Development

Developing the product

Google apps script in conjunction with google spreadsheets was chosen as the best way to store and modify these group quizzes because of the universal functionality provided by google products. In the web app, the student is prompted to provide their student ID number. If they are already listed in the spreadsheet, the app will show them all of their previous grades, average grade, and name. If they are not a returning student they will be prompted to provide their full name and will be added to the spreadsheet and then sent to the Group Quiz information screen, as if they had entered their ID number and been a returning student. This screen displays their name, group quiz average grade, all previous grades, and an input field for group quiz number and grade. Once values for both fields are entered and the submit button is clicked, or the enter key is pressed, if there is not already a grade for that quiz, the quiz will be added to the spreadsheet with its grade. If there is already a grade entered in the spreadsheet the message that retake grades are not submittable will appear in place of the header where the student name is shown. Once a new group quiz has been submitted the student name header is replaced by a confirmation that the grade has been recorded into the spreadsheet.

There are two documents which provide all of the code for this project within the google apps script spreadsheet modifier page; there is the google script file, code.gs, which runs on the google apps server, and the html file, index.html, which contains all of the required client side interaction such as CSS, Javascript, and HTML interface. Within the google script file there are several main functions: The web app populator (Image 1), the function which retrieves a list of all the current IDs, the function which retrieves all student information based on their ID (Image 2), the function to add a group quiz grade, and the function to add a new student to the spreadsheet. Within the index.html file there are three parts. The first part, the HTML, is the user interface. There are 3 classes which are each a separate window, and are displayed and hidden based on the desire of the following javascript. There is the idInput class (Image 3), which is displayed upon loading. The nameInput class is displayed if the previously entered ID does not match one of the IDs already in the spreadsheet. The gqInput class is displayed after there is confirmation from either of the two previous functions that the student is present in the spreadsheet and may enter group quiz grades. The second part, the CSS, modifies all of the divs in the html so that they are centered. Bootstrap styles were used heavily within the HTML to maintain adequate aesthetic. The third and final part of the index.html file is the javascript, where most of the work is done in the project.

Within the javascript code there are many functions which serve different purposes. Various buttons which correspond to their functions are present in the various classes in the html code. jQuery was used extensively to modify the appearance of the HTML to the user. The main functions present in the javascript are the addStudent, add GQ, and GetUserInfo, and id Check functions. The addStudent function calls one of the functions in the code.gs file to add a new student to the spreadsheet. The addGq function adds a gq to the student with a certain student ID number. The get id function receives an array of all of the current students to determine whether the student attempting to access the app is new or already has quizzes entered.

Image 1:

```
function doGet() {
  return HtmlService.createTemplateFromFile('EnterId').evaluate().setTitle('Group Quiz Entry System').setSandboxMode(HtmlService.SandboxMode.IFRAME);
              }
```

Image 2:

```
function getUserInfoInArray(column){
  Logger.log(typeof column, column)
 var ss = SpreadsheetApp.getActiveSpreadsheet();
  var sheet = ss.getSheets()[0];
  var arrayOfInfo = new Array();
  var lastRow = parseInt(lastValue(column))
  arrayOfInfo.push(sheet.getRange(1,column,lastRow).getValues().toString());
return arrayOfInfo

}
```

Image 3:

```
<div class="all">
  <idInput >
    <base target="_top">
    <div class="container">
      <div class="form-group row">
        <h1>IBESS Group Quiz Recording System</h1>
        <br>
        <label for="input">Student ID Number</label>
        <div class="width">
          <input   type="text" pattern="[0-9]*" maxlength="7" class="form-control" style="width: 300px" style="height: 300px" id="input" placeholder="0000000..."> </di
        <div class="form-group row">
        <br>
          <div class="offset-sm-2 col-sm-10">
            <button id="idSub" class="btn btn-primary" onclick="runid()">Submit</button>
          </div>
        </div>
      </div>
    </div>
```