



Progetto Ingegneria del Software

A.A. 2020-2021

Lorenzo Camilli

Indice

1	Testo del progetto	2
2	Analisi di contesto	3
3	Specifica dei requisiti	5
3.1	Requisiti sui dati	5
3.2	Requisiti funzionali	6
3.3	Requisiti non funzionali	7
3.4	Diagramma use-case	8
4	Analisi	12
4.1	Analisi nomi-verbi	12
4.2	Schede CRC	12
4.3	Stereotipi	17
4.4	Organizzazione in package	18
5	Progetto	20
5.1	Specifica dell'architettura	20
5.2	Classi di progetto	21
5.3	Sequence diagrams	23
5.4	Cicli di vita delle classi	28

Capitolo 1

Testo del progetto

Progettare (in breve max. 30 pagine) un sistema di prenotazione di posto a lezione, nel rispetto della (nuova) capienza delle aule (caratterizzate da un codice). La presenza degli studenti è organizzata in turni prenotabili in base alle ultime due cifre del numero di matricola (da 00 a 49 e da 50 a 99) a settimane alterne. La prenotazione per ciascun turno può essere fatta (e cancellata, per permettere a studenti in lista d'attesa di frequentare) solo dal lunedì a venerdì della settimana precedente. Al termine della prenotazione, il sistema produce una ricevuta (con aula, matricola ed orario) da scaricare/salvare. Per accedere al sistema servono le credenziali di Sapienza.

DA CONSEGNARE:

1. Analisi di contesto
2. Specifica dei requisiti
3. Analisi
 - a. Nomi/verbi
 - b. Schede CRC
 - c. Classi di analisi (stereotipi entity/boundary/control)
 - d. Organizzazione in package
4. Progetto
 - a. Specifica dell'architettura
 - b. Classi di progetto, sottosistemi, componenti
 - c. Realizzazione dei casi d'uso (sequence diagrams)
 - d. Cicli di vita di classi significative

Capitolo 2

Analisi di contesto

Il sistema che si vuole sviluppare si inserisce all'interno di un contesto dell'università La Sapienza di Roma la quale ha un bacino di utenza di centinaia di migliaia di utenti tra professori, studenti e personale interno. Il sistema di prenotazione permetterà di prenotare posti in aula comunicando con i seguenti sistemi:

- **Infostud:** il sistema di gestione delle tasse, voti e altre procedure di proprietà dell'università. Utile per permettere l'autenticazione tramite le credenziali già a disposizione dello studente, ottenere informazioni da usare nel sistema, come l'elenco delle lezioni e dei corsi oppure informazioni ulteriori sugli studenti.
- **Il database delle prenotazioni:** in cui vengono conservate tutte le informazioni inerenti alle prenotazioni (es. studente che l'ha effettuata, data, stato...).
- **Il database delle aule:** sistema in cui sono contenute informazioni su tutte le aule in cui è possibile fare lezione quali, l'edificio in cui sono situate, il nome, la capienza, ed altre.

La figura 2.1 riporta il diagramma di contesto del sistema.

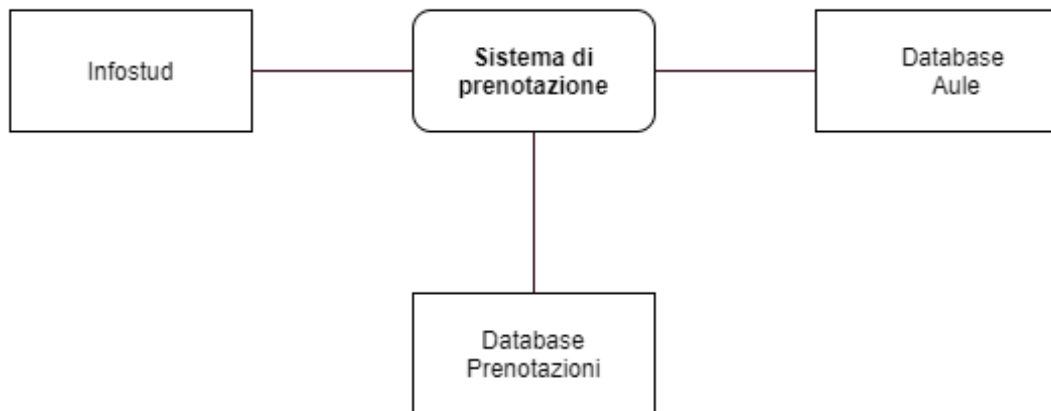


Figura 2.1: Figura che mostra il dominio in cui è situato il sistema di prenotazione.

Capitolo 3

Specifica dei requisiti

3.1 Requisiti sui dati

Vengono riportati i dati che interagiscono con il sistema.

1. Requisiti per gli **Studenti** (per studente si intende un utente, autenticato ed iscritto alla Sapienza), di cui interessa:
 - 1.1. Matricola
 - 1.2. Nome
 - 1.3. Cognome
 - 1.4. CF
 - 1.5. Credenziali ¹
 - 1.5.1. matricola
 - 1.5.2. password
 - 1.6. email
2. Requisiti sulle **Aule**:
 - 2.1. Capienza Covid
 - 2.2. Codice
 - 2.3. Nome
 - 2.4. Codice edificio
 - 2.5. Sede
 - 2.6. Ubicazione ²
 - 2.7. stato (es. agibile/non agibile)
3. Requisiti sulle **Prenotazioni**:
 - 3.1. Matricola studente prenotato (vedi req. 1)

¹Messe per completezza in realtà gestite da Infostud.

²Per ubicazione si intende dove è situata l'aula (es. Piazzale Aldo Moro 5, Via Scarpa...).

- 3.2. Stato prenotazione
 - 3.2.1. accettata/effettuata
 - 3.2.2. cancellata
 - 3.2.3. in attesa
- 3.3. Aula (vedi req. 2)
- 3.4. Data prenotazione
 - 3.4.1. Giorno
 - 3.4.2. Ora (da/a)
- 3.5. Posto
 - 3.5.1. Fila
 - 3.5.2. Posto

3.2 Requisiti funzionali

I requisiti funzionali che il sistema deve soddisfare sono:

1. AccessoInfostud:

- Descrizione: il sistema deve permettere l'accesso tramite le credenziali di Infostud.
- Priorità: Alta.

2. PrenotaMatricola:

- Descrizione: il sistema deve dare la possibilità di prenotarsi in base ai due numeri finali della matricola, se l'utente non è nella finestra temporale adatta non deve avere la possibilità di prenotarsi.
- Priorità: Alta.

3. Logout:

- Descrizione: il sistema deve permettere il logout in qualsiasi momento.
- Priorità: Alta.

4. CancellaPrenotazione:

- Descrizione: il sistema deve dare la possibilità all'utente di cancellare una prenotazione.
- Priorità: Alta.

5. VisualizzaPrenotazione:

- Descrizione: il sistema deve dare la possibilità all'utente di visualizzare le prenotazioni e le varie informazioni ad esse correlate (stato, data e ora, aula...).
- Priorità: Media.

6. NumeroPosti:

- Descrizione: se nell'aula non ci sono più posti disponibili, il sistema deve dare la possibilità all'utente di mettersi in attesa, solo se abilitato alla prenotazione e nella fascia di prenotazione corretta.

- Priorità: Media.

7. **VisualizzaRicevute:**

- Descrizione: il sistema deve permettere di visualizzare e scaricare in formato PDF la ricevuta di una prenotazione.
- Priorità: Media.

3.3 **Requisiti non funzionali**

1. **Disponibilità:**

- Descrizione: il sistema deve essere sempre disponibile e attivo, in caso contrario va segnalato.
- Priorità: Alta.

2. **Portabilità:**

- Descrizione: il sistema deve essere raggiungibile da ogni dispositivo, mobile o desktop garantendo sempre la stessa esperienza d'uso.
- Priorità: Alta.

3. **Comunicazione:**

- Descrizione: il sistema deve essere in grado di comunicare con la piattaforma Infostud per ottenere tutte le informazioni di cui necessita.
- Priorità: Alta.

4. **Interfaccia:**

- Descrizione: l'interfaccia del sistema deve essere intuitiva e semplice, fornendo solo le funzioni necessarie in modo comprensibile e senza ambiguità.
- Priorità: Media.

5. **TempoDiRisposta**

- Descrizione: il sistema deve rispondere agli input dell'utente il più velocemente possibile, in caso contrario deve segnalare l'errore.
- Priorità: Media.

