

# Contents

<b>1</b>	<b>Descrizione e Analisi del progetto</b>	<b>3</b>
1.1	Obiettivo . . . . .	3
1.2	Analisi del problema . . . . .	3
<b>2</b>	<b>Struttura del progetto</b>	<b>3</b>
<b>3</b>	<b>Progettazione concettuale</b>	<b>4</b>
3.1	Un po' di teoria . . . . .	4
3.2	Class diagram . . . . .	5
3.3	Class diagram completo . . . . .	6
<b>4</b>	<b>CRC Cards</b>	<b>6</b>
4.1	CRC cards UTILITIES . . . . .	7
4.1.1	Controller . . . . .	7
4.1.2	DBConnection . . . . .	7
4.1.3	EmailSender . . . . .	8
4.2	CRC cards DAO e DTO . . . . .	9
4.2.1	LaboratorioDAO e Laboratorio . . . . .	9
4.2.2	PostazioneDAO e Postazione . . . . .	10
4.2.3	PrenotazioneDAO e Prenotazione . . . . .	11
4.2.4	ResponsabileDAO e Responsabile . . . . .	12
4.2.5	SedeDAO e Sede . . . . .	13
4.2.6	StrumentoDAO e Strumento . . . . .	14
4.2.7	TeamDAO e Team . . . . .	15
4.2.8	TecnicoDAO e Tecnico . . . . .	16
4.2.9	UtenteDAO e Utente . . . . .	17
4.3	CRC cards GUI . . . . .	18
4.3.1	Application . . . . .	18
4.3.2	MainWindow . . . . .	18
4.3.3	NewBookingToolSelected . . . . .	19
4.3.4	IncreaseFont . . . . .	19
4.3.5	CalendarDialog . . . . .	20
4.3.6	UserHomePage . . . . .	20
4.3.7	NoScalingIcon . . . . .	21
4.3.8	ManagerHomePage . . . . .	21
4.3.9	LinkMouseOn . . . . .	22
4.3.10	ModOrganigramma . . . . .	22
4.3.11	MyBooking . . . . .	23
4.3.12	BookingSelectionListener . . . . .	23
4.3.13	MyTeam . . . . .	24
4.3.14	NewBooking . . . . .	24
4.3.15	SignInPageManager . . . . .	25
4.3.16	NewTeam . . . . .	25
4.3.17	OrganigramFiveTechnicians . . . . .	26
4.3.18	PasswordRecovery . . . . .	26

4.3.19	OrganigramTenTechnicians	27
4.3.20	PasswordRecoveryManager	27
4.3.21	UserLoginPage	28
4.3.22	ManagerLoginPage	28
4.3.23	SignInPage	29
4.3.24	ToolBookingSummary	29
4.3.25	MyBookingsTableModel	29
4.3.26	ToolsBookingsTableModel	30
4.3.27	ToolsTableModel	30
4.3.28	TableModelTeam	31
4.3.29	TablePanel	31
4.3.30	TextFieldBoderColor	31
<b>5</b>	<b>Sequence Diagram</b>	<b>32</b>
5.1	Sequence diagram della prima funzionalità a scelta	32
5.2	Sequence diagram della seconda funzionalità a scelta	33
<b>6</b>	<b>Piccolo esempio</b>	<b>34</b>
6.1	Login Utente	34
6.2	Registrazione Responsabile	36

# Documentazione

February 21, 2024

## 1 Descrizione e Analisi del progetto

### 1.1 Obiettivo

Ci è stato chiesto di sviluppare un sistema informativo composto da un applicativo Java dotato di GUI Swing per la gestione di un laboratorio scientifico.

### 1.2 Analisi del problema

E' stato sviluppato un sistema (**Lab-T**) utilizzabile da utenti e responsabili che, dopo un' opportuna registrazione o un semplice login, potranno visualizzare e utilizzare le rispettive funzionalità.

*La registrazione* è effettuabile tramite l'inserimento dei dati del tipo degli attributi dell'entità che vuole registrarsi, che sono diversi per l'utente e il responsabile <sup>1</sup>.

*Login* permette l'accesso al sistema tramite Email e password per l'utente, e tramite matricola e password per il responsabile. In questa pagina ci sarà la possibilità di registrarsi in caso di utente/responsabile nuovo, la possibilità di recuperare le credenziali o cambiare password.

*L'utente* come da traccia assegnata, avrà la possibilità di selezionare lo strumento per descrizione o per sede, e vederne il calendario di prenotazione, può effettuare una prenotazione per lo strumento selezionato e cancellarla o, eventualmente, modificarla. Inoltre per ogni strumento potrà visualizzare un riepilogo, su base mensile ed annuale, di quanto è stato usato e di quale utente lo ha usato di più.

*Il responsabile* può organizzare i tecnici in team, può creare, modificare e eliminare team di lavoro, nonché visualizzare un organigramma dei team a lui associati.

## 2 Struttura del progetto

Il nostro progetto è stato realizzato seguendo la logica del design pattern "**Model MVC**", in cui c'è una separazione delle responsabilità tra il modello (rappresentazione dei dati che nel nostro caso prende il nome di **DTO**, che sta per "Data Transfer Objects", e sono oggetti utilizzati per trasferire dati tra i vari componenti dell'applicazione, spesso tra il livello di presentazione e il livello di logica di business. Le DTO contengono solamente campi dati e i metodi getters e setters), la vista (rappresentazione dell'interfaccia utente, che nel nostro caso prende il nome di **GUI**) e il **controller** (gestione degli input utente e aggiornamento del modello) e seguendo anche il pattern **DAO** che sta

---

<sup>1</sup>Nelle pagine successive parleremo degli attributi delle entità individuate

per "Data Access Object". Il pattern DAO è un design pattern che fornisce un'interfaccia astratta per alcuni tipo di database. L'obiettivo principale di questo pattern è separare la logica di accesso ai dati dalla logica di business o di presentazione. Da quello detto fin'ora quindi si può capire che al nostro progetto è collegato un database locale implementato tramite il DBMS PostgreSQL e collegato tramite JDBC.

Il nostro progetto è composto da 4 package <sup>2</sup>:

1. GUI,
2. DTO,
3. DAO,
4. UTILITIES: in cui è presente il controller e altre classi di supporto, di cui parleremo in seguito

Prima di descrivere le classi presenti nei vari package, dobbiamo capire quali sono le entità coinvolte nel nostro progetto e le relazioni che intercorrono tra esse.

## 3 Progettazione concettuale

### 3.1 Un po' di teoria

Il modello utilizzato per descrivere le entità è l' **Unified Modeling Language (UML)** che è un linguaggio grafico per la modellazione di applicazioni software basate sulla programmazione ad oggetti. Il modello UML descrive i dati con questi concetti:

- **Classi**: una classe rappresenta la descrizione di un insieme di oggetti individuati da comportamenti e proprietà comuni.
- **Attributi**: sono proprietà definite da una classe e possiamo vedere che hanno un nome e un tipo
- **Metodi**: sono i comportamenti, le funzionalità di una classe
- **Associazioni**: esprimono una connessione logica tra classi. È possibile associare un nome ad un'associazione, specificare il ruolo che una certa classe ricopre nell'associazione e hanno una cardinalità.

---

<sup>2</sup>Un package è un meccanismo per organizzare classi Java logicamente correlate

## 3.2 Class diagram

Ecco le classi, e le relative associazioni, individuate da noi.

