

# **Progetto 2 - Gestionale per l'occupazione delle aule e delle stanze di studio e lavoro**

**"Software Engineering"  
Course a.a. 2019 - 2020**

Lecturer: Prof. Henry Muccini  
([henry.muccini@univaq.it](mailto:henry.muccini@univaq.it))

Date: 05 / 12 / 2019  
Deliverable: Milestone #1  
Team name: GitPub

# Table of contents

0   <u>List of challenging task / risky requirements</u>	3
1   <u>State of the art</u>	4
<u>Università Cattolica del Sacro Cuore (MI)</u>	4
<u>Università di Padova</u>	5
<u>The Space Cinema</u>	6
<u>Optix</u>	7
2   <u>Interview to Stakeholders</u>	10
3   <u>Requirements refining</u>	12
3.1   <u>Services</u>	12
3.2   <u>Non-functional requirements</u>	13
3.3   <u>Detailed Scenarios</u>	14
3.4   <u>Excluded Requirements</u>	18
3.5   <u>Assunzioni</u>	19
3.6   <u>Use Case Diagram</u>	20
4   <u>Software Architecture</u>	21
4.1   <u>Component Diagram</u>	21
4.2   <u>Sequence Diagram</u>	22
5   <u>Data and Data models</u>	24
6   <u>Design Decisions</u>	26
7   <u>Low Level Design</u>	27
7.1   <u>Class Diagram</u>	27
7.2   <u>API Design and Documentation</u>	27
8   <u>Explain how the FRs and the NFR are satisfied</u>	28
8.2   <u>Nonfunctional Requirements</u>	28
9   <u>Effort Recording</u>	29

<b>Team Members</b>		
<b>Name &amp; Surname</b>	<b>Matriculation Number</b>	<b>E-mail address</b>
Natan Cieplinski	251818	natan.cieplinski@student.univaq.it
Alessandro Lodi	253383	alelod96@gmail.com
Patrizia Villani	253382	patrizia.vil@icloud.com

## 0 | List of challenging task / risky requirements

Challenging task: Progettazione di un sistema di controllo di occupazione del posto

Day the task is identified: 26/11/2019

Day the challenge is resolved: 1/12/2019

Explanation on how the challenge has been managed: Si è deciso di utilizzare dei beacon bluetooth piuttosto che il Wi-Fi dell'università. Sono stati controllati i costi dei beacon prima di prendere la decisione finale.

Challenging task: Identificazione di un sistema di check-in

Day the task is identified: 26/11/2019

Day the challenge is resolved: 16/12/2019

Explanation on how the challenge has been managed: Si è ipotizzato di eliminare un sistema check-in e appoggiarsi al sistema di controllo della postazione per verificare se l'utente è uscito o entrato dall'aula. Questa possibilità è stata adottata a seguito di una discussione con il professore e all'adozione del bluetooth per validare la presenza in aula.

Challenging task: Sistema di segnalazione

Day the task is identified: 1/12/2019

Day the challenge is resolved: 2/12/2019

Explanation on how the challenge has been managed: Si è deciso di lasciare agli utenti la possibilità di fare una segnalazione solo riguardante la propria postazione, nel caso questa sia stata occupata da qualcuno.

Challenging task: Recupero dei dati personali dell'utente

Day the task is identified: 1/12/2019

Day the challenge is resolved: 4/12/2019

Explanation on how the challenge has been managed: Si è deciso di ottenere i dati personali dell'utente da parte dell'università in modo da evitare ridondanze nei database.

# 1 | State of the art

Abbiamo trovato ed analizzato principalmente tre sistemi: un progetto dell'Università Cattolica del Sacro Cuore, l'app Affluences utilizzata dall'università di Padova, il sistema di prenotazione dei posti nel cinema The Space Cinema di Montesilvano e la piattaforma Optix per la gestione degli spazi di coworking.

## Università Cattolica del Sacro Cuore (MI)

Dal 12 febbraio 2018 la Biblioteca offre la possibilità di utilizzare, in modalità sperimentale, il servizio StudyTime per la prenotazione di posti di studio individuale.

StudyTime consente di razionalizzare la fruizione dei posti di studio in Biblioteca, evitando che siano occupati in modo improprio e con tempi di permanenza eccessivi, e assicura agli utenti ulteriori vantaggi:

- Prenotare comodamente (in anticipo e a distanza) il proprio posto di studio.
- verificare la disponibilità di posti liberi e vederne indicata la posizione in sala.
- gestire le proprie prenotazioni, anche annullandole o mantenendole in caso di breve assenza dal posto utilizzato.

StudyTime funziona integrando diversi strumenti:

- il sito web **StudyTime** ([studytime.unicatt.it](http://studytime.unicatt.it)).
- l'app **Juno StudyTime** da installare nel proprio smartphone (prodotta da Telepen, disponibile per Apple iOS e Android).
- rilevatori **bluetooth** (beacons) distribuiti all'interno della sala.

**Come si utilizza StudyTime:**

### Prenotazione a distanza (tramite sito web):

- accedere al sito web [studytime.unicatt.it](http://studytime.unicatt.it)
- dove richiesto il numero della propria tessera universitaria, inserire la propria "matricola Biblioteca".
- seguire le istruzioni per selezionare la sala, il giorno, il posto e l'orario (tra quelli indicati disponibili) e infine confermare.
- attendere che al proprio indirizzo (@icatt.it) sia arrivata l'e-mail di conferma dell'avvenuta prenotazione.

### Conferma in sala della prenotazione (tramite app per smartphone):

Requisiti necessari:

Preliminarmente occorre installare nel proprio smartphone l'app Juno StudyTime (disponibile per Apple iOS e Android) e procedere alla propria registrazione nell'app (solo la prima volta) inserendo:

- "UCSC" come nome dell'istituzione di appartenenza.
- il proprio indirizzo e-mail istituzionale (@icatt.it).

Prima di iniziare a utilizzare il posto, è necessario recarsi nella sala per confermare la prenotazione entro 15 minuti dopo l'ora d'inizio della sua utilizzabilità:

- nel proprio smartphone aprire l'app Juno StudyTime e attivare la connessione bluetooth.
- connettersi ai beacons bluetooth

- selezionare "Conferma la tua prenotazione".
- confermare la prenotazione inquadrando fino a riconoscimento avvenuto il codice a barre posizionato sul tavolo.
  - iniziare a utilizzare il posto

### **Pausa durante l'utilizzazione del posto:**

Nel periodo di utilizzabilità è consentito assentarsi dal posto per un tempo massimo di 15 minuti (anche frazionabili).

Per avviare la pausa:

- entrare nell'app Juno StudyTime e scegliere "Conferma la tua prenotazione".
- inquadrare fino a riconoscimento avvenuto il codice a barre sul tavolo.
- avviare il contatore.

Per concludere la pausa:

- entrare nell'app Juno StudyTime e scegliere "Conferma la tua prenotazione".
- inquadrare fino a riconoscimento avvenuto il codice a barre sul tavolo.
- arrestare il contatore.

Se l'assenza sarà prolungata oltre i 15 minuti complessivi, la prenotazione verrà automaticamente annullata e il posto dovrà essere lasciato libero.

La **fine** della sperimentazione è avvenuto il 23 marzo 2018

### **Analisi**

Abbiamo ritenuto che il seguente sistema sia eccessivamente complicato per lo studente rispetto ai servizi che offre. Il meccanismo per fare una pausa è a nostro parere macchinoso (troppe scansioni del codice a barre sul tavolo) e non assicura che lo studente sia effettivamente in aula, in quanto la persona potrebbe decidere di fare una pausa più lunga di 15 minuti a patto che il telefono sia lasciato in aula. Tuttavia il sistema analizzato ci ha dato ispirazione ad implementare un meccanismo di gestione dei momenti di pausa.

Inoltre il sistema richiederebbe l'acquisto di diversi rilevatori bluetooth, una soluzione che potrebbe diventare molto dispendiosa nel caso in cui si voglia adattare il sistema per utilizzarlo in ogni aula di ogni polo della facoltà invece che solo in una biblioteca.

## **Università di Padova**

Dall'11 novembre 2019 la biblioteca utilizza, **in via sperimentale**, un App che ti permette di trovare e prenotare una postazione studio con Bloomberg o Datastream, il servizio è disponibile on-line con:

- l'app gratuita per Android e iPhone
- la pagina

Le postazioni sono prenotabili a distanza di massimo 7 giorni, per periodi di 30 minuti e fino a 2 ore complessive per singola prenotazione (che potrà essere ripetuta anche in giornata, in assenza di prenotazioni di altri utenti).

È consentita una pausa di non oltre 10 minuti, avvisando i bibliotecari in servizio al banco.

### **Come si utilizza Affluences**

- Aprire l'app e selezionare la biblioteca nella quale prenotare il posto, l'orario di inizio prenotazione e la durata della prenotazione.

- Confermare la prenotazione seguendo le istruzioni contenute nell'e-mail che Affluences invierà all'indirizzo indicato dall'utente durante la compilazione della richiesta.
- Una volta confermata la prenotazione, per poter accedere alla postazione l'utente deve mostrare ai bibliotecari in servizio al banco il codice contenuto nell'e-mail ricevuta da Affluences.

Alcune Screenshot dell'app:



#### Benvenuti su Affluences

L'app che vi comunica in tempo reale il numero di persone presenti in più di cento istituti.

Momentaneamente la sperimentazione è ancora in atto.

---

## Analisi

Il sistema ci è sembrato un'alternativa più semplice per l'utente rispetto a quello precedentemente analizzato, tuttavia non lo abbiamo ritenuto adatto per l'utilizzo anche in aule studio. Affluences è stato progettato per le biblioteche e di conseguenza si appoggia molto sul sorvegliante del quale ogni biblioteca dispone. Il sistema quindi richiederebbe l'assunzione di un sorvegliante per ogni aula dell'università che si vuole mettere a disposizione per studiare. Sebbene l'assunzione di un sorvegliante per aula sarebbe la soluzione più sicura dal punto di vista del controllo delle postazioni, riteniamo che renderebbe l'adozione del sistema troppo costosa per le università.

## The Space Cinema

Il cinema dispone di un sistema di prenotazione dei posti nelle sale.

Il servizio è disponibile utilizzando il sito web ufficiale di The Space.

