



LaureApp
Informatica e Tecnologie per la Produzione del Software
A.A 2022/2023

Realizzato da:

Clemente Diego 744404 ITPS d.clemente17@studenti.uniba.it

De Veteris Nicola 736332 ITPS n.deveteris@studenti.uniba.it

Lorusso Giuseppe 744407 ITPS g.lorusso127@studenti.uniba.it

Dipalma Pasquale 736346 ITPS p.dipalma4@studenti.uniba.it

Sommario

Cosa è LaureApp?	4
1. Scelte Progettuali	4
1.1. Navigazione	5
1.2. Database	5
1.3. Struttura Progetto	8
1.4. Liste	8
1.5. Servizi	8
1.6. Interfaccia	9
1.7. Tipologia di utenti	9
2. Descrizione funzionalità e tecnologie utilizzate	10
2.1. Registrazione Studente	10
2.2. Registrazione Docente	11
2.3. Login	12
2.4. Modifica Profilo	13
2.5. Registrazione Tesi	13
2.6. Condivisione delle informazioni pubbliche di una tesi	14
2.7. Generazione QR-code con informazioni della tesi	15
2.8. Modifica della tesi	16
2.9. Visualizzazione tesi di un docente	17
2.10. Visualizzazione di una richiesta	17
2.11. Visualizzazione delle tesi avviate	18
2.12. Aggiunta di un task	19
2.13. Visualizzazione dei task	20
2.14. Modifica stato task	21
2.15. Richiesta di un ricevimento	21
2.16. Visualizzazione e aggiunta ricevimenti	22
2.17. Visualizzazione e ricerca di una tesi - Ospite	23
2.18. Visualizzazione e ricerca di una tesi - Studente	23
2.19. Richiesta di una tesi	24
2.20. Comunicazione con il docente - Studente	25
2.21. Aggiunta di una tesi alla classifica	26
2.22. Visualizzazione della classifica	26
2.23. Visualizzazione della tesi in corso	27
2.24. Aggiunta di un co-relatore	27
3. Strumenti utilizzati	28

4. Progettazione dell'icona	28
5. Ruolo dei componenti	28
6. Limiti dell'app.....	29
7. Sviluppi futuri.....	29
8. Credenziali di accesso al sistema	29

Cosa è LaureApp?

LaureApp è un'applicazione sviluppata per dare supporto agli studenti nella scelta della tesi e ai professori nella gestione delle tesi e delle attività necessarie per la stesura di una tesi. Per quanto riguarda lo studente, l'app permette di visualizzare le tesi disponibili, richiedere una tesi e inoltre permette di richiedere dei ricevimenti con il docente. Il docente invece, ha la possibilità di rendere disponibili delle tesi, accettare le richieste di tesi, fissare dei ricevimenti e tenerne traccia, assegnare dei task allo studente per la stesura della tesi, caricare dei file inerenti ai task e monitorare lo stato di avanzamento delle tesi avviate. L'app prevede inoltre una modalità ospite, che permette di accedere al sistema senza creare un account, ma ha accesso solo alle funzionalità di ricerca, visualizzazione delle tesi e può contattare il docente tramite mail. Infine lo studente e il docente possono condividere le informazioni pubbliche di una tesi.

1. Scelte Progettuali

- Per ogni tesi è possibile aggiungere un task che può contenere uno o più file. Solo gli utenti autorizzati (docente che propone la tesi, tesisti e co-relatori) possono accedervi.
- La sezione “tesi avviate” contiene le tesi in fase di stesura. La sezione è stata progettata e realizzata in modo da permettere ai docenti di accedere in modo rapido a tutte le tesi avviate visualizzando il nome e il tesista. Per ognuno delle tesi vengono visualizzati i relativi tasks dando la possibilità di aggiungere nuovi task. Inoltre, cliccando su un task presente in questa sezione è possibile visualizzare la relativa descrizione, scadenza, stato e i file caricati; inoltre è possibile modificare lo stato.
- Quando un docente aggiunge una tesi essa viene automaticamente pubblicata e viene data la possibilità agli studenti di farne richiesta. Quando un docente riceve una richiesta e decide di accettarla, nella sezione “tesi avviate” sarà aggiunta la tesi per cui è appena stata accettata la richiesta.
- Il docente avrà la possibilità di aggiungere task, ricevimenti e co-relatori relativi ad una tesi solo nel momento in cui è stata accettata una richiesta per quella tesi.
- Un co-relatore viene aggiunto dal docente al momento della accettazione della tesi.
- Nella sessione relativa ai ricevimenti è stata progettata in modo da tener traccia di quelli che sono stati i ricevimenti. Durante l'aggiunta di un ricevimento è

infatti possibile specificare la data del ricevimento e selezionare i tasks affrontati durante lo stesso. I ricevimenti devono avere l'approvazione da parte del docente

- Lo studente ha la possibilità di aggiungere ogni tesi a una classifica, ordinandola in base alle proprie preferenze o ai vincoli specifici. Nella sezione profilo è possibile visualizzare le varie informazioni del profilo e modificarle, impostare il tema o effettuare il logout
- L'applicazione è stata progettata per funzionare in multilingua. L'app si adatta alla lingua di sistema così come il tema che è possibile modificarlo nella sezione del profilo (chiaro o scuro)

1.1. Navigazione

Le barre di navigazione utilizzate sono:

- AppBar
- BottomBar, per navigare tra le funzionalità principali

1.2. Database

L'app necessita di una connessione a Internet per funzionare correttamente, quindi i dati inerenti all'esecuzione dell'applicazione saranno scaricati da firebase a runtime. Per questi motivi l'utente deve essere connesso ad internet per poter utilizzare tutte le funzionalità dell'app.

Il database utilizzato è Cloud Firestore, un database NoSQL completamente gestito offerto da Firebase, il quale presenta una struttura di dati flessibile e scalabile. A differenza di un database relazionale tradizionale, Firestore organizza i dati in collezioni e documenti. Ogni documento è un insieme di campi chiave-valore e può contenere dati primitivi o strutture complesse come array e mappe.

Il database prende forma nel momento in cui sono inseriti dei dati da parte degli utenti. Ad esempio, se un utente si registra, qualora il database dovesse essere completamente vuoto, verrà creato in tempo reale il documento utente, con all'interno i dati e il tipo di utente appena registrato.