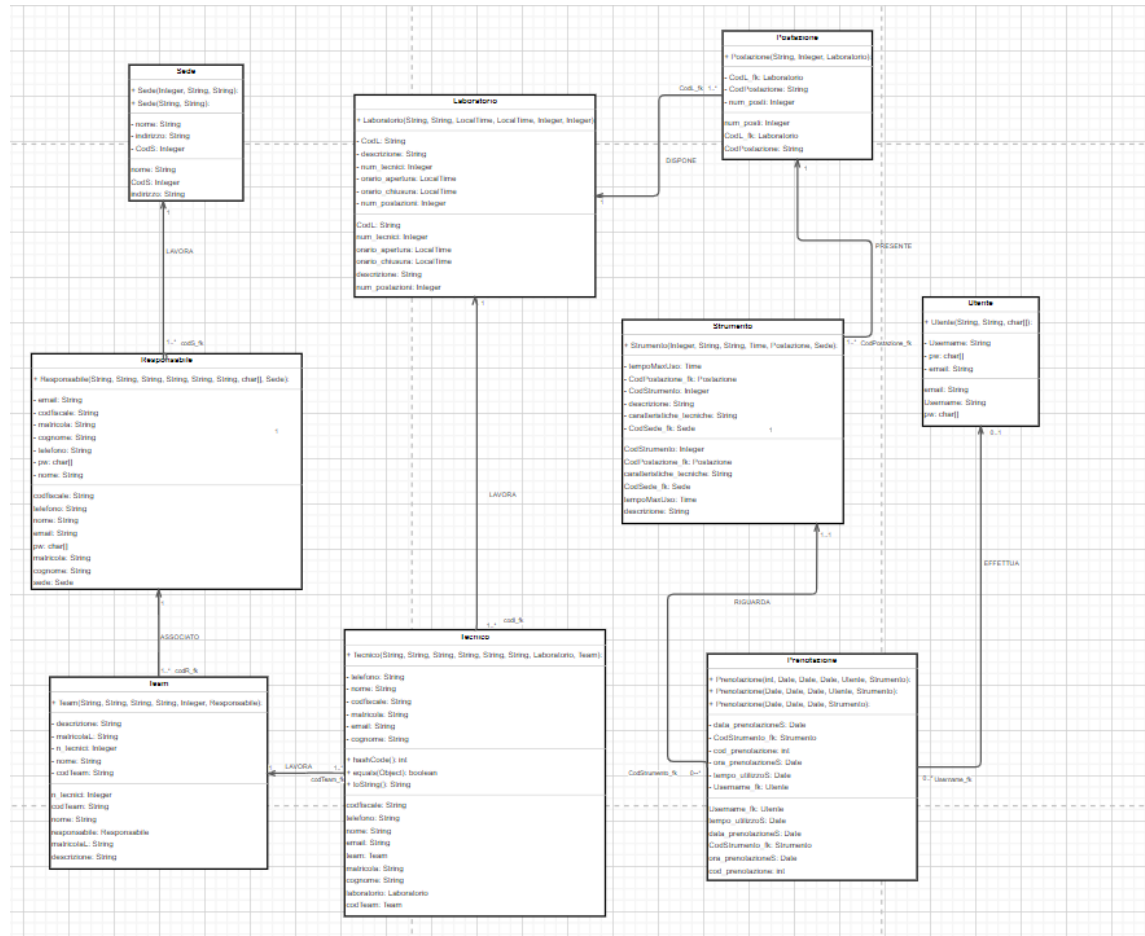


Figure 1: Class diagram entità

### 3.3 Class diagram completo

A seguire il class diagram di tutte le classi del nostro progetto, ovvero delle classi presenti nei packages GUI, DAO, DTO e UTILITIES

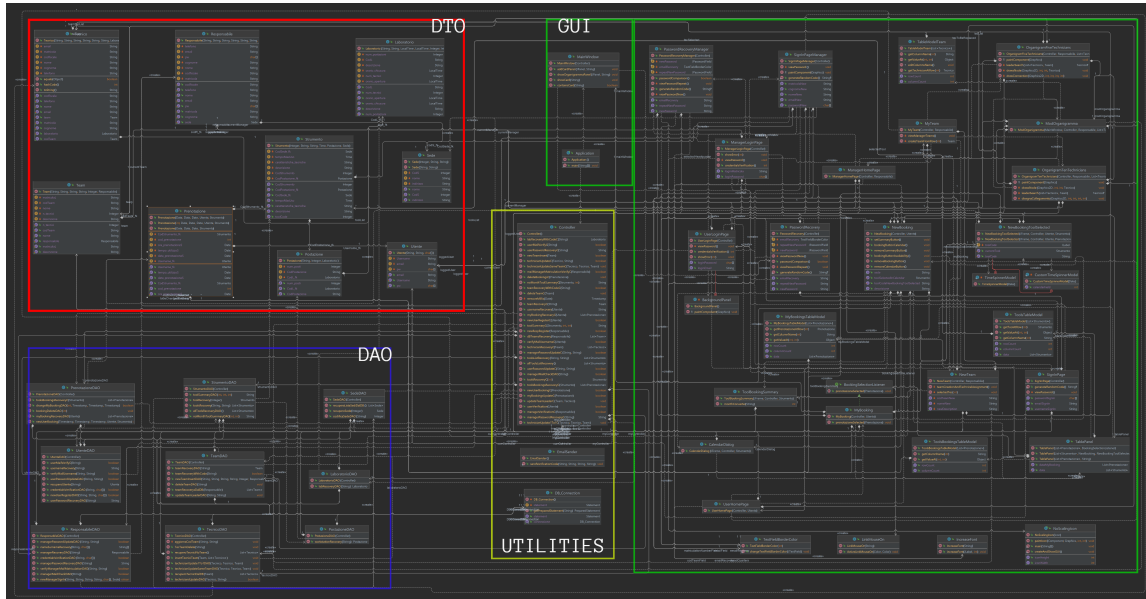


Figure 2: Class diagram completo

## 4 CRC Cards

Le CRC (**Class-Responsibility-Collaboration**) cards sono uno strumento utilizzato nel contesto della progettazione orientata agli oggetti per catturare, organizzare e discutere le informazioni relative alla progettazione delle classi e delle loro interazioni. Come dice il nome danno un focus su Responsabilità e Collaborazione, ovvero sulle responsabilità di ciascuna classe e sulle sue collaborazioni con altre classi. Di seguito troveremo le CRC CARDS di tutte le classi.

## 4.1 CRC cards UTILITIES

### 4.1.1 Controller

Class: <b>CONTROLLER</b>	
RESPONSIBILITIES	COLLABORATORS
<ol style="list-style-type: none"><li>1. Gestire le interazioni con le classi DAO (Data Access Object) come <b>UtenteDAO</b>, <b>StrumentoDAO</b>, <b>PrenotazioneDAO</b>, ecc.</li><li>2. Fornire metodi per l'accesso ai dati e la gestione delle operazioni del sistema.</li></ol>	<p>Collabora con le classi DAO per accedere e manipolare i dati relativi agli utenti, strumenti, prenotazioni, laboratori, responsabili, team e tecnici.</p>

Table 1: CRC CARD *Controller*

### 4.1.2 DBConnection

Class: <b>DBCONNECTION</b>	
RESPONSIBILITIES	COLLABORATORS
<p>Fornire una connessione al database</p>	<p>Collabora con la classe <b>DriveManager</b> per ottenere una connessione al database utilizzando le informazioni di URL, username e password, e con le classi <b>Statement</b> e <b>PreparedStatement</b> per creare oggetti che possono essere utilizzati per inviare query SQL al database.</p>

Table 2: CRC CARD *DBConnection*

#### 4.1.3 EmailSender

Class: <b>EMAILSENDER</b>	
RESPONSIBILITIES	COLLABORATORS
Inviare e-mail di verifica, utilizzando un server SMTP (Simple Mail Transfer Protocol), per scopi come la registrazione o il recupero della password.	Interagisce con le classi fornite dall'API JavaMail, come <b>Session</b> , <b>Authenticator</b> , <b>PasswordAuthentication</b> , <b>Message</b> , <b>MimeMessage</b> e <b>Transport</b> , per gestire la creazione e l'invio del messaggio di posta elettronica, e interagisce con il server SMTP di gmail.