E' consentito prenotare posti in aula scegliendoli dall'apposita mappa interattiva. I posti verranno assegnati alla persona nel suo biglietto virtuale e la persona avrà diritto ad utilizzare i posti prenotati. Nel caso la persona trovi il proprio posto occupato, può chiamare uno dei responsabili i quali agiranno di conseguenza.

## Come si effettua la prenotazione

- Aprire il sito e selezionare il film alla quale si è interessati.
- Selezionare l'orario al quale si è interessati.
- Dalla mappa dei posti, selezionare i posti che si vuole prenotare.

Alcune screenshots del sito:

The first screenshot shows a seating chart for a room labeled '1'. The chart is a grid of seats labeled A through U on the left and 1 through 10 at the top. Seats are marked with icons indicating their status: a black square for 'Il tuo posto' (your seat), grey squares for 'Posti occupati' (occupied seats), white squares for 'Posti non disponibili' (unavailable seats), an orange square for 'Poltrona VIP' (VIP seat), an empty square for 'Poltrone standard' (standard seats), and a blue square with a person icon for 'Spazio riservato alle persone disabili' (space reserved for disabled people). A legend at the bottom defines these symbols.

The second screenshot shows the movie details for 'CENA CON DELITTO - KNIVES OUT' (Azione). It includes the poster, a brief plot summary, and showtimes for Bologna on December 5th, 2019. The showtimes are 15:50, 18:00, 21:10, and 22:10. Each showtime has a 2D option. A button at the bottom right says '+ VEDI GLI ORARI DI TUTTI I GIORNI'.

The third screenshot shows the same movie details but for DOMANI GIO 05 DICEMBRE. The showtimes are 15:50, 18:00, 21:10, and 22:10, all in 2D. A button at the bottom right says '+ MOSTRA GLI ORARI DI TUTTI I GIORNI'.

## Analisi

Sebbene in questo caso il sistema analizzato sia stato progettato per un contesto molto diverso dal nostro, offre degli spunti interessanti.

L'interfaccia è molto intuitiva e permette di riservare un posto preciso, che nel caso del sistema per le aule sarebbe vantaggioso per un gruppo di persone che vogliono studiare insieme. Inoltre il cinema gestisce l'assegnazione dei posti con un solo sorvegliante per tutte le sale, che può essere chiamato nel caso in cui una persona occupi un posto che non gli spetta. Questo ci ha dato spunto per un sistema di segnalazioni da integrare nel nostro sistema, da utilizzare nel caso in cui uno studente trovi una persona seduta al posto che aveva prenotato.

## Optix

Optix è una piattaforma che permette di gestire gli spazi di coworking. Il servizio è disponibile tramite app e sito web. Le funzionalità sono:

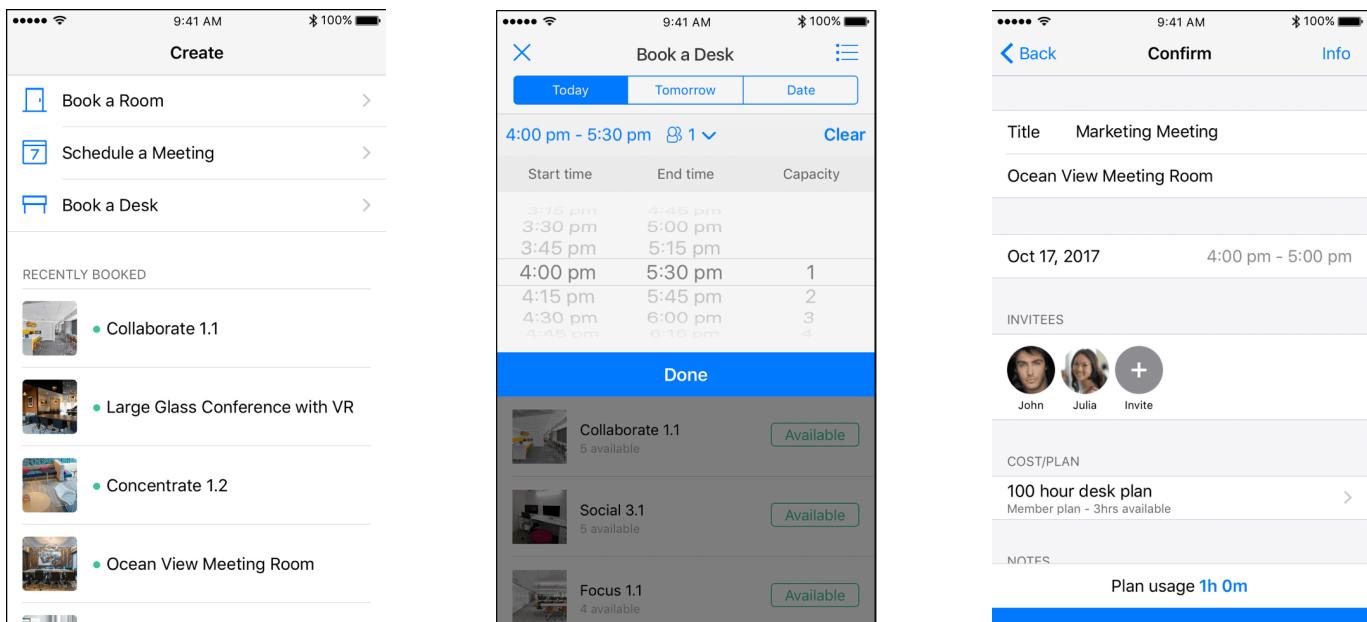
- Prenotazione di stanze.
- Prenotazione di scrivanie.
- Fatturazione delle prenotazioni.

- Analytics sui dati di prenotazione.
- Segnalazione di problemi.

## Come si utilizza Optix

### Prenotazione di una sala o di una scrivania:

- Aprire l'app e fare l'accesso con le credenziali del portale o con le credenziali fornite dalla propria organizzazione.
- Entrare nella sezione prenotazioni e decidere se prenotare una sala o una scrivania.
- Procedere alla prenotazione specificando l'ora di ingresso e di uscita, e la stanza se si intende prenotare una scrivania.



### Configurazione delle scrivanie da mettere a disposizione:

- Aprire la versione web desktop dell'app.
- Entrare nella sezione spaces.
- Creare un nuovo space, specificando che il tipo è una dedicated desk. Specificare il numero di posti della scrivania e una descrizione.

## Analisi

Il sistema analizzato è molto vicino ad una possibile soluzione al problema che ci è stato presentato con la specifica. In particolare:

- La funzione di prenotazione di una sala è uguale alla funzione di prenotazione di un'aula da parte di un professore.
- La funzione di prenotazione di una scrivania è uguale alla funzione di prenotazione di un posto da parte di uno studente.
- Il sistema permette di effettuare segnalazioni, una funzionalità che era già stata presa in considerazione con l'analisi del sistema di prenotazione dei biglietti del cinema The Space.
- Il sistema permette di definire le scrivanie a disposizione. Il servizio è da ispirazione per un sistema di configurazione delle mappe delle aule che non richieda l'inserimento di una mappa vera e propria.

Tuttavia la pecca principale che ha il sistema analizzato è la mancanza del monitoraggio di chi usufruisce delle postazioni. In questo caso gli utenti sono motivati a rispettare gli orari di prenotazione e ad usufruire delle postazioni dal fatto che il tempo prenotato viene pagato dall'utente. Nel caso del sistema che ci è stato richiesto, è necessario progettare un modo per controllare le postazioni.

## 2 | Interview to Stakeholders

Abbiamo intervistato gli stakeholder del sistema usando due metodologie.

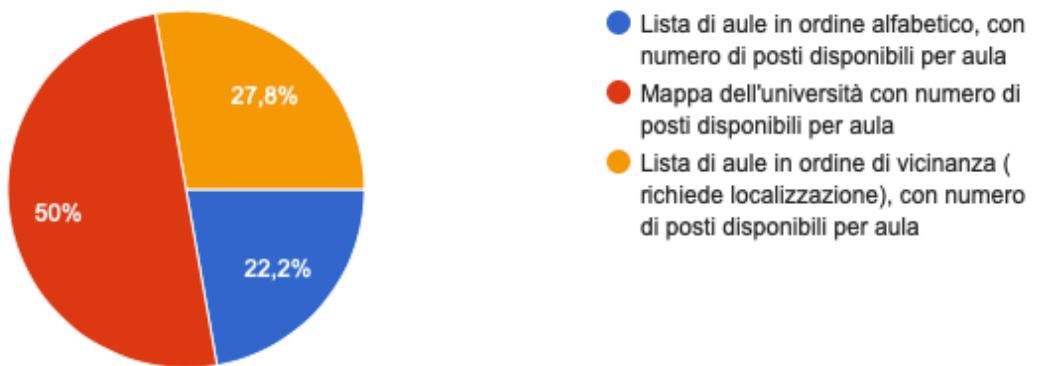
Per gli studenti abbiamo utilizzato un form Google con alcune domande, mentre per quanto riguarda i sorveglianti (ovvero le persone che si occupano di aprire e chiudere le aule nei vari edifici) abbiamo utilizzato un approccio diretto di conversazione.

### **Studenti:**

Dal form Google è emerso che la maggior parte degli studenti preferirebbe vedere una vista geolocalizzata delle aule rispetto a una lista delle stesse.

#### **Quale sarebbe il modo più comodo per visualizzare i posti disponibili?**

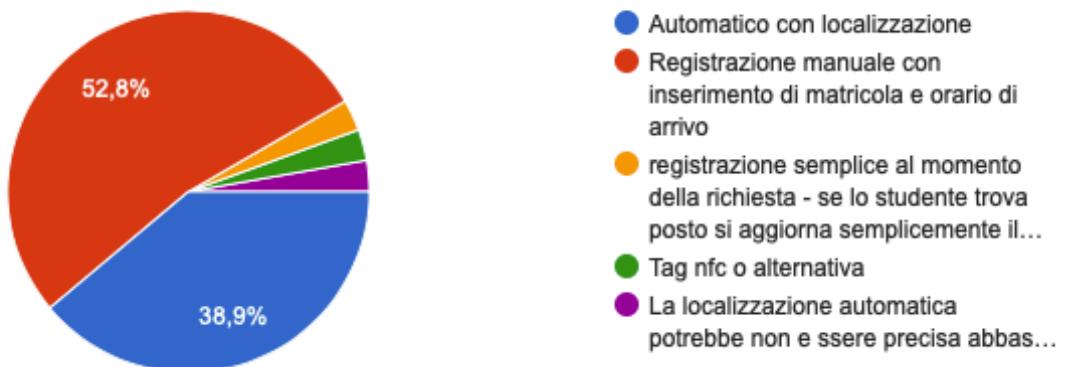
36 risposte



Inoltre gli studenti hanno mostrato di preferire un check-in manuale / fisico rispetto ad uno automatico che utilizzi la localizzazione.

