

LabManager

Documentazione

*A.A. 2022/2023
OBJECT ORIENTATION*

*GIOVANNI ADAMO
N86004180*

*NICOLA ABBATE
N86004088*

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE DIPARTIMENTO DI INGEGNERIA
ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE



CORSO DI LAUREA IN INFORMATICA

INSEGNAMENTO DI OBJECT ORIENTATION - ANNO ACCADEMICO 2022/2023

SISTEMA DI GESTIONE DI UN LABORATORIO SCIENTIFICO

Autori

Giovanni Adamo N86004180
giovanni.adamo@studenti.unina.it

Nicola Abbate N86004088
nicola.abbate@studenti.unina.it

Docente

Prof. Sergio Di Martino

Gennaio 2023

Sommario

0	Presentazione	2
1	Introduzione	4
2	Struttura del codice	5
2.1	Package: GUI	5
2.2	Package: DTO e Package: DAO	6
2.2	Package UTILITIES	7
3	Class Diagram.....	8
4	Sequence Diagram	9
4.1	Sequence: Diventa Responsabile	9
4.2	Sequence: Recupera Password.....	10
5	CRC Cards	11
5.1	CRC: UTILITIES.....	11
5.2	CRC: GUI	12
5.3	CRC: DTO e DAO.....	13
6	Breve esposizione	18
6.1	Avvio del Software e Login	18
6.2	Registrazione.....	19
6.3	Recupero Password.....	20
6.4	Schermata Generale	20

1 Introduzione

LabManager è una piattaforma per la gestione di un laboratorio scientifico, personalizzabile al 100% in base alle esigenze dell'utente che si interfaccia a tale applicativo. Sarà infatti possibile adattare l'utilizzo di LabManager ad un qualsiasi tipo di laboratorio, attraverso le funzionalità disponibili per i responsabili di laboratorio, come la creazione di nuove postazioni e, per ognuna di esse, l'aggiunta di nuovi strumenti.

Sarà inoltre possibile effettuare prenotazioni per tali strumenti, a patto che l'utente sia iscritto al singolo laboratorio, o comunque lavori nella sede in cui decide di effettuare la prenotazione, che potrà gestire in qualsiasi momento, modificandola o annullandola.

L'intero applicativo si presenta come un'interfaccia user-friendly, di estrema accessibilità per qualsiasi tipo di utente, con una chiara disposizione dell'interfaccia grafica, che agevola lo spostamento tra le numerose funzioni che LabManager ha da offrire. L'intero programma si basa su una prima finestra che gestirà il login/registrazione di un utente; una volta effettuato l'accesso, dalla finestra principale ci si potrà spostare al meglio all'interno della piattaforma, grazie ad un comodo menu situato sulla sinistra, dal quale è possibile accedere alle diverse schermate (Laboratori, Strumenti, Prenotazioni), tenendo conto dell'utente attualmente connesso.



Il software è scritto in linguaggio Java ed utilizza un database (locale) strutturato e gestito tramite il DBMS PostgreSQL, e collegato tramite JDBC. Il programma è realizzato da Giovanni Adamo e Nicola Abbate come progetto di tipo accademico per l'Università degli Studi di Napoli Federico II, insegnamento di Object Orientation del prof. Sergio Di Martino.

2 Struttura del codice

L'intero codice è suddiviso in quattro diversi package, con l'aggiunta di un quinto che conserva al suo interno i file multimediali utilizzati all'interno del codice. I package sono così suddivisi:

- **GUI:** contiene le classi necessarie alla gestione delle interfacce grafiche.
- **DTO:** sono contenute le classi che riconosciamo effettivamente come entità.
- **DAO:** speculare alla DTO, serve ad interfacciare le entità al Database.
- **UTILITIES:** classi di supporto e classe Controller.

Sono descritti nel dettaglio i package utilizzati e le classi contenute al loro interno.

2.1 Package: GUI

Le interfacce grafiche sono realizzate utilizzando le classi e i relativi metodi della libreria Java Swing, con alcune implementazioni di java.awt (come, ad esempio, per la classe Color). Il software si presenta suddiviso in una prima parte di login, per finire poi in quella che è la schermata principale; per quanto riguarda la prima parte, il tutto è gestito all'interno di tre diverse sottoclassi di JFrame da noi create, quali: LoginWindow per il login, RegisterWindow per la registrazione ed infine PasswordRecoveryWindow per il recupero della password. Successivamente, una volta effettuato il login, ci si muoverà all'interno della MainWindow (extends JFrame). Come per le finestre, i componenti utilizzati sono definiti in classi separate, in quanto sottoclassi di quelle fornite da Java Swing (ad esempio AvantiButton è una sottoclasse di JButton); ciò viene effettuato per tutti i componenti più importanti, o per quelli che verranno riutilizzati più volte (a differenza di, ad esempio, semplici JLabel che mostrano un normale testo, o dei JButton che non dovranno essere riutilizzati). Tutti i componenti non verranno adattati direttamente sui Frame, bensì sono state create diverse sottoclassi di JPanel che ospiteranno i componenti utilizzati, e saranno a loro volta inserite nei rispettivi Frame.

Il tutto viene poi richiamato ed istanziato attraverso il Controller.

2.2 Package: DTO e Package: DAO

Nel package DTO sono inserite tutte le classi Java che identifichiamo come effettive entità del programma; sono composte da un insieme di attributi (visibilità private) e dai rispettivi Getters() e Setters(), per poi essere richiamati e gestiti tramite il Controller.

Per quanto invece riguarda il package DAO, quest'ultimo è composto dallo stesso numero di classi di DTO (in quanto i due package sono speculari), consentendo di poter far interfacciare le singole entità al Database, selezionando o inserendo i dati richiesti.

