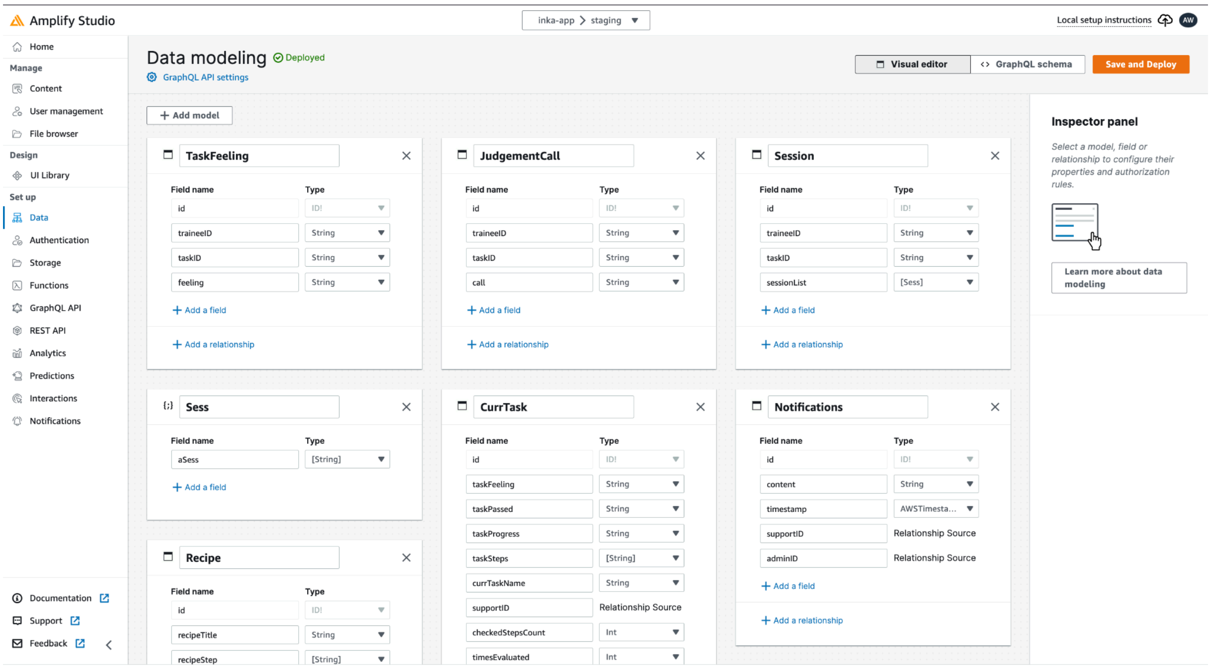
To whom it may concern,

This document is not an explanation to the codebase that you should have access to, rather it is a guide as to how we set up our AWS and the requirements you may need to do so. This should not require you to change anything within the codebase as such, this documentation is demonstrating to you, how we set up AWS.

1. Follow the steps below to set up AWS online
2. Configure AWS to your computer - this will include setting up the terminal/command line on your device
3. Ensure that when you pull the AWS Amplify studio to your device, that the AmplifyConfiguration file is now within the codebase - the codebase will not work properly if the AmplifyConfiguration file is not there
4. You should be able to test this out on a simulator of your choice
5. For this to function correctly on an iPad, you will need to add this to the Apple App Store. We cannot give you the instructions on this one, as we have not tried this ourselves, but we do understand that there is a screening process for the app to be uploaded to the Apple App Store and this may take some time.

## AWS Amplify

Please note that the creation of database should be done on AWS Amplify and this will be transferred over to AWS DynamoDB



## Data modeling

Deployed

GraphQL API settings

### TaskFeeling

| Field name | Type |
|---|---|
| id | ID! |
| traineeID | String |
| taskID | String |
| feeling | String |

### JudgementCall

| Field name | Type |
|---|---|
| id | ID! |
| traineeID | String |
| taskID | String |
| call | String |

### Session

| Field name | Type |
|---|---|
| id | ID! |
| traineeID | String |
| taskID | String |
| sessionList | [Sess] |

### Sess

| Field name | Type |
|---|---|
| aSess | [String] |

### CurrTask

| Field name | Type |
|---|---|
| id | ID! |
| taskFeeling | String |
| taskPassed | String |
| taskProgress | String |
| taskSteps | [String] |
| currTaskName | String |
| supportID | Relationship Source |
| checkedStepsCount | Int |
| timesEvaluated | Int |