6. **Posizionamento:**

- Descrizione: il sistema deve sempre far capire all'utente in quale parte del sistema si trova.
- Priorità: Bassa.

3.4 Diagramma use-case

Di seguito vengono riportati i diagrammi use-case, in modo da dare una descrizione delle funzioni o servizi offerti da un sistema dal punto di vista degli attori ³.

La figura 3.1 mostra una vista ad alto livello per gli use-case svolti da uno studente, nelle figure successive vengono presentate nel dettaglio i vari casi d'uso. Nella figura si può notare un unico attore (lo Studente) le cui azioni svolte sono: *Log-out*, per disconnettersi dal sistema, *Prenota*, che permette all'utente di svolgere varie azioni sulle prenotazioni, *Gestisci prenotazioni* per operare sulle prenotazioni e *Gestisci ricevute*. Alla destra della figura vengono riportati i sistemi utilizzati: Infostud per ottenere informazioni varie, il database delle prenotazioni ed il database delle aule per le varie informazioni sulle aule utili quando viene effettuata la prenotazione.

³I vari use-case sono stati separati per motivi di spazio e per garantire una migliore leggibilità.

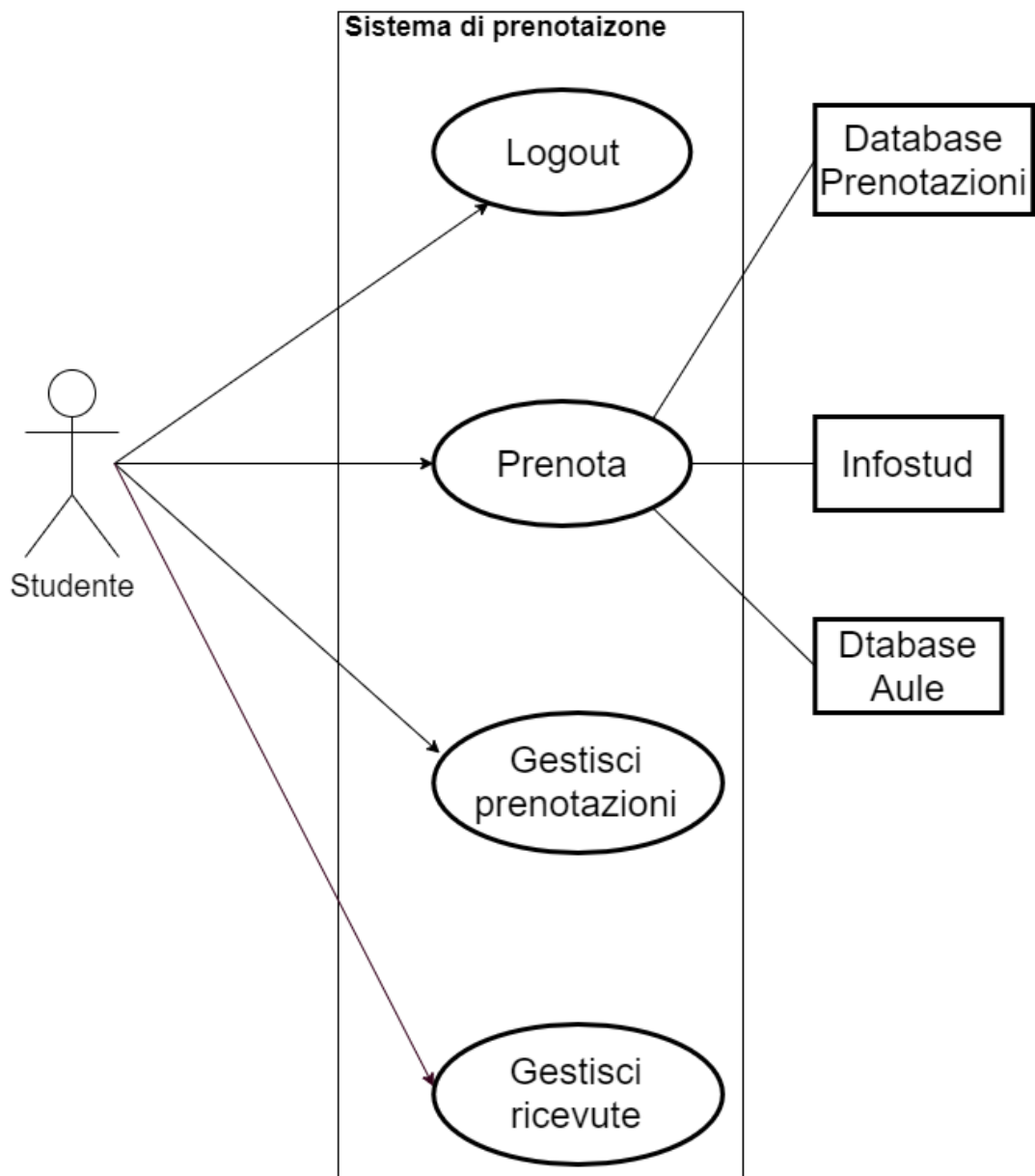


Figura 3.1: Vista ad alto livello degli use-case di uno studente.

La figura 3.2 mostra l'accesso al sistema da parte di un utente non autenticato.

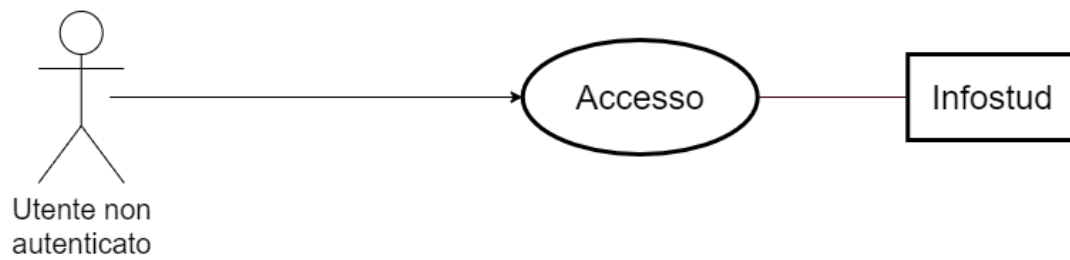


Figura 3.2: Rappresenta il diagramma use-case per l'accesso al sistema.

La figura 3.3 mostra nel dettaglio le azioni svolte dallo *Studente* per la gestione prenotazioni, in particolare può visionare le prenotazioni, effettuare una prenotazione e cancellare una prenotazione.

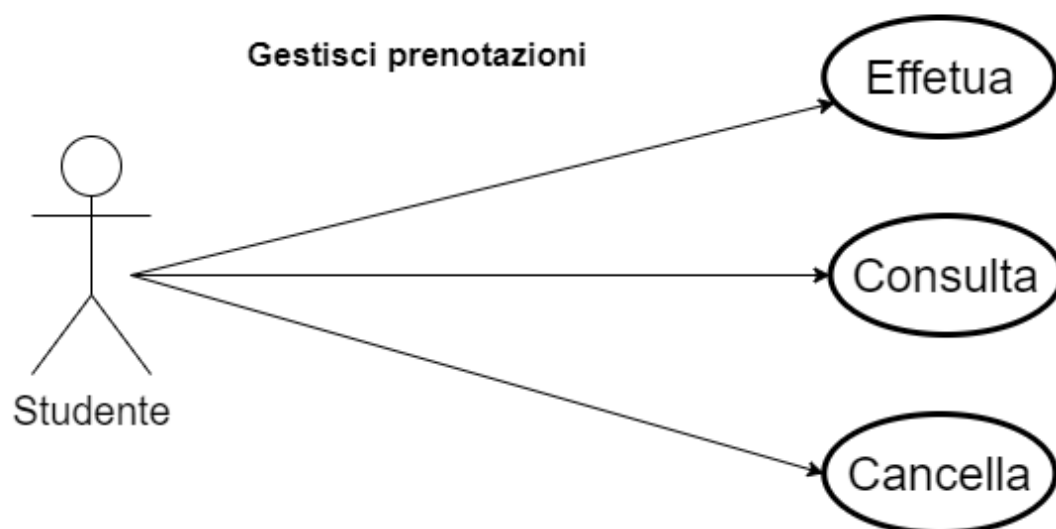


Figura 3.3: Rappresentazione nel dettaglio dell'use case *Gestione prenotazioni*.

