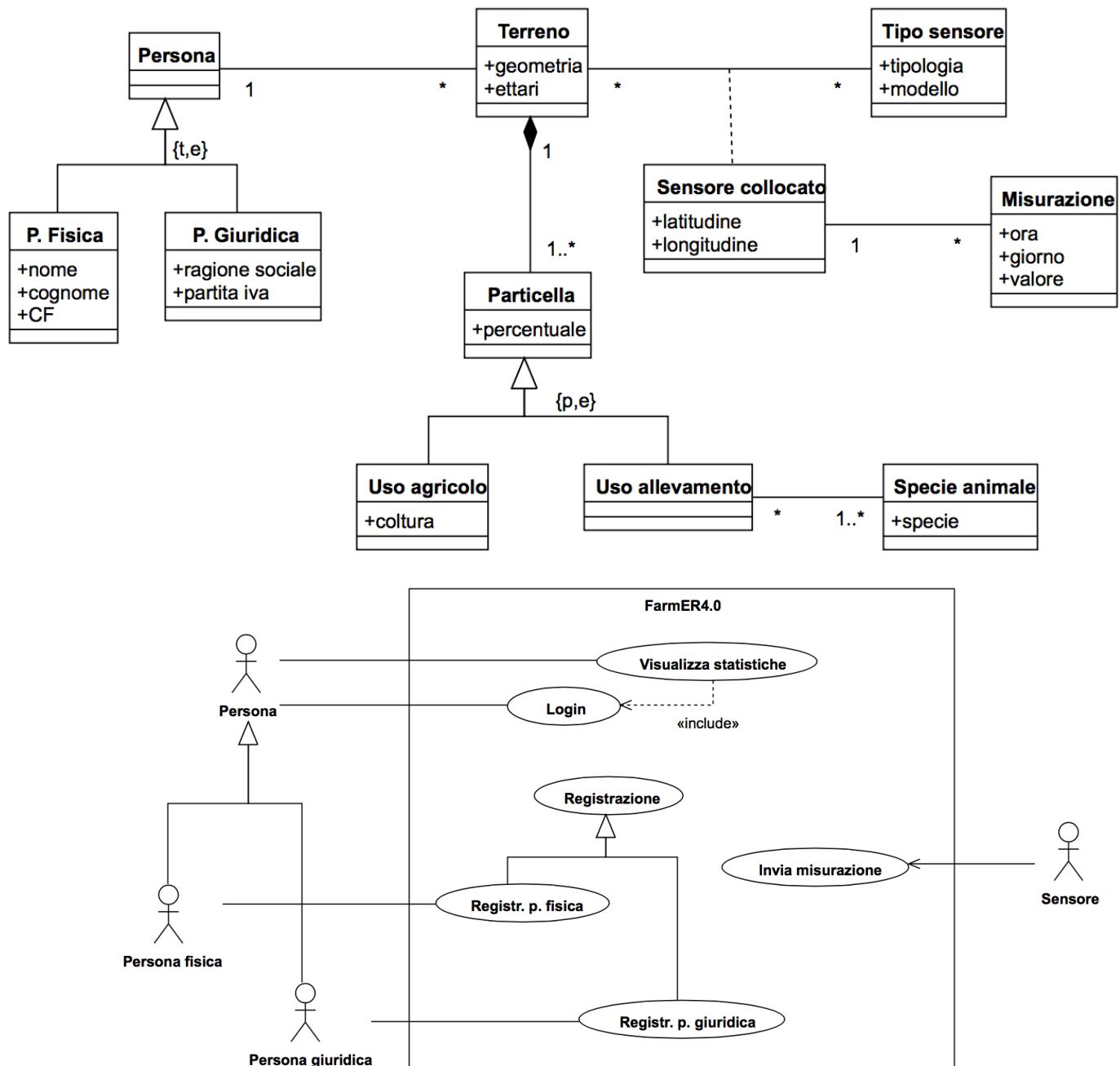
Utilizzando il formalismo UML, si disegnino il diagramma delle classi e dei casi d'uso per il dominio applicativo descritto.





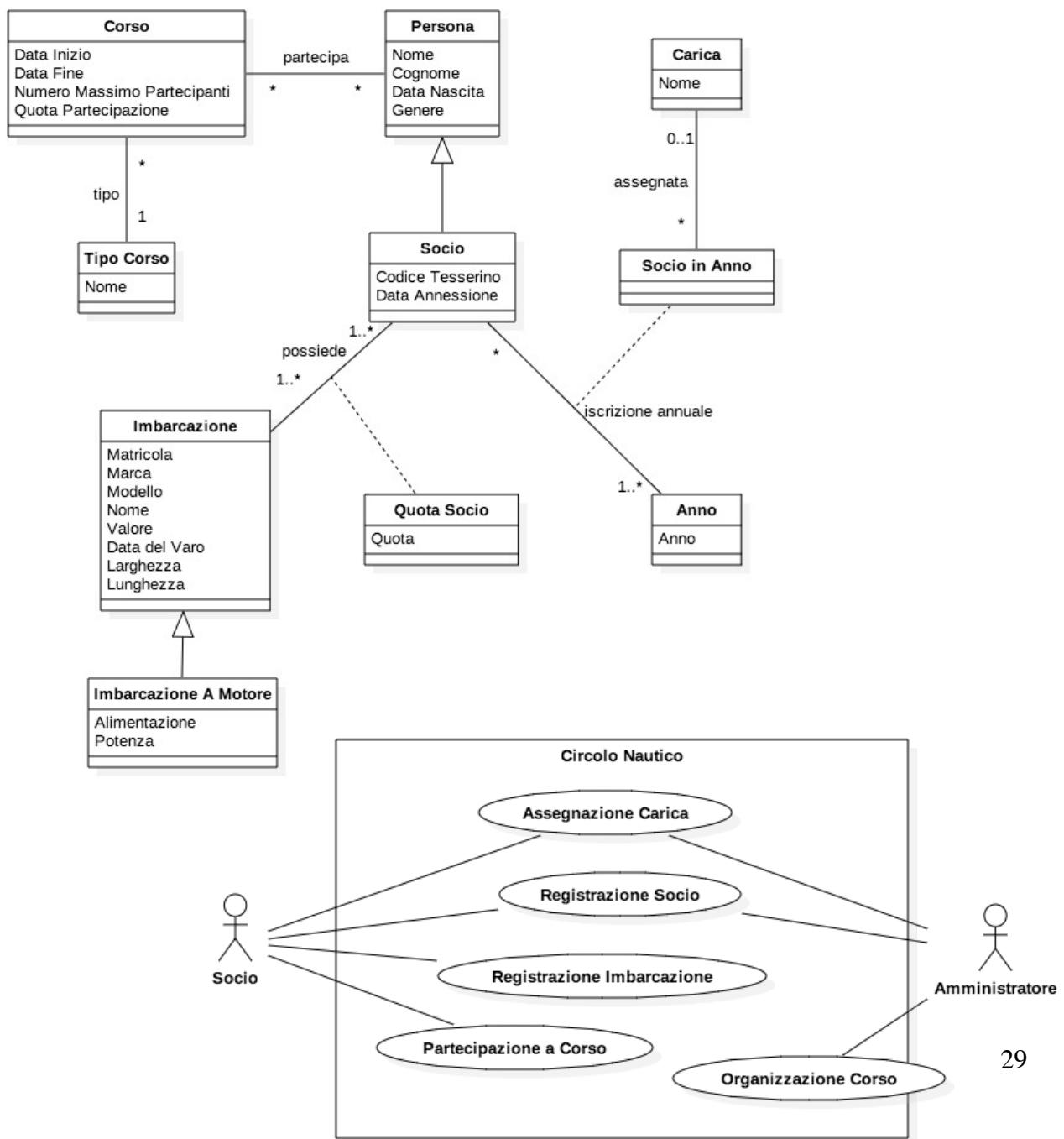
Il sistema FarmER4.0 gestisce le anagrafiche relative ai terreni utilizzati a scopo agricolo o di allevamento a livello regionale, insieme a dati statistici raccolti in tempo reale attraverso sensori in loco. Al sistema possono registrarsi sia persone fisiche (inviando nome, cognome e codice fiscale) che giuridiche (inviando ragione sociale e partita iva); ciascuna persona può possedere più terreni, di cui si memorizzano la geometria georeferenziata e l'estensione in ettari. Ogni terreno è composto da particelle, le quali coprono una determinata percentuale del terreno. Ogni particella è destinata ad un uso specifico, che può essere agricolo (nel qual caso si memorizza il tipo coltura), di allevamento (nel qual caso si memorizzano le specie animali presenti), o altro. Persone fisiche e giuridiche possono inoltre dotarsi di uno o più sensori (di cui esistono diverse tipologie e modelli), da collocare in un determinato terreno (un terreno non può essere dotato di più sensori dello stesso tipo). Per ogni sensore collocato, si vogliono conoscere le coordinate esatte e le misurazioni rilevate ogni giorno con cadenza oraria.

Utilizzando il formalismo UML, si disegnino il diagramma delle classi e dei casi d'uso per il dominio applicativo descritto.



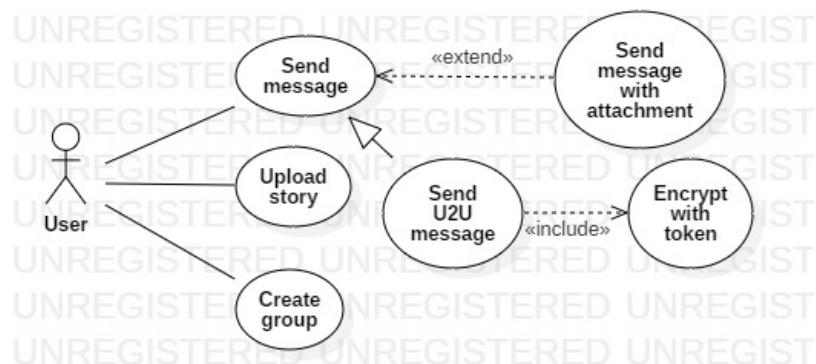
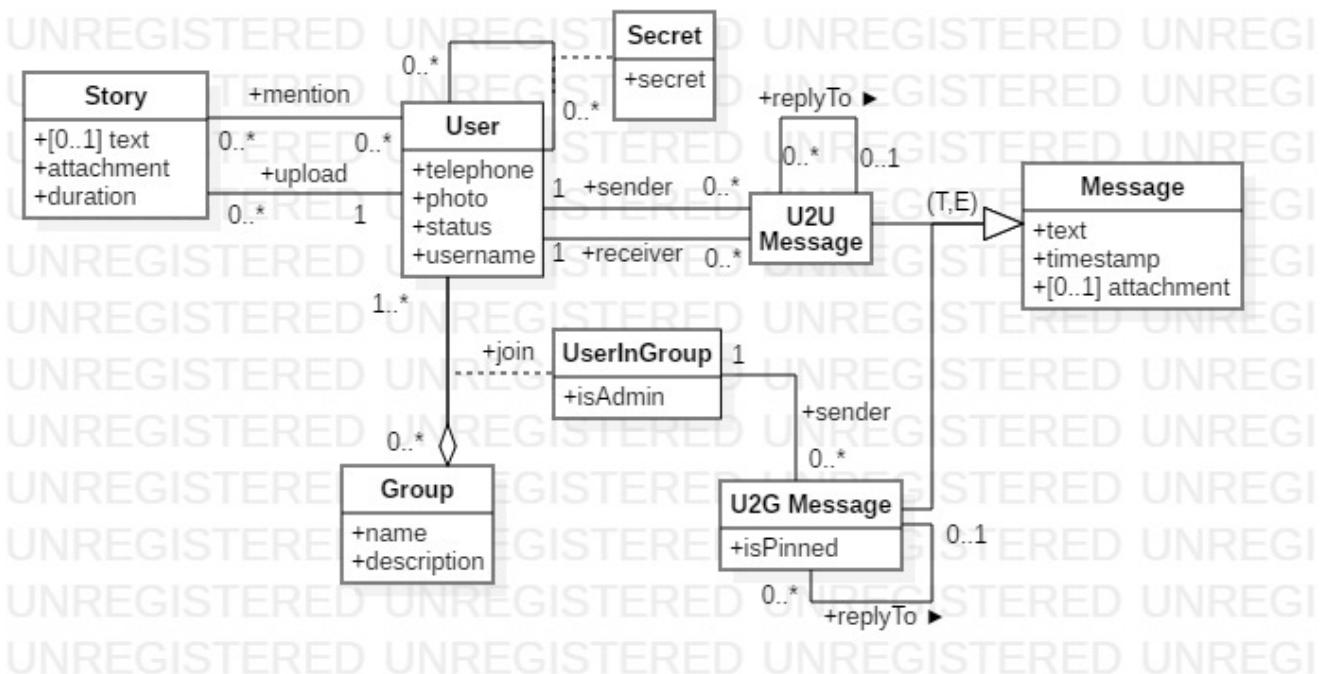
Un circolo nautico vuole realizzare un software per la gestione delle anagrafiche dei soci, dei corsi e delle imbarcazioni. Tutti i soci del circolo hanno un tesserino in cui è registrato il codice identificativo, i dati anagrafici (nome, cognome e data di nascita) e la data di annessione al circolo. Ogni anno vengono rinnovate le cariche sociali (presidente, segretario, consigliere, ...) che devono essere registrate (socio in carica, anno). Alcune cariche (es. consigliere) possono essere rivestite da più soci nello stesso anno ma un singolo socio però non può rivestire più cariche contemporaneamente. Ogni socio può possedere una o più imbarcazioni (del resto ciascuna imbarcazione può avere più proprietari, tutti soci, per ciascuno dei quali viene registrata una quota). I dati registrati per ciascuna imbarcazione sono: matricola, marca e modello, nome, valore, data del varo, dimensioni (larghezza e lunghezza), alimentazione e potenza del motore se è una barca a motore. Il circolo organizza inoltre dei corsi aperti a tutti (non solo soci) per i quali viene registrata la data di inizio e fine, il numero massimo di partecipanti, il tipo (es. "corso di vela base"), la quota di partecipazione e dati anagrafici degli iscritti.

Utilizzando il formalismo UML, si disegnino il diagramma delle classi e dei casi d'uso per il dominio applicativo descritto.



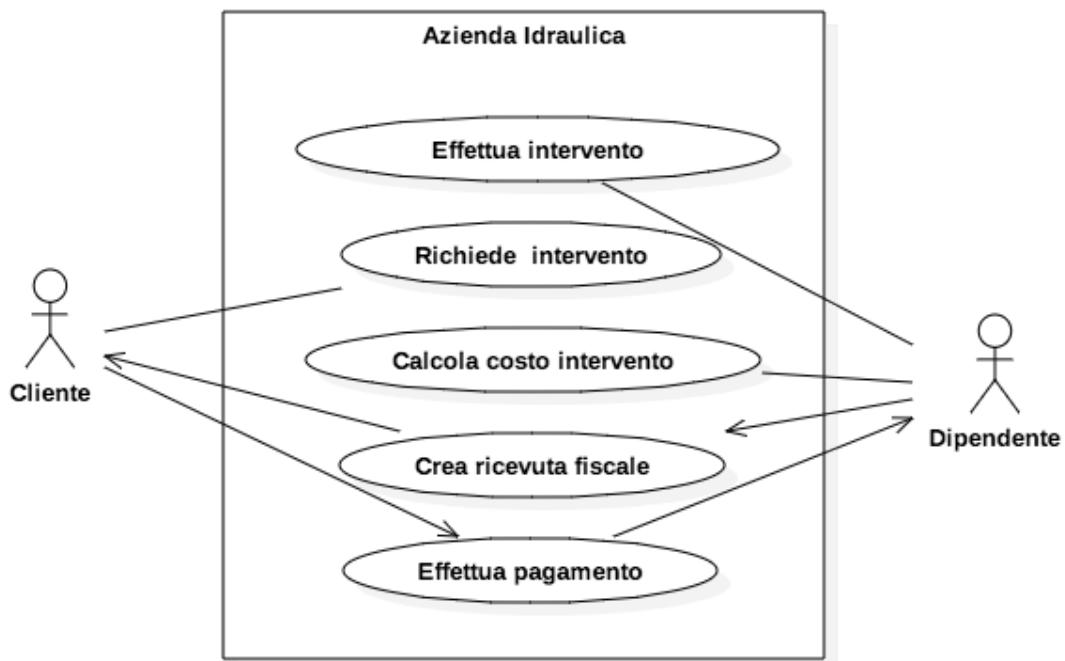
ChittyChat è un'applicazione che consente agli utenti di scambiare messaggi e pubblicare storie online. I messaggi possono essere scambiati tra singoli utenti o all'interno di gruppi di utenti. Di ogni utente si registrano l'username, il numero di telefono, la foto profilo e la descrizione. Di ogni gruppo si registrano il nome e la descrizione. Un utente può creare e partecipare ad un numero arbitrario di gruppi di cui può essere amministratore. Quando un utente crea un gruppo ne diventa amministratore. L'amministratore del gruppo può aggiungere a sua volta altri amministratori tra gli utenti del gruppo. Ogni utente può inviare messaggi a un singolo utente o all'interno di un gruppo. Dei messaggi inviati si conosce il testo, il timestamp (quando il messaggio è stato inviato), e un allegato multimediale opzionale. Un messaggio può essere scritto in risposta ad un messaggio precedente. I messaggi inviati all'interno di gruppi possono essere annotati come importanti. ChittyChat assegna ad ogni coppia di utenti singoli che scambia messaggi un token segreto utilizzato per criptare i messaggi scambiati (ciò non accade per i messaggi inviati all'interno di gruppi). Infine, ogni utente può eseguire l'upload di storie di cui si registrano l'allegato multimediale, la durata, e opzionalmente il testo e le menzioni di altri utenti.

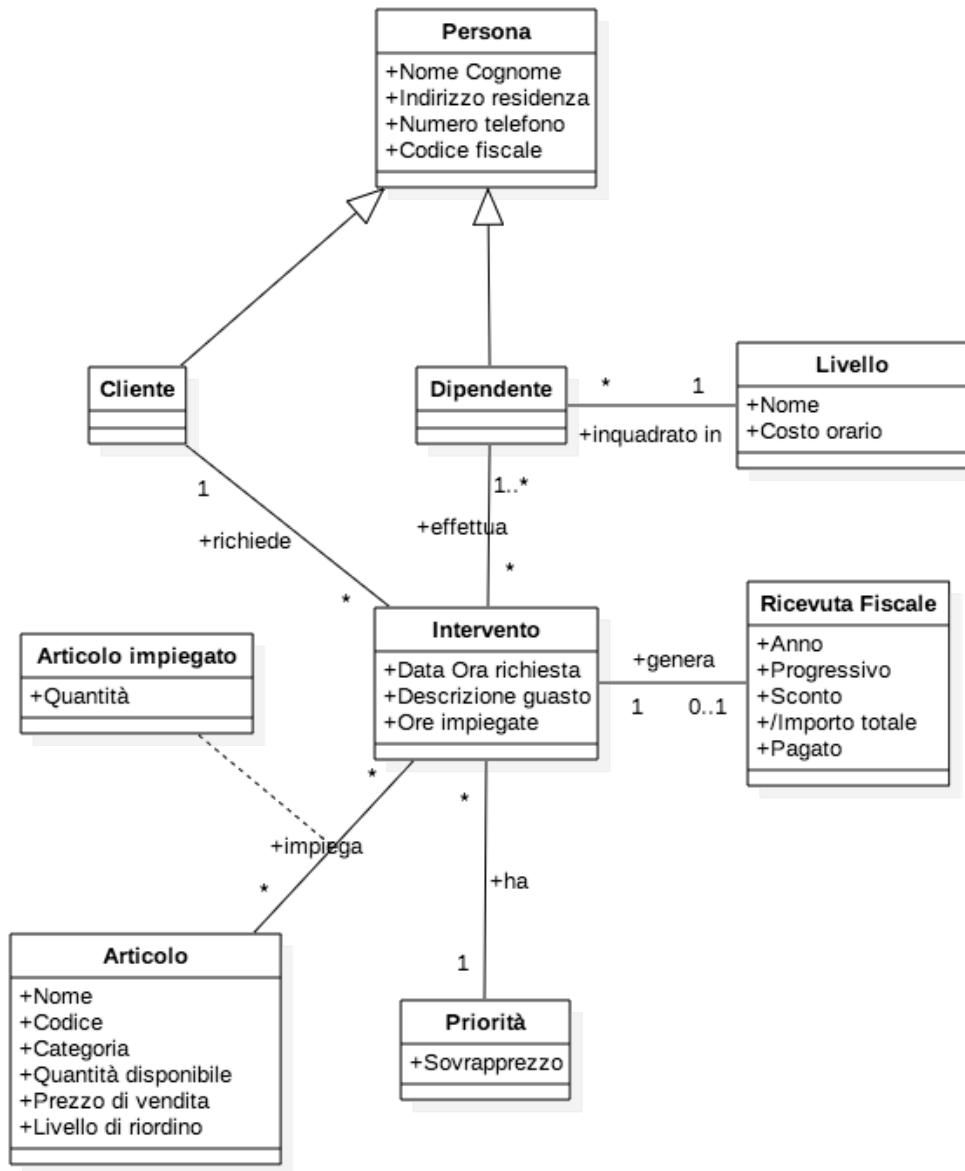
Utilizzando il formalismo UML, si disegnino il diagramma delle classi e dei casi d'uso per il dominio applicativo descritto.



Si vuole realizzare un software per la gestione degli interventi effettuati da un'azienda idraulica. Il sistema dovrà gestire il magazzino, i dipendenti, le prenotazioni degli interventi richiesti dai clienti, il tariffario, l'emissione di ricevute fiscali e i pagamenti. I clienti, dei quali si memorizzano nominativo, indirizzo, numero di telefono e codice fiscale, richiedono telefonicamente un intervento. Al momento della prenotazione, il cliente deve specificare la descrizione dell'intervento richiesto. Per ogni prenotazione si memorizzano inoltre la data e l'ora della richiesta, il cliente e la priorità (urgenza dell'intervento). Ogni intervento può essere effettuato da uno o più dipendenti e può prevedere l'impiego di materiale presente in magazzino. Per ciascun dipendente si registrano i consueti dati anagrafici e il livello (es. apprendista, operaio specializzato, ecc.). A ciascun livello corrisponde un costo orario di manodopera. Il costo totale di ogni intervento è calcolato in base ai materiali utilizzati, al costo di manodopera dei dipendenti intervenuti e al numero di ore impiegate. All'intervento può essere aggiunto un sovrapprezzo determinato dal livello di priorità. Il sistema deve gestire il materiale presente in magazzino, memorizzando per ciascun articolo un codice identificativo, il nome, la categoria, la quantità disponibile, il prezzo di vendita e il livello di riordino. A ogni intervento è associata l'emissione di una ricevuta fiscale contraddistinta da un numero progressivo e dall'anno di emissione. Al momento dell'emissione della ricevuta, l'azienda può decidere di applicare una percentuale di sconto sull'importo totale. Per ogni ricevuta emessa bisogna tenere traccia dell'eventuale pagamento.

Utilizzando il formalismo UML, si disegnino il diagramma delle classi e dei casi d'uso per il dominio applicativo descritto.

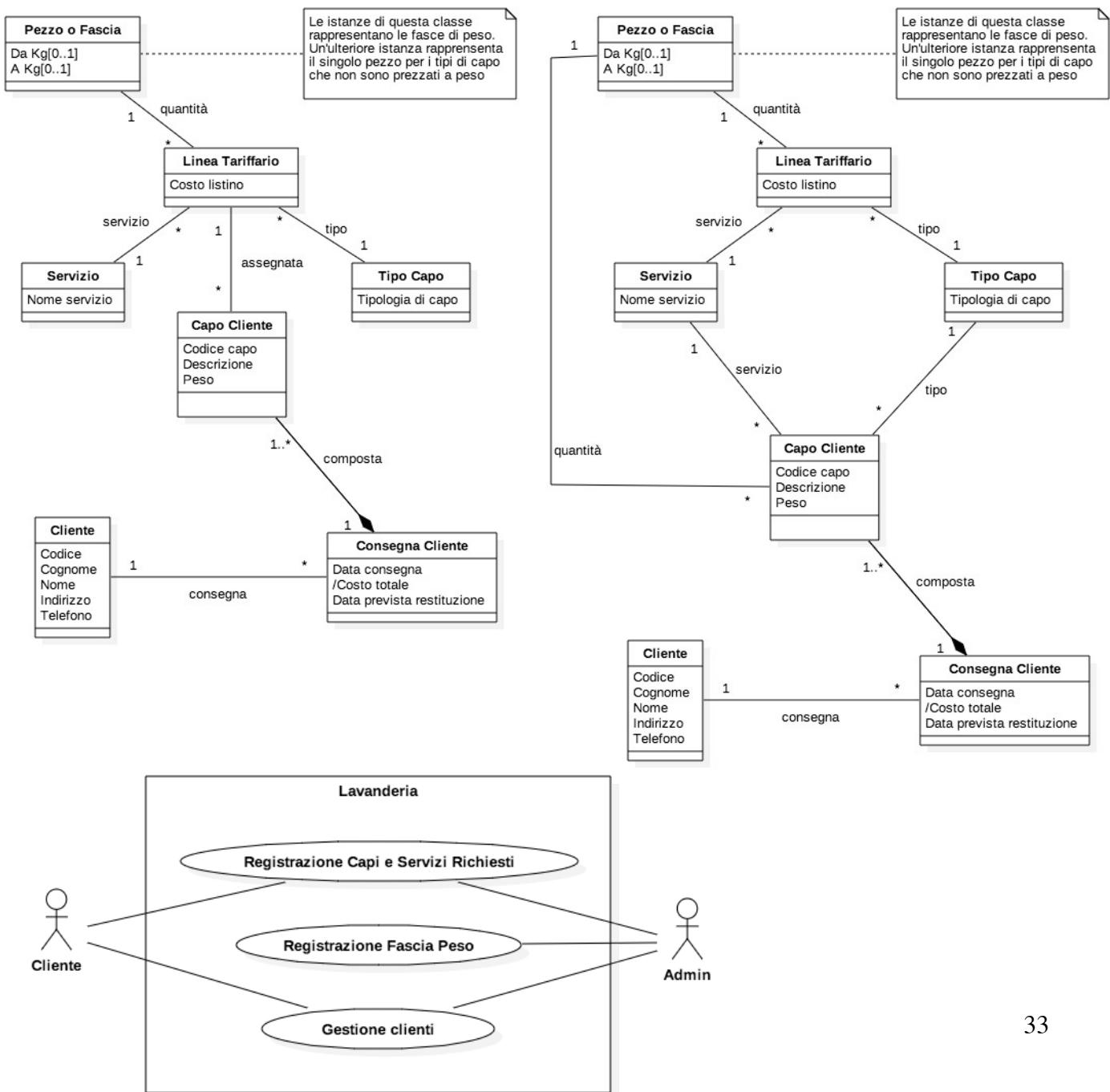




La lavanderia Arcobaleno vuole realizzare un sistema software per la gestione dei clienti, dei capi da lavare e del tariffario dei servizi offerti.

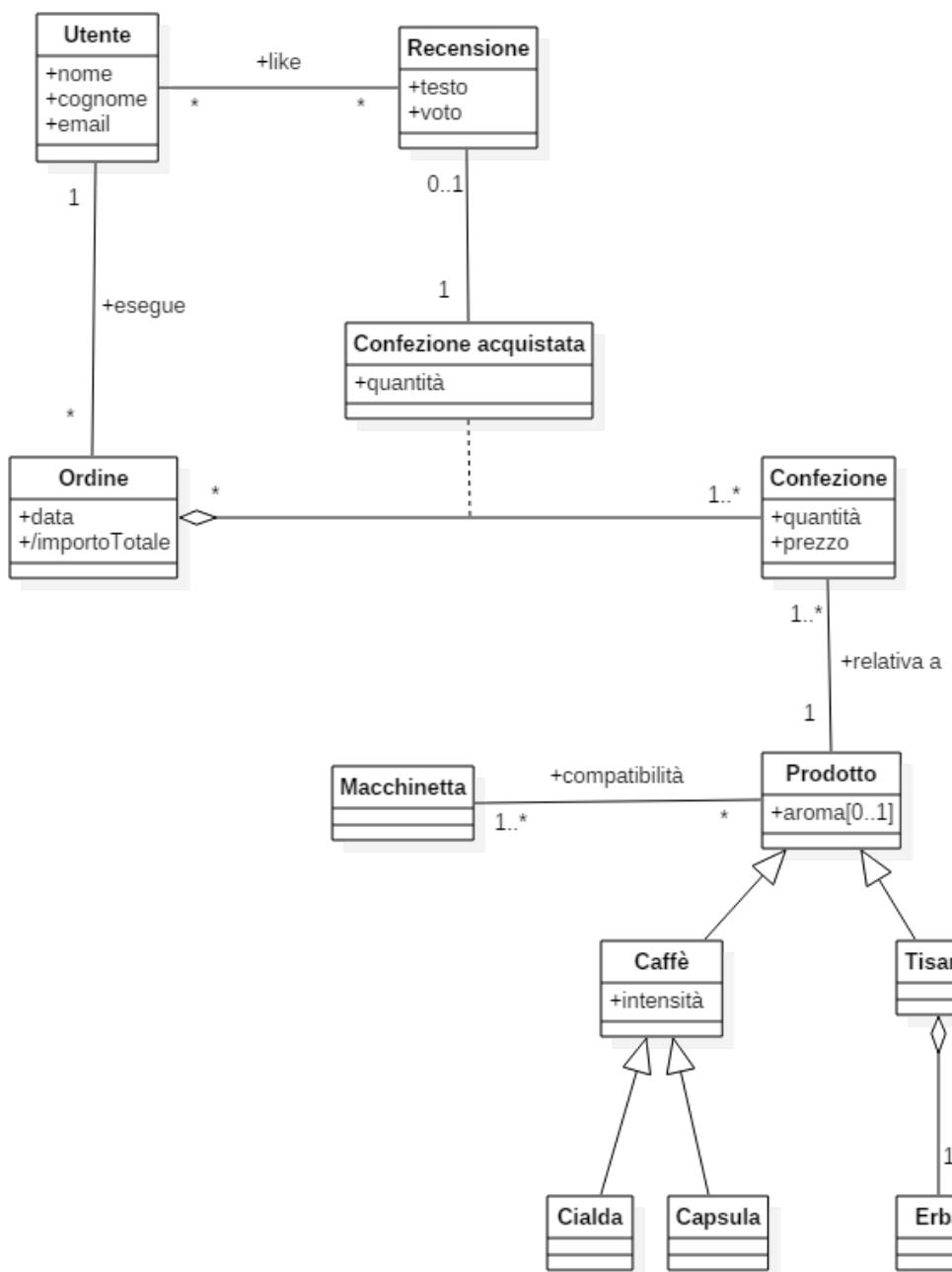
La lavanderia offre i servizi di lavaggio, in acqua o a secco, e stiratura. Il prezzo dei servizi offerti dipende dalla tipologia di capo da trattare (es. pantalone, gonna, giacca, ...). Per alcune tipologie di capi (es. tappeti, tende) il prezzo dei servizi dipende anche dalla fascia di peso. Le fasce, descritte da un peso minimo e un peso massimo (es. da 3 a 5 kg), devono essere registrate nel sistema. I clienti della lavanderia sono memorizzati in un archivio; per ciascuno si registrano un codice identificativo, il cognome, il nome, l'indirizzo e un recapito telefonico. Ogni volta che un cliente porta dei capi in lavanderia, nel sistema devono essere memorizzati la data, il costo totale e la data di restituzione prevista. Inoltre per ogni capo da trattare, identificato da un codice univoco, si memorizzano una descrizione, l'eventuale peso e i servizi richiesti. Naturalmente per uno stesso capo possono essere richiesti più servizi (es. lavaggio a secco e stiratura).