Table 3: CRC CARD *EmailSender*

*Per saltare tutte le CRC cards e passare alla sezione successiva clicca qua 5.*



## 4.2 CRC cards DAO e DTO

### 4.2.1 LaboratorioDAO e Laboratorio

Class: <b>LABORATORIDAO</b>	
RESPONSIBILITIES	COLLABORATORS
Recuperare informazioni sui laboratori dal database. Crea oggetti <b>Laboratorio</b> a partire dai risultati delle query eseguite sul database.	Collabora con la classe Controller e la classe <i>DBConnection</i>

Table 4: CRC CARD *LaboratorioDao*

Class: <b>LABORATORIO</b>	
RESPONSIBILITIES	COLLABORATORS
<ol style="list-style-type: none"><li>1. Rappresentare un oggetto <b>Laboratorio</b> con le sue caratteristiche principali, come il codice, la descrizione, gli orari di apertura e chiusura, il numero di tecnici e il numero di postazioni.</li><li>2. Fornire metodi getters e setters.</li></ol>	Collabora con la classe Controller e oggetti Java Time API

Table 5: CRC CARD *Laboratorio*

#### 4.2.2 PostazioneDAO e Postazione

Class: <b>POSTAZIONEDAO</b>	
RESPONSIBILITIES	COLLABORATORS
Recuperare informazioni sulle postazioni dal database. Crea oggetti <b>Postazione</b> a partire dai risultati delle query eseguite sul database.	Collabora con la classe Controller e la classe <i>DBConnection</i> .

Table 6: CRC CARD *PostazioneDao*

Class: <b>POSTAZIONE</b>	
RESPONSIBILITIES	COLLABORATORS
<ol style="list-style-type: none"> <li>1. Rappresentare un oggetto <b>Postazione</b> con le sue caratteristiche principali, come il codice della postazione, il numero di posti e il riferimento al laboratorio a cui appartiene.</li> <li>2. Fornire metodi getters e setters.</li> </ol>	Collabora con la classe Controller e <b>Laboratorio</b> .

Table 7: CRC CARD *Postazione*

#### 4.2.3 PrenotazioneDAO e Prenotazione

Class: <b>PRENOTAZIONEDAO</b>	
RESPONSIBILITIES	COLLABORATORS
<p>Recuperare informazioni sulle prenotazioni dal database. Crea oggetti <b>Prenotazione</b> a partire dai risultati delle query eseguite sul database.</p> <p>Effettuare operazione di recupero, inserimento, modifica e cancellazione.</p> <p>Gestire specifiche eccezioni associate a situazioni come la chiusura del laboratorio, la durata minima di utilizzo, sovrapposizioni di prenotazioni, e tentativi di prenotazione o modifica per date o orari passati.</p>	<p>Collabora con la classe Controller e la classe <i>DBConnection</i>.</p> <p>Collabora con le classi DAO come <b>UtenteDAO</b> e <b>StrumentoDAO</b> per recuperare informazioni su utenti e strumenti associati alle prenotazioni.</p>

Table 8: CRC CARD *PrenotazioneDao*

Class: <b>PRENOTAZIONE</b>	
RESPONSIBILITIES	COLLABORATORS
<ol style="list-style-type: none"> <li>1. Rappresentare un oggetto <b>Prenotazione</b> con le sue caratteristiche principali, come la data, l'ora, la durata, l'utente e lo strumento associato.</li> <li>2. Fornire metodi getters e setters.</li> </ol>	<p>Collabora con la classe Controller, con le classi <b>Utente</b> e <b>Strumento</b> per rappresentare le relazioni tra una prenotazione, l'utente e lo strumento associato e collabora con oggetti Date per rappresentare la data, l'ora e la durata della prenotazione.</p>

Table 9: CRC CARD *Prenotazione*

#### 4.2.4 ResponsabileDAO e Responsabile

Class: <b>RESPONSABILED AO</b>	
RESPONSIBILITIES	COLLABORATORS
<p>Recuperare informazioni sul responsabile dal database. Crea oggetti <b>Responsabile</b> a partire dai risultati delle query eseguite sul database.</p> <p>Effettuare operazione di verifica, recupero, inserimento, modifica e cancellazione.</p>	<p>Collabora con la classe Controller e la classe <i>DBConnection</i>.</p> <p>Collabora con la classe DAO <b>SedeDAO</b> per recuperare informazioni sulla sede del responsabile.</p>

Table 10: CRC CARD *ResponsabileDao*

Class: <b>RESPONSABILE</b>	
RESPONSIBILITIES	COLLABORATORS
<ol style="list-style-type: none"> <li>1. Rappresentare un oggetto <b>Responsabile</b> con le sue caratteristiche principali, come la matricola, il nome, il cognome, il codice fiscale, il telefono, l'email, la password e la sede di appartenenza.</li> <li>2. Fornire metodi getters e setters.</li> </ol>	<p>Collabora con la classe Controller, con le classi <b>Sede</b> per rappresentare la sede di appartenenza del responsabile.</p>

Table 11: CRC CARD *Responsabile*

#### 4.2.5 SedeDAO e Sede

Class: <b>SEDEDAO</b>	
RESPONSIBILITIES	COLLABORATORS
Recuperare informazioni sulle sedi dal database. Crea oggetti <b>Sede</b> a partire dai risultati delle query eseguite sul database.	Collabora con la classe Controller e la classe <i>DBConnection</i> .

Table 12: CRC CARD *SedeDao*

Class: <b>SEDE</b>	
RESPONSIBILITIES	COLLABORATORS
<ol style="list-style-type: none"><li>1. Rappresentare un oggetto <b>Sede</b> con le sue caratteristiche principali, come codice identificativo (CodS), il nome e l'indirizzo della sede.</li><li>2. Fornire metodi getters e setters.</li></ol>	Collabora con la classe Controller.

Table 13: CRC CARD *Sede*

#### 4.2.6 StrumentoDAO e Strumento

Class: <b>STRUMENTODAO</b>	
RESPONSIBILITIES	COLLABORATORS
Recuperare informazioni sugli strumenti dal database. Crea oggetti <b>Strumento</b> a partire dai risultati delle query eseguite sul database.	Collabora con la classe Controller e la classe <i>DBConnection</i> , con la classe <b>SedeDAO</b> per recuperare informazioni sulla sede associata a uno strumento e con la classe <b>PostazioneDAO</b> per recuperare informazioni sulla postazione associata a uno strumento.

Table 14: CRC CARD *StrumentoDao*

Class: <b>STRUMENTO</b>	
RESPONSIBILITIES	COLLABORATORS
<ol style="list-style-type: none"> <li>1. Rappresentare un oggetto <b>Strumento</b> con le sue caratteristiche principali, come codice identificativo, le caratteristiche tecniche, la descrizione, il tempo massimo di utilizzo, la postazione associata e la sede associata.</li> <li>2. Fornire metodi getters e setters.</li> </ol>	Collabora con la classe Controller, con la classe <b>Postazione</b> per rappresentare la postazione associata a uno strumento e con la classe <b>Sede</b> per rappresentare la sede associata a uno strumento.

Table 15: CRC CARD *Strumento*

#### 4.2.7 TeamDAO e Team

Class: <b>TEAMDAO</b>	
RESPONSIBILITIES	COLLABORATORS
<p>Recuperare informazioni sui team dal database. Crea oggetti <b>Team</b> a partire dai risultati delle query eseguite sul database.</p> <p>Effettuare operazione di recupero, inserimento, modifica e cancellazione.</p>	<p>Collabora con la classe Controller e la classe <i>DBConnection</i>, collabora con la classe <b>ResponsabileDAO</b> per recuperare informazioni sui responsabili associati ai team e con la classe <b>TecnicoDAO</b> per recuperare i tecnici associati ai team e dissociarli quando necessario</p>

Table 16: CRC CARD *TeamDao*

Class: <b>TEAM</b>	
RESPONSIBILITIES	COLLABORATORS
<ol style="list-style-type: none"> <li>1. Rappresentare un oggetto <b>Team</b> con le sue caratteristiche principali, come il codice del team, il nome, la descrizione, la matricola del leader, il numero di tecnici e il responsabile associato.</li> <li>2. Fornire metodi getters e setters.</li> </ol>	<p>Collabora con la classe Controller, e con la classe <b>Responsabile</b> per rappresentare e gestire l'informazione relativa al responsabile associato al team..</p>

