

PRODIGITX
SAPIENZA

ProdigitX

Progetto di Ingegneria del Software

Studente: Alessio Lucciola

Matricola: 1823638

Email: lucciola.1823638@studenti.uniroma1.it

Indice

Analisi di Contesto e Studio di Fattibilità	2
Contratto	6
Pianificazione Temporale	8
Glossario	15
Specifica dei Requisiti	17
Gestione dei Rischi	24
Specifica dei Casi d'Uso	29
Analisi dei Costi	41
Documento di Analisi	45
Documento di Design	60
Piano dei Test	74

Analisi di Contesto e Studio di Fattibilità

1. Prefazione

Lo scopo di questa prima parte è quella di definire gli obiettivi del sistema descrivendo quello che si andrà a progettare e di mostrare la fattibilità del progetto.

2. Analisi di contesto

2.1. Introduzione

In seguito allo stato di emergenza causato dall'attuale pandemia di Covid-19, l'Università degli Studi di Roma "La Sapienza" è intenzionata a creare un Portale Web che darà agli studenti la possibilità di prenotare un posto in aula nel rispetto della nuova capienza delle aule. Questo sistema prenderà il nome di ProdigitX. Il sistema si andrà a collocare nell'insieme degli strumenti messi a disposizione dell'ateneo per fronteggiare l'emergenza data dalla pandemia di Covid-19, non si esclude però che potrà essere riutilizzando in futuro ad emergenza finita.

2.2. Obiettivi del prodotto

Il sistema sviluppato sarà un'applicazione web e dovrà consentire di svolgere le funzionalità necessarie a garantire la prenotazione di un posto in aula. Il sistema sarà quindi rivolto a tutti gli studenti immatricolati a cui verrà data la possibilità di:

- Effettuare la prenotazione di un posto in aula nel giorno e negli orari desiderati;
- Visualizzare le prenotazioni effettuate;
- Cancellare una prenotazione effettuata;
- Produrre una ricevuta di prenotazione con la possibilità di salvarla;
- Iscriversi alla waiting list nel caso in cui i posti in aula siano terminati;
- Uscire dalla waiting list.

2.3. Interazione con altri sistemi

Per svolgere tutte le funzionalità elencate sopra lo studente dovrà autenticarsi. Le credenziali per effettuare il login su ProdigitX sono le stesse utilizzate su Infostud, per cui nel momento in cui si effettua il login, ProdigitX dovrà comunicare con il sistema Infostud per ricevere informazioni sull'utente che sta tentando di accedere al servizio e mostrare un messaggio di errore nel caso in cui l'autenticazione non vada a buon fine o, altrimenti, accedere al sistema.

Inoltre ProdigitX avrà un servizio di posta elettronica che permetterà di comunicare con lo studente per informarlo su aggiornamenti relativi alle sue prenotazioni (ad es. si è liberato un posto in aula).

2.4. Altri Requisiti e Vincoli

Viene richiesta la soddisfazione di ulteriori vincoli o requisiti:

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. Inoltre la prenotazione ad una lezione si deve necessariamente effettuare la settimana precedente a quella in cui si svolge la lezione e dal lunedì al venerdì, quindi uno studente deve essere in grado di effettuare una prenotazione solo se rispetta tali requisiti.

Il sistema dovrà risultare usabile, implementando un'interfaccia semplice ed intuitiva e permettendo sempre di capire quale operazione si sta svolgendo; veloce, rendendo la consultazione e l'utilizzo di informazioni immediata; e sicuro, evitando l'accesso e la divulgazione di informazioni private ad utenti non autorizzati. Inoltre dovrà essere disponibile sia in lingua italiana che inglese e dovrà essere compatibile con i principali browser (ad es. Chrome, Firefox, etc..).

Nota: Per una descrizione dettagliata dei requisiti si veda il documento “*Specifiche dei Requisiti*”;

3. Studio di Fattibilità

3.1. Background e formulazione del problema

In seguito all'attuale pandemia di Covid-19 lo Stato italiano ha varato delle norme con lo scopo di regolamentare lo svolgimento delle attività formative. L'obiettivo è quello di svolgere, ove possibile, le lezioni in presenza riducendo al minimo la probabilità di contagio di studenti e docenti. In particolare, l'allegato 18 del DPCM del 22 novembre 2020 recita:

“Le Università, sentito il Comitato Universitario Regionale di riferimento, predispongono, in base all'andamento del quadro epidemiologico, piani di organizzazione della didattica e delle attività curriculari, da svolgersi a distanza o in presenza, che tengono conto delle esigenze formative e dell'evoluzione del quadro pandemico territoriale e delle corrispondenti esigenze di sicurezza sanitaria [...]”

Le eventuali attività in presenza consentite si devono svolgere nel rispetto

- delle **linee guida** del Ministero dell'università e della ricerca;
- sulla base del **protocollo** per la gestione di casi confermati e sospetti di COVID-19;”

Fonte: <http://m.flcgil.it/attualita/dpcm-3-novembre-2020-sintesi-interventi-settori-conoscenza.flc#6>

Seguire le lezioni in presenza garantendo il distanziamento sociale comporta varie difficoltà:

1. Bisogna garantire una distanza interpersonale di almeno un metro tra ciascun individuo evitando assembramenti all'interno e all'esterno delle aule;
2. Mantenere la distanza tra individui durante le lezioni comporta una notevole riduzione dei posti in aula;
3. Non conoscere il numero di persone che si recheranno a lezione può portare all'impossibilità di parteciparvi se il numero di posti è già terminato. Questo implica ulteriori problemi come il doversi recare in sede per sapere se si può partecipare o meno alla lezione.

Nota: il documento delle norme messe in atto dall'Università degli Studi di Roma "La Sapienza" può essere visualizzato a questo [link](#);

3.2. Soluzione al problema

Una soluzione al problema è quella di creare un portale web che permetta di prenotare il posto in aula nel rispetto della nuova capienza delle aule.

Questo sistema risolve alcuni problemi precedentemente elencati infatti prenotare un posto in aula permette di sapere in anticipo se recarsi o meno in presenza evitando un inutile viaggio e riducendo al minimo la possibilità di creare assembramenti.

Ci sono anche degli svantaggi: oltre al costo del progetto occorre analizzare quelli che sono i reali problemi dell'applicazione. Uno di questi è che venga a crearsi il fenomeno del cosiddetto "Click Day". Nel momento in cui vengono aperte le prenotazioni delle aule, un elevato numero di studenti potrebbe provare a prenotare un posto provocando lunghe file d'attesa (col rischio di non riuscire a prenotarsi) ed eventuali rallentamenti al portale.

E' già stata trovata una parziale soluzione a questo problema ed è quella di organizzare i turni prenotabili in base alle ultime due cifre del numero di matricola (da 00 a 49 e da 50 a 99) a settimane alterne, in modo da ridurre, ogni settimana, il numero di studenti che possono effettuare una prenotazione.

3.3. Analisi di Mercato

Varie università italiane utilizzano già sistemi simili a ProdigitX infatti l'esistenza di tali portali può confermare l'attuabilità del progetto.

Un esempio è il sistema creato dall'Università degli Studi "Roma Tre" che permette di prenotare il posto in aula direttamente dall'area riservata dello studente.

Fonte: <https://portalestudente.uniroma3.it/infocovid/prenotazione-posto-aula/>

Contratto

1. Prefazione

In questo documento viene esposta una proposta di contratto per permettere al committente di accettare o meno l'accordo.

2. Proposta di contratto

2.1. Introduzione

Il sistema che si vuole realizzare è ProdigitX, un applicativo web che permetterà la prenotazione di un posto in aula. Il sistema viene descritto ad alto livello nel documento "*analisi di contesto e studio di fattibilità*" e i requisiti verranno descritti nel dettaglio nel documento "*specifica dei requisiti*".

2.2. Caratteristiche del sistema

La proposta di contratto è la seguente:

- Il progetto verrà realizzato utilizzando il metodo RUP (*Rational Unified Process*);
- Verrà eseguita l'analisi, il progetto e lo sviluppo del sistema rispettando i requisiti e i vincoli stabiliti dal committente;
- Il sistema verrà realizzato utilizzando un'interfaccia semplice e di facile utilizzo;
- Verrà rilasciata periodicamente della documentazione in modo da favorire il contatto tra membri del gruppo e committente ed approvare con maggiore facilità il lavoro svolto.

2.3. Tempi di realizzazione e costi

Vista l'attuale emergenza data dalla pandemia di Covid-19 e l'immediata necessità di permettere agli studenti di seguire in presenza, il sistema verrà realizzato entro due mesi dall'approvazione delle specifiche.

Il budget previsto è consultabile nel documento "*Analisi dei costi*". Nel budget non è inclusa la manutenzione, nuove versioni o funzionalità aggiuntive ma solo i costi di sviluppo del sistema.

2.4. Divieto di cessione

Il presente contratto è regolato dalla legge italiana e non può essere ceduto, in tutto o in parte, a terze parti, pena la nullità dell'atto di cessione.

2.5. Contatti

Nome	Cognome	Matricola	Email
Alessio	Lucciola	1823638	lucciola.1823638@studenti.uniroma1.it

Pianificazione Temporale

1. Prefazione

Questo documento ha un duplice obiettivo, quello di esporre il modello di processo software utilizzato e mostrare le iterazioni effettuate durante lo sviluppo del sistema.

2. Modello di Processo Software

Il modello di processo utilizzato nello sviluppo di ProdigitX è il *Rational Unified Process* (RUP).

In questo modello di processo, il ciclo di vita del software viene suddiviso in cicli di sviluppo, a loro volta scomposti in fasi. Queste fasi sono:

- *Inception*: Lo scopo iniziale è quello di delineare nel modo più accurato possibile il business case, ossia il ragionamento su cui si basa l'avvio del progetto. In questa fase si utilizzano vari strumenti quali un modello dei casi d'uso, la pianificazione iniziale del progetto e stime su costi e tempi;
- *Elaboration*: Si definisce la struttura complessiva del sistema. A questo punto si devono identificare la maggior parte dei requisiti e si devono effettuare delle stime più realistiche;
- *Construction*: Viene sviluppata buona parte del sistema e viene prodotta una prima *release*. Si inizia quindi la fase di test;
- *Transition*: Vengono effettuati gli ultimi test ed in seguito il prodotto viene rilasciato;

Ogni fase qui descritta viene nuovamente suddivisa in iterazioni, ognuna con una scadenza.

3. Iterazioni

Di seguito l'elenco delle iterazioni effettuate. Di ogni iterazione si identificano:

- Nome: nome dell'iterazione;
- Fase: fase del RUP di cui fa parte l'iterazione;
- Data di inizio/fine: data di inizio e fine dell'iterazione;
- Stato: può assumere 3 valori (pianificata, in corso o conclusa);
- Obiettivi: descrizione dei compiti svolti nell'iterazione;

Iterazione 1:

Nome	It_1
Fase	Inception
Inizio	07/12/2020
Fine	13/12/2020
Stato	Conclusa

Obiettivi	<ul style="list-style-type: none"> • Analisi del problema; • Inizio della stesura del documento “Analisi di contesto e studio di fattibilità”; • Inizio della stesura del documento “Contratto”; • Inizio della stesura del documento “Glossario”; • Verifica dello stato di avanzamento e conferma della pianificazione dell’iterazione 2;
-----------	--

Iterazione 2:

Nome	It_2
Fase	Inception
Inizio	14/12/2020
Fine	20/12/2020
Stato	Conclusa
Obiettivi	<ul style="list-style-type: none"> • Raffinamento del documento “Visione e studio di fattibilità”; • Raffinamento del documento “Contratto”; • Inizio della stesura del documento “Specifica dei Requisiti”; • Inizio della stesura del documento “Pianificazione Temporale”; • Aggiornamento del documento “Glossario”; • Verifica dello stato di avanzamento e conferma della pianificazione dell’iterazione 3; • Riunione con il committente per la verifica dei requisiti;

Iterazione 3:

Nome	It_3
Fase	Inception
Inizio	21/12/2020
Fine	27/12/2020
Stato	Conclusa
Obiettivi	<ul style="list-style-type: none"> • Correzione di eventuali errori riscontrati nella riunione con il committente; • Raffinamento del documento “Visione e studio di fattibilità”; • Raffinamento del documento “Specifica dei Requisiti”; • Inizio della stesura del documento “Analisi dei costi”; • Aggiornamento del documento “Pianificazione Temporale”; • Inizio della stesura del documento “Gestione dei rischi”; • Verifica dello stato di avanzamento e conferma della pianificazione dell’iterazione 4;

	<ul style="list-style-type: none"> • Riunione con il committente per la conferma degli obiettivi e dei requisiti chiave del sistema.
--	---

Tappa 1:

Nome	Ta_1
Fase	Inception
Inizio	27/12/2020
Fine	27/12/2020
Stato	Conclusa
Obiettivi	<ul style="list-style-type: none"> • Stabilire la fattibilità del progetto (Fatto); • Identificare i requisiti chiave (Fatto); • Identificare i rischi per il progetto (Fatto); • Stimare costi e tempi necessari al completamento del progetto (Fatto).

Iterazione 4:

Nome	It_4
Fase	Elaboration
Inizio	28/12/2020
Fine	31/01/2021
Stato	Conclusa
Obiettivi	<ul style="list-style-type: none"> • Stesura del documento "Specifica dei casi d'uso"; • Raffinamento del documento "Analisi dei costi"; • Realizzazione del diagramma UML dei casi d'uso; • Inizio della stesura del documento "Documento di Analisi"; • Inizio della stesura del documento "Documento di Design"; • Inizio della stesura del documento "Piano dei Test"; • Verifica dello stato di avanzamento e conferma della pianificazione dell'iterazione 5; • Riunione con il committente per la conferma degli use case e la specifica dei requisiti completa.

Tappa 2:

Nome	Ta_2
Fase	Elaboration
Inizio	31/12/2020

Fine	31/12/2020
Stato	Conclusa
Obiettivi	<ul style="list-style-type: none"> • Catturare nei casi d'uso buona parte dei requisiti funzionali (fatto); • Stimare concretamente costi, tempi e sforzo necessari al completamento del progetto (fatto); • Creare una base architeturale per il progetto (fatto); • Creare un piano di progetto per la fase di construction (fatto).

Iterazione 5:

Nome	It_5
Fase	Construction
Inizio	01/01/2021
Fine	07/01/2021
Stato	Conclusa
Obiettivi	<ul style="list-style-type: none"> • Analisi, progettazione, creazione dei test e implementazione dei casi d'uso con priorità 1: Effettua Login, Effettua Logout, Effettua Prenotazione; • Verifica dello stato di avanzamento e conferma della pianificazione dell'iterazione 6; • Riunione con il committente per aggiornamento sullo stato di avanzamento del progetto;

Iterazione 6:

Nome	It_6
Fase	Construction
Inizio	08/01/2021
Fine	10/01/2021
Stato	Conclusa
Obiettivi	<ul style="list-style-type: none"> • Correzione di eventuali errori riscontrati nella riunione con il committente; • Analisi, progettazione, creazione dei test e implementazione dei casi d'uso con priorità 2: Visualizza Prenotazioni Confermate, Stampa Ricevuta, Cancella Prenotazione Confermata; • Riunione con il committente per aggiornamento sullo stato di avanzamento del progetto.

Iterazione 7:

Nome	It_7
Fase	Construction
Inizio	11/01/2021
Fine	15/01/2021
Stato	Conclusa
Obiettivi	<ul style="list-style-type: none"> • Correzione di eventuali errori riscontrati nella riunione con il committente; • Analisi, progettazione, creazione dei test e implementazione dei casi d'uso con priorità 3: Visualizza Prenotazioni Non Confermate, Cancella Prenotazione Non Confermata; • Riunione con il committente per aggiornamento sullo stato di avanzamento del progetto.

Tappa 3:

Nome	Ta_3
Fase	Construction
Inizio	15/01/2020
Fine	15/01/2020
Stato	Conclusa
Obiettivi	<ul style="list-style-type: none"> • Completare l'identificazione e la realizzazione dei casi d'uso (fatto); • Completare l'analisi, la progettazione, l'implementazione e i test dei casi d'uso (fatto);

Iterazione 8:

Nome	It_8
Fase	Transition
Inizio	16/01/2021
Fine	20/01/2021
Stato	Corrente
Obiettivi	<ul style="list-style-type: none"> • Creazione della UI per l'interazione con il sistema; • Beta test del sistema;

	<ul style="list-style-type: none"> • Correzione di eventuali problemi riscontrati; • Rilascio del prodotto;
--	---

Tappa 4:

Nome	Ta_4
Fase	Transition
Inizio	20/01/2020
Fine	20/01/2020
Stato	Pianificata
Obiettivi	<ul style="list-style-type: none"> • Completare i beta test; • Completare la correzione dei problemi riscontrati; • Rilasciare il prodotto;

Glossario

1. Prefazione

Questo documento ha lo scopo di elencare le principali parole chiave utilizzate nel progetto e le relative definizioni.

2. Glossario

Di seguito una tabella in cui vengono identificati i principali termini del dominio e la relativa definizione:

Termine	Definizione
Ateneo	Istituto di istruzione superiore. In questo progetto si intende l'Università degli Studi di Roma "La Sapienza";
Aula	Sala che si trova all'interno di un dato edificio dell'ateneo. Ciascuna aula viene identificata con un codice.
Codice Fiscale	Stringa di 16 caratteri che viene usata da tutte le amministrazioni pubbliche in Italia e serve ad identificare il cittadino, il contribuente e tutti gli altri soggetti obbligati ad averlo;
Edificio	Una struttura attualmente in uso dall'ateneo;
Matricola	Un valore numerico associato univocamente ad ogni studente immatricolato presso l'ateneo;
Prenotazione Confermata	Prenotazione di un posto in aula in un determinato giorno e fascia oraria. La prenotazione è confermata per cui l'utente si può recare in aula esibendo la ricevuta.
Prenotazione Non Confermata	Prenotazione di un posto in aula in un determinato giorno e fascia oraria. La prenotazione non è confermata in quanto i posti in aula sono terminati. La prenotazione passa allo stato di confermata se qualcuno cancella la sua prenotazione nella stessa aula, giorno e fascia oraria della prenotazione non confermata.
Ricevuta di prenotazione	Ricevuta di una prenotazione confermata che lo studente può scaricare dal sistema. Attesta che lo studente può partecipare ad una lezione in presenza nell'aula e negli orari indicati in fase di prenotazione.
Studente	Persona iscritta all'ateneo, nonché unico attore del Sistema;
Utente Non Autenticato	Utente che non ha ancora effettuato l'autenticazione;

Waiting List	Lista delle Prenotazioni Non Confermate in un'aula, in un dato giorno e fascia oraria.
--------------	--

Specifica dei Requisiti

1. Prefazione

In questo documento verranno elencati i requisiti funzionali e i principali requisiti non funzionali precedentemente identificati che il sistema dovrà soddisfare.