La classe [Tecnico](#) gestisce l'utente iscritto alla piattaforma, raccogliendone i dati anagrafici, le credenziali di accesso e altri dati personali (come telefono ed e-mail); la classe [TecnicoDAO](#), invece, attraverso alcuni metodi consente di effettuare controlli sull'esistenza dei campi inseriti e la validità di questi ultimi, in modo conforme a quanto espresso nel Database; inoltre, consente di inserire nuovi tecnici, o di selezionarne di già esistenti (singolarmente o utilizzando una Collection).

La classe [Laboratorio](#) conserva al suo interno i dati di un laboratorio, che abbiamo identificato come un'entità composta da un tipo (ad esempio informatica, biologia, chimica) e la sede a cui è associato; per ogni laboratorio, oltre ad una descrizione e al suo codice identificativo, sono presenti un numero di tecnici variabile, e un singolo responsabile (abbiamo pensato di “ampliare” la consegna, fornendo ad un utente, se quest'ultimo è responsabile di un laboratorio, la possibilità di aggiungere nuove postazioni associate al laboratorio da lui gestito, e nuovi strumenti per ogni postazione, invece di rendere possibile il popolamento di questi due campi solo dal database). Nella classe [LaboratorioDAO](#), sono presenti metodi per iscrivere un tecnico ad un dato laboratorio, oltre alla possibilità di selezionarli e inserirne di nuovi.

Analogamente, come già anticipato, sarà possibile aggiungere nuove postazioni associate ad un dato laboratorio esclusivamente da un responsabile di quest'ultimo; ciò viene effettuato tramite le classi [Postazione](#) e [PostazioneDAO](#). Allo stesso modo, è possibile aggiungere dei nuovi strumenti alle postazioni esistenti, registrandone tutti i dati richiesti per identificarli, la loro locazione e un tempo massimo di utilizzo; tutto questo, grazie alle classi [Strumento](#) e [StrumentoDAO](#), che si interfaccia al DB.

Alcune informazioni inerente alle singole sedi, hanno bisogno di essere trattate separatamente; per questo utilizziamo le classi [Sede](#) e [SedeDAO](#) per ottenere tali informazioni, come ad esempio tutte le sedi in cui è iscritto un determinato Tecnico.

Le classi [Prenotazione](#) e [PrenotazioneDAO](#) sono invece utilizzate per gestire le nuove prenotazioni degli strumenti effettuate dagli utenti e inserirle nel Database, oppure selezionarle per verificare quali prenotazioni sono state effettuate; a differenza, le due classi [PrenotazionePersonale](#) e [PrenotazionePersonaleDAO](#) gestiscono le prenotazioni effettuate in precedenza da parte di un utente specifico; sono utilizzate particolarmente nella schermata “Prenotazioni”, consultabile dal menu laterale in qualsiasi momento.

Infine, per visualizzare le prenotazioni passate (concluse), utilizziamo le classi [Storico](#) e [StoricoDAO](#), grazie alle quali possiamo avere un riepilogo completo sull'utilizzo dei singoli strumenti, annualmente e mensilmente (tempo di utilizzo, top utilizzatore).

2.2 Package UTILITIES

In questo package sono contenute diverse classi utili alla gestione dei dati all'interno del codice e, indispensabile, il Controller che collega le varie classi del progetto attraverso i metodi definiti al suo interno.

La classe [ConnessioneDB](#) effettua la connessione al Database, utilizzando JDBC (il tutto avviene dopo l'implementazione di un plugin per PostgreSQL); tale classe, sarà collegata inoltre a tutte le classi DAO, in quanto esse si interfacciano in maniera diretta al DB.

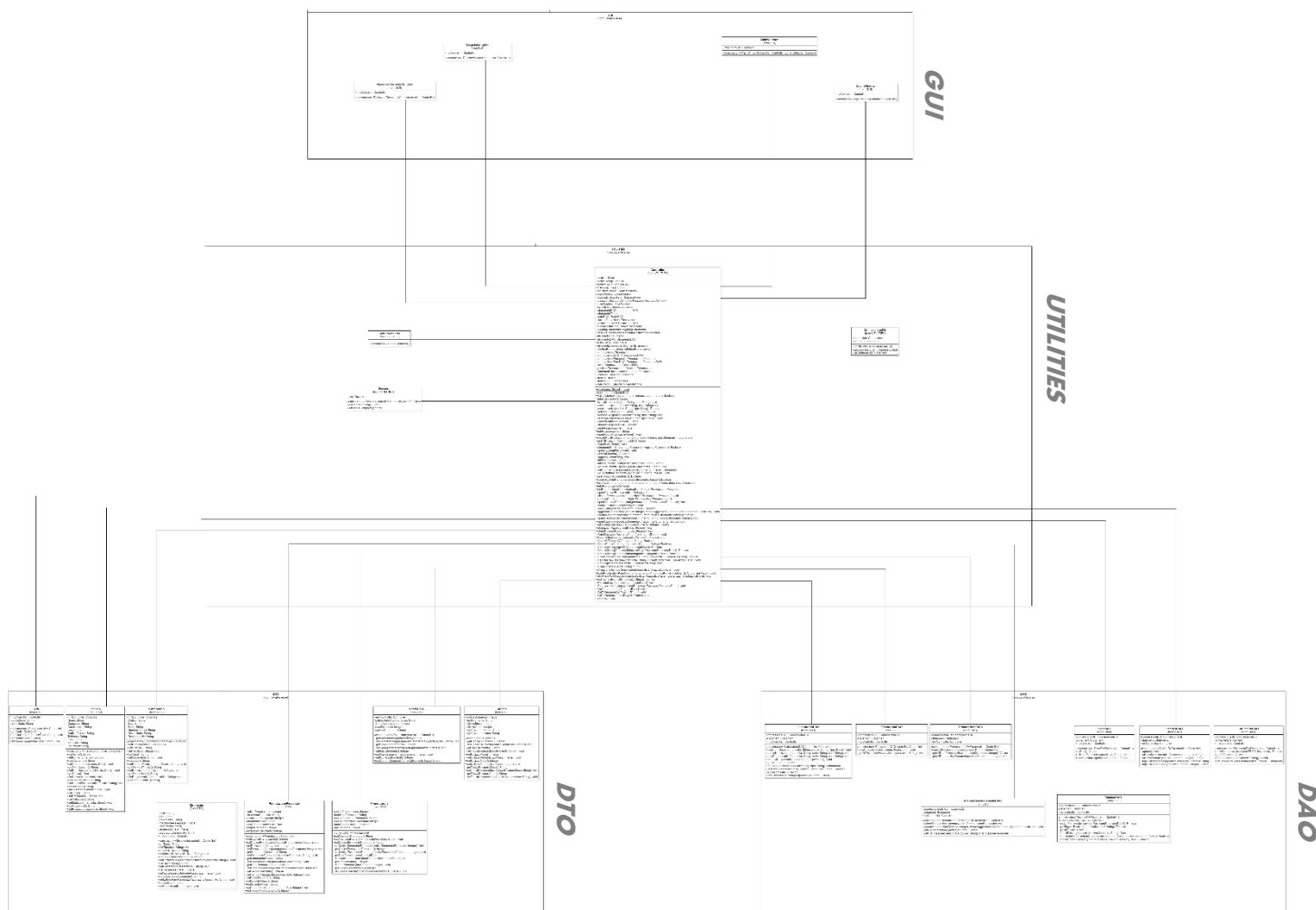
La classe [CenterRenderer](#) ridefinisce il metodo di allineamento delle caselle di una JTable (seppur estremamente piccola, è stata utilizzata ogni volta che abbiamo dovuto creare una nuova tabella; pertanto ci è sembrato utile definirla come classe).