Table 17: CRC CARD *Team*

#### 4.2.8 TecnicoDAO e Tecnico

Class: <b>TECNICODAO</b>	
RESPONSIBILITIES	COLLABORATORS
<p>Recuperare informazioni sui tecnici dal database. Crea oggetti <b>Tecnico</b> a partire dai risultati delle query eseguite sul database.</p> <p>Effettuare operazione di recupero, inserimento, modifica e cancellazione.</p>	<p>Collabora con la classe Controller e la classe <i>DBConnection</i>.</p>

Table 18: CRC CARD *TecnicoDao*

Class: <b>TECNICO</b>	
RESPONSIBILITIES	COLLABORATORS
<ol style="list-style-type: none"> <li>1. Rappresentare un oggetto <b>Tecnico</b> con le sue caratteristiche principali, come matricola, nome, cognome, codice fiscale, telefono, email, laboratorio e team associato.</li> <li>2. Fornire metodi getters e setters.</li> <li>3. Rappresentazione testuale: La classe fornisce un metodo toString.</li> <li>4. Confronto di uguaglianza: La classe implementa i metodi equals e hashCode.</li> </ol>	<p>Collabora con la classe Controller, con la classe <b>Laboratorio</b> per ottenere informazioni sul laboratorio associato al tecnico e con la classe <b>Team</b> per ottenere informazioni sul team associato al tecnico.</p>

Table 19: CRC CARD *Tecnico*



#### 4.2.9 UtenteDAO e Utente

Class: <b>UTENTEDAO</b>	
RESPONSIBILITIES	COLLABORATORS
Recuperare informazioni sugli utenti dal database. Crea oggetti <b>Utente</b> a partire dai risultati delle query eseguite sul database. Effettuare operazione di verifica, recupero, inserimento, modifica e cancellazione.	Collabora con la classe Controller e la classe <i>DBConnection</i> .

Table 20: CRC CARD *UtenteDao*

Class: <b>UTENTE</b>	
RESPONSIBILITIES	COLLABORATORS
<ol style="list-style-type: none"><li>1. Rappresentare un oggetto <b>Utente</b> con le sue caratteristiche principali, come username, l'email e la password.</li><li>2. Fornire metodi getters e setters.</li></ol>	Collabora con la classe Controller,

Table 21: CRC CARD *Utente*

## 4.3 CRC cards GUI

### 4.3.1 Application

Class: <b>APPLICATION</b>	
RESPONSIBILITIES	COLLABORATORS
Inizializzazione dell'applicazione.	Collabora con la classe <b>Controller</b> istanziando un oggetto di questa classe per gestire la logica dell'applicazione e con la classe <b>MainWindow</b> istanziando un oggetto di questa classe per rappresentare l'interfaccia utente principale.

Table 22: CRC CARD *Application*

### 4.3.2 MainWindow

Class: <b>MAINWINDOW</b>	
RESPONSIBILITIES	COLLABORATORS
<ol style="list-style-type: none"><li>1. Gestione dell'interfaccia utente principale dell'applicazione.</li><li>2. Si occupa della configurazione iniziale della finestra, impostando titolo, dimensioni, icona e altre proprietà.</li><li>3. Gestione del layout a schede.</li><li>4. La classe istanzia e gestisce vari pannelli.</li><li>5. Implementa metodi per aggiungere, mostrare e verificare la presenza di pannelli nel layout a schede.</li></ol>	<p>Collabora con la classe <b>Controller</b> poiché accetta un oggetto <b>Controller</b> come argomento nel costruttore di seguito la finestra principale interagisce con il controller per gestire gli eventi e le azioni dell'utente,</p> <p>con le classi <b>UserLoginPage</b> e <b>PasswordRecovery</b> incorporandole come pannelli nel layout a schede.</p> <p>Utilizza <b>CardLayout</b> per gestire un pannello che mostra diverse "schede" di contenuto in base al contesto dell'applicazione.</p>

Table 23: CRC CARD *MainWindow*

#### 4.3.3 NewBookingToolSelected

Class: <b>NEWBOOKINGTOOLSELECTED</b>	
RESPONSIBILITIES	COLLABORATORS
<ol style="list-style-type: none"> <li>1. E' responsabile della creazione e gestione dell'interfaccia utente per la prenotazione o la modifica di prenotazioni.</li> <li>2. Verifica che i dati inseriti dall'utente rispettino determinate condizioni.</li> <li>3. Invia email di conferma all'utente riguardo alle prenotazioni (incluso il riepilogo delle prenotazioni).</li> <li>4. Gestisce due modalità di utilizzo: la creazione di una nuova prenotazione (<b>NewBookingToolSelected</b>) e la modifica di una prenotazione esistente (<b>NewBookingToolSelected</b> con un costruttore diverso).</li> </ol>	<p>Collabora con la classe <b>Controller</b> per effettuare operazioni legate alle prenotazioni e alla gestione degli strumenti ,  con <b>EmailSender</b> per l'invio di email di conferma.</p>

Table 24: CRC CARD *NewBookingToolSelected*

#### 4.3.4 IncreaseFont

Class: <b>INCREASEFONT</b>	
RESPONSIBILITIES	COLLABORATORS
<p>E' responsabile per visualizzare testo con la possibilità di aumentare la dimensione del carattere.</p>	<p>Collabora con le altre classi <b>GUI</b> per creare etichette con caratteri personalizzati.</p>

Table 25: CRC CARD *IncreaseFont*

#### 4.3.5 CalendarDialog

Class: <b>CALENDARDIALOG</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile della creazione e visualizzazione di una finestra di dialogo che mostra il calendario delle prenotazioni per uno specifico strumento.	Collabora con la classe <b>TablePanel</b> per visualizzare la lista delle prenotazioni in forma tabellare, con il framework Swing di Java in particolare estende la finestra di dialogo ( <b>JDIALOG</b> ), collabora con un'istanza del <b>controller</b> (Controller) per recuperare le informazioni necessarie riguardanti le prenotazioni degli strumenti.

Table 26: CRC CARD *CalendarDialog*

#### 4.3.6 UserHomePage

Class: <b>USERHOMEPAGE</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile della creazione e visualizzazione della home page dedicata agli utenti dell'applicazione. Gestisce il cambio di pagina quando l'utente interagisce con i pulsanti per effettuare una nuova prenotazione o visualizzare le proprie prenotazioni.	Collabora con la classe <b>NewBooking</b> per gestire la creazione di una nuova prenotazione quando l'utente preme il pulsante "Nuova prenotazione", con la classe <b>MyBooking</b> per visualizzare le prenotazioni dell'utente quando l'utente preme il pulsante "Le mie prenotazioni", con la classe <b>MainWindow</b> per gestire il cambio di pannelli o schede nella finestra principale dell'applicazione e collabora con un'istanza del controller ( <b>myController</b> ) per gestire le operazioni relative alle prenotazioni, come il recupero delle prenotazioni dell'utente.

Table 27: CRC CARD *UserHomePage*

#### 4.3.7 NoScalingIcon

Class: <b>NOSCALINGICON</b>	
RESPONSIBILITIES	COLLABORATORS
Evita la scalatura dell'icona in base alle dimensioni del componente su cui viene visualizzata.	Collabora con altre classi <b>GUI</b> .

Table 28: CRC CARD *NoScalingIcon*

#### 4.3.8 ManagerHomePage

Class: <b>MANAGERHOMEPAGER</b>	
RESPONSIBILITIES	COLLABORATORS
<p>E' responsabile della creazione e visualizzazione della home page dedicata ai responsabili dell'applicazione.</p> <p>Gestisce il cambio di pagina quando il responsabile interagisce con il pulsante "Esci", effettuando il logout e ritornando alla pagina di accesso per i responsabili.</p>	<p>Collabora con la classe <b>ManagerLoginPage</b> per gestire il cambio di pagina quando il responsabile decide di effettuare il logout, con la classe <b>MyTeam</b> per visualizzare le informazioni relative ai team del responsabile quando l'utente preme il pulsante "I miei Teams", con la classe <b>MainWindow</b> per gestire il cambio di pannelli o schede nella finestra principale dell'applicazione e collabora con un'istanza del controller (<b>myController</b>) per gestire le operazioni relative alle informazioni del responsabile e alle azioni specifiche del responsabile.</p>

Table 29: CRC CARD *ManagerHomePage*

#### 4.3.9 LinkMouseOn

Class: <b>LINKMOUSEON</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile della creazione di un componente di testo Swing che simula l'aspetto di un link ipertestuale. L'effetto "link" include un cambio di colore del testo e la visualizzazione di un bordo inferiore al passaggio del mouse.	Collabora con le classi <b>GUI</b> in cui è presente un "link".