Uno dei vantaggi principali di Firestore è la sua capacità di sincronizzazione in tempo reale. Questo significa che è possibile impostare listener su documenti, collezioni o

query specifiche, e il database invierà automaticamente aggiornamenti ogni volta che i dati corrispondenti vengono modificati.

Struttura dei tipi di documenti presenti nel database:

DOCUMENTO UTENTE *con all'interno i dati e la tipologia dell'utente.*

laureapp-fa691	utente	10MN4vHMSjcm1FBLiApqWq1Xmeg1
+ Avvia raccolta	+ Aggiungi documento	+ Avvia raccolta
classifica	10MN4vHMSjcm1FBLiApqWq1Xmeg1 >	+ Aggiungi campo
corsi	DceS1qj1iwXuNNpkaUGTCem6W7b2	cognome: "Rossi"
ricevimenti	JozP1FZ3tMSMBtmxzkHYxnC5gN92	corso: "L-31 Informatica e tecnologie della produzione del software"
richiestatesi	Q3DgLeThbybYj5t0yF0TW80UrP93	dipartimento: "Informatica"
task	xucb24FOPuSb1Zba103z67gUrgK2	email: "studente2@gmail.com"
tesi	zqTzag4VLJe5yDxVzD5XUygTSgo2	matricola: "123457"
tesisti		nome: "Paolo"
utente >		tipo: "studente"

DOCUMENTO TESI *contiene i dati relativi alla tesi compresi eventuali vincoli.*

laureapp-fa691	tesi	3fpRJuThhX2jPwWVARpv
+ Avvia raccolta	+ Aggiungi documento	+ Avvia raccolta
classifica	3fpRJuThhX2jPwWVARpv >	+ Aggiungi campo
corsi	DB0SFJH37Cm81rXPXWd	corso: "LM-18 Computer science"
ricevimenti	DdmQcEcCo1i01HaJcANL	descrizione: "bb"
richiestatesi	JTqCtQXfQh7Es1GHRQGD	dipartimento: "Informatica"
task	OJcS1CN085qdTSgxt5U	docente: "JozPIFZ3tMSMBtmxzkHYxnC5gN92"
tesi	qY9qh3p8igx7H0caFBun	durata: "20"
tesisti		idTesi: "3fpRJuThhX2jPwWVARpv"
utente		media: "30"
		nome: "bb"

DOCUMENTO CLASSIFICA *contiene l'associazione di una tesi con un utente e la sua posizione relativa nella classifica.*

laureapp-fa691	classifica	2bgXnTk09Sn0ZdB5tM2c
+ Avvia raccolta	+ Aggiungi documento	+ Avvia raccolta
classifica >	2bgXnTk09Sn0ZdB5tM2c >	+ Aggiungi campo
corsi	5BgvoMLUxHh1WDZwqMuTH	durata: "150"
ricevimenti	WndytiL62sY7IwetSscK	idTesi: "DB0SFJH37Cm81rXPXWd"
richiestatesi	aQLqUs03L4rJzyWZWEv	media: "24"
task		nome: "Tesi 2"
tesi		pos: 1
tesisti		utente: "Q3DgLeThbybYj5t0yF0TW80UrP93"
utente		

DOCUMENTO CORSI *contiene le informazioni relativi al corso di studi.*

laureapp-fa691	corsi	0JfUwB0EEB87bc1d0nq6bX
+ Avvia raccolta	+ Aggiungi documento	+ Avvia raccolta
classifica	0JfUwB0EEB87bc1d0nq6bX >	+ Aggiungi campo
corsi	0JfUwB0EEB87bc1d0nq6bX	tipo: "Liceale"
ricevimenti		descrizione: "Politecnico"
richiestatesi		colore: "blu scuro"
task		
tesi		
tesisti		
utente		

DOCUMENTO TESISTI *contiene l'associazione tra tesi, relatore, studente e un possibile correlatore.*

laureapp-fa691	tesisti	QA00l1ucrEQ6RwjN1JNc
+ Avvia raccolta classifica corsi ricevimenti richiestatesi task tesi tesisti utente	+ Aggiungi documento QA00l1ucrEQ6RwjN1JNc SHUx8mmhv0Dc7HH2iD3p	+ Avvia raccolta + Aggiungi campo correlatore: null relatore: "JozPIFZ3tMSMBtmxzkHYxnC5gN92" studente: "DceS1qjliwXuNNpkaUGTCem6W7b2" tesi: "3fpRJJuThhX2jPwWVAPrv"

DOCUMENTO RICEVIMENTI *contiene le informazioni relative ad un ricevimento, compreso il suo stato e utente che ne ha fatto richiesta.*

laureapp-fa691	ricevimenti	2FKg4GzOMOR3dAgMWy5D
+ Avvia raccolta classifica corsi ricevimenti richiestatesi task tesi tesisti utente	+ Aggiungi documento 2FKg4GzOMOR3dAgMWy5D umBYMcByN2zqrBFeIZNd	+ Avvia raccolta + Aggiungi campo data: "07/08/2023" dettagli: "Richiesta ricevimento per task 1" stato: "Inviata" task: "Task 1" tesista: "SHUx8mmhv0Dc7HH2iD3p"

DOCUMENTO RICHIESTATESI *contiene le informazioni relative a una richiesta di tesi.*

laureapp-fa691	richiestatesi	N2d4dmk6S0x3M2onUsfh
+ Avvia raccolta classifica corsi ricevimenti richiestatesi task tesi tesisti utente	+ Aggiungi documento N2d4dmk6S0x3M2onUsfh	+ Avvia raccolta + Aggiungi campo descrizione: "Richiesta tesi" docente: "zgTZag4VLJe5yDxVzD5XUygTSgo2" studente: "1OMN4vHMSjcmIFBLiApqWq1Xmeg1" tesi: "DdmQcEcCoii01HaJcANL"

DOCUMENTO TASK *contiene le informazioni relative a un compito assegnato, compreso lo stato, scadenza e relativo utente che lo deve svolgere.*

laureapp-fa691	task	6PajpmZBIyd1XuovQ1sq
+ Avvia raccolta classifica corsi ricevimenti richiestatesi task tesi tesisti utente	+ Aggiungi documento 6PajpmZBIyd1XuovQ1sq CBxHkFBjIIXW0wPUh1q7 fL6Y5rg1SHJm6JbRYdso gSUP1C2Ica5EwTntrArx kKixYxdORsrBpgzNcFBw p1dDUtuaRwKeUBQR6UFd	+ Avvia raccolta + Aggiungi campo descrizione: "sss" file nome: "aa" scadenza: "24/08/2023" stato: "Iniziato" tesista: "QA00l1ucrEQ6RwjN1JNc"