#### **E' necessario registrare l'arrivo in aula, quale metodo ti sembra più comodo?**

36 risposte



Altre idee che qualche studenti hanno suggerito sono state:

- Possibilità di prenotare per gruppi o cercare gruppi di studio.

- Classificare le aule per livello di rumorosità consentito.
- Far indicare agli studenti se hanno intenzione di utilizzare o meno la presa di corrente in modo da poter sapere in anticipo se saranno disponibili o meno delle prese
- Modificare il tempo massimo per il quale una postazione è prenotabile a seconda dell'affollamento dell'aula, per cercare di massimizzare il numero di persone che possono usufruire del servizio.

## Sorveglianti:

Il primo problema che ci siamo posti è stato capire come funzionano l'apertura e la chiusura delle aule quando non ci sono lezioni. Abbiamo parlato con il personale presente ai box informazione dei poli di Coppito 1 e Coppito 2, il quale si occupa anche dell'apertura e chiusura delle aule.

Ci è stato detto che l'aula normalmente viene lasciata aperta anche quando la lezione finisce, per lasciare la possibilità agli studenti di poter studiare. Tuttavia non c'è un iter preciso, e la chiusura dell'aula è a discrezione del personale.

A seguito di queste informazioni, abbiamo deciso che le aule, durante le lezioni verranno mostrate aperte ma occupate, mentre a fine lezione verranno mostrate aperte e saranno i sorveglianti a decidere quando chiuderle, inserendo l'informazione successivamente nel sistema. Inoltre i sorveglianti hanno espresso la preferenza di poter specificare per quale motivo l'aula è stata chiusa (es. manutenzione, pulizia, orario di chiusura, comportamento scorretto da parte di studenti ecc.).

Un'altra problematica che abbiamo analizzato è stata sapere con quale certezza si potesse prenotare giorni prima un'aula normale in orario non di lezione, dato che non sapevamo se le aule si potessero considerare sempre aperte o aperte solo per le lezioni.

Inizialmente avevamo pensato di lasciare la possibilità di prenotare in anticipo solo nelle stanze certamente aperte (es. biblioteca), e di rendere aperte a prenotazioni le altre aule solo nel caso in cui il sorvegliante avesse deciso di lasciarla aperte.

A seguito della discussione con il personale, dato che ci è stato comunicato che normalmente le aule vengono lasciate aperte anche quando non c'è lezione, possiamo considerare tutte le aule come sempre aperte (al pari della biblioteca) e di conseguenza possiamo permettere agli studenti di prenotare un posto anche giorni prima.

## 3 | Requirements refining

### 3.1 | Services

---

#### Sistema di login

Il sistema dovrà permettere di effettuare il login con le credenziali della propria università. Successivamente, tramite delle API fornite dall'università, verranno inviate al sistema le seguenti informazioni sull'utente:

- nome
- cognome
- ruolo (studente, professore, amministratore, sorvegliante)

Importanza: Alta

Complessità: Alta

---

#### Mappa universitaria

Il sistema dovrà permettere di aggiungere una planimetria dell'università. Le aree interattive sulla mappa verranno impostate definendo i vertici di ogni aula. Una volta definite le aule, il sistema scaricherà tramite API fornite dall'università l'orario delle lezioni per definire quali aule sono aperte, e quali aperte ma occupate. I dati verranno scaricati periodicamente in modo da assicurare che il sistema sia aggiornato rispetto ai cambiamenti di orario delle lezioni.

Importanza: Alta

Complessità: Alta

---

#### Configurazione dell'aula

Il sistema permetterà di configurare la distribuzione dei posti all'interno di un'aula. La configurazione verrà fatta utilizzando dei modelli predefiniti di banchi. I modelli predefiniti di banchi (es. fila da 10, fila da 20, banco da 8 ecc.) verranno creati alla prima configurazione del sistema e potranno essere modificati in seguito. Si indicheranno inoltre in quante file e colonne i banchi sono disposti e il loro orientamento (verticale o orizzontale).

Il sistema in base alla configurazione dei posti, costruirà una mappa.

Importanza: bassa

Complessità: media

*Spiegazione dettagliata in Design Decisions*

---

#### Apertura e chiusura aula

Il sistema permetterà di modificare lo stato di un'aula da aperta a chiusa e viceversa. Lo stato di default di ogni aula sarà aperta. Ogni aula avrà un orario di apertura e uno di chiusura di default che viene impostato alla prima configurazione

Importanza: bassa

Complessità: bassa

---

#### Prenotazione

Il sistema permetterà ad uno studente di prenotare un posto in aula. La prenotazione sarà costituita da una data e da una durata. Ogni prenotazione sarà annullabile. Il posto verrà scelto tramite la mappa interattiva e allo studente verrà comunicato il numero di banco che gli è stato assegnato.

Per un professore invece, ci sarà la possibilità di prenotare un'intera aula. La prenotazione di un professore avrà priorità su quella degli studenti, pertanto quando un professore prenoterà un'aula, tutte le prenotazioni degli studenti in quell'aula verranno cancellate, e gli studenti verranno notificati.

Importanza: alta.

Complessità: media.

---

## Sistema di segnalazioni

Il sistema permetterà agli studenti di segnalare quando una persona occupa il posto che lo studente ha prenotato.

Inoltre, il sorvegliante avrà la possibilità di segnalare un utente per comportamenti inadatti. Dopo una quantità di segnalazioni (il cui valore verrà definito in fase di configurazione) un utente potrà essere bannato.

---

## Check-in e check-out

Il sistema permetterà allo studente di effettuare il check-in in aula. Il check-in verrà eseguito connettendosi a dei beacon bluetooth.

Il beacon bluetooth inoltre verrà utilizzato per controllare l'effettiva presenza dello studente in aula. Il beacon trasmetterà un segnale bluetooth che dovrà essere ricevuto dall'app per validare la presenza dello studente in aula. Il professore non avrà bisogno della validazione del beacon per fare il check-in.

Sia al professore che allo studente sarà permesso di fare check-out tramite un semplice pulsante nell'applicazione, nel caso in cui si finisca prima del previsto e non si necessiti più di un'aula o una postazione di studio. Se lo studente dimentica di fare il checkout, l'assenza di ricezione dal beacon bluetooth verrà utilizzata per effettuare un checkout automatico. A seguito di un check-out automatico il sistema manderà una notifica allo studente.

Importanza: alta

Complessità: media

---

## Sistema di pause

Lo studente avrà la possibilità di fare una pausa. La durata massima della pausa verrà definita in fase di configurazione del software. Durante la pausa, lo studente potrà allontanarsi dall'aula, scollegandosi dal beacon bluetooth. Alla fine della pausa, lo studente si ricollegherà al beacon bluetooth. In caso contrario, la prenotazione del suo posto verrà cancellata e lo studente verrà notificato.

---

## Definizione delle priorità su prenotazione

Il sistema permetterà in fase di configurazione di definire i processi con i quali gestire le prenotazioni. Sarà possibile scegliere se non stabilire alcun processo, stabilire la priorità di un professore, di uno studente e stabilire i tempi massimi o minimi entro i quali effettuare una prenotazione.

## 3.2 | Non-functional requirements

### Usability

**Learnability:** il sistema per gli utente è utilizzabile tramite un sistema di "mappe". E' pertanto molto intuitivo da utilizzare.

### Security

**Confidentiality:** Il sistema gestisce l'accesso alla visualizzazione delle prenotazioni solo al personale autorizzato

### Maintainability

**Reusability:** il sistema può essere facilmente esteso per diventare un sistema di prenotazione delle aule anche per le lezioni ordinarie, e per registrare le presenze nelle lezioni a frequenza obbligatoria.

### Portability

**Adaptability:** il sistema è pensato per essere utilizzato da più università tramite delle API.

## 3.3 | Detailed Scenarios

### 3.3.1 | Actors

Sono stati identificati i seguenti attori:

- Utenti:
  - Studenti
  - Professori
  - Amministratori
  - Sorveglianti
- Sistemi informatici universitari

### 3.3.2 | Scenarios

**Nome:** Prenotazione aula da parte di un utente

**Partecipanti:** Studenti, Professori

**Flusso di eventi:**

- L'utente apre l'app
- L'utente seleziona la data e l'ora per il quale ha intenzione di prenotare
- Il sistema in base al processo per tipo di utente definito in fase di configurazione decide come trattare la prenotazione
- In caso la prenotazione venga accettata l'utente riceve una mail di notifica di avvenuta prenotazione

**Nome:** Pausa da parte di uno studente

**Partecipanti:** Studente

**Flusso di eventi:**

- Lo studente apre l'app
- Lo studente apre la schermata per la pausa e indica che è in pausa
- L'app fa partire il timer di tempo definito in fase di configurazione
- Al termine della pausa l'app cerca il segnale del beacon bluetooth. Se non lo trova, annulla la prenotazione dello studente e gli invia una notifica

**Nome:** Check-in in aula da parte di un utente

**Partecipanti:** Studenti, Professori

**Flusso di eventi:**

- L'utente entra in aula
- L'utente apre l'app per fare il check-in
  - L'app comincia la ricerca del segnale bluetooth trasmesso dal beacon per validare il check-in
    - L'app riceve il segnale dal beacon e comunica con il server che il telefono dello studente risulta in aula
    - Lo studente viene registrato come presente in aula
    - Se lo studente non effettua il check-in entro la quantità di tempo configurata, la prenotazione viene annullata.
  - Se l'utente è un professore, segnala che è arrivato in aula senza bisogno di collegarsi al beacon
- La prenotazione dell'utente viene confermata
- Se entro una quantità di tempo definita in fase di configurazione lo studente non esegue il check-in la sua prenotazione viene cancellata e lo studente viene notificato.

**Nome:** Check-out da parte di un utente

**Partecipanti:** Studenti, Professori

**Flusso di eventi:**

- L'utente apre l'app
- L'utente fa il checkout
- Il sistema modifica l'orario di fine prenotazione della prenotazione attiva nel momento corrente per rendere il posto o l'aula disponibile.
- Se lo studente dimentica di fare il checkout il sistema lo esegue in automatico una volta che lo studente si è scollegato dal beacon.
- Viene inviata una notifica allo studente.