La figura 3.4 mostra le azioni che deve svolgere l'utente per effettuare una prenotazione, ovvero scegliere il giorno, l'ora e l'aula d'interesse.

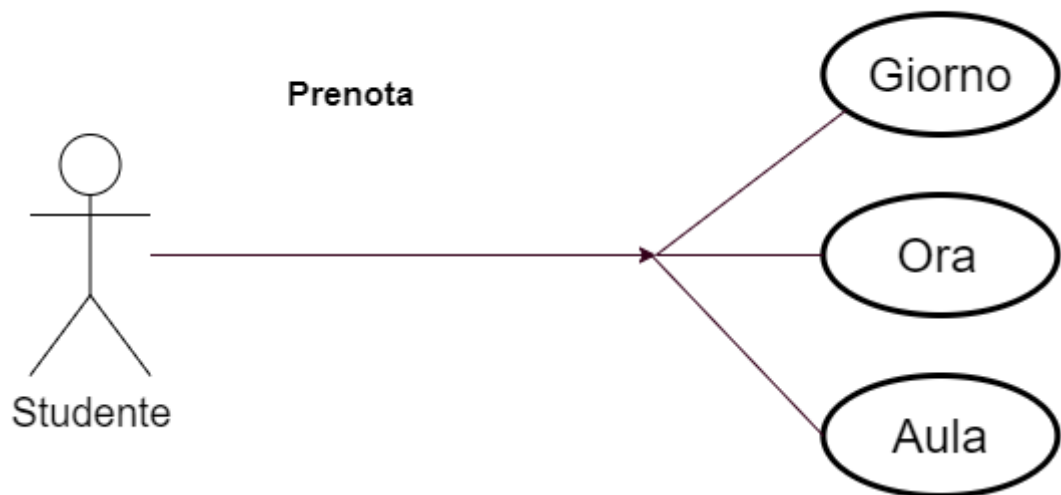


Figura 3.4: Rappresentazione nel dettaglio dell'use-case *Prenota*.

La figura 3.5 mostra che lo studente può di visualizzare o scaricare le ricevute delle prenotazioni.

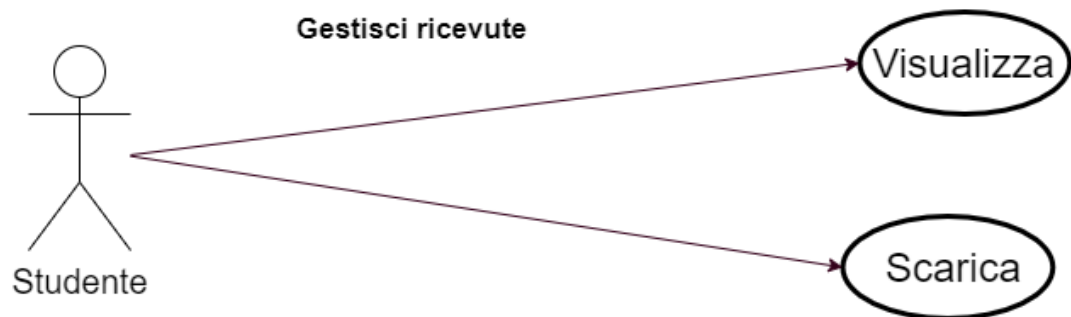


Figura 3.5: Rappresentazione nel dettaglio dell'use case *Gestione ricevute*.

Capitolo 4

Analisi

4.1 Analisi nomi-verbi

A questo punto partendo dalla specifica si vogliono identificare gli oggetti e le operazioni nel sistema, individuando alcune parole chiave:

- In **grassetto** vengono evidenziati i nomi, che potranno essere classi.
- In *corsivo* vengono evidenziati i verbi, che identificano operazioni o servizi, i metodi delle classi.
- In sottolineato gli attributi.

Progettare un sistema di prenotazione di **posto** a lezione, nel rispetto della (nuova) capienza delle **aule** (caratterizzate da un codice). La presenza degli **studenti** è organizzata in **turni** *prenotabili* in base alle ultime due cifre del numero di matricola (da 00 a 49 e da 50 a 99) a settimane alterne. La prenotazione per ciascun turno può essere *fatta* (e *cancellata*, per permettere a studenti in lista d'attesa di *frequentare*) solo dal lunedì a venerdì della settimana precedente. Al termine della prenotazione, il sistema *produce* una **ricevuta** (con aula, matricola ed orario) da *scaricare*/*salvare*. Per *accedere* al sistema servono le **credenziali** di Sapienza.

4.2 Schede CRC

Le schede CRC (*Class, Responsibility, Collaboration*) descrivono in modo semplice e veloce le classi di analisi estratte con l'analisi nomi-verbi. Ciascuna scheda descrive una classe (o un oggetto) indicando ⁴:

1. Il nome della classe (*class*).
2. Le sue superclassi e sotto classi.
3. Le sue responsabilità, ovvero le azioni che permette di svolgere o che svolge (*responsibility*).
4. Il nome di altre classi con cui questa classe collabora per svolgere i compiti di cui è responsabile (*collaboration*).

⁴Alcune implementazioni delle schede CRC presentano anche gli attributi dell'oggetto o della classe, in questo caso non sono stati inseriti ma sono visibili nella sezione 3.1 inerente alla specifica dei dati.

Nome	Studente
Descrizione	Classe che rappresenta un utente autenticato nel sistema.
SuperClasse	-
SottoClasse	-
Responsabilità	-
Collaborazioni	-

Nome	Utente non autenticato
Descrizione	Classe che rappresenta un utente non ancora autenticato nel sistema.
SuperClasse	-
SottoClasse	-
Responsabilità	-
Collaborazioni	-

Nome	Prenotazione
Descrizione	Classe che modella l'oggetto prenotazione, contiene tutte le informazioni che identificano una prenotazione.
SuperClasse	-
SottoClasse	-
Responsabilità	-
Collaborazioni	-

Nome	Aula
Descrizione	Classe che astrae il concetto di aula, contiene tutte le informazioni inerenti ad un aula.
SuperClasse	-
SottoClasse	-
Responsabilità	-
Collaborazioni	-

Nome	RicevutaPrenotazione
Descrizione	Rappresenta l'oggetto ricevuta della prenotazione
SuperClasse	-
SottoClasse	-
Responsabilità	-
Collaborazioni	-

Nome	Handler	Effetua Prenotazioni
Descrizione	Classe che permette la gestione di una prenotazione.	
SuperClasse	-	
SottoClasse	-	
Responsabilità	<ul style="list-style-type: none"> • Effettua dei controlli sulla capienza dell'aula. • Effettua i controlli sulla possibilità di effettuare una prenotazione (settimana e matricola). • Permette di effettuare una prenotazione in base ai dati presi in input. • Crea una ricevuta come output della prenotazione con tutte le informazioni necessarie. 	
Collaborazioni	Studente, Prenotazione, Aula, RicevutaPrenotazione, HandlerInviaMail.	

Nome	Handler	Cancella
Descrizione	Classe che gestisce l'eliminazione di prenotazioni in corso.	
SuperClasse	-	
SottoClasse	-	
Responsabilità	<ul style="list-style-type: none"> • Permette di cancellare una prenotazione. 	
Collaborazioni	Prenotazione.	

Nome	Handler	Visualizza
Descrizione	Classe che gestisce la visualizzazione delle prenotazioni.	
SuperClasse	-	
SottoClasse	-	
Responsabilità	<ul style="list-style-type: none"> • Permette di visualizzare le prenotazioni in corso con tutte le informazioni. • Permette di visualizzare le prenotazioni scadute. • Permette di visualizzare lo stato di una prenotazione. 	
Collaborazioni	Prenotazione, Studente.	

Nome	HandlerScarica
Descrizione	Classe che gestisce la visualizzazione e il download delle ricevute.
SuperClasse	-
SottoClasse	-
Responsabilità	<ul style="list-style-type: none"> • Permette di visualizzare la ricevuta di una prenotazione. • Permette di scaricare in locale la ricevuta di una prenotazione in formato PDF.
Collaborazioni	Prenotazione, RicevutaPrenotazione, HandlerEffetuaPrenotazioni, HandlerInviaMail.

Nome	HandlerGestisciCoda
Descrizione	Classe che aggiunge rimuove e gestisce gli utenti in coda.
SuperClasse	-
SottoClasse	-
Responsabilità	<ul style="list-style-type: none"> • Inserisce un utente in coda se non è possibile effettuare una prenotazione. • Rimuove quando necessario il primo utente dalla coda, aggiornando lo stato della prenotazione.
Collaborazioni	Prenotazione, Aula, Studente, HandlerInviaMail.

