McDonalds eLearning Application

DESIGN MANUAL

William Nolan | C00216986 | November 1st, 2019 | Chris Meudec Date

Contents

1. Introduction

The aim of this project is to create a mobile and web application that enables McDonald's employees to complete their onboarding and continuous training through gamified eLearning rather than paper form *Station Observation Checklists* (SOCs). The application would also provide additional functionality for Crew Trainers and Managers to oversee employees' training progress and results.

The purpose of this document is to describe how McDonald's eLearning will work. This document is a continuation of the functional specification, where the application's functionalities were described.

This document will include the following diagrams:

1. High Level Diagram
2. Domain Model Diagram
3. Design Class Diagram
4. System Sequence Diagram
5. Sequence Diagram

This document will also outline the basic user interface design and the database implementation. This design document should enable any other developer to implement the functionalities as described successfully.

2. High Level Diagram

3. Domain Model Diagram

leaderboard

date
name
score

soc

name
description
percent

question

name

has ►

1   1   1   1..n

user

email
fname
lname

result

date
result
total
incorrect

answer

name
isAnswer

0..n

has ►

1

1   has ►   0..n

1

has ►

1..n

1

has ►

1..n

1

has ►

1..4

feedback

feedback
senderName
date

4. Design Class Diagram

```
                                answer                                                                    question
+------------------------------------------------------------+                         +------------------------------------------------------------------------------+
| -id: string                                                |                         | -id: string                                                                  |
| -name: string                                              |                         | -name: string                                                                |
| -isAnswer: Boolean                                         |        has ►            +------------------------------------------------------------------------------+
+------------------------------------------------------------+      1..4        1       | +socQuestions(): object                                                      |
| +socAnswers(): object                                      |                         | +fetchQuestions(socId: string): object                                       |
| +fetchAnswers(socId: string, questionId: string): object  |                         | +getQuestion(socId: string, questionId: string): object                      |
| +createAnswer(socId: string, questionId: string, name:    |                         | +createQuestion(socId: string, name: string, answers: any[]): object         |
|   string, isAnswer: boolean): object                       |                         | +updateQuestion(socId: string, questionId: string, name: string,             |
| +updateAnswer(socId: string, questionId: string,          |                         |   answers: any[]): object                                                    |
|   answerId: string, name: string, isAnswer: boolean): obj  |                         | +deleteQuestion(socId: string, questionId: string): object                   |
| +deleteAnswer(socId: string, questionId: string,          |                         +------------------------------------------------------------------------------+
|   answerId: strin): object                                 |
+------------------------------------------------------------+
```

1..n

has ►

1

```
                         leaderboard                                                                       soc
+----------------------------------------------------+                  +------------------------------------------------------------------------------------+
| -id: string                                        |    has ►         | -id: string                                                                        |
| -name: string                                      |   1        1     | -name: string                                                                      |
| -score: number                                     |                 | -description: string                                                               |
| -date: Date                                        |                 | -percent: number                                                                   |
+----------------------------------------------------+                 +------------------------------------------------------------------------------------+
| +leaderboard(): object                             |                 | +socs(): object                                                                    |
| +addLeaderboard(socId: string, score: number): obj |                 | +fetchSocs(): object                                                               |
| +fetchLeaderboard(socId: string): object           |                 | +getSoc(id: string): object                                                        |
| +compareScores(socId: string, name: string): void  |                 | +createSoc(name: string, description: string, percent: number, questions: any[]): object |
+----------------------------------------------------+                 | +updateSoc(socId: string, name: string, description: string, percent: number, questions: any[]): object |
                                                                        | +deleteSoc(socId: string): object                                                  |
                                                                        +------------------------------------------------------------------------------------+
```