Utilizzando il formalismo UML, si disegnino il diagramma delle classi e dei casi d'uso per il dominio applicativo descritto.



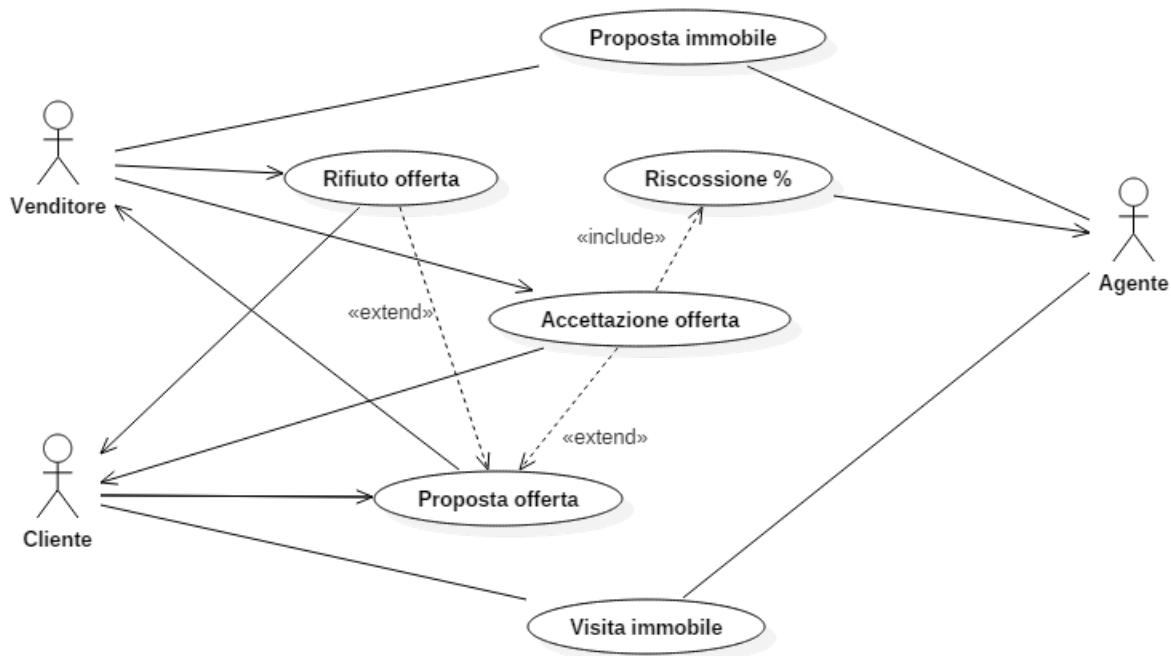
Il sito kaffe.it permette agli utenti registrati di ordinare alcuni tipi di prodotto (caffè o tisane) per macchine da caffè; il caffè (la cui intensità può essere bassa, media o alta) può essere fornito in capsule o in cialde, mentre le tisane (contenenti una serie di erbe) sono disponibili solo in cialde. Ogni prodotto è compatibile con diversi modelli di macchinette e può essere contraddistinto da un aroma particolare (per esempio, vaniglia o cocco). Il sito non mette in vendita i singoli esemplari dei prodotti, bensì delle confezioni che contengono una determinata quantità di un prodotto ad un determinato prezzo. Gli utenti (di cui si conoscono le generalità e le credenziali di login) possono effettuare ordini in cui includere una o più unità di almeno una confezione; per ogni ordine vengono memorizzati la data e l'importo totale delle confezioni acquistate. Il sito offre inoltre agli utenti la possibilità di fornire delle recensioni sulle confezioni acquistate, comprensive di un testo libero e di una valutazione numerica da 1 (scarsa) a 5 (ottima). Le recensioni più utili possono essere infine segnalate dagli utenti attraverso l'apposizione di un "like".

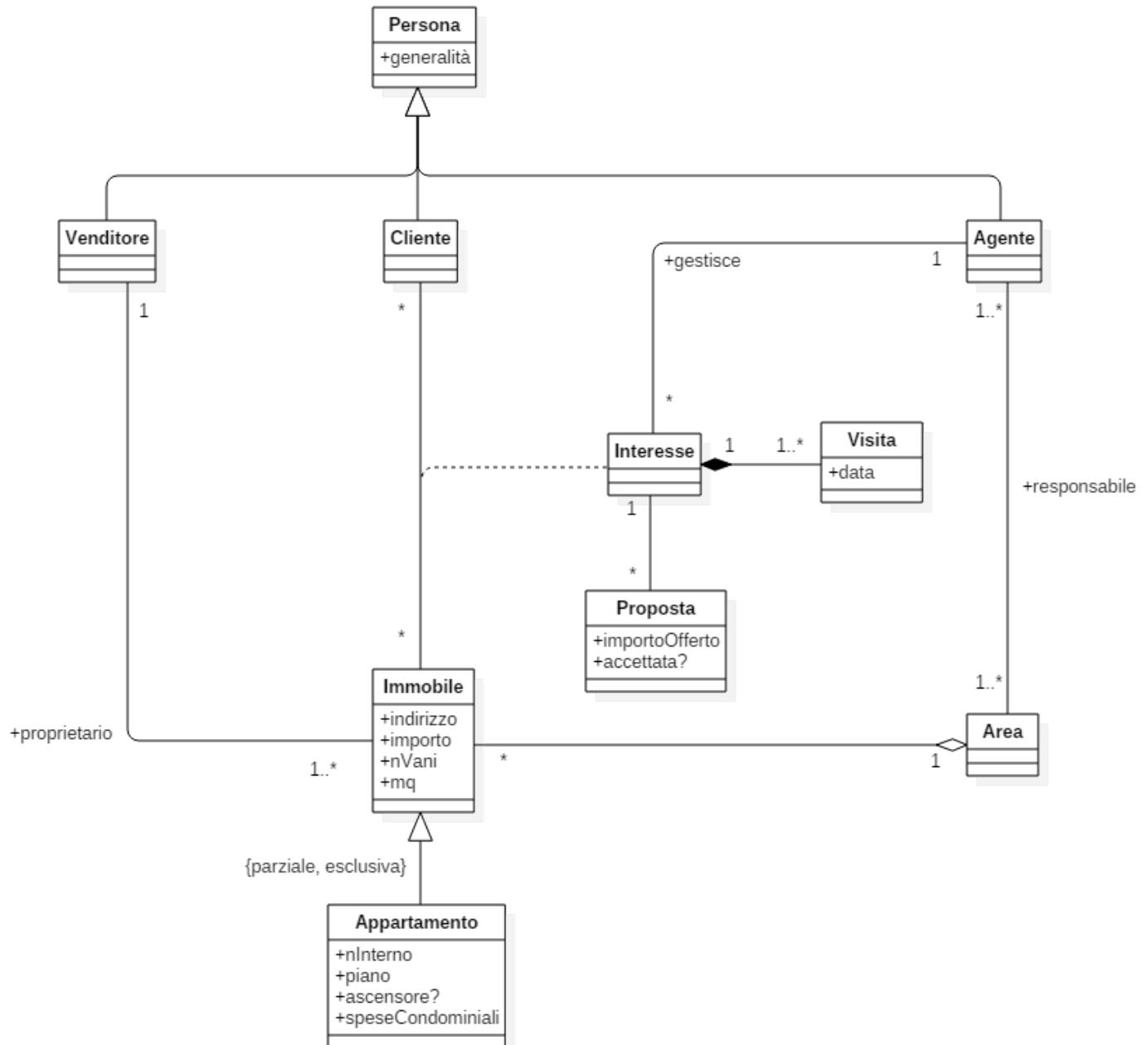
Utilizzando il formalismo UML, si disegni il diagramma delle classi per il dominio applicativo descritto.



L'agenzia immobiliare DanceCasa necessita di un software per gestire la compravendita di immobili a Cesena. Di ogni immobile si conosce l'indirizzo, l'importo richiesto, il numero di vani ed i metri quadrati commerciali, oltre alle generalità del venditore. Nel caso di appartamenti, si conosce anche il numero dell'appartamento all'interno del condominio, il piano, l'eventuale presenza dell'ascensore e le spese condominiali. Gli immobili sono suddivisi in diverse aree geografiche della città, ciascuna delle quali è gestita da almeno un agente immobiliare. Gli agenti si occupano di gestire le visite dei clienti nei diversi immobili. Un cliente può visitare diversi immobili (anche più volte), ma non può visitare lo stesso immobile con agenti diversi; inoltre, si vogliono memorizzare le date di ogni singola visita. Il software deve gestire anche le proposte di acquisto di un immobile presentate da un cliente, indicando l'importo che è stato offerto e l'eventuale accettazione della proposta da parte del venditore. Se la proposta viene rifiutata, il cliente è libero di effettuare una nuova proposta; se la proposta viene accettata, l'agente che ha seguito il cliente durante le visite riceve una percentuale sull'importo della vendita.

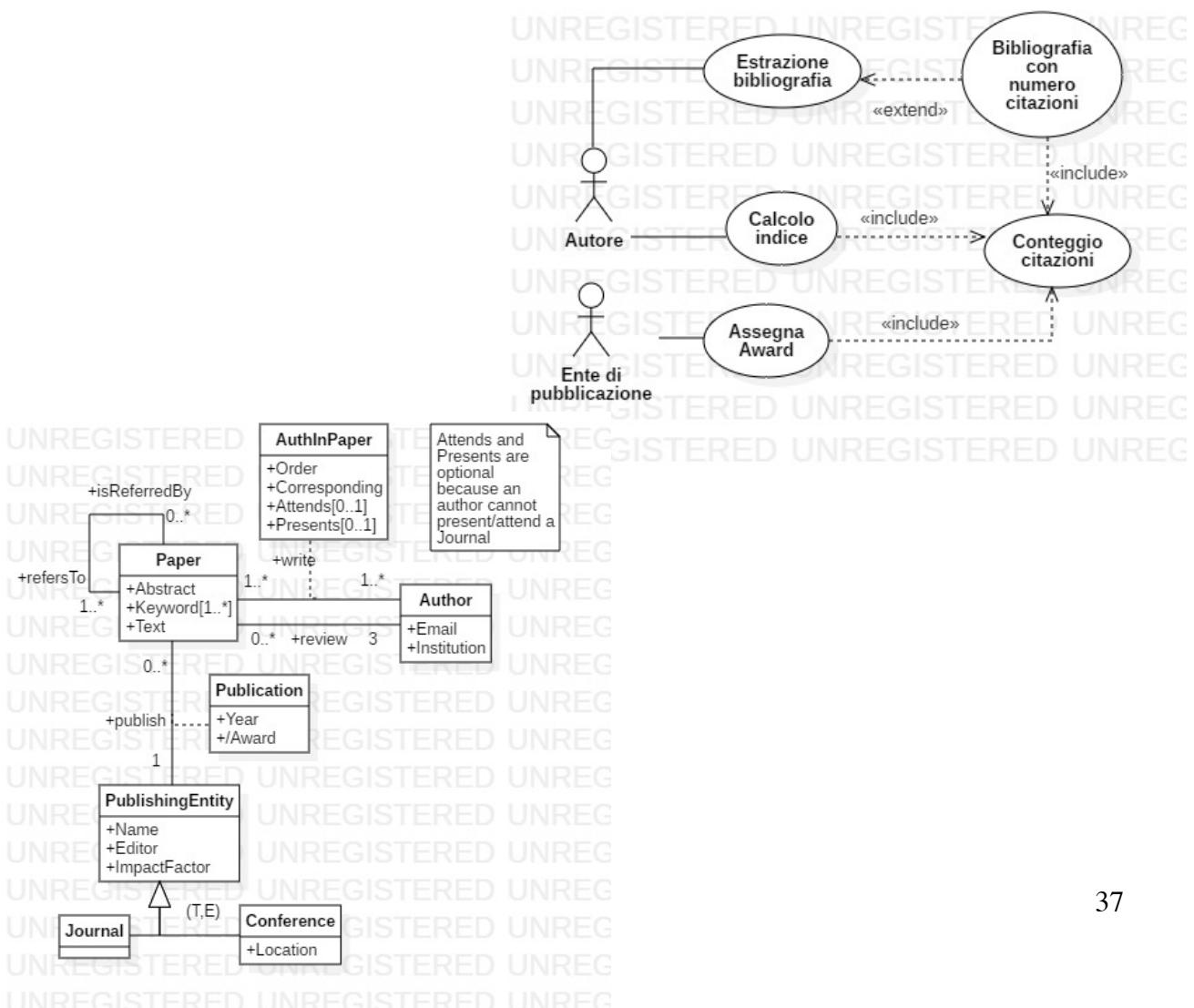
Utilizzando il formalismo UML, si disegnino il diagramma delle classi e dei casi d'uso per il dominio applicativo descritto.





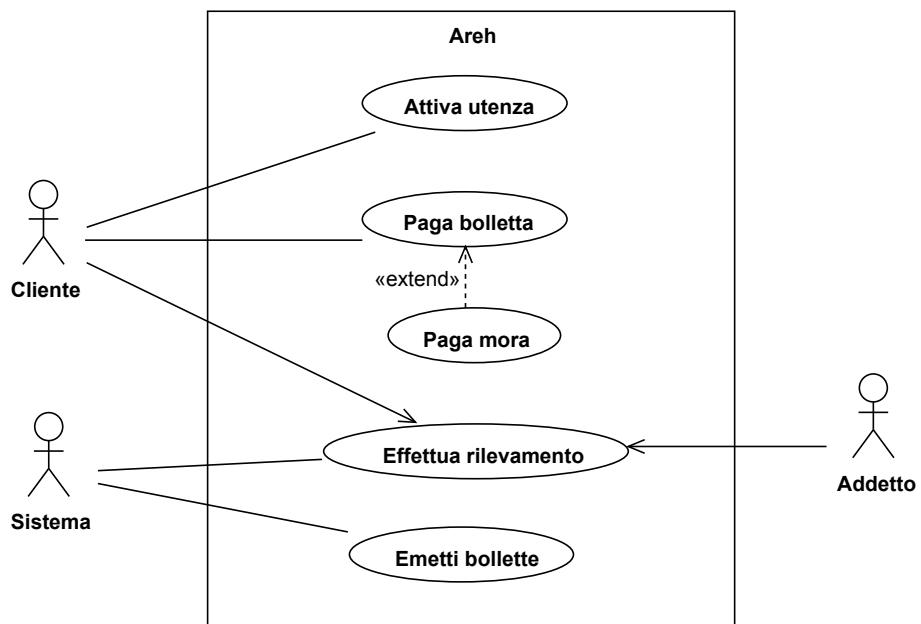
Si vuole modellare la banca dati che gestisce gli articoli pubblicati da ricercatori, al fine di automatizzare la gestione della bibliografia (cioè l'elenco delle pubblicazioni per ricercatore) e il calcolo dell'indice di produttività dei singoli ricercatori (indice basato sul numero di citazioni per articolo). Ogni articolo è descritto da un titolo, da una lista di parole chiave e da un sommario, cita almeno un altro articolo e può essere citato da altri articoli. Un articolo è scritto da almeno un autore di cui si conoscono email e istituto di affiliazione. Per ogni articolo si memorizzano l'ordine degli autori (primo, secondo, terzo, etc.) e un flag che rappresenta il "corresponding author", ossia l'autore cui indirizzare eventuale corrispondenza relativa all'articolo (ci possono essere più corresponding author per ogni articolo). Ogni articolo è pubblicato da un ente di pubblicazione (di cui si conoscono il nome e l'editor) che può essere di tipo "Rivista" o "Conferenza"; in quest'ultimo caso, si conosce anche il luogo in cui la conferenza si svolge. Un articolo può essere pubblicato una sola volta e della pubblicazione si conoscono l'anno e il fattore di impatto (numerico) dell'ente di pubblicazione. Per poter essere pubblicato, un articolo deve essere valutato da 3 revisori (che sono a loro volta ricercatori). Nel caso cui l'ente di pubblicazione sia di tipo Conferenza, è obbligatorio per almeno uno degli autori dell'articolo partecipare alla conferenza. Tra gli autori partecipanti alla conferenza, si memorizza quello che presenterà oralmente l'articolo pubblicato. Infine, ogni anno l'ente di pubblicazione premia l'articolo (pubblicato dallo stesso ente di pubblicazione 10 anni prima) con il maggior numero di citazioni.

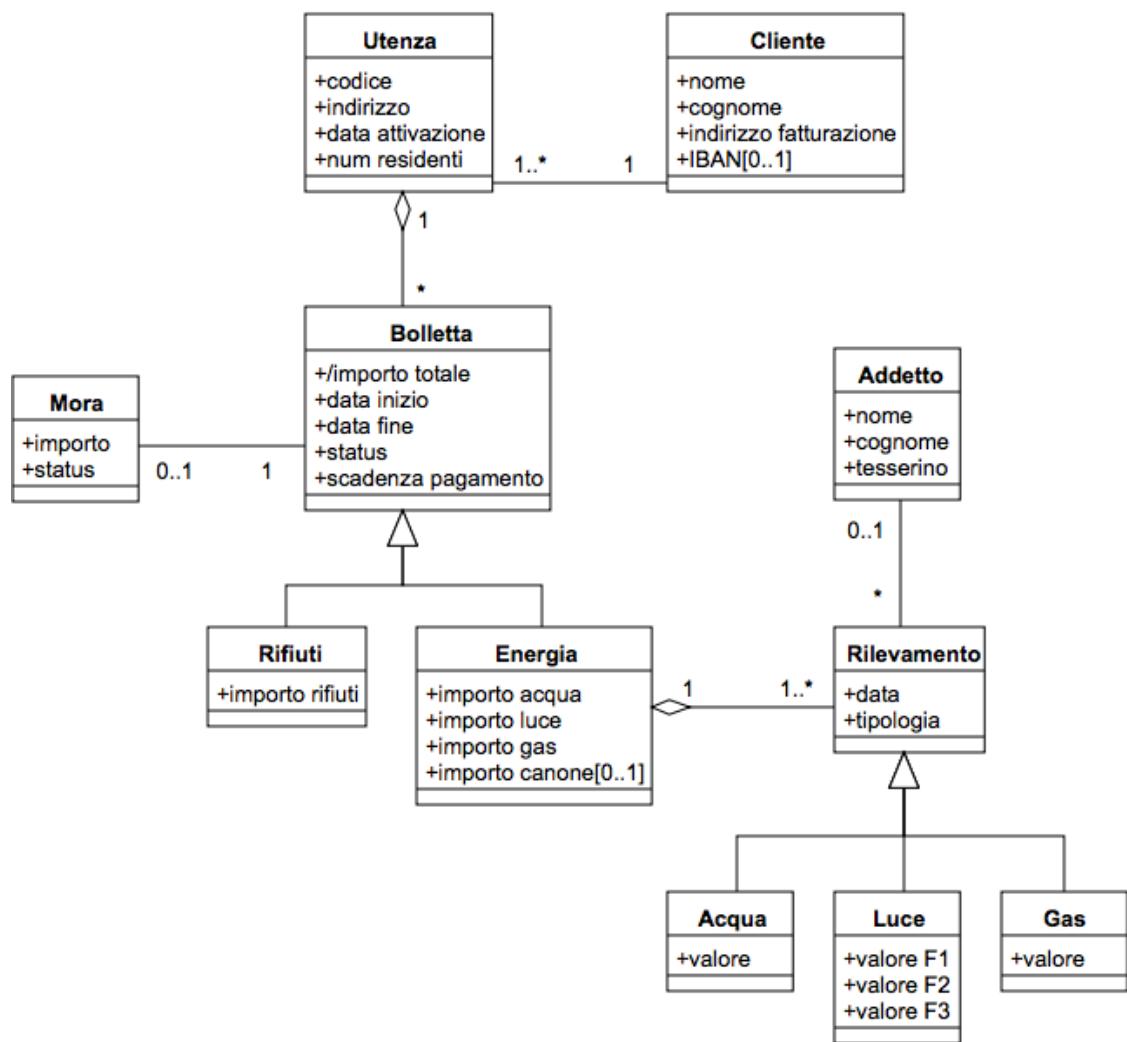
Utilizzando il formalismo UML, si disegnino il diagramma delle classi e dei casi d'uso per il dominio applicativo descritto.



La società Areh gestisce i contratti relativi ai consumi energetici nel territorio emiliano-romagnolo. Un contratto (o utenza) viene stipulato da un cliente, del quale si conoscono dati anagrafici, indirizzo di fatturazione ed eventuale IBAN su cui addebitare automaticamente le bollette. L'utenza (di cui si memorizzano il codice, l'indirizzo effettivo, il numero di residenti nell'indirizzo e la data di attivazione) comprende un insieme di bollette, che vengono solitamente emesse con cadenza bimestrale. Ogni bolletta indica la data di inizio, quella di fine, l'importo totale, lo status (pagata/non pagata) e la data di scadenza del pagamento; se il cliente tarda nel pagamento, viene emessa una mora (di cui si memorizzano l'importo e lo status). Le bollette riportano anche gli importi di dettaglio, i quali sono limitati ad un'unica voce nel caso in cui la bolletta riguardi i rifiuti, mentre si scompongo in importi per acqua, luce e gas nel caso in cui la bolletta riguardi l'energia; a questi ultimi può essere aggiunto anche l'importo del canone tv. Ad ogni bolletta fanno riferimento un insieme di rilevamenti eseguiti all'indirizzo dell'utente per verificare i consumi effettivi. Di ogni rilevamento si conoscono la data di esecuzione, la tipologia (eseguito dal cliente/eseguito dall'addetto/stimato) ed i valori rilevati sui contatori. Se il rilevamento è stato eseguito da un addetto, si memorizzano nome, cognome e tesserino di quest'ultimo. I valori rilevati dipendono dal tipo di energia: nel caso di acqua e gas si tratta di un unico valore, mentre nel caso della luce si tratta di tre valori distinti (fascia1, fascia2 e fascia3).

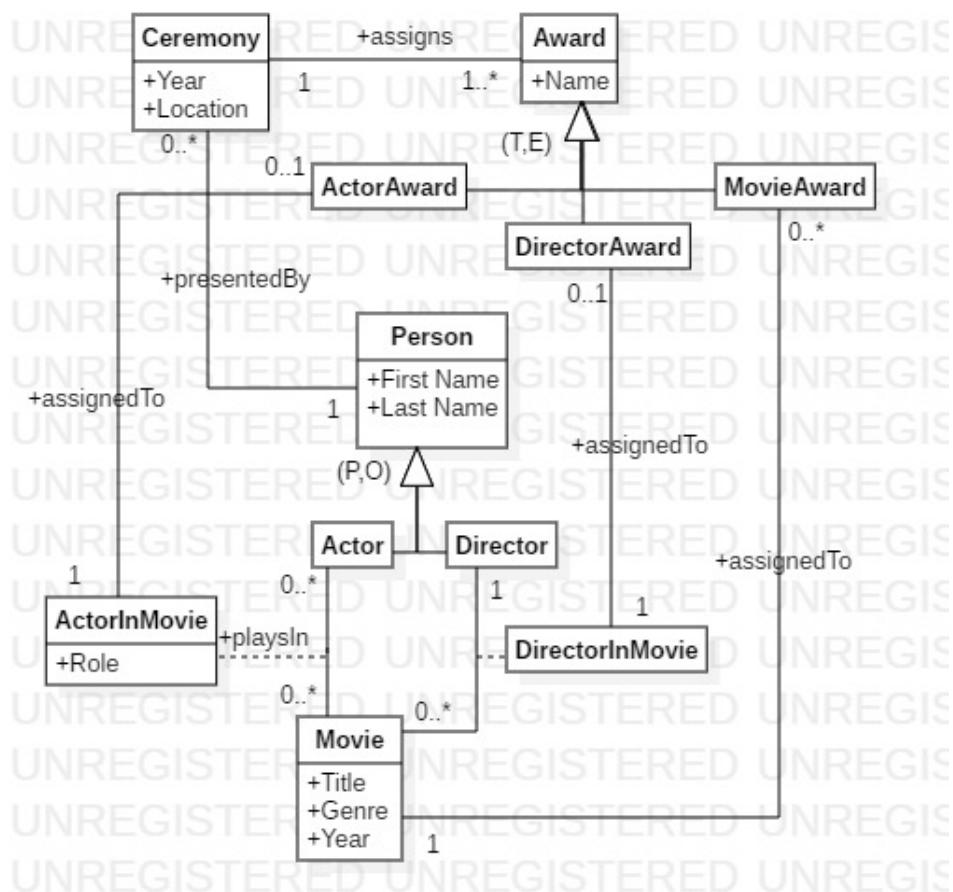
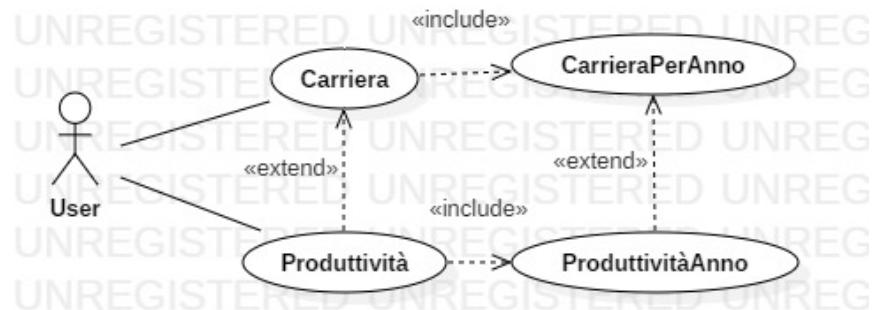
Si modellino le specifiche sopra riportate in UML attraverso un diagramma dei casi d'uso e un diagramma delle classi.





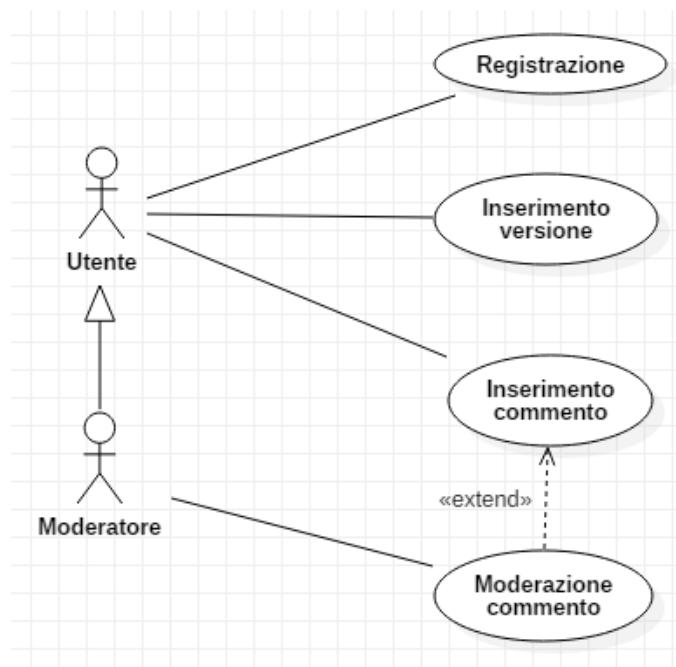
Si vuole modellare la banca dati che gestisce l'assegnazione dei premi Oscar ad attori, registi e film, con l'obiettivo di monitorare, per ogni attore/regista, la carriera (in termini del numero di Oscar vinti) e la produttività (numero di film), a livello sia annuale sia complessivo. La cerimonia di assegnazione degli Oscar si tiene ogni anno in una location diversa. Ogni cerimonia ha un presentatore, che può anche non essere un attore o un regista. Di ogni attore e regista si registrano nome e cognome (un attore può anche essere regista), di ogni film si conoscono titolo, genere e anno di produzione. Per ogni film si registrano il regista e tutti gli attori partecipanti con il rispettivo ruolo. Durante la cerimonia vengono assegnati in modo esclusivo diversi premi a un film (per esempio miglior film o migliore colonna sonora), a un attore (miglior attore protagonista e non protagonista) o un regista (miglior regista). Attori e registi possono ricevere un solo riconoscimento per film, mentre un film può riceverne più di uno.

