



System Design Document

Purpose of this Document

The purpose of this document is to provide an overview of the design choices for this App.

This application is for Android devices. It utilizes Google firebase for a real-time database, cloud storage, and user authentication.

System Description

Using deep learning, we can maximize group performance, while making the work experience more pleasurable.

We created an Android application that facilitates effective group matching via deep learning. Specifically, we matched group members pairing complementary traits based on existing studies and personality analysis; namely, the Myers-Briggs Type Indicator test, and custom skills matrix. Students who have already made an account register to classes using a unique code provided by a supervisor or professor, which prompts them to then take a series of skill and personality assessments. At the end of the term, students rate their group members on how well they collaborated. Professors have the option to create classes, and then when all students have joined the class, they can click “Create Groups”, which will contact the server with the neural net script and automatically create groups and send the result to firebase database.

We used Google Firebase to store the results from tests and for group matching, as well as user profile information. Deep learning can be used to improve group matching through the rating system.

Versions and Libraries

Below is a snippet of the Gradle file:

```
android {  
    compileSdkVersion 28  
    defaultConfig {
```

```

        applicationId "wpi.jtkaplan.teamup"
        minSdkVersion 27
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'),
            'proguard-rules.pro'
        }
    }
}

```

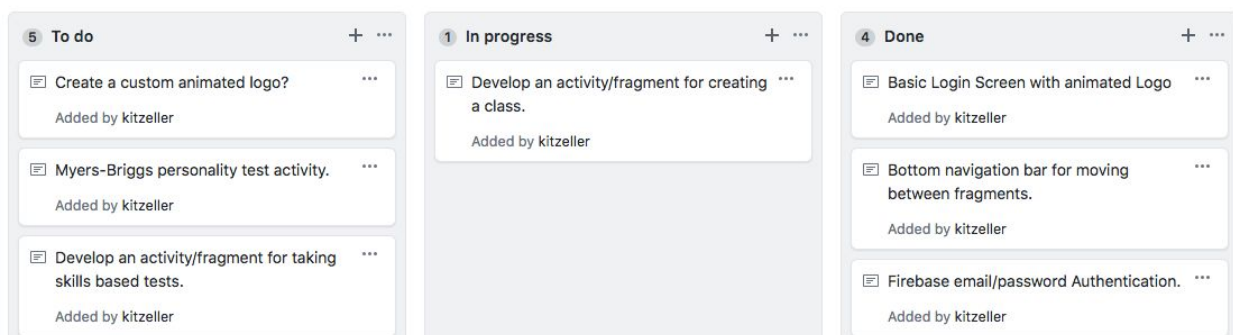
Dependencies include:

```

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support:design:28.0.0'
    implementation 'android.arch.lifecycle:extensions:1.1.1'
    implementation 'com.android.support:cardview-v7:28.0.0'
    implementation 'com.android.support:recyclerview-v7:28.0.0'
    implementation 'com.github.clans:fab:1.6.4'
    implementation 'com.squareup.picasso:picasso:2.71828'
    implementation 'com.google.firebase:firebase-database:11.0.4'
    implementation 'com.google.firebase:firebase-auth:11.0.4'
    implementation 'com.google.firebase:firebase-storage:11.0.4'
    implementation "com.hootsuite.android:nachos:1.1.1"
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
}

```

Task management using Github Projects:



Initial Prototypes

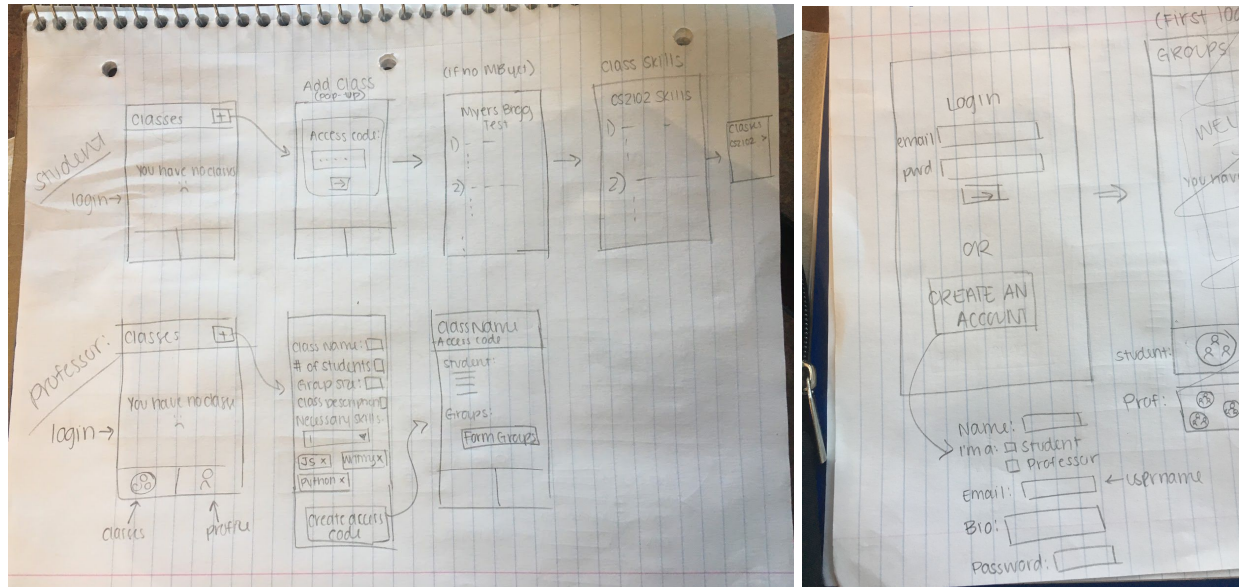


Fig 1. UI Design

Design

Naming Conventions:

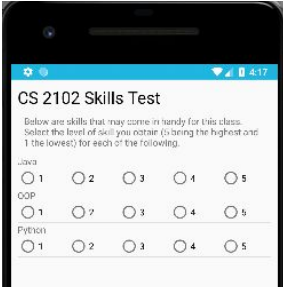
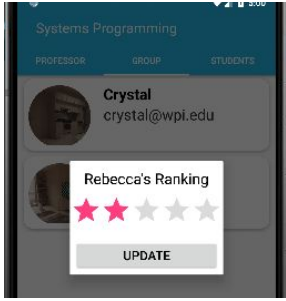
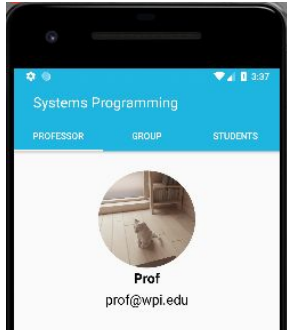
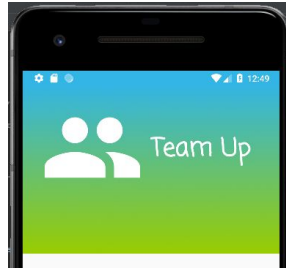
Field ID's are prefixed with the initialism for the activity e.g. for "Create Activity" CA_btnSubmit

Feature List:

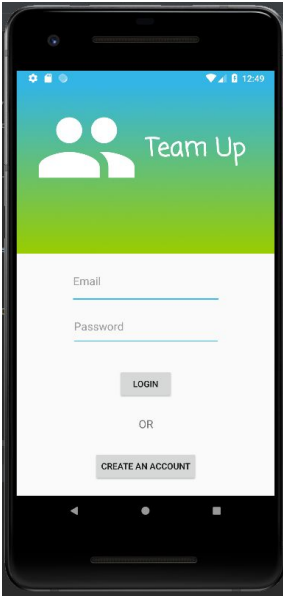
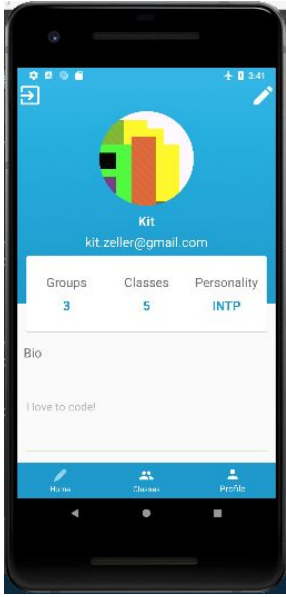
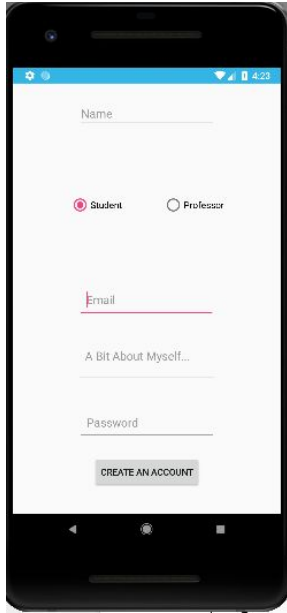
Chips for Skills tags	Drop down Option Menu	RecyclerView with CardViews	Custom DialogFragment
Dynamically	Group member rating	Custom profile picture	Animated color Login

