

System Manual

BuddyU

The purpose of the BuddyU app is to provide a social platform for college students to collaborate on their schedules in order to make class time more efficient, collaborative and fun.

The current BuddyU app is currently designed for the Android platform and leverages the Firebase set of services for much of its functionality.

File: registration.java

Class: registration

XML: activity_registration.xml

- Code allowing a user to register with the BuddyU service.
- We currently support Google, Facebook, or email authentication.
- This is an Android activity. The method **onCreate()** is used to initialize many of the data fields of the registration activity including Firebase database references and making connections to UI elements defined in the XML. In addition, a listener is registered for authentication state changes.
- The **userLogin()** method takes the email and password entered by the user and makes the appropriate Firebase calls to register the user.
- The **onActivityResult(int requestCode, int resultCode, Intent data)** method is how we get the results of a Facebook or Google sign-in attempt.
- The **firebaseAuthWithGoogle(GoogleSignInAccount acct)** method registers the Google user with Firebase.
- The **handleSignInResult(GoogleSignInResult result)** method gives the result of signing in with the account using Google credentials.
- The **signIn()** method pops up an Android activity that gathers the Google credentials.
- The **handleFacebookAccessToken(AccessToken token)** registers the Facebook user with Firebase.

File: LoginScreen.java

Class: LoginScreen

XML: activity_login_screen.xml

- Code allowing user to have an option of logging in with either Google, Facebook, or email.
- This is an Android activity. The method **onCreate()** is used to initialize many of the data fields of the LoginScreen activity including Firebase database references and making connections to UI elements defined in the XML. In addition, a listener is registered for authentication state changes.
- The method **userLogin()** is what actually reads in the email and password.
 - The method first checks to make sure that the user has entered both their email and password if neither of these things are true then the program outputs an error message.
 - If all the information is entered correctly the program then lets the user know they are being signed in, it then calls the firebase sign in method.

- If the user is successfully able to be signed in then the program navigates the Main Activity.
- If the user's login was unsuccessful then an error message is outputted to the user.
- The **firebaseAuthWithGoogle(GoogleSignInAccount acct)** attempts to sign-in the user with Google credentials.
- The **handleSignInResult(GoogleSignInResult result)** method gives the result of signing in with the account using Google credentials.
- The **signIn()** method pops up an Android activity that gathers the Google credentials.
- The **handleFacebookAccessToken(AccessToken token)** signs-in the Facebook user with Firebase.

File: MainActivity.java

Class: MainActivity

- The MainActivity class has code that allows for navigation between screens from the sidebar/navigation drawer. Specifically with the method **onNavigationItemSelected(Menuitem item)**:
- If home is clicked in the sidebar which has the id **nav_home**, then it will move to the home fragment.
- If classes is clicked in the sidebar which has the id **nav_classes**, then it will move to the classes fragment.
- If schedule is clicked in the sidebar (id: **nav_schedule**), then it will move to the schedule fragment.
- If profile is clicked in the sidebar (id: **nav_profile**), then it will move to the profile fragment.
- If settings is clicked in the sidebar (id: **nav_settings**), then it will move to the settings fragment.
- If the logout button is clicked in the sidebar (id: **nav_logout**) then it will sign the current user out with the line: **firebaseAuth.signOut()**; and moves the user back to the registration screen using an intent.

File: HomeFragment.java

Class: HomeFragment

XML: fragment_home.xml

- This is the first screen that the user will be greeted with when they sign into the app.
- The **onViewCreated(View view, Bundle savedInstanceState)** method allows navigation for changing to the About BuddyU screen if the button **about_app** is pressed and Terms and Conditions if the **terms_and_conditions** button is pressed.