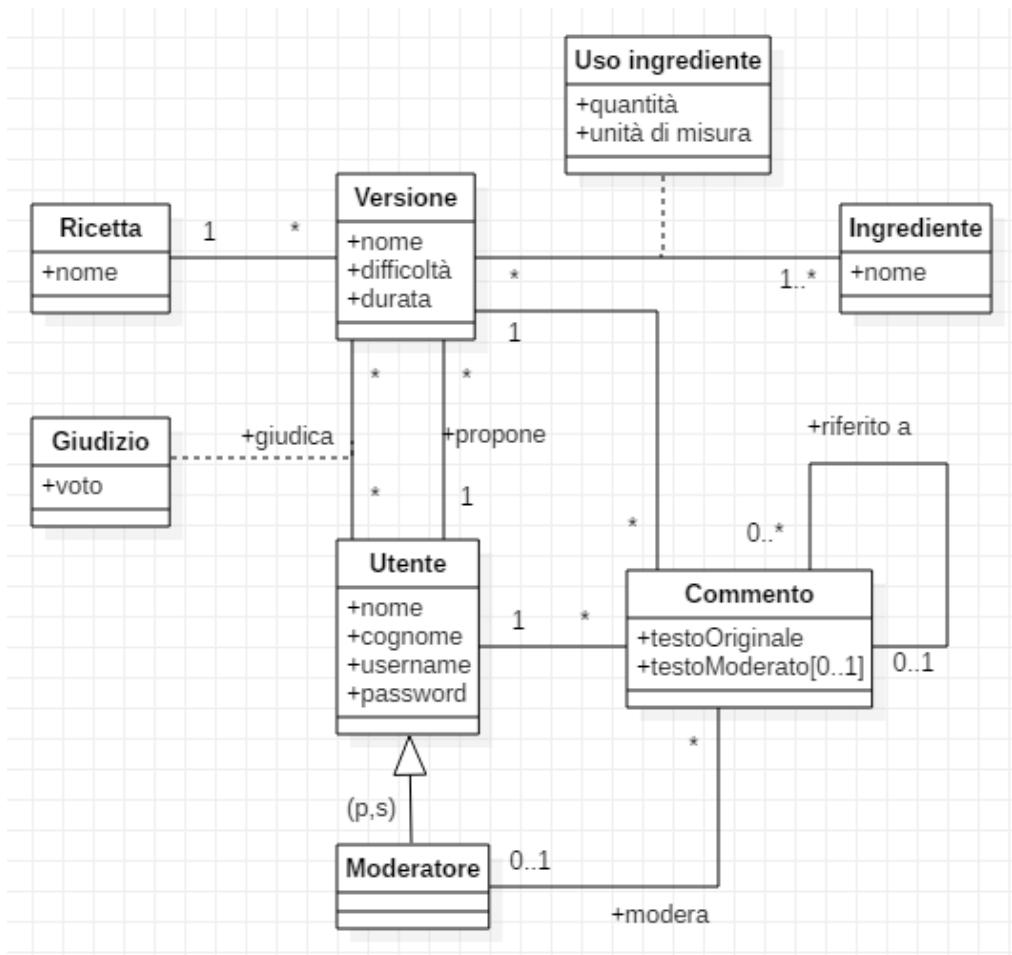
Utilizzando il formalismo UML, si disegnino il diagramma delle classi e dei casi d'uso per il dominio applicativo descritto.



Il sito verdepesto.it permette agli utenti registrati di inserire ricette, valutarle e commentarle. In fase di registrazione, gli utenti (alcuni dei quali possono svolgere il ruolo di moderatore) devono inserire nome, cognome, username e password. Il sito contiene un'anagrafica di ricette standard (identificate da un nome) di cui gli utenti possono proporre la propria versione. Ogni versione inserita deve riportare un nome ed un'indicazione della durata e del livello di difficoltà, oltre all'elenco degli ingredienti utilizzati; di ogni ingrediente va indicata la quantità e l'unità di misura (in grammi o in unità). Una volta inserita sul sito, la versione di una ricetta può essere valutata e commentata dagli altri utenti. In particolare, un utente può esprimere un giudizio (con un voto da 1 a 10; non si può esprimere più di un giudizio per versione) o lasciare dei commenti sulle versioni. I commenti possono essere effettuati direttamente sulla versione della ricetta, oppure possono essere lasciati in risposta ad un altro commento. I moderatori possono intervenire sui commenti pubblicati modificandone il testo per censurare URL e linguaggio non consono; in tal caso, il sistema tiene traccia sia del testo originale che di quello moderato.

Utilizzando il formalismo UML, si disegnino il diagramma delle classi e dei casi d'uso per il dominio applicativo descritto.

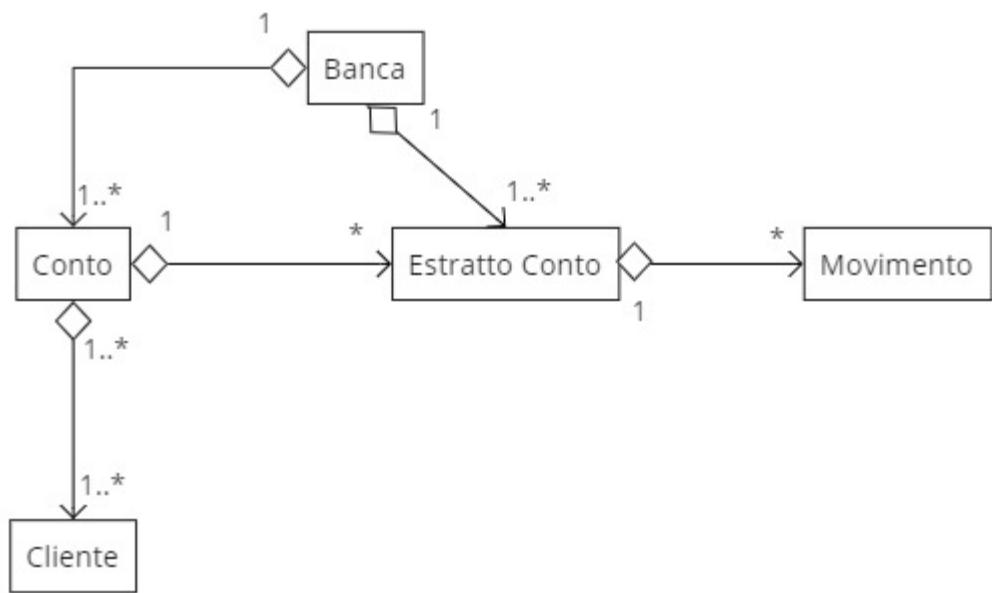
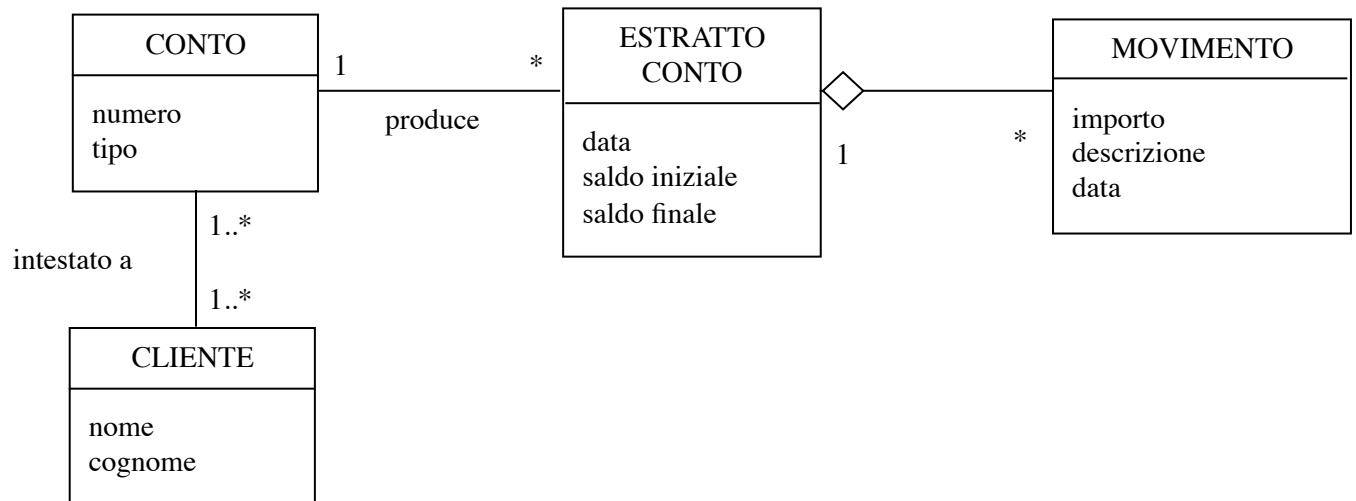




Progettazione delle associazioni

È dato il diagramma delle classi UML in figura, che descrive gli estratti conto inviati da una banca. Si effettui la progettazione delle associazioni considerando il seguente carico di lavoro:

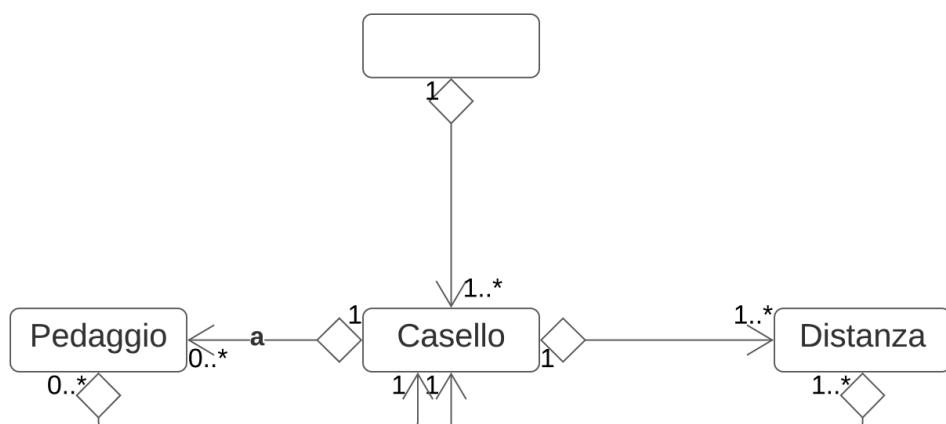
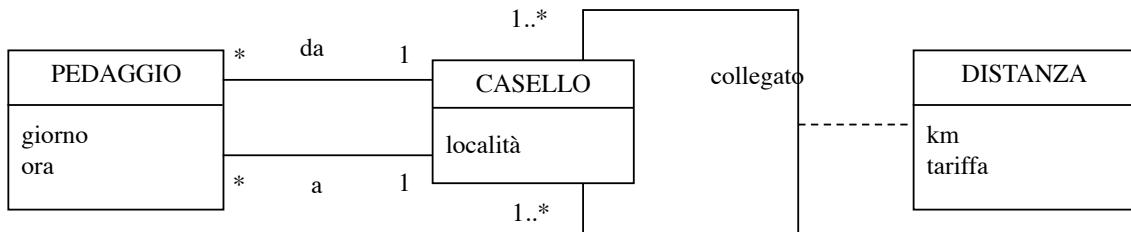
- calcolo del saldo finale di un estratto conto dato il saldo iniziale e i movimenti che lo costituiscono (100 volte al mese)
- stampa, per un dato cliente, di un resoconto annuale che riporti il saldo complessivo dei suoi conti correnti al termine di ciascuno dei 12 mesi (100 volte all'anno)
- dato un numero di conto corrente, stampa dei dati degli intestatari e del più recente estratto conto (10 volte al mese).



mirò

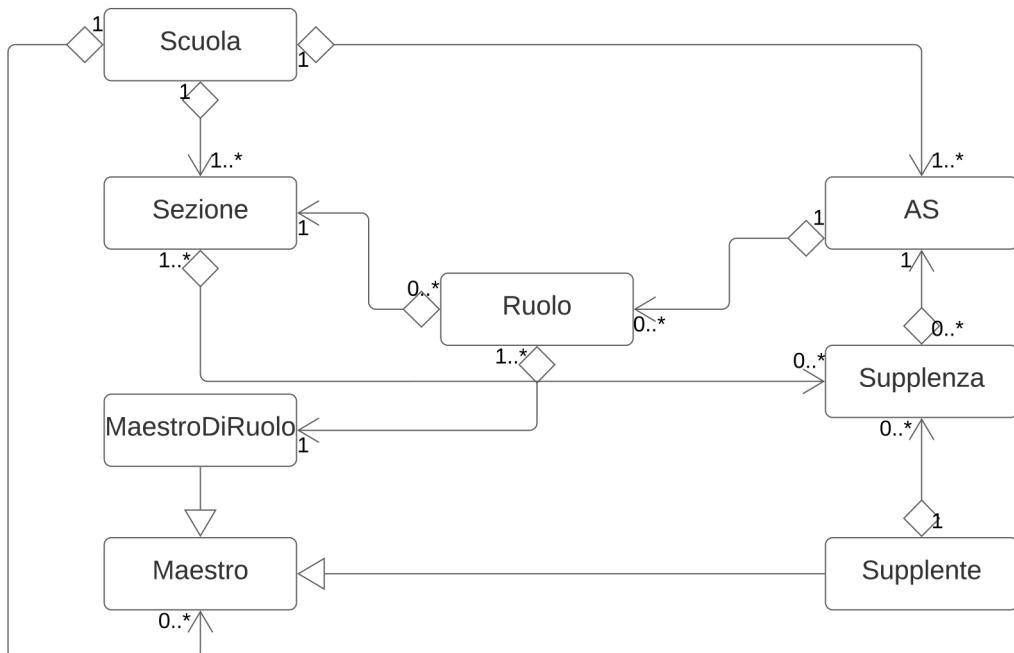
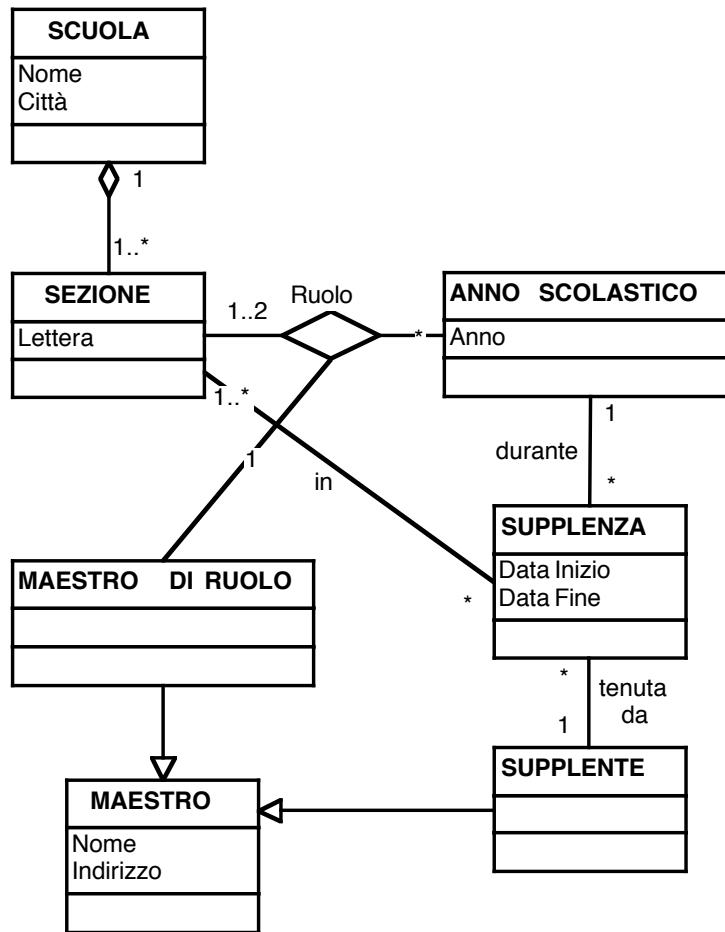
È dato il diagramma delle classi UML in figura, che descrive il sistema di pedaggi di un'autostrada. Si suppone che, nell'ottica di una gestione completamente centralizzata, il database sia ospitato in un'unica copia presso la sede della società autostradale. Si effettui la progettazione delle associazioni considerando il seguente carico di lavoro:

- riepilogo degli incassi effettuati presso un dato casello (1 volta al giorno)
- inserimento di un nuovo pedaggio (presso un dato casello e dato il casello di provenienza) e calcolo della tariffa (1000 volte al giorno)
- rincaro di una data percentuale per tutte le tariffe (1 volta all'anno)



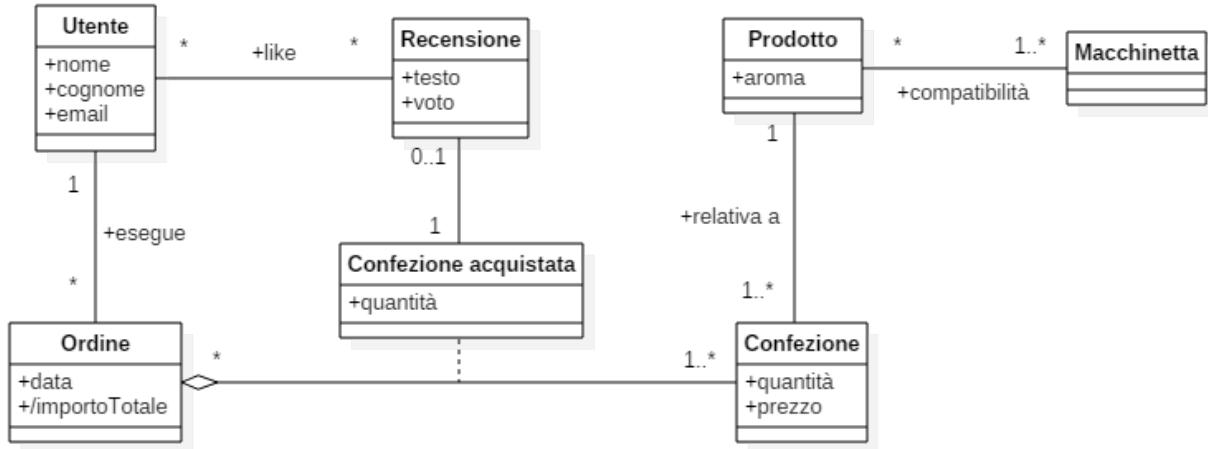
Dato il diagramma delle classi in figura, si effettui il progetto delle associazioni considerando un carico di lavoro così composto:

1. Elenco dei maestri supplenti e, per ciascuno, calcolo del numero di supplenze tenute nell'anno scolastico corrente.
 2. Elenco dei maestri di ruolo nel corrente anno scolastico, riportando per ciascun maestro le sezioni in cui insegna.

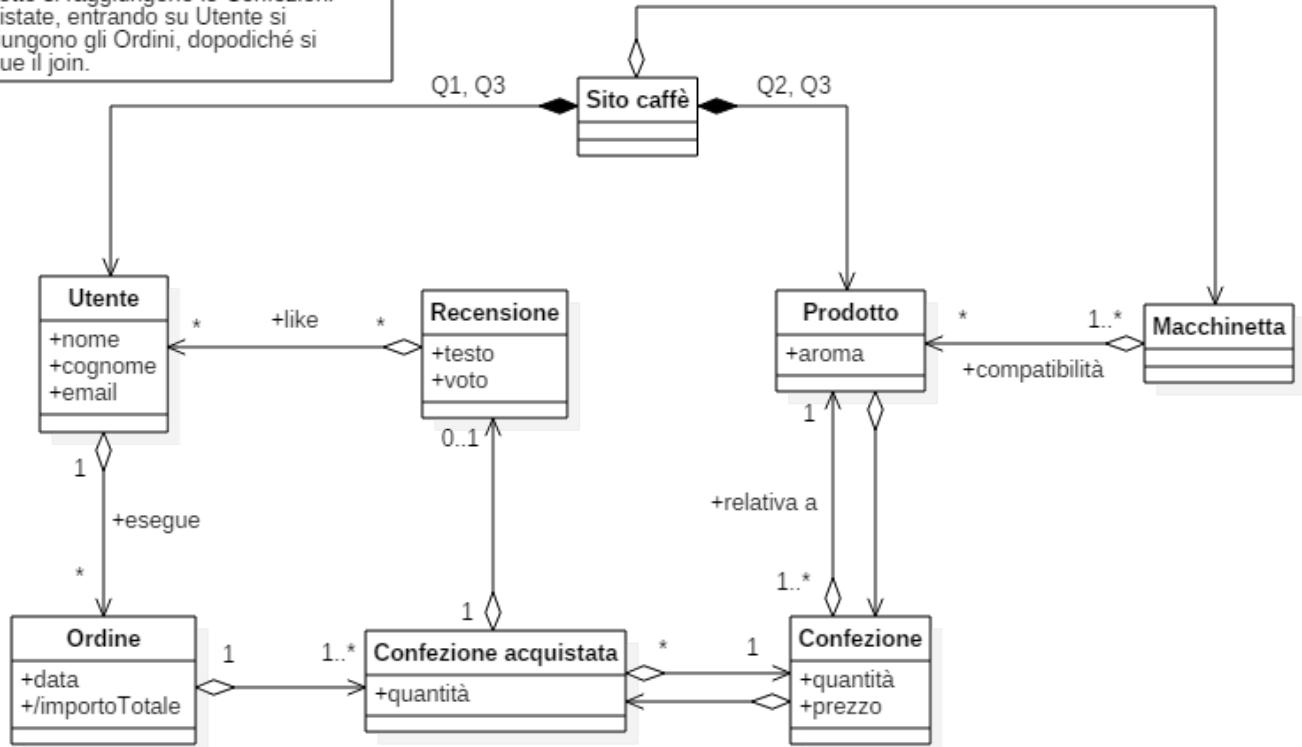


Dato il diagramma delle classi in figura, il quale rappresenta la gestione degli ordini da parte di un rivenditore di caffè, effettuare il progetto delle associazioni tenendo conto del seguente carico di lavoro:

- Visualizzare gli ultimi prodotti acquistati da un determinato utente (più volte al giorno).
- Visualizzare la media dei voti ricevuti da un dato prodotto (1 volta alla settimana).
- Visualizzare i prodotti che hanno ricevuto meno ordini (1 volta al mese).

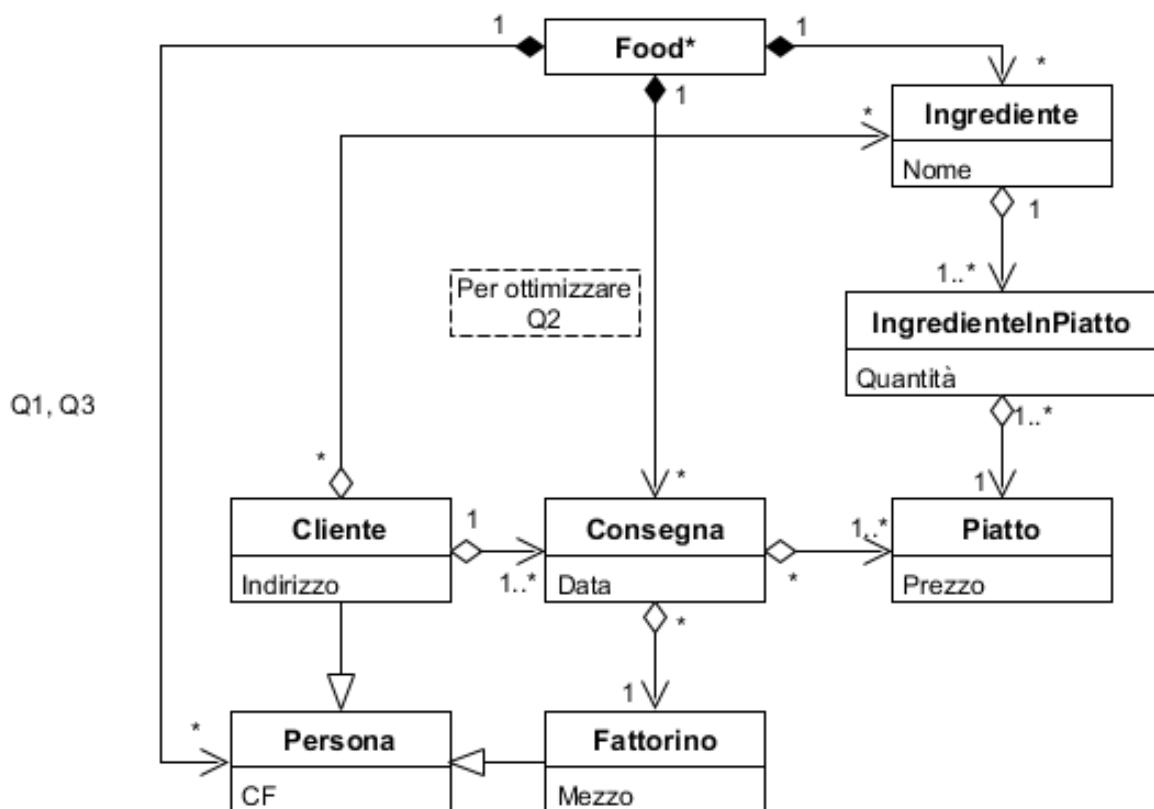
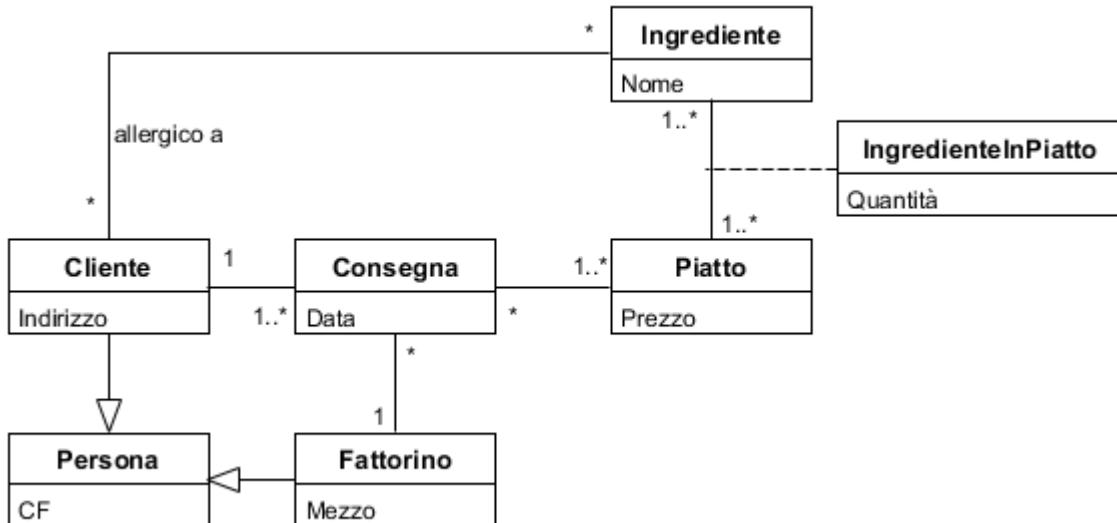


Q2 è ottimizzata entrando su Prodotto.
Q3 non è ottimizzata: entrando su Prodotto si raggiungono le Confezioni acquistate, entrando su Utente si raggiungono gli Ordini, dopodiché si esegue il join.



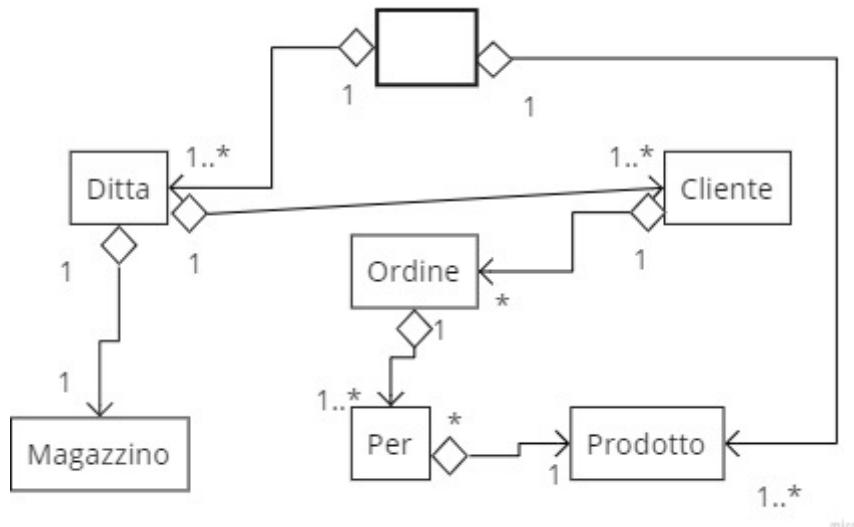
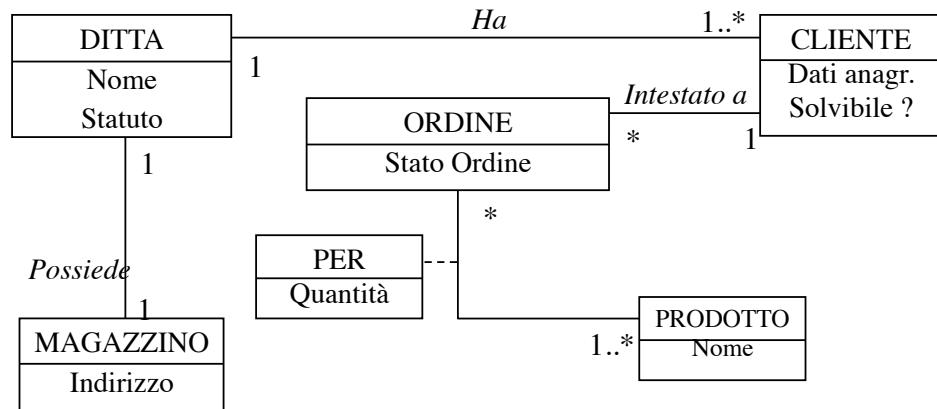
Dato il diagramma delle classi in figura, che rappresenta le consegne effettuate dalla società “Food*”, effettuare il progetto delle associazioni tenendo conto del seguente carico di lavoro:

1. Visualizzare tutti gli ingredienti a cui un dato cliente è allergico (per ogni consegna).
2. Visualizzare l'elenco di tutte le consegne effettuate in un dato intervallo temporale (1 volta alla settimana);
3. Visualizzare i piatti preferiti di ogni cliente (1 volta al mese)



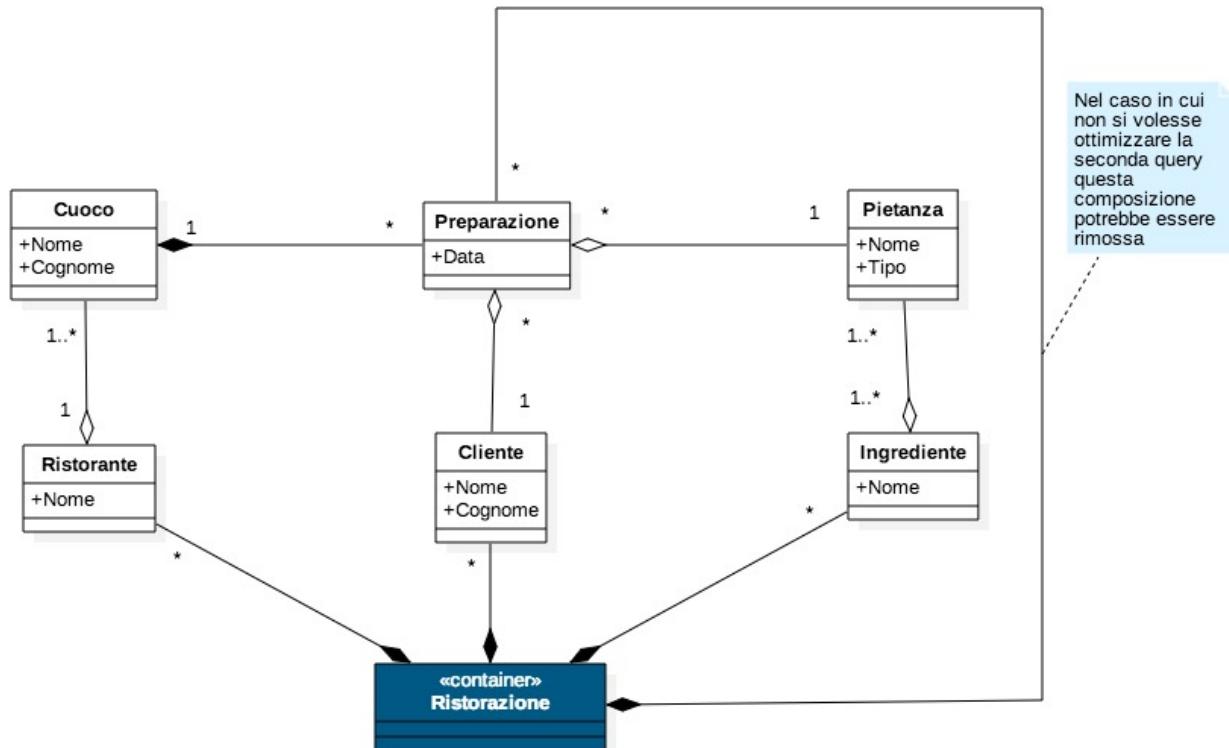
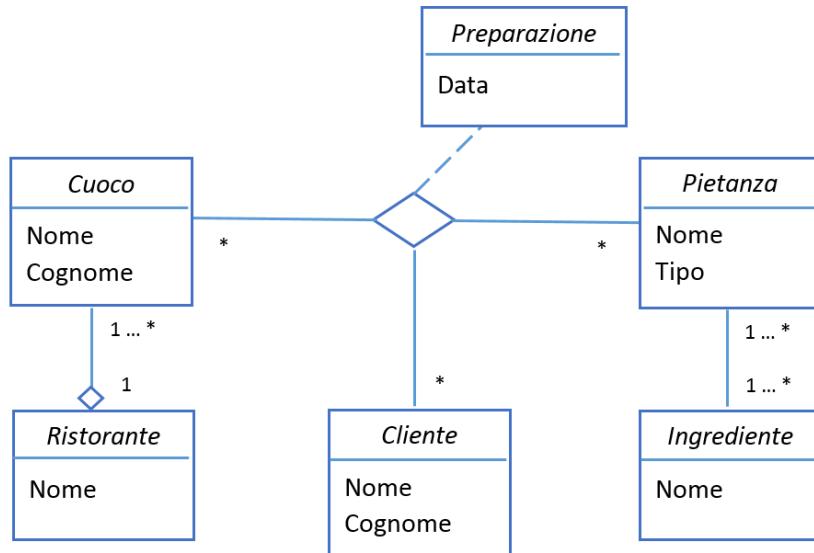
Dato il diagramma delle classi in figura, si effettui il progetto delle associazioni considerando un carico di lavoro così composto:

- Per una data ditta, stampa di tutti gli ordini con stato “Evaso”, includendo i dati anagrafici dell'intestatario e i prodotti ordinati con le relative quantità (1 volta al giorno).
- Stampa, per ciascun prodotto, della quantità complessiva di quel prodotto prevista da ordini con stato “Inevaso” (1 volta alla settimana).