1.3. Struttura Progetto

L'app è stata realizzata all'interno di un unico modulo. I package hanno la seguente struttura:

- Modelli
all'interno presenta delle classi e dei package dedicati alla gestione dei dati relativi alle diverse strutture e agli adapter. I package sono:
 - adapterFile, adapter per i file presenti nei task
 - adapterRicevimenti, adapter per i ricevimenti disponibili
 - adapterTask, adapter per i task (lato docente e lato studente)
 - compare, compare relativi all'ordinamento della classifica
 - menuHome, struttura della card con relativo adapter
- Registrazione
contiene le funzionalità di Login e Registrazione
- Utenti
 - Docenti
all'interno presenta delle classi che si occupano della gestione delle funzionalità corrispondenti al docente.
 - Ospite
all'interno presenta delle classi che si occupano della gestione delle funzionalità corrispondenti all'ospite.
 - Studenti
all'interno presenta delle classi che si occupano della gestione delle funzionalità corrispondenti allo studente.

1.4. Liste

La maggior parte delle sezioni è costituita da RecyclerView utile per gestire il riciclo dei componenti durante lo scrolling e diverse modalità di visualizzazione del layout degli elementi.

1.5. Servizi

Nel progetto è stato utilizzato Firebase come servizio per:

- per l'autenticazione, utilizzando nello specifico il servizio Firebase Authenticator
- per il salvataggio dei file per i progetti di tesi caricati, utilizzando nello specifico Firebase Storage

1.6. Interfaccia

Tutte le interfacce sono state realizzate manualmente seguendo i principi del Material Design.

I colori utilizzati sono il bianco, grigio chiaro, l'azzurro e il nero.

Il bianco e il grigio chiaro vengono utilizzati come sfondo e BottomBar, mentre l'azzurro per l'ActionBar e il nero per scritte ed icone.

1.7. Tipologia di utenti

L'app prevede due tipologie di utenti:

- Studente
- Docente
- Ospite

Questo ci ha permesso di sviluppare le funzionalità previste per le due tipologie di utenti separatamente e in modo più efficiente.

2. Descrizione funzionalità e tecnologie utilizzate

Sono di seguito descritte le principali funzionalità dell'app.

2.1. Registrazione Studente

Lo studente non registrato deve inserire email, password, matricola, nome, cognome, il dipartimento e il corso che frequenta (i corsi visualizzati verranno filtrati in base al dipartimento selezionato in precedenza) per poter accedere al sistema.

The image displays three sequential screenshots of the 'Registrazione Studente' (Student Registration) form, illustrating the user interface and its interactive elements.

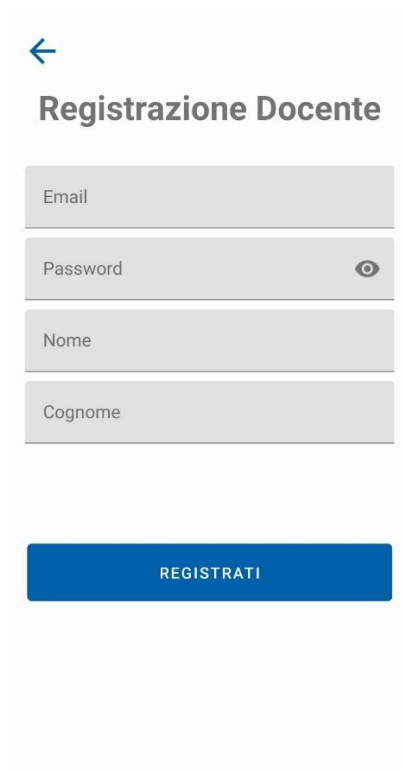
Screenshot 1 (Left): Shows the initial form with input fields for Email, Password (with a toggle icon), Matricola, Nome, and Cognome. Below these are dropdown menus for Dipartimento and Corso. A blue button labeled 'REGISTRATI' is at the bottom.

Screenshot 2 (Middle): Shows the Dipartimento dropdown menu open, displaying a list of options: Policlinico, Informatica, and Lettere. The other fields remain unchanged.

Screenshot 3 (Right): Shows the Corso dropdown menu open, displaying a list of filtered course options: LM-18 Computer science, L-31 Informatica, and L-31 Informatica e tecnologie della pr.. The Dipartimento dropdown is now closed and shows 'Informatica' as the selected value.

2.2. Registrazione Docente

Il docente non registrato deve inserire email, password, nome e cognome per poter accedere al sistema.

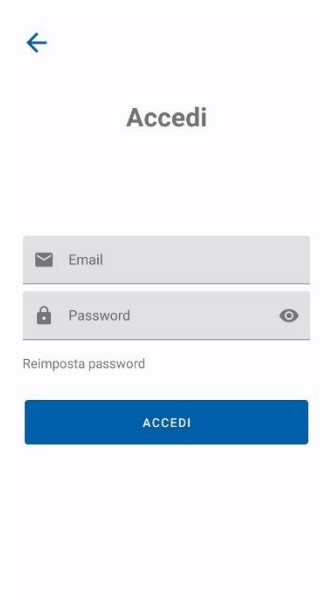


A mobile application registration form for a teacher. The form is displayed on a light blue background. At the top left, there is a blue back arrow icon. Below it, the title "Registrazione Docente" is written in bold black text. The form consists of four input fields stacked vertically: "Email", "Password", "Nome", and "Cognome". The "Password" field has a small eye icon to its right, indicating a toggle for password visibility. Below the input fields, there is a blue button with the text "REGISTRATI" in white capital letters.

2.3. Login

L'utente già registrato deve inserire email e password per accedere al sistema.

Se un utente non dovesse ricordare la password e possibile cliccare su "Reimposta password" attraverso il servizio di reimpostazione della password di Firebase Authentication, è possibile inviare un'email per consentirgli di reimpostarla utilizzando la funzione `sendPasswordResetEmail(email)`.



