*Titolo capitolo: 20pt – Titolo sottocapitolo: 16pt – paragrafo: 12pt*

*Titoli e parole chiave: grassetto – Nomi propri di variabili, campi, classi, funzioni, ecc: corsivo*

Descrizione progetto

Architettura (architettura generale + db diagramma relazionale)

* Operazione 0: Autenticazione studente (Sequenza operazioni)
* Operazione 1: Inserimento nuovo appello (Sequenza operazioni)
* Operazione 2: Inserimento nuova prenotazione dello studente (Sequenza operazioni)
* Operazione 3: Richiesta visione appelli disponibili (Sequenza operazioni)
* Operazione 4: Richiesta visione appelli prenotati dello studente (Sequenza operazioni)

Dettagli implementativi dei client/server

* Operazione 0: Autenticazione studente (Dettaglio operazioni + Classi librerie)
* Operazione 1: Inserimento nuovo appello (Dettaglio operazioni + Classi librerie)
* Operazione 2: Inserimento nuova prenotazione dello studente (Dettaglio operazioni + Classi librerie)
* Operazione 3: Richiesta visione appelli disponibili (Dettaglio operazioni + Classi librerie)
* Operazione 4: Richiesta visione appelli prenotati dello studente (Dettaglio operazioni + Classi librerie)

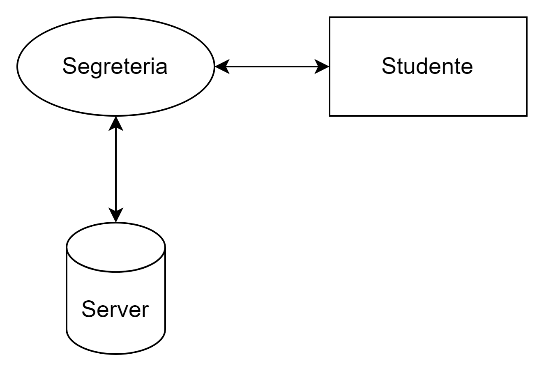
Parti rilevanti del codice sviluppato

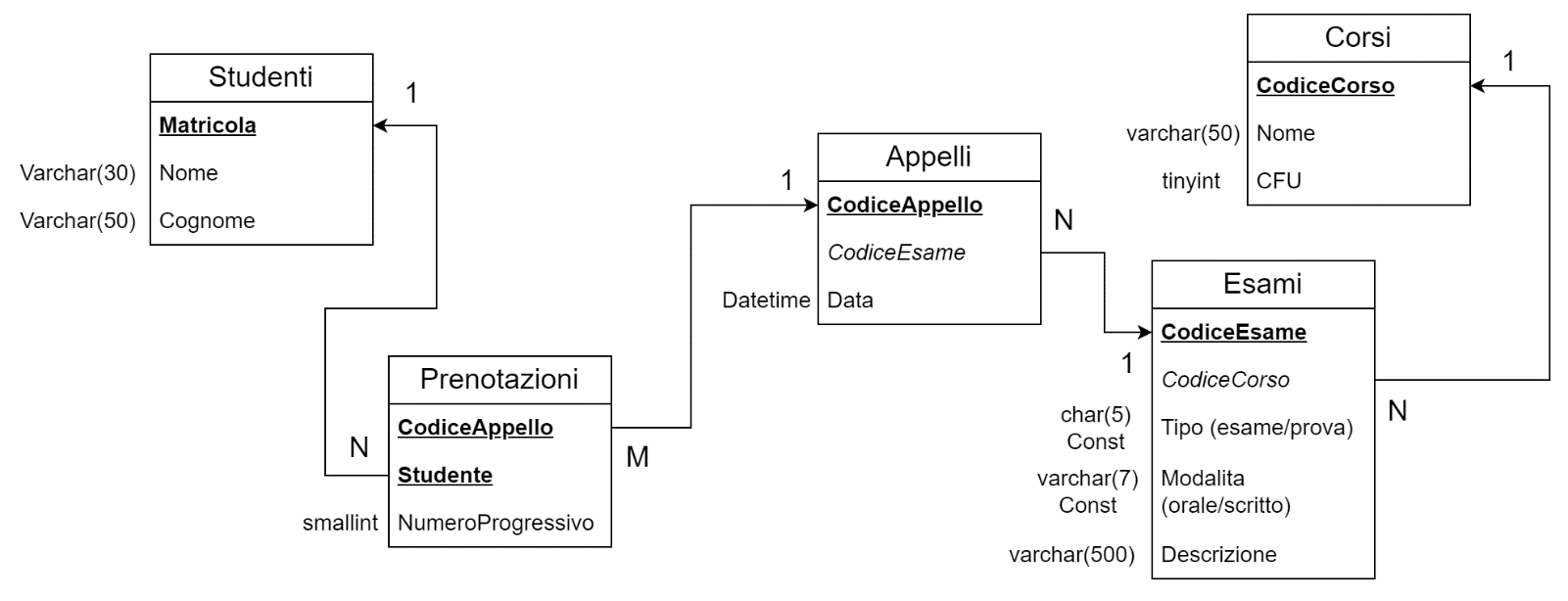
Manuale utente con le istruzioni su compilazione ed esecuzione