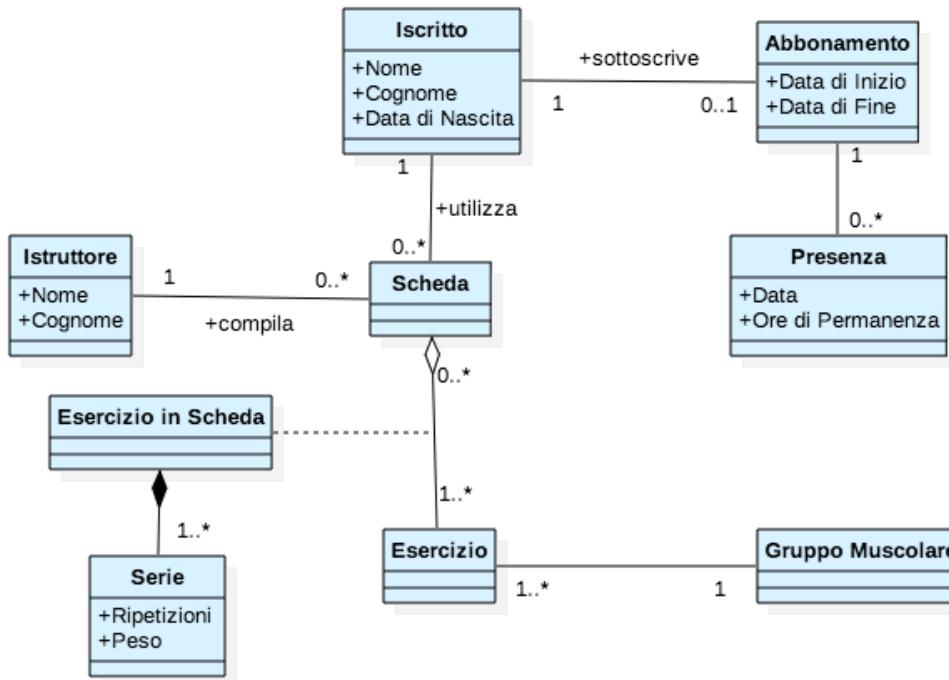
Dato il diagramma delle classi in figura, effettuare il progetto delle associazioni tenendo conto del seguente carico lavoro:

- Visualizzare tutte le pietanze preparate da tutti i cuochi di un dato ristorante (1 volta al giorno).
- Visualizzare l'elenco di tutte le preparazioni effettuate in un dato intervallo temporale (1 volta alla settimana).
- Visualizzare tutte le pietanze che utilizzano un dato ingrediente (1 volta al mese).

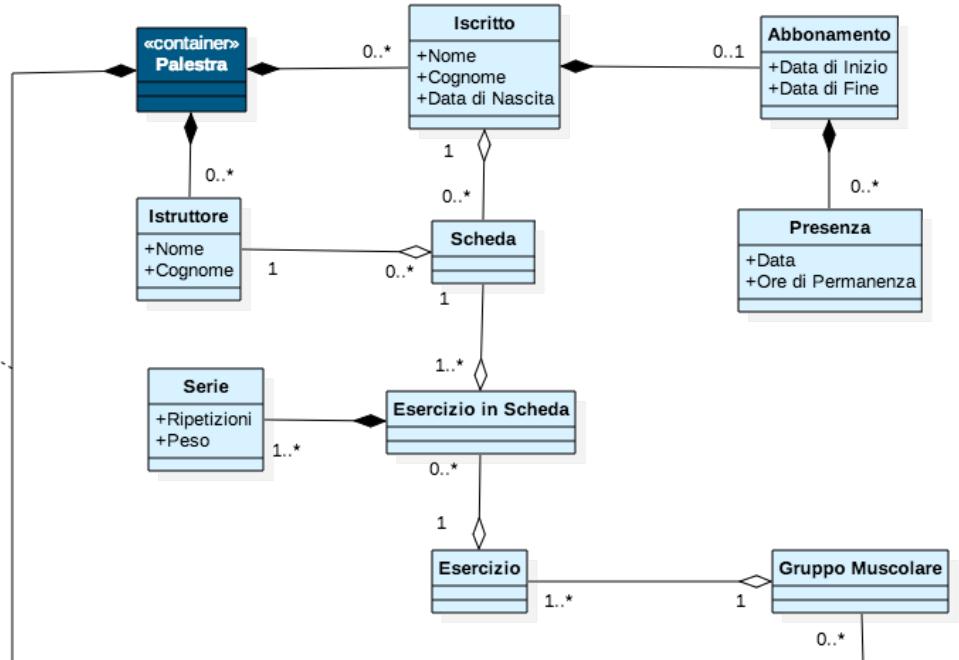


Dato il diagramma delle classi in figura relativo alla gestione di una palestra, svolgere il progetto delle associazioni tenendo conto del seguente carico di lavoro:

- Calcolare le ore totali di permanenza per ogni iscritto (una volta alla settimana).
- Dato un iscritto, elencare le schede che utilizza e gli istruttori che le hanno compilate (due volte alla settimana).
- Elencare i cinque esercizi prescritti più di frequente per un dato gruppo muscolare (una volta ogni sei mesi).

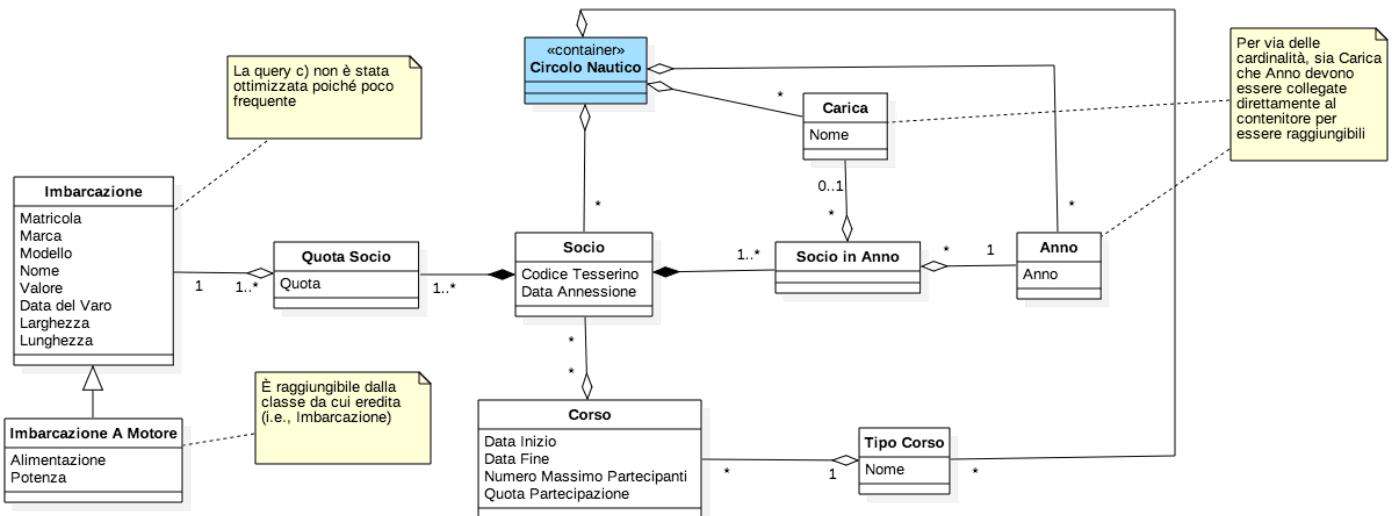
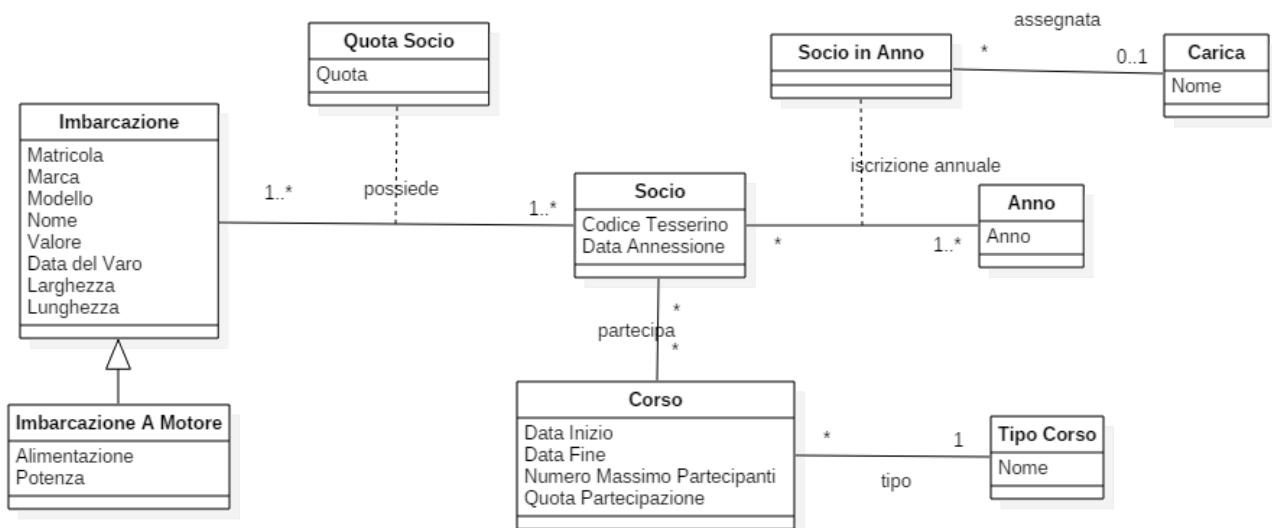


La terza query viene ottimizzata poiché sarebbe comunque necessario aggiungere un riferimento dal contenitore ad Esercizio (o a Gruppo Muscolare) per via dell'ozionalità tra Esercizio e Esercizio in Scheda.



Dato il diagramma delle classi in figura, effettuare il progetto delle associazioni tenendo conto del seguente carico di lavoro:

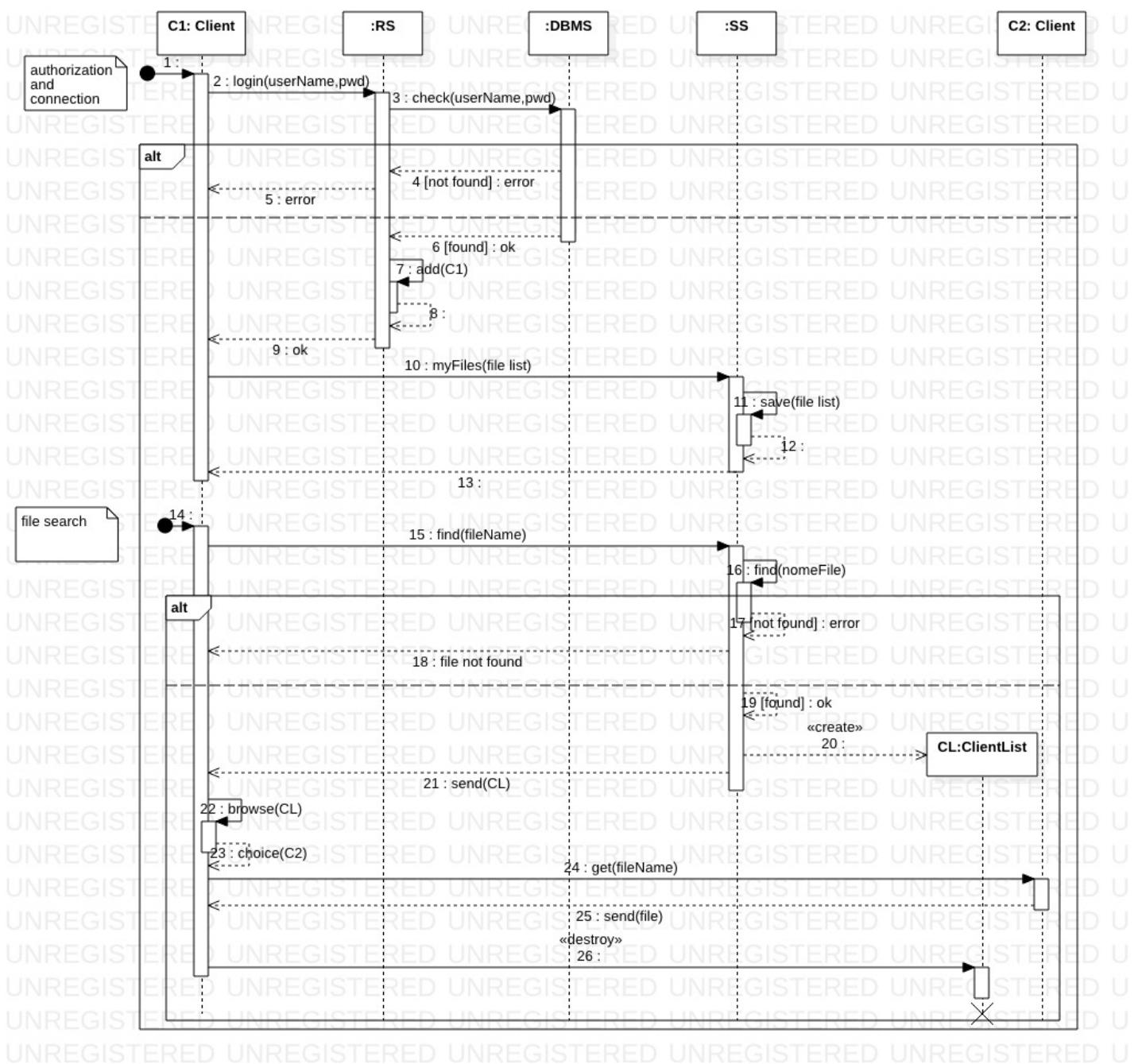
- Visualizzare il valore totale delle imbarcazioni possedute da un dato socio (1 volta al giorno).
- Per ogni tipo di corso visualizzare il numero medio di partecipanti e il numero medio di anni di appartenenza al circolo dei soci che hanno partecipato a corsi di quel tipo (1 volta alla settimana).
- Data la matricola di un'imbarcazione, verificare se ad almeno uno dei soci a cui appartiene è mai stata assegnata una carica (1 volta all'anno).



Diagrammi di sequenza

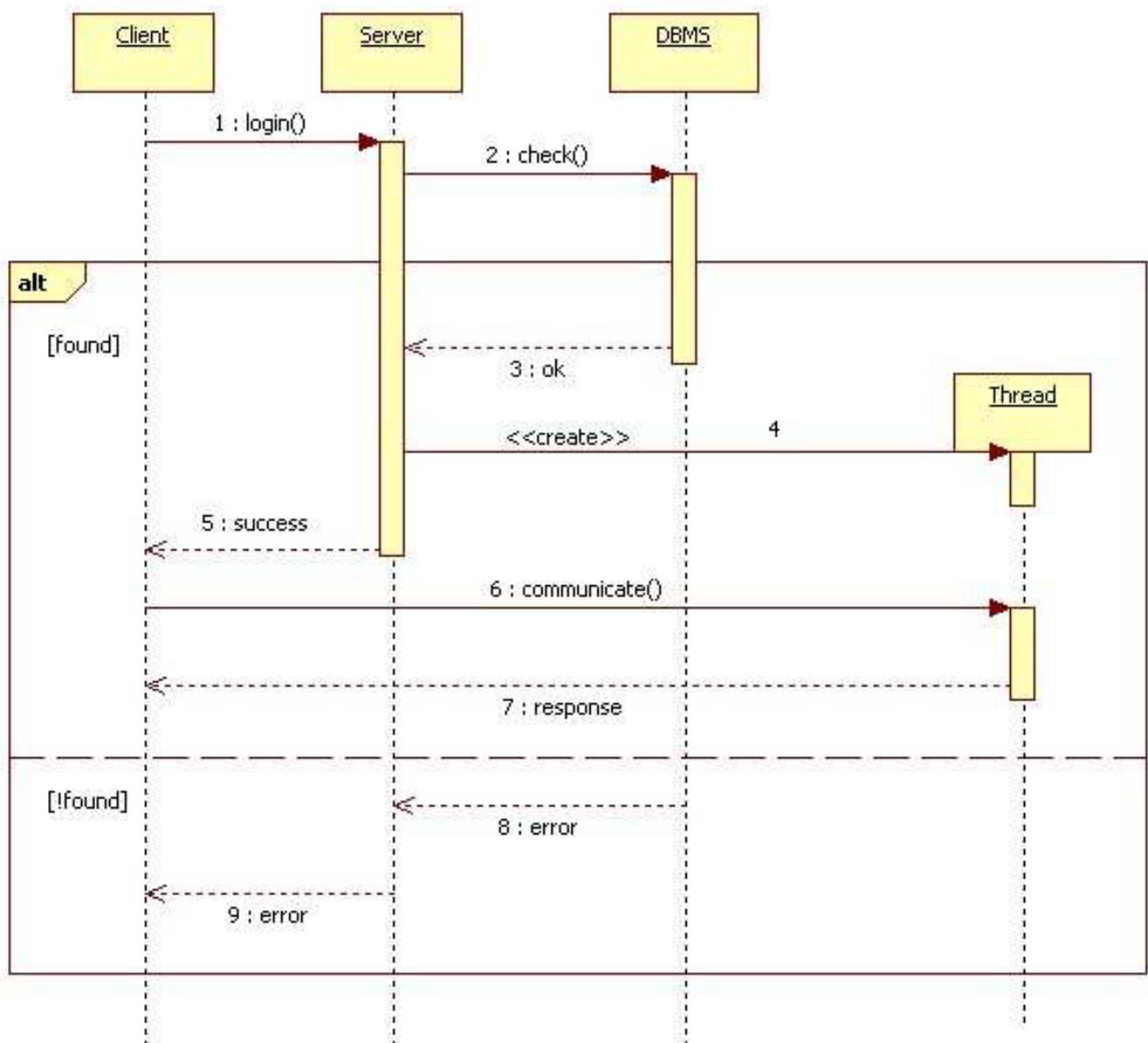
Realizzare un diagramma di sequenza UML relativo al seguente dominio applicativo:

In un sistema peer to peer lo scambio di file fra due client richiede l'esecuzione di diversi passi. Per prima cosa è necessario che un client (C1) si colleghi ad un server di registrazione (RS) tramite l'invio del proprio username e della propria password. RS, prima di inserire C1 nel sistema, interroga un server DBMS per controllare la correttezza dei dati inseriti. Nel caso in cui la ricerca abbia esito negativo RS invia un messaggio d'errore a C1. Altrimenti, una volta inserito, C1 invia le informazioni relative ai file in proprio possesso ad un server di ricerca (SS) il quale le immagazzinerà in un proprio database. Le ricerche avvengono inviando le richieste a SS il quale, interrogando il proprio database, restituirà un messaggio di errore o creerà una lista di client in grado di soddisfare la richiesta, che invierà a C1. Una volta selezionato dalla lista il client che preferisce, C1 comunicherà direttamente con quest'ultimo per il download effettivo del file.



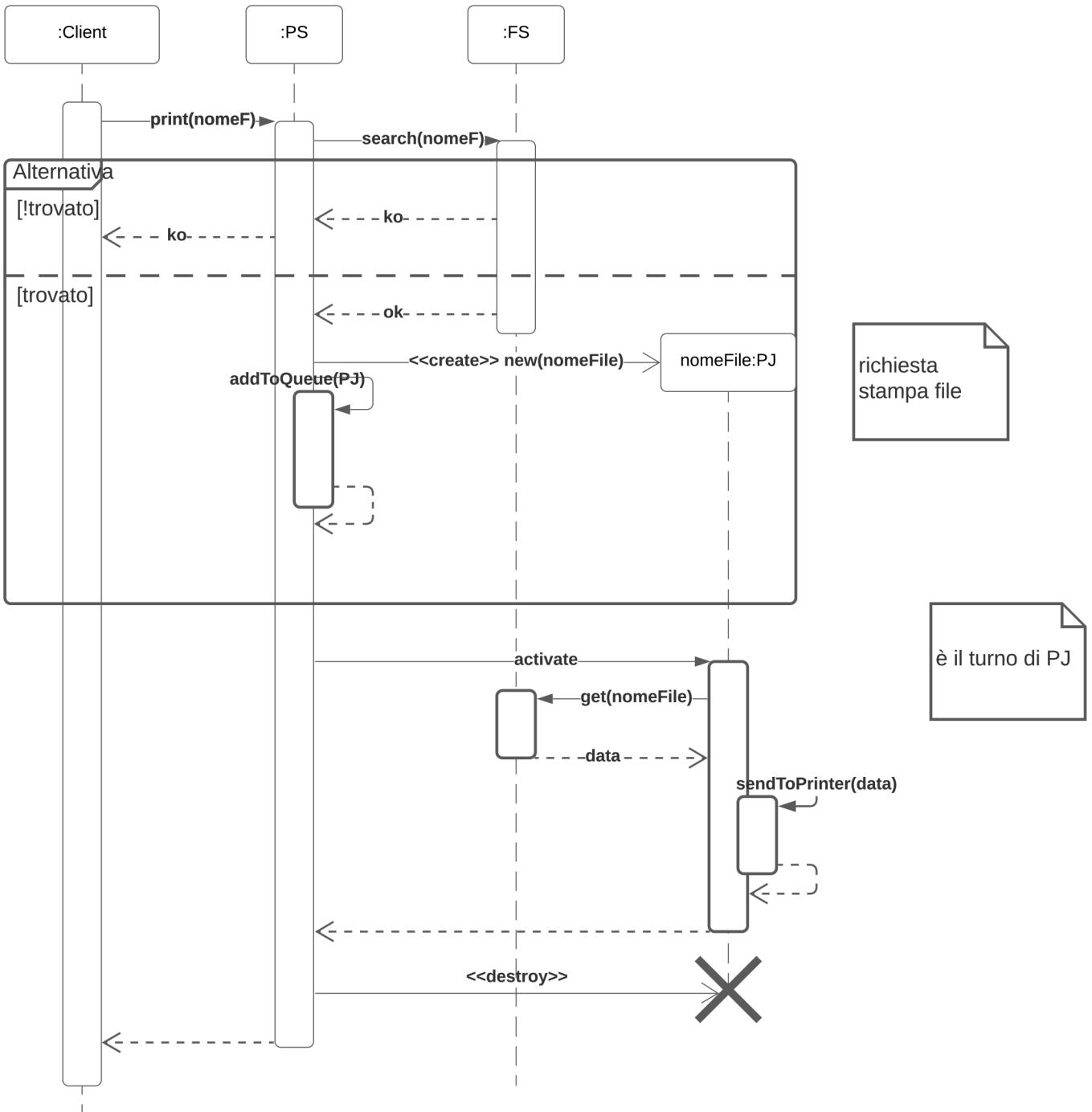
Realizzare un diagramma di sequenza UML che modelli il seguente dominio:

In un'applicazione client-server, l'autenticazione di ogni client presso il server avviene mediante inserimento della propria login e della propria password in una form di registrazione. Il controllo viene eseguito inviando i dati ad un server di autenticazione che a sua volta interroga un server DBMS. In caso di riconoscimento dell'utente il server di autenticazione procede alla creazione di un thread per la comunicazione con il particolare client.



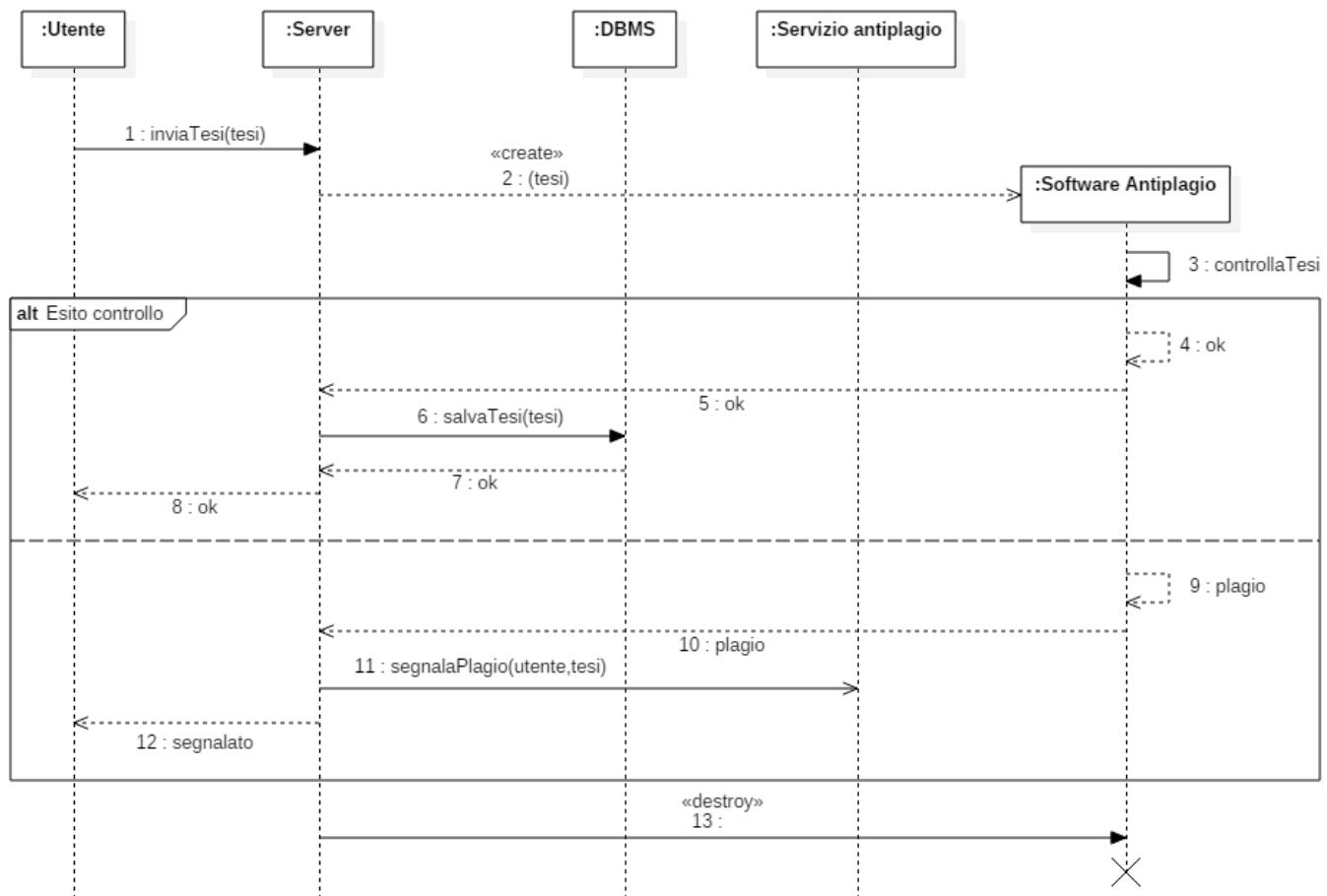
In un'applicazione client-server, un client, per poter stampare un file, deve innanzitutto effettuare una richiesta al Print Server (PS). Il PS interroga il File Server (FS) per individuare il file da stampare. Se la ricerca non ha successo il procedimento termina, altrimenti il PS genera un nuovo Print Job (PJ) e, richiamando un proprio metodo, lo inserisce nella coda dei job pendenti. Quando giunge il suo turno il PJ viene attivato e, dopo aver letto i dati da stampare dal FS, provvede al loro invio alla stampante. Terminata la stampa il PJ viene eliminato e viene notificato al client l'esito dell'operazione.