Table 30: CRC CARD *LinkMouseOn*

#### 4.3.10 ModOrganigramma

Class: <b>MODORGANIGRAMMA</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile della gestione dell'interfaccia grafica per la modifica dell'organigramma di un team.	Collabora con classi <b>Swing</b> , con un'istanza del controller ( <b>myController</b> ) per effettuare le operazioni di aggiornamento necessarie nel database quando viene confermata la sostituzione di un tecnico, con due tabelle di tecnici, consentendo la selezione di un tecnico da sostituire dalla prima tabella e un tecnico sostituito dalla seconda tabella e con altre classi GUI come <b>MainWindow</b> , <b>MyTeam</b> , e <b>TableModelTeam</b> .

Table 31: CRC CARD *ModOrganigramma*

#### 4.3.11 MyBooking

Class: <b>MYBOOKING</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile della visualizzazione delle prenotazioni effettuate da un utente.	Collabora con classi <b>Swing</b> , con un'istanza del controller ( <b>myController</b> ) per recuperare e gestire le prenotazioni dell'utente, implementa l'interfaccia <b>BookingSelectionListener</b> per ricevere notifiche quando viene selezionata una prenotazione nella tabella delle prenotazioni. In risposta a tale selezione, la classe aggiorna la visibilità dei bottoni di modifica ed eliminazione e infine collabora con altre classi GUI come <b>MainWindow</b> , <b>BookingFrame</b> , <b>TablePanel</b> .

Table 32: CRC CARD *MyBooking*

#### 4.3.12 BookingSelectionListener

Class: <b>BOOKINGSELECTIONLISTENER</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile per essere implementata da classi interessate a ricevere notifiche quando una prenotazione è stata selezionata.	Collabora con la classe <b>Prenotazione</b> per specificare che le classi che la implementano devono gestire la selezione di un'istanza di <b>Prenotazione</b> .

Table 33: CRC CARD *BookingSelectionListener*

#### 4.3.13 MyTeam

Class: <b>MYTEAM</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile della visualizzazione dei team associati a un responsabile.	Collabora con classi <b>Swing</b> , con un'istanza del controller ( <b>myController</b> ) per recuperare e gestire i team associati a un responsabile, con una tabella per visualizzare i dettagli dei team ( <b>JTable</b> ) e con altre classi GUI, come <b>MainWindow</b> , <b>NewTeam</b> , <b>OrganigramFiveTechnicians</b> , <b>OrganigramTenTechnicians</b> e <b>ManagerHomePage</b> .

Table 34: CRC CARD *MyTeam*

#### 4.3.14 NewBooking

Class: <b>NEWBOOKING</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile della gestione dell'interfaccia utente per la creazione di nuove prenotazioni.	Collabora con classi <b>Swing</b> , con un'istanza del controller ( <b>myController</b> ) per recuperare dati, come la lista degli strumenti e le informazioni sulle prenotazioni e con altre classi GUI, come <b>MainWindow</b> , <b>BookingFrame</b> , <b>CalendarDialog</b> , <b>ToolBookingSummary</b> e <b>TablePanel</b> .

Table 35: CRC CARD *NewBooking*



#### 4.3.15 SignInPageManager

Class: <b>SIGNINPAGEMANAGER</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile della creazione dell'interfaccia utente grafica per la registrazione di un nuovo responsabile.	Collabora con classi <b>Swing</b> , con un oggetto di tipo <b>Controller</b> per eseguire operazioni logiche come la verifica della matricola e dell'email, con la classe <b>EmailSender</b> per l'invio di email di verifica, con oggetti di tipo <b>SedeDAO</b> per ottenere informazioni sulle sedi dal database e con altre classi GUI come <b>ManagerLoginPage</b> , quando l'utente preme il pulsante "Indietro" per tornare alla pagina di accesso.

Table 36: CRC CARD *SignInPageManager*

#### 4.3.16 NewTeam

Class: <b>NEWTEAM</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile della creazione dell'interfaccia grafica per l'inserimento di un nuovo team e l'inserimento di tecnici nel team.	Collabora con classi <b>Swing</b> , con un oggetto di tipo <b>Controller</b> e con altre classi come <b>TextFieldBorderColor</b> , <b>Tecnico</b> , <b>Responsabile</b> , <b>Team</b> , <b>TecnicoDAO</b> .

Table 37: CRC CARD *NewTeam*

#### 4.3.17 OrganigramFiveTechnicians

Class: <b>ORGANIGRAMFIVETECHNICIANS</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile della rappresentazione un organigramma con cinque livelli gerarchici. L'organigramma visualizza i dettagli di un team, tra cui il leader, il coordinatore e tre responsabili. Include anche la possibilità di modificare la struttura del team.	Collabora con classi <b>Swing</b> , con la classe <b>Controller</b> , con la classe <b>Responsabile</b> per ottenere informazioni sul responsabile loggato, con la classe <b>Team</b> per ottenere informazioni sul team per il quale viene visualizzato l'organigramma, con la classe <b>Tecnico</b> per ottenere la lista dei tecnici membri del team. La classe collabora con questa lista per ottenere informazioni sui membri del team e mostrarle nell'organigramma, con la classe <b>ModOrganigramma</b> per consentire la modifica della struttura del team e con la classe <b>MainWindow</b> per navigare tra le schermate dell'applicazione.

Table 38: CRC CARD *Organigramma5Panel*

#### 4.3.18 PasswordRecovery

Class: <b>PASSWORDRECOVERY</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile per la gestione della procedura di recupero della password dell'utente nell'applicazione.	Collabora con classi <b>Swing</b> , con la classe <b>Controller</b> per gestire il recupero della password, con le classi <b>TextFieldBorderColor</b> , <b>IncreaseFont</b> e <b>EmailSender</b> .

Table 39: CRC CARD *PasswordRecovery*

#### 4.3.19 OrganigramTenTechnicians

Class: <b>ORGANIGRAMFIVETECHNICIANS</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile della rappresentazione un organigramma con dieci livelli gerarchici. L'organigramma visualizza i dettagli di un team, tra cui il leader, due coordinatori e sette responsabili. Include anche la possibilità di modificare la struttura del team.	Collabora con classi <b>Swing</b> , con la classe <b>Controller</b> , con la classe <b>Responsabile</b> per ottenere informazioni sul responsabile loggato, con la classe <b>Team</b> per ottenere informazioni sul team per il quale viene visualizzato l'organigramma, con la classe <b>Tecnico</b> per ottenere la lista dei tecnici membri del team. La classe collabora con questa lista per ottenere informazioni sui membri del team e mostrarle nell'organigramma, con la classe <b>ModOrganigramma</b> per consentire la modifica della struttura del team e con la classe <b>MainWindow</b> per navigare tra le schermate dell'applicazione.

Table 40: CRC CARD *OrganigramTenTechnicians*

#### 4.3.20 PasswordRecoveryManager

Class: <b>PASSWORDRECOVERYMANAGER</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile per la gestione della procedura di recupero della password del responsabile nell'applicazione.	Collabora con classi <b>Swing</b> , con la classe <b>Controller</b> per gestire il recupero della password, con le classi <b>TextFieldBorderColor</b> , <b>IncreaseFont</b> e <b>EmailSender</b> .