**Nome:** Annullamento di una prenotazione da parte di un utente

**Partecipanti:** Studenti, Professori

**Flusso di eventi:**

- L'utente apre l'app
- L'utente apre la sua lista di prenotazioni
- L'utente decide la prenotazione da eliminare e la elimina
  - Se l'utente è uno studente, il posto viene reso disponibile
  - Se l'utente è un professore, l'aula viene resa disponibile

**Nome:** Segnalazione di "Prenotazione falsa" da parte di un sorvegliante

**Partecipanti:** Sorveglianti, Studenti

**Flusso di eventi:**

- Il sorvegliante apre l'app
- Trova un posto prenotato ma senza studente
- Verifica che il posto non mostri lo studente come "in pausa"
- Il sorvegliante clicca il pulsante "segnala assenza"
- Il sistema invia una notifica di segnalazione allo studente che aveva prenotato
- Il sistema blocca l'utente dalle future di prenotazioni dopo un numero di segnalazioni stabilito in fase di configurazione

**Nome:** Segnalazione di "posto occupato" da parte di uno studente

**Partecipanti:** Sorveglianti, Studenti

**Flusso di eventi:**

- Lo studente trova il suo posto occupato
- Lo studente apre l'app e effettua una segnalazione
- Il sorvegliante riceve una notifica con la segnalazione
- Il sorvegliante va in aula e allontana lo studente

**Nome:** Apertura di un aula

**Partecipanti:** Sorvegliante, Studenti

**Flusso di eventi:**

- Il sorvegliante chiude o apre un'aula
- Il sorvegliante entra nell'applicazione, seleziona l'aula che ha aperto o chiuso e ne cambia lo stato
  - Se l'aula è stata aperta, viene inviata una notifica agli studenti che hanno acconsentito a riceverla

**Nome:** Accesso da parte di un utente

**Partecipanti:** Studenti, Professori, Sorveglianti, Amministratori, Sistemi informatici universitari

**Flussi di eventi:**

- L'utente apre l'app
- L'utente inserisce le credenziali che l'università gli ha fornito
- Il sistema comunica con il server dell'università selezionata dall'utente attraverso apposite API
- Il server dell'università comunica attraverso JSON se il login è andato a buon fine e i dati relativi all'utente, nonché la tipologia di utenza (studente, professore, sorvegliante, amministratore) e un token JWT
- Il sistema logga l'utente in base alla sua tipologia

**Nome:** Logout da parte di un utente

**Partecipanti:** Professori, Studenti, Amministratori, Sorveglianti

**Flusso di eventi:**

- L'utente apre l'app
- L'utente effettua il logout

**Nome:** Aggiornamento della disponibilità dell'aula

**Partecipanti:** Sistemi informatici universitari, Studenti

**Flusso di eventi:**

- Il sistema richiede tramite API al sistema informatico dell'università la disponibilità delle aule
- Il sistema aggiorna il proprio database
- Nel caso in cui un'aula occupata risulti disponibile, il sistema la mostra come disponibile
  - Il sistema invia una notifica agli studenti che hanno acconsentito a riceverla
- Nel caso in cui l'aula risulti non più disponibile, il sistema annulla le prenotazioni di tutti gli studenti che hanno prenotato per lo stesso orario e stessa data.
- Nel caso precedente il sistema invia una notifica agli studenti che avevano prenotato per quell'ora e quella data

**Nome:** Aggiunta/AggIORNAMENTO della mappa degli edifici

**Partecipanti:** Amministratori

**Flusso di eventi:**

- L'amministratore apre il pannello di controllo

- L'amministratore decide se aggiungere o modificare la mappa
  - L'amministratore carica un'immagine della mappa
- L'amministratore clicca il pulsante salva
- Il sistema aggiorna il database

**Nome:** Aggiunta/Aggiorramento delle aule nella mappa da parte dell'amministratore

**Partecipanti:** Amministratori

**Flusso di eventi:**

- L'amministratore apre il pannello di controllo
- L'amministratore seleziona la mappa
- L'amministratore decide se aggiungere o aggiornare un'aula
  - Se l'amministratore decide di aggiungere un'aula, crea attraverso un apposito strumento l'area corrispondente all'aula selezionando i vertici sulla mappa
  - Se l'amministratore decide di modificare un'aula, sposta attraverso un apposito strumento i vertici che definiscono l'area di un'aula
- L'amministratore dà un nome all'aula
- L'amministratore clicca il pulsante salva
- Il sistema aggiorna il database

**Nome:** Configurazione delle aule

**Partecipanti:** Amministratori

**Flusso di eventi:**

- L'amministratore apre il pannello di controllo
- L'amministratore seleziona la mappa
- L'amministratore seleziona l'aula
- Indica in quante file e/o colonne i banchi sono organizzati
- L'amministratore inserisce il numero di banchi presenti per ogni tipologia (banchi da 4, da 5, etc..) e la loro disposizione (verticale o orizzontale).
- Specifica se l'aula di default è aperta o chiusa
- Il sistema costruisce automaticamente una mappa dell'aula utilizzando le informazioni date dall'amministratore
- L'amministratore clicca il pulsante salva
- Il sistema aggiorna il database

**Nome:** Configurazione delle tipologie di banchi

**Partecipanti:** Amministratore

**Flusso di eventi:**

- L'amministratore apre il pannello di controllo
- L'amministratore seleziona la tipologia dei banchi
- L'amministratore configura le varie tipologie di banchi con il numero di posti per ogni tipo
- L'amministratore clicca il pulsante salva
- Il sistema aggiorna il database

**Nome:** Configurazione degli intervalli di tempo

**Partecipanti:** Amministratore

**Flusso di eventi:**

- L'amministratore apre il pannello di controllo
- L'amministratore definisce il tempo massimo di una pausa
- L'amministratore definisce il tempo massimo di ritardo tra l'inizio di una prenotazione e il check-in

- L'amministratore definisce l'antico massimo con il quale una prenotazione può essere effettuata

**Nome:** Configurazione delle priorità nel processo di prenotazione

**Partecipanti:** Amministratore

**Flusso di eventi:**

- L'amministratore sceglie la priorità tra utente e professore
- L'amministratore sceglie il giorno massimo e minimo entro il quale va fatta una prenotazione

**Nome:** Configurazione iniziale

**Partecipanti:** Amministratore

**Flusso di eventi:**

- L'amministratore entra nel sistema. Vengono presentate le diverse fasi di configurazione in quest'ordine:

1. Aggiunta della mappa degli edifici
2. Aggiunta delle aule nella mappa
3. Configurazione delle tipologie di banchi
4. Configurazione delle aule
5. Configurazione priorità nel processo di prenotazione
6. Configurazione degli intervalli di tempo

## 3.4 | Excluded Requirements

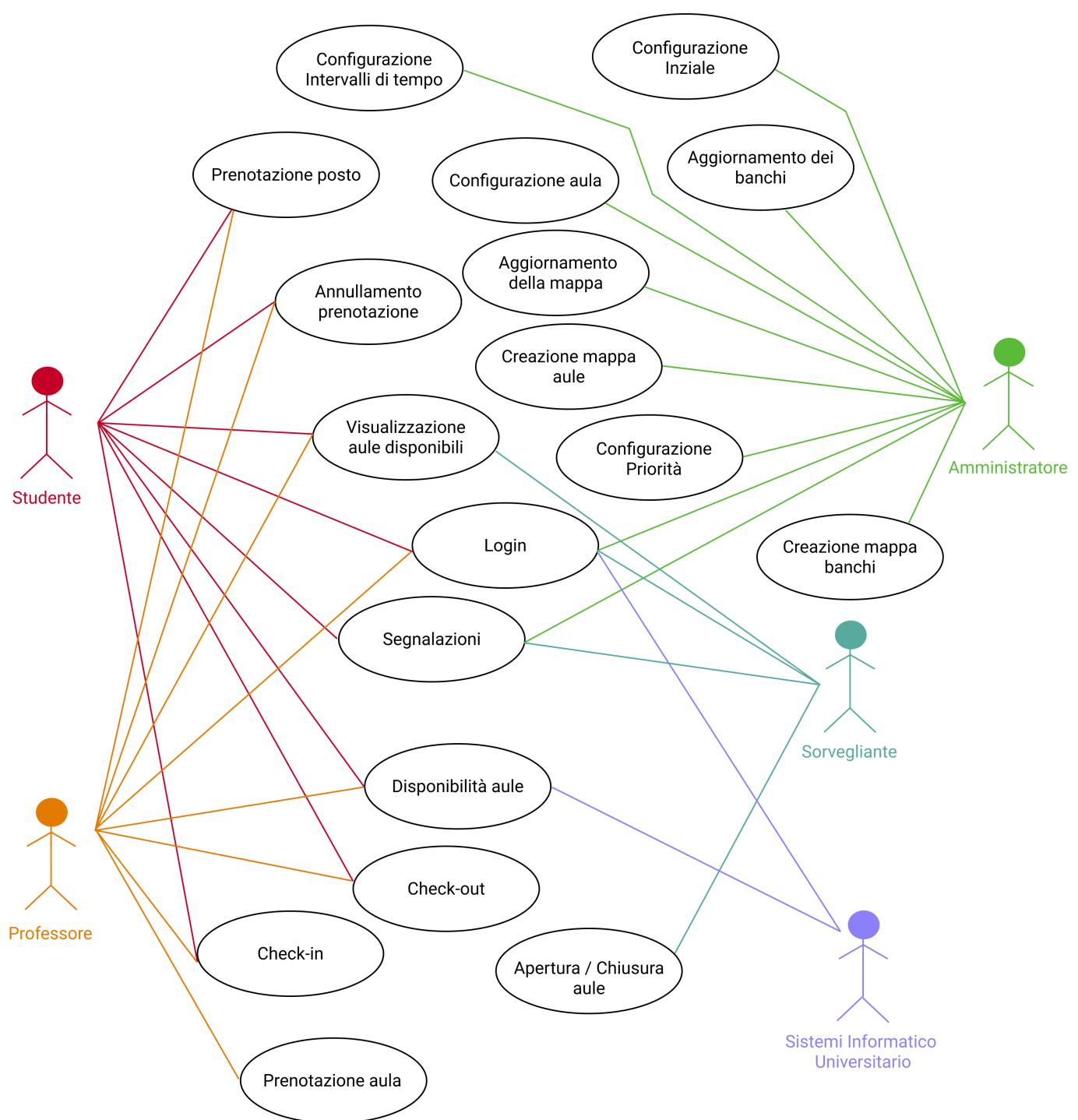
- **Utilizzo del RFID per il check-in:** il servizio avrebbe richiesto l'utilizzo di hardware apposito. Inoltre sarebbe nato il problema della distribuzione e della gestione dei tag. Tag fisici di grande dimensioni (es. card) avrebbero generato problemi di acquisto, gestione e smarrimento. Tag adesivi, sebbene più comodi, oltre al costo iniziale di acquisto avrebbero generato un costo di manutenzione.
- **Utilizzo codice QR per il check-in:** In assenza di un sistema di controllo della presenza nel banco, il sistema avrebbe permesso di fare una screenshot al proprio codice QR che si sarebbe potuta inviare ad un'altra persona per falsificare il check-in. Inoltre abbiamo ritenuto che ad un costo molto simile sia una migliore scelta l'acquisto di BLE, i quali oltre al check-in permettono di fare il controllore della presenza in aula.
- **Creazione di gruppi di studio:** il servizio sebbene molto utile, avrebbe complicato inutilmente il sistema dovendo poi aggiungere una gestione delle prenotazioni interne ai gruppi e di approvazioni delle prenotazioni fatte a nome di tutto il gruppo. La proposta della creazione di gruppi di studio, è stata fatta da un solo studente in fase di intervista, quindi abbiamo ritenuto non fosse considerata come feature fondamentale da tutta l'utenza. Inoltre abbiamo ritenuto che una soluzione alternativa sia dare la possibilità agli utenti di organizzarsi per prenotare posti tra loro vicini.
- **Utilizzo del Wi-Fi universitario per il check-in e controllo della presenza:** il servizio avrebbe negato l'utilizzo del sistema agli utenti in diversi casi. Il Wi-Fi non può essere considerato sempre funzionante, e non tutti gli studenti sono disposti a utilizzare il Wi-Fi universitario, perché meno performante rispetto ad un hotspot personale o per evitare distrazioni durante lo studio. Inoltre abbiamo pensato che un sistema di check-in del genere sarebbe stato fortemente dipendente dall'architettura di rete della rete universitaria. Nell'interesse di sviluppare un sistema portatile e indipendente dalla singola università, la soluzione è stata scartata.
- **Visualizzazione dei posti disponibili in università come lista di aule:** nel sondaggio condiviso con gli stakeholder, gli utenti hanno indicato di non preferire questa opzione