File:ProfileFragment.java

Class: ProfileFragment

XML: fragment_profile.xml

- The **onViewCreated(View view, Bundle savedInstanceState)** method initializes the class data fields including connecting to the UI elements defined in the XML as well as connecting to Firebase storage which we will use for storing our profile picture for the user. A button handler for the “upload image” button is registered and launches a gallery intent if clicked to allow the user to take a photo using the camera or select a photo from their device.
- The **loadProfilePicture()** method retrieves the image url for the user’s stored profile picture from Firebase. If successful, it creates a DownloadImageTask object.
- The **onActivityResult(int requestCode,int resultCode, Intent data)** method gets the results of what the user selected in the gallery intent and uploads the image to Firebase.

Class: DownloadImageTask

- This class downloads the actual image from Firebase given its url and displays it in an ImageView.

Class:YourBuddiesFragment.java

XML: fragment_your_buddies.xml

Class: Buddy

- This class is used by an array list to represent each buddy displayed in the list box.
- The method **getName()** returns the name of the buddy.
- The method **setName()** sets the name of the buddy.
- The method **getUID()** returns the uid of the buddy.
- The method **setUID()** sets the uid of the buddy.
- The method **isSelected()** returns whether the buddy has been selected in the list.
- The method **setSelected()** sets the state of whether the buddy has been selected in the list.

Class:YourBuddiesFragment

- This fragment allows you to see all the users registered with the app from Firebase and lets you select which ones you want to select as your buddies.
- The **onViewCreated(View view, Bundle savedInstanceState)** method initializes the class data fields including connecting to the UI elements defined in the XML and calls the method **displayListView()**.
- **displayListView()** initializes more of the class data fields and retrieves the list of users from the Firebase database and displays them.

Class: MyCustomAdapter

- Allows for customizing the list box appearance and behavior.

Class:ViewHolder

- Class represents UI elements used within a given list item.

- The **getView(int position, View convertView, ViewGroup parent)** registers event handlers with the UI elements in the list box so we can respond to selecting/unselecting buddies. If a buddy is selected or unselected the uids for the current selected buddies are stored in the database.

Class:Course.java

- This class is used by an array list to represent each course displayed in the list box.
- The method **getName()** returns a string for the courses name.
- The method **setName()** allows the courses name to be set.
- The method **getCRN()** returns a string for the crn.
- The method **setCRN()** allows for setting the crn.
- The method **isSelected()** returns whether the course has been selected in the list.
- The method **setSelected()** allows setting the state of whether the course is selected in the list.

File: ClassesFragment.java

Class: ClassesFragment

XML: fragment_classes.xml

- This fragment allows you to see classes (courses) from the College of Computing and Informatics, College of Engineering, and College of Arts and Sciences. When you select a class it will store it in Firebase.
- The **onViewCreated(View view, Bundle savedInstanceState)** method initializes the class data fields including connecting to the UI elements defined in the XML. It retrieves the list of courses from the Firebase database and displays them.

Class: MyCustomAdapter2

- Allows for customizing the list box appearance and behavior.
Class:ViewHolder
 - Class represents UI elements used within a given list item.
- The **getView(int position, View convertView, ViewGroup parent)** registers event handlers with the UI elements in the list box so we can respond to selecting/unselecting courses. If a course is selected or unselected the crns for the current selected courses are stored in the database.
- The method **getFilter()** allows for filtering the list of courses given a search string.

Class:ScheduleFragment.java

XML: fragment_schedule.xml

Class:Item

- This class is used by an array list to represent a single list item in the list box.
- Each class item is stored as a private variable as they don't need to be modified outside of the class only accessed.
- The method **getDay()** returns a string of the private variable day
- The method **getDescription()** returns a string of the course description
- The method **getTime()** returns an int of the time.
- Finally **getCRN()** returns a string with the crn.

Class: ScheduleFragment

- This fragment displays the list of courses that a user has selected for themselves or can display a buddy's schedule.
- The **onViewCreated(View view, Bundle savedInstanceState)** method initializes the class data fields including connecting to the UI elements defined in the XML. It retrieves the list of classes selected from the Firebase database and displays them.
- The method **itemsForDay()** returns text for all the courses for a given day suitable for displaying in a rich text field (setting to HTML).
- The method **updateItems()** first sorts the items in order of their time, then compiles each of the classes into their respective days and displays all the classes within a text view.