### Notifications

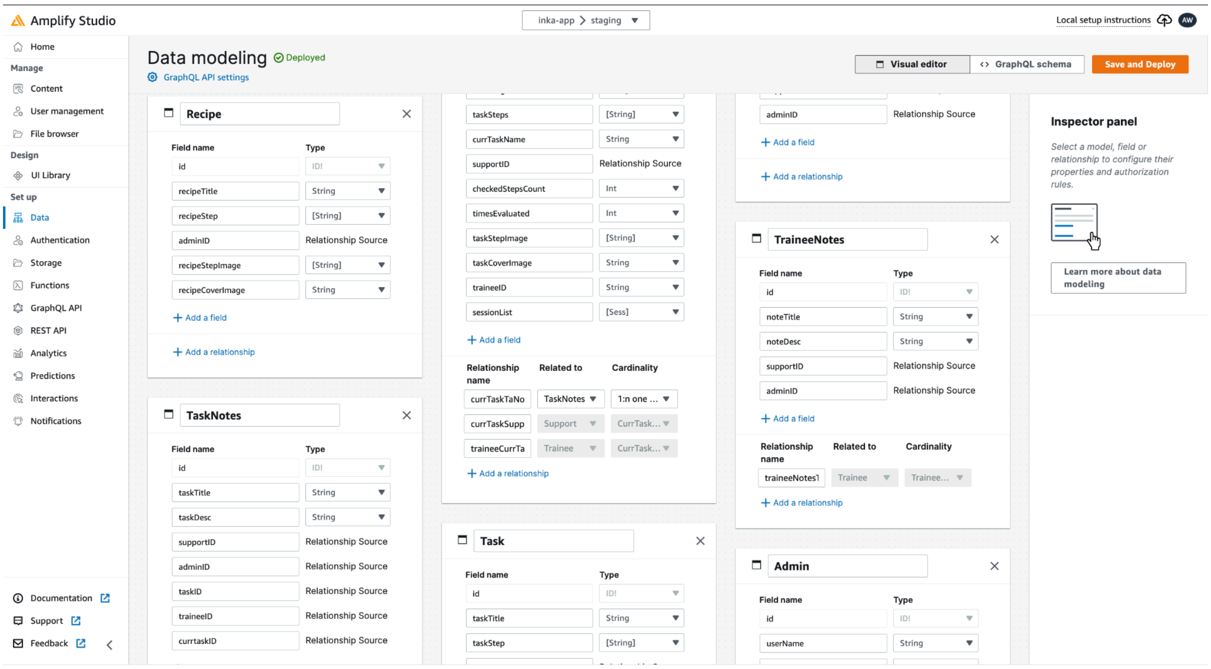
| Field name | Type |
|---|---|
| id | ID! |
| content | String |
| timestamp | AWSTimesta… |
| supportID | Relationship Source |
| adminID | Relationship Source |

### Recipe

| Field name | Type |
|---|---|
| id | ID! |
| recipeTitle | String |
| recipeStep | [String] |

### Inspector panel

*Select a model, field or relationship to configure their properties and authorization rules.*



## Data modeling

Deployed

GraphQL API settings

### Recipe

| Field name | Type |
|---|---|
| id | ID! |
| recipeTitle | String |
| recipeStep | [String] |
| adminID | Relationship Source |
| recipeStepImage | [String] |
| recipeCoverImage | String |

### TaskNotes

| Field name | Type |
|---|---|
| id | ID! |
| taskTitle | String |
| taskDesc | String |
| supportID | Relationship Source |
| adminID | Relationship Source |
| taskID | Relationship Source |
| traineeID | Relationship Source |
| currtaskID | Relationship Source |

| | |
|---|---|
| taskSteps | [String] |
| currTaskName | String |
| supportID | Relationship Source |
| checkedStepsCount | Int |
| timesEvaluated | Int |
| taskStepImage | [String] |
| taskCoverImage | String |
| traineeID | String |
| sessionList | [Sess] |

| Relationship name | Related to | Cardinality |
|---|---|---|
| currTaskTaNo | TaskNotes | 1:n one … |
| currTaskSupp | Support | CurrTask… |
| traineeCurrTa | Trainee | CurrTask… |

### Task

| Field name | Type |
|---|---|
| id | ID! |
| taskTitle | String |
| taskStep | [String] |

| | |
|---|---|
| adminID | Relationship Source |

### TraineeNotes

| Field name | Type |
|---|---|
| id | ID! |
| noteTitle | String |
| noteDesc | String |
| supportID | Relationship Source |
| adminID | Relationship Source |