Nome	HandlerInviaMail
Descrizione	Classe che gestisce l'invio delle mail agli studenti.
SuperClasse	-
SottoClasse	-
Responsabilità	<ul style="list-style-type: none"> • Invia per mail la ricevuta della prenotazione. • Invia per mail eventuali aggiornamenti sullo stato della prenotazione.
Collaborazioni	Prenotazione, Studente, RicevutaPrenotazione, HandlerGestisciCoda, HandlerEffetuaPrenotazioni.

Nome	HandlerLogin
Descrizione	Classe che gestisce il login dell'utente.
SuperClasse	-
SottoClasse	-
Responsabilità	<ul style="list-style-type: none"> • Permette di effettuare il login. • Segnala con un errore eventuali problemi.
Collaborazioni	Studente.

Nome	HandlerLogout
Descrizione	Classe che gestisce il logout dell'utente.
SuperClasse	-
SottoClasse	-
Responsabilità	<ul style="list-style-type: none"> • Permette di effettuare il logout. • Segnala con un errore eventuali problemi.
Collaborazioni	Studente.

Nome	UILogin
Descrizione	Classe che rappresenta l'interfaccia che permette il login all'interno del sistema.
SuperClasse	-
SottoClasse	-
Responsabilità	-
Collaborazioni	HandlerLogin.

Nome	UILogout
Descrizione	Classe che rappresenta l'interfaccia che permette il logout dal sistema.
SuperClasse	-
SottoClasse	-
Responsabilità	-
Collaborazioni	HandlerLogout.

Nome	UIPrenotazione
Descrizione	Classe che rappresenta l'insieme degli elementi dell'interfaccia che permettono di effettuare una prenotazione e gestire le prenotazioni da parte di uno studente.
SuperClasse	-
SottoClasse	-
Responsabilità	-
Collaborazioni	HandlerEffetuaPrenotazioni, HandlerCancella, HandlerVisualizza, HandlerScarica, HandlerGestisciCoda, HandlerInviaMail.

4.3 Stereotipi

- **Entity:** classe che mantiene informazioni di lunga durata, tipicamente persistenti
- **Boundary:** classe che modella l'interazione tra sistema e attori, spesso è un'astrazione di un elemento software di interfaccia
- **Control:** classe che si occupa del controllo e coordinamento di altri oggetti, tipicamente è associabile a un caso d'uso.

La figura 4.1 mostra il diagramma.

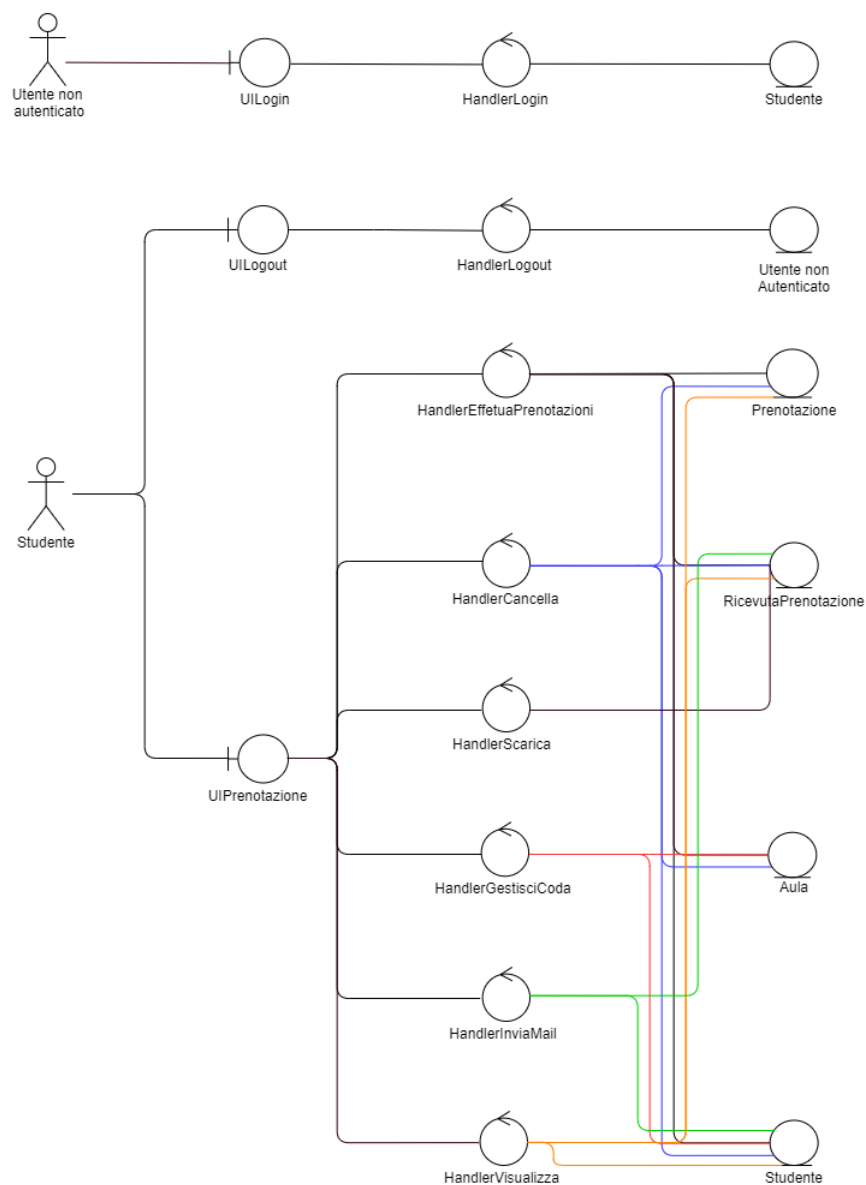


Figura 4.1: Diagrammi EBC.

4.4 Organizzazione in package

A questo punto gli elementi individuati nelle sezioni precedenti vengono raggruppati ad alto livello in package comunicanti, la suddivisione è visibile nella figura 4.2

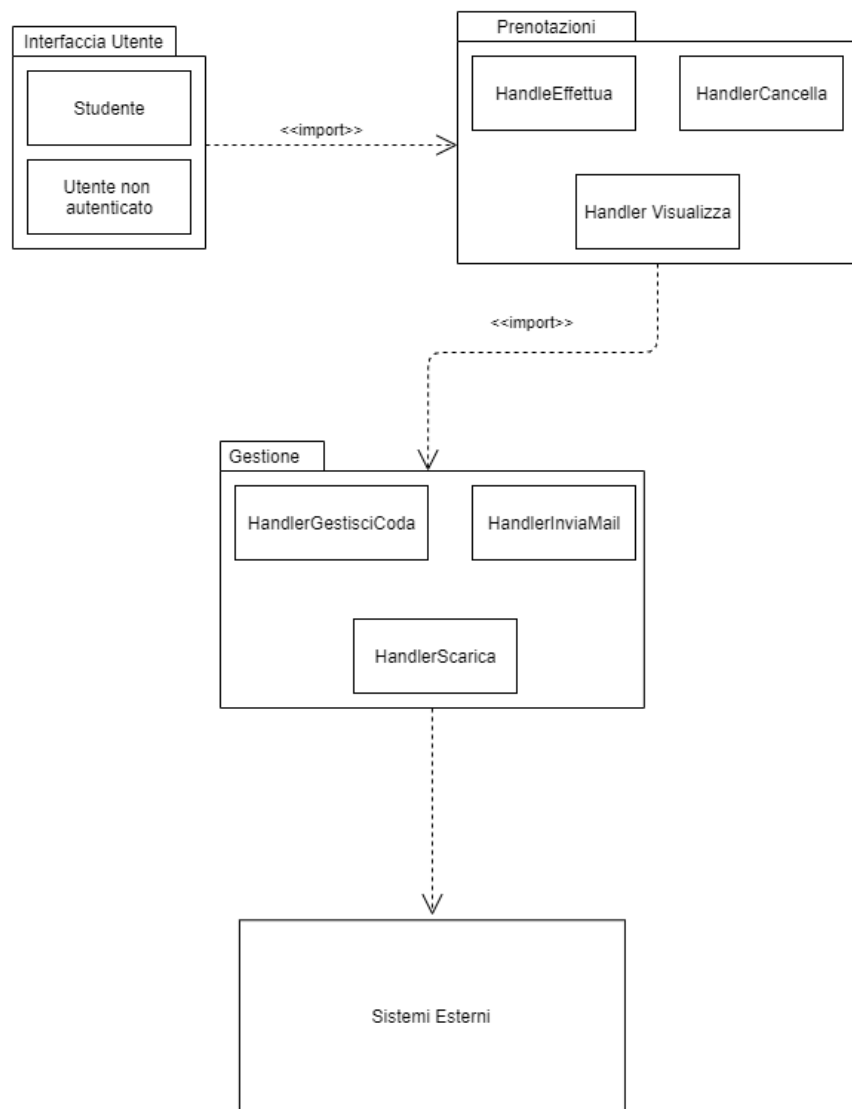


Figura 4.2: Organizzazione in package.