Costruire il diagramma di sequenza UML che rappresenta il dominio descritto.



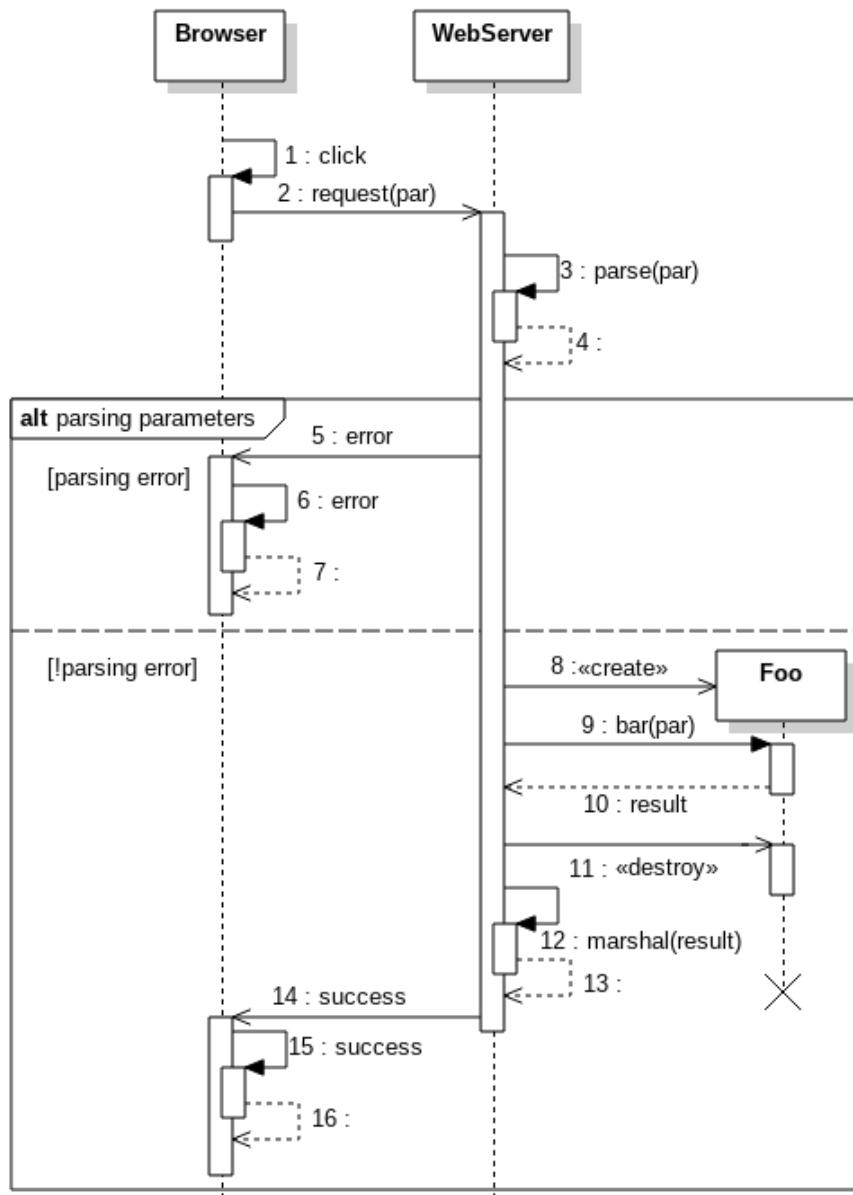
Modellare tramite *diagramma di sequenza UML* il seguente scenario, in cui viene descritta la sottomissione di una tesi di laurea da parte di uno studente.

Una volta collegato al sistema di gestione della carriera universitaria (si supponga che il login sia già stato effettuato), lo studente invia al server il file PDF della propria tesi di laurea. Prima di inserirla nel database, il server avvia un'istanza del software antiplagio per verificare l'originalità della tesi. Se il controllo non riscontra irregolarità, la tesi viene inviata al DBMS e l'utente viene notificato dell'avvenuta sottomissione. Qualora venga invece riscontrato un plagio, il server invia al Servizio Antiplagio dell'Università una debita segnalazione (inclusiva dei dati dello studente e del documento di tesi) e notifica lo studente dell'avvenuta segnalazione. In entrambi i casi, il server distrugge l'istanza del software a controllo terminato.



Modellare tramite *diagramma di sequenza UML* il seguente scenario che modella in maniera astratta la modalità di interazione client server in una specifica applicazione multi-tier.

Il diagramma descrive l'esecuzione di codice server-side a partire dal click di un bottone in una pagina web. Il browser (client) reagisce alla pressione del bottone invocando la funzione click() da cui è inviata una richiesta asincrona al web server. Il web server esegue quindi il parsing dei parametri: in caso di fallimento restituisce l'errore al browser, il quale invoca la funzione error(); in caso di successo, il web server istanzia un oggetto di classe Foo su cui invoca il metodo bar(parameters). Al termine dell'esecuzione il web server distrugge l'oggetto creato, procede con il preparazione del risultato e lo restituisce al browser che invoca la funzione success().

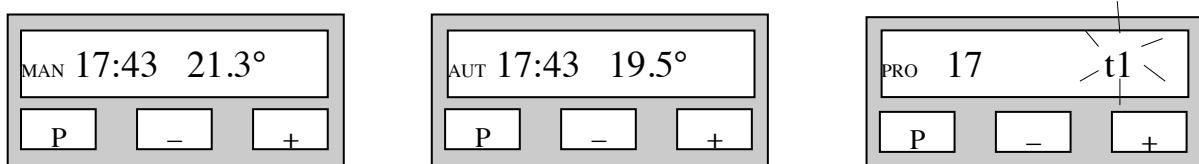


Diagrammi degli stati

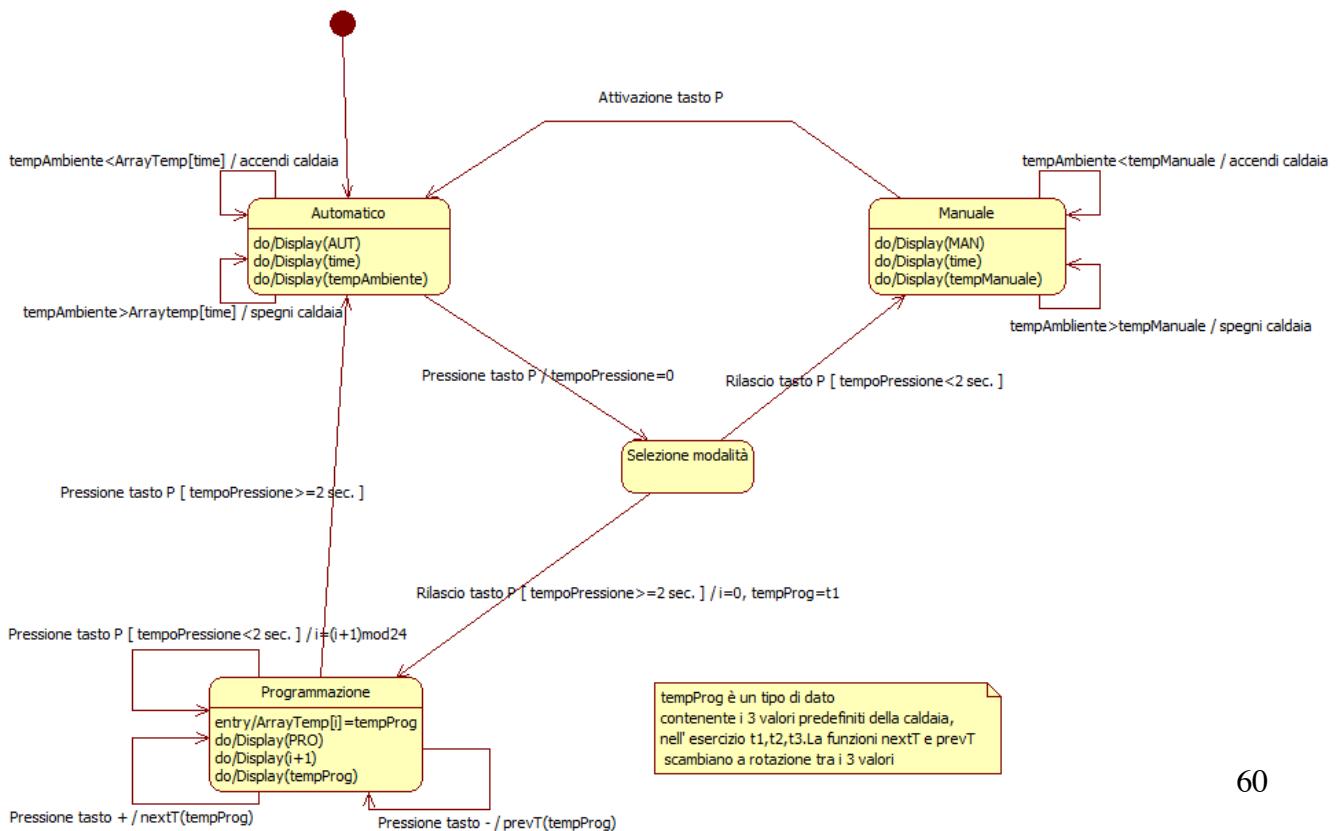
Un cronotermostato programmabile C da appartamento, da collegare al sistema di riscaldamento, può lavorare in due modalità: manuale e automatica, tra cui commuta a seguito della pressione del tasto P.

- **Modalità manuale:** l'utente seleziona una temperatura desiderata sul display, e C si impegna a mantenerla 24 ore su 24. Per selezionare la temperatura, espressa in gradi centigradi con 1 cifra di precisione decimale (es. 21.3 gradi), si utilizzano i tasti + e -. In questa modalità, il display visualizza l'ora e la temperatura desiderata.
- **Modalità automatica:** l'utente può programmare la temperatura desiderata per ciascuna delle 24 ore, scegliendola tra 3 valori disponibili pre-programmati: t1, t2 e t3. Quando C lavora in questa modalità, il display visualizza l'ora e la temperatura corrente dell'ambiente. Per la programmazione si procede come segue. Si entra in programmazione attraverso una pressione prolungata (almeno 2 secondi) del tasto P; a questo punto il display visualizza la prima delle 24 ore (00) e, lampeggiante, la temperatura (t1, t2 o t3) attualmente impostata, che l'utente può modificare a rotazione con i tasti + e -. Premendo il tasto P si passa alla prossima ora e si ripete il procedimento. Si può uscire in qualunque momento dalla programmazione, tornando in modalità automatica, se si preme P per almeno 2 secondi.

Sia in modalità manuale sia automatica, per mantenere nell'ambiente la temperatura t desiderata, C comanda un relè che accende e spegne la caldaia: quando la temperatura ambiente è inferiore a t la caldaia viene accesa, quando è maggiore di t viene spenta.

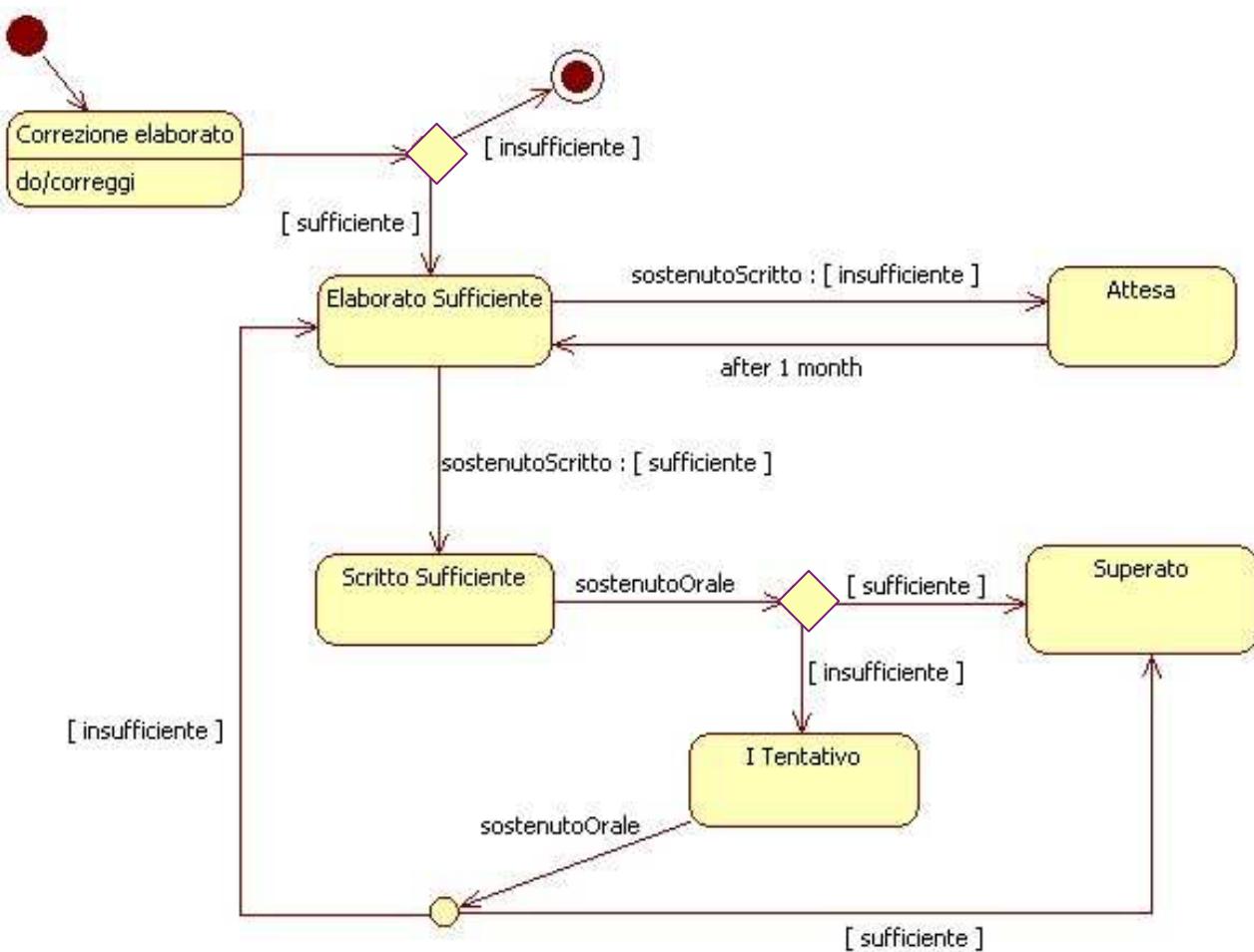


Si disegni un diagramma degli stati UML che modelli il cronotermostato, supponendo che le temperature programmate per la giornata siano contenute in un array temp[0..23] e che una funzione time restituisca, quando chiamata, l'ora attuale (compresa tra 0 e 24).



L'esame di un corso universitario prevede una prova scritta e una prova orale. Per poter effettuare l'esame è indispensabile consegnare preliminarmente un elaborato; se l'elaborato viene ritenuto sufficiente, si può sostenere la prova scritta. Se l'esito della prova scritta è positivo si può sostenere l'orale. Se lo scritto è insufficiente lo si può tentare di nuovo durante un appello successivo, a patto che la distanza tra i due appelli non sia inferiore a un mese; l'elaborato resta comunque valido. Nel caso in cui l'orale sia insufficiente, lo si può sostenere di nuovo durante un qualsiasi appello successivo; in caso di ulteriore esito negativo, lo studente dovrà sostenere di nuovo anche la prova scritta.

Si disegni il diagramma di stato UML della classe *Esame*.

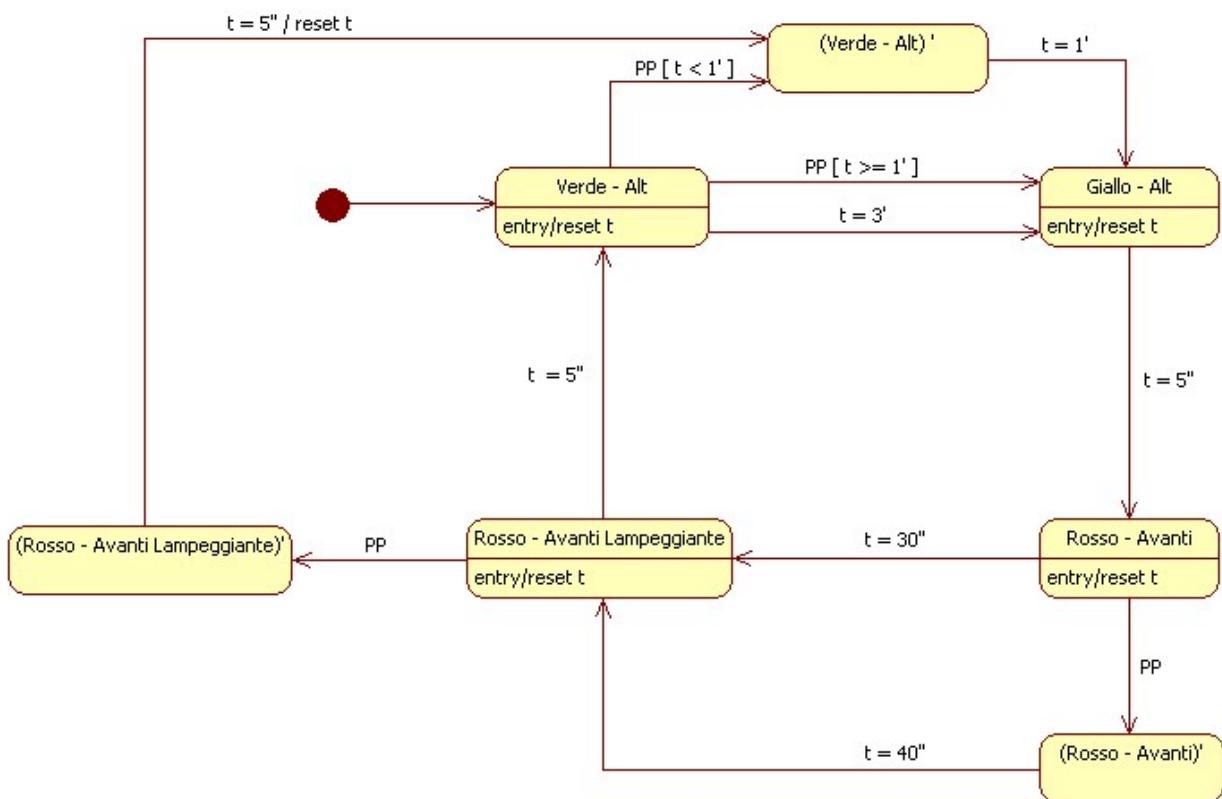


Su una strada è posto un semaforo pedonale, regolato come mostrato in tabella. La durata del verde per le auto è determinata da un pulsante di chiamata pedonale: quando il pulsante viene premuto, se il verde sussiste da almeno 1 minuto avviene immediatamente il passaggio al giallo, in caso contrario il passaggio al giallo avviene solo allo scadere del minuto; se il pulsante non viene premuto, la durata del verde è di 3 minuti. Se il pulsante viene premuto durante il rosso (lato auto) la durata del rosso, normalmente pari a 30 secondi, viene prolungata di 10 secondi, a meno che "avanti" stia già lampeggiando, nel qual caso il verde successivo dura solo 1 minuto.

<i>lato auto</i>	<i>lato pedoni</i>	<i>tempo</i>
verde	alt	1-3 min
giallo	alt	5 sec
rosso	avanti	30-40 sec
rosso	avanti lamp	5 sec

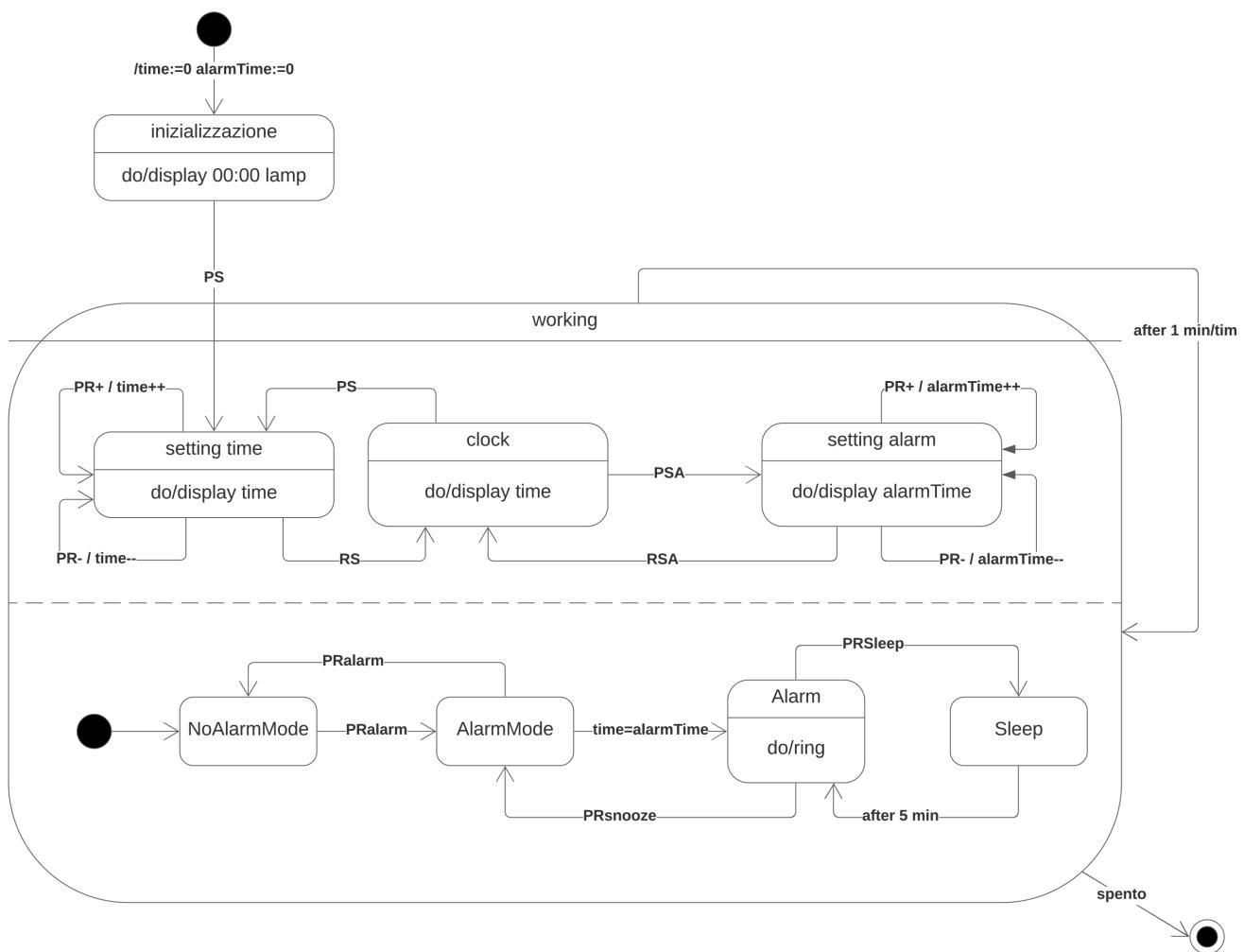
Si disegni il diagramma di stato UML della classe *semaforo*.

Considero la variabile 't' come cronometro, e l'evento PP per 'pulsante premuto'



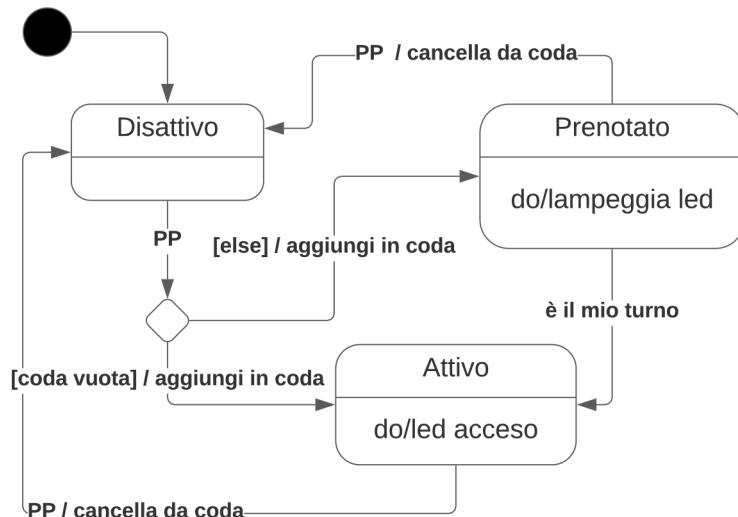
Un orologio digitale da comodino funziona come segue. Quando viene acceso, sul quadrante lampeggia 00:00. Per prima cosa occorre impostare l'ora attuale. Ci sono 3 pulsanti: SET, +, -. Se, tenendo premuto SET, si premono + o -, l'ora sul quadrante viene aumentata o diminuita di un minuto. Rilasciando SET compare l'ora impostata, in formato "OO:MM" con il simbolo ":" lampeggiante. Naturalmente, l'ora può essere reimpostata in qualunque momento. Una volta impostata l'ora, è possibile impostare l'ora della sveglia in modo analogo (pulsanti SET ALARM, +, -). Un pulsante ALARM ON/OFF permette di attivare e disattivare la sveglia. Quando la sveglia suona, si possono utilizzare i pulsanti SNOOZE oppure SLEEP. Il primo tacita definitivamente la sveglia (ma non la disattiva!), il secondo la tacita solo per i 5 minuti successivi. In qualunque momento, l'orologio può essere spento.

Si disegni il diagramma di stato UML della classe *orologio*.



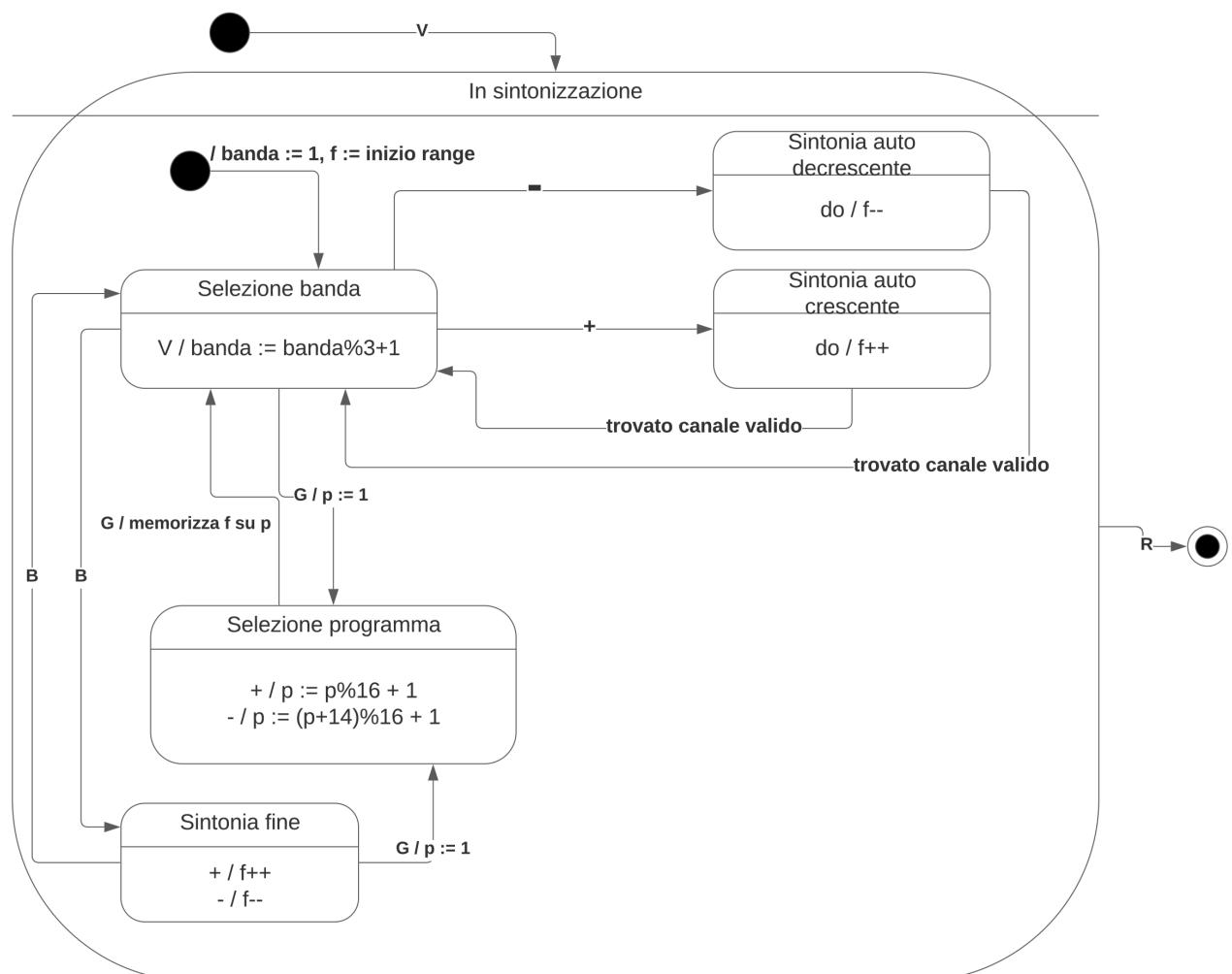
Una sala riunioni è dotata di un sistema di 4 microfoni, ciascuno corrispondente a una postazione e quindi a un oratore. Solo un microfono può essere attivo in ciascun istante. Ogni microfono è dotato di un led colorato e di un tasto P, premendo il quale il relatore prenota il suo intervento (il led comincia a lampeggiare). La prenotazione viene accodata alle eventuali altre. Quando è il proprio turno di parlare (ovvero il relatore precedente spegne il microfono), il led si illumina stabilmente indicando che il microfono è attivo. Evidentemente, se nessun altro microfono è attivo al momento della prenotazione, il microfono viene immediatamente attivato. Al termine del proprio intervento, il relatore spinge di nuovo il tasto P disattivando il suo microfono (il led si spegne). Se P viene premuto dopo aver prenotato un intervento ma prima di iniziare a parlare, la prenotazione viene annullata.

Si modelli il comportamento dei microfoni attraverso un diagramma degli stati UML .



