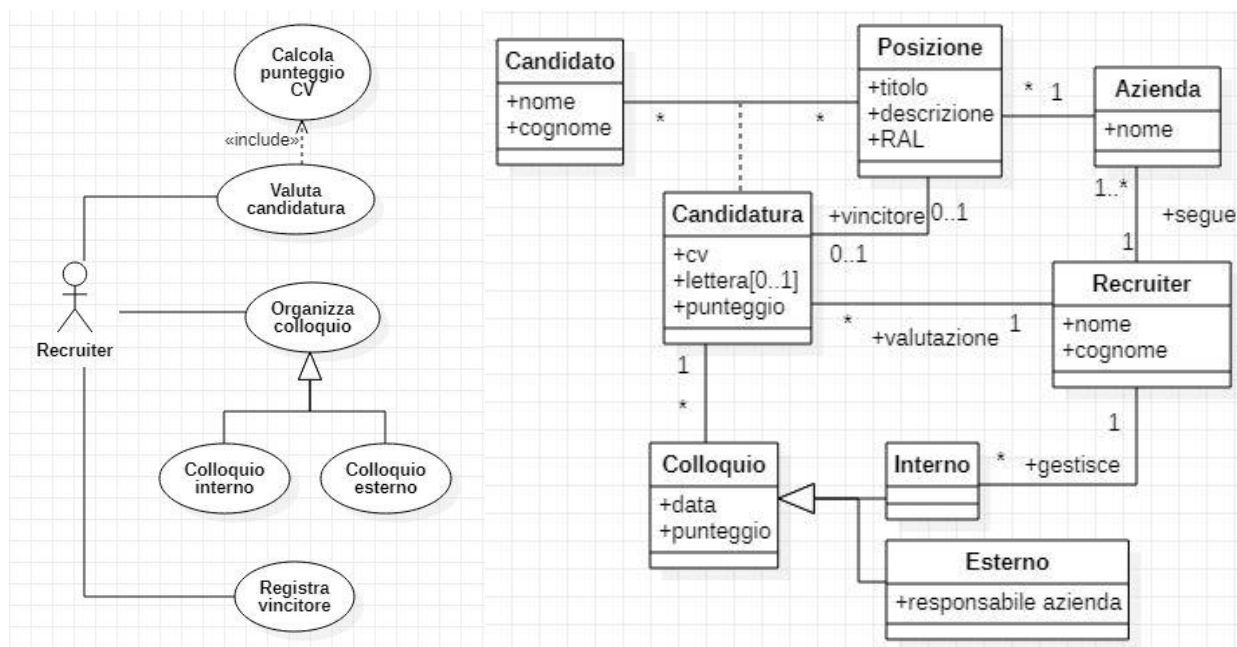


1) Si vuole modellare il sistema informativo di un'agenzia interinale che seleziona candidati per posti di lavoro su richiesta di aziende terze. Le aziende che vogliono aprire delle posizioni ne comunicano i dettagli all'agenzia (titolo della posizione, descrizione e RAL - reddito annuo lordo); una posizione si intende aperta per una sola persona. L'agenzia mantiene l'anagrafica delle persone che si candidano per una posizione e delle candidature sottomesse (comprendenti di CV e di una lettera di presentazione opzionale); una persona può candidarsi a più posizioni, ma un'unica volta per la stessa posizione. L'agenzia assegna un recruiter di riferimento per ciascuna azienda; il recruiter si occupa di valutare le candidature ricevute per una determinata posizione, assegnando un punteggio da 1 a 100. In base ai punteggi assegnati, il recruiter decide l'organizzazione di uno o più colloqui. I colloqui possono essere interni o esterni; nel primo caso vi partecipa un recruiter dell'agenzia (che non corrisponde necessariamente a quello che ha selezionato le candidature), nel secondo caso vi partecipa un responsabile indicato dall'azienda che offre il posto di lavoro. Di ogni colloquio si vogliono memorizzare la data di svolgimento ed il punteggio assegnato, anch'esso da 1 a 100. Al termine del percorso di selezione si vuole tenere conto della candidatura eventualmente selezionata per il posto di lavoro offerto. Si noti che il sistema è a uso esclusivo dell'agenzia (non è prevista un'interfaccia web a cui i candidati abbiano accesso).