Capitolo 5

Progetto

5.1 Specifica dell'architettura

Il sistema è accessibile tramite browser, in particolare sfrutta un'architettura di tipo *client-server*, il client è il dispositivo da cui accede l'utente, che comunicano con Web Browser tramite protocollo *HTTPS* in modo da garantire uno scambio dei dati sicuro tra le due estremità. Il sistema principale si interfaccia (tramite *HTTPS* e *TCP/IP*) con gli altri sistemi che sono: il **Database delle Aule** tramite un *DBMS* che viene interrogato all'occorrenza, il **Database delle prenotazioni** contenente tutte le prenotazioni, il sistema **Infostud** tramite il quale verifica le credenziali d'accesso e richiede informazioni necessarie sullo studente che sono contenute nel **Database degli studenti**. Infine è presente un **server di posta elettronica** (lo stesso che usa Infostud) per l'invio delle e-mail. La figura 5.1 mostra un grafico esplicativo del design architetturale.

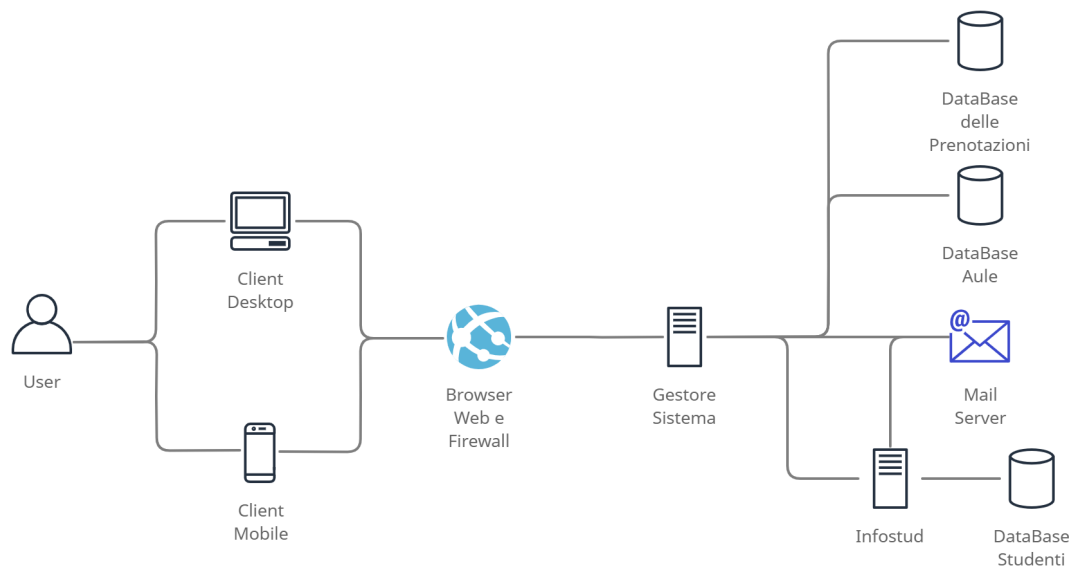


Figura 5.1: Diagramma dell'architettura.

5.2 Classi di progetto

La classe **Studiante** ha solo il metodo per ottenere i dati, (es. matricola o indirizzo mail) **GetInfo()**, non ha altri metodi per aggiornare le informazioni poichè queste vengono modificate all'interno del sistema Infostud.

La classe **Aula** ha diversi metodi per la gestione, in particolare: **SetState()** che permette di aggiornare lo stato dell'aula (se non dovesse essere disponibile), **GetState()** permette di ottenere lo stato dell'aula, **GetCapacity()**, la capienza Covid dell'aula e **SetCapacity()** imposta la capienza dell'aula (se si ha la necessità di dover ampliare o ridurre la capienza), i metodi **GetInfo()** e **SetInfo()** servono rispettivamente per ottenere e aggiornare le informazioni inerenti all'aula.

La classe **Prenotazione**, usa il metodo **CreatePrenotazione()** per creare una prenotazione riferita ad uno studente e ad un'aula, **CreateRicevuta()** crea la ricevuta a partire dalla prenotazione creata, ricevuta da mandare per mail tramite il metodo **SendMail()**, **SendAlert()** serve per inviare un messaggio d'errore in caso l'utente non sia abilitato a prenotarsi, oppure nel caso in cui si prova a fare una prenotazione multipla dell'aula nella stessa fascia oraria o nel caso ci sia un errore nell'inserimento dei dati.

La classe **GestisciPrenotazioni** è in dipendenza di tipo «use» alla classe **Prenotazioni** poiché essa necessita della seconda per operare. La classe ha i seguenti metodi: **DeletePrenotazione()** per cancellare una determinata prenotazione, **ViewPrenotazione()** per vedere le prenotazioni effettuate, **ViewStatePrenotazione()** per ottenere lo stato (effettuata, cancellata o in attesa)

di una determinata prenotazione, **DownloadRicevuta()** per scaricare la ricevuta di una prenotazione e **ViewRicevuta()** per visualizzare la ricevuta.

La classe **Coda** si occupa di gestire le code per le prenotazioni delle aule, con il metodo **AddToQueue()** aggiunge lo studente alla coda nel caso in cui l'aula sia piena, **RemoveFromQueue()** rimuove uno studente dalla coda (aggiornandola) nel caso in cui abbia cancellato la prenotazione riferita all'aula, oppure nel caso in cui il primo studente della coda viene aggiunto alle prenotazioni effettuate (per scorrimento), **UpdateStatoPrenotazione()** aggiorna lo stato di una prenotazione quando si esce dalla coda (es. effettuata o cancellata), infine con **SendMail()** viene inviata una mail di notifica nel caso in cui si verifichi un cambiamento all'interno della coda. La figura 5.2 riporta quanto detto ma sotto forma di diagramma delle classi.

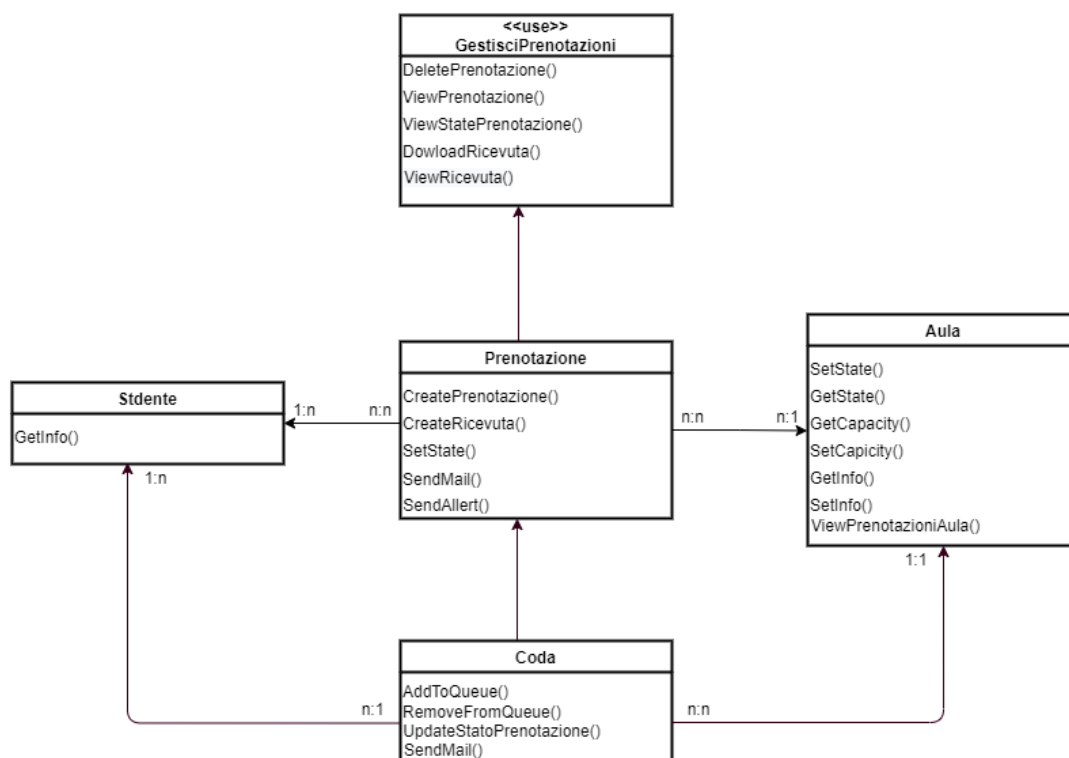


Figura 5.2: Diagramma delle classi.