- **Segnalazione da parte di uno studente a una postazione che non sia la sua:**  
Inizialmente avevamo pensato di non creare un meccanismo automatico per il controllo delle postazioni occupate a vuoto, e lasciare agli studenti la possibilità di effettuare segnalazioni alle altre postazioni. Tuttavia non si poteva essere certi che gli studenti si sarebbero segnalati tra di loro e il servizio avrebbe potuto generare troppe segnalazioni delle quali non si poteva verificare l'autenticità. Un abuso del sistema di segnalazione avrebbe solamente disturbato il lavoro dei sorveglianti. Avrebbe richiesto inoltre l'impiego di troppo personale di controllo.
- **Caricamento di una mappa che rappresenti in modo realistico i posti dentro un'aula:**  
la funzionalità avrebbe richiesto maggiore lavoro in fase di configurazione rispetto ad una mappa dei posti generata. Prenotare i singoli posti scegliendone la posizione è una funzionalità che è stata aggiunta per permettere alle persone che vogliono studiare insieme di poter prenotare posti affini. Pertanto non è importante sapere dove dei posti sono posizionati nell'aula, è importante solamente sapere se si trovano vicini e allo stesso tavolo. Per raggiungere questa funzionalità, l'inserimento manuale di una mappa dei posti dell'aula che rispecchi le posizioni dell'aula avrebbe presentato alcune problematiche:
  - La mappa andrebbe modificata ogni volta che i banchi vengono spostati, anche se rimangono dello stesso tipo
  - Se viene aggiunto un nuovo banco, invece di aggiungere un banco dal pannello di controllo bisognerebbe ricaricare una nuova mappa dell'aula
  - Per gestire il numero di aule richieste dalla documentazione (circa 1000) ogni università avrebbe dovuto creare una planimetria di 1000 aule, una soluzione inutilmente dispendiosa rispetto ai vantaggi che avrebbe offerto rispetto a una mappa auto generata.

## 3.5 | Assunzioni

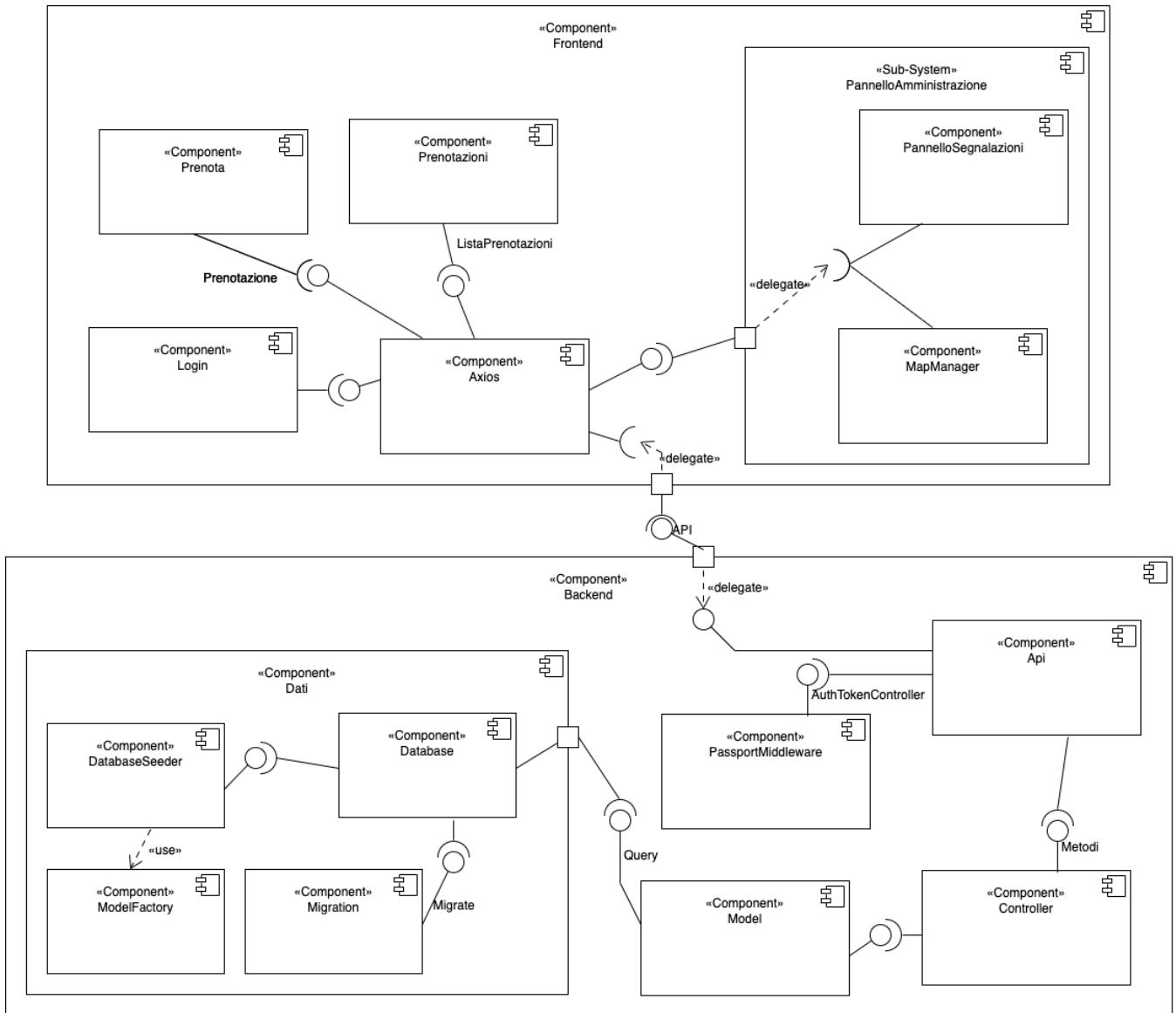
- L'università deve fornire delle API per ottenere informazioni sugli utenti e delle API per ottenere le prenotazioni delle aule per le lezioni
- E' necessario almeno un sorvegliante per edificio che si occupi delle mansioni di apertura e chiusura delle aule, e che gestisca le segnalazioni eventualmente potendo allontanare studenti che infrangono le regole
- Ogni aula che si vuole aggiungere al servizio necessita di un beacon bluetooth.
- Assumiamo che lo studente/professore/sorvegliante abbia un telefono funzionante durante tutta la durata della prenotazione
- Assumiamo che ogni utente che voglia utilizzare il servizio sia presente nel database dell'università.
- Assumiamo che il sorvegliante sia reperibile durante le segnalazioni
- Assumiamo che ci sia una persona addetta alla costruzione delle mappe in fase di configurazione iniziale (spiegata negli scenari) e che le costruisca correttamente
- Assumiamo che ogni studente abbia a disposizione un telefono