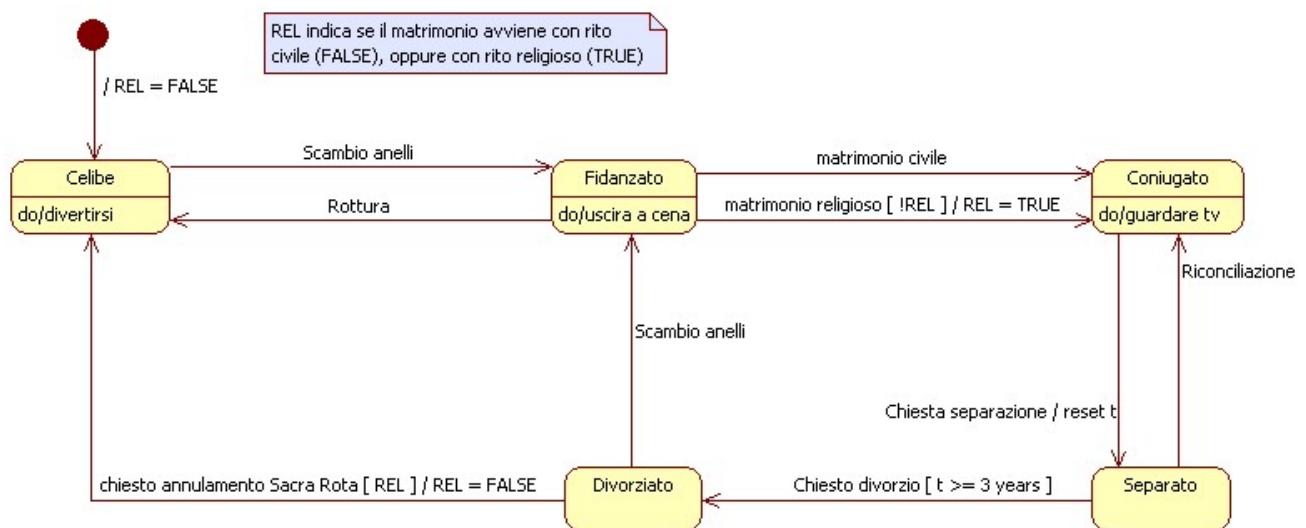
Il telecomando di un televisore è dotato di 2 tasti (+) e (-) e di 4 tasti colorati: verde (V), giallo (G), blu (B), rosso (R), che vengono usati per sintonizzare i canali nel modo seguente. La procedura di sintonizzazione si avvia premendo V; la sintonia si posiziona inizialmente sulla banda 1, passando a rotazione alla 2 e alla 3 a ogni successiva pressione di V. Premendo (+) o (-) si inizia la ricerca automatica (rispettivamente su frequenze via via crescenti e decrescenti); la ricerca si arresta quando viene trovato un canale valido. A questo punto, premendo G si seleziona il programma tv (da 1 a 16) su cui memorizzare il canale: inizialmente viene proposto il programma 1, che può essere modificato premendo (+) e (-); a una nuova pressione di G il canale viene memorizzato sul programma corrente. Premendo B si attiva la sintonia fine: le frequenze vengono scandite manualmente tenendo premuto (+) o (-). Si esce dalla procedura di sintonizzazione premendo R.

Si disegni un diagramma degli stati che modelli il comportamento del sintonizzatore.



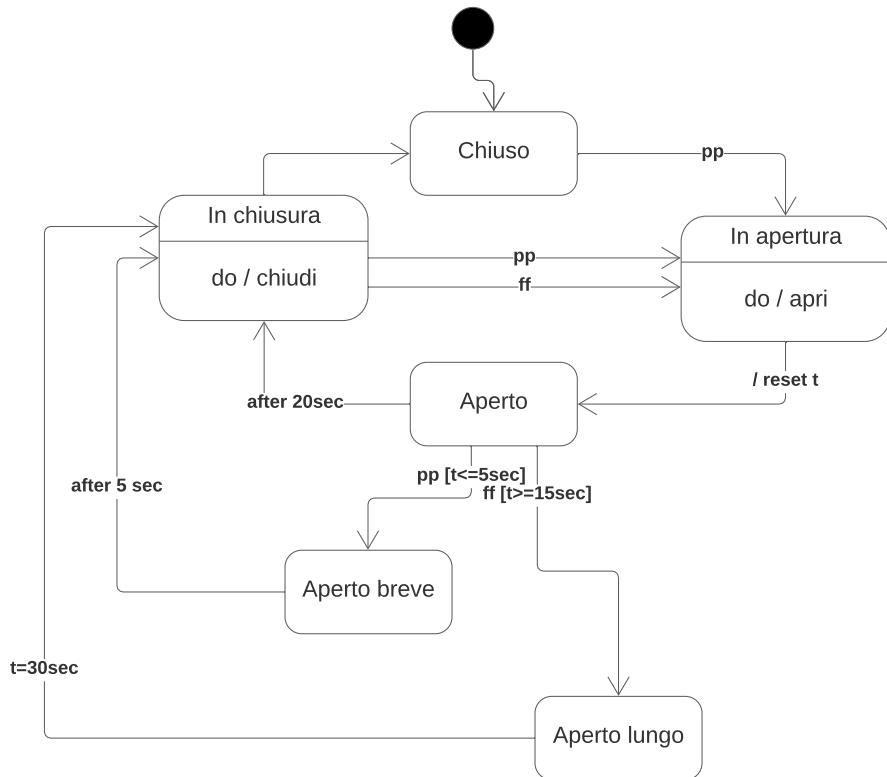
Una persona, alla nascita, è “celibe”. Durante il celibato, la sua principale attività consiste nel divertirsi. Può poi fidanzarsi, compiendo lo scambio degli anelli; durante il fidanzamento, la sua principale attività consiste nell’uscire a cena con il partner. A questo punto la persona può contrarre matrimonio, divenendo “coniugata”. La principale occupazione dei coniugati è guardare la televisione. Se il matrimonio avviene con rito civile, la persona può in qualunque momento chiedere la separazione. Dopo almeno tre anni dalla separazione si può chiedere il divorzio; oppure, in qualunque momento, si può procedere alla riconciliazione ritornando a tutti gli effetti coniugati. Un divorziato può nuovamente fidanzarsi e sposarsi. Se il matrimonio è invece avvenuto con rito religioso, che implica anche il matrimonio civile, in caso di separazione e divorzio si perde la possibilità di risposarsi in chiesa, a meno di ottenere l’annullamento della Sacra Rota. Si tenga presente che l’annullamento della Sacra Rota non ha alcun effetto dal punto di vista dell’unione civile.

Si disegni un diagramma di stato UML che modelli le specifiche illustrate.



Il funzionamento di un automatismo per l'apertura di un cancello può essere descritto come segue. L'utente ha a disposizione un telecomando dotato di un pulsante, premendo il quale il cancello si apre; l'apertura richiede 10 secondi. Il cancello, una volta completamente aperto, normalmente comincia a richiudersi dopo 20 secondi. Se durante la chiusura si preme il pulsante del telecomando, il cancello si riapre. Se, entro 5 secondi da quando il cancello si è completamente aperto, viene premuto il pulsante del telecomando, il cancello si richiude anticipatamente dopo solo 5 secondi. Sulla soglia del cancello è montata una fotocellula. Se il fascio della fotocellula viene interrotto durante la chiusura del cancello, il cancello comincia immediatamente a riaprirsi. Se il fascio viene interrotto durante gli ultimi 5 secondi di apertura del cancello, l'apertura viene prolungata di 10 secondi.

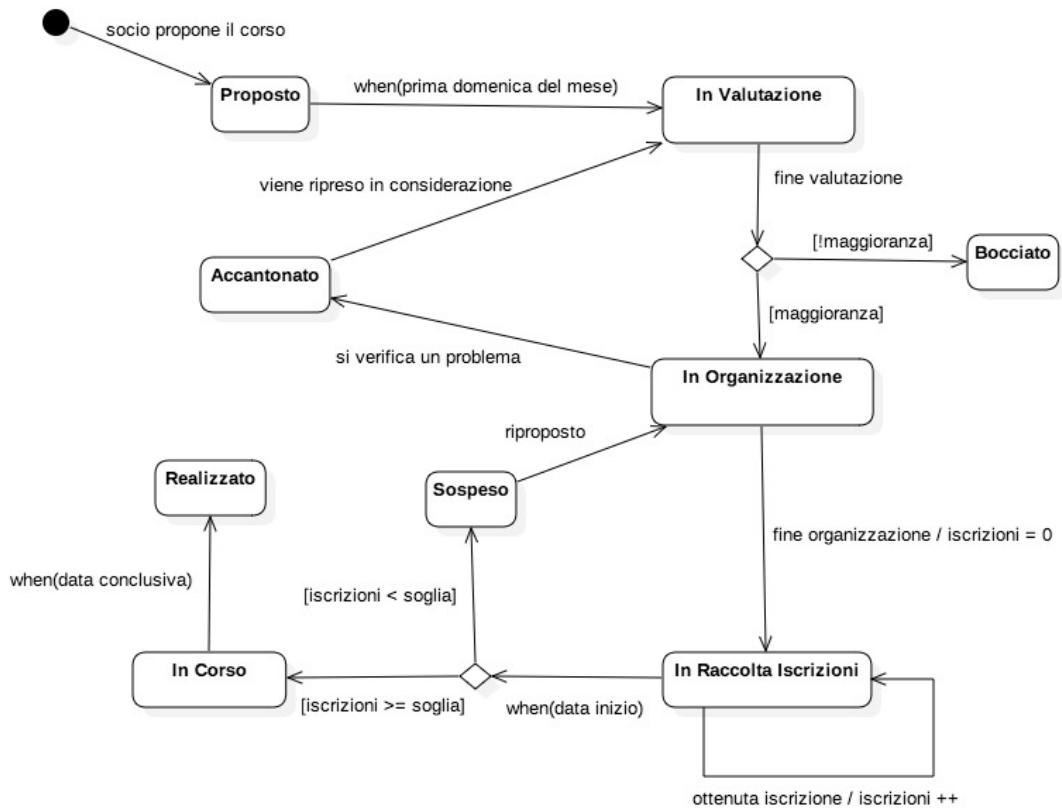
Si disegni un diagramma di stato UML che modelli le specifiche illustrate.



Un circolo nautico vuole realizzare un software per la gestione delle anagrafiche dei soci, dei corsi e delle imbarcazioni. Tutti i soci del circolo hanno un tesserino in cui è registrato il codice identificativo, i dati anagrafici (nome, cognome e data di nascita) e la data di annessione al circolo. Ogni anno vengono rinnovate le cariche sociali (presidente, segretario, consigliere, ...) che devono essere registrate (socio in carica, anno). Alcune cariche (es. consigliere) possono essere rivestite da più soci nello stesso anno ma un singolo socio però non può rivestire più cariche contemporaneamente. Ogni socio può possedere una o più imbarcazioni (del resto ciascuna imbarcazione può avere più proprietari, tutti soci, per ciascuno dei quali viene registrata una quota). I dati registrati per ciascuna imbarcazione sono: matricola, marca e modello, nome, valore, data del varo, dimensioni (larghezza e lunghezza), alimentazione e potenza del motore se è una barca a motore. Il circolo organizza inoltre dei corsi aperti a tutti (non solo soci) per i quali viene registrata la data di inizio e fine, il numero massimo di partecipanti, il tipo (es. "corso di vela base"), la quota di partecipazione e dati anagrafici degli iscritti.

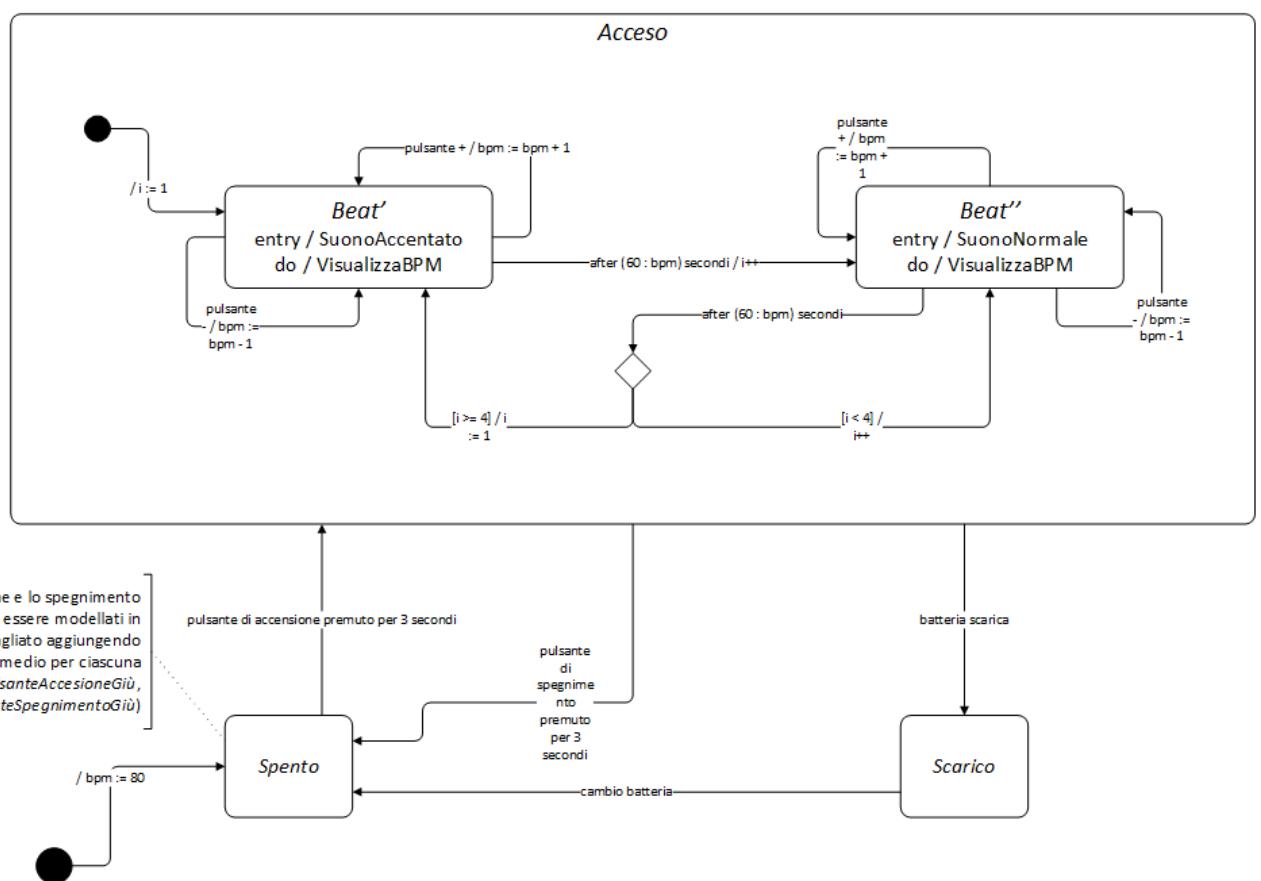
In questo contesto, modellare il diagramma degli stati relativo ai corsi.

I soci del circolo possono proporre in qualsiasi momento un corso da realizzare in un determinato periodo. La prima domenica di ogni mese i soci si riuniscono per discutere ed eventualmente approvare le proposte di corsi raccolte. In particolare, per poter approvare un corso è necessario il parere positivo della maggioranza dei soci. Se il corso viene approvato, il socio proponente ha il compito di finalizzarne l'organizzazione. In caso si verifichino problemi tali per cui il corso non può più essere svolto (per esempio nessun istruttore risulta disponibile), allora questo verrà accantonato per poter essere eventualmente rivalutato e riorganizzato in un secondo momento. Dopo che il corso è stato approvato e organizzato è necessario raccogliere un numero sufficiente di iscritti prima delle date effettive del corso. In caso non vengano raccolte abbastanza iscrizioni, il corso sarà temporaneamente sospeso per poter essere riproposto (senza rivalutazione) in futuro. Se invece il numero di iscrizioni risulta sufficiente, il corso verrà realizzato nelle date concordate.



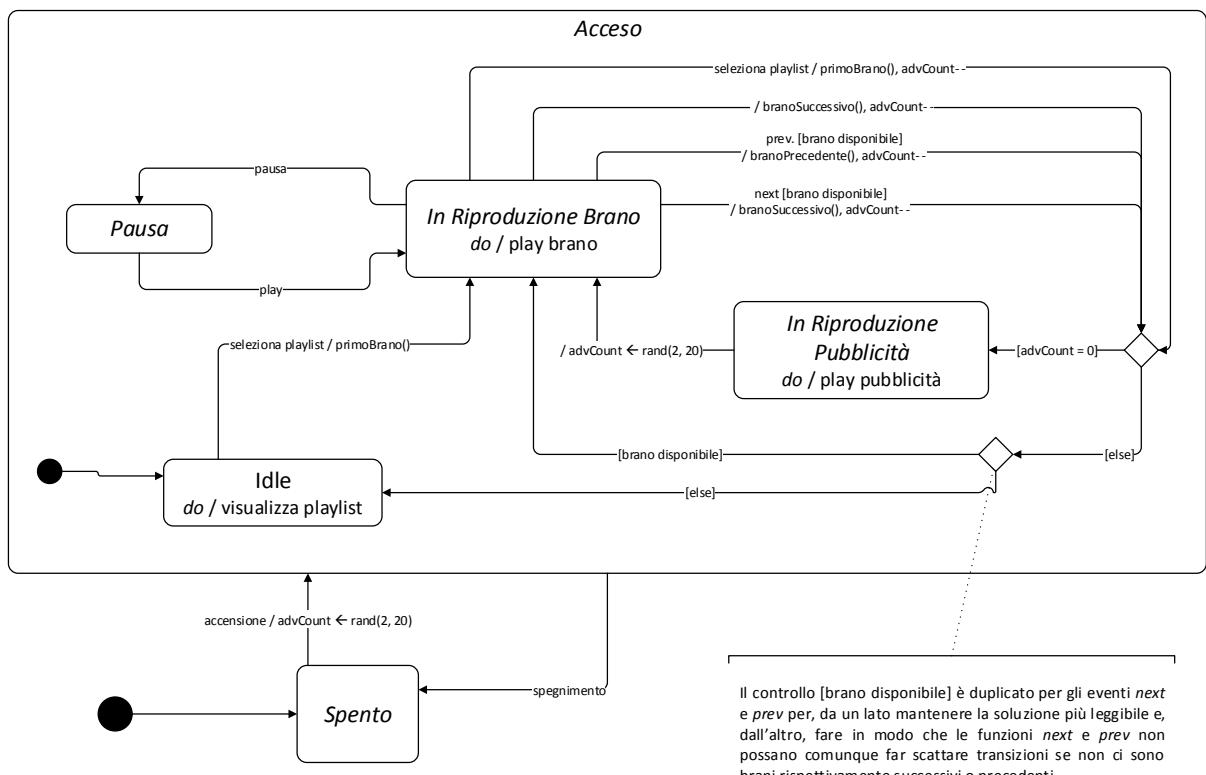
Il metronomo è uno strumento usato in musica per misurare il tempo e scandire la ritmica. Inizialmente il metronomo è spento e può essere acceso tenendo premuto l'apposito pulsante per tre secondi. Una volta acceso, il metronomo emette un suono (beat) con una frequenza detta Beats Per Minute (BPM) il cui valore viene visualizzato tramite apposito display; più alti sono i BPM e minore è il periodo di tempo tra un beat e l'altro. Il valore dei BPM di default (ovvero alla prima accensione) è pari a ottanta ma può essere aumentato o diminuito dall'utente tramite appositi comandi in ogni momento in cui il metronomo è acceso. Anche in caso di spegnimento il valore corrente dei BPM deve essere mantenuto in modo da essere utilizzato all'accensione successiva. Nel caso in cui la batteria che alimenta il metronomo si esaurisca, quest'ultimo cessa di funzionare fino a quando essa non viene sostituita.

Si disegni il diagramma di stato UML della classe Metronomo.

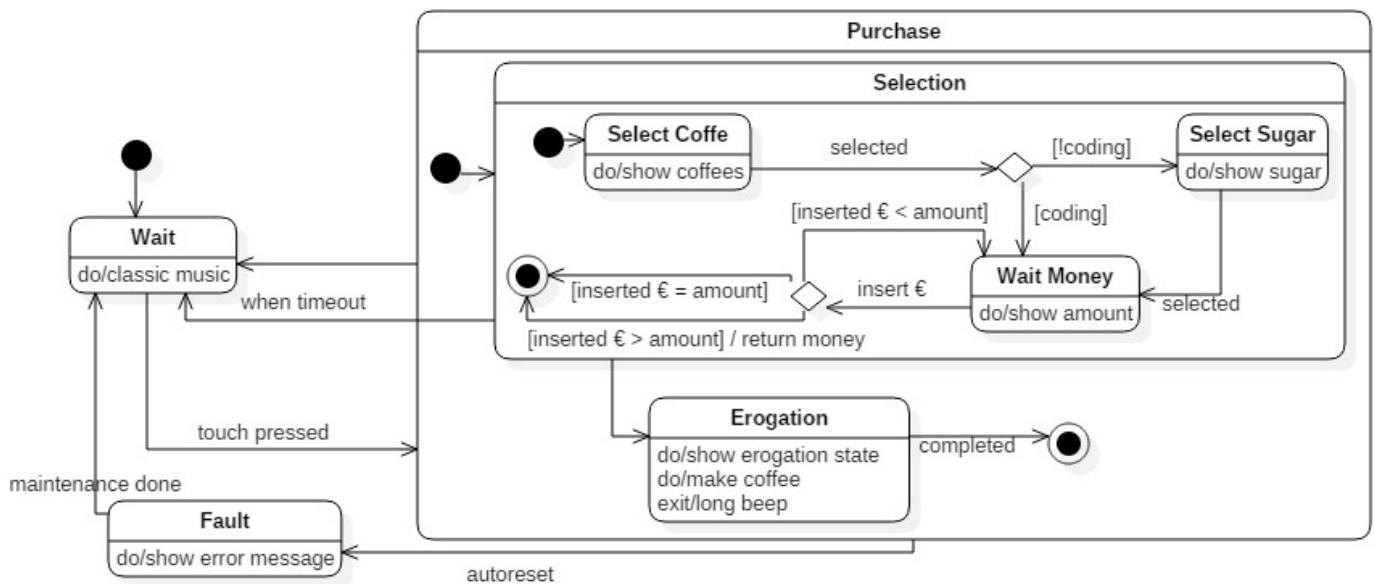


Si modelli tramite *diagramma degli stati* il seguente scenario relativo ad un'applicazione per lo streaming di musica.

Dopo che l'applicazione è stata aperta, per poter iniziare lo streaming è necessario selezionare una delle playlist presenti in catalogo; i brani della playlist selezionata vengono poi riprodotti in ordine. Durante la riproduzione dei brani l'utente può mettere in pausa (e riprendere in seguito l'ascolto), selezionare un'altra playlist o, se presente, passare al brano successivo o precedente tramite apposite funzioni next e previous. L'applicazione è fruibile gratuitamente ma sono previsti intermezzi pubblicitari tra la riproduzione di un brano e l'altro. Nello specifico, un spot pubblicitario viene riprodotto ogni n brani, dove n è un valore random compreso tra 2 e 20; si noti che n viene inizializzato all'accensione e successivamente rivalorizzato dopo ogni riproduzione di pubblicità. Ai fini della gestione degli intermezzi pubblicitari si contano come brani riprodotti anche tutti quei brani che sono stati saltati tramite funzioni next e previous. Si fa notare infine che la pubblicità non può né essere saltata, né essere messa in pausa (l'applicazione può comunque essere chiusa in ogni momento).

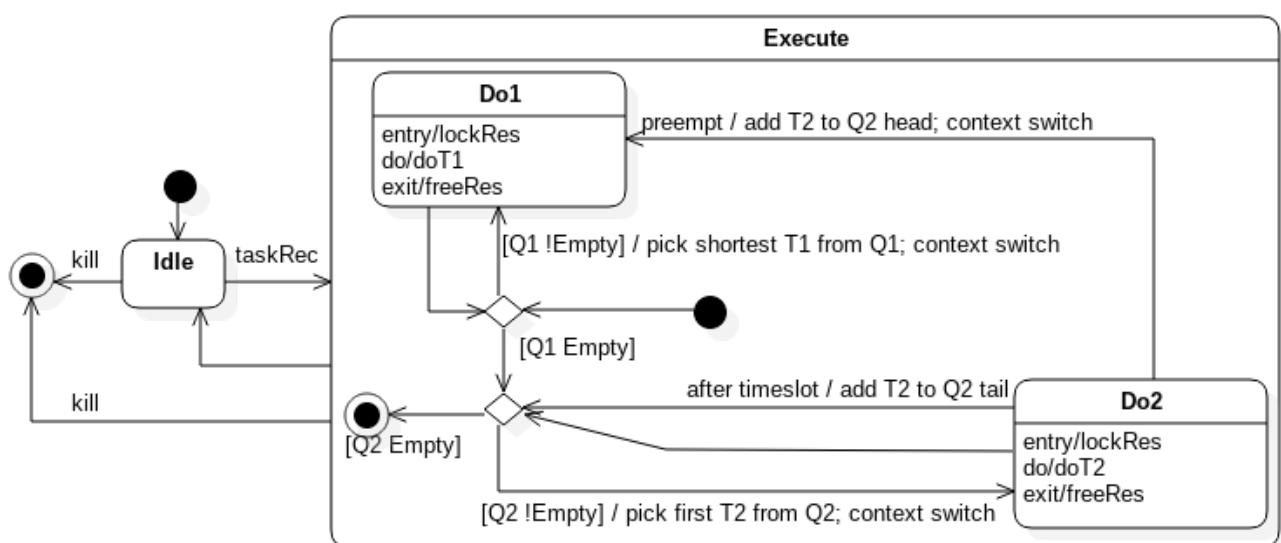


Si vuole modellare con diagramma degli stati il funzionamento di una macchinetta per l'erogazione di caffè dotata di touch screen e prodotta dall'azienda Kaffee. Ad accensione avvenuta, la macchinetta entra in attesa e suona una leggiadra melodia classica. Quando un utente preme il touch screen, la macchinetta interrompe la musica ed entra in selezione. In questa fase, all'utente è mostrata la lista dei caffè disponibili. Tra i possibili tipi di caffè selezionabili, il "coding coffee" è un particolare caffè che non richiede l'aggiunta di zucchero. Se il caffè selezionato non è di tipo coding coffee, all'utente è mostrata la schermata di selezione zucchero. Una volta scelta la quantità di zucchero, la macchinetta visualizza l'importo del caffè e attende che l'utente completi il pagamento. Se l'utente seleziona il coding coffee, la macchinetta salta la selezione dello zucchero, visualizza l'importo del caffè e attende che l'utente inserisca l'importo. Se l'importo inserito è maggiore di quello richiesto, la macchinetta restituisce il resto. Saldato l'importo, la fase di selezione si conclude e la macchinetta eroga il caffè. Durante l'erogazione la macchinetta mostra lo stato dell'erogazione. Al termine dell'erogazione, la macchinetta emette un beep e ritorna in stato di attesa. La fase di selezione è gestita con un timer, se l'utente non completa la selezione prima dello scadere del timer, la macchinetta ritorna in stato di attesa. Infine, la macchinetta è spesso soggetta a guasti. In qualsiasi momento delle due fasi la macchinetta può auto-resettersi. In tal caso la macchinetta entra in stato di guasto, mostra un messaggio di errore, e attende che un operatore dell'azienda Kaffee esegua la manutenzione. A manutenzione avvenuta, la macchinetta torna in stato di attesa.

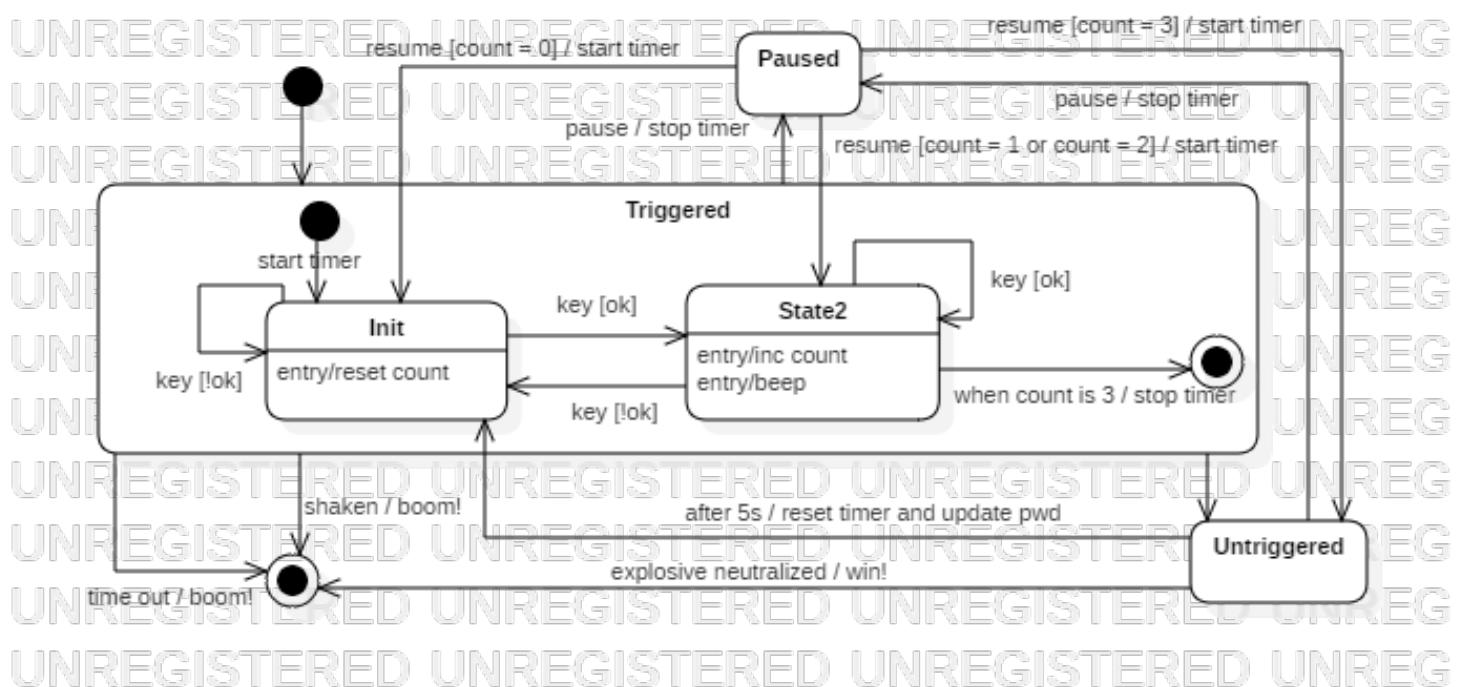


Si modelli tramite *diagramma degli stati UML* il seguente scenario che modella il funzionamento di uno scheduler.