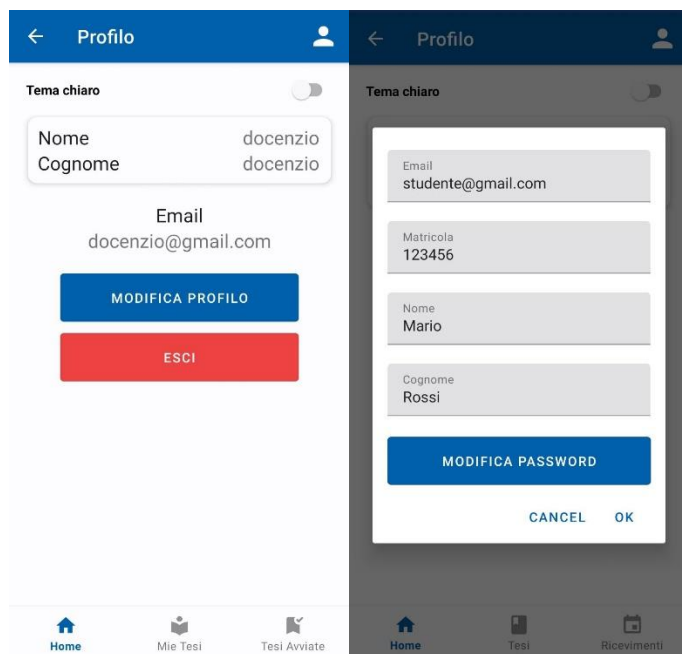
La pagina di login è uguale sia per lo studente sia per il docente, infatti una volta verificate le credenziali con firebase authentication mediante la funzione `signInWithEmailAndPassword(email, password)` viene chiamata la funzione `checkUserType()` che verifica il tipo di utente ed avvia la relativa activity

```
private void checkUserType() {
    FirebaseUser user = mAuth.getCurrentUser();
    String userId = user.getId();

    FirebaseFirestore firestore = FirebaseFirestore.getInstance();
    firestore.collection("utente").document(userId).get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
        @Override
        public void onSuccess(DocumentSnapshot documentSnapshot) {
            String role = documentSnapshot.getString("tipo");
            if (role.equals("studente")) {
                Toast.makeText(Login.this, "Autenticazione come studente.",
                    Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(Login.this, HomeStudente.class);
                intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP | Intent.FLAG_ACTIVITY_NEW_TASK);
                startActivity(intent);
                finishAffinity();
            } else if (role.equals("docente")) {
                Toast.makeText(Login.this, "Autenticazione come docente.",
                    Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(Login.this, HomeDocente.class);
                intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP | Intent.FLAG_ACTIVITY_NEW_TASK);
                startActivity(intent);
                finishAffinity();
            }
        }
    });
}
```

2.4. Modifica Profilo

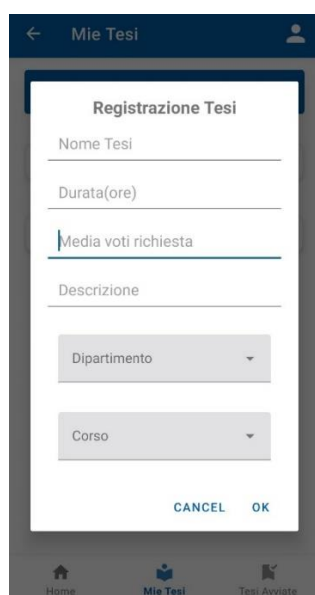
Lo studente e il docente registrati possono modificare le informazioni del proprio profilo. La modifica dei campi avviene mediante un DialogFragment contenente i solo i campi modificabili.



2.5. Registrazione Tesi

Il docente registrato può registrare una nuova tesi inserendo il nome, la durata (in ore), la media dei voti dello studente, una descrizione; Il dipartimento e il corso di laurea (opzionali).

I vincoli relativi alla tesi sono la media voti e la durata, e questi non devono essere necessariamente rispettati.



2.6. Condivisione delle informazioni pubbliche di una tesi

È possibile condividere le informazioni pubbliche di una tesi, cliccando il pulsante apposito, tramite WhatsApp, Gmail, Messaggi ecc.

Per far questo viene avviato un intent per le app di condivisione, che mostra all'utente un elenco di app tra cui scegliere per la condivisione e il messaggio sarà formato da un'immagine (codice qr) e delle informazioni della tesi.



2.7. Generazione QR-code con informazioni della tesi

Per ogni tesi, viene generato automaticamente, alla creazione della tesi, un codice QR. Scansionandolo verranno mostrate le informazioni sulla tesi.

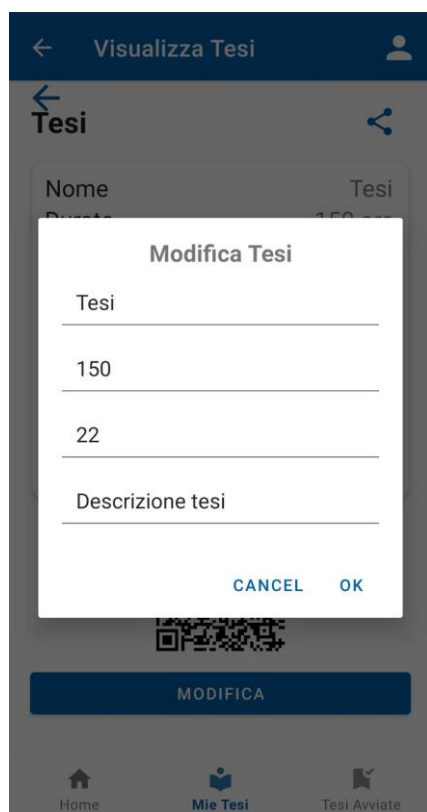


Per generare il qr vengono prese le informazioni relative alla tesi, e vengono codifica in un codice QR utilizzando la libreria ZXing utilizzando MultiFormatWriter (il testo viene codificato in una matrice di bit) e BarcodeEncoder (utilizzato per convertire la matrice di bit in un oggetto Bitmap), e quindi visualizza il codice QR generato all'interno di un'ImageView nell'interfaccia dell'app Android.

```
String text = "Nome": " + nome + " "+"Corso": " + corso + " "+"Media Richiesta": " + media + " "+"  
MultiFormatWriter multiFormatWriter = new MultiFormatWriter();  
try {  
    BitMatrix matrix = multiFormatWriter.encode(text, BarcodeFormat.QR_CODE, width: 400, height: 400);  
    BarcodeEncoder encoder = new BarcodeEncoder();  
    bitmap = encoder.createBitmap(matrix);  
    imageQr.setImageBitmap(bitmap);  
} catch (WriterException e) {  
    throw new RuntimeException(e);  
}
```