0..n

has ►

1

```
                            user                                                                            result
+------------------------------------------------------------+             has ►       +------------------------------------------------------------------------------+
| -id: string                                                |           1      0..n   | -id: string                                                                  |
| -fname: string                                             |                         | -date: Date                                                                  |
| -lname: string                                             |                         | -result: number                                                              |
| -email: string                                             |                         | -total: number                                                               |
+------------------------------------------------------------+                         | -incorrect: string[]                                                         |
| +userIsAuthenticated(): Boolean                            |                         +------------------------------------------------------------------------------+
| +userId(): string                                          |                         | +results(): object                                                           |
| +currUser(): object                                        |                         | +fetchResults(userId: string, socId: string): object                         |
| +users(): object                                           |                         | +addResult(userId: string, socId: string, result: number, total: number,     |
| +fetchUsers(): object                                      |                         |   incorrect: string[]): object                                               |
| +getUser(id: string): object                               |                         | +getResult(id: string, socId: string, userId: string): object                |
| +signUp(email: string, password: string): object          |                         +------------------------------------------------------------------------------+
| +createUser(userId: string, email: string, fname: string, |
|   lname: string, role: number): object                     |
| +login(email: string, password: string): object           |
| +updateCurrUser(id: string): object                        |
| +logout(): void                                            |
| +private setUserData(userData: AuthResponseData): void     |
+------------------------------------------------------------+
```

1

has ►

1..n

```
                            feedback
+------------------------------------------------------------------------------+
| -id: string                                                                  |
| -feedback: string                                                            |
| -senderName: string                                                          |
| -date: Date                                                                  |
+------------------------------------------------------------------------------+
| +feedback(): object                                                          |
| +fetchResults(userId: string, socId: string): object                         |
| +addFeedback(feedback: string, senderName: string, userId: string,           |
|   socId: string, resultId: string): object                                   |
| +getResultObject(userId: string): object                                     |
+------------------------------------------------------------------------------+
```

5. Project Structure

The application makes use of different project files to handle different functionalities within the project.

5.1. MODELS

Models are used to represent classes in this project. The class is defined within the model with its different attributes and exported for use throughout the project. This allows the use of custom objects, such as a SOC object, throughout the project.

5.2. SERVICES

Services are used to handle communication between the application and Firebase (the back end). For organisational purposes, most of the services handle communication for a specific object, i.e. the SOCs service. Services handle fetching, posting, updating and deletion of data in Firebase.

There is one service, the question service, that is solely used for keeping track of values and completing tasks throughout the Take SOC process. It does not communicate with the back end.

