

Premesse generali del progetto:

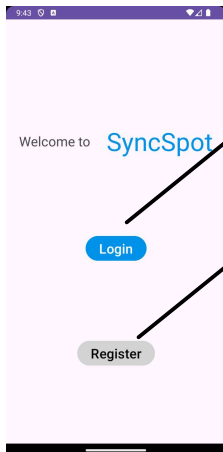
- In tutte le Classi Activity del progetto è presente il view/data binding
- In tutte le Classi Activity del progetto è presente un controllo sulla autenticazione dell'utente

```
@Override
protected void onStart() {
    super.onStart();
    FirebaseAuth user = FirebaseAuth.getInstance().getCurrentUser();
    if (user != null) {
        Log.i( tag: "User UID", user.getId());
        Intent itn = new Intent( packageContext: WelcomePage.this, HomePage.class);
        startActivity(itn);
    }
}
```

Controllo l'effettiva esistenza dell'utente e ,nel caso non ci fosse, riporta l'utente alla schermata iniziale

La sessione viene in automatico gestita da firebase

Prima schermata visualizzata dall'utente - WelcomePage

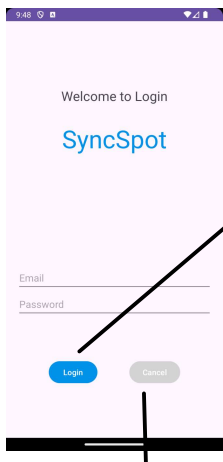


```
pageWelcomeBinding.buttonLogin.setOnClickListener(v -> {
    Intent itn = new Intent( packageContext: WelcomePage.this, LoginPage.class);
    startActivity(itn);
});

pageWelcomeBinding.buttonRegister.setOnClickListener(v -> {
    Intent itn = new Intent( packageContext: WelcomePage.this, RegisterPage.class);
    startActivity(itn);
});
```

Il bottone Login rimanda alla Activity specifica
Il bottone Register rimanda alla Activity specifica

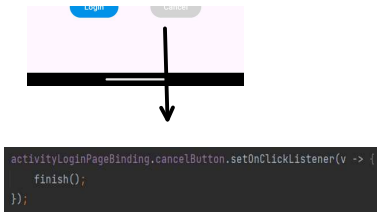
Pagina di login



```
if(String.valueOf(activityLoginPageBinding.emailEditText.getText()).equals("") ||
    String.valueOf(activityLoginPageBinding.passwordEditText.getText()).equals("")){
    Toast.makeText( context: this, text: "Attenzione: assicurarsi di aver inserito email e password", Toast.LENGTH_SHORT).show();
}
```

Controlla effettivamente che l'utente abbia inserito qualcosa

```
}else{
    mAuth.signInWithEmailAndPassword(String.valueOf(activityLoginPageBinding.emailEditText.getText())
        , String.valueOf(activityLoginPageBinding.passwordEditText.getText()))
        .addOnCompleteListener( activity: this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()) {
                    FirebaseAuth user = mAuth.getCurrentUser();
                    Log.i( tag: "User UID", user.getId());
                    Intent itn = new Intent( packageContext: LoginPage.this, HomePage.class);
                    startActivity(itn);
                } else {
                    Toast.makeText( context: LoginPage.this, text: "Attenzione: email o password errati", Toast.LENGTH_SHORT).show();
                }
            }
        })
}
```



Torna indietro

```
startActivity(itn);
} else {
    Toast.makeText(context: LoginPage.this, text: "Attenzione: email o password errati", Toast.LENGTH_SHORT).show();
}
});
}
```

La funzione `signInWithEmailAndPassword` viene utilizzata per loggare all'interno del sistema di firebase, tutti i controlli e le autorizzazioni vengono eseguiti da quest'ultimo

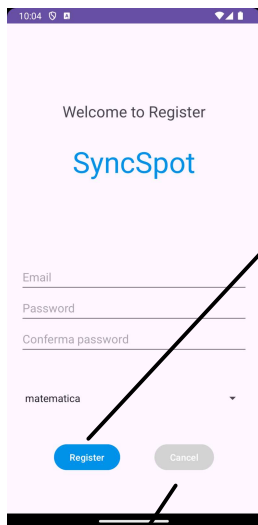
Una volta completata l'operazione, controllo lo stato di quest'ultima:

- Se andata a buon fine passo alla schermata principale del progetto
- Altrimenti avviso l'utente del proprio errore

La variabile `mAuth` è globale e deve essere definita ad inizio Activity

```
mAuth = FirebaseAuth.getInstance();
```

Pagina di Registrazione



```
ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(context: this, R.array.facolta_array, android.R.layout.simple_spinner_item);
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
activityRegisterPageBinding.facoltaSpinner.setAdapter(adapter);
```

Popolo lo Spinner con un array personalizzato

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="facolta_array">
        <item>matematica</item>
        <item>fisica</item>
        <item>chimica</item>
        <item>informatica</item>
    </string-array>
</resources>
```

L'array personalizzato è salvato su un file XML all'interno Della cartella values