2.8. Modifica della tesi

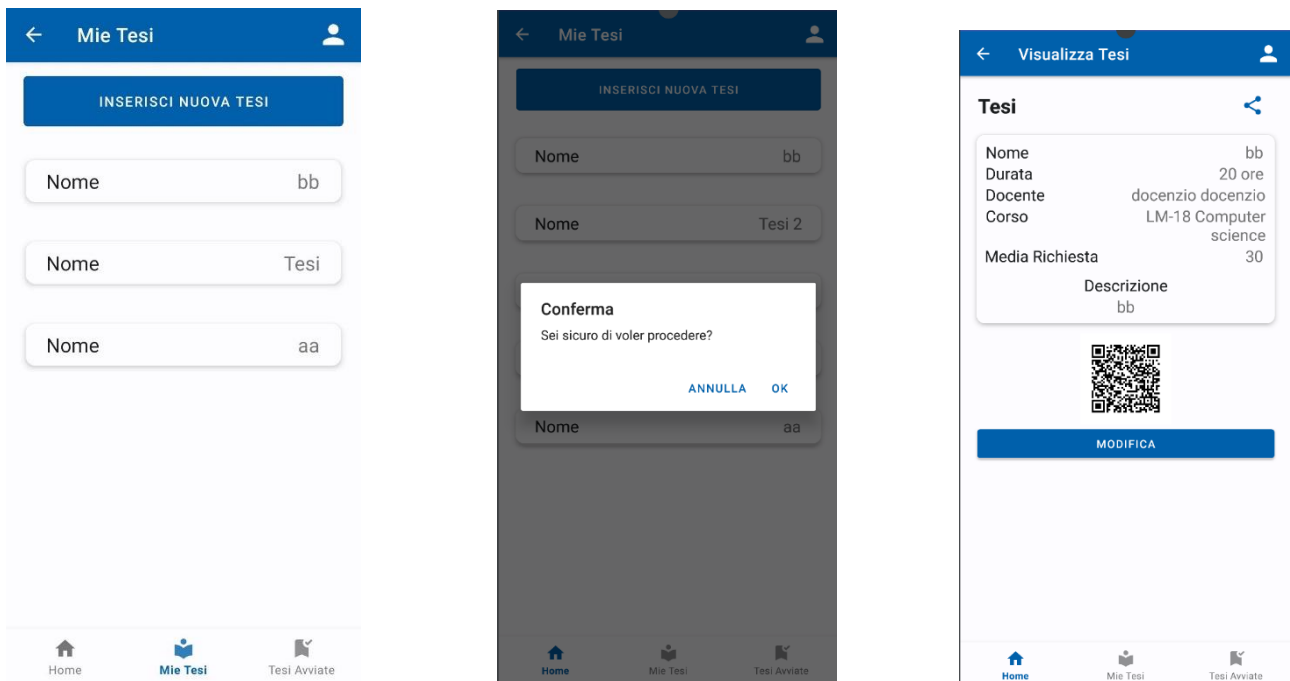
Il docente può modificare le informazioni di una tesi.



2.9. Visualizzazione tesi di un docente

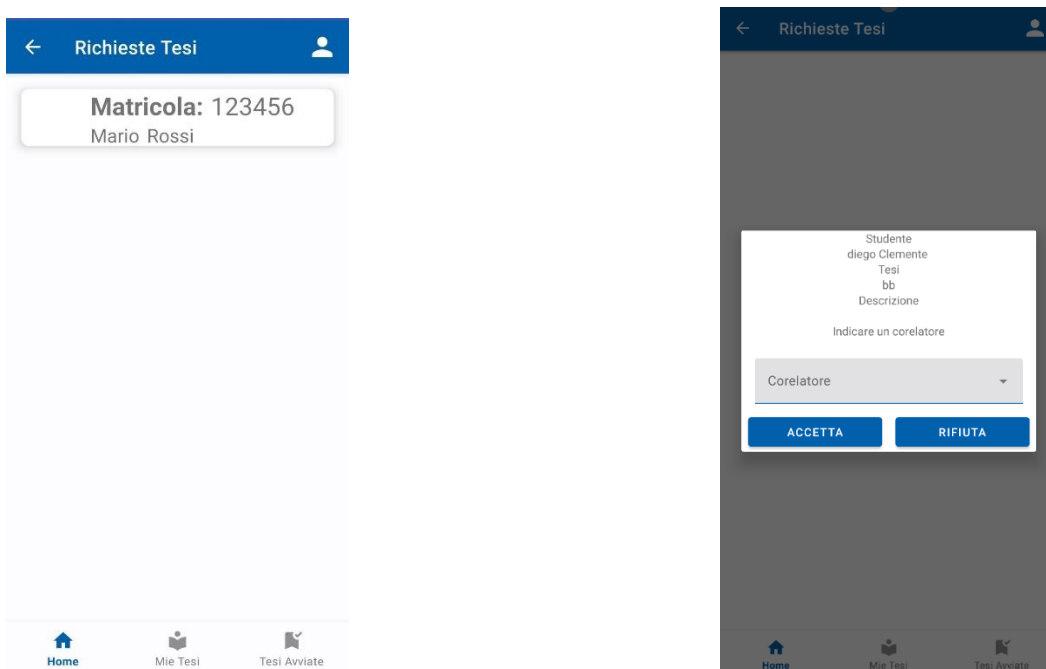
Il docente può visualizzare le tesi che ha inserito nel sistema.

Se clicca su una di essa verrà caricato un fragment che mostrerà la tesi, mentre se i terrà premuto la potrà cancellare (mostrerà un popup di conferma). Queste operazioni vengono gestite mediante le funzioni `onItemLongClick(int position)` e `onItemClick(int position)`.



2.10. Visualizzazione di una richiesta

Il docente può visualizzare le richieste inviate per le sue tesi e può decidere se accettare o rifiutare la richiesta mediante un `DialogFragment`.



2.11. Visualizzazione delle tesi avviate

Il docente può visualizzare le tesi in corso di stesura con le informazioni degli studenti a cui sono state assegnate le tesi



2.12. Aggiunta di un task

Il docente per le tesi in corso può aggiungere un task che il tesista deve svolgere e può caricare dei file relativi al task. I task devono avere una data di scadenza e uno stato (non iniziato, in corso, terminato).