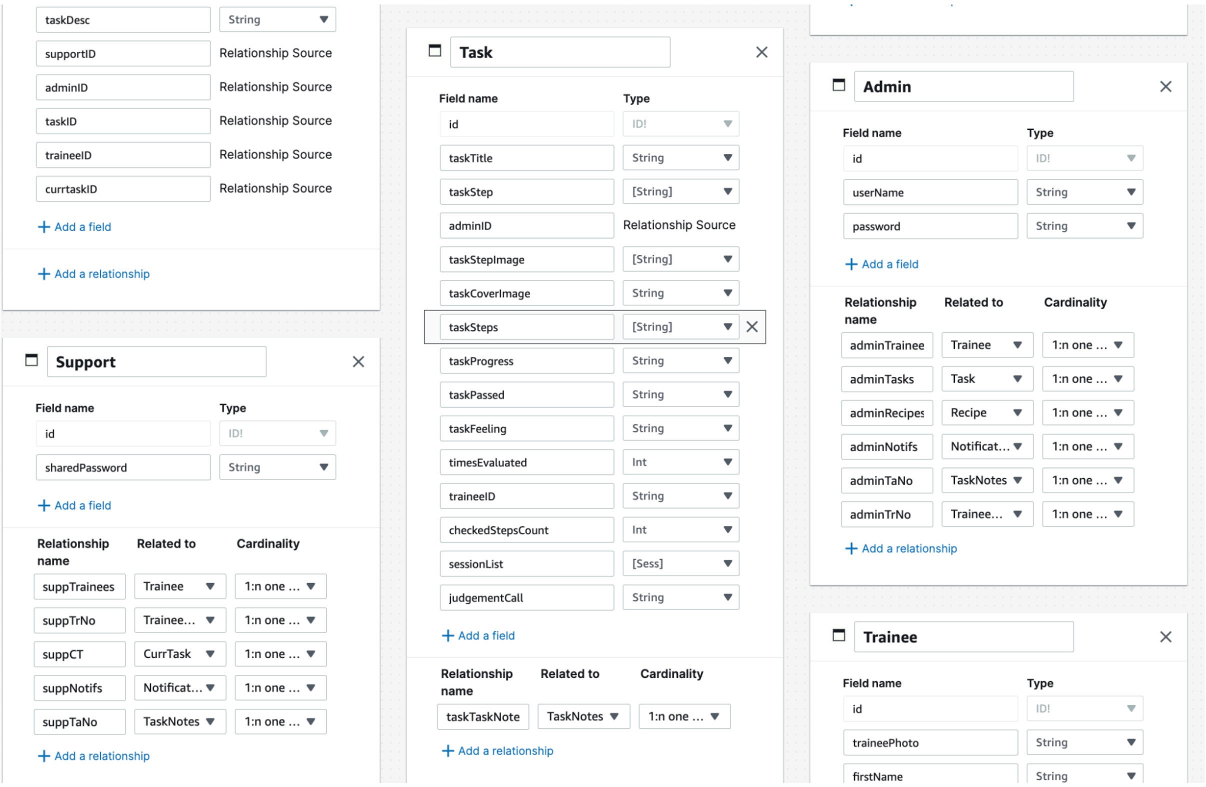
| Relationship name | Related to | Cardinality |
|---|---|---|
| traineeNotes1 | Trainee | Trainee… |

### Admin

| Field name | Type |
|---|---|
| id | ID! |
| userName | String |

### Inspector panel

*Select a model, field or relationship to configure their properties and authorization rules.*



| Field name | Type |
|---|---|
| taskDesc | String |
| supportID | Relationship Source |
| adminID | Relationship Source |
| taskID | Relationship Source |
| traineeID | Relationship Source |
| currtaskID | Relationship Source |

Support

| Field name | Type |
|---|---|
| id | ID! |
| sharedPassword | String |

| Relationship name | Related to | Cardinality |
|---|---|---|
| suppTrainees | Trainee | 1:n one … |
| suppTrNo | Trainee… | 1:n one … |
| suppCT | CurrTask | 1:n one … |
| suppNotifs | Notificat… | 1:n one … |
| suppTaNo | TaskNotes | 1:n one … |

Task

| Field name | Type |
|---|---|
| id | ID! |
| taskTitle | String |
| taskStep | [String] |
| adminID | Relationship Source |
| taskStepImage | [String] |
| taskCoverImage | String |
| taskSteps | [String] |
| taskProgress | String |
| taskPassed | String |
| taskFeeling | String |
| timesEvaluated | Int |
| traineeID | String |
| checkedStepsCount | Int |
| sessionList | [Sess] |
| judgementCall | String |

| Relationship name | Related to | Cardinality |
|---|---|---|
| taskTaskNote | TaskNotes | 1:n one … |