Dopo aver effettuato il login, i dati del tecnico attualmente collegato alla piattaforma vengono inseriti in un oggetto *currentSession* della classe [Session](#), molto utile per poter attingere ai dati dell'utente in maniera semplice e diretta.

A collegare le numerose classi del progetto, comunicare tra di loro e definire metodi importanti per la gestione, controllo e modellazione dei dati, vi è la classe [Controller](#) (di cui definiamo all'interno delle diverse classi un oggetto *myController*). Il costruttore di quest'ultimo, prima di istanziare una pagina di login (che comparirà all'avvio del programma), richiamerà il metodo [update\(\)](#) della classe [StoricoDAO](#); tale metodo, esegue una *CALL* di una procedura definita sul Database, la quale controllerà che la data delle prenotazioni non sia passata rispetto a quella odierna; in tal caso, la prenotazione sarà conclusa e passerà nella tabella *STORICO*.

3 Class Diagram

Di seguito, verrà mostrato il Class Diagram delle classi che compongono il progetto; il diagramma è compreso di tutti gli attributi e metodi utilizzati, rispettandone la visibilità e i tipi. Anche il Class Diagram è suddiviso in packages, rispettando quelli usati all'interno del codice; si noti inoltre che, per quanto riguarda il package GUI, sono state inserite solo le classi dei quattro frame che compongono l'applicativo, in quanto, come espresso in precedenza, ogni componente è descritto in una classe a sé e sarebbero fin troppe da rappresentare, generando confusione nella lettura del diagramma. Si idealizza dunque che sia tutto identificato nelle quattro classi frame indicate, così da rendere l'idea di come è suddiviso il codice, tralasciando i dettagli dell'interfaccia grafica. Il Class Diagram è rappresentato come segue:

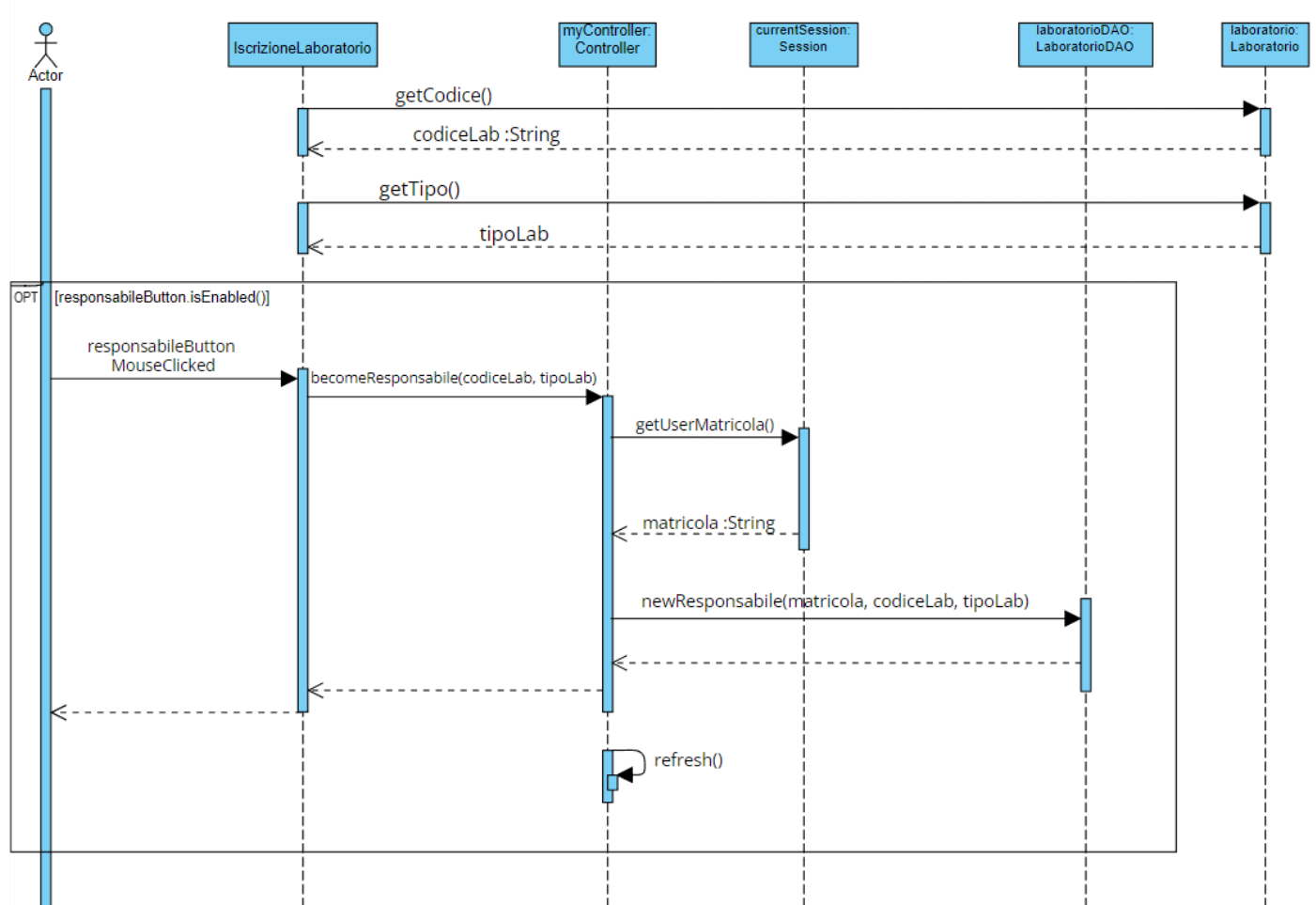


4 Sequence Diagram

Successivamente è riportato, come da consegna, il Sequence Diagram di due funzionalità a scelta; per la rappresentazione, abbiamo scelto di mostrare la funzione che consente ad un utente di diventare responsabile di laboratorio, e quella del recupero password. I due Sequence, sono descritti come segue

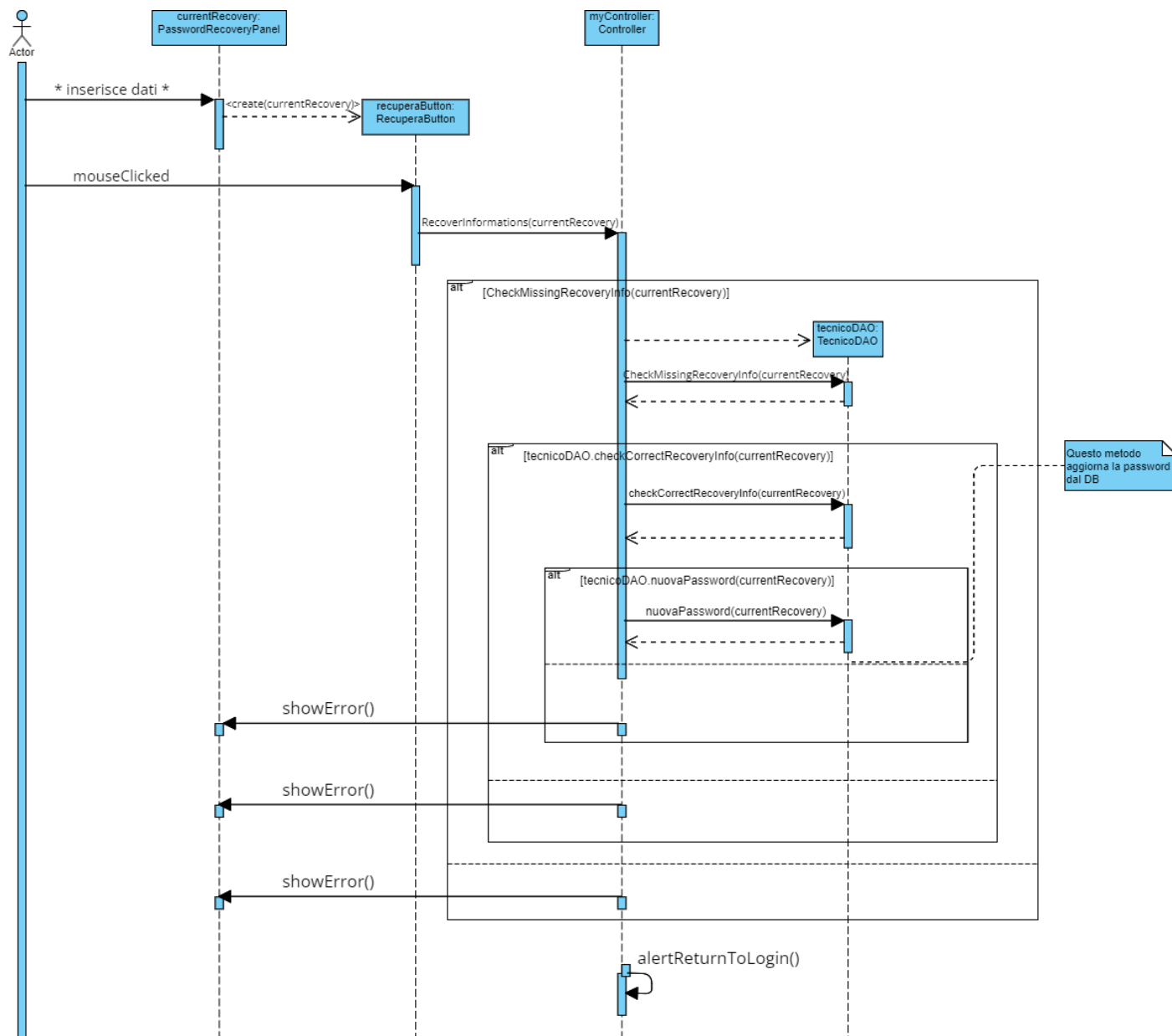
4.1 Sequence: Diventa Responsabile

Ad un utente semplice (tecnico) è concessa la possibilità di diventare responsabile di un laboratorio, a patto che sia iscritto al laboratorio in questione e che non sia già responsabile di altri; avere tale titolo consente di poter aggiungere nuove postazioni e inserire nuovi strumenti ad esse (inserendo per entrambe i parametri necessari, come il massimo numero di strumenti per una postazione, il numero di ore consecutive massimo per cui è possibile prenotarsi ad uno strumento e le caratteristiche di quest'ultimo). Le funzioni necessarie affinché tale funzionalità possa essere realizzabile, sono espresse nel seguente Sequence Diagram:



4.2 Sequence: Recupera Password

Se la propria password viene dimenticata, è possibile recuperarla mediante la funzionalità di recupero password descritta; dopo aver inserito i propri dati e aver premuto il tasto *recupera*, verrà inviata una richiesta al **Controller** il quale, dopo una verifica sulla completezza e correttezza dei dati inseriti, oltre a controllare se questi ultimi siano effettivamente associati all'account che si vuole recuperare, invia una richiesta alla classe **TecnicoDAO** per il cambio della password; dopodiché si verrà reindirizzati alla schermata di login.



5 CRC Cards

Di seguito, sono elencate le CRC Cards per i vari package del progetto; per quanto riguarda il package GUI, come espresso già in precedenza, saranno indicati solo i quattro frame che sostengono l'applicativo, ignorando le singole classi di componenti definite separatamente; tuttavia, sono indicati tra i collaboratori delle classi frame, i panels principali utilizzati al loro interno. Per la classe controller, non è possibile elencare tutti i collaboratori, o definire tutte le funzionalità che esso può effettuare (per ovvi motivi). Le Cards sono descritte come segue

5.1 CRC: UTILITIES

CLASS: Session	
RESPONSIBILITIES	COLLABORATORS
	Controller
Definitore di sessioni create dall'avvio del programma	Tecnico

CLASS: ConnessioneDB	
RESPONSIBILITIES	COLLABORATORS
Effettua la connessione al Database tramite JDBC	

CLASS: CenterRenderer	
RESPONSIBILITIES	COLLABORATORS
Definizione dell'allineamento di celle di una JTable	

CLASS: Controller	
RESPONSIBILITIES	COLLABORATORS
Contiene tutte i metodi necessari al programma	*

5.2 CRC: GUI

CLASS: MainWindow	
RESPONSIBILITIES	COLLABORATORS
	Controller
Consente di eseguire tutte le funzionalità una volta loggato	LeftMenu
	LaboratoriPage
	StrumentiPage
	PrenotazioniPage

CLASS: LoginWindow	
RESPONSIBILITIES	COLLABORATORS
	Controller
Consente di effettuare il login (dal LoginPanel)	LoginPanel
	LeftSidePanel

CLASS: RegisterWindow	
RESPONSIBILITIES	COLLABORATORS
	Controller
Consente di effettuare la registrazione (dai panel associati)	RegisterPanel (Anagrafica + Credenziali)
	LeftSidePanel

CLASS: PasswordRecoveryWindow	
RESPONSIBILITIES	COLLABORATORS
	Controller
Consente di recuperare la password associata all'account	LeftSidePanel
	PasswordRecoveryPanel

5.3 CRC: DTO e DAO

CLASS: Laboratorio	
RESPONSIBILITIES	COLLABORATORS
	Controller
Definizione di oggetti Laboratorio	

CLASS: LaboratorioDAO	
RESPONSIBILITIES	COLLABORATORS
	ConnessioneDB
Gestione dei Laboratori da parte di un tecnico	Controller

CLASS: Postazione	
RESPONSIBILITIES	COLLABORATORS
	Controller
Definizione di oggetti Postazione	

CLASS: PostazioneDAO	
RESPONSIBILITIES	COLLABORATORS
	ConnessioneDB
Gestione dei dati delle postazioni dal DB	Controller

CLASS: Prenotazione	
RESPONSIBILITIES	COLLABORATORS
	Controller
Definizione di oggetti Prenotazione	LocalDate

CLASS: PrenotazioneDAO	
RESPONSIBILITIES	COLLABORATORS
	ConnessioneDB
Gestione delle operazioni di prenotazioni sul DB	Controller

CLASS: PrenotazionePersonale	
RESPONSIBILITIES	COLLABORATORS
	Controller
Definizione oggetti PrenotazionePersonale	LocalDate

CLASS: PrenotazionePersonaleDAO	
RESPONSIBILITIES	COLLABORATORS
	ConnessioneDB
Gestione delle prenotazioni associate ad un singolo tecnico	Controller

CLASS: Sede	
RESPONSIBILITIES	COLLABORATORS
	Controller
Definizione di oggetti Sede	

CLASS: SedeDAO	
RESPONSIBILITIES	COLLABORATORS
	ConnessioneDB
Gestione dei i dati delle sedi tramite DB	Controller

CLASS: Storico	
RESPONSIBILITIES	COLLABORATORS
	Controller
Definizione di oggetti Storico	

CLASS: StoricoDAO	
RESPONSIBILITIES	COLLABORATORS
	ConnessioneDB
Gestione dei dati storici delle prenotazioni concluse dal DB	Controller

CLASS: Strumento	
RESPONSIBILITIES	COLLABORATORS
	Controller
Definizione di oggetti Strumento	

CLASS: StrumentoDAO	
RESPONSIBILITIES	COLLABORATORS
	ConnessioneDB
Gestione dei dati degli strumenti dal il DB	Controller

CLASS: Tecnico	
RESPONSIBILITIES	COLLABORATORS
	Controller
Definizione di oggetti Tecnico	

CLASS: TecnicoDAO	
RESPONSIBILITIES	COLLABORATORS
	ConnessioneDB
Gestione dei dati dei tecnici con il DB	Controller