Team 1005

TeamUp

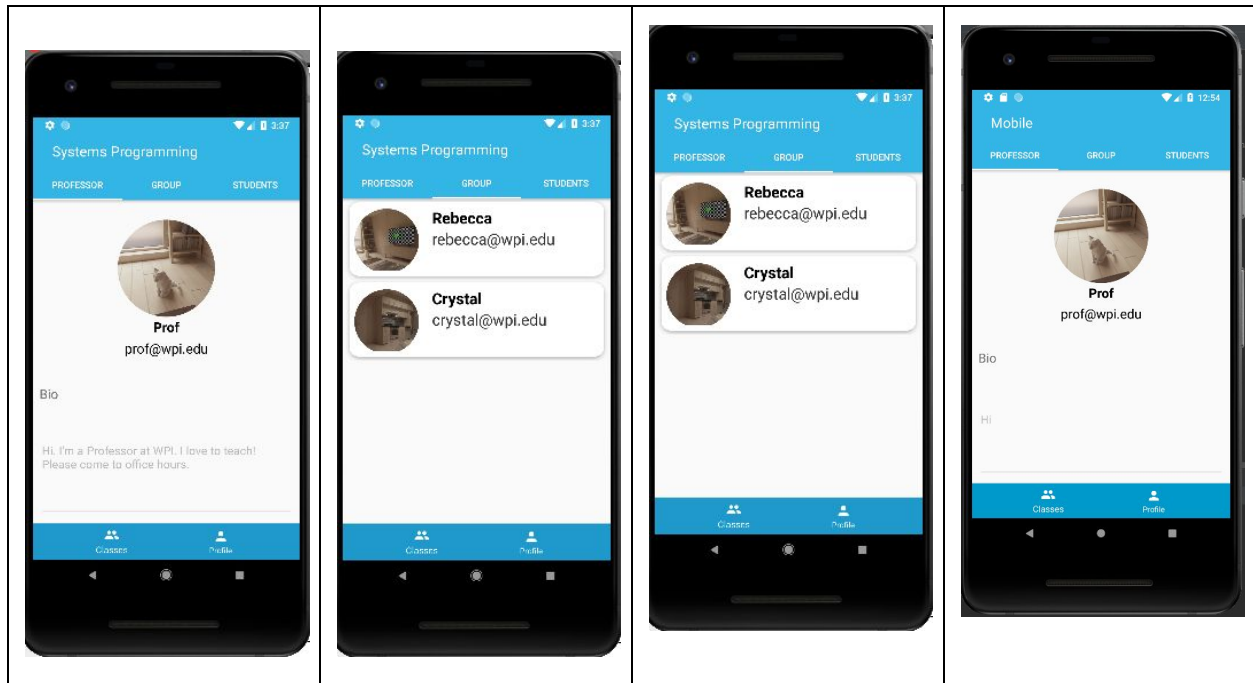
generated skills test	dialog	from Camera	Screen
			