← Task tesi

Nuovo taskTesi

Nome
Task 1

Progettazione della soluzione

Scadenza:

Seleziona data
10/08/2023

CARICA FILE

AGGIUNGI COMPITO

Home Mie Tesi Tesi Arviate

Per il caricamento di un file abbiamo bisogno di controllare richiedere i permessi relativi all'autorizzazione per accedere ai file e ai documenti sul dispositivo dell'utente. Mediante l'utilizzo di `private OpenMultipleDocuments()` consente all'utente di selezionare più file, evita duplicati nell'elenco degli URI dei file selezionati e aggiorna l'interfaccia con il numero di file selezionati.



```
private ActivityResultLauncher<String[]> filePickerLauncher =
    registerForActivityResult(new ActivityResultContracts.OpenMultipleDocuments(), uris -> {
        if (uris != null && uris.size() > 0) {
            if (!fileUris.addAll(uris)) {
                Toast.makeText(getContext(), "Il file selezionato è già presente", Toast.
            }
            Log.d("a", fileUris.toString());
            String testo = "File selezionati: " + fileUris.size();
            buttonScegliFile.setText(testo);
            updateSelectedFilesList();
        }
    });

1 usage 1 DiegoCle1
private void scegliFile() {
    filePickerLauncher.launch(new String[] {"/"});
}
```

2.13. Visualizzazione dei task




Docente e studente possono visualizzare i task assegnati per il completamento della tesi

Lato docente

 Task tesi 

INSERISCI NUOVO TASKTESI

Nome Task	Task 1
Scadenza	10/08/2023
Stato	Iniziato

 Home  Mie Tesi  Tesi Avviate

Lato studente

 Compiti Tesi 

Tesi

Nome	Tesi 2
Durata	150 ore
Iniziata	Iniziata
Descrizione	
Descrizione tesi	

Compiti Assegnati

Nome Task	Task 1
Scadenza	10/08/2023
Stato	Iniziato

Compiti Svolti

 Home  Tesi  Ricevimenti

2.14. Modifica stato task

Lo studente e il docente possono modificare lo stato del task impostandolo come completato o iniziato

The screenshot shows the 'Compiti Tesi' screen. At the top, there is a back arrow and a user icon. The main content area displays the following information: 'Nome Task 1', 'Scadenza 10/08/2023', 'Descrizione Progettazione della soluzione', and 'Stato Iniziato'. Below this information are two blue buttons: 'IMPOSTA COME COMPLETATO' and 'IMPOSTA COME INIZIATO'. At the bottom, there is a navigation bar with three icons: 'Home', 'Tesi', and 'Ricevimenti'.

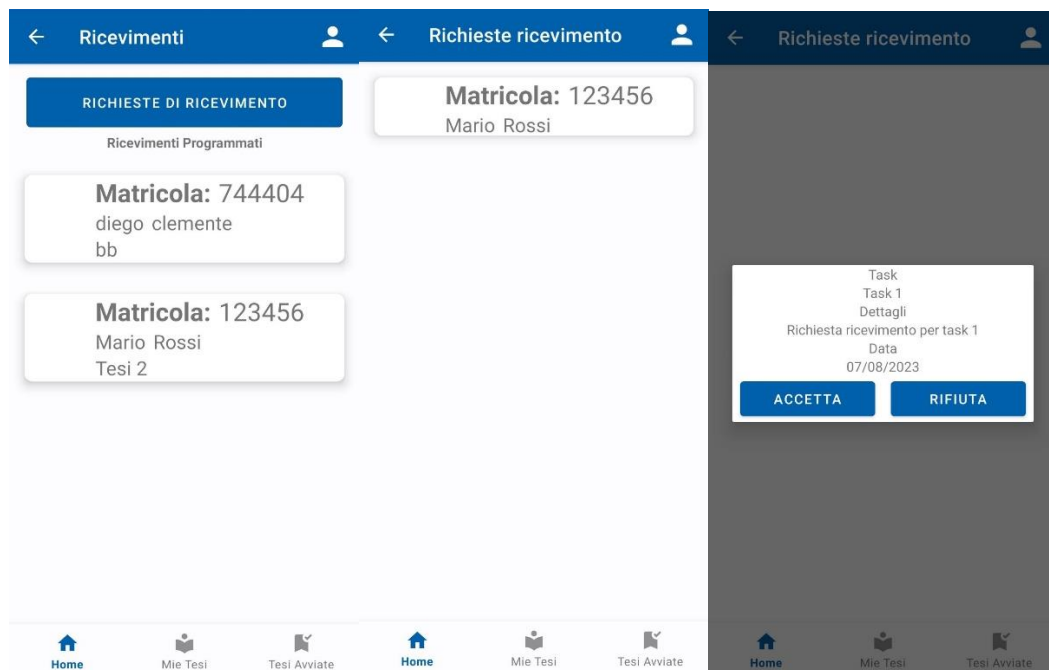
2.15. Richiesta di un ricevimento

Lo studente può richiedere un ricevimento associandolo ad un task, inserendo un possibile giorno di ricevimento e una descrizione. Nella pagina ricevimenti relativa allo studente e' possibile visualizzare tutte le richieste inviate e il loro relativo stato.

The left screenshot shows the 'Ricevimenti' screen with a 'RICHIEDI RICEVIMENTO' button at the top. Below it is a form titled 'Nuovo ricevimento'. The form has a dropdown menu for 'Argomento' with 'Task 1' selected, a text input field containing 'Richiesta ricevimento task 1', a 'Data:' label, a date picker showing '07/08/2023', and an 'INVIA RICHIESTA' button at the bottom. The right screenshot shows the 'Ricevimenti' screen with a 'RICHIEDI RICEVIMENTO' button at the top. Below it is a list of received requests. The first request is shown with the following details: 'Data 07/08/2023', 'Task di riferimento Task 1', 'Stato Inviata', and 'Dettagli Richiesta ricevimento per task 1'. At the bottom, there is a navigation bar with three icons: 'Home', 'Tesi', and 'Ricevimenti'.

2.16. Visualizzazione e aggiunta ricevimenti

Il docente può visualizzare i ricevimenti programmati e accettare una richiesta di ricevimento, effettuata dallo studente, associata ad un task.