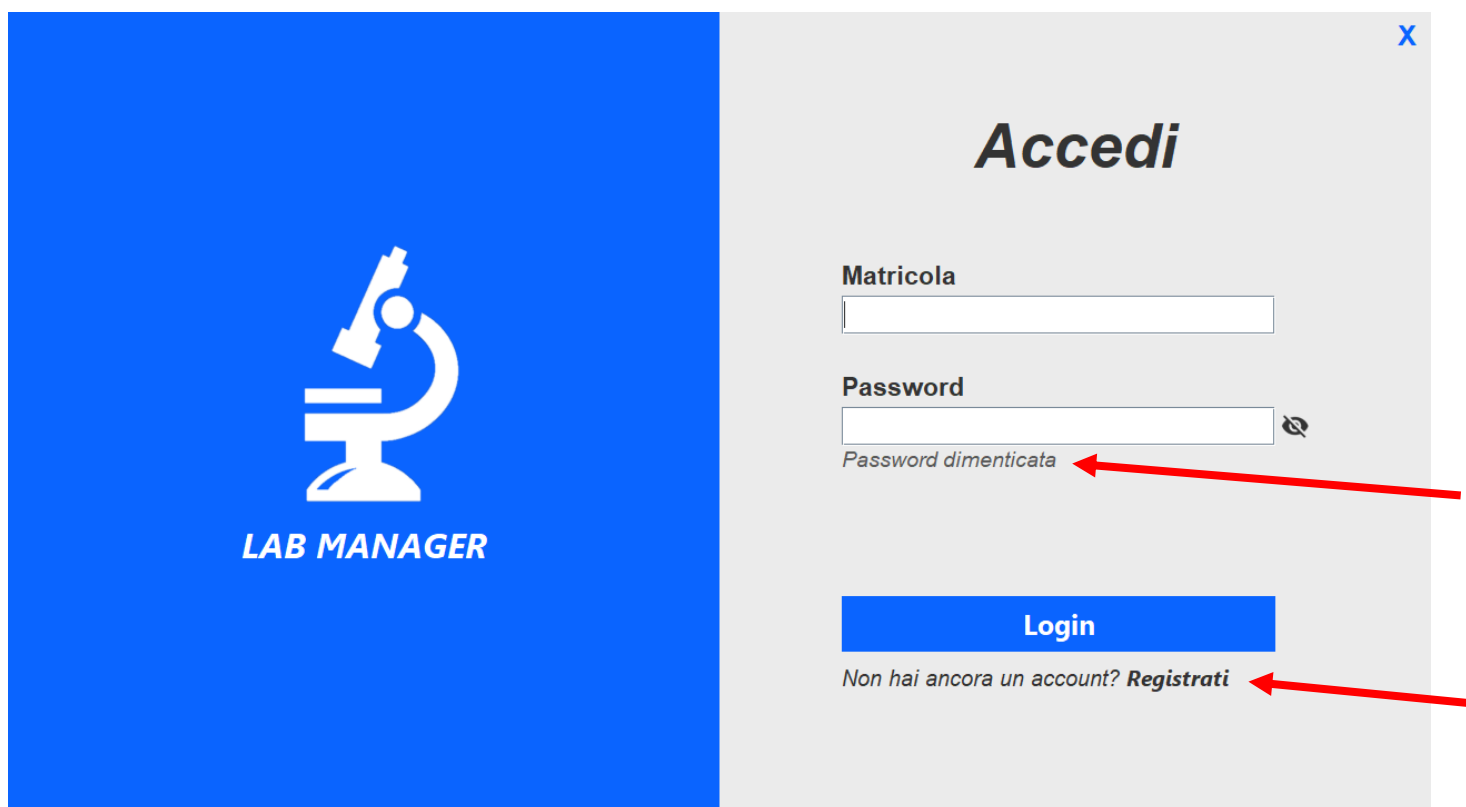
6 Breve esposizione

Dopo aver esposto il funzionamento generale dell'applicativo, sono riproposte di seguito alcune schermate per mostrare come il software si presenta all'utente, e dare un'idea dei pattern e della disposizione degli elementi che saranno presenti all'interno dell'applicativo. Si ricorda che il tutto è strutturato su quattro frame (i primi tre che, attraverso i Panel e componenti associati gestiscono login/register/recovery, e una MainWindow che sarà il cuore dell'applicazione); saranno tuttavia presenti numerose sottoclassi di JDialog da noi definite.

Di seguito, sono riportate alcune schermate dei quattro frame a disposizione.