**Descrizione progetto**

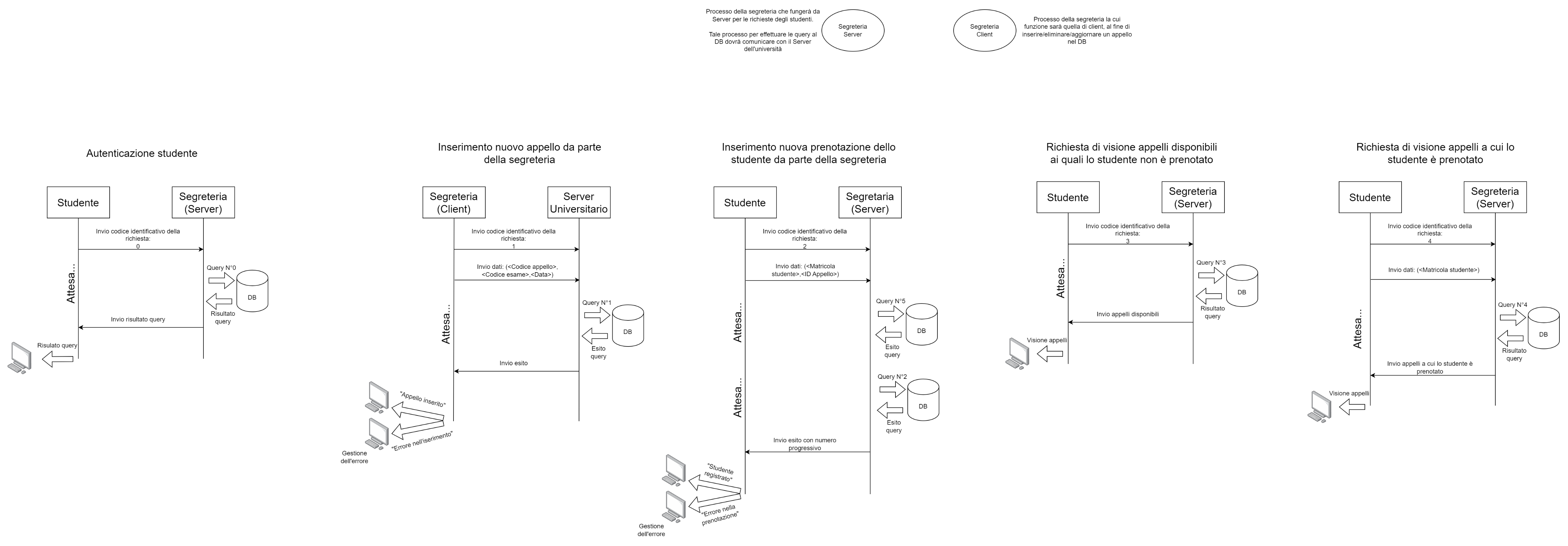
Si vuole scrivere un applicazione client/server concorrente per gestire gli esami universitari. Le tre entità sono: segreteria, studente e server universitario. La **segreteria** inserisce gli appelli sul server dell'università, inoltra la richiesta di prenotazione degli studenti al server universitario e infine fornisce allo studente le date degli esami disponibili per l'esame scelto dallo studente. Lo **studente** chiede alla segreteria se ci sono appelli disponibili per un corso ed infine invia una richiesta di prenotazione di un esame alla segreteria dalla quale riceve il numero progressivo della prenotazione. Infine il **server universitario** riceve l'aggiunta di nuovi esami e riceve la prenotazione di un esame.