2.17. Visualizzazione e ricerca di una tesi - Ospite

L'ospite può visualizzare e cercare una tesi senza dover effettuare la registrazione e l'accesso al sistema. Una volta visualizzata la tesi potrà condividerla o contattare il docente

←

Q Cerca qui

Welcome

REGISTRAZIONE STUDENTE

REGISTRAZIONE DOCENTE

ACCEDI

Visualizza tesi

Nome bb

Nome Tesi 2

Nome Tesi

Nome cc

Nome aa

2.18. Visualizzazione e ricerca di una tesi - Studente

Lo studente può visualizzare e cercare una tesi relativa al suo corso, mediante l'utilizzo di una recyclerview popolata dalle tesi relative al dipartimento dello studente.

← Ricerca Tesi

Q Cerca qui

Nome bb

Nome Tesi 2

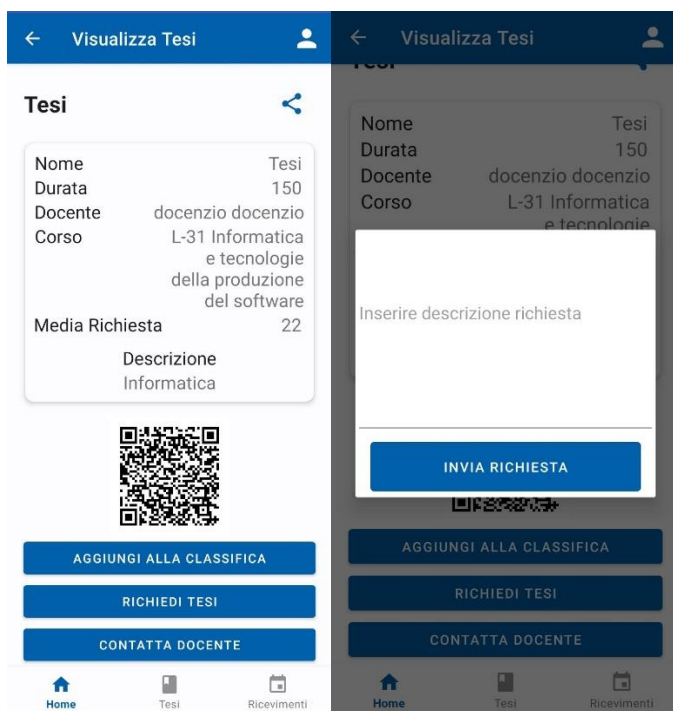
Nome Tesi

Nome cc

Nome aa

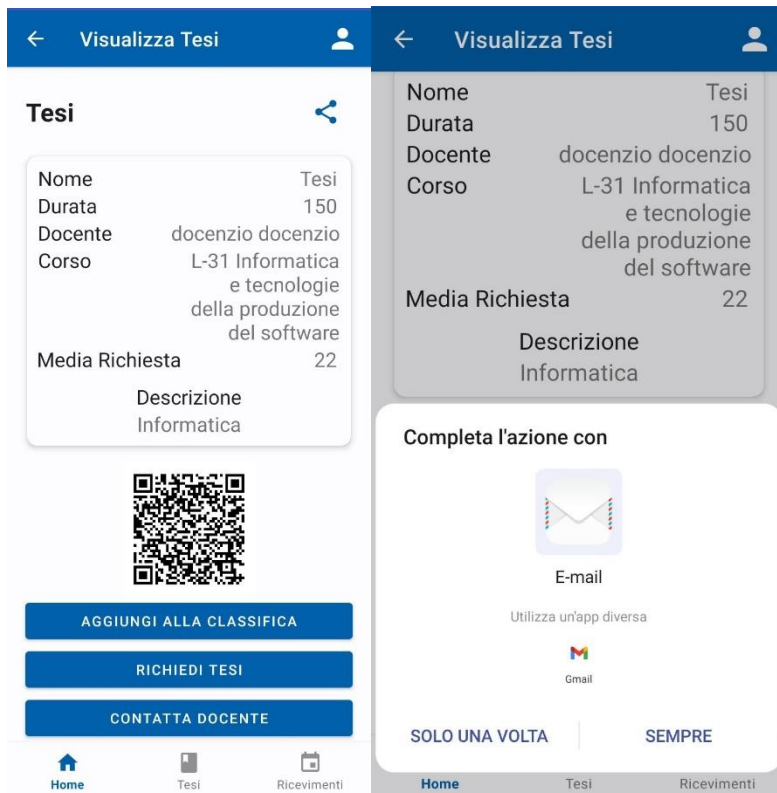
2.19. Richiesta di una tesi

Lo studente può effettuare la richiesta di una tesi. Inserendo nella descrizione eventuali vincoli che non rispetta



2.20. Comunicazione con il docente - Studente

Lo studente può comunicare con il docente tramite email.



Per far questo Viene creato un' intent per inviare un'email utilizzando l'azione `Intent.ACTION_SENDTO`. Viene impostato l'indirizzo email del docente come destinatario (mediante `Intent.EXTRA_EMAIL`) e viene definito un oggetto per l'email (`Intent.EXTRA_SUBJECT`), che include il nome della tesi. Viene verificato se c'è un'app di gestione email installata nel dispositivo utilizzando `resolveActivity()`. Se c'è un'app email installata, l'intento viene avviato per consentire all'utente di comporre e inviare l'email. Se nessuna app email è installata, viene mostrato un messaggio di avviso.

```
contattaButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        database.collection("utente").document(docente).get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
            @Override
            public void onSuccess(DocumentSnapshot documentSnapshot) {
                String docentemail = documentSnapshot.getString("email");

                Intent intent = new Intent(Intent.ACTION_SENDTO);
                intent.setData(Uri.parse("mailto:"));
                Log.d("email", docentemail);
                intent.putExtra(Intent.EXTRA_EMAIL, new String[]{docentemail});
                intent.putExtra(Intent.EXTRA_SUBJECT, "Informazioni riguardo tesi: "+nome);

                if(intent.resolveActivity(getPackageManager()) != null){
                    startActivity(intent);
                }else{
                    Toast.makeText(getApplicationContext(), "Nessuna app email installata", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
});
```