Le molteplicità tra la classe **Stdente** e **Prenotazione**: $(1:n)$ indica che ad uno studente corrispondono n prenotazioni, $(n:n)$ che ad n prenotazioni corrispondono n studenti; quella tra le classi **Prenotazione** e **Aula** indicano: $(n:1)$ che ad un'aula corrispondono n prenotazioni, ma $(n:n)$ che ad n prenotazioni corrispondono n aule.

Le molteplicità tra la classe **Coda** e la classe **Stdente** rappresenta il fatto che uno studente può trovarsi in n code $(1:n)$ e che in una coda possono esserci n studenti $(n:1)$. Le molteplicità

tra le classi **Coda** ed **Aula** indicano che ad un'aula corrisponde una ed una sola coda (*1:1*) mentre ad *n* code corrispondono *n* aule (*n:n*).

5.3 Sequence diagrams

In questa sezione ci interessa descrivere le azioni svolte per un determinato caso d'uso, raffigurandole tramite i diagrammi di sequenza.

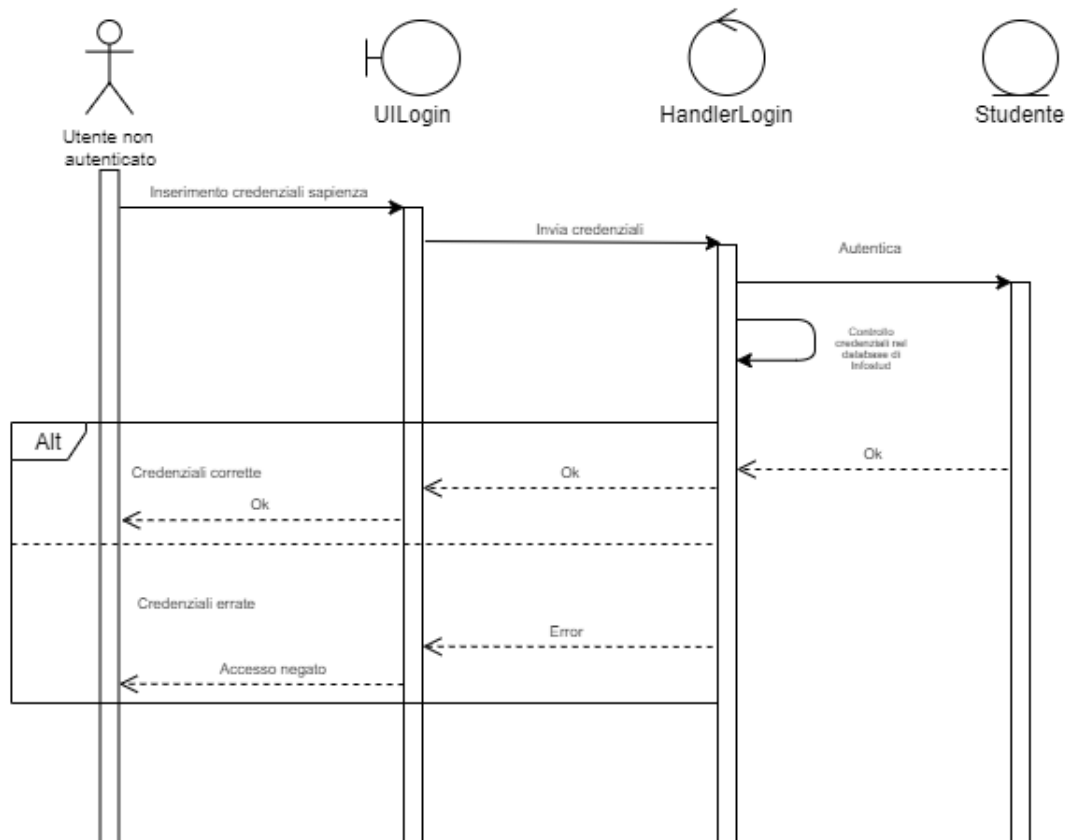


Figura 5.3: Diagramma di sequenza per il login.

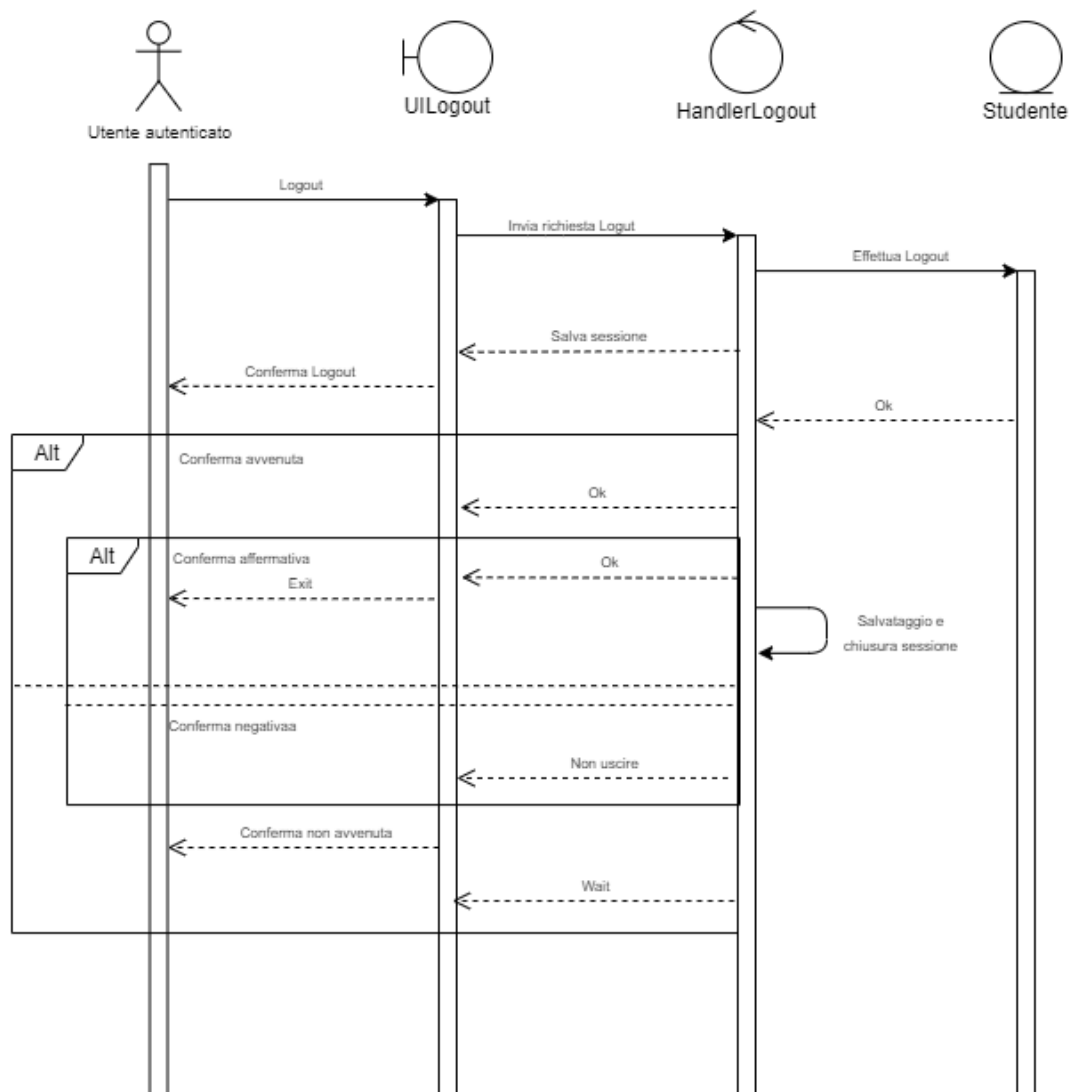


Figura 5.4: Diagramma di sequenza per il logout.