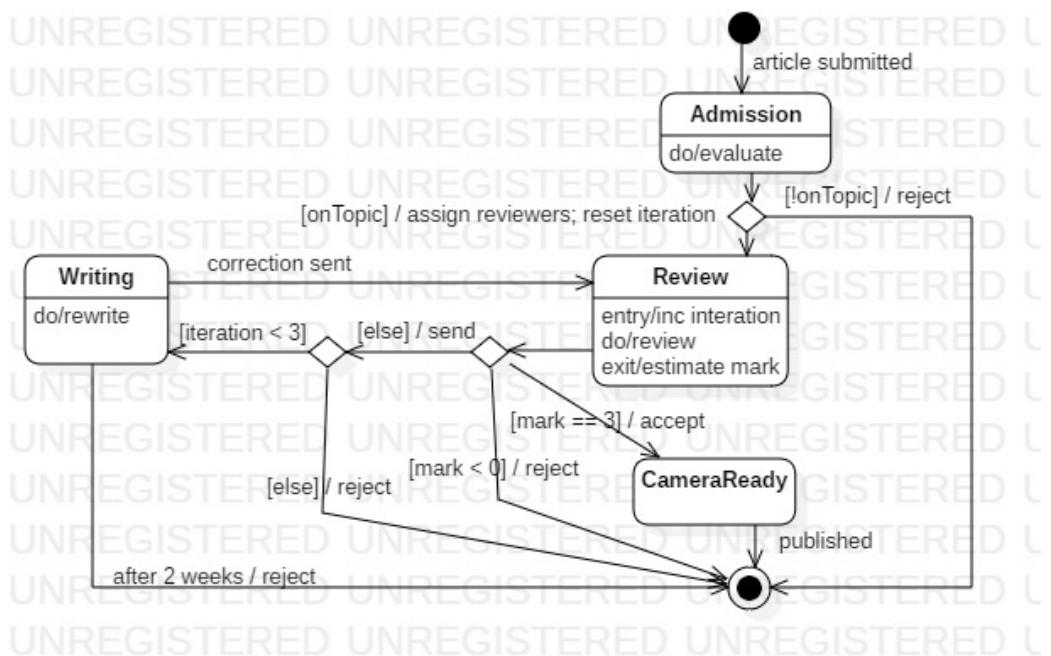
Lo scheduler ha il compito di eseguire dei task accodati in due liste ($Q1$ e $Q2$) in base alla priorità: i task appartenenti a $Q1$ sono eseguiti sempre prima dei task presenti in $Q2$. Una volta acceso, lo scheduler attende nuovi task da eseguire (idle). Ricevuti uno o più task, lo scheduler entra nell'articolata fase di esecuzione. In prima istanza, lo scheduler verifica la presenza di task in $Q1$. Se ci sono task ($Q1$ non è vuota), lo scheduler preleva il task più corto ($T1$), esegue un'operazione di "context switch" ed esegue $T1$. Altrimenti lo scheduler verifica la presenza di task in $Q2$. Se ci sono task ($Q2$ non è vuota), lo scheduler preleva il primo elemento della lista ($T2$), esegue un'operazione di "context switch" e procede con l'esecuzione di $T2$, altrimenti termina la fase di esecuzione e ritorna in attesa di nuovi task. L'esecuzione di $T1$ richiede di: ottenere il lock sulle risorse necessarie, eseguire il task, e rilasciare i lock precedentemente ottenuti. L'esecuzione di $T2$ prevede gli stessi passi di $T1$, tuttavia presenta due differenze sostanziali: se in questa fase lo scheduler riceve un segnale "preempt" (cioè lo scheduler riceve un task $T1$ durante l'esecuzione di $T2$), lo scheduler sospende $T2$, lo inserisce in testa a $Q2$, esegue un "context switch" e procede con l'esecuzione del nuovo task $T1$; se il tempo di esecuzione ha raggiunto la soglia "timeslot", lo scheduler accoda $T2$ in coda a $Q2$ e procede con la selezione del prossimo elemento in $Q2$. In qualsiasi momento lo scheduler può essere terminato con segnale "kill".



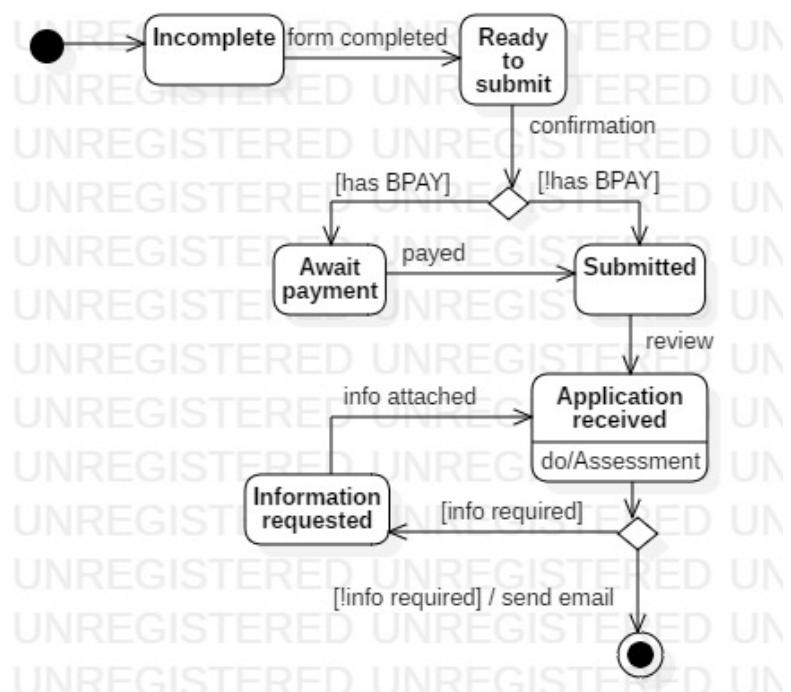
Si vuole modellare con un diagramma a stati il funzionamento dell'applicazione B00m!, gioco alla moda per smartphone in cui il giocatore deve disinnescare un ordigno senza farlo esplodere. All'avvio della partita l'ordigno è innescato. Attenzione: quando l'ordigno è innescato lo scadere del timeout e le vibrazioni dello smartphone lo fanno esplodere! Il giocatore deve indovinare una password di 3 caratteri per disinnescare l'ordigno. A ogni inserimento di un carattere corretto il gioco emette un beep ed è possibile procedere con l'inserimento del carattere successivo. Se l'inserimento è errato, i caratteri inseriti sono resettati e il giocatore deve ricominciare dall'inserimento del primo carattere (ma la password non cambia). Se il terzo carattere inserito è corretto allora il timeout si ferma e l'ordigno è disinnescato. Una volta disinnescato l'ordigno, l'utente ha 5 secondi per neutralizzarlo. Se l'ordigno è neutralizzato la partita termina con vittoria. In caso contrario, allo scadere dei 5 secondi, l'ordigno entra nuovamente in stato di innescos, il timeout è ripristinato e la password è resettata. Il giocatore deve quindi ricominciare con l'inserimento della nuova password. In ogni momento il gioco può essere sospeso e ripreso nell'esatto punto in cui l'utente l'ha sospeso.



Si vuole modellare con un diagramma degli stati la pubblicazione di articoli presso riviste scientifiche. La procedura inizia con l'invio dell'articolo alla rivista scientifica da parte degli autori. Quando l'articolo è ricevuto dalla rivista, esso entra in fase di ammissione. In questa fase l'editor della rivista valuta preliminarmente l'articolo. Se l'articolo non rispetta il topic della rivista esso viene rifiutato e la procedura si conclude. Se l'articolo rispetta il topic, l'editor lo assegna a 3 revisori e l'articolo entra in fase di revisione. Ogni revisore assegna un voto all'articolo (da -3 a +3). Quando i revisori concludono la revisione, il sistema calcola il voto medio. Se tutti gli autori hanno assegnato il voto massimo (cioè 3) all'articolo, l'articolo è accettato ed entra in fase di camera ready, la quale termina quando la rivista viene pubblicata. Se il voto medio è insufficiente (cioè < 0), l'articolo viene rifiutato e la procedura si conclude. Altrimenti, l'esito delle revisioni è inviato agli autori i quali hanno 2 settimane per preparare una nuova versione dell'articolo e inviarla alla rivista. Se la nuova versione non viene inviata entro 2 settimane, l'articolo viene rifiutato e la procedura si conclude. Quando la nuova versione dell'articolo è ricevuta dalla rivista, essa entra direttamente in fase di revisione e la procedura riprende come descritto in precedenza per un massimo di 2 volte. Alla terza mancata accettazione l'articolo viene automaticamente rifiutato.



Si vuole modellare con diagramma degli stati il ciclo di vita della richiesta di un visto australiano. La richiesta di visto si basa sul completamento e sulla validazione di una form con i dati dell'utente richiedente. Quando un utente inizia una richiesta di visto, questa è identificata come incompleta. Una volta completata la form, la richiesta entra in stato di pronto. Quando l'utente ne conferma l'invio, la richiesta viene inviata al dipartimento degli affari interni, previo pagamento di una tassa governativa prevista solo per alcune tipologie di visto (ossia quelle soggette a clausola "BPAY"). In particolare, se la richiesta è soggetta a clausola "BPAY" essa rimane in attesa di pagamento fino a quando l'utente non salda il costo della pratica. Quando la richiesta è presa in carico dal dipartimento degli affari interni, essa è identificata come ricevuta. Al termine di questa fase, la richiesta può essere finalizzata (la decisione di concessione/rifiuto è notificata via e-mail all'utente) o informazioni aggiuntive possono essere richieste all'utente. In questa evenienza, una volta forniti i dati richiesti questi devono essere validati. Se le informazioni fornite sono sufficienti, la richiesta è finalizzata e l'esito è comunicato all'utente, altrimenti ulteriori informazioni sono richieste.

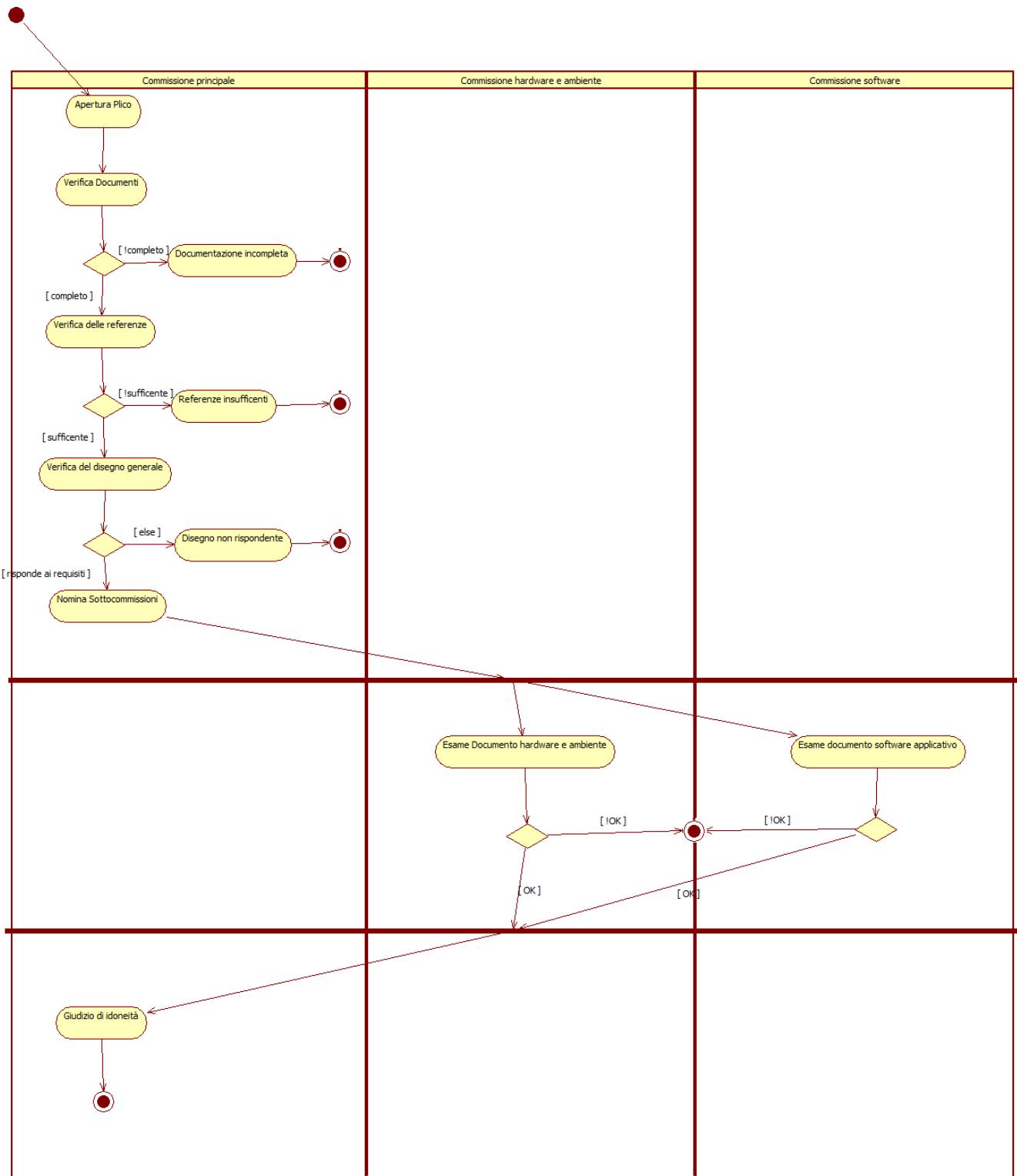


Diagrammi di attività

La commissione di valutazione relativa alla idoneità di un progetto di un sistema informativo di un'azienda manifatturiera procede come segue:

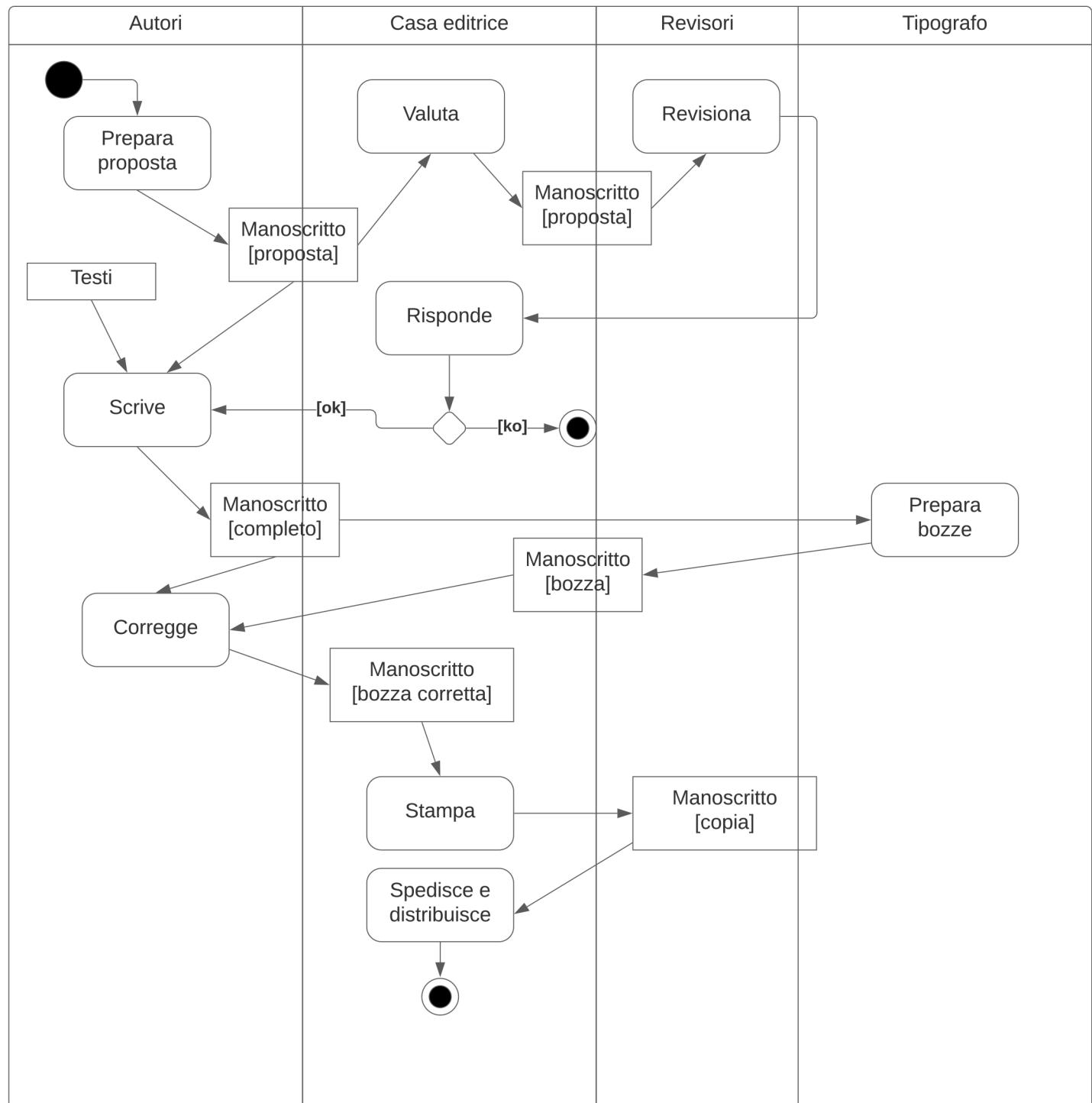
- a) *apre il plico contenente i documenti di progetto e verifica se esistono tutti i documenti richiesti; procede alla fase successiva solo se la verifica ha avuto esito positivo altrimenti emette giudizio “non idoneo: documenti non conformi”;*
- b) *verifica il documento di referenze allegato; se le esperienze maturate dal proponente nel settore specifico non sono sufficienti emette giudizio “non idoneo: esperienze non significative” altrimenti procede al punto c;*
- c) *esamina il documento che contiene il disegno generale; in caso di non rispondenza ai requisiti emette giudizio “non idoneo: disegno generale non rispondente” altrimenti procede al punto d;*
- d) *nomina due sottocommissioni tecniche che procedono quindi in parallelo; la prima esamina il documento relativo all'hardware, al software di base e di ambiente, la seconda che esamina il documento relativo al software applicativo; il progetto viene ritenuto idoneo se e solo se entrambe le commissioni emettono un giudizio di idoneità sul punto di loro competenza.*

Si disegni un diagramma di attività UML che modelli le specifiche illustrate.

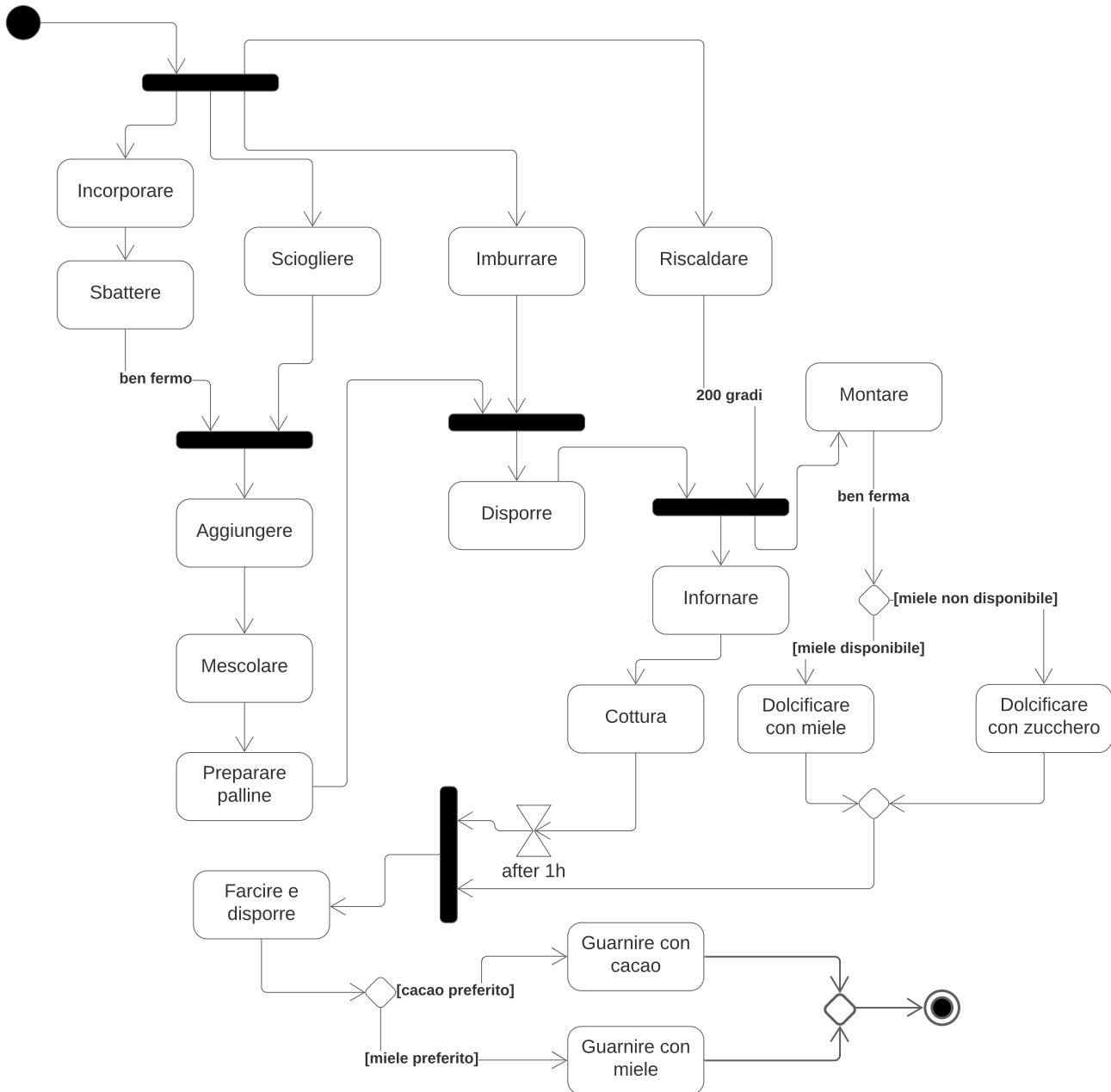


Il processo di scrittura e pubblicazione di un libro scientifico è piuttosto complesso. Inizialmente, gli autori preparano una proposta, consistente nell'indice dell'intera trattazione e nella bozza di un paio di capitoli; inviano poi il materiale predisposto all'attenzione della casa editrice prescelta. La casa editrice valuta il progetto, avvalendosi della consulenza di revisori esterni (alcuni dei quali suggeriti dagli autori stessi); se le revisioni sono positive, il progetto viene accettato. A questo punto, gli autori iniziano la fase di scrittura vera e propria, che in genere comporta la consultazione di una gran quantità di altri testi specializzati e articoli. Una volta terminata l'opera, gli autori inviano il manoscritto alla casa editrice. Il tipografo della casa editrice prepara le bozze del lavoro e le spedisce agli autori per la correzione. L'attività di correzione delle bozze, svolta dagli autori, consiste nel raffronto tra il manoscritto originale e la bozza preparata dal tipografo; ogni inconsistenza o errore tipografico viene scrupolosamente annotato basandosi su un vocabolario di segni grafici appositi. Le bozze corrette vengono infine rispedite alla casa editrice, che stampa la copie. Alcune copie omaggio vengono spedite a esperti del settore e, naturalmente, agli autori stessi; le altre copie vengono distribuite ai negozi per la vendita.

Si modellino le specifiche illustrate attraverso un diagramma di attività UML.



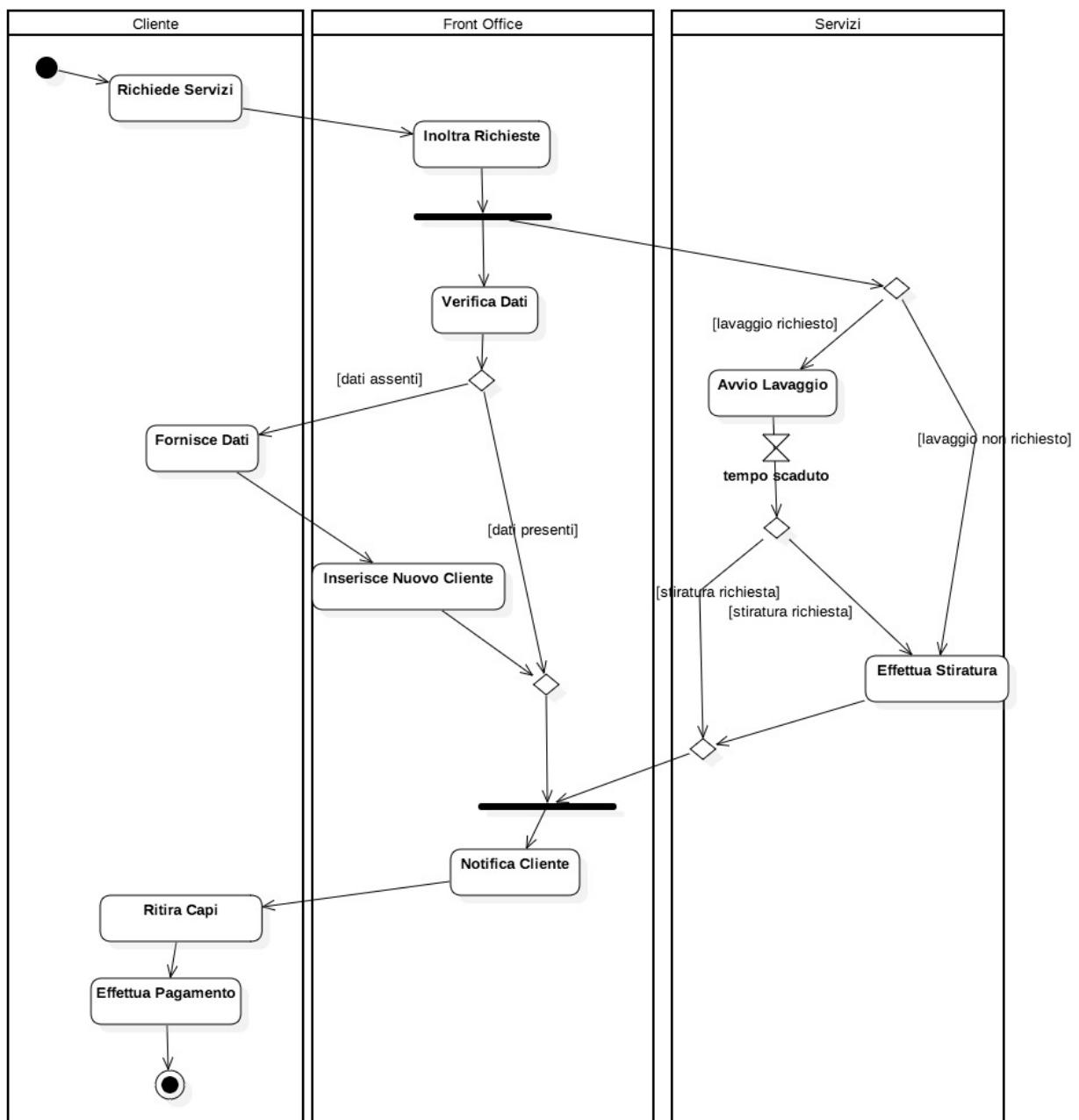
Ricetta dei Bignè alla Rumbaugh. Ingredienti: uova, farina, marsala, lievito, zucchero, cacao, miele, panna. Incorporare 3 uova a 1 hg di farina e sbattere per alcuni minuti. Quando il composto è ben fermo, aggiungere 1 dl di marsala in cui avrete precedentemente fatto sciogliere 2 cucchiai di zucchero e 1 bustina di lievito. Dopo aver mescolato, preparare tante palline di circa 3 cm di diametro e disponetele su una teglia che avrete precedentemente imburrato. Inforntate in forno già caldo a 200° per 1 ora. Nel frattempo, montate la panna ben ferma e dolcificate con 1 cucchiaino di miele (in alternativa, potete usare 1 cucchiaino di zucchero). Quando i bignè sono cotti, farciteli con la panna e disponeteli sul piatto di portata. A scelta, guarnite con cacao oppure con miele.



La lavanderia Arcobaleno vuole realizzare un sistema software per la gestione dei clienti, dei capi da lavare e del tariffario dei servizi offerti. La lavanderia offre i servizi di lavaggio, in acqua o a secco, e stiratura. Il prezzo dei servizi offerti dipende dalla tipologia di capo da trattare (es. pantalone, gonna, giacca, ...). Per alcune tipologie di capi (es. tappeti, tende) il prezzo dei servizi dipende anche dalla fascia di peso. Le fasce, descritte da un peso minimo e un peso massimo (es. da 3 a 5 kg), devono essere registrate nel sistema. I clienti della lavanderia sono memorizzati in un archivio; per ciascuno si registrano un codice identificativo, il cognome, il nome, l'indirizzo e un recapito telefonico. Ogni volta che un cliente porta dei capi in lavanderia, nel sistema devono essere memorizzati la data, il costo totale e la data di restituzione prevista. Inoltre per ogni capo da trattare, identificato da un codice univoco, si memorizzano una descrizione, l'eventuale peso e i servizi richiesti. Naturalmente per uno stesso capo possono essere richiesti più servizi (es. lavaggio a secco e stiratura).