2. Elenco dei requisiti

2.1. Introduzione

Di seguito viene descritto il formato utilizzato per identificare i requisiti:

- **Requisiti funzionali:** ogni requisito funzionale ha il seguente identificativo “RF_NUM_OP_OGG” dove:
 - RF: indica che si tratta di un requisito funzionale;
 - NUM: valore numerico assegnato univocamente al requisito;
 - OP: operazione che il sistema dovrà effettuare;
 - OGG: oggetto su cui verte il requisito;
- **Requisiti non funzionali:** ogni requisito non funzionale ha il seguente identificativo “RNF_NUM_CT_OGG” dove:
 - RNF: indica che si tratta di un requisito non funzionale;
 - NUM: valore numerico assegnato univocamente al requisito;
 - CT: tipologia di requisito non funzionale (ad esempio “sicurezza”, “usabilità”, etc..);
 - OGG: oggetto su cui verte il requisito;

Ogni requisito (sia funzionale che non funzionale) è accompagnato da un nome che permette di identificarlo in maniera più semplice.

Ogni requisito è composto dalle seguenti sezioni:

- **ID:** identificativo del requisito;
- **Priorità:** indica il grado di importanza del requisito. Questo campo può assumere 4 categorie di priorità:
 - *Must:* Il requisito è essenziale e senza di esso il sistema non può funzionare;
 - *Should:* il requisito non è essenziale ma è desiderabile. Senza tale requisito il sistema può funzionare ma ne risente a livello funzionale;
 - *Could:* il requisito non è essenziale ma è apprezzabile. Senza tale requisito il sistema può funzionare ma ne risente a livello qualitativo;
 - *Would:* il requisito non verrà implementato e potrà essere considerato in aggiornamenti futuri;
- **Mutabilità:** indica la possibilità che il requisito cambi nel tempo. Può essere:
 - *Durevole:* il requisito è stabile nel tempo, non mutabile;

- *Volatile*: il requisito può variare in seguito all'uso o durante lo sviluppo;
- Stato: stringa che indica lo stato del requisito. Può assumere tre valori:
 - *Approvato*: il committente ha visionato il requisito e ritiene che lo si debba realizzare;
 - *Non approvato*: il committente ha visionato il requisito e non ritiene che il sistema debba soddisfarlo pertanto non verrà realizzato;
 - *In sospeso*: il committente deve ancora visionare il requisito;
- Descrizione: descrizione del requisito;

2.2. Requisiti funzionali

Login:

ID	RF_1_Effettua_Login
Priorità	Must
Mutabilità	Durevole
Stato	Approvato
Descrizione	Il sistema dovrà permettere ad uno Studente immatricolato alla Sapienza di autenticarsi nel sistema sfruttando le stesse credenziali che utilizza nei servizi dell'ateneo.

Logout:

ID	RF_2_Effettua_Logout
Priorità	Must
Mutabilità	Durevole
Stato	Approvato
Descrizione	Il sistema dovrà permettere allo Studente di effettuare il logout ritornando allo stato di utente non autenticato.

Effettua Prenotazione:

ID	RF_3_Effettua_Prenotazione
Priorità	Must
Mutabilità	Durevole
Stato	Approvato
Descrizione	Il sistema dovrà permettere allo studente di prenotare un posto in aula. Lo studente si può prenotare solo se rispetta alcuni requisiti (si vedano i vincoli nell'analisi di contesto). Se non vi sono posti

	disponibili allora si dovrà permettere allo studente di iscriversi alla waiting list.
--	---

Visualizzazione Prenotazioni Effettuate:

ID	RF_4_Visualizzazione_PrenotazioniEffettuate
Priorità	Must
Mutabilità	Durevole
Stato	Approvato
Descrizione	Il sistema dovrà permettere allo studente di visualizzare la lista delle prenotazioni da lui effettuate.

Effettua Cancellazione Prenotazione:

ID	RF_5_Effettua_CancellazionePrenotazione
Priorità	Must
Mutabilità	Durevole
Stato	Approvato
Descrizione	Il sistema dovrà permettere allo studente di cancellare una prenotazione, dopo aver visualizzato la lista delle prenotazioni effettuate. Nel momento in cui si cancella una prenotazione, se vi sono studenti nella waiting list, il sistema deve permettere a tali studenti di prenotarsi. Si può cancellare una prenotazione fino al venerdì della settimana precedente in cui si svolge la lezione.

Salva Ricevuta Prenotazione:

ID	RF_6_Salvataggio_RicevutaPrenotazione
Priorità	Must
Mutabilità	Durevole
Stato	Approvato
Descrizione	Il sistema dovrà permettere allo studente di salvare in locale la ricevuta di prenotazione di un posto in aula.

Visualizzazione Iscrizioni Waiting List:

ID	RF_7_Visualizzazione_IscrizioniWaitingList
Priorità	Should

Mutabilità	Durevole
Stato	Approvato
Descrizione	Il sistema dovrà permettere allo studente di visualizzare tutte le sue iscrizioni alle waiting list.

Cancella Iscrizione Waiting List:

ID	RF_8_Effettua_CancellazioneWaitingList
Priorità	Should
Mutabilità	Durevole
Stato	Approvato
Descrizione	Il sistema dovrà permettere allo studente di disiscriversi da una waiting list a cui era iscritto.

2.3. Requisiti non funzionali

Sicurezza:

ID	RNF_1_Sicurezza_Comunicazione
Priorità	Must
Mutabilità	Durevole
Stato	Approvato
Descrizione	Il sistema garantirà la confidenzialità delle comunicazioni (ad es. utilizzando il protocollo HTTPS).

Riservatezza:

ID	RNF_2_Sicurezza_Riservatezza
Priorità	Must
Mutabilità	Durevole
Stato	Approvato
Descrizione	Il sistema garantirà la riservatezza delle informazioni personali degli utenti garantendo così la privacy.

Usabilità Interfaccia:

ID	RNF_3_Usabilità_Interfaccia
----	-----------------------------

Priorità	Should
Mutabilità	Durevole
Stato	Approvato
Descrizione	Il sistema garantirà un'interfaccia intuitiva per permettere a tutti gli utenti di usare senza difficoltà tutte le funzionalità del sistema.

Lingua:

ID	RNF_4_Usabilità_Lingua
Priorità	Should
Mutabilità	Durevole
Stato	Approvato
Descrizione	Il sistema dovrà essere disponibile sia in lingua italiana che in lingua inglese.

Disponibilità Sistema:

ID	RNF_5_Disponibilità_Sistema
Priorità	Must
Mutabilità	Durevole
Stato	Approvato
Descrizione	Il sistema dovrà essere sempre disponibile e garantire il servizio nei periodi in cui si svolgono le lezioni in presenza.

Compatibilità Sistema:

ID	RNF_6_Compatibilità_Sistema
Priorità	Must
Mutabilità	Durevole
Stato	Approvato
Descrizione	Il sistema dovrà essere compatibile con i principali browser (ad es. Google Chrome, Firefox, Microsoft Edge, Opera, etc.).

Gestione dei Rischi

1. Prefazione

In questo documento verrà effettuata la gestione dei rischi. Per rischio si intende un problema che può minacciare un progetto, il software in via di sviluppo o l'organizzazione stessa. In generale, si distinguono tre categorie di rischio:

1. Rischi per il progetto: influiscono sulle risorse e sulla tempistica del progetto;
2. Rischi per il prodotto: influiscono sulla qualità o sulle prestazioni del software che si sta sviluppando;
3. Rischi per l'azienda: influiscono sull'azienda che sta sviluppando o acquistando il software;

L'obiettivo è quindi quello di gestire eventuali imprevisti, calcolarne i danni e cercare di limitarli il più possibile tenendo conto di eventuali aumenti di costo e ritardi sulla pianificazione temporale del progetto.

2. Modello RMMM

Per la valutazione e la gestione dei rischi si utilizza il modello RMMM (Risk, Mitigation, Monitoring, Management). Qui di seguito viene mostrata la struttura delle tabelle dove ogni tabella rappresenta un rischio:

- Nome: nome del rischio;
- ID: identificatore univoco del rischio;
- Probabilità: indica la probabilità che un rischio possa verificarsi. Può assumere i seguenti valori:
 - *Molto Alta*: la probabilità che si verifichi è sopra al 75%;
 - *Alta*: la probabilità che si verifichi è compresa tra il 50 e il 75%;
 - *Moderata*: la probabilità che si verifichi è compresa tra il 25 e il 50%;
 - *Bassa*: la probabilità che si verifichi è compresa tra il 10 e il 25%;
 - *Molto Bassa*: la probabilità che si verifichi è sotto al 10%;
- Effetti: indica la gravità degli effetti sul progetto. Può assumere i seguenti valori:
 - Catastrofici: minacciano la sopravvivenza stessa del progetto;
 - Gravi: provocano notevoli aumenti sui costi e ritardi e, se non gestiti correttamente, possono minacciare la sopravvivenza del progetto;
 - Tollerabili: provocano aumenti sui costi e ritardi entro i margini consentiti;
 - Insignificanti: non hanno conseguenze rilevanti e in genere non comportano aumenti sui costi o sui tempi di completamento del progetto;
- Descrizione: fornisce una breve descrizione del rischio;

- Mitigation: fornisce un modo per cercare di impedire il verificarsi del rischio adottando alcune strategie;
- Monitoring: fornisce delle tecniche di osservazione che permettono di accorgersi del rischio in tempo utile;
- Management: fornisce delle proposte per gestire il rischio nel caso in cui si verifichi;

3. Elenco dei Rischi

Nome	Ritardi
ID	Risk_01
Probabilità	Alta
Effetti	Gravi
Descrizione	Presenza di ritardi causati da una cattiva gestione del tempo, da una sottostima del tempo necessario a completare il sistema o da imprevisti che rallentano lo sviluppo del progetto.
Mitigation	Anticipare le scadenze in base ai tempi stabiliti nella pianificazione temporale del progetto permette di avere più tempo per gestire eventuali ritardi.
Monitoring	Controllare periodicamente che i tempi stabiliti nella pianificazione temporale del progetto siano rispettati.
Management	Se non è possibile rispettare i tempi stabiliti per un'iterazione, si possono posticipare delle attività nelle iterazioni successive in modo da rispettare le scadenze. Se il lavoro inizia ad accumularsi si può contattare il committente per concordare una nuova data di rilascio.

Nome	Requisiti Errati
ID	Risk_02
Probabilità	Bassa
Effetti	Catastrofici
Descrizione	Le richieste espresse dal committente possono essere mal interpretate rendendo i requisiti errati.
Mitigation	Far controllare frequentemente la specifica dei requisiti al committente in modo da correggere preventivamente eventuali errori presenti nei requisiti.

Monitoring	Effettuare riunioni periodiche con il committente per aggiornarlo su ogni cambiamento effettuato ai requisiti e progettare il lavoro futuro.
Management	Cercare di correggere i requisiti errati limitando gli effetti sugli altri requisiti.

Nome	Requisiti Modificati
ID	Risk_03
Probabilità	Moderata
Effetti	Gravi
Descrizione	I requisiti vengono modificati dal committente durante lo sviluppo.
Mitigation	Discutere con il committente le richieste meno chiare, evitando di dover aggiungere o modificare i requisiti nelle riunioni successive.
Monitoring	Fare continue riunioni con il committente in modo da evitare incomprensioni sulle richieste e quindi un'eventuale modifica dei requisiti in corso d'opera.
Management	Discutere le modifiche con il committente in maniera approfondita, concordare una nuova scadenza e riorganizzare la pianificazione temporale del progetto.

Nome	Personale indisposto
ID	Risk_04
Probabilità	Moderata
Effetti	Gravi
Descrizione	Alcuni membri del team potrebbero assentarsi per lunghi periodi di tempo.
Mitigation	Nella pianificazione temporale del progetto si tiene conto di eventuali assenze (per malattia o altri motivi) di ogni membro del team. Ogni membro è tenuto ad informare in anticipo di eventuali assenze.
Monitoring	Le assenze improvvise non sono monitorabili.
Management	Riorganizzare il team in modo che ci sia una maggiore sovrapposizione di lavoro e ogni membro del team possa

	capire il lavoro degli altri. Qualora non fosse possibile, si dovrà organizzare una riunione per valutare l'impatto sul progetto e riprogrammare le scadenze.
--	---

Nome	Complessità sottovalutata
ID	Risk_05
Probabilità	Bassa
Effetti	Catastrofici
Descrizione	Porre poca attenzione nella raccolta dei requisiti potrebbe portare a una sottovalutazione della complessità del sistema.
Mitigation	Dare particolare attenzione all'attività di analisi in modo che sia coerente con quello che si andrà a progettare e rispetti la reale complessità del sistema.
Monitoring	Revisionare periodicamente il progetto in modo da scovare delle complessità con maggiore probabilità e quindi poterle gestire in tempo.
Management	Cercare di negoziare i requisiti con il committente per ridurre la complessità del sistema. Se ciò non fosse possibile si deve cercare di distribuire correttamente il lavoro nelle iterazioni successive evitando ritardi.

Nome	Problemi con il database
ID	Risk_06
Probabilità	Bassa
Effetti	Catastrofici
Descrizione	Il database potrebbe presentare malfunzionamenti o non essere in grado di gestire il numero di transazioni.
Mitigation	Sviluppare dei test per verificare il corretto funzionamento del database.
Monitoring	Effettuare dei test periodici su ogni funzionalità sviluppata tenendo conto di tutte le possibili situazioni come i momenti di picco.
Management	Organizzare una riunione con i membri del team per pianificare la risoluzione dei problemi con il database. Questo potrebbe comportare dei ritardi e quindi una nuova pianificazione temporale, oltre che ad un possibile aumento dei costi.

Specifica dei Casi d'Uso

1. Prefazione

In questo documento vengono elencati ed analizzati nel dettaglio gli attori ed i casi d'uso del sistema che realizzano i requisiti funzionali elencati nel documento precedente. Si descrivono in maniera dettagliata le modalità di interazione tra le varie entità ed il sistema stesso, esibendo anche un diagramma UML.

2. Elenco degli attori

2.1. Introduzione

Ogni attore del sistema viene rappresentato da una tabella con le seguenti sezioni:

- **ID**: stringa univoca che identifica l'attore. E' del tipo "A_N" dove:
 - A: indica che si tratta di un attore;
 - N: valore numerico associato univocamente all'attore;
- **Nome**: nome dell'attore che permette di identificare il suo ruolo;
- **Genitore**: ID dell'attore genitore (se l'attore non ha un genitore allora assume il valore "/");
- **Tipo**: tipologia dell'attore. Può assumere i seguenti valori: "Primario", "Secondario", "Di supporto";
- **Classe**: classificazione dell'attore. Può assumere i seguenti valori: "Umano", "Sistema";
- **Descrizione**: descrizione del ruolo dell'attore;

2.2. Elenco degli attori

Gli attori che interagiscono con il sistema sono i seguenti:

- Utente non autenticato;
- Studente;
- Infostud;
- Posta elettronica;

ID	A_1
Nome	Utente non autenticato
Genitore	/
Tipo	Primario
Classe	Umano
Descrizione	Utente non ancora autenticato, non può svolgere alcuna funzionalità se non visualizzare la pagina di login.

ID	A_2
Nome	Studente
Genitore	/
Tipo	Primario
Classe	Umano
Descrizione	Utente autenticato al sistema. L'unico attore umano del sistema rappresenta uno Studente.

ID	A_3
Nome	Infostud
Genitore	/
Tipo	Di Supporto
Classe	Sistema
Descrizione	Il Sistema vi comunicherà nel momento in cui un utente non autenticato prova ad effettuare il login. Fornisce l'esito del login.

ID	A_4
Nome	Posta Elettronica
Genitore	/
Tipo	Di Supporto
Classe	Sistema
Descrizione	Sistema di posta elettronica utilizzato per inviare informazioni relative alle prenotazioni agli utenti del sistema.

3. Casi d'uso

3.1. Introduzione

Il caso d'uso è una tecnica usata per effettuare in maniera esaustiva e non ambigua, la raccolta dei requisiti al fine di produrre software di qualità. Essa consiste nel valutare ogni requisito focalizzandosi sugli attori che interagiscono col sistema, valutandone le varie interazioni.

Ogni caso d'uso viene rappresentato da una tabella con le seguenti sezioni:

- *ID*: stringa univoca che identifica il caso d'uso. E' del tipo "CU_N" dove:
 - CU: indica che si tratta di un caso d'uso;
 - N: valore numerico associato univocamente al caso d'uso;
- *Nome*: nome del caso d'uso;
- *Sottosistema*: sottosistema di cui fa parte il caso d'uso;
- *Priorità*: priorità nella realizzazione del caso d'uso. Può assumere un valore da 1 (alta) a 3 (bassa);
- *Attori*: lista degli attori coinvolti nel caso d'uso;
- *Descrizione*: descrizione del caso d'uso;
- *Precondizioni*: condizioni che devono essere vere prima dell'esecuzione del caso d'uso e che permettono l'accesso al caso d'uso;
- *Flusso Principale*: sequenza di azioni svolte dal sistema e dagli attori coinvolti nel caso d'uso nello scenario di successo;
- *Postcondizioni*: condizioni che devono essere vere dopo l'esecuzione del flusso principale del caso d'uso;
- *Flusso Alternativo n*: sequenza di azioni svolte dal sistema e dagli attori coinvolti nel caso d'uso in scenari che si diramano dal flusso principale. Il valore "n" identifica il numero del flusso alternativo di uno stesso caso d'uso;

3.1. Elenco dei casi d'uso

ID	CU_1
Nome	Effettua Login
Sottosistema	Gestione Accesso
Priorità	1
Attori	Utente non autenticato, Infostud
Descrizione	L'Utente non autenticato effettua il login con lo stesso account utilizzato sui servizi dell'ateneo e quindi utilizzando le stesse credenziali.
Precondizioni	Il Sistema non ha una sessione aperta relativa ad uno Studente.
Flusso Principale	<ol style="list-style-type: none"> 1. L'Utente non autenticato richiede il fare il login collegandosi alla pagina web di ProdigitX; 2. Il Sistema richiede le credenziali dell'utente non autenticato (matricola e password) attraverso il modulo di login; 3. L'Utente non autenticato inserisce le credenziali nei relativi moduli; 4. L'Utente non autenticato conferma i dati inseriti; 5. Il Sistema comunica con Infostud per verificare che i dati

	inseriti corrispondano ad un profilo esistente; 6. Infostud conferma che le credenziali inserite sono corrette; 7. Il Sistema indirizza l'Utente non autenticato alla pagina principale;
Postcondizioni	L'Utente è ora autenticato ed ha una sessione aperta come Studente.
Flusso Alternativo 1	Si riprende dal punto 5: 1. Il Sistema verifica che non vi sono studenti che corrispondono alla matricola inserita. Il sistema genera un errore; 2. Il Sistema riparte dal punto 2 del flusso principale.
Flusso Alternativo 2	Si riprende dal punto 5: 1. Il Sistema rifiuta la credenziali e segnala l'errore; 2. Il Sistema riparte dal punto 2 del flusso principale;

ID	CU_2
Nome	Effettua Logout
Sottosistema	Gestione Accesso
Priorità	1
Attori	Studente
Descrizione	Lo Studente effettua il logout.
Precondizioni	Il Sistema ha una sessione aperta relativa allo Studente.
Flusso Principale	1. Lo Studente chiede al sistema di effettuare il logout; 2. Il Sistema disconnette lo Studente.
Postcondizioni	Il Sistema non ha una sessione aperta relativa allo Studente.