**Schema architetturale**

Lo schema architetturale si basa su tre entità: il **server dell’università**, la **segreteria** e lo **studente**. Lo studente non si interfaccia direttamente con il server dell’università, ma lo fa attraverso la segreteria. Quindi quest’ultima altro non è che un **tramite**, quindi inoltra le richieste dello studente al server dell’università è poi inoltra i dati restituiti dal server allo studente. La segreteria è coinvolta attivamente solo nell’inserimento di un nuovo appello.

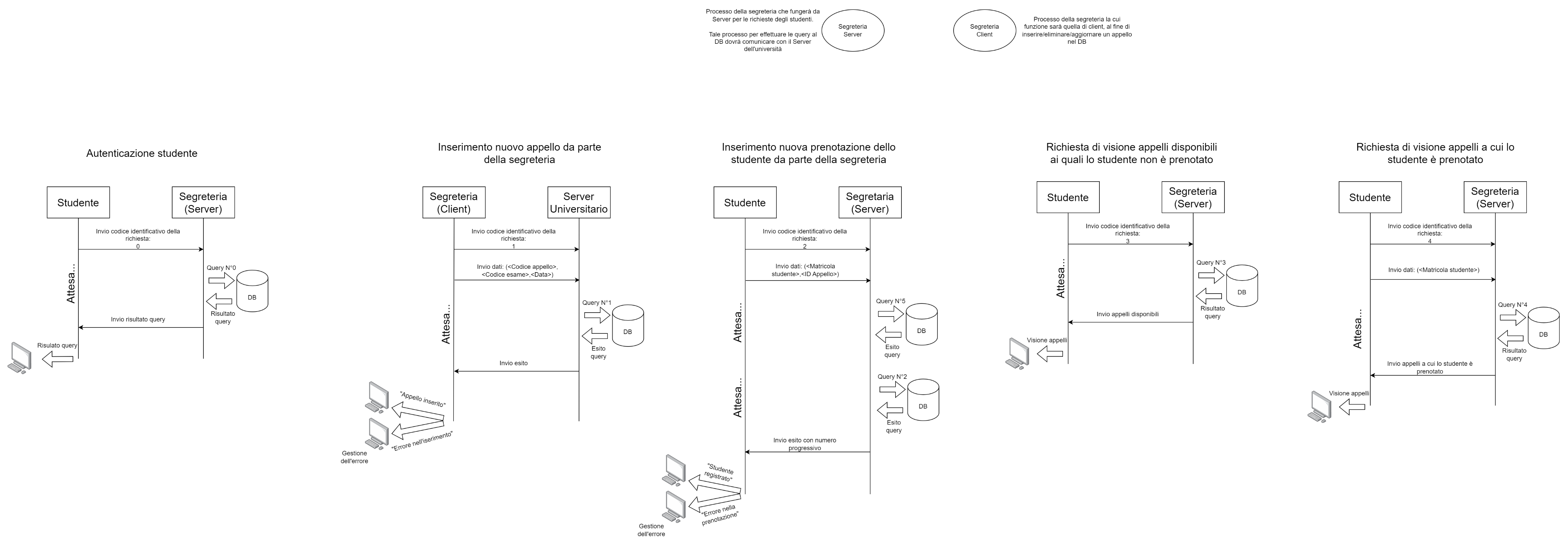
Data la presenza di dati strutturati si è scelto di utilizzare il database relazionale SQL, in particolare SQLite che possiede un’ottima libreria compatibile con il linguaggio C. Si è scelto di utilizzare il linguaggio C++ principalmente per due motivi: è orientato ad oggetti e le lezioni di laboratorio del corso sono in C.