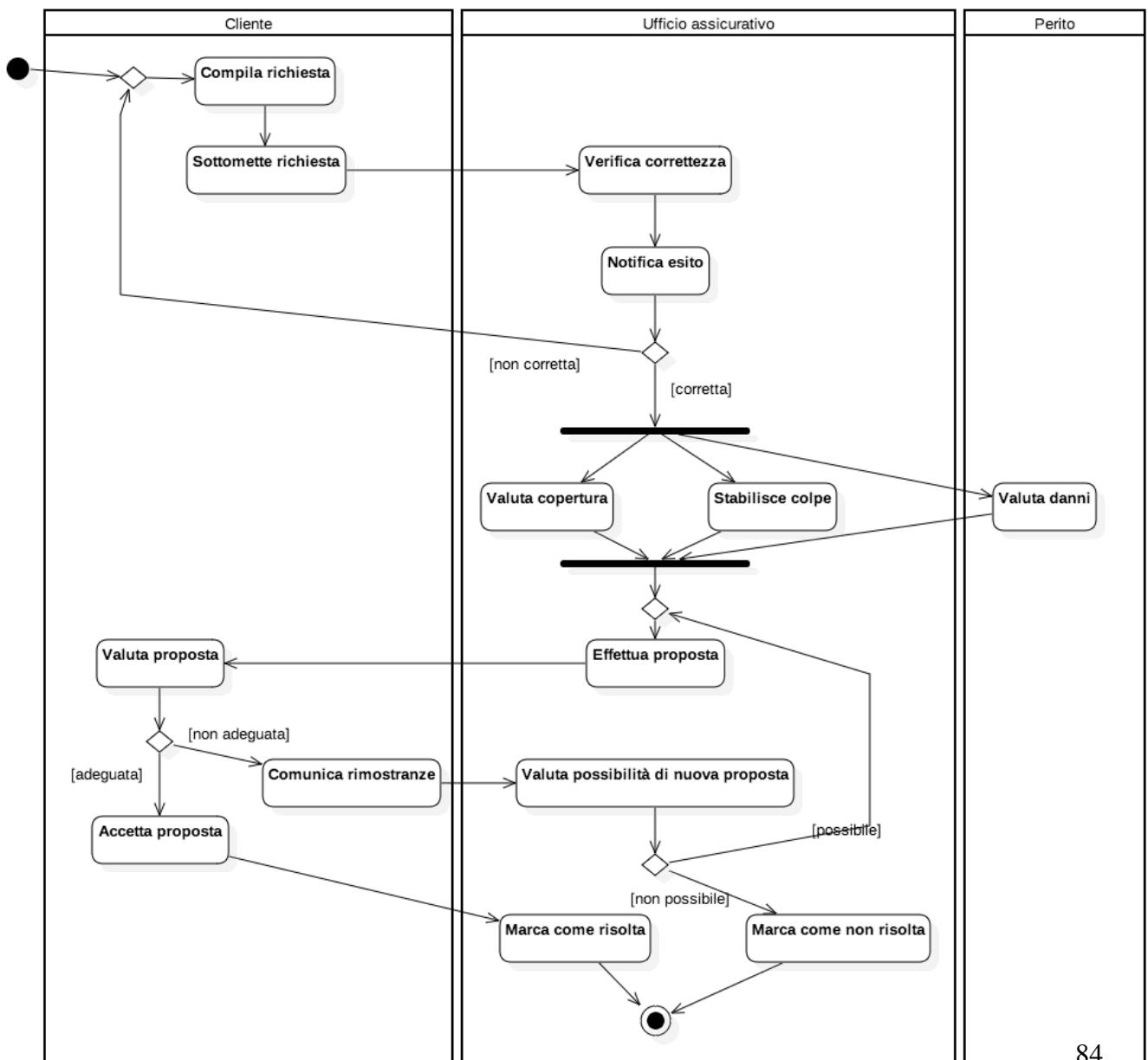
In questo contesto, modellare tramite diagramma delle attività il seguente scenario in cui viene descritta la richiesta di servizi da parte di un cliente.

Dopo aver ricevuto i capi e i dettagli relativi ai servizi richiesti (lavaggio a secco, stiratura, etc.), l'addetto al front office passa il tutto al personale designato e controlla se i dati relativi al cliente sono già presenti in archivio. In caso negativo, il cliente dovrà fornire i propri dati che in seguito verranno registrati. Nel frattempo, il personale addetto ai servizi si occuperà di eseguire le richieste ricevute. In particolare avvierà il processo di lavaggio e allo scadere del tempo necessario effettuerà la stiratura (notare che il lavaggio o la stiratura potrebbero non essere richiesti). Una volta terminate le precedenti operazioni verrà inviata una notifica al cliente, il quale dovrà poi ritirare i propri capi ed effettuare il dovuto pagamento.

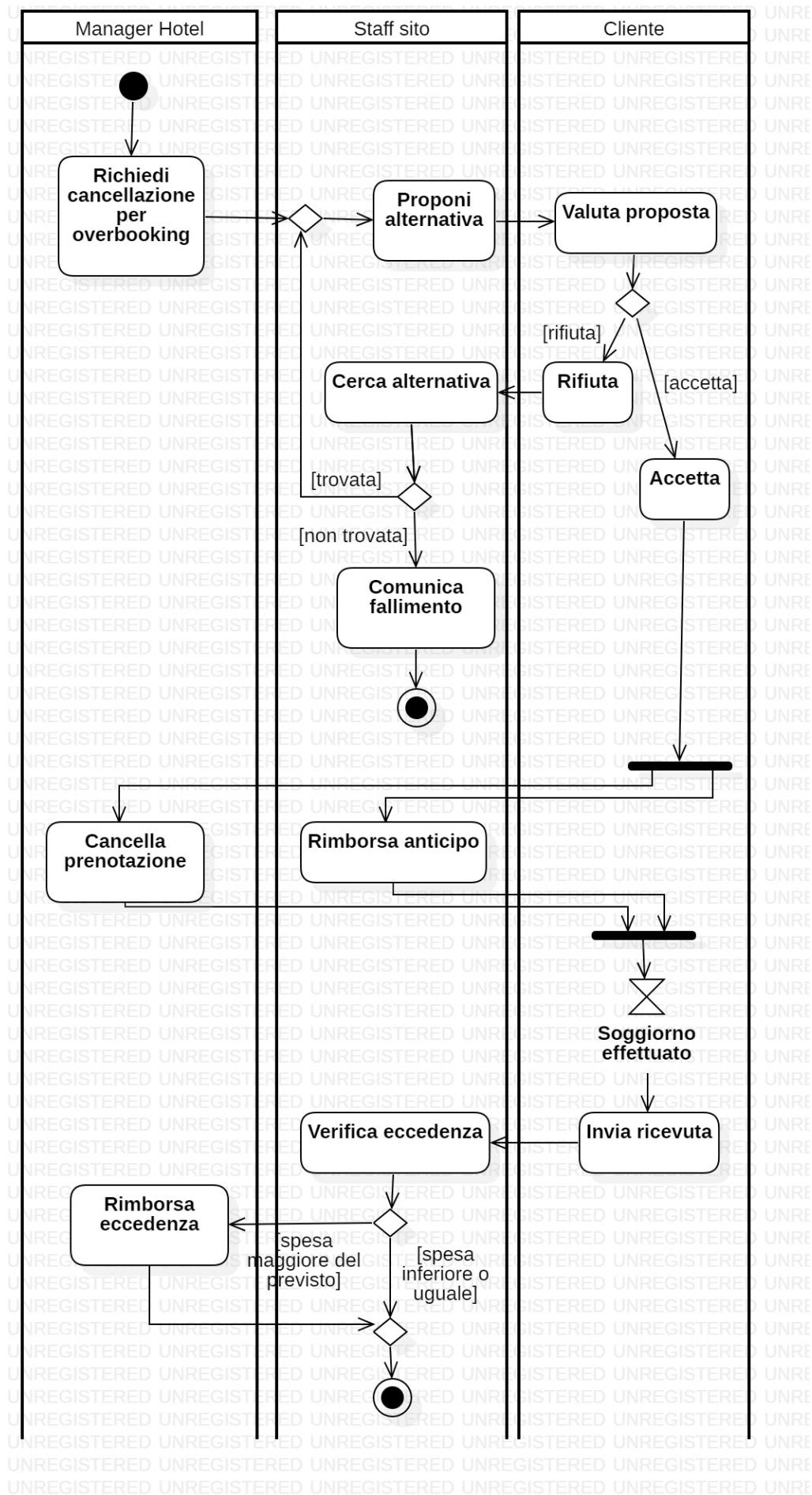


Modellare tramite diagramma di attività il seguente scenario in cui viene descritta la sottomissione e la gestione di un reclamo assicurativo relativo ad un sinistro stradale.

Per prima cosa il cliente della compagnia assicurativa deve compilare e sottomettere un modulo di richiesta in cui vengono descritti i dettagli del sinistro. In seguito l'ufficio assicurativo prende in carico la richiesta e ne verifica la correttezza. L'esito della verifica viene poi notificato al cliente che, in caso di errori di compilazione, dovrà effettuare una nuova sottomissione. Se invece il reclamo risulta formalmente corretto, l'ufficio provvede da un lato a valutare la copertura assicurativa del cliente, dall'altro a stabilire le colpe del sinistro. Nel frattempo un perito ha il compito di valutare l'entità dei danni. Al termine di queste operazioni l'ufficio effettua una proposta di risarcimento che il cliente valuterà se accettare o meno. In caso di accettazione il reclamo viene marcato come risolto e viene effettuato il versamento come stabilito nella proposta. In caso di rifiuto il cliente deve comunicare le proprie rimostranze all'ufficio assicurativo che poi valuterà se è possibile o meno fare una nuova proposta (ci possono essere più iterazioni di proposta e rimostranze). Nel caso in cui l'ufficio non voglia fare ulteriori proposte il reclamo viene marcato come irrisolto.

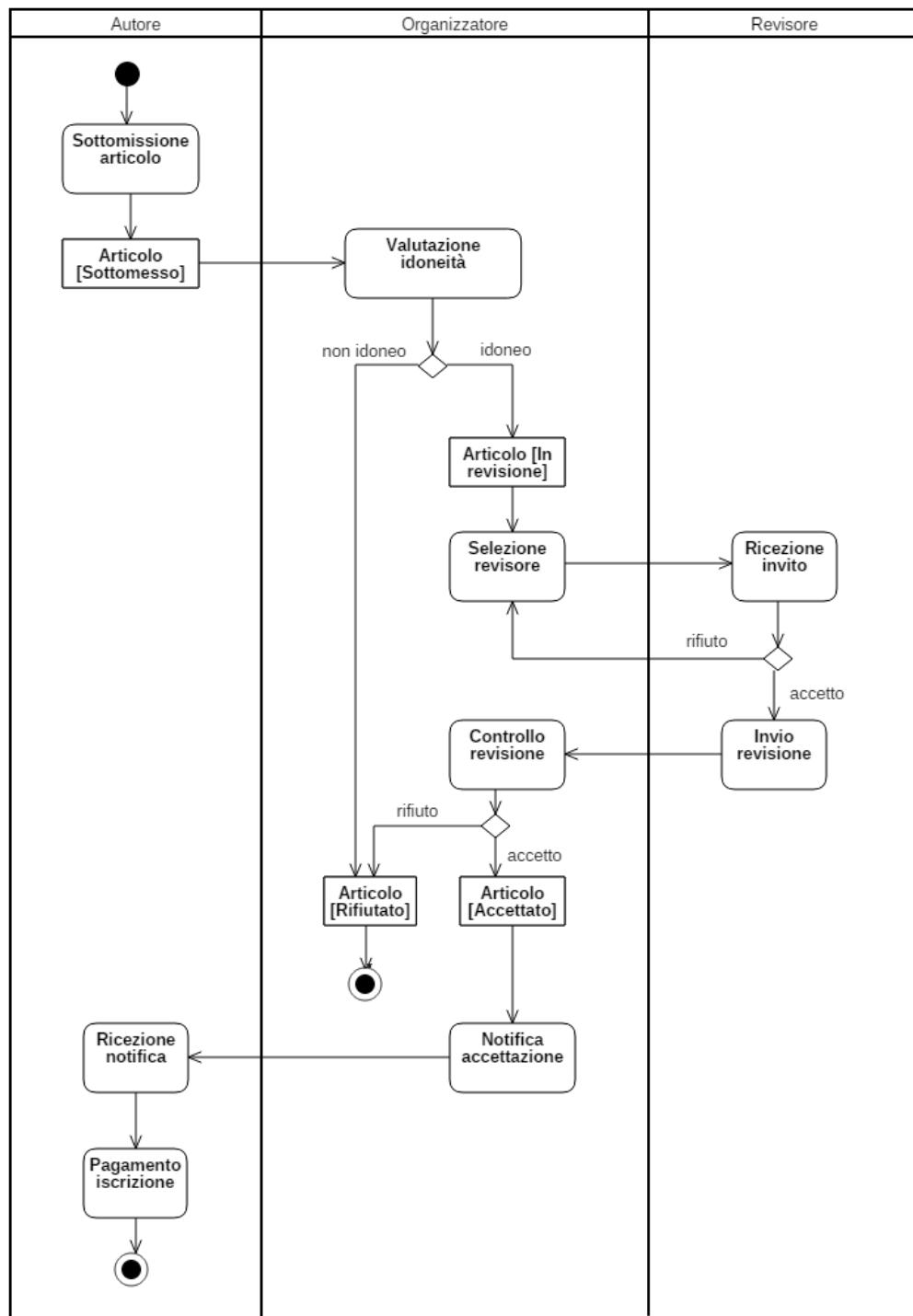


Si vuole modellare con un diagramma di attività la casistica in cui un hotel vada in overbooking sul sito prenotando.com. La procedura viene avviata dal manager dell'hotel, il quale richiede allo staff del sito di cancellare una determinata prenotazione a causa di overbooking. Ricevuta la richiesta, lo staff contatta lo sfortunato cliente proponendogli una soluzione alternativa. Il cliente, una volta presa visione dell'offerta, può decidere se accettarla o meno. In caso di rifiuto, lo staff del sito cerca un'alternativa e sottopone al cliente una nuova proposta; il processo è iterativo e si interrompe quando il cliente accetta una proposta o quando lo staff non riesce a trovare nuove alternative. In quest'ultimo caso, la procedura viene conclusa dallo staff, il quale comunica al manager dell'hotel di non essere riuscito a cancellare la prenotazione (sarà poi cura del manager cercare un modo di risolvere la situazione). Qualora il cliente accetti una proposta alternativa, lo staff cancella la prenotazione e, allo stesso tempo, il manager provvede a rimborsare il cliente di quanto già pagato. Una volta effettuato il soggiorno presso la struttura alternativa, il cliente invierà infine la fattura del soggiorno allo staff del sito; in caso la spesa risulti maggiore rispetto a quella inizialmente prevista, lo staff richiedere al manager dell'hotel di rimborsare al cliente anche le spese in eccesso

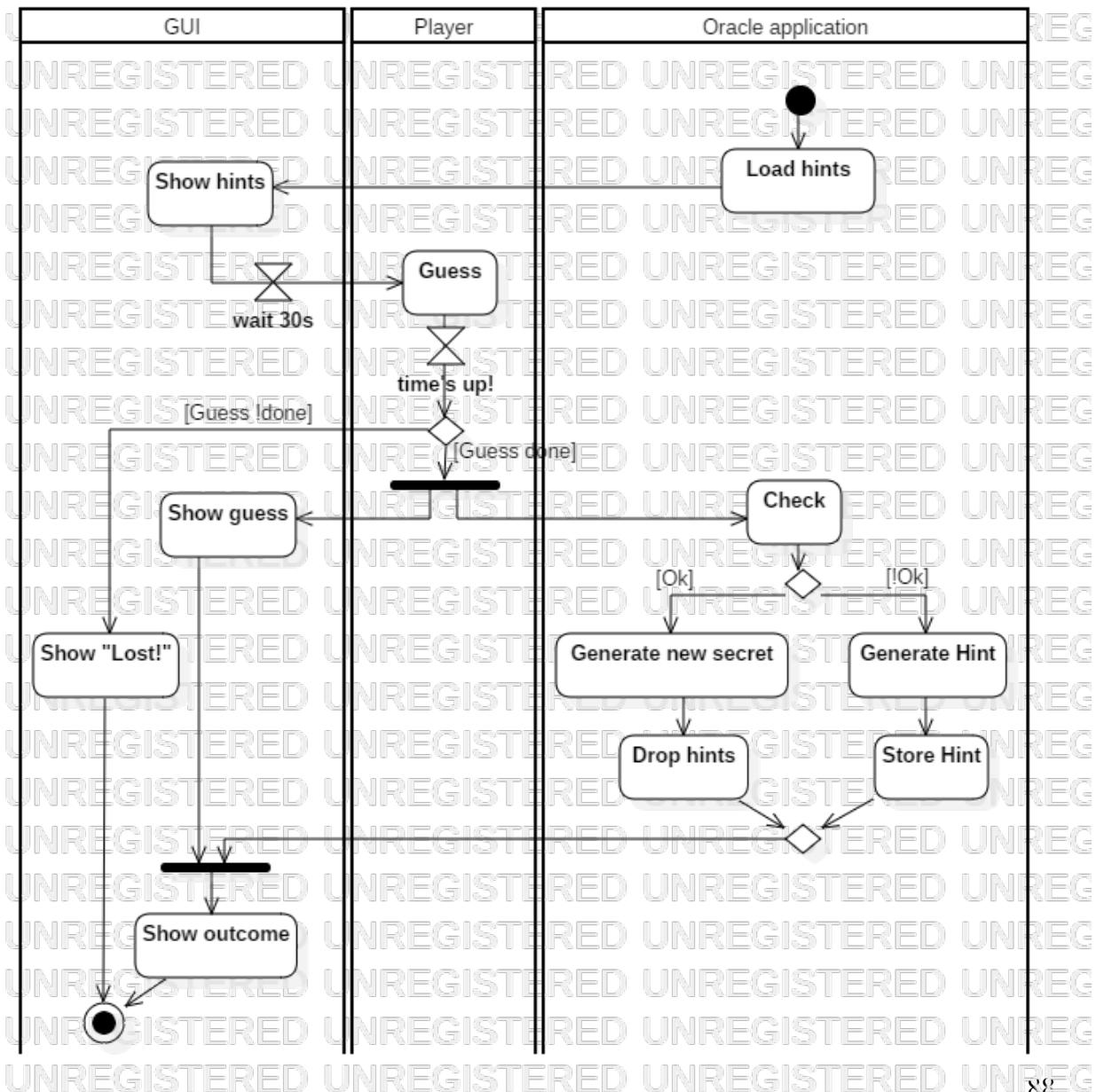


Modellare tramite *diagramma di attività UML* il seguente scenario, in cui viene descritta la sottomissione di un articolo scientifico all'organizzatore di una conferenza. Nel diagramma si modelli anche lo stato dell'articolo attraverso il processo.

Previa autenticazione presso il sito della conferenza, l'autore invia l'articolo che vuole sottomettere. In prima istanza, l'organizzatore verifica che l'argomento dell'articolo sia coerente con l'ambito della conferenza. In caso negativo, l'articolo viene semplicemente rifiutato; in caso positivo, l'articolo entra nel processo di revisione. L'organizzatore individua quindi un revisore qualificato da incaricare per la valutazione dell'articolo. Ricevuta notifica, il revisore decide se accettare o meno l'incarico (in caso contrario, l'organizzatore deve individuare un nuovo revisore). Il revisore che ha accettato l'incarico invia la revisione all'organizzatore, il quale decide se accettare o rifiutare l'articolo. In caso di accettazione, l'autore viene notificato e conclude l'iter pagando la tassa di iscrizione alla conferenza.

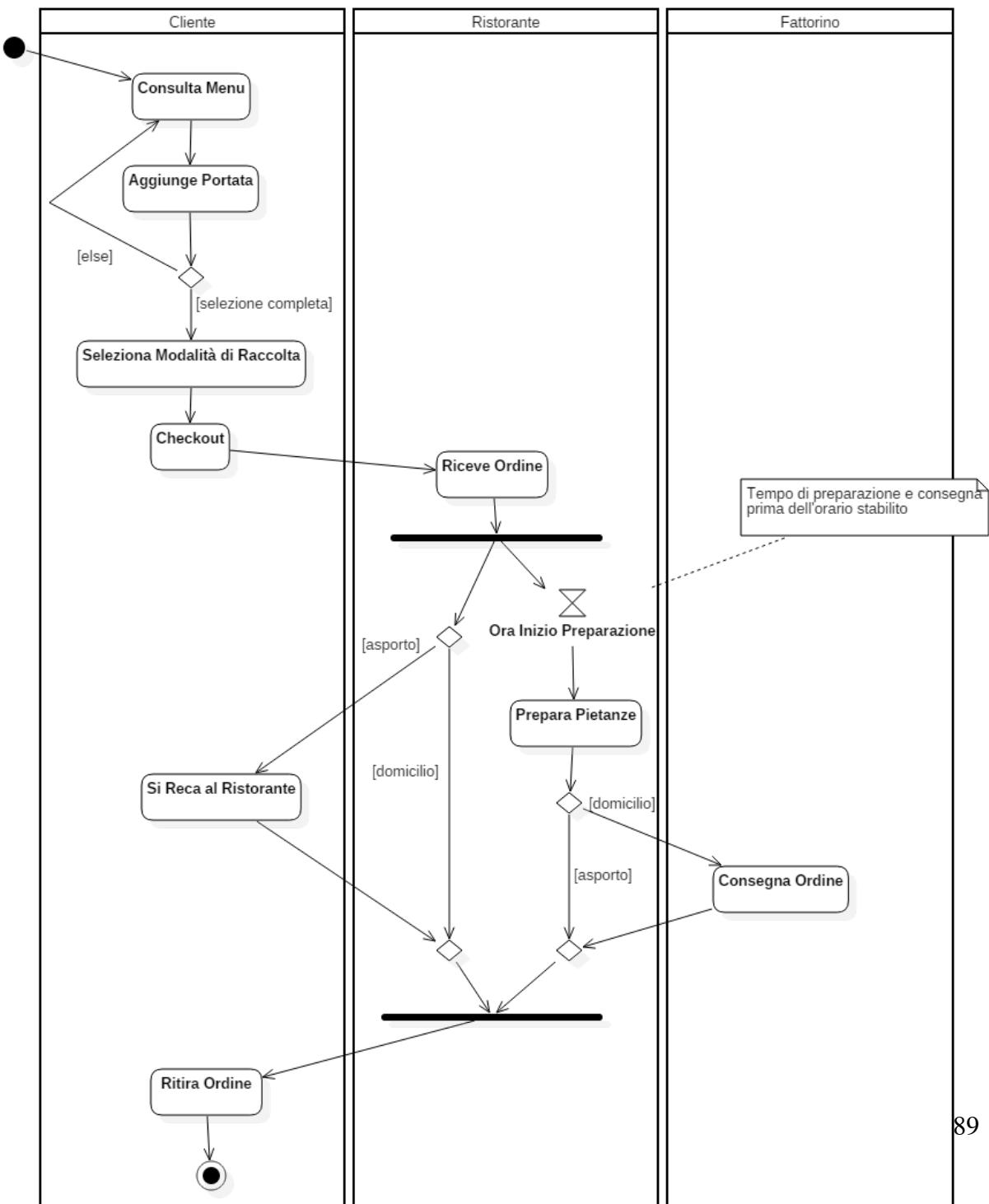


Si vuole modellare con *diagramma delle attività UML* il funzionamento di 0r4cl3, applicazione i cui giocatori si sfidano per indovinare un token segreto. L'applicazione è composta da una componente visuale (GUI) e da un oracolo, il cui compito è quello di dispensare saggi consigli per indirizzare i “guess” dei giocatori. Quando un giocatore inizia la partita, l'oracolo carica i consigli precedentemente dispensati e ne demanda la visualizzazione alla GUI. I consigli sono visualizzati per 30s. Allo scadere dei 30s il giocatore ha un breve lasso temporale (timeout) per inserire il guess. Se il giocatore non inserisce il guess in tempo, la GUI mostra il messaggio “Lost!” e la partita termina. Se il guess è stato inserito in tempo, la GUI visualizza il guess inserito e l'oracolo lo controlla. Se il guess è corretto, l'oracolo genera un nuovo segreto e rimuove i consigli precedentemente dispensati. Se il guess è errato, l'oracolo genera un nuovo consiglio e lo accoda a quelli esistenti. Una volta mostrato e verificato il guess, la GUI visualizza l'esito e la partita termina.



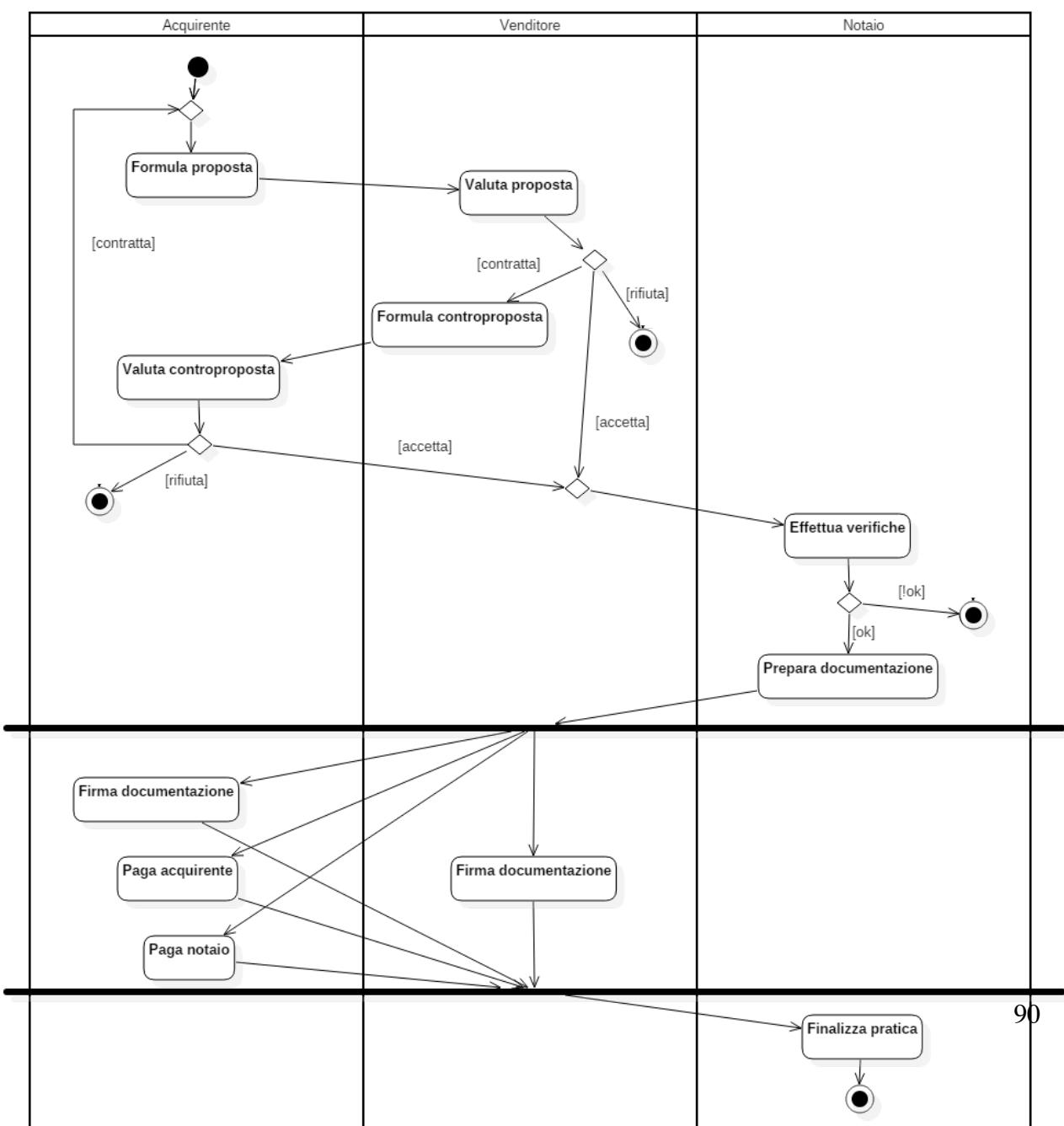
Si modelli tramite diagramma di attività il seguente scenario relativo all'ordinazione di pietanze tramite l'applicativo descritto nell'esercizio su GiastIt.

Dopo aver consultato il menu, il cliente aggiunge una ad una tutte le pietanze desiderate utilizzando l'apposita interfaccia messa a disposizione dall'applicazione. Quando tutte le portate sono state aggiunte il cliente seleziona la modalità di ritiro dell'ordine che preferisce (asporto o a domicilio) e procede alla conferma dell'ordine tramite pagamento (check-out). Una volta ricevuto l'ordine il ristorante attende il momento giusto (in modo tale che le pietanze non si raffreddino) prima di iniziare a preparare le pietanze richieste e, una volta pronte, in caso di ordine a domicilio vengono passate al fattorino incaricato che provvede a consegnarle al cliente. In caso l'ordine sia da asporto il cliente si reca al ristorante per ritirare le pietanze richieste (notare che il cliente può recarsi al ristorante mentre le pietanze sono ancora in preparazione).



Modellare tramite diagramma di attività UML il seguente scenario, in cui viene descritta la sottomissione di una proposta di acquisto di un immobile da parte di un soggetto acquirente a un soggetto venditore e il seguente iter burocratico.

Inizialmente l'acquirente formula una proposta di acquisto relativa un immobile di interesse, che viene valutata dal rispettivo venditore. Il venditore può decidere se accettare la proposta, rifiutarla o se formulare una controproposta da sottoporre all'acquirente. In quest'ultimo caso, l'acquirente può decidere a sua volta se accettare la controproposta, rifiutarla o se ripartire con una nuova proposta. Una volta che acquirente e venditore hanno trovato un accordo, la proposta viene presa in gestione da un notaio, il quale effettua tutte le verifiche necessarie e, in caso di accertamenti positivi, prepara la documentazione della compravendita. A questo punto, acquirente e venditore possono procedere con la compravendita, firmando (entrambi) la documentazione e pagando (l'acquirente) gli importi relativi alla compravendita e alle spese notarili. Il notaio procede alla finalizzazione delle pratiche non appena tutte le attività precedenti sono state eseguite.



Si vuole modellare tramite un diagramma di attività l'acquisto di prodotti tramite l'applicazione PO\$, un servizio di gestione di carte prepagate. L'acquisto inizia con la selezione del prodotto da parte del cliente. Una volta selezionato il prodotto, PO\$ attende l'inserimento del PIN dell'utente e, nel frattempo, mostra il messaggio "wait". Una volta inserito il PIN, PO\$ si interfaccia con la banca presso cui la carta è registrata, la quale verifica che il PIN inserito sia corretto. Se il PIN inserito non è corretto, PO\$ richiede nuovamente il PIN all'utente ripetendo la procedura di inserimento. L'utente ha 3 tentativi per inserire un PIN corretto. Al terzo inserimento errato, la banca blocca la carta, PO\$ segnala l'errore di "transazione fallita" all'utente e conclude il pagamento. Se il PIN inserito è corretto, la banca verifica il credito disponibile sulla carta. Se il credito non è sufficiente, PO\$ segnala l'errore di "transazione fallita" all'utente e conclude il pagamento. Se il credito è sufficiente, la banca aggiorna il credito residuo sulla carta, e, infine, PO\$ notifica all'utente la conclusione del pagamento.