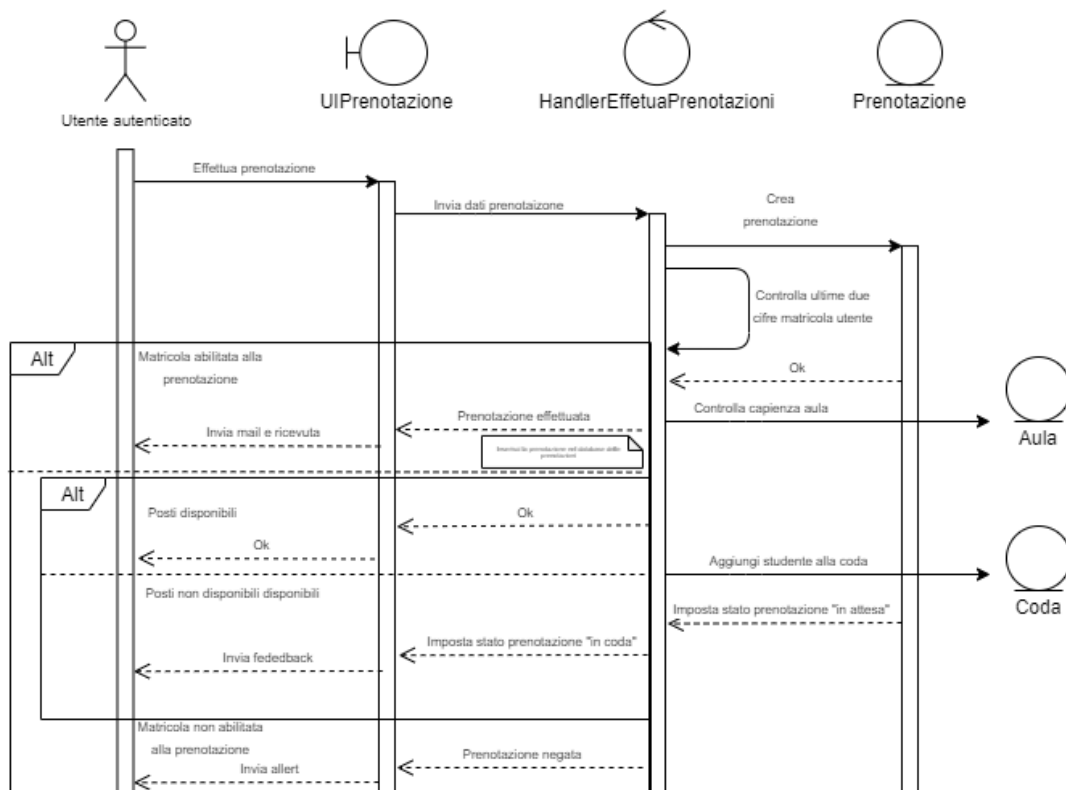


Figura 5.5: Diagramma di sequenza che mostra come effettuare la prenotazione.

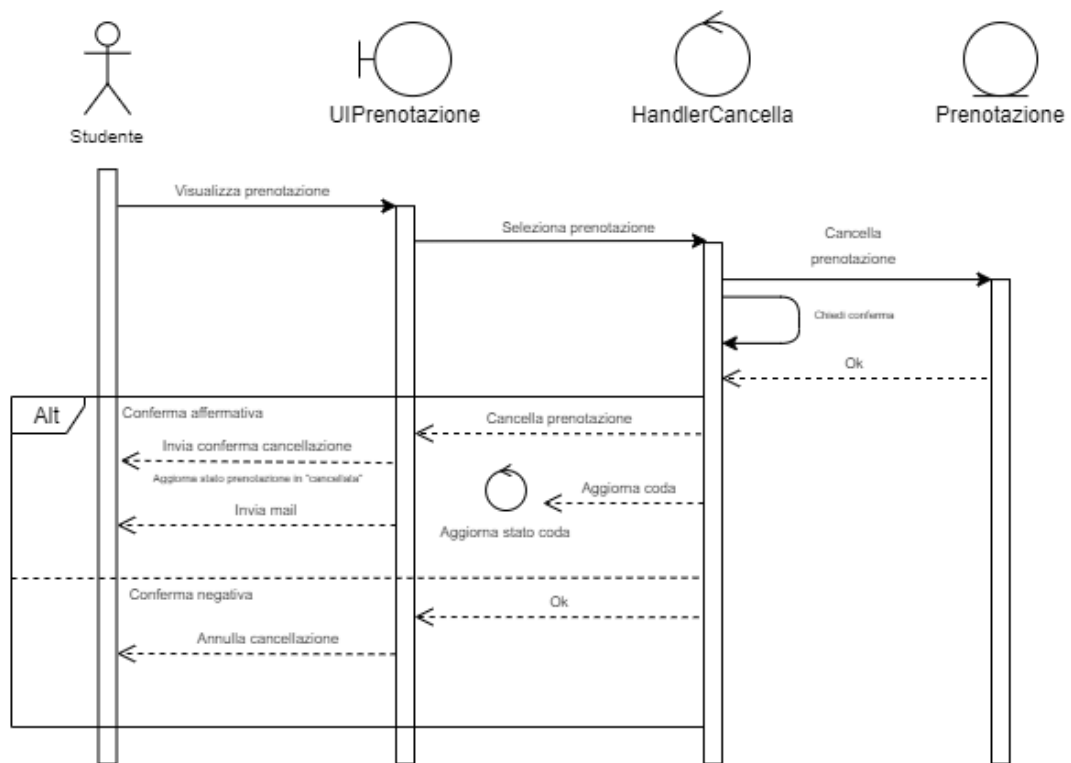


Figura 5.6: Diagramma di sequenza cancellazione prenotazione.

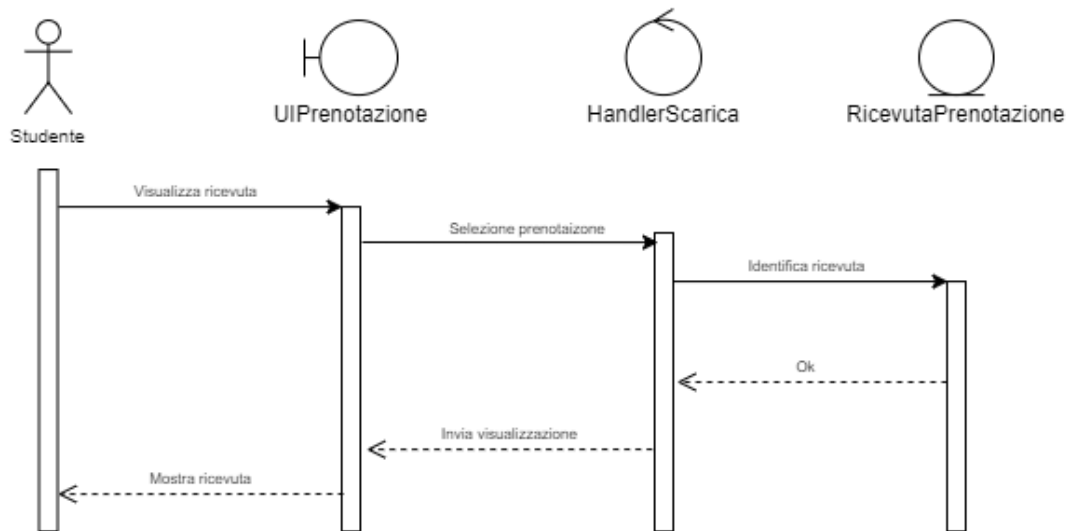


Figura 5.7: Diagramma di sequenza visualizzazione prenotazione.