```
if(String.valueOf(activityRegisterPageBinding.emailEditText.getText()).equals("") ||
    String.valueOf(activityRegisterPageBinding.passwordEditText.getText()).equals("")){
    Toast.makeText(context: this, text: "Attenzione: assicurarsi di aver inserito email e password", Toast.LENGTH_SHORT).show();
}
```

Controlla effettivamente che l'utente abbia inserito qualcosa

```
}else if (!String.valueOf(activityRegisterPageBinding.passwordEditText.getText())
    .equals(String.valueOf(activityRegisterPageBinding.passwordConfirmEditText.getText()))){
    Toast.makeText(context: this, text: "Attenzione: le due password non corrispondono", Toast.LENGTH_SHORT).show();
}
```

Controlla effettivamente che l'utente abbia inserito correttamente la password

```
mAuth.createUserWithEmailAndPassword(String.valueOf(activityRegisterPageBinding.emailEditText.getText()),
    String.valueOf(activityRegisterPageBinding.passwordEditText.getText()))
```

La funzione `createUserWithEmailAndPassword` viene utilizzata per registrarsi all'interno del sistema di firebase, tutti i controlli e le autorizzazioni vengono eseguiti da quest'ultimo

```
activityLoginPageBinding.cancelButton.setOnClickListener(v -> {
    finish();
});
```

Torna indietro

```

.addOnCompleteListener( activity: this, new OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if (task.isSuccessful()) {
            FirebaseAuth user = mAuth.getCurrentUser();
            Map<String, String> userInfo = new HashMap<>();
            userInfo.put("uidUtente", user.getId());
            userInfo.put("facolta", activityRegisterPageBinding.facoltaSpinner.getSelectedItem().toString());
            db.collection( collectionPath: "facoltaUtente" ) CollectionReference
                .add(userInfo) Task<DocumentReference>
                .addOnSuccessListener(new OnSuccessListener<DocumentReference>() {
                    @Override
                    public void onSuccess(DocumentReference documentReference) {
                        Intent itn = new Intent( packageContext: RegisterPage.this, HomePage.class);
                        startActivity(itn);
                    }
                })
        }
    }
});

```

Se l'utente viene registrato correttamente, inserisco in firestore un riferimento che collega l'id dell'utente con la facoltà che ha scelto, per poi spostarlo nella pagina principale del progetto

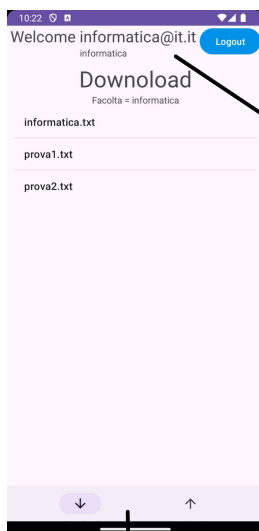
```

.addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) {
        Toast.makeText( context: RegisterPage.this, e.getMessage(), Toast.LENGTH_SHORT).show();
    }
});
} else {
    Toast.makeText( context: RegisterPage.this, text: "Attenzione: qualcosa è andato storto durante la registrazione", Toast.LENGTH_SHORT).show();
}
}

```

Nel caso in cui il sistema non riesca a registrare l'utente oppure non riesca ad associare l'id alla facoltà, avviso l'utente

Pagina di Homepage della applicazione



```

activityHomePageBinding.logoutButton.setOnClickListener(v -> {
    FirebaseAuth.getInstance().signOut();
    onStart();
});

```

Bottone di logout per far cadere la sessione

```

if (user != null) {
    Log.i( tag: "User UID", user.getId());
    activityHomePageBinding.welcomeTitle.setText("Welcome " + user.getEmail());
}

```

Se l'utente logga con successo visualizzo la sua email

```

db.collection( collectionPath: "facoltaUtente" ) CollectionReference
    .whereEqualTo( field: "uidUtente", mAuth.getId()) Query
    .get() Task<QuerySnapshot>
    .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
        @Override
        public void onComplete(@NonNull Task<QuerySnapshot> task) {
            if (task.isSuccessful()) {
                for (QueryDocumentSnapshot document : task.getResult()) {
                    Log.i( tag: "Result321", msg: document.getId() + " => " + document.getData());
                    facolta = (String) document.getString( field: "facolta");
                    activityHomePageBinding.facoltaTextView.setText(facolta);
                    applicaFragment( title: "download");
                }
            } else {
                Log.i( tag: "Result321", msg: "Error getting documents: ", task.getException());
            }
        }
    });

```

Per estrapolare la facoltà associata all'utente, interrogo firestore e assegno ad una variabile globale la facoltà dell'utente

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/nav_download"
        android:icon="@drawable/download"
        android:title="download"/>
    <item
        android:id="@+id/nav_upload"
        android:icon="@drawable/upload"
        android:title="upload"/>
</menu>
```