**Operazione 0: Autenticazione studente**

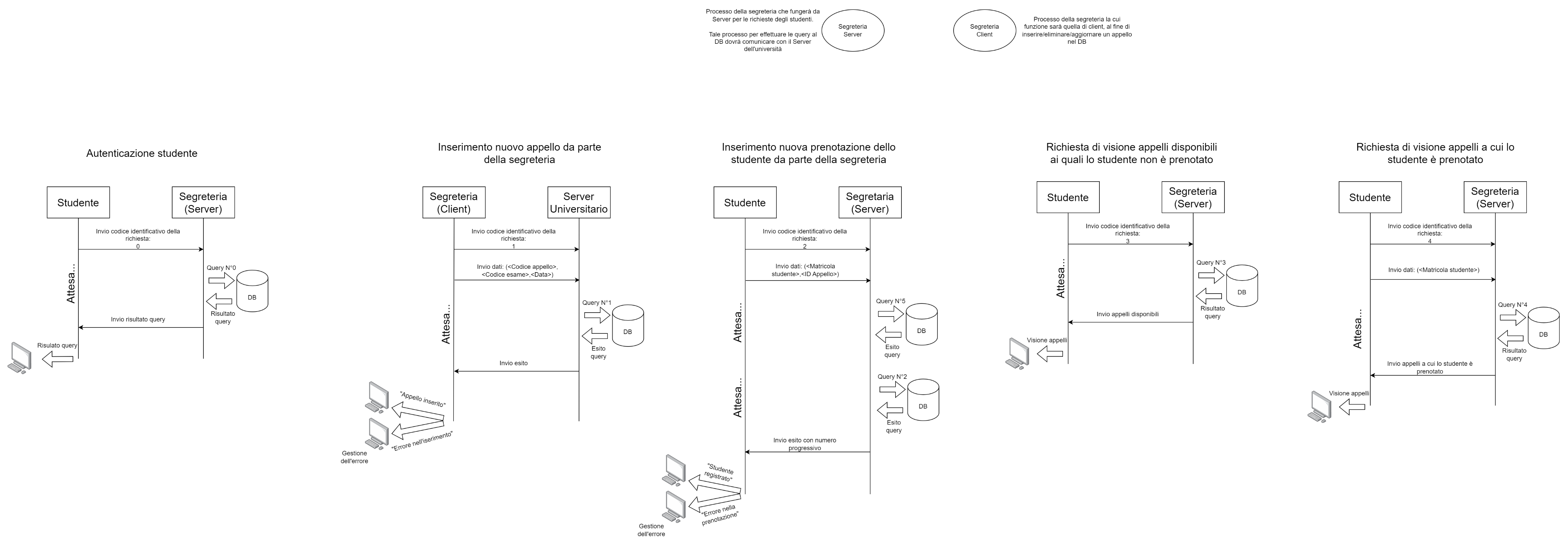
All’inizio, lo studente deve autenticarsi prima di poter effettuare un’operazione. Ciò avviene **una sola volta** e, per semplicità di implementazione, avviene scambiando solamente la matricola.

L’operazione inizia con lo studente che invia alla segreteria il codice dell’operazione da eseguire (l’autenticazione è identificata dal valore 0). La segreteria riceve la richiesta e la inoltra al server dell’università che poi verifica se lo studente è autenticato (controlla nel database). Infine il server restituisce il risultato (se lo studente è autenticato oppure no) alla segreteria che, a sua volta, lo inoltra allo studente. Mentre avviene lo scambio di dati tra la segreteria e il server, lo studente resta in attesa della risposta.