6. Sequence Diagram

6.1. SOCS SERVICE
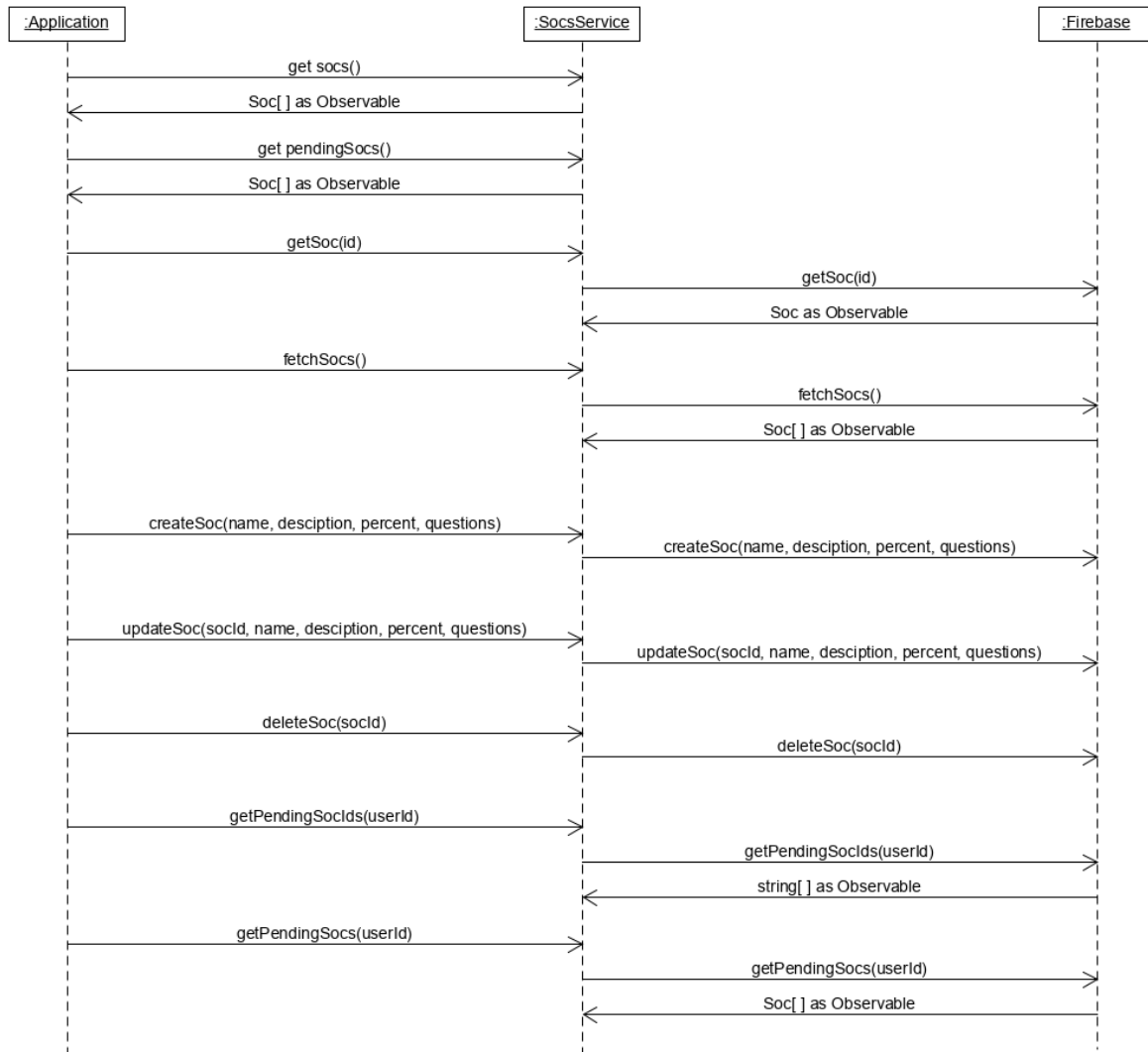
The SOCs Service is used to handle all communication to the back end that relates to the Soc object. The application calls the service method which makes a request using Angular's HttpClient module.

As an example, we will look at the `fetchSocs()` method. This method is used to fetch all the SOCs from the back end. The application calls this method and the service sends a get request to the back end. If the request is successful, it will return a list of SOCs and store it as a `BehaviorSubject` within the service. The application can then access this list of SOCs by calling `get soc()` and subscribing to it. All other services follow a similar process, aside from the Question Service.

6.2. QUESTION SERVICE

```
        :Application                              :QuestionService
             │                                          │
             │─────────────────reset()─────────────────▶│
             │                                          │
             │                                          │
             │──addIncorrectQuestion(questionID: string, questionName: string)──▶│
             │                                          │
             │                                          │
             │───────────getIncorrectQuestions()───────▶│
             │◀─────────────────string[ ]───────────────│
             │                                          │
             │                                          │
             │────────getFinalIncorrectQuestionIDs()───▶│
             │◀─────────────────string[ ]───────────────│
             │                                          │
             │                                          │
             │───────getFinalIncorrectQuestionNames()──▶│
             │◀─────────────────string[ ]───────────────│
             │                                          │
             │                                          │
             │───────────removeIncorrectQuestion()─────▶│
             │                                          │
             │────────────────isFirstRun()─────────────▶│
             │◀─────────────────boolean──────────────────│
             │                                          │
             │────────────────firstRunDone()───────────▶│
             │                                          │
             │────────────────addResult()──────────────▶│
             │                                          │
             │────────────────getResult()──────────────▶│
             │◀─────────────────number───────────────────│
             │                                          │
             │────────────────addProgress()────────────▶│
             │                                          │
             │────────────────getProgress()────────────▶│
             │◀─────────────────number───────────────────│
             │                                          │
             │────────────────addScore(bonus)──────────▶│
             │                                          │
             │────────────────getScore()───────────────▶│
             │◀─────────────────number───────────────────│
             │                                          │
             │                                          │
             │────────────────resetStreak()────────────▶│
             │                                          │
```

The Question Service is used to handle all actions for the Take SOC process. This service allows the application to keep track of values in between questions during the process. It stores the user's result, score, any incorrect questions, and other key values. This is the only service that differs from the rest.