ID	CU_3
Nome	Effettua Prenotazione
Sottosistema	Gestione Prenotazioni
Priorità	1
Attori	Studente
Descrizione	Lo Studente effettua la prenotazione di un posto in aula. Si chiede di selezionare il codice dell'edificio, il codice dell'aula, il

	giorno e la fascia oraria di interesse. Nell'effettuare una prenotazione il sistema deve verificare che vi sia almeno un posto in aula, che nel giorno corrente lo studente possa prenotare in base al turno di cui fa parte e che non sia stata già effettuata la medesima prenotazione (stessa aula e stesso giorno/orario). Se i controlli vanno a buon fine si istanzia una nuova prenotazione confermata. Se non vi sono posti in aula allora si deve permettere allo Studente di iscriversi alla waiting list istanziando una nuova prenotazione non confermata.
Precondizioni	1. Il Sistema ha una sessione aperta relativa ad uno Studente.
Flusso Principale	<ol style="list-style-type: none"> 1. Lo Studente chiede al Sistema di avviare la procedura relativa alla prenotazione di un posto in aula; 2. Il Sistema controlla che nel giorno corrente le prenotazioni sono attive per il turno dello Studente; 3. Il Sistema mostra la lista degli edifici; 4. Lo Studente seleziona l'edificio in cui si trova l'aula che vuole prenotare; 5. Il Sistema risponde mostrando la lista delle aule nell'edificio selezionato; 6. Lo Studente seleziona l'aula di interesse; 7. Il Sistema risponde mostrando un form in cui selezionare giorno e fascia oraria in cui si vuole prenotare un posto in aula. Vengono mostrati solo i giorni della settimana successiva; 8. Lo Studente seleziona il giorno e la fascia oraria e conferma la scelta; 9. Il Sistema chiede conferma della prenotazione del posto in aula; 10. Lo Studente conferma la scelta; 11. Il Sistema controlla che lo Studente non abbia già effettuato la medesima prenotazione; 12. Il Sistema controlla che vi siano posti liberi in aula; 13. Il Sistema conferma la prenotazione; 14. Si ritorna alla homepage;
Postcondizioni	Viene generata una prenotazione confermata.
Flusso Alternativo 1	<p>Si riprende dal punto 2:</p> <ol style="list-style-type: none"> 1. Il Sistema verifica che lo Studente non fa parte del turno corretto oppure sta effettuando la prenotazione il sabato o la domenica; 2. L'operazione di prenotazione di un posto in aula viene annullata, si ritorna alla homepage;
Flusso Alternativo 2	<p>Si riprende dal punto 10:</p> <ol style="list-style-type: none"> 1. Lo Studente non conferma la scelta; 2. Si ritorna al punto 7 del flusso principale;

Flusso Alternativo 3	<p>Si riprende dal punto 11:</p> <ol style="list-style-type: none"> 1. Il Sistema verifica che lo Studente ha già prenotato la lezione; 2. Viene mostrato un messaggio di errore; 3. Si ritorna al punto 7 del flusso principale;
Flusso Alternativo 4	<p>Si riprende dal punto 12:</p> <ol style="list-style-type: none"> 1. Il Sistema verifica che non ci sono più posti liberi in aula; 2. Il Sistema chiede allo Studente di iscriversi alla Waiting List; 3. Lo studente conferma; 4. Il Sistema genera una prenotazione non confermata relativa allo studente; 5. Si ritorna al punto 3 del flusso principale. <p>Postcondizione: Viene generata una prenotazione non confermata.</p> <p>Vi è un ulteriore flusso alternativo incapsulato:</p> <ol style="list-style-type: none"> 1. Si riparte dal punto 2 del flusso alternativo 4; 2. Lo studente non conferma la volontà di iscriversi alla waiting list; 3. Si ritorna al punto 7 del flusso principale;

ID	CU_4
Nome	Visualizza Prenotazioni Confermate
Sottosistema	Gestione Prenotazioni
Priorità	2
Attori	Studente
Descrizione	Lo Studente visualizza le prenotazioni confermate attualmente attive (il giorno corrente precede il giorno della prenotazione).
Precondizioni	Il Sistema ha una sessione aperta relativa ad uno Studente.
Flusso Principale	<ol style="list-style-type: none"> 1. Lo Studente chiede al sistema di poter visualizzare le prenotazioni confermate; 2. Il Sistema risponde mostrando la lista delle prenotazioni confermate attualmente attive;
Postcondizioni	/
Flusso Alternativo 1	<p>Si riparte dal punto 2:</p> <ol style="list-style-type: none"> 1. Non vi sono prenotazioni confermate attualmente attive. Viene mostrata una lista vuota;

ID	CU_5
Nome	Salva Ricevuta
Sottosistema	Gestione Prenotazioni
Priorità	2
Attori	Studente
Descrizione	Lo Studente salva in locale la ricevuta di prenotazione in cui compaiono varie informazioni quali aula, matricola, data/fascia oraria della prenotazione e informazioni sullo studente quali nome, cognome, matricola e codice fiscale.
Precondizioni	<ol style="list-style-type: none"> 1. Il Sistema ha una sessione aperta relativa ad uno Studente; 2. Lo Studente ha almeno una prenotazione confermata attualmente attiva (il giorno corrente precede il giorno della prenotazione)
Flusso Principale	<ol style="list-style-type: none"> 1. Lo Studente chiede di poter salvare la ricevuta di prenotazione; 2. Il Sistema risponde scaricando la ricevuta di prenotazione;
Postcondizioni	/

ID	CU_6
Nome	Cancella Prenotazione Confermata
Sottosistema	Gestione Prenotazioni
Priorità	2
Attori	Studente, Posta Elettronica
Descrizione	Lo Studente cancella la prenotazione confermata. Il Sistema recupera la lista degli Studenti iscritti alla waiting list (prenotazioni non confermate) relativa a prenotazioni nella stessa aula e nello stesso giorno e fascia oraria della prenotazione cancellata in ordine crescente per data di richiesta. Viene confermata la prenotazione del primo studente nella lista e gli viene inviata una email per informarlo che la sua richiesta di prenotazione è stata confermata (la relativa prenotazione non confermata viene quindi cancellata).
Precondizioni	<ol style="list-style-type: none"> 1. Il Sistema ha una sessione aperta relativa ad uno Studente; 2. Lo Studente ha almeno una prenotazione confermata

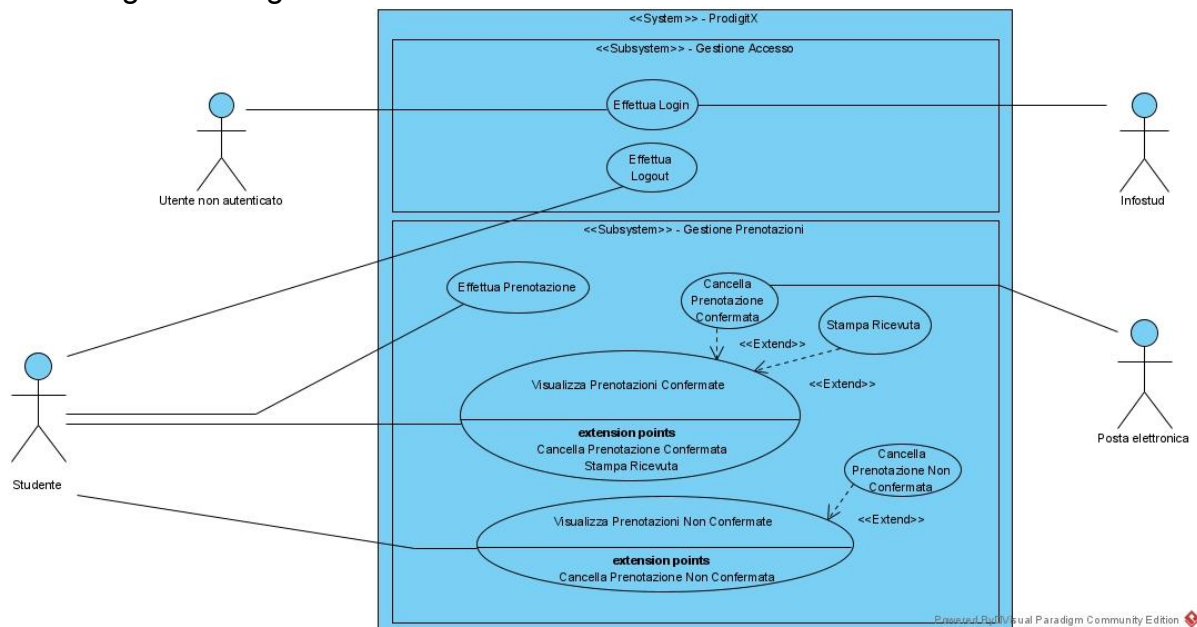
	attualmente attiva (il giorno corrente precede il giorno della prenotazione).
Flusso Principale	<ol style="list-style-type: none"> 1. Lo Studente chiede di poter eliminare una prenotazione attiva; 2. Il Sistema chiede una conferma allo Studente; 3. Lo Studente conferma la volontà di eliminare la prenotazione; 4. Il Sistema cancella la prenotazione; 5. Il Sistema recupera la lista degli Studenti iscritti alla waiting list ossia la lista delle prenotazioni non confermate nella stessa aula e nello stesso giorno e fascia oraria della prenotazione cancellata ordinata in ordine crescente per data di richiesta; 6. Il Sistema estrae la prenotazione del primo Studente nella lista e la conferma; 7. Il Sistema rimuove lo Studente dalla waiting list eliminando la relativa prenotazione non confermata; 8. Il Sistema invia una email allo Studente per informarlo che la sua prenotazione è stata confermata.
Postcondizioni	<ol style="list-style-type: none"> 1. La prenotazione cancellata viene rimossa dalla lista delle prenotazioni effettuate dello Studente; 2. Viene rimossa la prenotazione non confermata dello Studente nella waiting list che viene aggiunta nella sua lista delle prenotazioni confermate.
Flusso Alternativo 1	<p>Si riparte dal punto 2:</p> <ol style="list-style-type: none"> 1. Lo Studente non conferma la volontà di eliminare la prenotazione; 2. Il Sistema annulla la cancellazione della prenotazione e si torna alla visualizzazione della lista delle prenotazioni confermate;
Flusso Alternativo 3	<p>Si riparte dal punto 1:</p> <ol style="list-style-type: none"> 1. Il Sistema genera un errore perché non è più possibile cancellare la prenotazione (una prenotazione confermata si può cancellare solo dal lunedì al venerdì della settimana precedente a quella della prenotazione); 2. Si ritorna alla visualizzazione delle prenotazioni confermate.
Flusso Alternativo 3	<p>Si riparte dal punto 8:</p> <ol style="list-style-type: none"> 1. La lista è vuota quindi non vi sono Studenti nella waiting list; 2. Il Sistema termina la cancellazione della prenotazione.
ID	CU_7

Nome	Visualizza Prenotazioni Non Confermate
Sottosistema	Gestione Prenotazioni
Priorità	3
Attori	Studente
Descrizione	Lo Studente visualizza le prenotazioni non confermate attualmente attive (il giorno corrente precede il giorno della prenotazione).
Precondizioni	Il Sistema ha una sessione aperta relativa ad uno Studente.
Flusso Principale	<ol style="list-style-type: none"> 1. Lo Studente chiede al sistema di poter visualizzare le prenotazioni non confermate; 2. Il Sistema risponde mostrando la lista delle prenotazioni non confermate attualmente attive;
Postcondizioni	/
Flusso Alternativo 1	<p>Si riparte dal punto 2:</p> <ol style="list-style-type: none"> 1. Non vi sono prenotazioni non confermate attualmente attive. Viene mostrata una lista vuota.

ID	CU_8
Nome	Cancella Prenotazione Non Confermata
Sottosistema	Gestione Prenotazioni
Priorità	3
Attori	Studente
Descrizione	Lo Studente cancella la prenotazione non confermata uscendo dalla relativa waiting list.
Precondizioni	<ol style="list-style-type: none"> 1. Il Sistema ha una sessione aperta relativa ad uno Studente; 2. Lo Studente ha almeno una prenotazione non confermata attualmente attiva (il giorno corrente precede il giorno della prenotazione).
Flusso Principale	<ol style="list-style-type: none"> 1. Lo Studente chiede di poter eliminare una prenotazione non confermata; 2. Il Sistema chiede una conferma allo Studente; 3. Lo Studente conferma la volontà di eliminare la prenotazione; 4. Il Sistema cancella la prenotazione non confermata;

	5. Si torna alla visualizzazione della lista delle prenotazioni non confermate;
Postcondizioni	1. La prenotazione cancellata viene rimossa dalla lista delle prenotazioni non confermate effettuate dallo Studente;
Flusso Alternativo 1	<p>Si riparte dal punto 2:</p> <ol style="list-style-type: none"> 1. Lo Studente non conferma la volontà di eliminare la prenotazione; 2. Il Sistema annulla la cancellazione della prenotazione non confermata e si torna alla visualizzazione della lista delle prenotazioni non confermate;

Qui di seguito il diagramma UML dei casi d'uso:



Analisi dei Costi

1. Prefazione

In questo documento verrà svolta l'analisi dei costi per la realizzazione del sistema ProdigitX.

2. Stima dei costi

2.1. Introduzione

Per effettuare una stima dei costi software verrà utilizzato il metodo UCP (*Use Case Points*), un modo semplice e pratico per ottenere delle stime sufficientemente accurate della dimensione del progetto. Questo metodo prevede il calcolo dei seguenti elementi:

- Unadjusted Use Case Points (UUCP): il valore quantitativo grezzo relativo alle funzionalità che devono essere esposte dall'applicazione;
- Technical Complexity Factor (TCF): una misura della complessità tecnologica dell'applicazione da realizzare;
- Environmental Complexity Factor (ECF): una misura della complessità legata al contesto in cui l'applicazione viene sviluppata;
- Productivity Factor (PF): coefficiente che permette la conversione da generici Use Case Points a ore uomo.

2.2. Unadjusted Use Case Points

Si stima ora la complessità degli attori e dei casi d'uso.

Gli attori vengono classificati su una scala di 3 valori:

- Simple (Peso 1): L'attore è un sistema esterno che utilizza una API;
- Average (Peso 2): L'attore è un sistema esterno che utilizza un protocollo (ad es. TCP/IP, FTP, SMTP, HTTP);
- Complex (Peso 3): L'attore è un utente che utilizza una GUI;

La stima del costo degli attori è data dalla seguente formula:

$$UAW = (NoSimpleActors \times 1) + (NoAvgActors \times 2) + (NoComplexActors \times 3)$$

Di seguito la tabella con i pesi di ogni attore:

Attore	Peso
Utente non autenticato	3
Studente	3
Infostud	2
Posta Elettronica	2
UAW	10

I casi d'uso vengono classificati su una scala di 3 valori:

- Simple (Peso 5): Un numero di transazioni compreso tra 1 e 3;
- Average (Peso 10): Un numero di transazioni compreso tra 4 e 7;
- Complex (Peso 15): Un numero di transazioni maggiore o uguale ad 8;

La stima del costo dei casi d'uso è data dalla seguente formula:

$$UUCW = (NoSimpleUC \times 5) + (NoAvgUC \times 10) + (NoComplexUC \times 15)$$

Di seguito la tabella con i pesi di ogni caso d'uso:

Caso d'uso	Peso
Effettua Login	5
Effettua Logout	5
Effettua Prenotazione	15
Visualizza Prenotazioni Confermate	5
Stampa Ricevuta	5
Cancella Prenotazione Confermata	10
Visualizza Prenotazioni Non Confermate	5
Cancella Prenotazione Non Confermata	5
UUCW	55

2.2. Technical Complexity Factor

Si stima ora la complessità tecnica del sistema. Questo fattore è determinato assegnando un valore compreso fra 0 (irrilevante) e 5 (essenziale) a 13 aspetti tecnici prefissati ad ognuno dei quali viene associato un peso. Il TCF è dato dalla seguente formula:

$$TCF = 0.6 + (TF \div 100)$$

dove

$$TF = \sum (i=1, n=13) FactorWeight(i) \times FactorScore(i)$$

Di seguito la tabella degli aspetti tecnici presi in considerazione nel TCF con peso e valore che gli viene assegnato:

Caso d'uso	Valore	Peso
Distributed System	2	4
Response Time	1	5

End-user Efficiency	1	5
Internal Processing Complexity	1	1
Code Reusability	1	3
Easy to Install	0.5	1
Easy to Use	0.5	5
Portability to Other Platforms	2	4
System Maintenance	1	2
Parallel Processing	1	3
Security Features	1	4
Access for Third Parties	1	0
End User Training	1	0
TF		42
TCF		1.02

2.3. Environmental Complexity Factor

Questo fattore è determinato assegnando un valore tra 0 (nessuna esperienza) e 5 (esperto) a 8 aspetti ambientali, ad ognuno dei quali va associato un peso. L'ECF è dato dalla seguente formula:

$$ECF = 1.4 + (-0.03 \times EF)$$

dove

$$EF = \sum (i=1, n=8) \text{FactorWeight}(i) \times \text{FactorScore}(i)$$

Di seguito la tabella degli aspetti ambientali presi in considerazione nell'ECF con peso e valore che gli viene assegnato:

Caso d'uso	Valore	Peso
Familiarity with Development Process Used	0.5	1.5
Application Experience	0.5	0.5
Object-oriented Experience of Team	1	5

Lead Analyst Capability	0.5	5
Motivation of the Team	1	5
Stability of Requirements	2	2
Part-time staff	-1	0
Difficult Programming Language	-1	1
EF		-17.75
EFC		1.9325

2.4. Calcolo UCP

Il valore finale degli Use Case Points (UCP) è dato dalla seguente formula:

$$UCP = (UAW + UUCW) * TCF * ECF = (10 + 55) * 1.02 * 1.9325 = 128,12475$$

2.5. Stima dello sforzo

Una volta calcolata la dimensione del progetto si può calcolare lo sforzo necessario al suo completamento. Stimando che per ogni UCP siano necessarie 25 ore/uomo allora lo sforzo totale è dato dalla seguente formula:

$$EE = UCP * (ore/UCP) = 128,12475 * 25 = 3203.118 \text{ ore/uomo}$$

Calcolando che il lavoro venga svolto da un team di cinque persone, si prevede di terminare il lavoro entro 2 mesi, includendo anche eventuali ritardi.

2.6. Stima del costo

Calcolando uno stipendio di 20€/ora si stima un costo del progetto pari a 64062,26€ circa.

Documento di Analisi

1. Prefazione

In questo documento si andranno ad analizzare la specifica del sistema per estrarre le classi di analisi descritte dalle schede CRC, verranno poi identificati i package di analisi e si descriveranno le relazioni tra le classi di analisi mediante diagrammi *entity*, *boundary* e *control*.