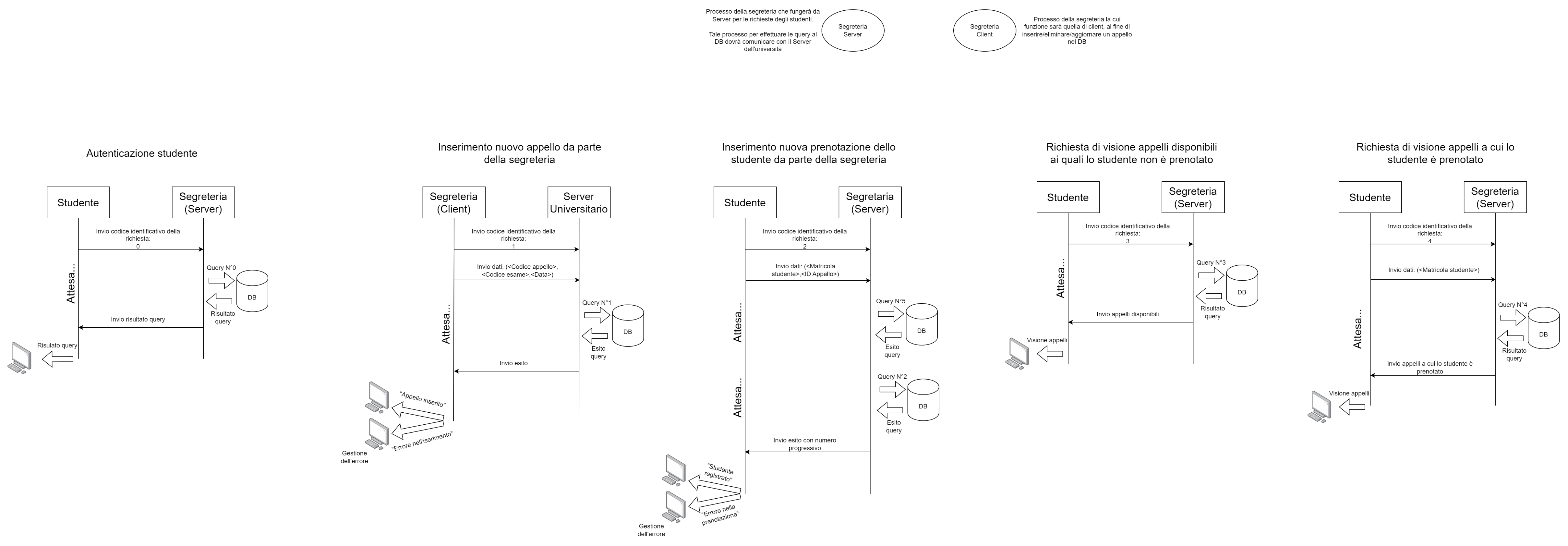
**Operazione 1: Inserimento nuovo appello**

L’inserimento di un nuovo appello può essere svolto solamente dalla segreteria. Questa operazione inizia con la segreteria che invia al server dell’università l’operazione da svolgere (la numero 1 corrisponde all’inserimento di un nuovo appello) e poi invia i dati dell’appello da inserire. Il server dell’università inserisce i dati nel database e poi restituisce alla segreteria l’esito dell’operazione (se è andata a buon fine oppure no). Durante lo scambio dei dati tra la segreteria e il server dell’università, lo studente resta in attesa di una risposta.