Common/Non-User Specific Views

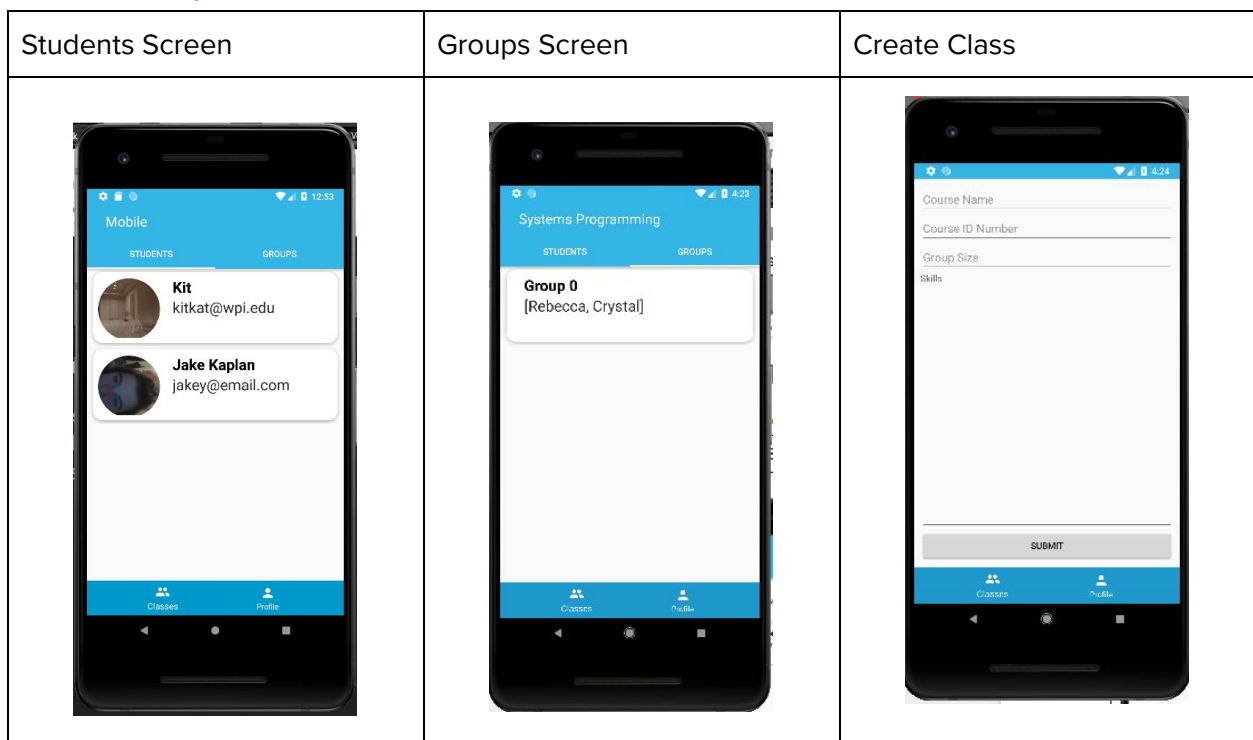
Login Screen	Profile Screen	Create Account	Classes
			

Student-only Views

Class Screen	Students Screen	Group Screen	Professor Screen
--------------	-----------------	--------------	------------------