2. Analisi Nomi-Verbi

Un metodo utilizzato per estrarre le classi di analisi è l'analisi nomi-verbi. Per effettuare l'analisi verranno prese in considerazione le descrizioni degli use-case presenti nel documento "specificazione dei casi d'uso". Si utilizzano le seguenti convenzioni:

- Una **sottolineatura rossa** indica un nome, ossia un possibile candidato a diventare una classe o un attributo;
- Una **sottolineatura verde** indica un verbo, ossia un possibile candidato a diventare una responsabilità di classe;
- Una **sottolineatura blu** indica un predicato, ossia un possibile candidato a diventare un attributo;

Di seguito l'analisi nomi-verbi sugli use-case:

1. L'**Utente non autenticato** **effettua** il login con lo stesso account utilizzato sui servizi dell'ateneo e quindi utilizzando le stesse **credenziali**;
2. Lo **Studente** **effettua** il logout;
3. Lo **Studente** **effettua** la prenotazione di un posto in aula. Si chiede di selezionare il **codice dell'edificio**, il **codice dell'aula**, il **giorno** e la **fascia oraria** di interesse. Nell'effettuare una prenotazione il sistema deve **controllare** che vi sia almeno un **posto** in **aula**, che nella **giorno** corrente lo **studente** possa prenotare in base al **turno** di cui fa parte e che non sia stata già **effettuata la medesima prenotazione** (stessa **aula** e stesso **giorno/orario**). Se i controlli vanno a buon fine si **istanzia una nuova prenotazione confermata**. Se non vi sono **posti** in **aula** allora si deve permettere allo **Studente** di **iscriversi alla waiting list** istanziando una nuova **prenotazione non confermata**.
4. Lo **Studente** **visualizza** le **prenotazioni confermate** attualmente attive (il **giorno** corrente precede il **giorno della prenotazione**).
5. Lo **Studente** **stampa** la **ricevuta di prenotazione** in cui compaiono varie informazioni quali **aula**, **matricola**, **data/fascia oraria** della **prenotazione confermata** e informazioni sullo **studente** quali **nome**, **cognome**, **matricola** e **codice fiscale**;
6. Lo **Studente** **cancella** la **prenotazione confermata**. Il Sistema **recupera** la lista degli **Studenti iscritti alla waiting list** (**prenotazioni non confermate**) relativa a prenotazioni nella stessa **aula** e nello stesso **giorno** e **fascia oraria** della **prenotazione cancellata** in ordine crescente per data di richiesta. **Viene confermata la prenotazione** del primo **studente** nella lista e gli viene **inviata una email** per informarlo dell'avvenuta prenotazione.
7. Lo **Studente** **visualizza** le **prenotazioni non confermate** attualmente attive (il

- giorno corrente precede il giorno della prenotazione).
8. Lo **Studente** **cancella** la **prenotazione non confermata** uscendo dalla relativa waiting list.

Si procede ora alla scelta di quali classi mantenere. Di ogni classe verranno elencate le responsabilità e gli attributi associati.

3. Schede CRC

3.1. Introduzione

Le schede CRC (Class, Responsibility, Collaboration) sono uno strumento utilizzato per descrivere le classi di analisi ricavate con l'analisi nomi-verbi. Ogni scheda CRC viene rappresentata come una tabella in cui sono presenti le seguenti sezioni:

- Nome: nome della classe;
- Descrizione: descrizione della classe;
- Superclassi: nomi di eventuali superclassi della classe;
- Sottoclassi: nomi di eventuali sottoclassi della classe;
- Attributi: elenco degli attributi della classe di cui si evidenziano:
 - Nome: nome dell'attributo;
 - Descrizione: descrizione dell'attributo;
- Responsabilità: responsabilità della classe di cui si evidenziano:
 - Nome: nome della responsabilità;
 - Collaboratori: nomi di eventuali classi che collaborano alla responsabilità;

3.2. Elenco delle schede CRC

Nome	Studente
Superclassi	/
Sottoclassi	/
Descrizione	Classe che rappresenta un generico studente.
Attributi	Di seguito la lista degli attributi: <ul style="list-style-type: none">• Matricola: matricola dello studente;• Nome: nome dello studente;• Cognome: cognome dello studente;• Email: email istituzionale dello studente;• Codice Fiscale: codice fiscale dello studente;• Elenco prenotazioni: lista delle prenotazioni effettuate dall'utente;
Responsabilità	/

Nome	Prenotazione
Superclassi	/
Sottoclassi	Prenotazione Confermata, Prenotazione Non Confermata
Descrizione	Classe che rappresenta una generica prenotazione effettuata da uno studente;
Attributi	<p>Di seguito la lista degli attributi:</p> <ul style="list-style-type: none"> ● Studente: studente che ha effettuato la prenotazione; ● Aula: aula associata alla prenotazione. ● Data: giorno relativo alla prenotazione di un posto in aula; ● Inizio: orario in cui inizia la validità della prenotazione; ● Fine: orario in cui termina la validità della prenotazione;
Responsabilità	/

Nome	Prenotazione Confermata
Superclassi	Prenotazione
Sottoclassi	/
Descrizione	Classe che rappresenta una prenotazione confermata. Una prenotazione confermata garantisce che lo studente possa recarsi a lezione nell'aula e negli orari indicati in fase di prenotazione. E' possibile generare una ricevuta di una prenotazione confermata.
Attributi	<p>Di seguito la lista degli attributi:</p> <ul style="list-style-type: none"> ● Istante conferma: istante in cui la prenotazione è stata confermata.
Responsabilità	/

Nome	Prenotazione Non Confermata
Superclassi	Prenotazione
Sottoclassi	/
Descrizione	Classe che rappresenta una prenotazione non confermata. Una prenotazione non confermata corrisponde all'iscrizione alla waiting list relativa agli studenti che intendono prenotare un posto in aula negli orari della prenotazione non confermata. Ogni Studente nella waiting ha una posizione che dipende dall'istante in cui è stata fatta la richiesta di iscrizione alla waiting list.

Attributi	Di seguito la lista degli attributi: <ul style="list-style-type: none"> • Istante richiesta: istante in cui lo Studente si è iscritto alla waiting list della relativa prenotazione.
Responsabilità	/

Nome	Aula
Superclassi	/
Sottoclassi	/
Descrizione	Classe che rappresenta un'aula dell'ateneo.
Attributi	Di seguito la lista degli attributi: <ul style="list-style-type: none"> • Codice aula: codice associato all'aula; • Posti: numero di posti prenotabili nel rispetto della nuova capienza delle aule in seguito alla pandemia di Covid-19; • Edificio: Codice dell'edificio in cui si trova l'aula.
Responsabilità	/

Nome	Edificio
Superclassi	/
Sottoclassi	/
Descrizione	Classe che rappresenta un edificio dell'ateneo.
Attributi	Di seguito la lista degli attributi: <ul style="list-style-type: none"> • Codice edificio: codice associato all'edificio; • Indirizzo: indirizzo dell'edificio; • Lista aule: lista delle aule all'interno dell'edificio.
Responsabilità	/

Nome	HandlerLogin
Superclassi	/
Sottoclassi	/
Descrizione	Classe che gestisce l'autenticazione di un utente.
Attributi	/

Responsabilità	<p>Di seguito la lista delle responsabilità:</p> <ul style="list-style-type: none"> • Contatta il sistema Infostud inviando i dati inseriti dall'Utente; • Riceve l'esito della verifica sulla correttezza dei dati inseriti. <p>I Collaboratori sono: ManagerInfostud, Studente.</p>
----------------	---

Nome	HandlerLogout
Superclassi	/
Sottoclassi	/
Descrizione	Classe che gestisce il logout.
Attributi	/
Responsabilità	<p>Di seguito la lista delle responsabilità:</p> <ul style="list-style-type: none"> • Effettua il logout. <p>I Collaboratori sono: Studente.</p>

Nome	HandlerEffettuaPrenotazione
Superclassi	/
Sottoclassi	/
Descrizione	Classe che gestisce la creazione di una nuova prenotazione.
Attributi	/
Responsabilità	<p>Di seguito la lista delle responsabilità:</p> <ul style="list-style-type: none"> • Effettua una prenotazione confermata se lo studente soddisfa i vari requisiti; • Iscrive lo studente alla waiting list se i posti in aula sono terminati effettuando una nuova prenotazione non confermata; <p>I Collaboratori sono: Studente, PrenotazioneConfermata, PrenotazioneNonConfermata, Edificio, Aula.</p>

Nome	HandlerSalvataggioRicevuta
Superclassi	/
Sottoclassi	/

Descrizione	Classe che gestisce il salvataggio della ricevuta di prenotazione.
Attributi	/
Responsabilità	<p>Di seguito la lista delle responsabilità:</p> <ul style="list-style-type: none"> • Salva la ricevuta di prenotazione. <p>I Collaboratori sono: PrenotazioneConfermata.</p>

Nome	HandlerCancellazionePrenotazioneConfermata
Superclassi	/
Sottoclassi	/
Descrizione	Classe che gestisce la cancellazione di una prenotazione confermata.
Attributi	/
Responsabilità	<p>Di seguito la lista delle responsabilità:</p> <ul style="list-style-type: none"> • Cancella una prenotazione confermata; • Contatta l'handlerListaPrenotazioni per ottenere la lista degli Studenti iscritti alla waiting list della prenotazione cancellata e conferma la prenotazione del primo studente nella waiting list; • Invia una email al relativo studente. <p>I Collaboratori sono: PrenotazioneConfermata, PrenotazioneNonConfermata, HandlerListaPrenotazioni, ManagerPostaElettronica.</p>

Nome	HandlerListaPrenotazioni
Superclassi	/
Sottoclassi	/
Descrizione	Classe che gestisce la visualizzazione della lista delle prenotazioni effettuate da uno studente, sia confermate che non confermate.
Attributi	/
Responsabilità	<p>Di seguito la lista delle responsabilità:</p> <ul style="list-style-type: none"> • Visualizza la lista delle prenotazioni confermate di uno studente; • Visualizza la lista delle prenotazioni non confermate di

	<p>uno studente.</p> <p>I Collaboratori sono: PrenotazioneConfermata, PrenotazioneNonConfermata.</p>
--	--

Nome	HandlerCancellazionePrenotazioneNonConfermata
Superclassi	/
Sottoclassi	/
Descrizione	Classe che gestisce la cancellazione di una prenotazione non confermata.
Attributi	/
Responsabilità	<p>Di seguito la lista delle responsabilità:</p> <ul style="list-style-type: none"> • Cancella la prenotazione non confermata; <p>I Collaboratori sono: PrenotazioneNonConfermata.</p>

Nome	InterfacciaInfostud
Superclassi	/
Sottoclassi	/
Descrizione	Classe che permette la comunicazione con il sistema Infostud.
Attributi	/
Responsabilità	<p>Di seguito la lista delle responsabilità:</p> <ul style="list-style-type: none"> • Invia le richieste al sistema Infostud; • Riceve le risposte dal sistema Infostud.

Nome	InterfacciaPostaElettronica
Superclassi	/
Sottoclassi	/
Descrizione	Classe che permette la comunicazione il sistema di posta elettronica.
Attributi	/
Responsabilità	Di seguito la lista delle responsabilità:

	<ul style="list-style-type: none"> • Invia le email.
--	---

Nome	ManagerPostaElettronica
Superclassi	/
Sottoclassi	/
Descrizione	Classe che gestisce la formattazione del testo delle email.
Attributi	/
Responsabilità	<p>Di seguito la lista delle responsabilità:</p> <ul style="list-style-type: none"> • Formatta l'email in base alla richiesta; • Invia l'email. <p>I Collaboratori sono: InterfacciaPostaElettronica.</p>

Nome	ManagerInfostud
Superclassi	/
Sottoclassi	/
Descrizione	Classe che gestisce la comunicazione con il sistema Infostud.
Attributi	/
Responsabilità	<p>Di seguito la lista delle responsabilità:</p> <ul style="list-style-type: none"> • Riceve richieste dagli handler e le inoltra al sistema Infostud; • Riceve risposte dal sistema Infostud e le inoltra agli handler adeguati. <p>I Collaboratori sono: InterfacciaInfostud.</p>

Nome	UILogin
Superclassi	/
Sottoclassi	/
Descrizione	Classe che rappresenta l'insieme degli elementi dell'interfaccia che si occupa dell'accesso al sistema.

Attributi	/
Responsabilità	<p>Di seguito la lista delle responsabilità:</p> <ul style="list-style-type: none"> • Visualizza il form per l'inserimento dei dati necessari per accedere al sistema; • Invia i dati inseriti alle corrispondenti classi handler; • Visualizza i messaggi di risposta; <p>Di seguito i collaboratori della classe: HandlerLogin</p>

Nome	UILogout
Superclassi	/
Sottoclassi	/
Descrizione	Classe che rappresenta l'insieme degli elementi dell'interfaccia che si occupa del logout dal sistema.
Attributi	/
Responsabilità	<p>Di seguito la lista delle responsabilità:</p> <ul style="list-style-type: none"> • Visualizza il bottone per richiedere il logout; • Richiede il logout alla corrispondente classe handler; <p>Di seguito i collaboratori della classe: HandlerLogout;</p>

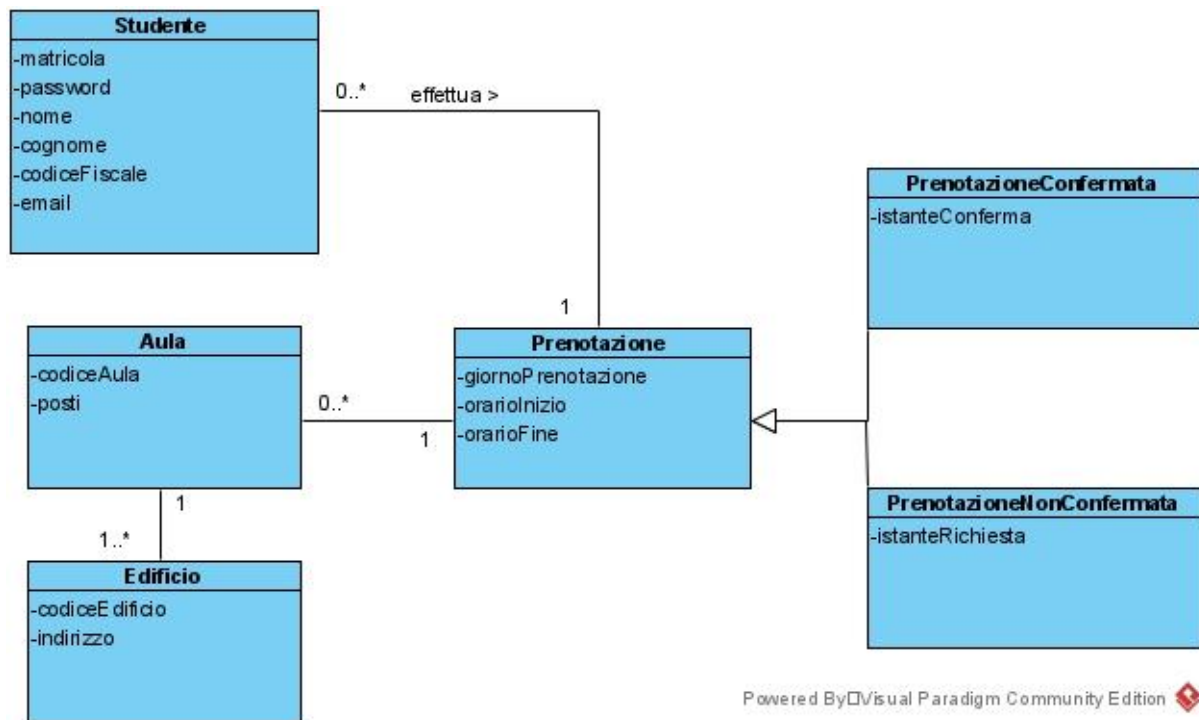
Nome	UIPrenotazione
Superclassi	/
Sottoclassi	/
Descrizione	Classe che rappresenta l'insieme degli elementi dell'interfaccia che si occupa della prenotazione di una lezione.
Attributi	/
Responsabilità	<p>Di seguito la lista delle responsabilità:</p> <ul style="list-style-type: none"> • Visualizza i form per l'inserimento dei dati necessari ad effettuare una prenotazione; • Invia i dati inseriti alle corrispondenti classi handler; • Visualizza eventuali messaggi informativi o di errore generati durante la prenotazione. <p>Di seguito i collaboratori della classe: HandlerEffettuaPrenotazione.</p>

Nome	UIVisualizzazionePrenotazioniConfermate
Superclassi	/
Sottoclassi	/
Descrizione	Classe che rappresenta l'insieme degli elementi dell'interfaccia che si occupa di visualizzare la lista delle prenotazioni confermate effettuate da uno studente.
Attributi	/
Responsabilità	<p>Di seguito la lista delle responsabilità:</p> <ul style="list-style-type: none"> • Visualizza la lista delle prenotazioni confermate; • Visualizza il bottone per scaricare la ricevuta di prenotazione; • Visualizza l'interfaccia relativa alla cancellazione di una prenotazione; • Invia i dati alle relative classi handler; <p>Di seguito i collaboratori della classe: HandlerListaPrenotazioni, HandlerStampaRicevuta, HandlerCancellazionePrenotazioneConfermata.</p>

Nome	UIVisualizzazionePrenotazioniNonConfermate
Superclassi	/
Sottoclassi	/
Descrizione	Classe che rappresenta l'insieme degli elementi dell'interfaccia che si occupa di visualizzare la lista delle prenotazioni non confermate effettuate da uno studente.
Attributi	/
Responsabilità	<p>Di seguito la lista delle responsabilità:</p> <ul style="list-style-type: none"> • Visualizza la lista delle prenotazioni non confermate; • Visualizza l'interfaccia relativa alla cancellazione di una prenotazione non confermata; • Invia i dati alle relative classi handler; <p>Di seguito i collaboratori della classe: HandlerListaPrenotazioni, HandlerCancellazionePrenotazioneNonConfermata.</p>

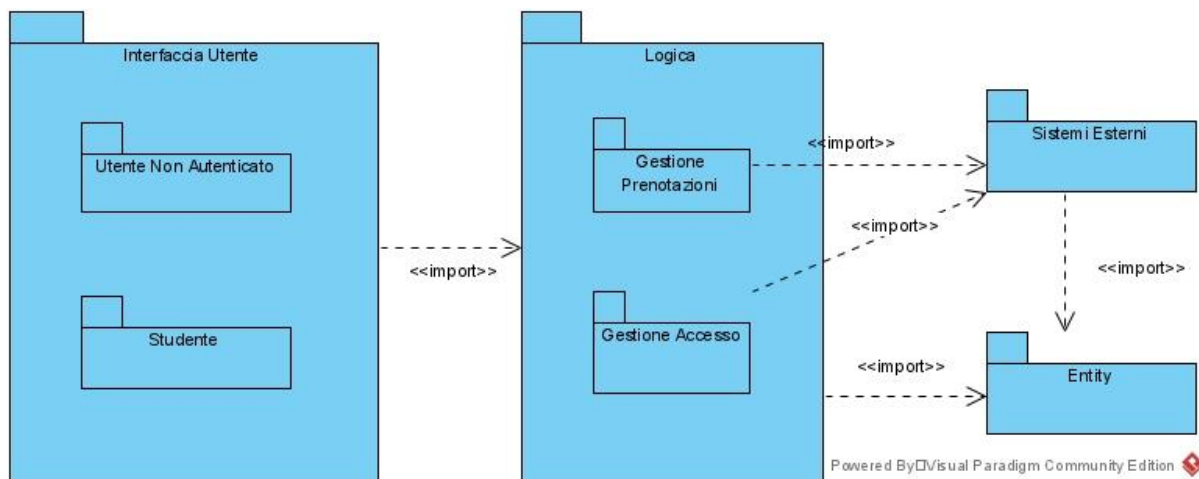
4. Diagramma delle classi entity

Il seguente diagramma mostra le associazioni tra le classi entity:



5. Package di analisi

Il seguente diagramma mostra la suddivisione in package di analisi adottata in questo progetto:



6. Stereotipi EBC

6.1. Introduzione

Per raffinare l'analisi si utilizzano gli stereotipi EBC (Entity, Boundary, Control). Il modello EBC organizza le responsabilità delle classi in base al loro ruolo nella realizzazione del caso d'uso:

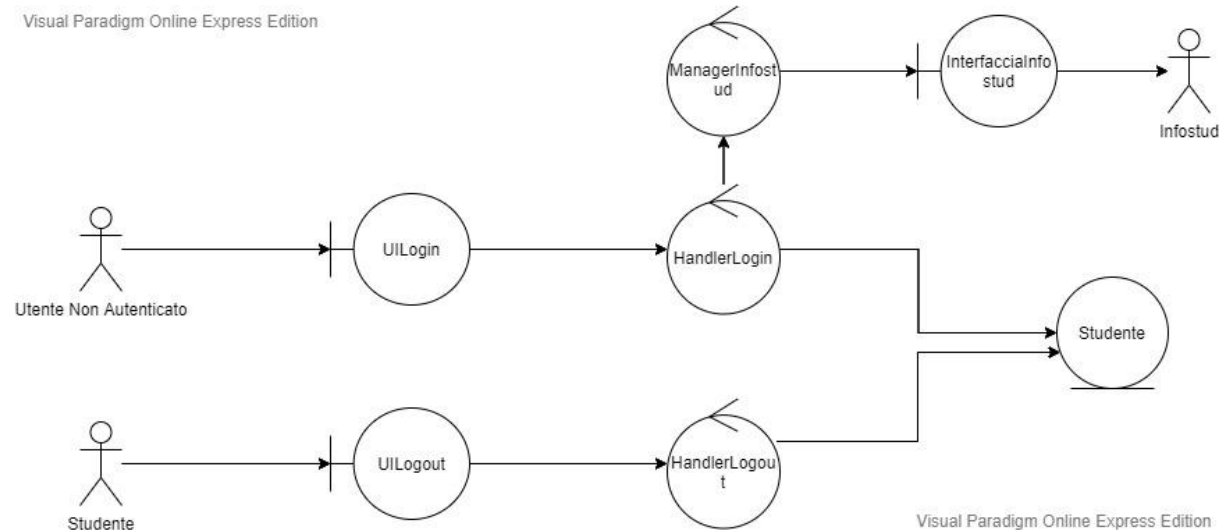
- Entity: rappresenta informazioni di lunga durata rilevanti per gli stakeholder (ovvero derivate principalmente da oggetti di dominio, di solito persistenti). Una classe entity può comunicare con classi control o altre classi entity;
- Boundary: rappresenta l'interazione con attori esterni (utenti o sistemi esterni). Una classe boundary può comunicare con classi control;
- Control: racchiude le procedure richieste per l'esecuzione di un caso d'uso e la sua logica di business, e controlla altri oggetti coinvolti nel caso d'uso. Una classe control può comunicare con altre classi control, entity o boundary.

I seguenti diagrammi mostrano gli stereotipi EBC per ogni sottosistema.

Le associazioni tra le classe entity vengono omesse perché già trattate nel relativo diagramma delle classi.

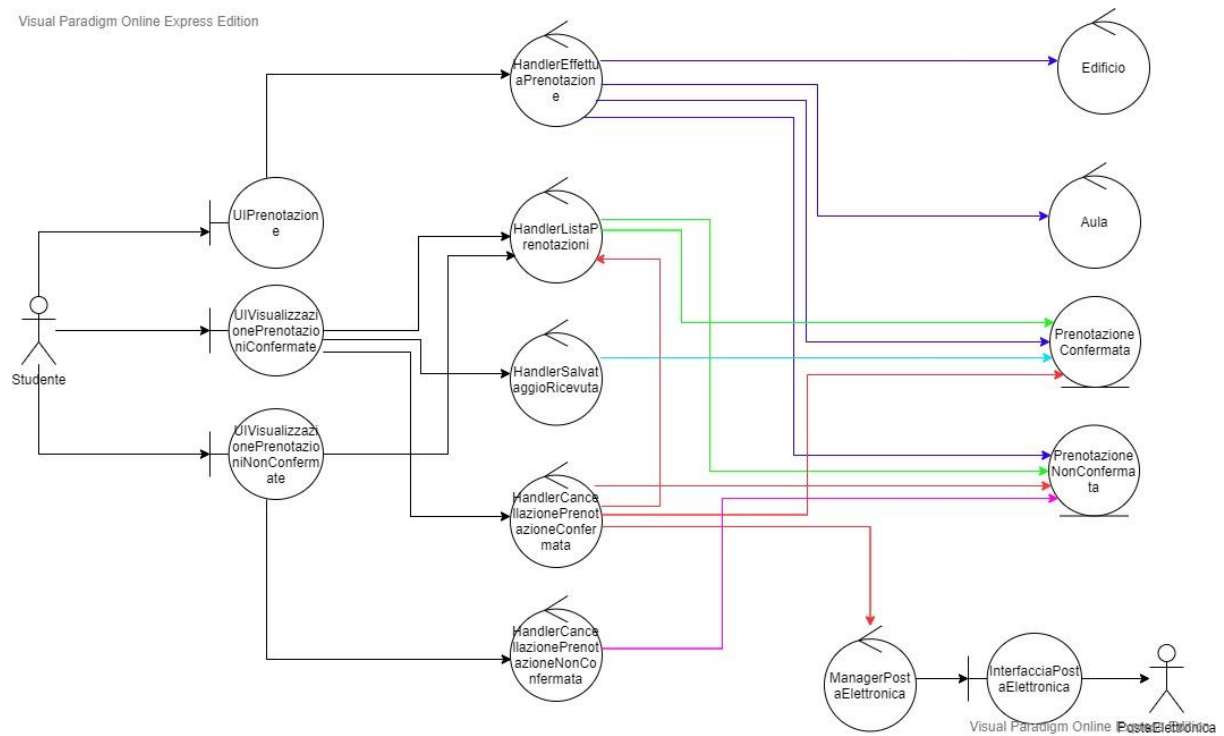
6.2. Diagrammi degli stereotipi

Sottosistema "Gestione Accesso":



Sottosistema "Gestione Prenotazioni":

Visual Paradigm Online Express Edition



Documento di Design

1. Prefazione

In questo documento si andrà ad evolvere il documento di analisi aggiungendo dettagli relativi all'implementazione. Verrà illustrata l'architettura utilizzata, si raffineranno le classi di analisi in classi di progettazione ed infine verranno illustrati i diagrammi di sequenza e i cicli di vita di classi significative.

2. Specifica dell'architettura

ProdigitX è una web application che utilizza il paradigma client-server. Il server è composto da un *application server* che contiene la logica dell'applicazione e da un *web server* che gestisce le richieste provenienti dai client. Vi è inoltre un *database manager* che consente la manipolazione dei dati nel database.

Nella gestione del server viene utilizzato *Apache Tomcat 9*, un'applicazione software open source sviluppato dalla Apache Software Foundation che implementa le specifiche servlet, fornendo quindi una piattaforma software per l'esecuzione di applicazioni web sviluppate in linguaggio Java.

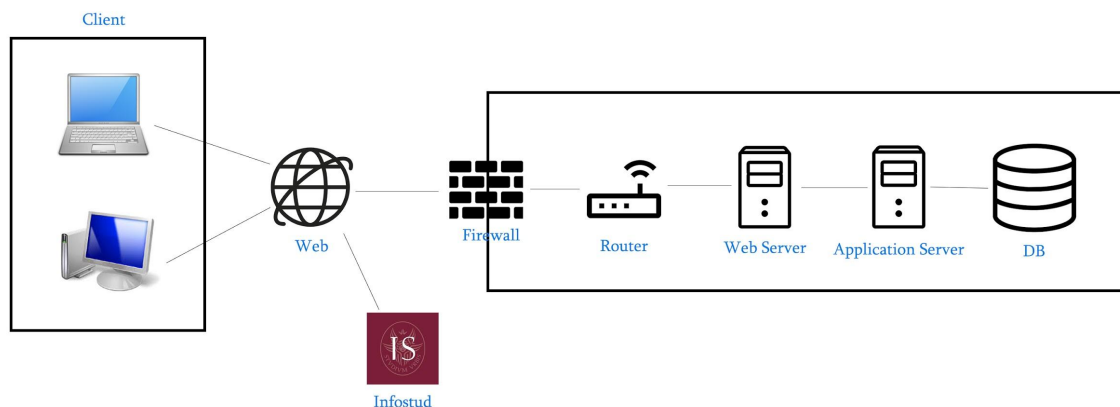
Per il database manager viene utilizzato *PostgreSQL 13*, un DBMS ad oggetti open source. Il punto di forza di PostgreSQL è la programmabilità infatti rende più semplice costruire applicazioni per il mondo reale, utilizzando i dati prelevati dalla base di dati.

Nel momento in cui si effettua il login, il server comunica con il sistema Infostud per autenticare l'utente.

Il sistema è protetto da un firewall che controlla e filtra le richieste dei client per offrire una maggiore sicurezza.

Tutte le comunicazioni tra server e client avverranno utilizzando il protocollo Https con TLS 1.3, per garantire la sicurezza nello scambio di dati.

Qui di seguito viene mostrato un diagramma che riassume l'architettura del sistema:



3. Classi di progetto

3.1. Introduzione

Di seguito il raffinamento delle classi di analisi descritte nel documento di analisi. Ogni classe identificata viene inserita nel relativo package che rappresenta una parte del sottosistema. Di ogni classe vengono esposti gli attributi, i metodi e i tipi di ritorno. Sono state aggiunte le classi Servlet che hanno il compito di ricevere le richieste ed affidarle agli handler appropriati che la gestiranno. Inoltre, viene adottato il pattern DAO (*Data Access Object*): si tratta di una classe con relativi metodi utilizzata nelle applicazioni web per stratificare ed isolare l'accesso ad una tabella del database tramite query (poste all'interno dei metodi della classe) creando un maggiore livello di astrazione.

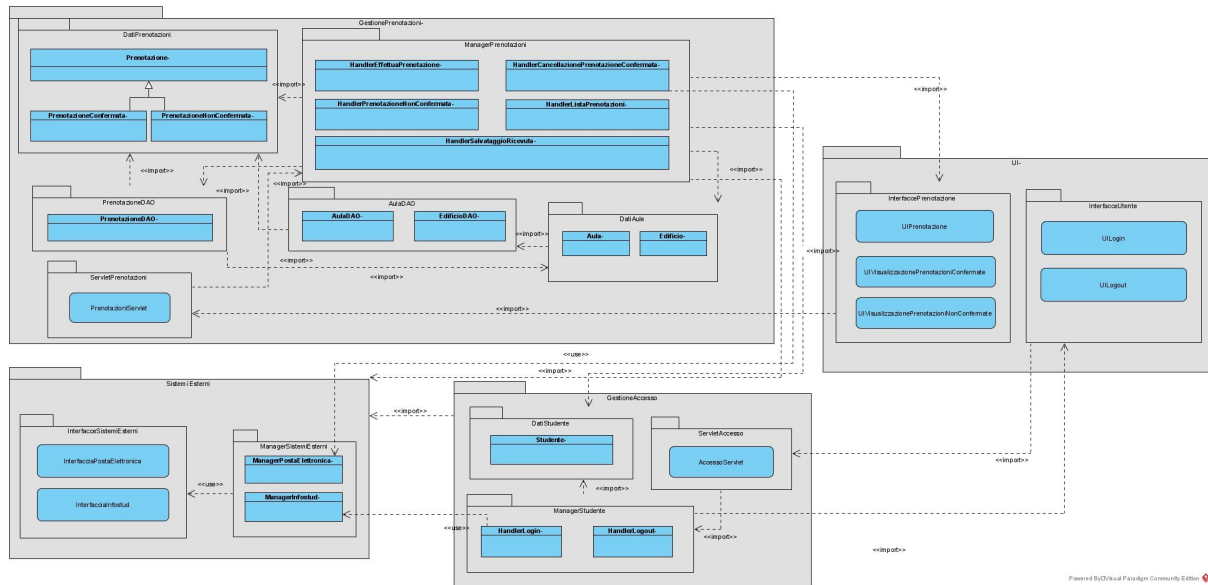
Nei diagrammi di sequenza verrà mostrato a grandi linee il funzionamento delle classi DAO.

Vengono quindi aggiunti i seguenti package:

- `javax.servlet.http.*`: libreria che gestisce le richieste generate da uno o più client, attraverso lo scambio di messaggi tra il server ed i client che hanno effettuato la richiesta;
- `java.sql.*`: libreria per la manipolazione dei dati nel database;

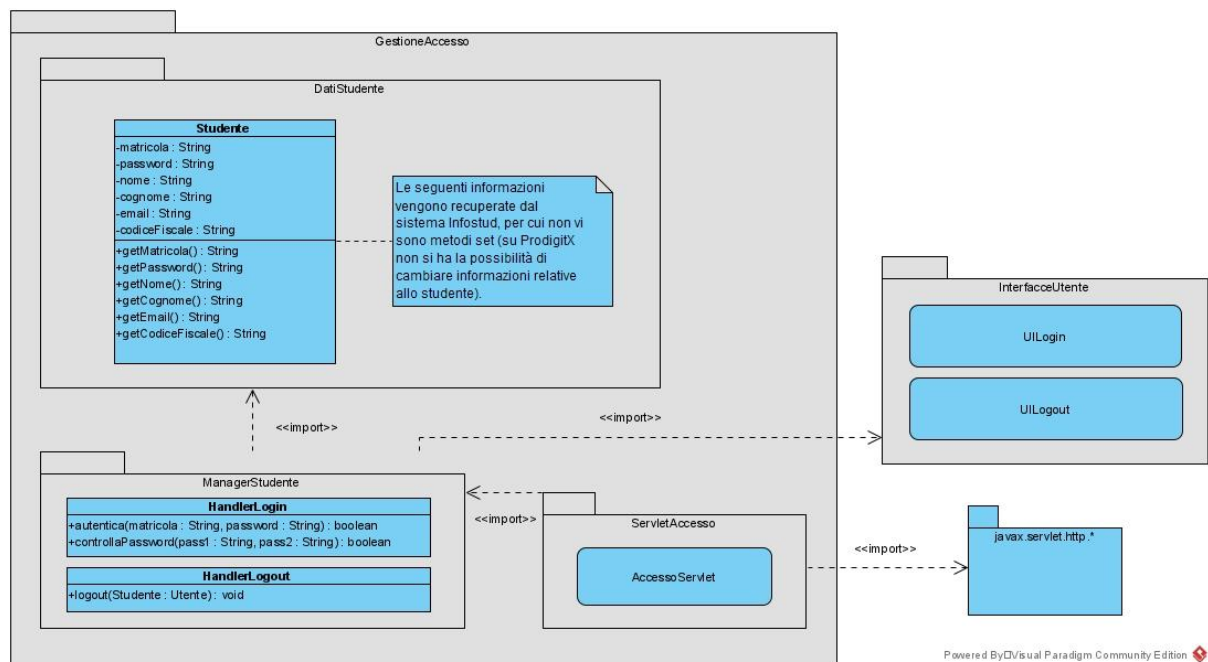
3.2. Lista dei diagrammi dei package:

Qui di seguito un diagramma che mostra le dipendenze tra i vari package:

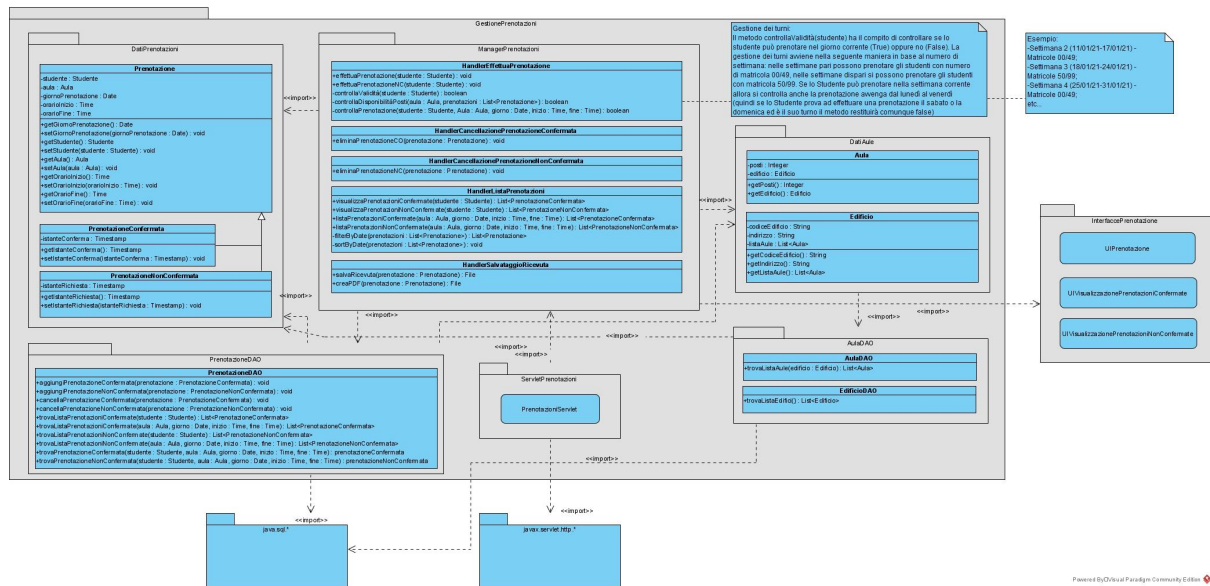


Vengono mostrati ora i vari package nel dettaglio:

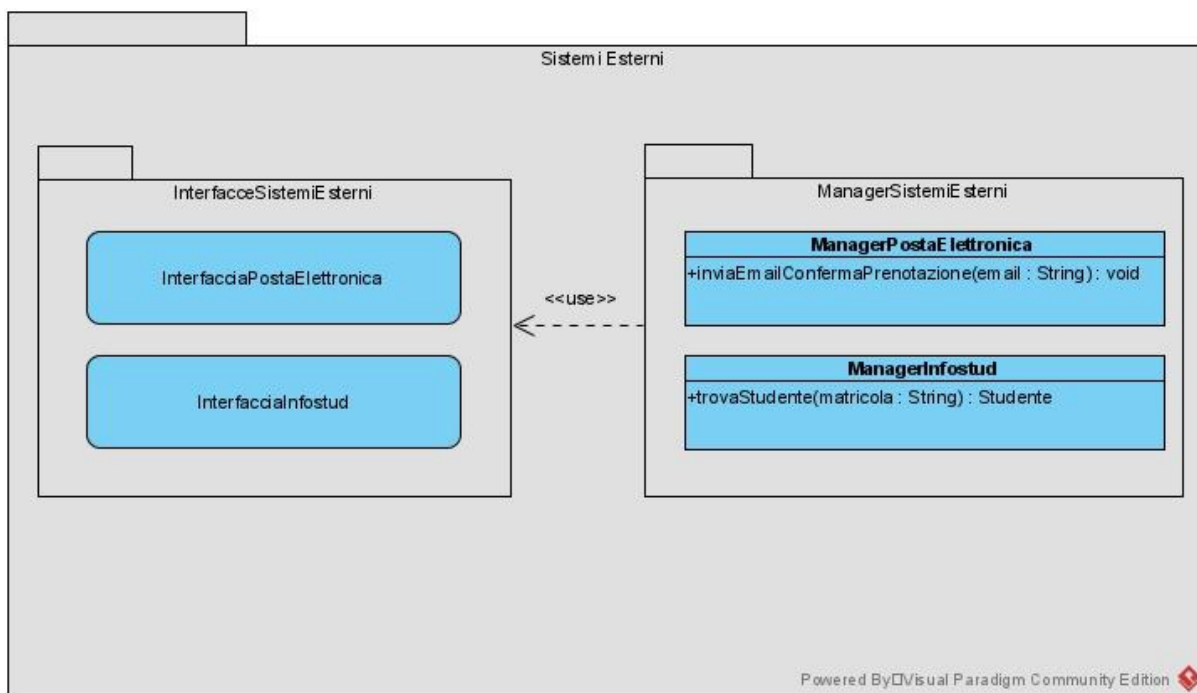
Gestione Accesso:



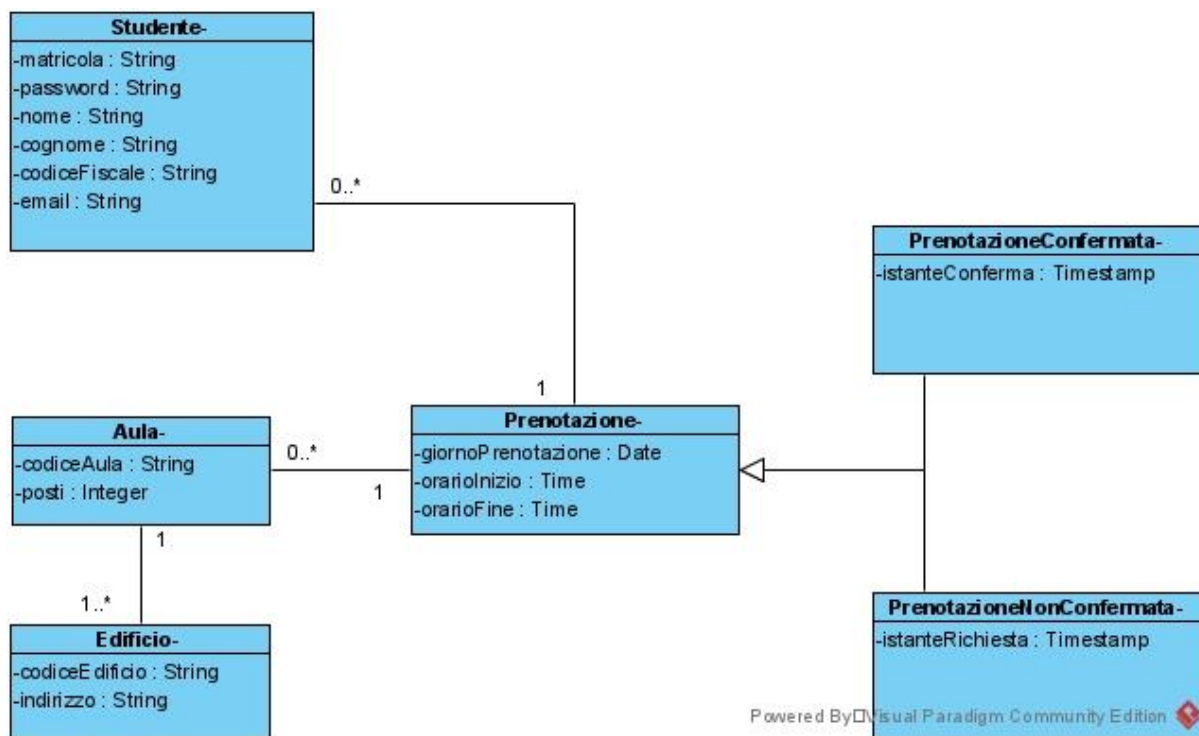
Gestione Prenotazioni:



Sistemi Esterni:



Qui di seguito il diagramma delle classi entity modificato per la fase di design:



4. Diagrammi di sequenza

4.1. Introduzione

I diagrammi di sequenza mostrano l'interazione fra le classi durante l'esecuzione dei casi d'uso. Ciascun diagramma è identificato da:

- *ID*: valore numerico associato univocamente al diagramma;
- *Nome*: nome del caso d'uso a cui il diagramma fa capo;

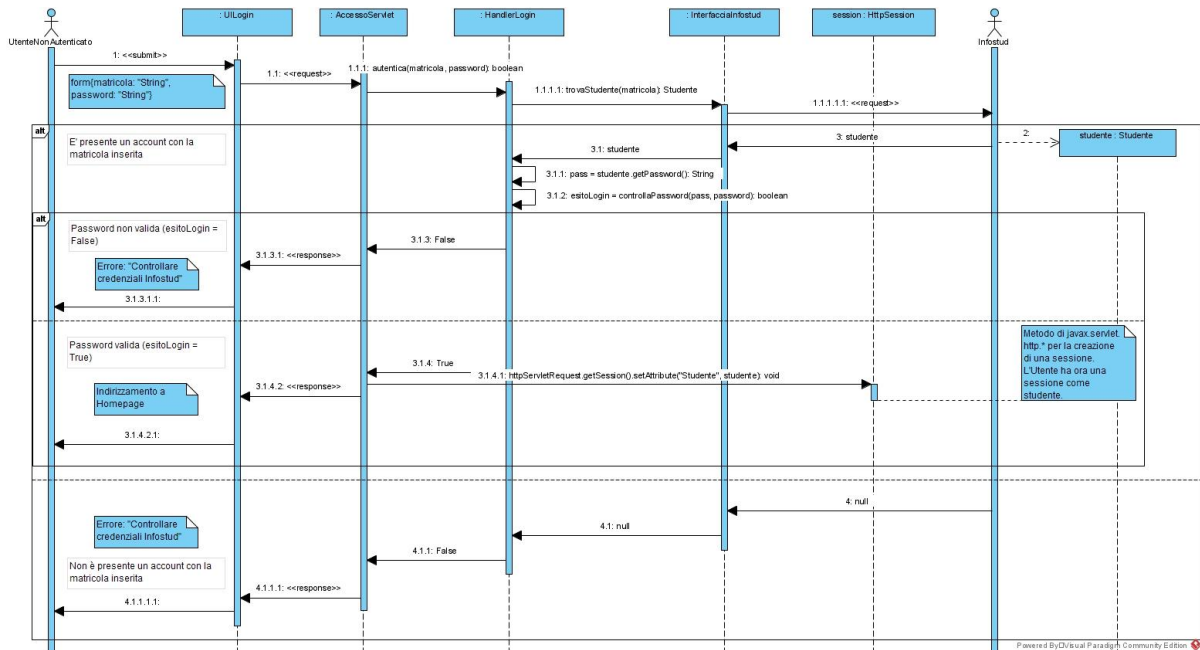
La comunicazione tra le varie UI e le servlet viene descritta tramite stereotipi:

- <<click>>: l'attore interagisce con il sistema facendo click su un particolare punto dell'interfaccia;
- <<compile>>: l'attore interagisce con il sistema compilando un form;
- <<submit>>: comunicazione dell'interfaccia alla servlet in seguito alla compilazione di un form;
- <<request>>: comunicazione dell'interfaccia alla servlet in seguito ad un click;
- <<response>>: risposta del servlet all'interfaccia.

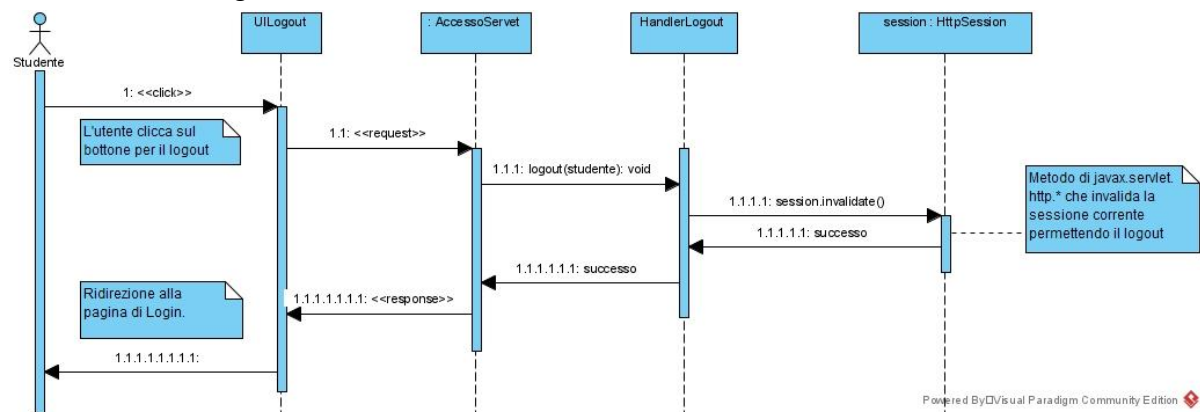
4.2. Lista dei diagrammi di sequenza

Di seguito i diagrammi di sequenza:

DS1 - EffettuaLogin:



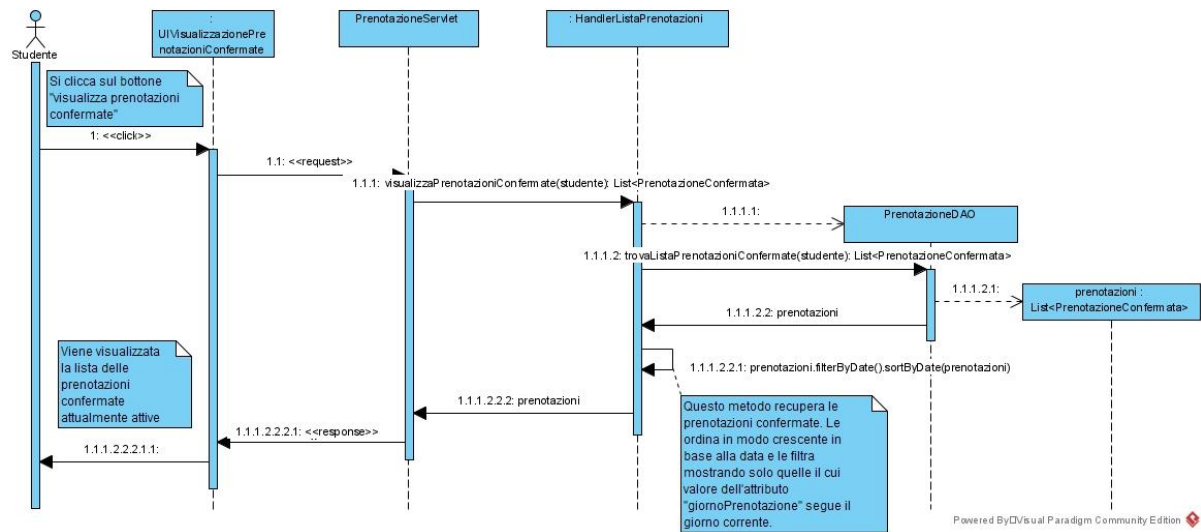
DS2 - EffettuaLogout:



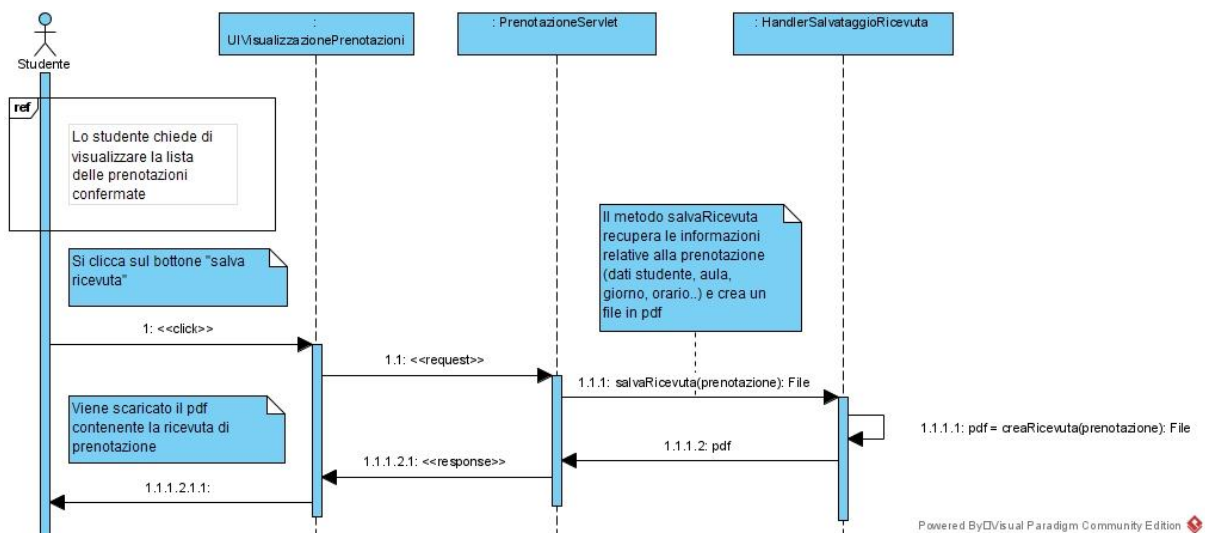
DS3 - EffettuaPrenotazione:



DS4 - VisualizzaPrenotazioniConfermate:

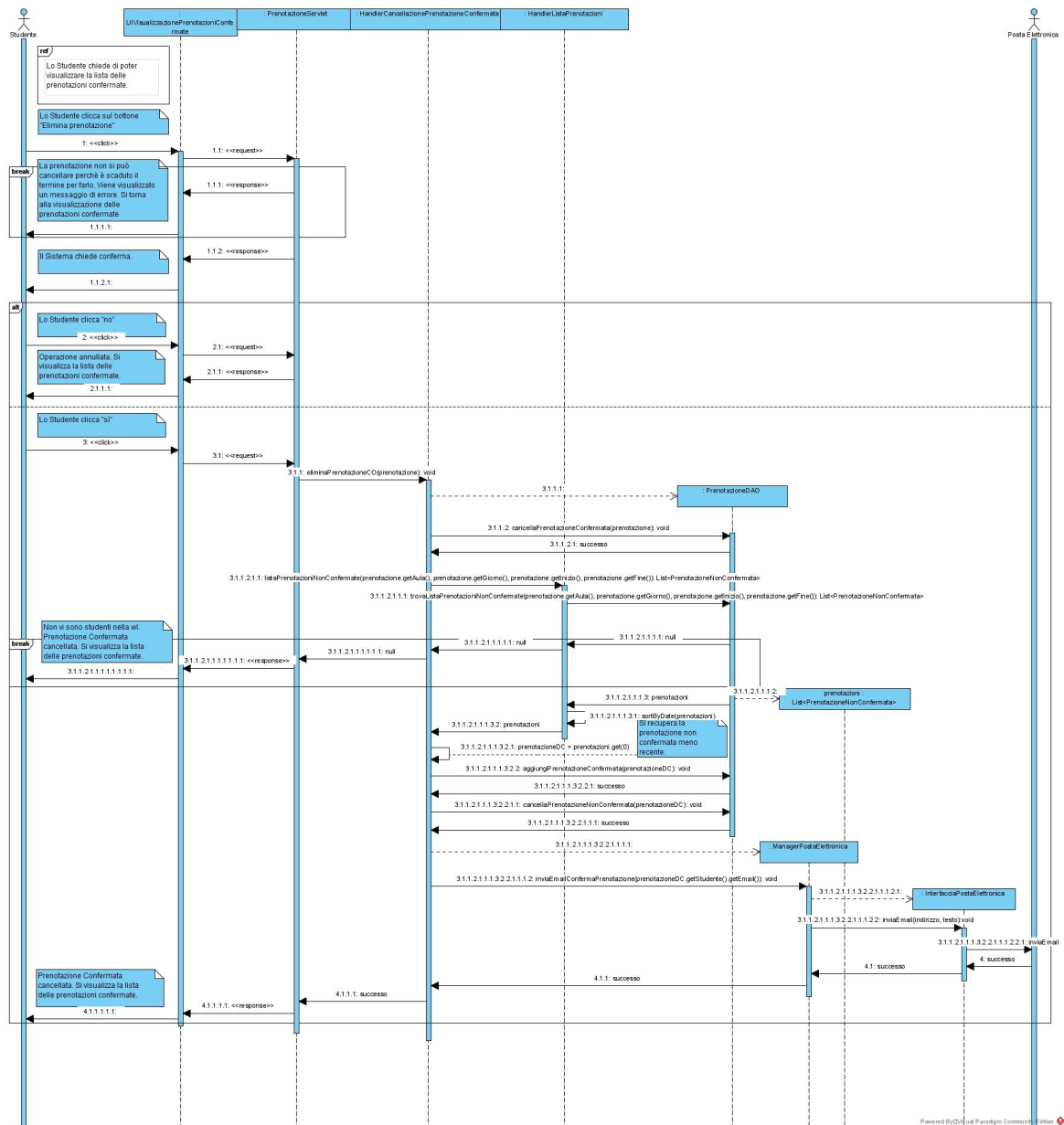


DS5 - SalvaRicevuta:



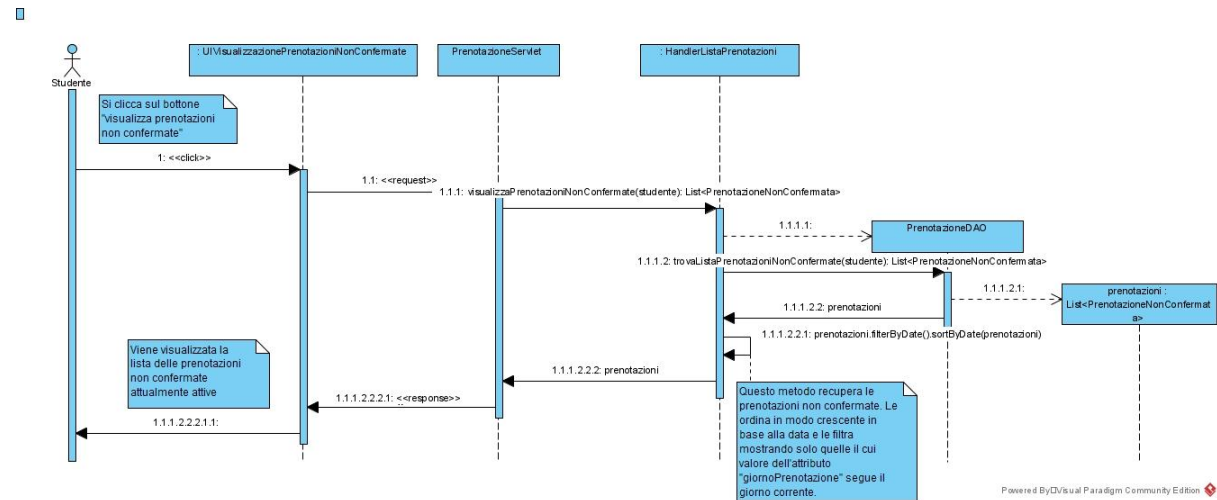
DS6 - CancellaPrenotazioneConfermata:

1

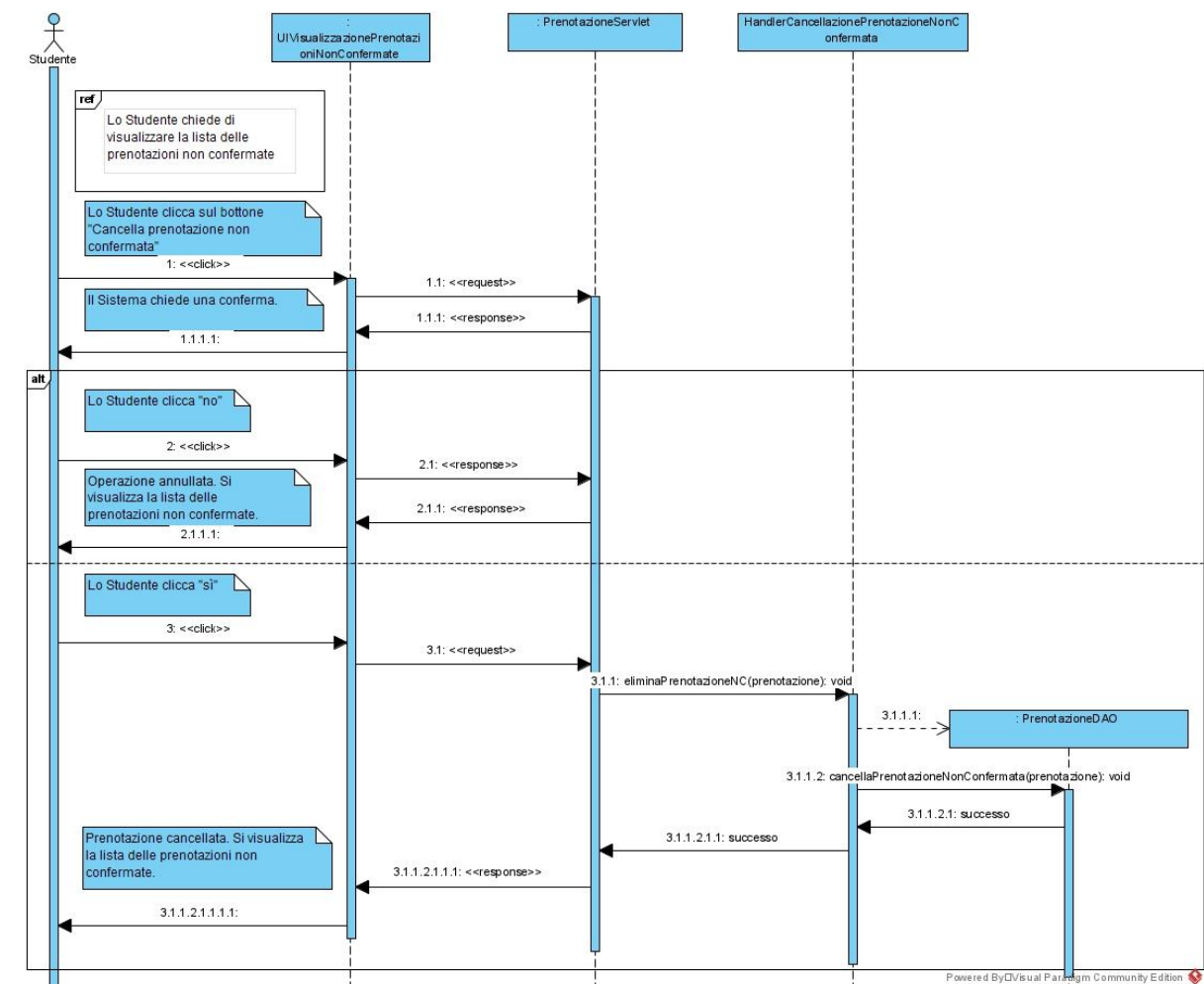


Powered By D4U - UML Paradigm Community Edition

DS7 - VisualizzaPrenotazioniNonConfermate:



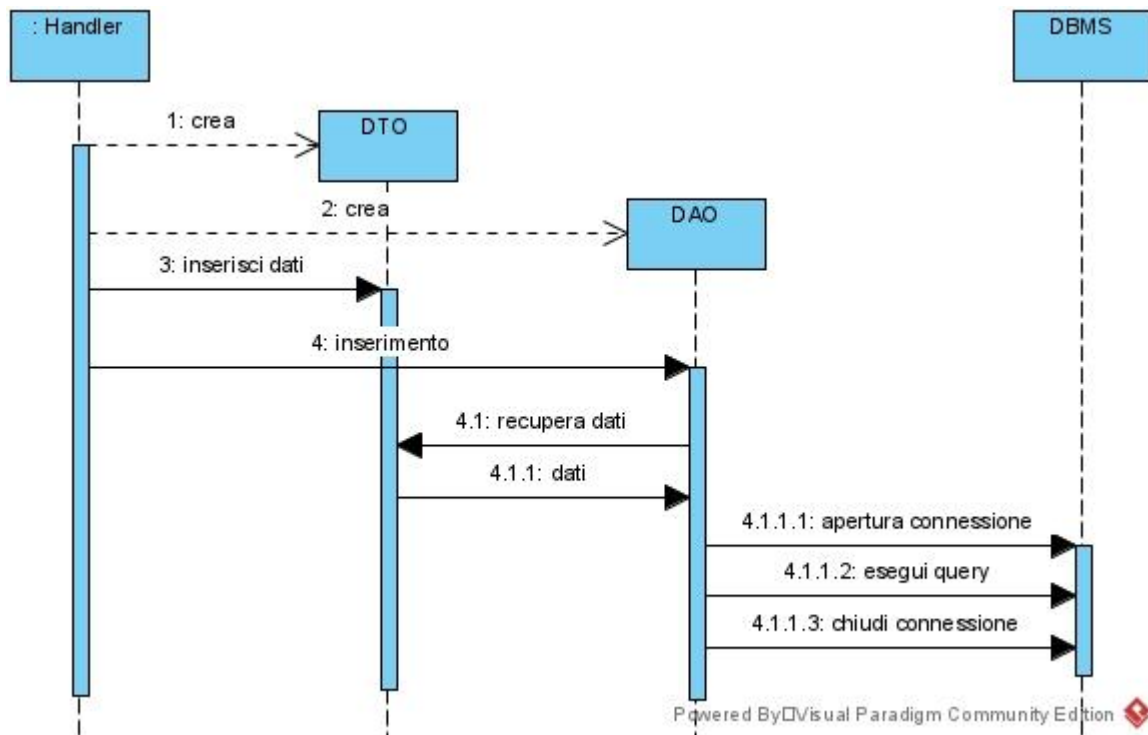
DS8 - CancellaPrenotazioneNonConfermata:



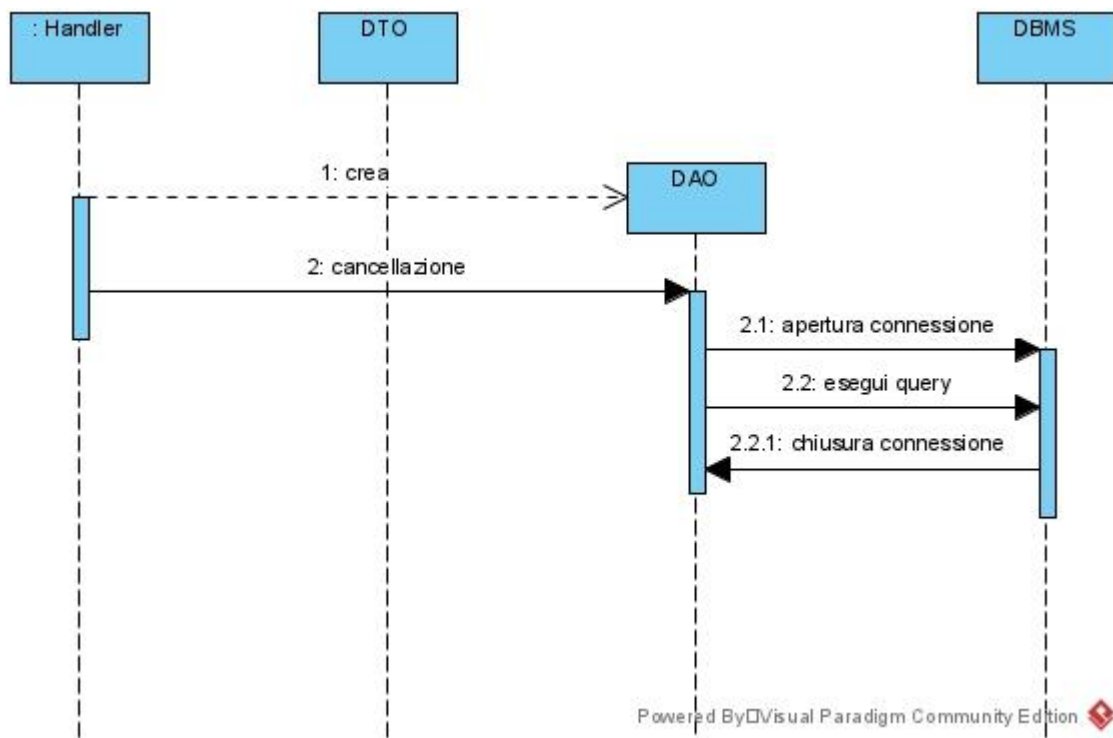
Qui di seguito viene mostrato a grandi linee il funzionamento delle classi DAO, dove:

- Handler: rappresenta una generica classe handler;
- DAO (Data Access Object): rappresenta una generica classe DAO che offre all'handler un accesso trasparente ai dati e si occupa della comunicazione con il database;
- DTO (Data Transfer Object): classe che si occupa del trasferimento ed il recupero dei dati tra la classe DAO e gli handler;
- DBMS: database manager che si occupa della gestione dei dati (nel nostro caso PostgreSQL);

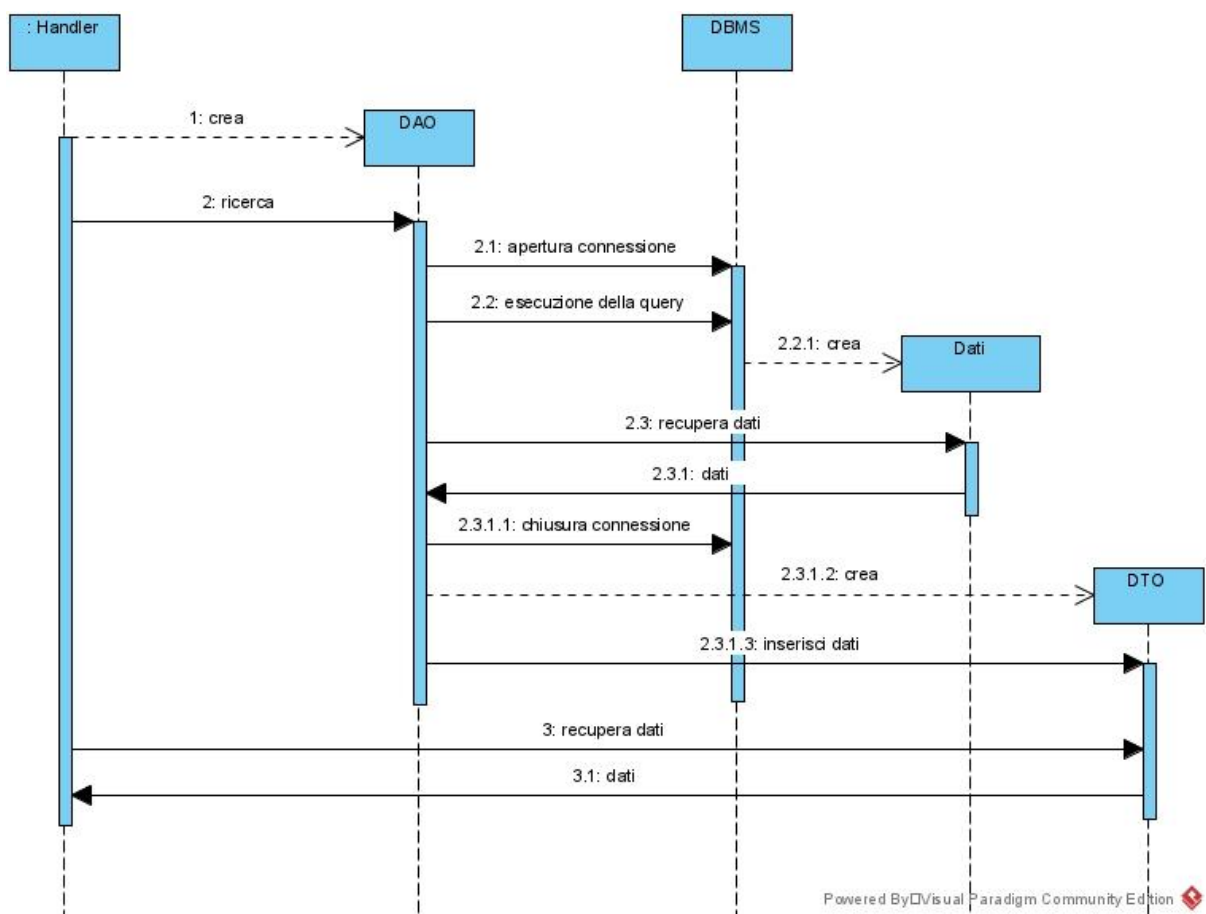
Inserimento:



Cancellazione:

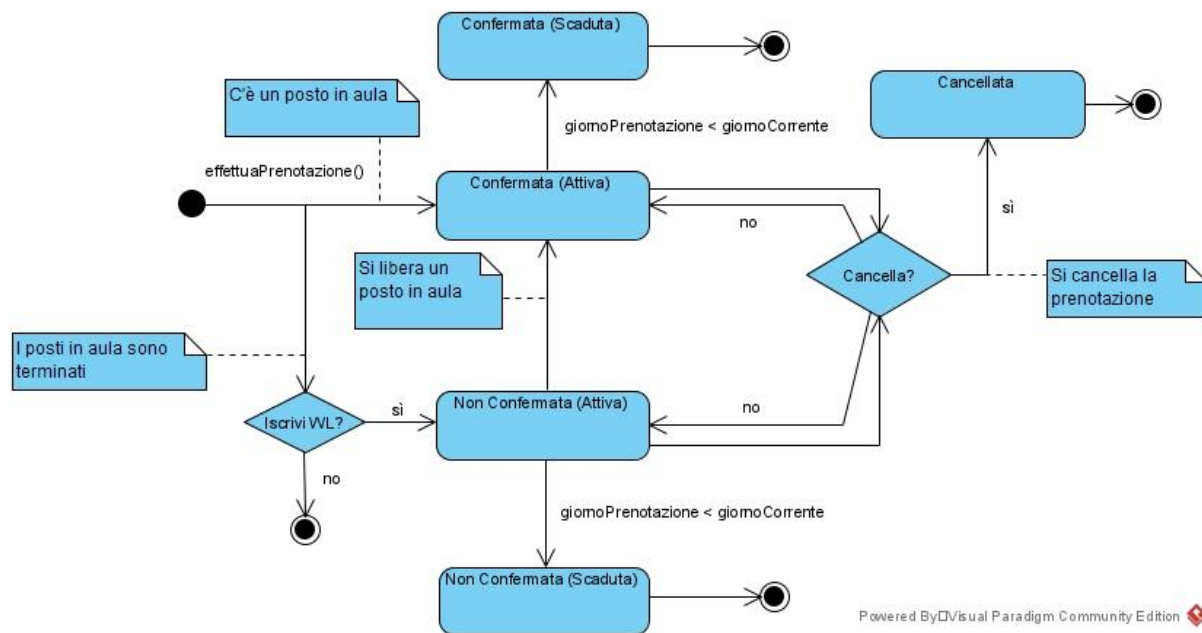


Ricerca:



5. Cicli di vita di classi significative

I cicli di vita si riferiscono a classi i cui oggetti (istanze) possono cambiare stato a seguito di azioni. Viene ora analizzato il ciclo di vita delle prenotazioni (che rappresentano la classe più significativa del sistema) mediante l'utilizzo di un diagramma di stato:



Piano dei Test

1. Prefazione

In questo documento si definiscono i test funzionali necessari per garantire il corretto funzionamento del sistema.

2. Elenco dei test

2.1. Introduzione

Ogni test viene rappresentato da una tabella in cui si identificano:

- ID: stringa univoca del tipo T_n dove:
 - T: indica che si tratta di un test;
 - n: valore univoco associato al test;
- UC: Caso d'uso a cui il test si riferisce;
- Descrizione: breve descrizione di cosa si andrà a testare;
- Precondizioni: asserzioni vere sullo stato del sistema prima dell'esecuzione del test;
- Azioni: sequenza di azioni da eseguire per il corretto svolgimento del test;
- Post-condizioni: asserzioni vere sullo stato del sistema dopo l'esecuzione del test.

2.2. Test

Di seguito la lista dei test:

Test T_1 (CU_1 - Effettua Login):

ID	T_1.1
Descrizione	Verifica che il sistema autentichi correttamente uno studente.
Precondizioni	Uno studente s1 è registrato su Infostud con matricola m1 e password p1. Non vi è una sessione aperta per lo studente s1.
Azioni	<ol style="list-style-type: none">1. Si dà avvio alla procedura di login;2. Si inserisce matricola m1 e password p1 nel form;3. Si confermano i dati inseriti;4. Verificare che il sistema autentichi lo studente s1 e lo reindirizzi alla homepage.
Postcondizioni	Si è autenticati nel sistema come studente s1.

ID	T_1.2
Descrizione	Verifica che il sistema neghi l'accesso inserendo credenziali errate.

Precondizioni	Uno studente s1 è registrato su Infostud con matricola m1 e password p1. Non vi è una sessione aperta per lo studente s1.
Azioni	<ol style="list-style-type: none"> 1. Si dà avvio alla procedura di login; 2. Si inserisce matricola m2 e password p2 nel form; 3. Si confermano i dati inseriti; 4. Verificare che il sistema restituisca un errore e chieda di inserire nuovamente i dati; 5. Si inserisce la matricola m1 nel form; 6. Verificare che il sistema restituisca un errore e chieda di inserire nuovamente i dati; 7. Si inserisce la password p1 nel form; 8. Verificare che il sistema restituisca un errore e chieda di inserire nuovamente i dati;
Postcondizioni	/

Test T_2 (CU_2 - Effettua Logout):

ID	T_2.1
Descrizione	Verifica che il sistema effettui correttamente il logout.
Precondizioni	Vi è una sessione aperta per lo studente s1.
Azioni	<ol style="list-style-type: none"> 1. Si dà avvio alla procedura di logout; 2. Verificare che il sistema comunichi allo studente la disconnessione dal sistema e che venga visualizzata la pagina di login.
Postcondizioni	Non vi è più una sessione aperta per lo studente s1.