Table 41: CRC CARD *PasswordRecoveryManager*

#### 4.3.21 UserLoginPage

Class: <b>USERLOGINPAGE</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile della creazione e organizzazione della pagina di accesso dell'applicazione.	Collabora con classi <b>Swing</b> , con la classe <b>Controller</b> per gestire le operazioni principali del sistema, inclusa la gestione delle autenticazioni, con la classe

Table 42: CRC CARD *UserLoginPage*

#### 4.3.22 ManagerLoginPage

Class: <b>MANAGERLOGINPAGE</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile per l'accesso di un responsabile.	Se le credenziali sono corrette collabora con la classe <b>ManagerHomePage</b> . Nel caso in cui un responsabile ha dimenticato la password collabora con la classe <b>PasswordRecoveryManager</b> . Quando un responsabile vuole registrarsi collabora con la classe <b>SignInPageManager</b> . Collabora con un oggetto Controller ( <b>myController</b> ).

Table 43: CRC CARD *ManagerLoginPage*

#### 4.3.23 SignInPage

Class: <b>SIGNINPAGE</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile della registrazione di un nuovo utente.	Collabora con un oggetto Controller ( <b>myController</b> ) per eseguire operazioni come la verifica dell'esistenza di email o username nel database. Collabora con la classe <b>UserLoginPage</b> per consentire all'utente di tornare alla pagina di accesso e con la classe <b>EmailSender</b> per inviare un codice di verifica all'utente via email.

Table 44: CRC CARD *SignInPage*

#### 4.3.24 ToolBookingSummary

Class: <b>TOOLBOOKINGSUMMARY</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile della presentazione di un dialogo che consente all'utente di selezionare un mese e un anno per ottenere un riepilogo di uno specifico strumento.	Collabora con un oggetto Controller ( <b>myController</b> ) per richiedere il riepilogo dello strumento con le informazioni selezionate dall'utente e con un oggetto <b>Strumento</b> che rappresenta lo strumento per il quale l'utente richiede il riepilogo.

Table 45: CRC CARD *ToolBookingSummary*

#### 4.3.25 MyBookingsTableModel

Class: <b>MYBOOKINGSTABLEMODEL</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile della gestione e della presentazione dei dati delle prenotazioni in una tabella.	Collabora principalmente con i componenti Swing, oggetti <b>SimpleDateFormat</b> per formattare le date e le ore nelle colonne corrispondenti e collabora con l'estensione e i suoi metodi <b>AbstractTableModel</b> , fornendo un'implementazione di base per un modello di tabella astratto.

Table 46: CRC CARD *MyBookingsTableModel*

#### 4.3.26 ToolsBookingsTableModel

Class: <b>TOOLSBOOKINGSTABLEMODEL</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile della gestione e della presentazione dei dati delle prenotazioni relative agli strumenti in una tabella.	Collabora principalmente con i componenti Swing, oggetti <b>SimpleDateFormat</b> per formattare le date e le ore nelle colonne corrispondenti e collabora con l'estensione e i suoi metodi <b>AbstractTableModel</b> , fornendo un'implementazione di base per un modello di tabella astratto e con una lista di oggetti <b>Prenotazione</b> specifici per gli strumenti come dati del modello.

Table 47: CRC CARD *ToolsBookingsTableModel*

#### 4.3.27 ToolsTableModel

Class: <b>TOOLSTABLEMODEL</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile della gestione e della presentazione dei dati degli strumenti in una tabella.	Collabora principalmente con i componenti Swing, oggetti <b>SimpleDateFormat</b> per formattare le date e le ore nelle colonne corrispondenti e collabora con l'estensione e i suoi metodi <b>AbstractTableModel</b> , fornendo un'implementazione di base per un modello di tabella astratto e con una lista di oggetti <b>Strumento</b>

Table 48: CRC CARD *ToolsTableModel*

#### 4.3.28 TableModelTeam

Class: <b>TABLEMODELTEAM</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile della gestione e della presentazione dei dati dei tecnici di un team in una tabella.	Collabora principalmente con i componenti Swing, collabora con l'estensione e i suoi metodi <b>AbstractTableModel</b> , fornendo un'implementazione di base per un modello di tabella astratto e con una lista di oggetti <b>Tecnico</b> .

Table 49: CRC CARD *TableModelTeam*

#### 4.3.29 TablePanel

Class: <b>TABLEPANEL</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile della presentazione e gestione delle interazioni utente con i dati degli strumenti e delle prenotazioni.	Collabora principalmente con i modelli di <b>tabelle descritte sopra</b> , con l'implementazione di un <b>ListSelectionListener</b> per gestire gli eventi di selezione delle tabelle e con la classe <b>NewBooking</b> .

Table 50: CRC CARD *TablePanel*

#### 4.3.30 TextFieldBoderColor

Class: <b>TEXTFIELDBORDERCOLOR</b>	
RESPONSIBILITIES	COLLABORATORS
E' responsabile del cambiamento di colore del bordo del campo di testo in risposta agli eventi di focus.	Collabora con altre classi <b>GUI</b> .

Table 51: CRC CARD *TextFieldBorderColor*

## 5 Sequence Diagram

### 5.1 Sequence diagram della prima funzionalità a scelta

Il sequence diagram descrive gli eventi quando il bottone registrati viene cliccato. Quando l'utente fa clic sul pulsante, vengono gestiti alcuni casi: se email, username e password, sono vuoti si visualizza un messaggio d'avviso, altrimenti viene creato un nuovo oggetto Utente con le informazioni fornite dall'utente, viene verificato se l'email e l'username sono già presenti nel database. Se non lo sono, viene generato un codice casuale e inviato via email all'utente che poi dovrà inserire. Se il codice inserito corrisponde a quello generato, l'utente viene registrato correttamente e viene visualizzato un messaggio di conferma. In caso contrario, viene visualizzato un messaggio di errore. Se l'email o l'username sono già presenti nel database, viene visualizzato un messaggio di avviso.