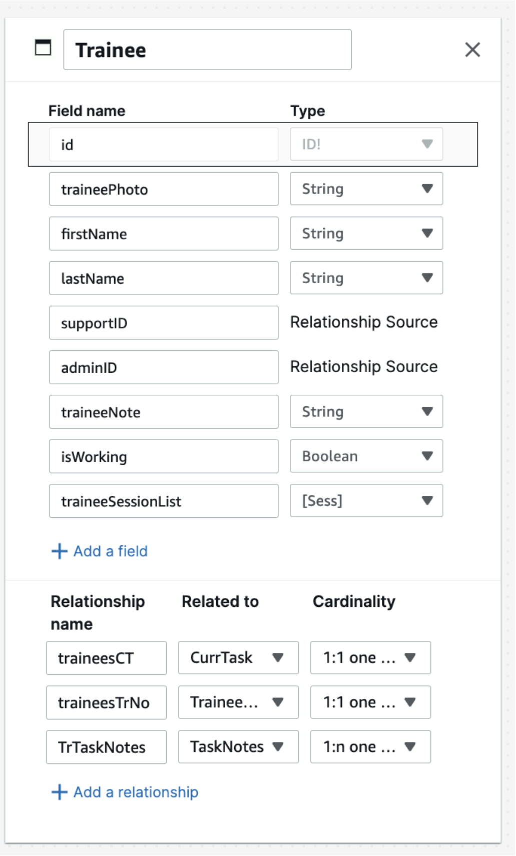
Admin

| Field name | Type |
|---|---|
| id | ID! |
| userName | String |
| password | String |

| Relationship name | Related to | Cardinality |
|---|---|---|
| adminTrainee | Trainee | 1:n one … |
| adminTasks | Task | 1:n one … |
| adminRecipes | Recipe | 1:n one … |
| adminNotifs | Notificat… | 1:n one … |
| adminTaNo | TaskNotes | 1:n one … |
| adminTrNo | Trainee… | 1:n one … |

Trainee

| Field name | Type |
|---|---|
| id | ID! |
| traineePhoto | String |
| firstName | String |



Trainee

| Field name | Type |
|---|---|
| id | ID! |
| traineePhoto | String |
| firstName | String |
| lastName | String |
| supportID | Relationship Source |
| adminID | Relationship Source |
| traineeNote | String |
| isWorking | Boolean |
| traineeSessionList | [Sess] |

| Relationship name | Related to | Cardinality |
|---|---|---|
| traineesCT | CurrTask | 1:1 one … |
| traineesTrNo | Trainee… | 1:1 one … |
| TrTaskNotes | TaskNotes | 1:n one … |

## **AWS DynamoBD**

Requires no extra set up, simply follow the screenshots above and recreate the database in amplify studio. Make sure to save and deploy.

## AWS S3

- Once signed into AWS Management Console, navigate to S3.
- Click on the bucket name to which the bucket policy should be applied.
- Click on the “Permissions” tab.
- Scroll down to the Bucket Policy section and click the “Edit” button.
- Copy paste the following the JSON policy, making sure to replace ‘YOUR BUCKET NAME’ with the name of your S3 bucket.

```
{
  "Version": "2012-10-17",
  "Id": "PolicyExample",
  "Statement": [
    {
      "Sid": "AllowGetAndPut",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::YOUR_BUCKET_NAME/*"
    },
    {
      "Sid": "DenyNonSSLRequests",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::YOUR_BUCKET_NAME",
        "arn:aws:s3:::YOUR_BUCKET_NAME/*"
      ],
      "Condition": {
        "Bool": {
          "aws:SecureTransport": "false"
        }
      }
    }
  ]
}
```

## AWS Cognito

*note you will need Administrator Access to fully implement the Authentication.

- There should be a userpool generated once Amplify has been created

- Create your users, in this case there should be one for admin and one for support workers (so two total)
- Connect a common email address to this user and a temporary password will be sent there
- Force Change the password on your terminal/command line with this prompt
- aws cognito-idp admin-set-user-password --user-pool-id '[REPLACE WITH YOUR USER POOL ID' --username '[REPLACE WITH YOUR USERNAME]' --password '[REPLACE WITH PASSWORD YOU WANT TO CHANGE TO' --permanent
- Please note that for a MAC terminal you need the quotation marks around the user pool id, username and password. This may be different for Windows Command Line
- Once the password change happens, the Confirmation Status should go from 'Force change password' to 'Confirmed'
- From here your passwords should be set up for the user

Now that you have two users - support & admin

1. You must create two groups - one is Admin and the other is Support
2. Most importantly you must create a custom attribute called 'custom:role' and make the value support for the support worker and admin for the admin worker (please ensure that the value is exactly 'support' or 'admin' or you may need to edit the codebase
3. You should be able to create the custom attributes on the user pool and then add the attributes specifically to the user when selecting the user