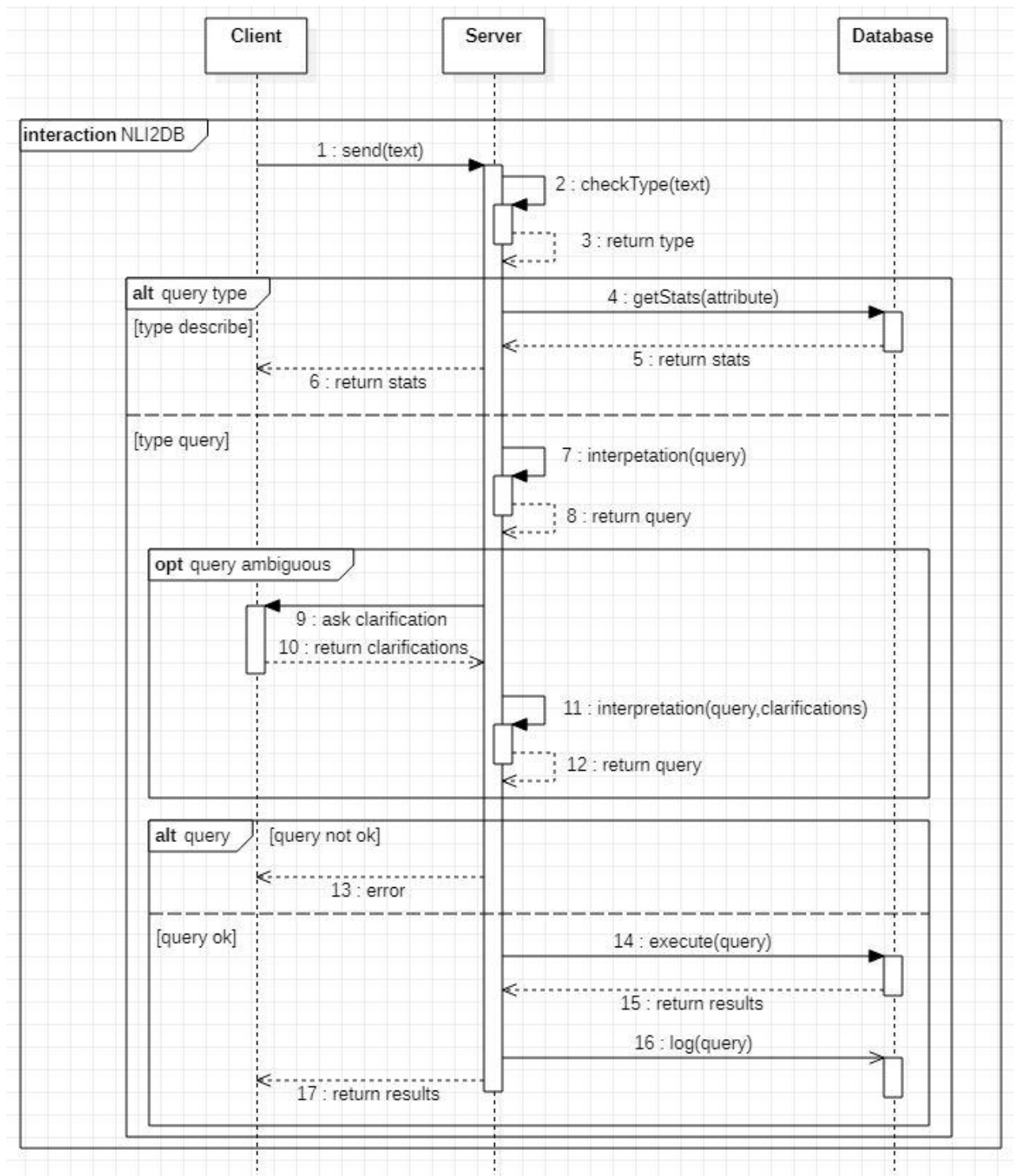
Si modellino le specifiche sopra riportate in UML attraverso:

- 1.1) un *diagramma dei casi d'uso* (3/32 punti)
- 1.2) un *diagramma delle classi* (9/32 punti)



2) Si vuole modellare con un *diagramma di sequenza* il funzionamento di NLI2DB, un sistema in grado di tradurre una stringa di testo (espressa in linguaggio naturale) in una query da eseguire su un database. L'interazione viene iniziata dal client, il quale invia una stringa di testo al server. Il server analizza la stringa per capire se si tratta di una richiesta di tipo *describe* o di tipo *query*. Nel primo caso, la stringa contiene semplicemente il nome di un attributo di una certa tabella di cui l'utente vuole conoscere i valori più utilizzati; il server esegue quindi una query sul database per recuperare i valori più utilizzati e li restituisce al client. Nel secondo caso, la stringa deve essere interpretata dal sistema per tradurre la richiesta da linguaggio naturale a query SQL. L'interpretazione del testo può essere ostacolata dalla presenza di alcune ambiguità (ad esempio, l'utente può menzionare un attributo "città" di cui esistono due varianti: "città di nascita" e "città di residenza"); in questo caso, il server interroga il client per risolvere le ambiguità e prova nuovamente ad interpretare il testo considerando anche le informazioni aggiuntive ricevute. In ogni caso, non è scontato che

l'interpretazione riesca a generare una query SQL valida; in tal caso, viene restituito un errore al client. Se la query risulta invece valida, questa viene eseguita sul database ed i risultati vengono inoltrati al client. Infine, il server salva (in maniera asincrona) l'SQL della query generata in una tabella di log del database (8/32 punti).



3) Si illustri il ruolo dello *studio di fattibilità* all'interno del ciclo di vita del software, precisandone obiettivi e risultati (6/32 punti).

4) Si spieghi cos'è un *package* e si illustrino struttura e utilizzo dei *diagrammi dei package* (6/32 punti).