## 3.6 | Use Case Diagram



# 4 | Software Architecture

## 4.1 | Component Diagram



**Axios:** Componente che si occupa delle richieste alle API nel frontend

**Login:** Pannello di login

**Prenota:** Pannello di visualizzazione dell'aula, che permette la prenotazione di un posto

**Prenotazioni:** Pannello per visualizzare la lista di prenotazioni dell'utente ed eliminarne una

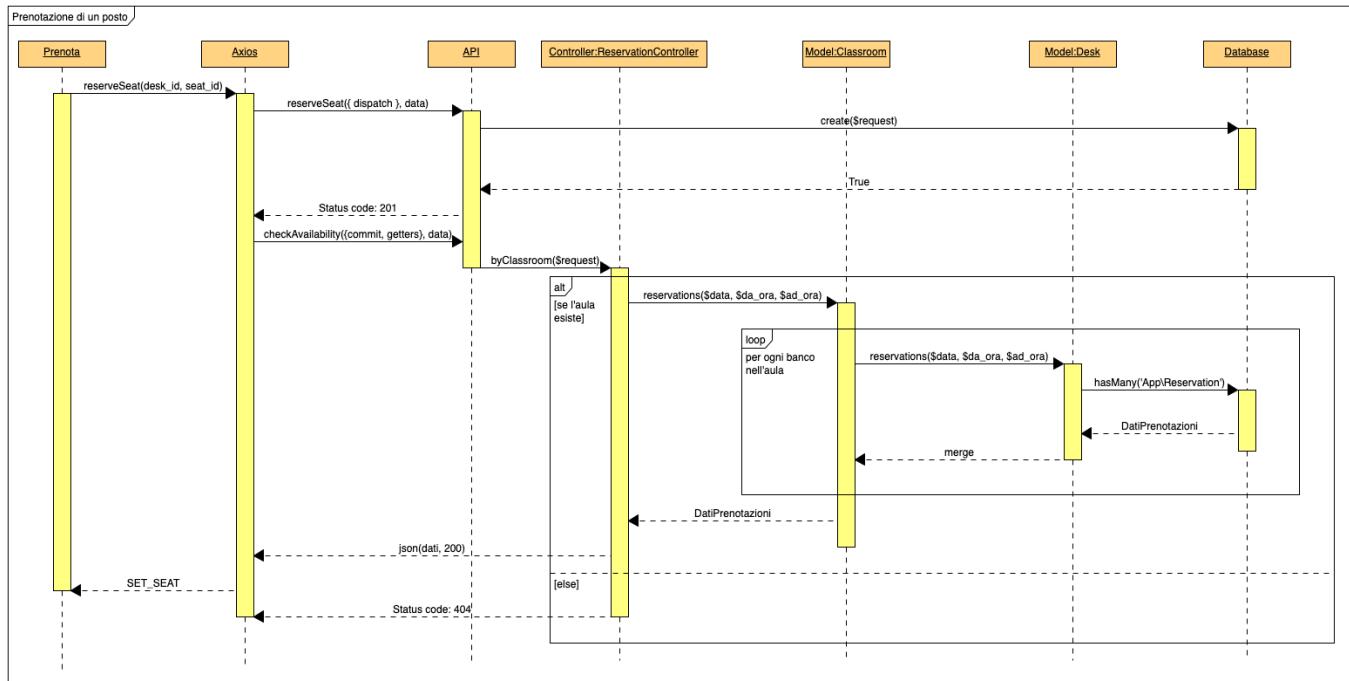
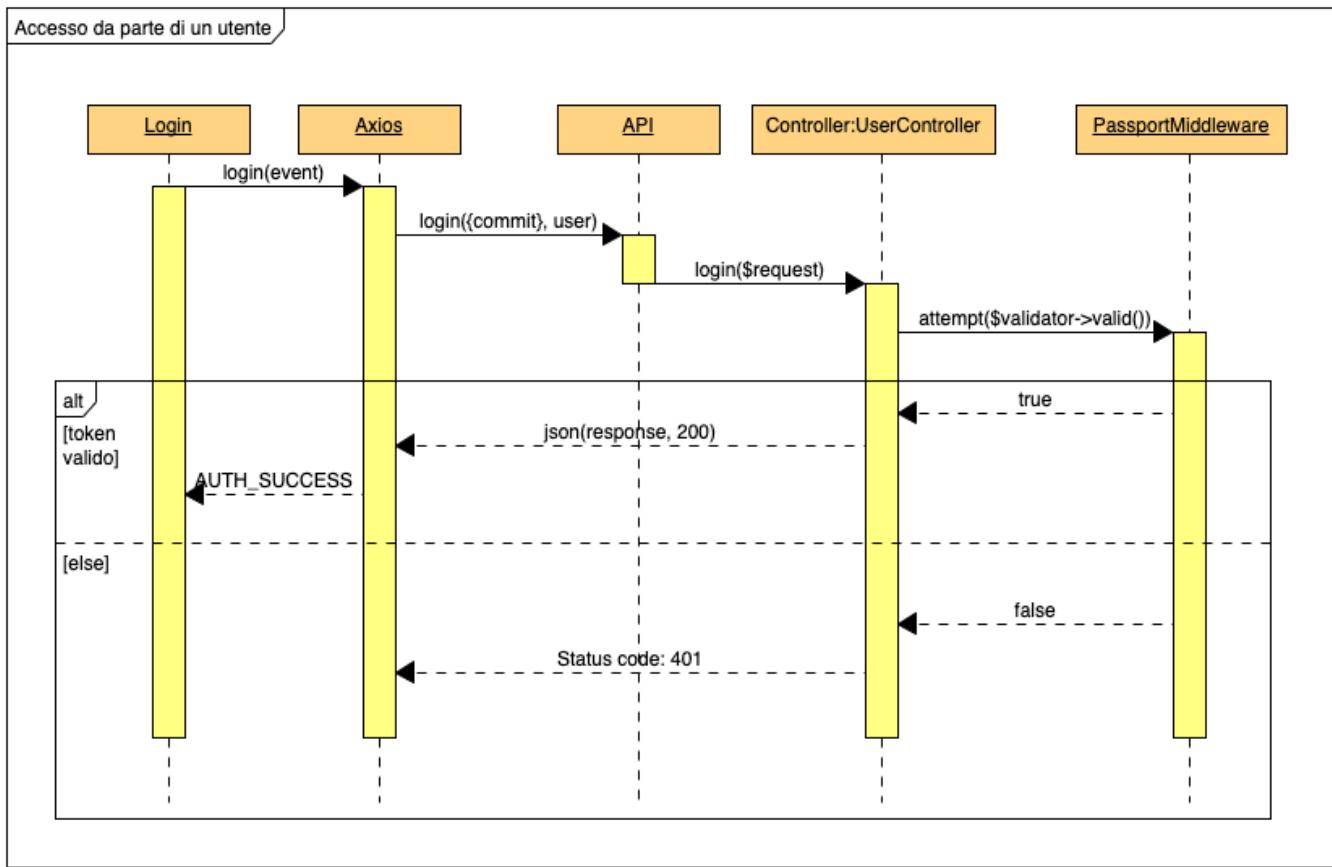
**PassportMiddleware:** Componente che si occupa della gestione del token di autenticazione per l'utente. Esegue i controlli nelle chiamate ad API protette.

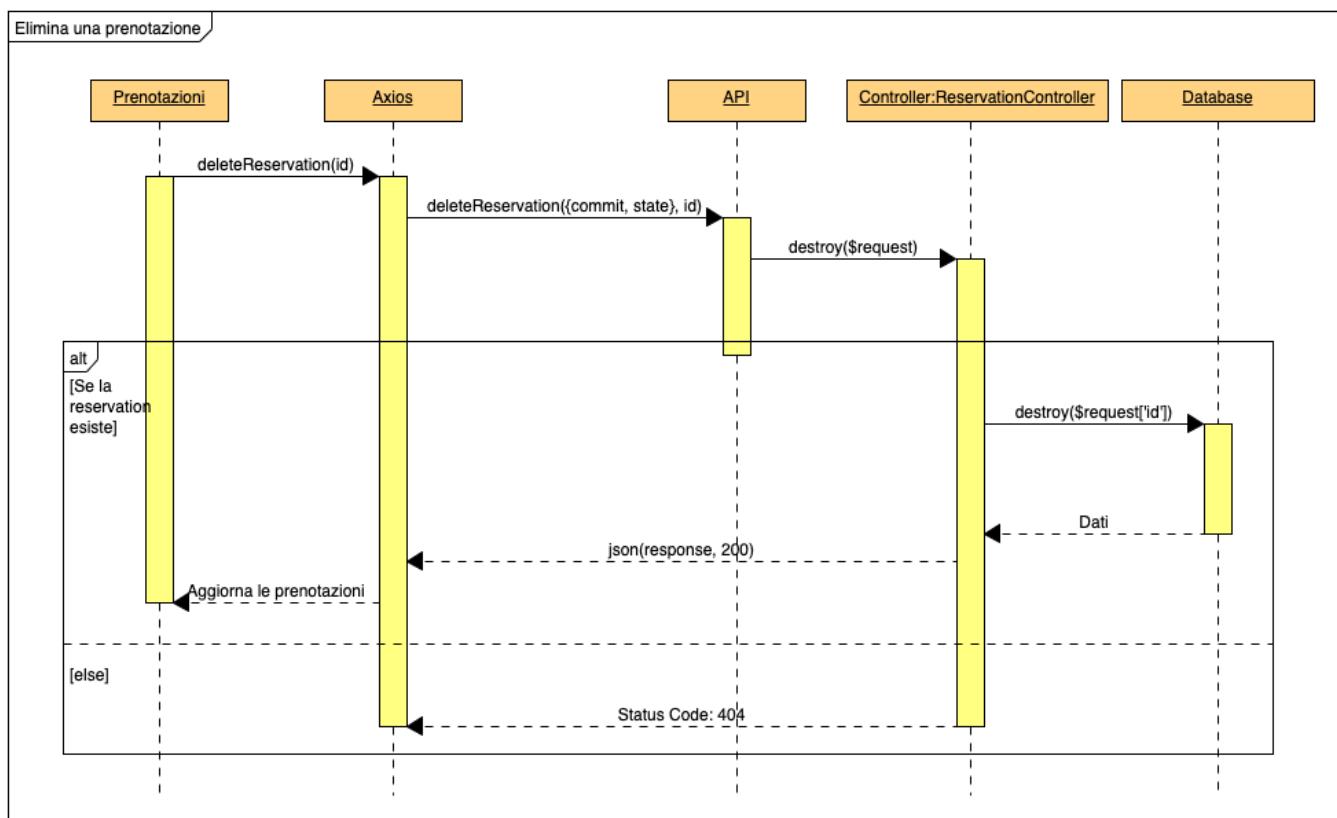
**DatabaseSeeder:** Componente che si occupa di caricare i dati di test nel database