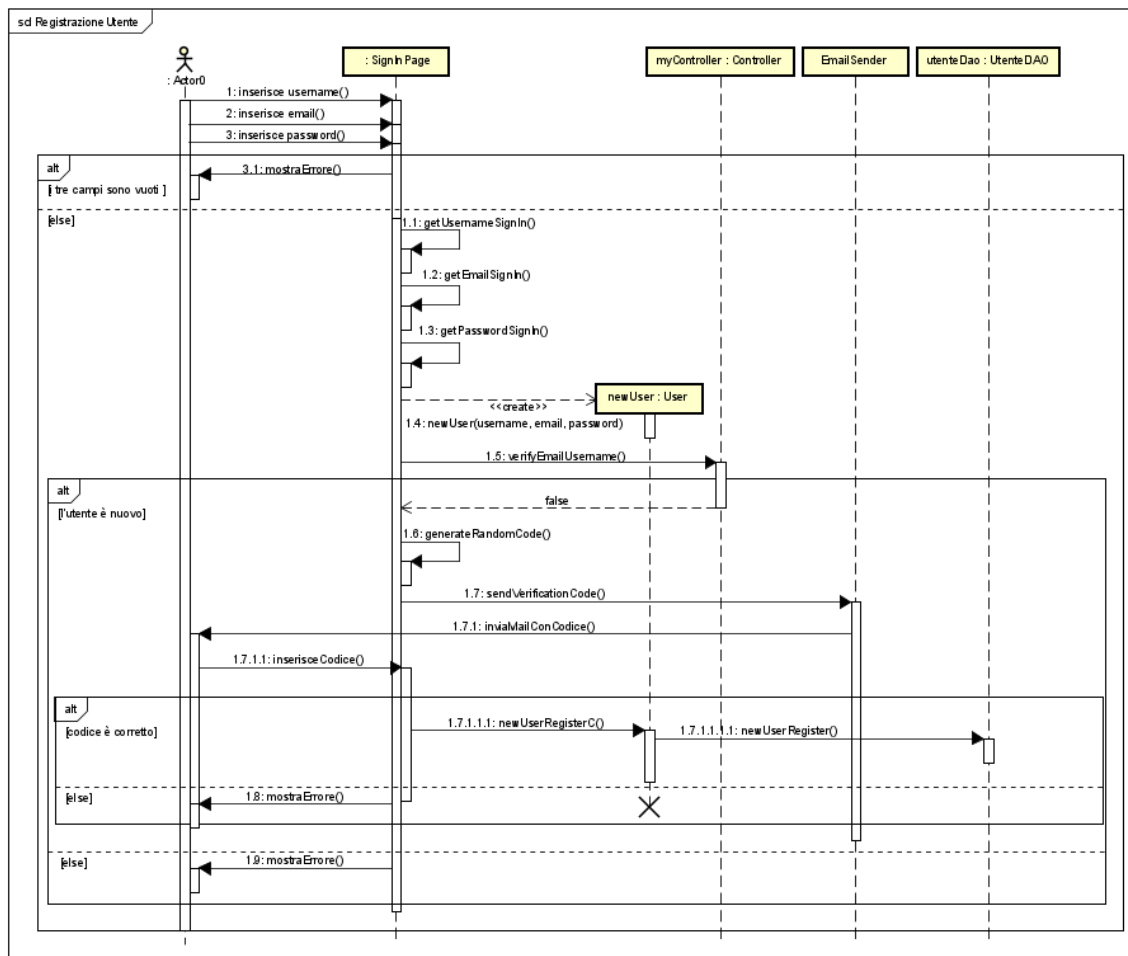


Figure 3: Sequence diagram prima funzionalità



## 5.2 Sequence diagram della seconda funzionalità a scelta

Il secondo sequence diagram gestisce gli eventi quando il bottone Cerca viene premuto. Quando l'utente fa clic sul pulsante, si controlla prima se sia stato inserito un valore per la sede e per la descrizione. Se entrambi i campi non sono vuoti, viene chiamato un metodo per recuperare una lista di strumenti in base ai campi forniti. Se la lista non è vuota, vengono mostrati gli strumenti in una tabella. Viene aggiunta anche la possibilità di annullare la ricerca e ripristinare la tabella di visualizzazione degli strumenti disponibili. Se la lista è vuota, viene visualizzato un messaggio di avviso. Se nessuno dei due campi è inserito, viene mostrato un messaggio di avviso.<sup>3</sup>

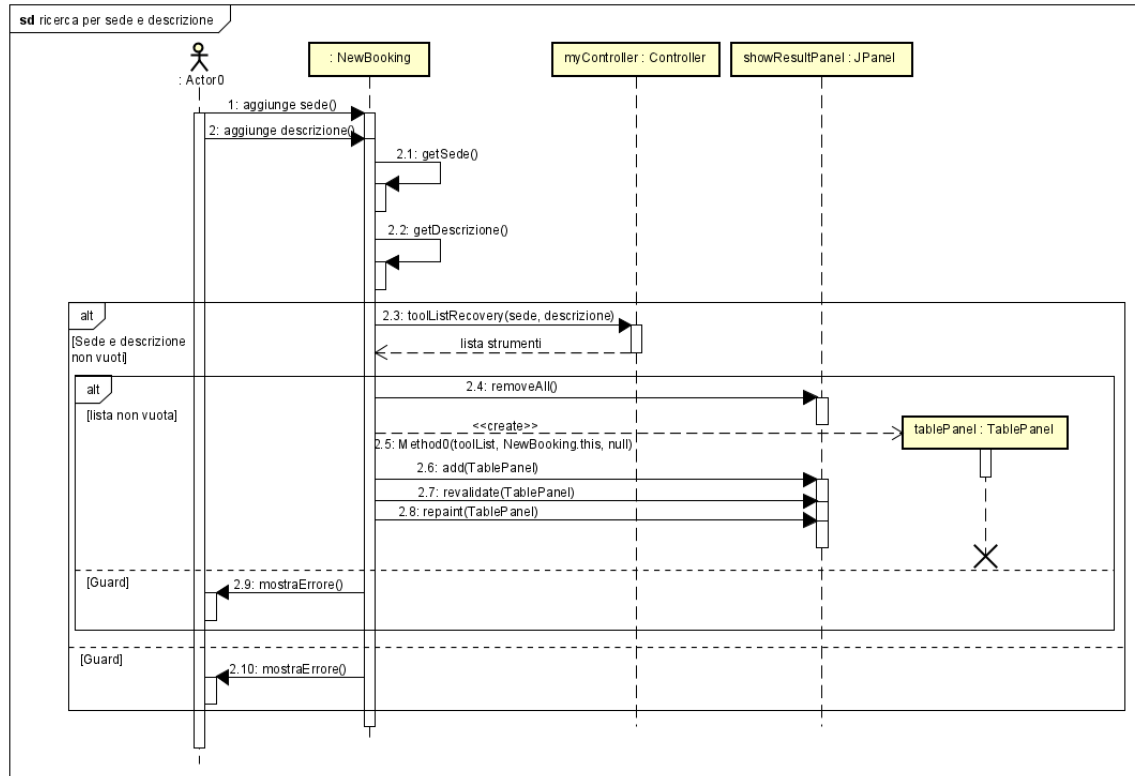


Figure 4: Sequence diagram seconda funzionalità

<sup>3</sup>C'è un altro caso da gestire, ovvero se solo uno dei due campi (sede o descrizione) è inserito, viene effettuata una ricerca solo per quel campo. Nel sequence diagram questo caso non è stato rappresentato in quanto la logica era simile a quella descritta fin'ora e poi il sequence diagram sarebbe diventato troppo lungo.

## 6 Piccolo esempio

Dopo aver spiegato le entità, le responsabilità, le collaborazioni e l'organizzazione del nostro progetto a grandi linee, sono riportate di seguito le schermate principali del progetto. La schermata login per l'utente e la schermata della registrazione per il responsabile.<sup>4</sup> <sup>5</sup>

### 6.1 Login Utente

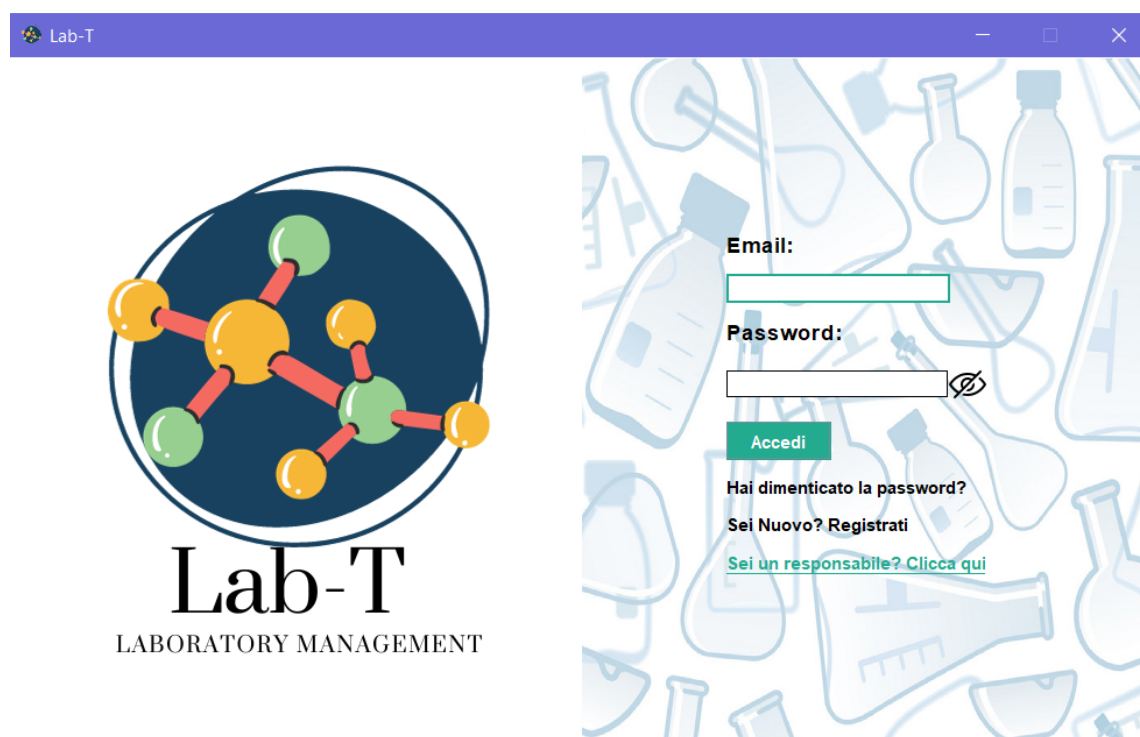


Figure 5: Login Utente

Si può notare che da questa pagina si può recuperare le credenziali, registrarsi e accedere all'area responsabile nel caso ad accedere nel nostro sistema sia proprio un responsabile. A seguito di questa pagina, dopo il login con tutte le credenziali corrette, si passerà alla pagina che ci permette di prenotare uno strumento e si presenta così

---

<sup>4</sup>Schermate prese a caso, giusto per far vedere un po' il nostro progetto

<sup>5</sup>Altra nota: abbiamo scelto di non permettere il ridimensionamento della finestra e di fissare dimensioni specifiche per tutte le finestre.