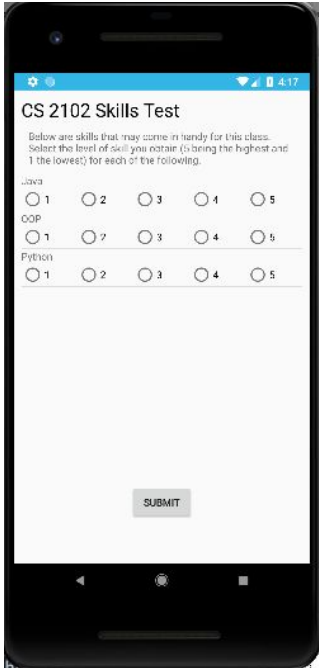


Professor-only Views



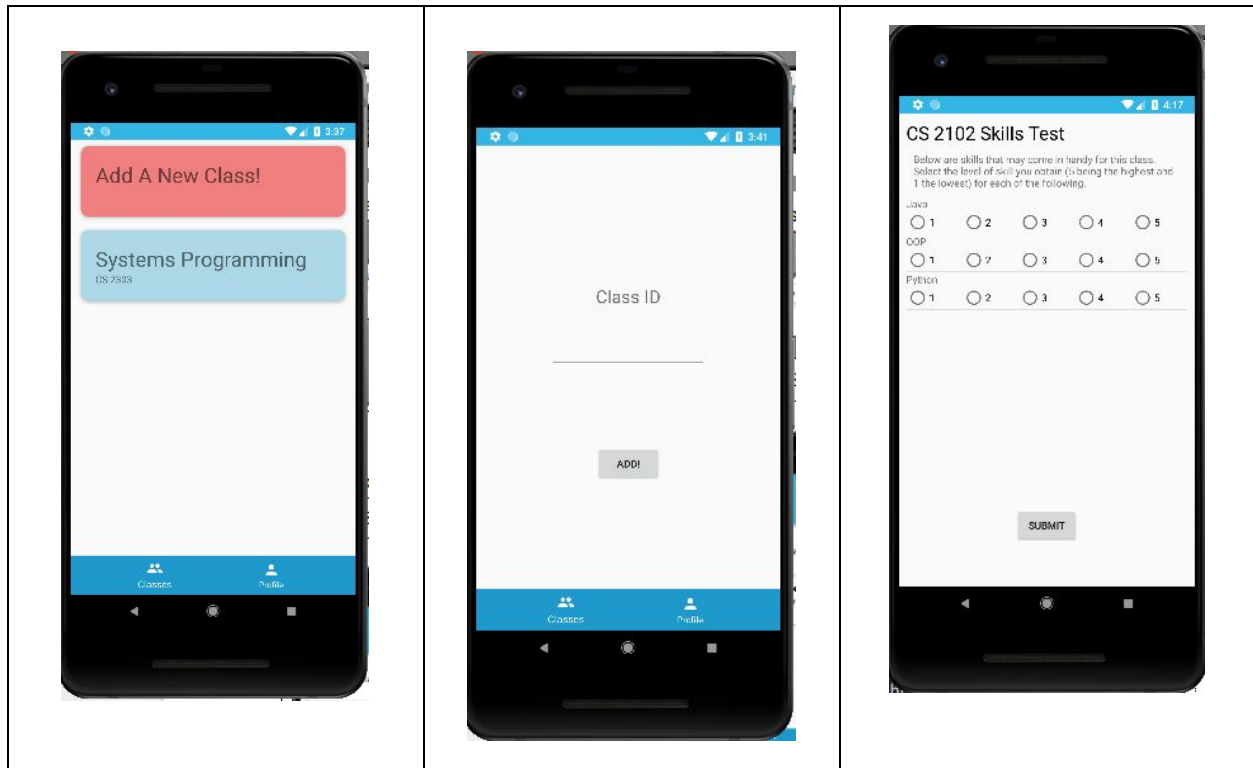
Flow of Events

Student Joins a Class

1. The student clicks "Add class"	2. The student enters unique class UID	3. The student completes skills assessment
		

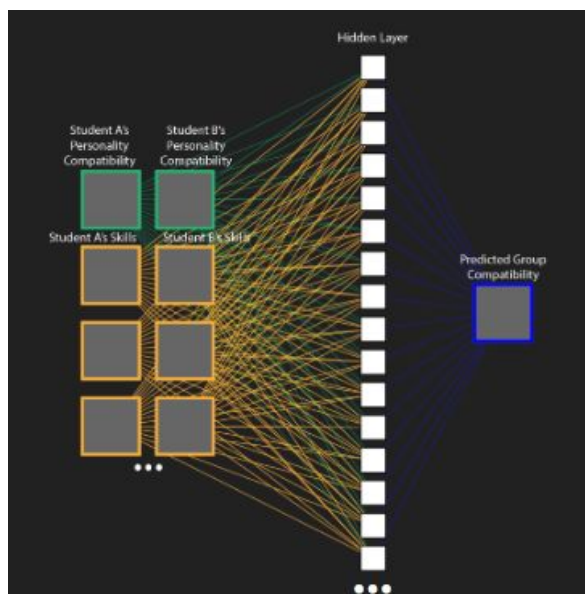
Professor Creates a Class

1. The professor clicks add class	2. The professor enters class details	3. The student completes skills assessment
-----------------------------------	---------------------------------------	--

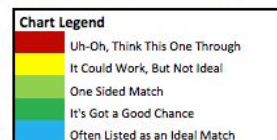
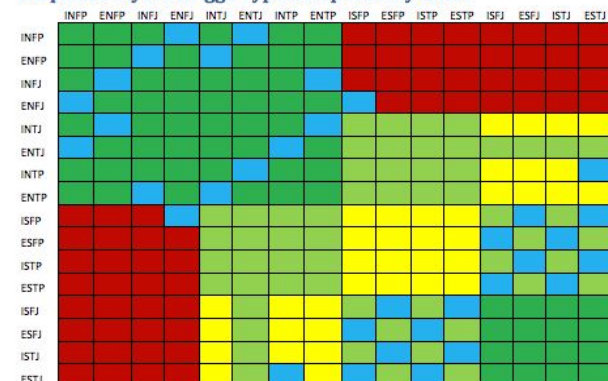