Creo un menu che mi permetta di navigare tra i due fragment

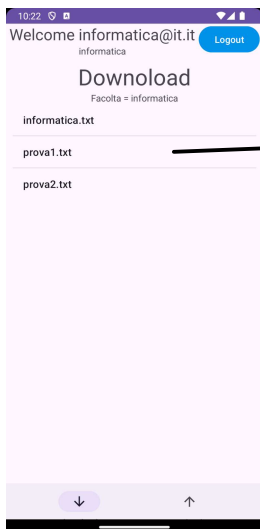
```
activityHomePageBinding.bottomNav.setOnItemSelectedListener(item -> {
    applicaFragment(item.getTitle().toString());
    return true;
});
```

Nella pagina di Homepage associo una funzione applicaFragment ai due bottoni presenti nel menu

```
protected void applicaFragment(String title){
    if(title.equals("download")){
        download downloadFragment = download.newInstance("Facolta = ", facolta);
        FragmentManager fragmentManager = getSupportFragmentManager();
        fragmentManager.beginTransaction()
            .replace(R.id.fragment_container, downloadFragment, tag: null)
            .setReorderingAllowed(true)
            .addToBackStack( name: "name")
            .commit();
    }else{
        upload uploadFragment = upload.newInstance("Facolta = ", facolta);
        FragmentManager fragmentManager = getSupportFragmentManager();
        fragmentManager.beginTransaction()
            .replace(R.id.fragment_container, uploadFragment, tag: null)
            .setReorderingAllowed(true)
            .addToBackStack( name: "name")
            .commit();
    }
}
```

Tale funzione modifica e visualizza i fragment all'interno del fragment container. Ad ogni fragment passo la facoltà associata all'utente

Fragment di Download



```
FirebaseStorage storage = FirebaseStorage.getInstance();

StorageReference listRef = storage.getReference().child( pathString: "/" + mParam2);

listRef.listAll()
    .addOnSuccessListener(new OnSuccessListener<ListResult>() {
        @Override
        public void onSuccess(ListResult listResult) {
            storageList = new ArrayList<String>();
            storageReferences = new ArrayList<StorageReference>();

            for (StorageReference item : listResult.getItems()) {

                Log.i( tag: "Storage123Item", item.getPath());
                storageReferences.add(item);
                storageList.add(item.getName());
            }
        }
    });
```

Istanzio una connessione con firebase storage, e navigo all'interno della cartella specifica associata alla facoltà

Una volta all'interno, popolo una lista con tutti i file contenuti in essa

```
ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>(getView().getContext(), android.R.layout.simple_list_item_1, storagelist);
ListView v = getView().findViewById(R.id.listView);
v.setAdapter(arrayAdapter);
```

Popolo la listview

```
v.setOnItemClickListener((parent, view1, position, id) -> {

    StorageReference islandRef = storage.getReference().child(storageReferences.get(position).getPath());

    String name = storageReferences.get(position).getName();
    String[] fileInfo = name.split( regex: "\\.");
```

Estrapolo le informazioni del file che l'utente ha scelto di scaricare

```
islandRef.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
    @Override
    public void onSuccess(Uri uri) {
        DownloadManager.Request request = new DownloadManager.Request(uri);

        request.setNotificationVisibility(DownloadManager.Request.VISIBILITY_VISIBLE_NOTIFY_COMPLETED);
        request.setDestinationInExternalFilesDir(getContext(), Environment.DIRECTORY_DOWNLOADS, name);

        DownloadManager downloadManager = (DownloadManager) getContext().getSystemService(Context.DOWNLOAD_SERVICE);
        downloadManager.enqueue(request);

        Toast.makeText(getContext(), text: "Documento scaricato con successo", Toast.LENGTH_SHORT).show();
    }
}
```

Estrapolo il link di download associato al file e lo scarico sul dispositivo

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

Si per leggere che per scrivere sul dispositivo abilito i seguenti permessi

```
}).addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception exception) {
        Toast.makeText(getContext(), text: "Errore durante il download del file", Toast.LENGTH_SHORT).show();
    }
});
});
}
)
addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) {
        Log.i( tag: "ErroreStorage", e.getMessage());
        Toast.makeText(getContext(), text: "Errore durante la visualizzazione dei file", Toast.LENGTH_SHORT).show();
    }
});
});
```

Gestisco gli errori associati alla lettura e download dei file

Fragment di Download



```
Button b = getView().findViewById(R.id.uploadButton);
b.setOnClickListener(this::chooseFile);
}

1 usage
private void chooseFile(View view){
    Intent intent = new Intent();
    intent.setAction(Intent.ACTION_GET_CONTENT);
    intent.setType("*/*");
    intent = Intent.createChooser(intent, "Choose a file");
    startActivityForResult(intent, requestCode: 100);
}
```

Associo al bottone Carica una funzione che apre un Intent personalizzato che permette all'utente di selezionare un file dal proprio archivio interno

```
@Override
public void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if(data != null){
        Uri fileUri = data.getData();
        UploadTask uploadTask = listRef.child( pathString: "/"+"mParam2"+"/"+fileUri.getLastPathSegment())
            .putFile(fileUri);

        uploadTask.addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>(){
            @Override
            public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                Toast.makeText(getContext(), text: "Documento caricato con successo", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

Una volta conclusa l'operazione di selezione, controllo che l'utente abbia effettivamente selezionato un file e lo salvo nella directory di firebase storage associata alla facoltà specifica dell'utente