**ModelFactory:** Componente che si occupa di generare dati fintizi per ogni modello

**Migration:** Componente che si occupa della definizione delle tabelle del Database

## 4.2 | Sequence Diagram





## 5 | Data and Data Models

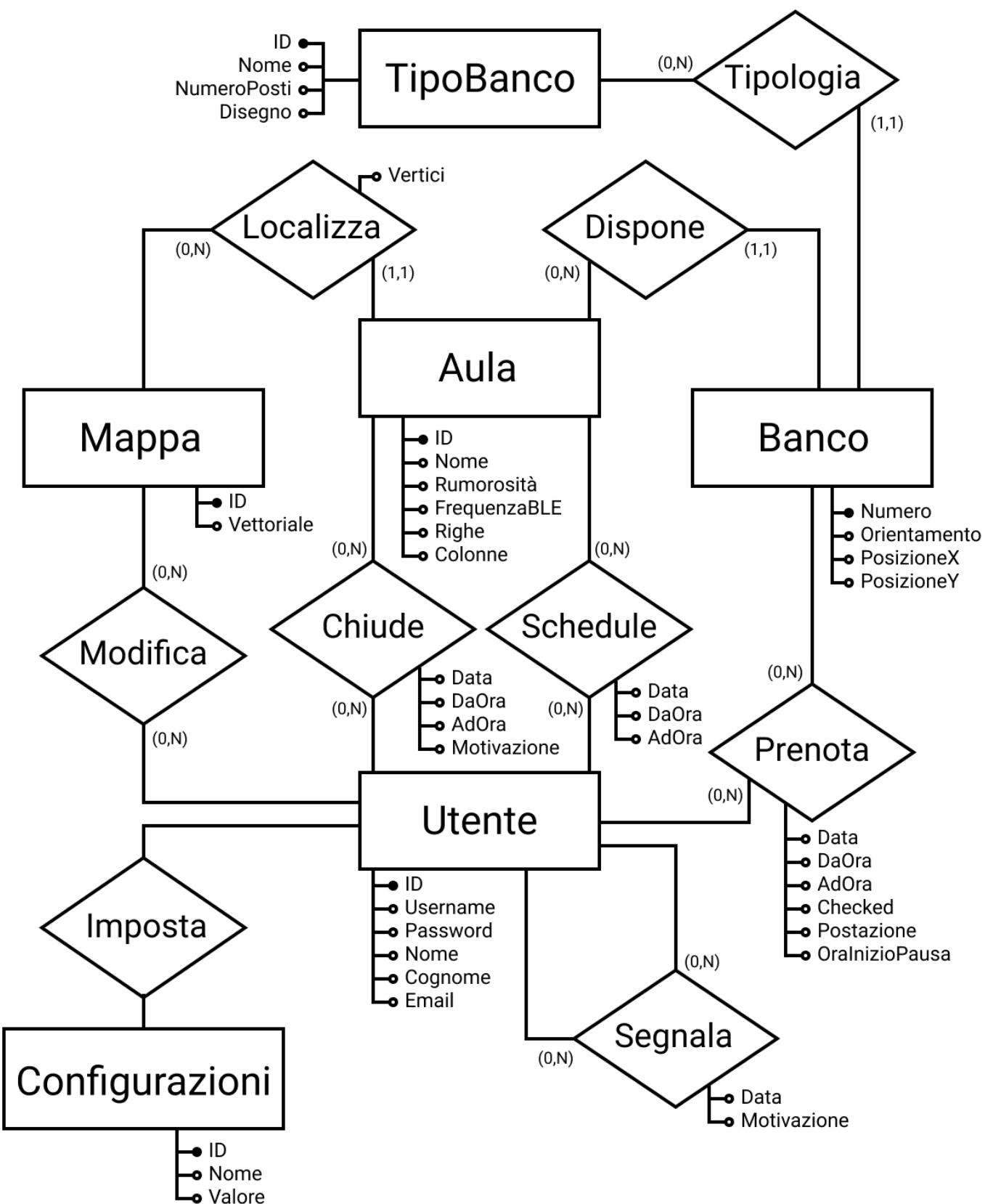
I dati necessari al sistema sono:

- Planimetrie dell'università
- Orario delle lezioni
- Planimetrie delle aule
- Lista della tipologia di banchi utilizzati dall'università
- Informazioni personali sugli utenti

Il DB dell'università ci può fornire solo l'orario delle lezioni e le informazioni personali degli utenti. Quindi il resto dei dati deve essere inserito manualmente nel sistema. Tuttavia i dati relativi alle planimetrie, sono generalmente statici, ovvero soggetti raramente a cambiamenti nel tempo. Questo vuol dire che sebbene il nostro sistema richieda un data entry manuale, i dati richiesti dovranno essere inseriti solo in fase di configurazione o in raro caso di modifica.

### 5.1.2 | Schema ER

- Le tabelle di prenotazione tra aula (schedule) e posti (prenota) sono separate per migliorare le performance quando si necessita l'accesso alle prenotazioni delle sole aule
- Le aule hanno uno stato di default. Alcune aperte e altre chiuse, i custodi poi giorno per giorno possono modificare lo stato. Inoltre ogni aula ha un orario di default di apertura e uno di chiusura
- L'entità utente non dispone di un flag "tipo" poiché l'identity manager salva temporaneamente le sue informazioni al momento del login e le associa ad un token che l'utente utilizzerà per effettuare operazioni privilegiate.
- I dati presi dall'API per le lezioni, vengono inseriti nella tabella schedule



## 6 | Design Decisions

- Il sistema non si occupa degli orari delle lezioni perchè le policy di prenotazione delle aule per le lezioni possono essere molto diverse da università a università. Utilizzare delle API permette di astrarre il sistema dalle policy di ogni università. Inoltre, nella specifica è richiesto che il sistema sia indirizzato alla prenotazione e all'uso di postazioni per studiare. Pertanto riteniamo che implementare un backend o frontend per gli orari di lezioni, esca dal dominio del sistema.
- Si permette di prenotare i singoli posti perchè così è possibile riconoscere in modo semplice se una persona sta occupando una postazione senza prenotazione o in modo abusivo. Inoltre prenotare un posto preciso permette alle persone che vogliono studiare insieme di prenotare posti che sono vicini tra di loro. La complessità del sistema aumenta, tuttavia questo meccanismo permette alle persone che vogliono studiare in gruppo di prenotare posti vicini tra di loro, soddisfacendo così la richiesta da parte degli studenti di avere un modo per poter studiare in gruppo. Inoltre, il costo hardware è irrisorio rispetto ai vantaggi appena nominati. Sarà infatti necessario solamente munire ogni banco di un numero identificativo.

### **Configurazione dell'aula:**

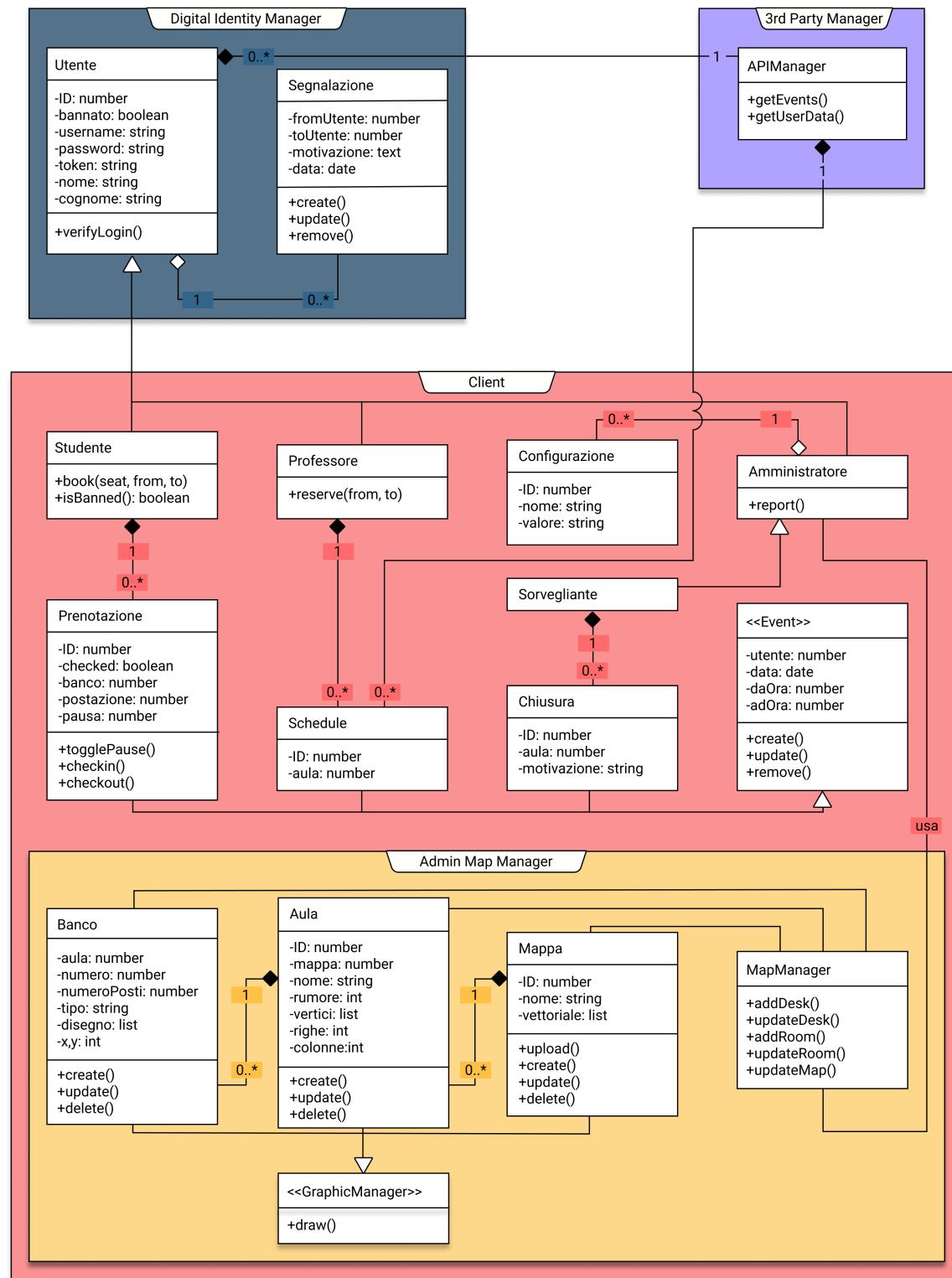
Abbiamo notato che le aule solitamente utilizzano un set di banchi che si può ritenere standard per l'università (banchi da 4, banchi da 6, fila da 10, fila da 20, ecc..). Questo ci ha portato a pensare che non c'è bisogno di una planimetria precisa della disposizione dei posti per ogni aula. Per poter permettere allo studente di selezionare un posto, è sufficiente una mappa che mostri come i vari posti sono divisi tra i tavoli. In questo modo lo studente avrà la possibilità di prenotare posti affianco alle persone con le quali vuole studiare, perchè la mappa sarà sufficientemente dettagliata da mostrargli in che banco si trova, senza però la necessità di dover inserire una vera planimetria nel sistema per mostrargli dove il banco si trova nell'aula. Questo sistema di configurazione dei posti, permette inoltre di modificare velocemente una planimetria di un'aula, aggiungendo semplicemente un nuovo tipo di banco, senza la necessità di inserire nel sistema una planimetria completamente nuova.