**Operazione 2: Inserimento nuova prenotazione dello studente**

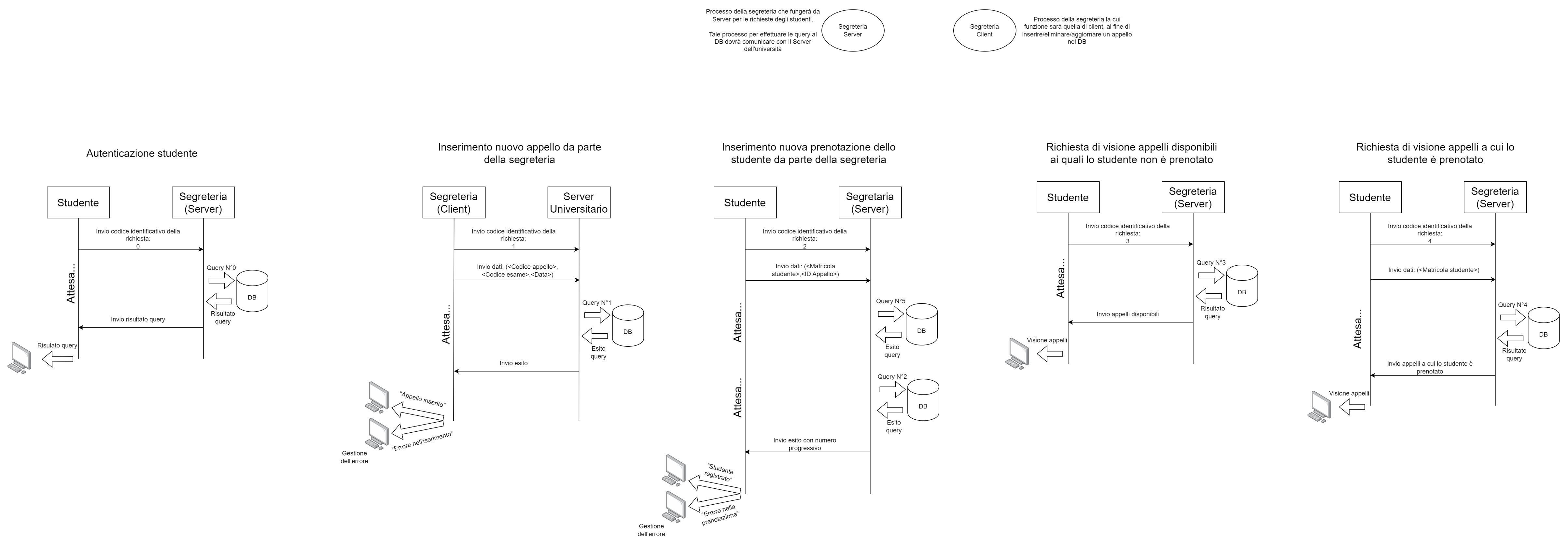
****L’operazione inizia con lo studente che sceglie l’operazione da svolgere. Poi un codice viene inviato alla segreteria (per l’operazione di inserimento di un appello il codice è 2) e subito dopo viene inviata anche la matricola dello studente e il codice dell’appello al quale prenotarsi. Dopodiché questi dati vengono inoltrati al server dell’università il quale genera il numero progressivo del prenotato a partire da quelli salvati e poi inserisce l’appello nel database. Il server dell’università invierà alla segreteria una risposta contenente il numero progressivo dello studente che si è appena prenotato. Infine la segreteria inoltra il numero progressivo allo studente. Durante lo scambio dei dati tra la segreteria e il server dell’università, lo studente resta in attesa di una risposta.

**Operazione 3: Visione appelli disponibili**

Con questa operazione lo studente vuole ottenere gli appelli disponibili e, ovviamente, non vengono mostrati quelli ai quali è già prenotato.

Tutto inizia con la scelta dell’operazione dello studente la quale viene inviata alla segreteria sempre sottoforma di codice (per questa operazione è 3). La segreteria inoltra la richiesta al server dell’università il quale preleva i dati dal database. Dopodiché il server invia i dati prelevati alla segreteria che, a sua volta, li inoltra allo studente. Durante lo scambio dei dati tra la segreteria e il server dell’università, lo studente resta in attesa di una risposta.

**Operazione 4: Visione appelli prenotati**

Con questa operazione lo studente vuole ottenere gli appelli ai quali è prenotato.

Il tutto inizia sempre con la scelta dell’operazione che poi viene inviata alla segreteria (il valore corrispondente è 3). La segreteria inoltra la richiesta al server dell’università il quale preleva i dati dal database per quello specifico studente. Infine il server restituisce i dati alla segreteria che a sua volta li inoltra allo studente. Durante lo scambio dei dati tra la segreteria e il server dell’università, lo studente resta in attesa di una risposta.