Test T_3 (CU_3 - Effettua Prenotazione):

ID	T_3.1
Descrizione	Verifica che il sistema effettui correttamente una prenotazione considerando che è il turno dello studente di prenotare, non è già stata effettuata la medesima prenotazione (confermata o non confermata) e vi sono posti in aula.
Precondizioni	Vi è una sessione aperta per lo studente s1. Il sistema contiene i dati su un edificio e1, dove all'interno si hanno tre aule a1, a2. Il sistema contiene i dati su un edificio e2, dove all'interno si hanno tre aule a3, a4.
Azioni	<ol style="list-style-type: none"> 1. Si dà avvio alla procedura di prenotazione; Verificare che il sistema mostri la lista degli edifici; 2. Si seleziona l'edificio e1; 3. Verificare che il sistema mostri la lista delle aule in cui compaiono le aule a1 e a2;

	<ol style="list-style-type: none"> 4. Si seleziona l'aula a1; 5. Sia S_n il numero di settimana corrente, verificare che che il sistema mostri una lista contenente i giorni $\{g_1, \dots, g_7\}$ della settimana S_{n+1} (la settimana successiva), e due campi inizio in1 e fine fn1 in cui selezionare la fascia oraria; 6. Si seleziona il giorno g1, inizio h1 e fine h2; 7. Verificare che il sistema chieda una conferma; 8. Si conferma la scelta; 9. Verificare che il sistema torni alla homepage;
Postcondizioni	Il Sistema contiene una nuova prenotazione confermata pc1 con attributi studente s1, aula a1, dataPrenotazione g1, inizio h1, fine h1.

ID	T_3.2
Descrizione	Verifica che il sistema generi un errore in fase di prenotazione considerando che non è il turno dello studente di prenotare.
Precondizioni	Vi è una sessione aperta per lo studente s1.
Azioni	<ol style="list-style-type: none"> 1. Si dà avvio alla procedura di prenotazione; 2. Verificare che il sistema restituisca un errore e torni alla homepage.
Postcondizioni	/

ID	T_3.3
Descrizione	Verifica che il sistema generi un errore in fase di prenotazione considerando che è il turno dello studente di prenotare ma è già stata effettuata la medesima prenotazione.
Precondizioni	<p>Vi è una sessione aperta per lo studente s1.</p> <p>Il sistema contiene i dati su un edificio e1, dove all'interno si hanno tre aule a1, a2. Il sistema contiene i dati su un edificio e2, dove all'interno si hanno tre aule a3, a4. Il Sistema contiene una prenotazione confermata pc1 con attributi studente s1, aula a1, dataPrenotazione g1, inizio h1, fine h1.</p>
Azioni	<ol style="list-style-type: none"> 1. Si dà avvio alla procedura di prenotazione; 2. Verificare che il sistema mostri la lista degli edifici; 3. Si seleziona l'edificio e1; 4. Verificare che il sistema mostri la lista delle aule in cui compaiono le aule a1 e a2; 5. Si seleziona l'aula a1; 6. Sia S_n il numero di settimana corrente, verificare che il sistema mostri una lista contenente i giorni $\{g_1, \dots, g_7\}$

	<p>della settimana S_{n+1} (la settimana successiva), e due campi inizio $in1$ e fine $fn1$ in cui selezionare la fascia oraria;</p> <ol style="list-style-type: none"> 7. Si seleziona il giorno $g1$, inizio $h1$ e fine $h2$; 8. Verificare che il sistema chieda una conferma; 9. Si conferma la scelta; 10. Verificare che il sistema generi un errore e si ritorni all'inserimento del giorno e della fascia oraria.
Postcondizioni	/

ID	T_3.4
Descrizione	Verifica che il sistema generi un errore in fase di prenotazione considerando che è il turno dello studente di prenotare ma lo studente è già iscritto alla waiting list ossia ha una prenotazione non confermata nell'aula e negli orari indicati in fase di prenotazione.
Precondizioni	<p>Vi è una sessione aperta per lo studente $s1$.</p> <p>Il sistema contiene i dati su un edificio $e1$, dove all'interno si hanno tre aule $a1$, $a2$. Il sistema contiene i dati su un edificio $e2$, dove all'interno si hanno tre aule $a3$, $a4$. Il Sistema contiene una prenotazione non confermata $nc1$ con attributi studente $s1$, aula $a1$, dataPrenotazione $g1$, inizio $h1$, fine $h1$.</p>
Azioni	<ol style="list-style-type: none"> 1. Si dà avvio alla procedura di prenotazione; 2. Verificare che il sistema mostri la lista degli edifici; 3. Si seleziona l'edificio $e1$; 4. Verificare che il sistema mostri la lista delle aule in cui compaiono le aule $a1$ e $a2$; 5. Si seleziona l'aula $a1$; 6. Sia S_n il numero di settimana corrente, verificare che il sistema mostri una lista contenente i giorni $\{g1, \dots, g7\}$ della settimana S_{n+1} (la settimana successiva), e due campi inizio $in1$ e fine $fn1$ in cui selezionare la fascia oraria; 7. Si seleziona il giorno $g1$, inizio $h1$ e fine $h2$; 8. Verificare che il sistema chieda una conferma; 9. Si conferma la scelta; 10. Verificare che il sistema generi un errore e si ritorni all'inserimento del giorno e della fascia oraria.
Postcondizioni	/

ID	T_3.5
Descrizione	Verifica che il sistema generi un errore in fase di prenotazione

	considerando che è il turno dello studente di prenotare, che lo studente non ha già effettuato la medesima prenotazione (confermata o non confermata), non vi sono posti in aula e lo studente acconsente ad entrare nella waiting list.
Precondizioni	Vi è una sessione aperta per lo studente s1. Il sistema contiene i dati su un edificio e1, dove all'interno si hanno tre aule a1, a2. Il sistema contiene i dati su un edificio e2, dove all'interno si hanno tre aule a3, a4. Il Sistema contiene una prenotazione confermata pc1 con attributi studente s1, aula a1, dataPrenotazione g1, inizio h1, fine h1. Il Sistema contiene una prenotazione non confermata nc1 con attributi studente s1, aula a1, dataPrenotazione g2, inizio h3, fine h4.
Azioni	<ol style="list-style-type: none"> 1. Si dà avvio alla procedura di prenotazione; 2. Verificare che il sistema mostri la lista degli edifici; 3. Si seleziona l'edificio e2; 4. Verificare che il sistema mostri la lista delle aule in cui compaiono le aule a3 e a4; 5. Si seleziona l'aula a3; 6. Sia Sn il numero di settimana corrente, verificare che il sistema mostri una lista contenente i giorni {g1,...,g7} della settimana Sn+1 (la settimana successiva), e due campi inizio in1 e fine fn1 in cui selezionare la fascia oraria; 7. Si seleziona il giorno g3, inizio h5 e fine h6; 8. Verificare che il sistema chieda una conferma; 9. Si conferma la scelta; 10. Verificare che il sistema generi un pop-up che chieda allo studente se si vuole iscrivere alla waiting list; 11. Si conferma la scelta; 12. Verificare che si ritorni alla homepage.
Postcondizioni	Il Sistema contiene una nuova prenotazione non confermata nc2 con attributi studente s1, aula a2, dataPrenotazione g3, inizio h5, fine h6.

ID	T_3.6
Descrizione	Verifica che il sistema generi un errore in fase di prenotazione considerando che è il turno dello studente di prenotare, che lo studente non ha già effettuato la medesima prenotazione (confermata o non confermata), non vi sono posti in aula e lo studente non acconsente ad entrare nella waiting list.
Precondizioni	Vi è una sessione aperta per lo studente s1. Il sistema contiene i dati su un edificio e1, dove all'interno si hanno tre aule a1, a2. Il sistema contiene i dati su un edificio

	e2, dove all'interno si hanno tre aule a3, a4. Il Sistema contiene una prenotazione confermata pc1 con attributi studente s1, aula a1, dataPrenotazione g1, inizio h1, fine h1. Il Sistema contiene una prenotazione non confermata nc1 con attributi studente s1, aula a1, dataPrenotazione g2, inizio h3, fine h4.
Azioni	<ol style="list-style-type: none"> 1. Si dà avvio alla procedura di prenotazione; 2. Verificare che il sistema mostri la lista degli edifici; 3. Si seleziona l'edificio e2; 4. Verificare che il sistema mostri la lista delle aule in cui compaiono le aule a3 e a4; 5. Si seleziona l'aula a3; 6. Sia Sn il numero di settimana corrente, verificare che il sistema mostri una lista contenente i giorni {g1,...,g7} della settimana Sn+1 (la settimana successiva), e due campi inizio in1 e fine fn1 in cui selezionare la fascia oraria; 7. Si seleziona il giorno g3, inizio h5 e fine h6; 8. Verificare che il sistema chieda una conferma; 9. Si conferma la scelta; 10. Verificare che il sistema generi un pop-up che chieda allo studente se si vuole iscrivere alla waiting list; 11. Non si conferma la scelta; 12. Verificare che si ritorni alla selezione del giorno e della fascia oraria.
Postcondizioni	/

ID	T_3.7
Descrizione	Verifica che il sistema generi un errore in fase di prenotazione considerando che è il turno dello studente di prenotare ma non conferma la prenotazione.
Precondizioni	Vi è una sessione aperta per lo studente s1. Il sistema contiene i dati su un edificio e1, dove all'interno si hanno tre aule a1, a2. Il sistema contiene i dati su un edificio e2, dove all'interno si hanno tre aule a3, a4.
Azioni	<ol style="list-style-type: none"> 1. Si dà avvio alla procedura di prenotazione; 2. Verificare che il sistema mostri la lista degli edifici; 3. Si seleziona l'edificio e1; 4. Verificare che il sistema mostri la lista delle aule in cui compaiono le aule a1 e a2; 5. Si seleziona l'aula a1; 6. Sia Sn il numero di settimana corrente, si verifica che il sistema mostri una lista contenente i giorni {g1,...,g7} della settimana Sn+1 (la settimana successiva), e due

	campi inizio in1 e fine fn1 in cui selezionare la fascia oraria; 7. Si seleziona il giorno g1, inizio h1 e fine h2; 13. Verificare che il sistema chieda una conferma; 14. Non si conferma la scelta; 15. Verificare che si ritorni alla selezione del giorno e della fascia oraria.
Postcondizioni	/

Test T_4 (CU_4 - Visualizza Prenotazioni Confermate):

ID	T_4.1
Descrizione	Verifica che il sistema visualizzi le prenotazioni confermate dello studente che ne ha fatto richiesta.
Precondizioni	Il sistema contiene uno studente s1, uno studente s2, una prenotazione confermata co1 legata allo studente s1 e una prenotazione confermata co2 legata allo studente s2. Vi è una sessione aperta per lo studente s1.
Azioni	1. Si dà avvio alla procedura di visualizzazione delle prenotazioni confermate legate allo studente s1; 2. Verificare che il sistema mostri la lista in cui compare solo la prenotazione confermata co1.
Postcondizioni	/

ID	T_4.2
Descrizione	Verifica che il sistema visualizzi una lista vuota nel caso in cui lo studente non abbia alcuna prenotazione confermata.
Precondizioni	Il sistema contiene uno studente s1 e non contiene nessuna prenotazione confermata legata allo studente s1. Vi è una sessione aperta per lo studente s1.
Azioni	1. Si dà avvio alla procedura di visualizzazione delle prenotazioni confermate dello studente s1; 2. Verificare che il sistema mostri una lista vuota.
Postcondizioni	/

Test T_5 (CU_5 - Stampa Ricevuta):

ID	T_5.1
Descrizione	Verifica che il sistema scarichi un pdf della ricevuta di prenotazione.

Precondizioni	Il sistema contiene una prenotazione confermata pc1 nell'aula a1, nel giorno g1 e nella fascia oraria h1/h2 legata allo studente s1. Vi è una sessione aperta per lo studente s1.
Azioni	<ol style="list-style-type: none"> 1. Si dà avvio alla procedura di scaricamento della ricevuta di prenotazione pc1; 2. Verificare che il sistema scarichi in locale la ricevuta relativa alla prenotazione pc1.
Postcondizioni	/

Test T_6 (CU_ 6 - Cancella Prenotazione Confermata):

ID	T_6.1
Descrizione	Verifica che il sistema cancelli una prenotazione confermata nel caso in cui non vi siano studenti nella waiting list.
Precondizioni	Il sistema contiene una prenotazione confermata pc1 nell'aula a1, nel giorno g1 e nella fascia oraria h1/h2 legata allo studente s1. Vi è una sessione aperta per lo studente s1. Il Sistema non contiene una prenotazione non confermata nc1 nell'aula a1, nel giorno g1 e nella fascia oraria h1/h2.
Azioni	<ol style="list-style-type: none"> 1. Si dà avvio alla procedura di cancellazione della prenotazione confermata pc1; 2. Verificare che il sistema chieda una conferma; 3. Si conferma la scelta; 4. Verificare che il sistema torni a visualizzare la lista delle prenotazioni confermate.
Postcondizioni	Il Sistema non contiene una prenotazione confermata pc1 nell'aula a1, nel giorno g1 e nella fascia oraria h1/h2 legata allo studente s1.

ID	T_6.2
Descrizione	Verifica che il sistema cancelli una prenotazione confermata nel caso in cui vi siano studenti nella waiting list. In tal caso il sistema confermerà la prenotazione del primo studente nella waiting list.
Precondizioni	<p>Il sistema contiene una prenotazione confermata pc1 nell'aula a1, nel giorno g1 e nella fascia oraria h1/h2 legata allo studente s1. Vi è una sessione aperta per lo studente s1.</p> <p>Il Sistema contiene una prenotazione non confermata nc1 nell'aula a1, nel giorno g1 e nella fascia oraria h1/h2 relativa allo studente s2.</p> <p>Il Sistema contiene una prenotazione non confermata nc2</p>

	nell'aula a1, nel giorno g1 e nella fascia oraria h1/h2 relativa allo studente s3. La prenotazione nc1 è stata fatta nell'orario o1, la prenotazione nc2 è stata fatta nell'orario nc2 ed $nc1 < nc2$.
Azioni	<ol style="list-style-type: none"> 1. Si dà avvio alla procedura di cancellazione della prenotazione confermata pc1; 2. Verificare che il sistema chieda una conferma; 3. Si conferma la scelta; 4. Verificare che il sistema torni a visualizzare la lista delle prenotazioni confermate.
Postcondizioni	<p>Il Sistema non contiene una prenotazione confermata pc1 nell'aula a1, nel giorno g1 e nella fascia oraria h1/h2 legata allo studente s1.</p> <p>Il Sistema contiene una prenotazione confermata nc1 nell'aula a1, nel giorno g1 e nella fascia oraria h1/h2 legata allo studente s2.</p> <p>Il Sistema non contiene una prenotazione non confermata nc1 nell'aula a1, nel giorno g1 e nella fascia oraria h1/h2 legata allo studente s2.</p>

ID	T_6.3
Descrizione	Verifica che il sistema annulli l'operazione di cancellazione di una prenotazione confermata se lo studente non conferma la scelta.
Precondizioni	Il sistema contiene contiene una prenotazione confermata pc1 nell'aula a1, nel giorno g1 e nella fascia oraria h1/h2 legata allo studente s1. Vi è una sessione aperta per lo studente s1.
Azioni	<ol style="list-style-type: none"> 1. Si dà avvio alla procedura di cancellazione della prenotazione confermata pc1; 2. Verificare che il sistema chieda una conferma; 3. Non si conferma la scelta; 4. Verificare che il sistema torni a visualizzare la lista delle prenotazioni confermate.
Postcondizioni	/

ID	T_6.4
Descrizione	Verifica che il sistema annulli l'operazione di cancellazione di una prenotazione confermata se questa viene fatta oltre i termini non consentiti.
Precondizioni	Il sistema contiene contiene una prenotazione confermata pc1 nell'aula a1, nel giorno g1 e nella fascia oraria h1/h2 legata

	allo studente s1. Vi è una sessione aperta per lo studente s1.
Azioni	<ol style="list-style-type: none"> 1. Si dà avvio alla procedura di cancellazione della prenotazione confermata pc1; 2. Verificare che il sistema chieda una conferma; 3. Si conferma la scelta; 4. Verificare che il sistema mostri un errore e che si torni a visualizzare la lista delle prenotazioni confermate.
Postcondizioni	/

Test T_7 (CU_ 7 - Visualizza Prenotazioni Non Confermate):

ID	T_7.1
Descrizione	Verifica che il sistema visualizzi le prenotazioni non confermate dello studente che ne ha fatto richiesta.
Precondizioni	Il sistema contiene uno studente s1, uno studente s2, una prenotazione non confermata nc1 legata allo studente s1 e una prenotazione non confermata nc2 legata allo studente s2. Vi è una sessione aperta per lo studente s1.
Azioni	<ol style="list-style-type: none"> 1. Si dà avvio alla procedura di visualizzazione delle prenotazioni non confermate legate allo studente s1; 2. Verificare che il sistema mostri la lista in cui compare solo la prenotazione non confermata nc1.
Postcondizioni	/

ID	T_7.2
Descrizione	Verifica che il sistema visualizzi una lista vuota nel caso in cui lo studente non abbia alcuna prenotazione non confermata.
Precondizioni	Il sistema non contiene nessuna prenotazione non confermata legata allo studente s1. Vi è una sessione aperta per lo studente s1.
Azioni	<ol style="list-style-type: none"> 1. Si dà avvio alla procedura di visualizzazione delle prenotazioni non confermate legate allo studente s1; 2. Verificare che il sistema mostri una lista vuota.
Postcondizioni	/

Test T_8 (CU_ 8 - Cancella Prenotazione Non Confermata):

ID	T_8.1
Descrizione	Verifica che il sistema cancelli correttamente una prenotazione

	non confermata.
Precondizioni	Il sistema contiene una prenotazione non confermata nc1 legata allo studente s1. Vi è una sessione aperta per lo studente s1.
Azioni	<ol style="list-style-type: none"> 1. Si dà avvio alla procedura di cancellazione della prenotazione non confermata nc1; 2. Verificare che il sistema chieda una conferma; 3. Si conferma la scelta; 4. Verificare che si torni alla visualizzazione della lista delle prenotazioni non confermate.
Postcondizioni	Il sistema non contiene una prenotazione non confermata nc1 legata allo studente s1.

ID	T_8.2
Descrizione	Verifica che il sistema non cancelli una prenotazione non confermata nel caso in cui lo studente non confermi la scelta.
Precondizioni	Il sistema contiene una prenotazione non confermata nc1 legata allo studente s1. Vi è una sessione aperta per lo studente s1.
Azioni	<ol style="list-style-type: none"> 1. Si dà avvio alla procedura di cancellazione della prenotazione non confermata nc1; 2. Verificare che il sistema chieda una conferma; 3. Non si conferma la scelta; 4. Verificare che si torni alla visualizzazione della lista delle prenotazioni non confermate.
Postcondizioni	/