6.1 Avvio del Software e Login

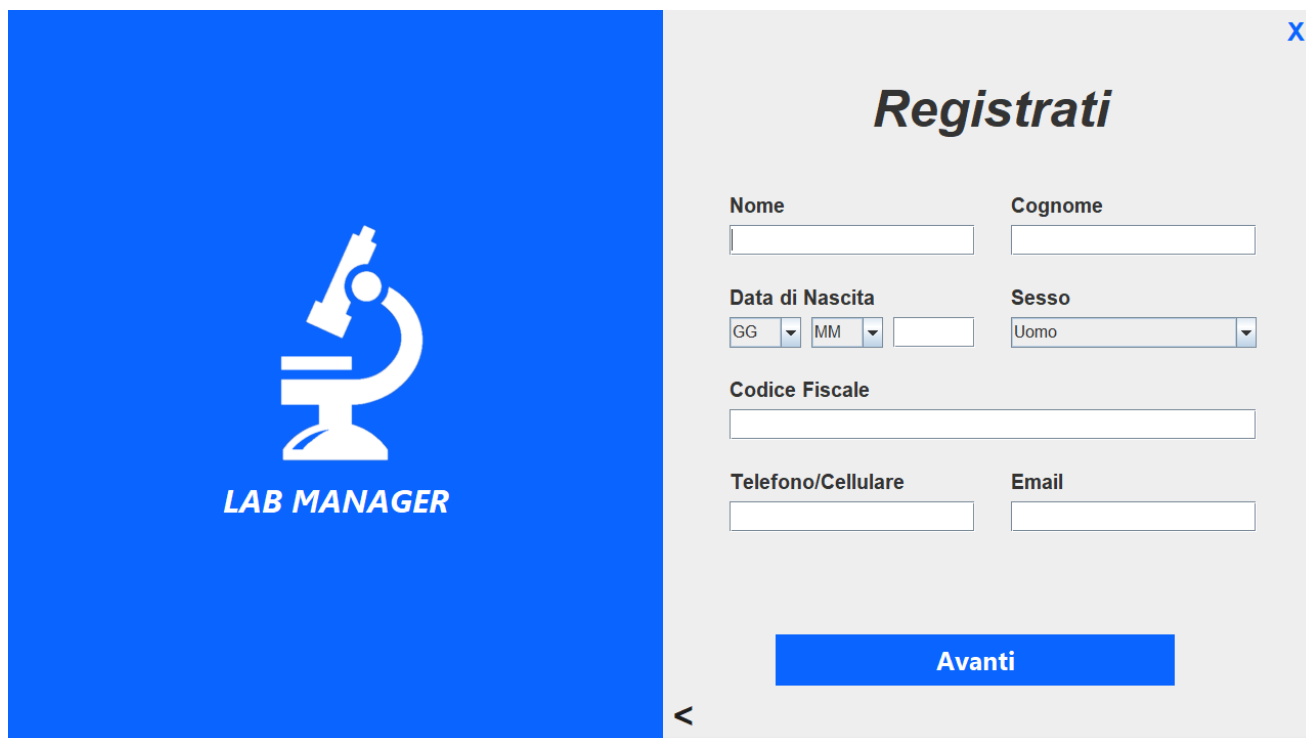
È possibile accedere inserendo le proprie credenziali e cliccando sul tasto Login. Altrimenti, ci si può registrare mediante il tasto Registrati in fondo alla pagina; se la password è stata dimenticata, si può accedere alla schermata di recupero password dall'apposita scritta.



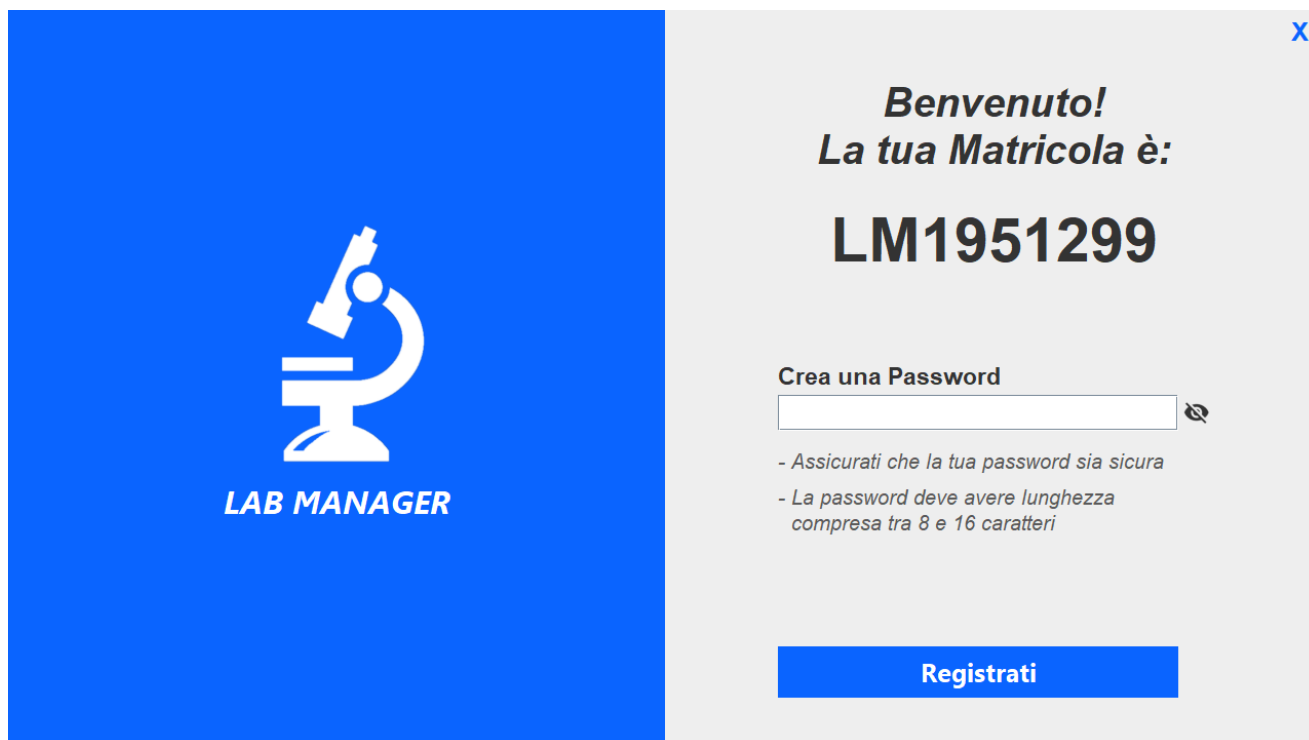
The screenshot displays the login interface of the LabManager application. It is divided into two main sections. The left section has a solid blue background and features a white icon of a microscope with a circular element at the eyepiece, with the text "LAB MANAGER" centered below it. The right section has a light gray background and is titled "Accedi" in a large, bold, italicized font. Below the title, there are two input fields: "Matricola" and "Password". The "Password" field includes a small eye icon for toggling visibility. Below the "Password" field, there is a link "Password dimenticata" with a red arrow pointing to it. Below the input fields is a blue "Login" button. At the bottom of the right section, there is a link "Non hai ancora un account? Registrati" with a red arrow pointing to it. A small blue "X" icon is visible in the top right corner of the right panel.