Neural Network Design

- Based on Myers Briggs Personality Reference and Skills Assessment Scores
 - Myers Briggs used for personality compatibility of students
 - Skills assessment used to match pairs with different strong skillsets



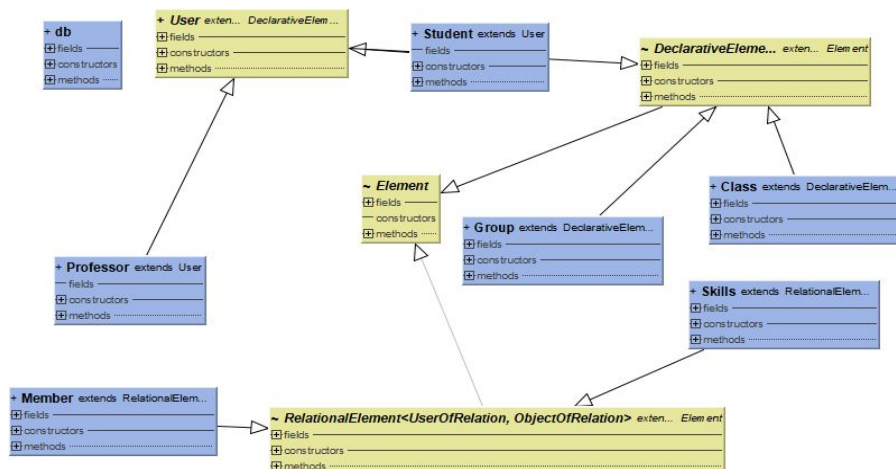
Simplified Myers Briggs Type Compatibility Chart



- Developed Model
 - For a model on N skills, the input layer size (s) is $2(N+1)$.
 - The extra 1 is to account for each student's M.B. score.
 - The hidden layer is size $2s$.
 - The output layer is always size 1.
 - This gives us a dynamic method to quantify the compatibility score of StudentA with StudentB

Firestore Design

- Flat JSON Tree Structure for quick access
- Use of UIDs
 - User
 - Student
 - Professor
 - Members
 - Skills
 - Classes
 - Groups
- Multiple Endpoints
 - Android App
 - Python ANN Server



Team 1005

TeamUp

```
teamup-224504
├── Classes
│   ├── -LTJ7pYqPOSDAZOmDnhQ
│   ├── -LTJCTOjsV15Ch98Ebnz
│   └── -LTLv0W64k93tfNoo4-
├── E2U
├── Members
│   ├── -LTJ97KGF32BMRx9uzn:-LTJ7pYqPOSDAZOmDnhQ
│   ├── -LTJ97KGF32BMRx9uzn:-LTJCTOjsV15Ch98Ebnz
│   ├── -LTJJ77kIDAmC5i3_43:-LTJCTOjsV15Ch98Ebnz
│   └── -LTLm_6lDmSRJU8uZuBx:-LTJ7pYqPOSDAZOmDnhQ
├── Professors
│   ├── -LTJ7k4KyAENZf1yzC89
│   └── -LTJAY3lmi0CAOkelNvG
├── Skills
│   ├── -LTJ97KGF32BMRx9uzn:-LTJ7pYqPOSDAZOmDnhQ
│   ├── -LTJ97KGF32BMRx9uzn:-LTJCTOjsV15Ch98Ebnz
│   ├── -LTJJ77kIDAmC5i3_43:-LTJCTOjsV15Ch98Ebnz
│   └── -LTLm_6lDmSRJU8uZuBx:-LTJ7pYqPOSDAZOmDnhQ
└── Students
    ├── -LTJ7ClS_YXvWaNeyJQO
    ├── -LTJ97KGF32BMRx9uzn
    ├── -LTLm_6lDmSRJU8uZuBx
    ├── -LTPpYnRYBxuUOrxDxRv
    └── -LTPrQZlgbpNAXXT-1r
```