Lab-T

Sede:

Descrizione:

Codice Strumento	Caratteristiche	Descrizione	Tempo Max Uso	Cod. Postazione	Sede
1	Intel Core i7, 16 GB RA...	Computer Desktop	01:30	P1	CERN
2	Dell U2415	Monitor 24 pollici	01:30	P1	CERN
3	Apple Magic Mouse 2	Mouse wireless	01:30	P1	CERN
4	Apple Keyboard	Tastiera Wireless	01:30	P11	CERN
5	Duplicatore forence	Tableau TD2u	01:30	P1	CERN
6	Raspberry Pi 4	Mini computer single-b...	01:30	P11	CERN
7	Capacità 500 mL, Prec...	Pipetta automatica	00:30	P2	Lawrence Berkeley
8	Intervallo di misura 0-1...	Termometro digitale	00:30	P15	Institut Pasteur
9	Intervallo di misura pH...	pH-metro	00:30	P16	Nazionale Argonne
10	Volume max 100 mL, ...	Buretta	00:30	P16	Institut Pasteur
11	Ingrandimento 40-1600...	Microscopio elettronico	00:50	P3	Max Planck
12	Ingrandimento 40-400x...	Microscopio ottico	00:50	P17	Salk Institute
13	Risoluzione 2D/4D, Lu...	Ecografo	02:00	P4	Scripps Research
14	Risoluzione 16-slice, T...	Tomografo a raggi X	02:00	P14	Scripps Research
15	Velocità di scansione 2...	Analizzatore di conta d...	00:10	P5	Oak Ridge
16	Precisione $\pm 1 \mu\text{L}$ , Inter...	Microlitro	00:10	P5	Synchrotron Light Sour...
17	Ingrandimento 40-1600...	Microscopio elettronico	00:10	P12	Fraunhofer
18	Intervallo di misura 0-1...	Termometro	00:10	P12	Oak Ridge
19	Ingrandimento 40-400x...	Microscopio ottico	00:10	P13	Oak Ridge
20	Intervallo di misura 0-1...	Medisana	00:10	P13	Fraunhofer
21	Precisione $\pm 0,1 \mu\text{g}$ , Int...	Sonda ELISA	00:50	P6	Turing
22	Precisione $\pm 0,5 \text{ mL}$ , V...	Pipetta a volume variab...	00:50	P6	Von Neumann
23	Formato max A3, Sta...	Stampante a getto di in...	03:30	P7	Max Planck

Indietro

Prenota

Calendario Prenotazioni

Riepilogo

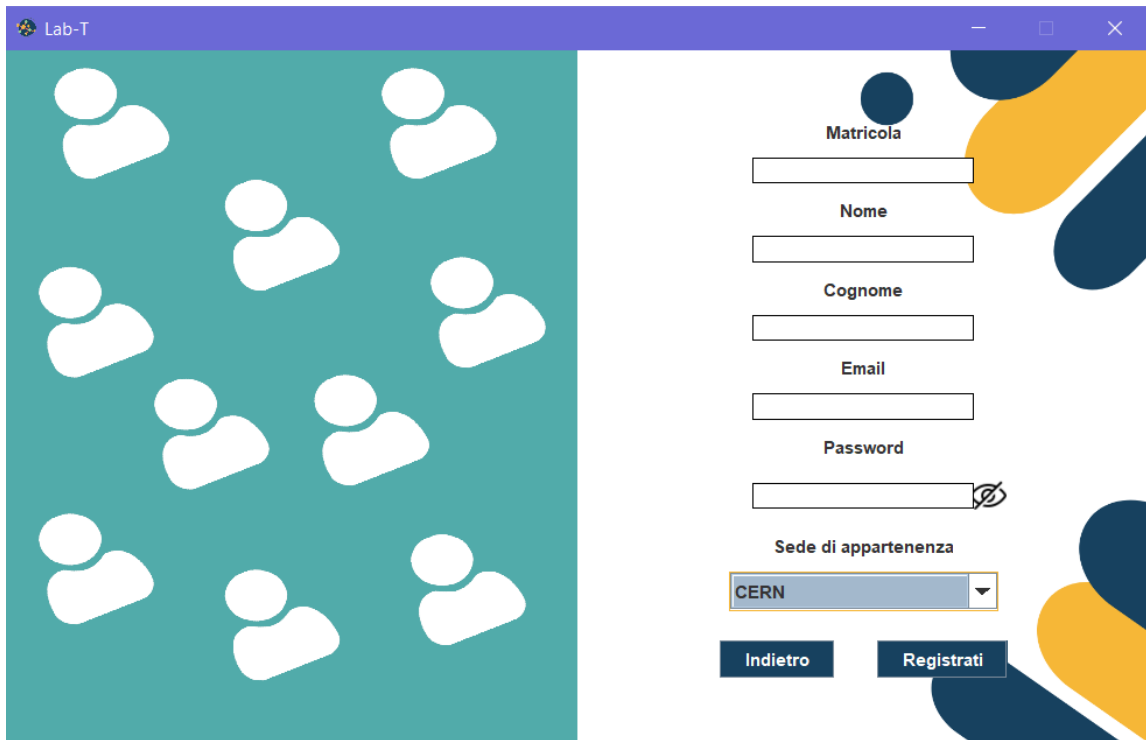
Cerca

Figure 6: Prenotazione strumento

Possiamo vedere che tramite un JTextField si può cercare uno strumento per sede e/o per descrizione. Una volta selezionato lo strumento si potrà vedere la disponibilità di quello strumento, il riepilogo e si potrà prenotare cliccando il bottone Conferma Prenotazione una volta inseriti i dati delle prenotazione.

## 6.2 Registrazione Responsabile

Una volta cliccato sul link "Sei un responsabile?Clicca qui" che si trova nella pagina di login dell'utente 5, veniamo indirizzati nella parte che riguarda responsabile. Ci apparirà per prima la pagina di login del responsabile, che sostanzialmente è simile alla pagina di login dell'utente, quindi si potrà accedere alla propria area personale, recuperare le credenziali e registrarsi. In questo caso vogliamo farvi vedere la finestra inerente alla registrazione di un nuovo responsabile.



The screenshot shows a web browser window with the title 'Lab-T'. The page is divided into two main sections. On the left, there is a teal-colored area with several white, stylized human icons. On the right, there is a white registration form. The form contains the following fields and elements:

- Matricola**: A text input field.
- Nome**: A text input field.
- Cognome**: A text input field.
- Email**: A text input field.
- Password**: A text input field with a toggle icon (an eye with a slash) to the right, indicating a password field.
- Sede di appartenenza**: A dropdown menu (JComboBox) with 'CERN' selected.
- Indietro**: A dark blue button with white text.
- Registrati**: A dark blue button with white text.

The background of the right section features abstract, overlapping shapes in dark blue and yellow.

Figure 7: Registrazione responsabile

Come si può vedere troviamo i campi matricola,nome,... e la sede di appartenenza che è rappresentata mediante una JComboBox, ovvero un menu a discesa che consente ai responsabili di selezionare una sede dall'elenco. Poi ci sono i due bottoni, uno che permette di tornare alla pagina di login del responsabile e l'altro, dopo aver compilato tutti i campi, di registrarsi. Dopo essersi loggato o registrato apparirà questa schermata,da qui si può visualizzare, una volta cliccato il team, il suo organigramma, eliminare il team selezionato oppure aggiungerne uno nuovo.