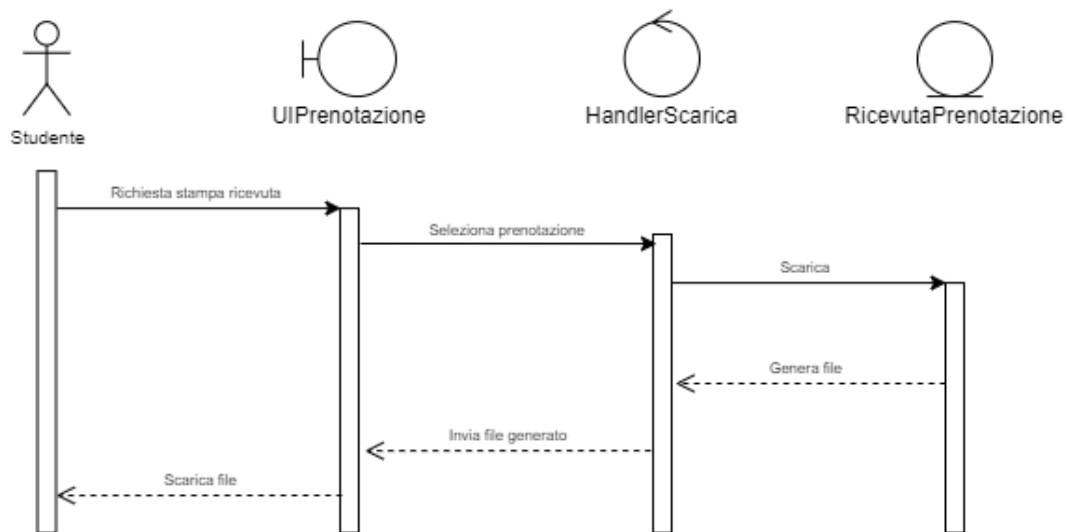


Figura 5.8: Diagramma di sequenza per lo scaricamento della ricevuta in PDF.

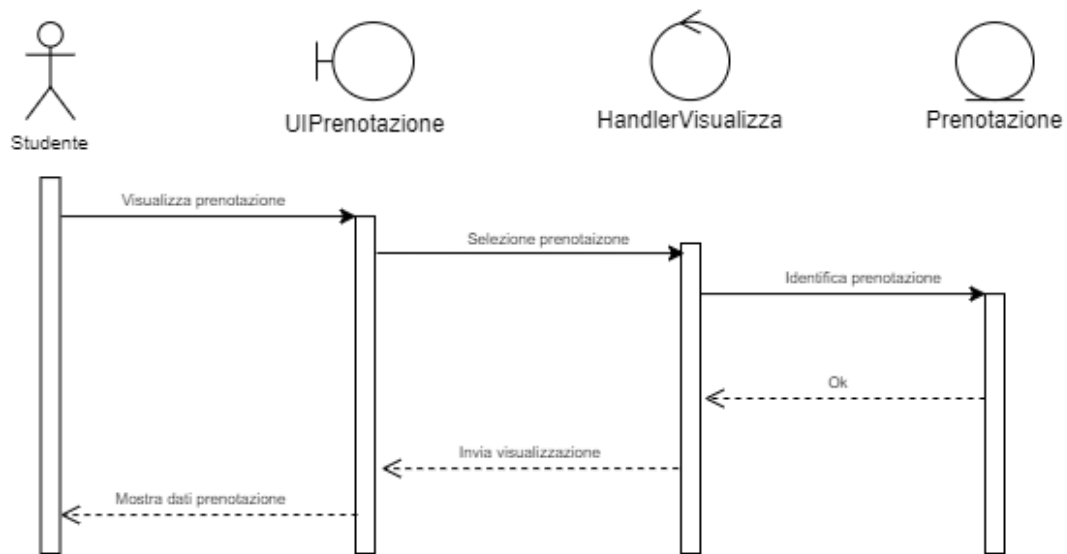


Figura 5.9: Diagramma di sequenza visualizzazione ricevuta.

5.4 Cicli di vita delle classi

Per identificare i cicli di vita delle classi ci si deve riferire agli oggetti che possono cambiare stato a seguito di eventi, in particolare in questo caso si farà riferimento solo alla classe **Prenotazione** (figura 5.10) e alla classe **Aula** (figura 5.2).

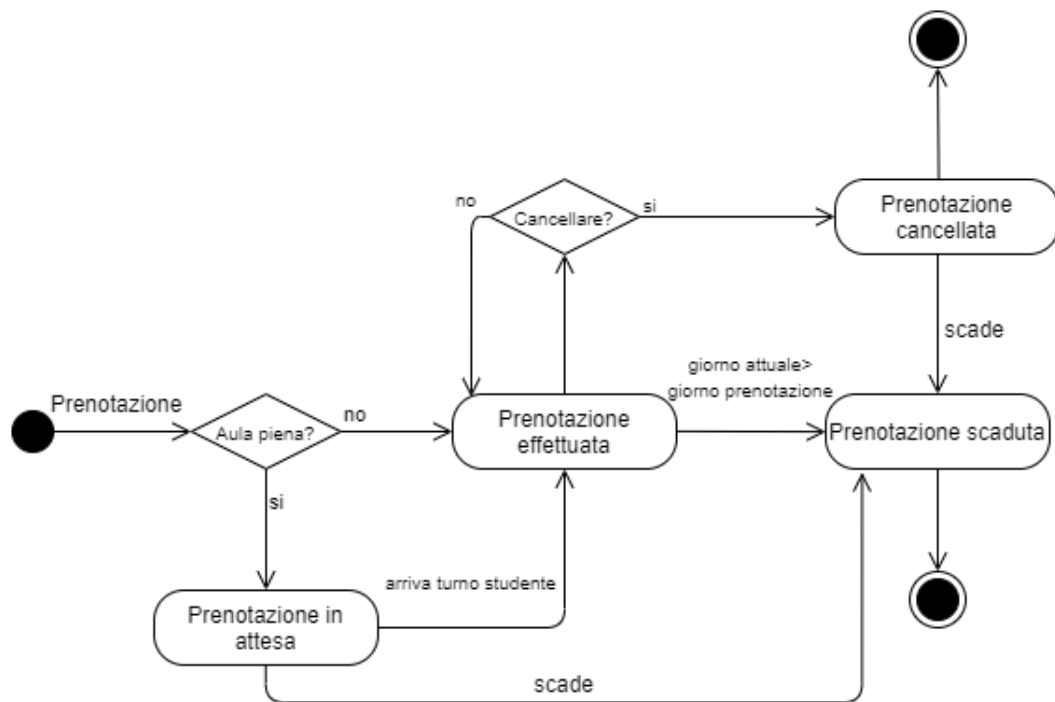


Figura 5.10: Ciclo di vita della classe **Prenotazione**

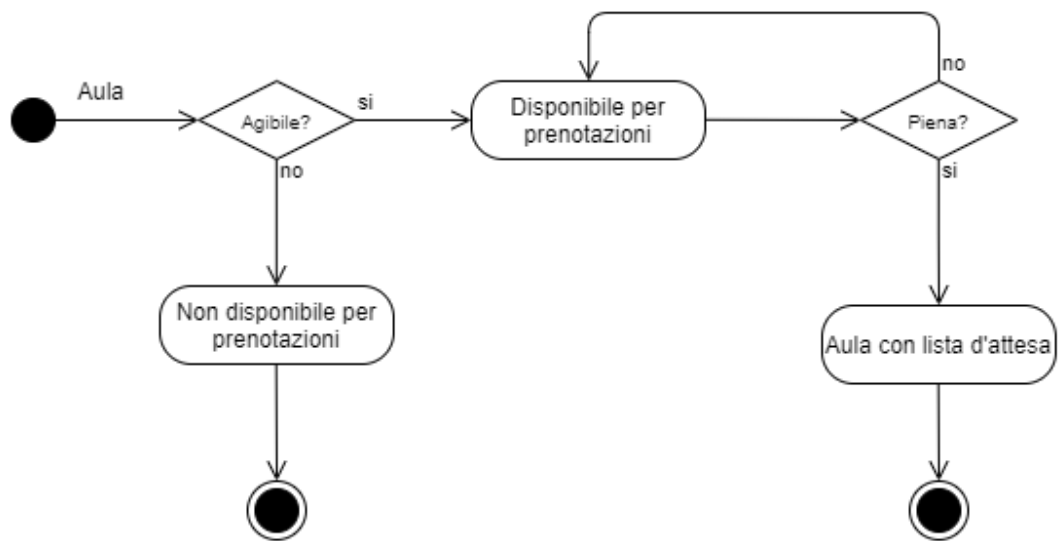


Figura 5.11: Ciclo di vita della classe **Aula**