6.2 Registrazione

Una volta inseriti tutti i dati, verrà controllato che questi ultimi siano validi e, soprattutto, che non ci siano campi lasciati vuoti.

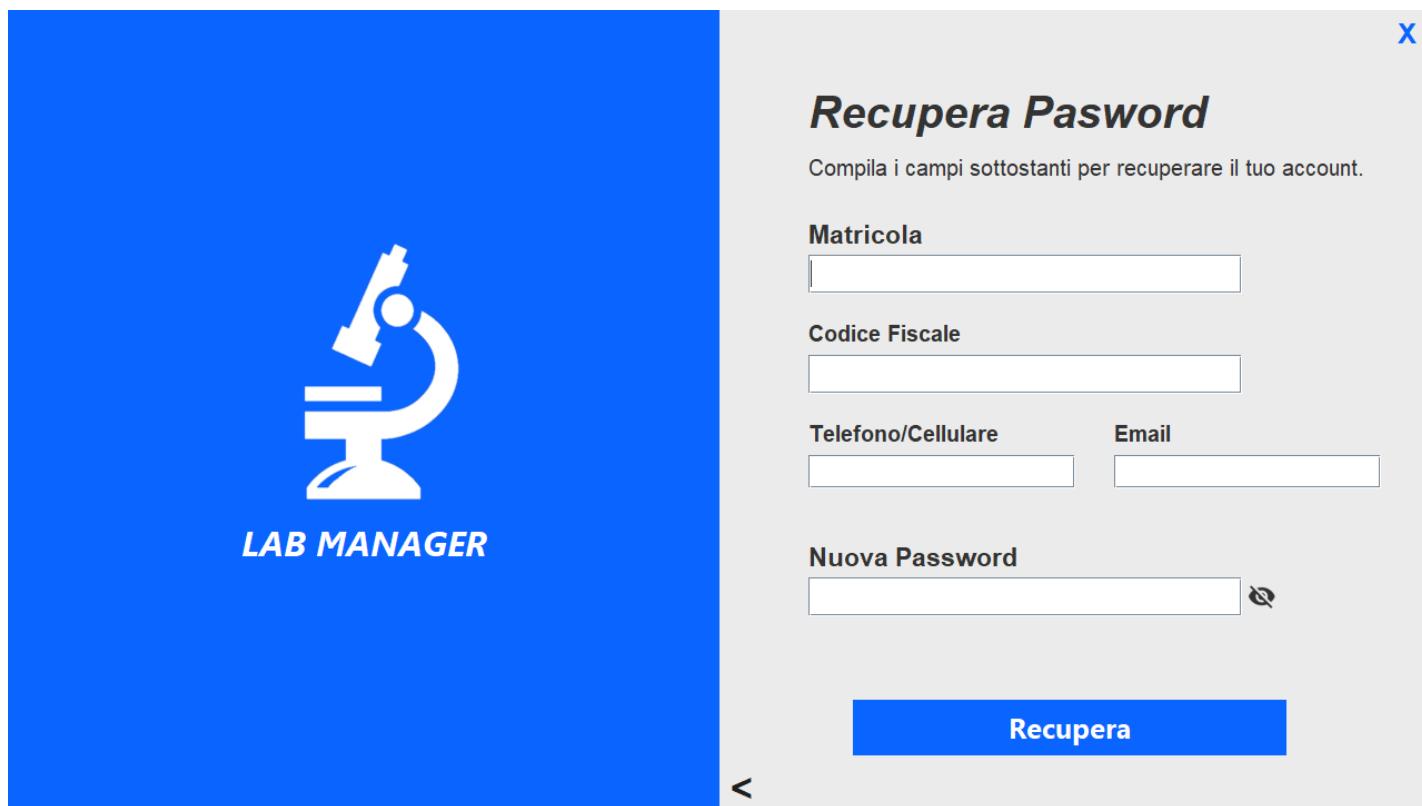


Una volta che i controlli sono stati effettuati, se andati a buon fine, si passerà all'assegnazione della matricola e all'inserimento di una password. Si noti che la matricola assegnata sarà sempre differente (l'identificativo è sempre composto da LM + una sequenza numerica).



6.3 Recupero Password

Se l'utente ha dimenticato la sua password di accesso, è possibile recuperarla tramite l'apposita schermata, inserendo tutti i campi richiesti e, dopo un adeguato controllo di questi ultimi, la nuova password viene aggiornata (come mostrato nella sezione 4.2).



Recupera Password


Compila i campi sottostanti per recuperare il tuo account.

Matricola

Codice Fiscale

Telefono/Cellulare **Email**

Nuova Password

Recupera

6.4 Schermata Generale

Da qui, sarà possibile accedere a tutte le funzionalità del Software, attraverso il comodo menu sulla sinistra, utilizzabile in qualsiasi momento e da qualsiasi schermata; dalla sezione **Laboratori**, si accede alla lista di tutti i laboratori attualmente disponibili, e consente ad un utente che è responsabile di un laboratorio di effettuare operazioni come la creazione di una nuova postazione o l'aggiunta di nuovi strumenti (a postazioni già esistenti). La sezione **Strumenti**, analogamente, riporta tutti gli strumenti attualmente registrati, ed offre la possibilità all'utente di effettuare una prenotazione (qualora egli fosse iscritto al laboratorio corrispondente a quello strumento). Infine, la sezione **Prenotazioni** mostra tutte le prenotazioni legate all'utente attualmente connesso, permettendogli di modificare o eliminare una prenotazione esistente, o di effettuarne di nuove. In basso a destra, in qualunque schermata della MainWindow è riportata la data attuale, in modo da consentire al tecnico che effettua una prenotazione di potersi regolare di conseguenza. La schermata principale è riportata di seguito.