2.21. Aggiunta di una tesi alla classifica

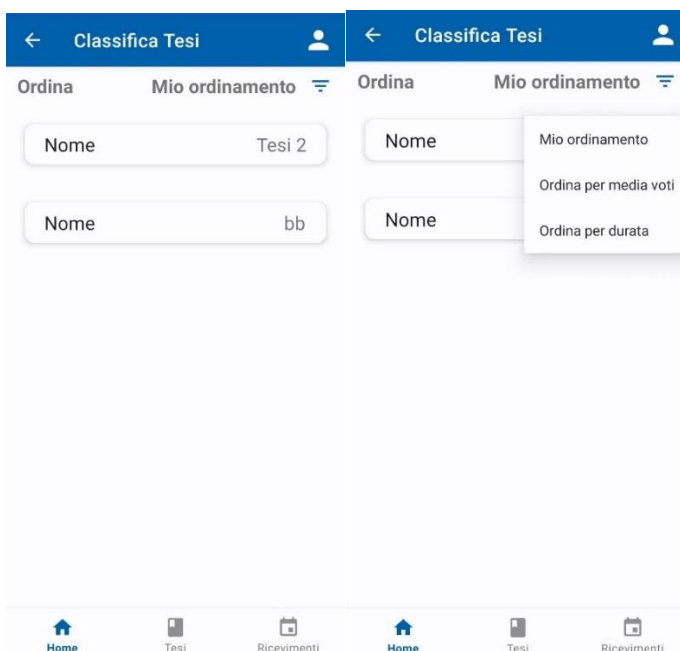
Lo studente può aggiungere una tesi alla classifica delle sue tesi preferite.



2.22. Visualizzazione della classifica

Lo studente può visualizzare la classifica delle tesi preferite, potendole ordinare manualmente mediante la funzione `onSwiped(@NonNull RecyclerView.ViewHolder viewHolder, int direction)`, per media dei voti o per durata (in ore).

Se clicca sulla tesi verranno mostrate la tesi mentre se tiene premuto verrà cancellata dalla classifica



2.23. Visualizzazione della tesi in corso

Lo studente può visualizzare le informazioni e i task relative alla sua tesi.

The screenshot shows the 'Compiti Tesi' (Thesis Tasks) screen. At the top, there is a blue header with a back arrow, the title 'Compiti Tesi', and a user profile icon. Below the header, the screen is divided into three main sections: 'Tesi', 'Compiti Assegnati', and 'Compiti Svolti'. The 'Tesi' section contains a card with the following information: 'Nome' (Nome), 'Durata' (Durata), 'Iniziata' (Iniziata), 'Descrizione' (Descrizione), and 'Descrizione tesi' (Descrizione tesi). The 'Compiti Assegnati' section contains a card with the following information: 'Nome Task' (Nome Task), 'Scadenza' (Scadenza), 'Stato' (Stato), 'Task 1' (Task 1), '10/08/2023' (10/08/2023), and 'Iniziato' (Iniziato). The 'Compiti Svolti' section is currently empty. At the bottom of the screen, there is a navigation bar with three icons: 'Home', 'Tesi', and 'Ricevimenti'.

Tesi	
Nome	Tesi 2
Durata	150 ore
Iniziata	Iniziata
Descrizione	
Descrizione tesi	

Compiti Assegnati	
Nome Task	Task 1
Scadenza	10/08/2023
Stato	Iniziato

Compiti Svolti

2.24. Aggiunta di un co-relatore

Un docente ha la possibilità di aggiungere un co-relatore all'interno di una determinata tesi.

The screenshot shows the 'Richieste Tesi' (Thesis Requests) screen. At the top, there is a dark blue header with a back arrow, the title 'Richieste Tesi', and a user profile icon. Below the header, the screen displays a card with the following information: 'Studente' (Studente), 'Paolo Rossi' (Paolo Rossi), 'Tesi' (Tesi), 'Tesi 3' (Tesi 3), 'Descrizione' (Descrizione), 'Richiesta tesi' (Richiesta tesi), and 'Indicare un corelatore' (Indicare un corelatore). Below the card, there is a dropdown menu labeled 'Corelatore' with a downward arrow. At the bottom of the card, there are two buttons: 'ACCETTA' (Accept) and 'RIFIUTA' (Reject). At the bottom of the screen, there is a navigation bar with three icons: 'Home', 'Mie Tesi', and 'Tesi Avviate'.

Studente
Paolo Rossi
Tesi
Tesi 3
Descrizione
Richiesta tesi
Indicare un corelatore

Corelatore

ACCETTA RIFIUTA

3. Strumenti utilizzati

Gli strumenti utilizzati per la realizzazione dell'applicazione sono stati:

- GitHub
- Android Studio
- Firebase Storage
- Firebase Authenticator
- Cloud Firestore

4. Progettazione dell'icona

Nella fase di progettazione dell'icona del prodotto, abbiamo deciso di utilizzare un cappello accademico in quanto permette di comunicare velocemente qual è l'intento del prodotto. Il cappello accademico è rappresentato con una forma netta e linee ortogonali, inoltre abbiamo optato per un design semplice in modo da avere un'icona che risulti chiara e leggibile anche a dimensioni ridotte. Per quanto riguarda i colori, abbiamo scelto il blu per lo sfondo e il bianco per il cappello accademico in quanto, questi colori, evocano un senso di fiducia che riteniamo sia importante per un'applicazione di questo tipo.



5. Ruolo dei componenti

La suddivisione dei compiti è stata eseguita nel seguente modo:

Sviluppo Funzionalità:

- Diego Clemente
- Nicola De Veteris

Sviluppo Layout:

- Diego Clemente
- Giuseppe Lorusso

Gestione Database:

- Pasquale Dipalma

Scrittura Documentazione:

- Pasquale Dipalma

6. Limiti dell'app

Di seguito sono riportati i limiti dell'app, che saranno oggetto di miglioramento in futuro:

- Assenza di un contesto universitario che permette l'utilizzo di account universitari
- Gestione delle notifiche

7. Sviluppi futuri

In base ai limiti dell'app che sono stati riscontrati possiamo definire quelli che potrebbero essere alcuni sviluppi futuri dell'applicazione:

- Sviluppo di funzionalità per l'utilizzo di account universitari

8. Credenziali di accesso al sistema

Per accedere all'app per scopi di test, come studente, possono essere utilizzate le seguenti credenziali:

- E-Mail: studente@gmail.com
- Password: studente123

Studente senza tesi assegnata

- E-Mail: studente1@gmail.com
- Password: studente

Per accedere all'app per scopi di test, come docente, possono essere utilizzate le seguenti credenziali:

- E-Mail: docenzio@gmail.com
- Password: docenzio

- E-Mail: docente2@gmail.com
- Password: docente2