La mappa viene realizzata dal sistema costruendo una matrice che rappresenta l'aula (colonne e righe sono specificate in fase di configurazione). I tipi di banco sono successivamente collocati all'interno delle celle della matrice, specificando qual è il tipo di banco e il suo orientamento. A questo punto il sistema può generare una mappa sufficientemente accurata. Graficamente il sistema funzionerà come mostrato nello stato dell'arte della piattaforma Optix.

**Utilizzo dei BLE per il check-in e controllo delle postazioni:** Tra tutte le soluzioni analizzate, l'utilizzo dei BLE ci è sembrata quella con i maggiori vantaggi rispetto ai costi. Il loro costo si aggira intorno alla cifra di uno scanner di codice QR, ma con i BLE oltre che del check-in possiamo occuparci del controllo della postazione. A seguito di un incontro con il committente abbiamo ragionato sui possibili problemi di piazzamento per orientare al meglio il segnale dei BLE, problemi che abbiamo supposto risolvibili con l'utilizzo di multipli BLE per aula in caso di necessità. Le motivazioni per le quali abbiamo ritenuto che l'utilizzo del Wi-Fi non sia ideale sono state precise negli excluded requirements.

# 7 | Low Level Design

## 7.1 | Class Diagram



## 7.2 | API Documentation

Le API sono state realizzate seguendo lo standard [OpenAPI 3.0](#).

Possono essere consultate a:

<https://documenter.getpostman.com/view/5342440/SWTD8x88?version=latest>

# 8 | Explain how the FRs and the NFR are satisfied by design

## 8.2 | Nonfunctional Requirements

### Usability

**User error protection:** il sistema utilizza sistemi informatici esterni per fornire dati sugli utenti, pertanto non è possibile avere dei dati errati sugli utenti nel sistema. Gli unici dati che possono essere inseriti errati sono quelli sulla mappa dell'università e le segnalazioni.

**Learnability:** il sistema per gli utente è utilizzabile tramite un sistema di "mappe". E' pertanto molto intuitivo da utilizzare. La costruzione delle mappe avviene inserendo immagini, o specificando di che tipo sono i banchi presenti in aula. Le prenotazioni vengono fatte tramite un sistema di click della postazione. La curva di apprendimento del sistema quindi è molto ripida.

### Security

**Confidentiality:** Il sistema gestisce l'accesso alla visualizzazione delle prenotazioni solo al personale autorizzato

**Accountability:** il sistema prende i dati dal sistema informatico dell'università, la quale verifica le identità digitali utilizzando un documento di riconoscimento. L'identità si può quindi considerare univoca per ogni entità.

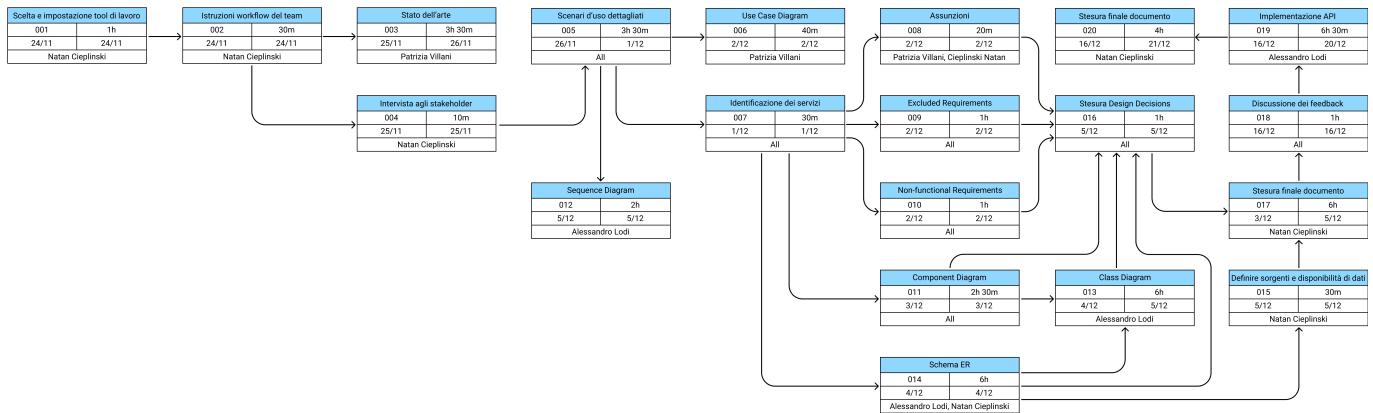
### Maintainability

**Reusability:** il sistema può essere facilmente esteso per diventare un sistema di prenotazione delle aula anche per le lezioni ordinarie, e per registrare le presenze nelle lezioni a frequenza obbligatoria.

### Portability

**Adaptability:** il sistema è pensato per essere utilizzato da più università tramite delle API. Non si è scelto di usare il sistema per le lezioni perchè ogni università può avere la propria policy di prenotazione delle aule.

# 9 | Effort Recording



## Natan Work Log

Date gg/mm	Time Spent	Description of the task	Category	Subcategory
24/11	1h	Impostazione dei tool di lavoro: repo su GitHub, board su Trello	Doing	Work Organization
24/11	30m	Stesura delle istruzioni sul workflow del team. Identificazione delle macro categorie di task	Doing	Work Organization
25/11	10m	Creazione e diffusione di un questionario per raccogliere opinioni da possibili utenti	Doing	Raffinamento dei requisiti
30/11	2h 30m	Use case diagrams, class diagrams. Overview of UML	Learning	UML
1/12	2h 30m	Functional Requirements, Nonfunctional Requirements, Use case, Scenarios, Actors	Learning	Functional & Non functional Requirements
4/12	7h	Stesura, impaginazione e raffinamento del documento finale	Doing	Documentazione
5/12	1h	Stesura PERT	Doing	Documentazione
16/12	30 m	Integrazione stato dell'arte come richiesto nei feedback	Doing	Stato dell'arte
16/12	1h	Correzioni della documentazione in base ai feedback (FR and Interview to Stakeholders)	Doing	Documentazione
18/12	30m	Correzioni della documentazione in base ai feedback (Scenarios, Excluded Requirement)	Doing	Documentazione
19/12	1h	Correzioni della documentazione in base ai feedback (Scenarios, Excluded Requirement)	Doing	Documentazione
21/12	1h	Revisione finale documentazione	Doing	Documentazione
30/12	4h	Frontend prototipo	Doing	Implementazione prototipo
31/12	1h	Aggiornamento component e sequence diagram, stesura finale documentazione	Doing	Documentazione

## Alessandro Work Log

Date gg/mm	Time Spent	Description of the task	Category	Subcategory
4/12	3h 30m	Creazione dello schema ER	Doing	Dati modellazione
4/12	3h	Creazione del Class Diagram	Doing	Design di basso livello
5/12	3h	Creazione del Class Diagram	Doing	Design di basso livello
5/12	2h	Creazione del Sequence Diagram	Doing	Architettura software
16/12	1h	API Design	Learning	Implementazione prototipo
16/12	1h	Definire le risorse delle API	Doing	Implementazione prototipo
20/12	2h	Definire le URL delle API	Doing	Implementazione prototipo
20/12	2h 30m	Creazione della documentazione delle API con OpenAPI 3.0	Doing	Implementazione prototipo
30/12	4h	Creazione modulo autenticazione	Doing	Implementazione prototipo
31/12	16h	Implementazione backend e fine frontend	Doing	Implementazione prototipo

## Patrizia Work Log

Date gg/mm	Time Spent	Description of the task	Category	Subcategory
26/11	3h 30m	Ricerca e analisi dello stato dell'arte e stesura di un mini report	Doing	Stato dell'arte
1/12	2h 30m	Use case diagrams, class diagrams. Overview of UML	Learning	UML
2/12	40m	Stesura use case diagrams	Doing	Use case Diagram
2/12	20m	Stesura delle assunzioni	Doing	Raffinamento dei requisiti
3/12	30m	Stesura del Component Diagram	Doing	Component Diagram
5/12	3h	Trasferimento del Use case diagram e Component diagram da draw.io a figma	Doing	UML
19/12	1h 30m	API Design	Learning	API
20/12	1h 30m	Controllare se le università hanno un calendario di lezioni online	Doing	Raffinamento dei requisiti
21/12	1h	Revisione delle API	Doing	Implementazione prototipo

## Team Work Log

Date gg/mm	Total Time	Description of the task	List of people who worked on the task	Category	Subcategory
23	11 30m	Briefing organizzativo	Patrizia Villani, Natan Cieplinski	Doing	Work Organization
26	11 2h	Discussione degli scenari d'uso e chiarimenti da chiedere al committente	Patrizia Villani, Natan Cieplinski, Alessandro Lodi	Doing	Raffinamento dei requisiti
1	12 1h 30m	Definizione degli scenari d'uso dettagliati	Patrizia Villani, Natan Cieplinski, Alessandro Lodi	Doing	Raffinamento dei requisiti
1	12 30m	Identificazione dei servizi	Patrizia Villani, Natan Cieplinski, Alessandro Lodi	Doing	Raffinamento dei requisiti
2	12 2h	Discussione sugli Excluded Requirements, Non functional Requirement	Patrizia Villani, Natan Cieplinski, Alessandro Lodi	Doing	Raffinamento dei requisiti
3	12 2h	Discussione sul Component Diagram	Patrizia Villani, Natan Cieplinski, Alessandro Lodi	Doing	Architettura del Software
4	12 2h 30m	Discussione sullo schema ER	Natan Cieplinski, Alessandro Lodi	Doing	Dati e Modellazione
16	12 1h	Discussione feedback e rivalutazione requisiti funzionali	Patrizia Villani, Natan Cieplinski, Alessandro Lodi	Doing	Raffinamento dei requisiti
19	12 1h	Rivisitazione schemi secondo i requisiti aggiornati	Natan Cieplinski, Alessandro Lodi	Doing	Documentazione

## Work Log:

Total hours spent by the group: 111 hours 10 minutes

Time spent learning: 8 hours 30 minutes

Time spent doing: 102 hours 40 minutes