7. User Interface

The front end of McDonalds eLearning needed to provide an appealing, rich experience for end users.

It was developed using web technologies:

1. Ionic Framework, as mentioned before its custom components are styled very well

2. CSS, particularly used to customise some of Ionic components to give a unique style to the application

As the application has the potential to display large amounts of information it is important that the screen is not too cluttered and is clearly presented to the user.

It is also important that the application carries the same style throughout. Components of similar functionalities should carry a similar style.

Finally, navigating the application should be simple. This would be achieved through the ease of access of the 'hamburger' menu throughout the application.

7.1. SCREENFLOW

7.1.1. Login

The login page is clear and concise. All that is required to be captured here is the user's email and password.
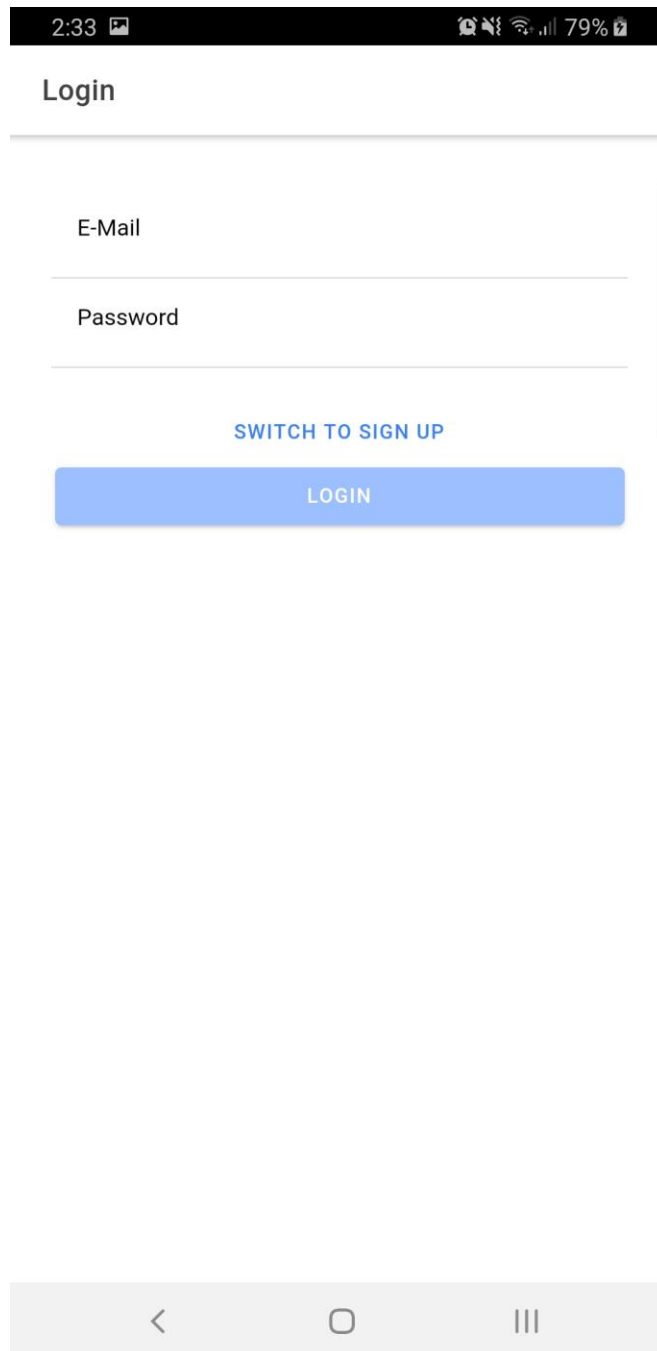
7.1.2. Sign Up

The signup page, similar to the login page, is clear and concise. All that is required to be captured here is the user's first name, last name, email, and password.
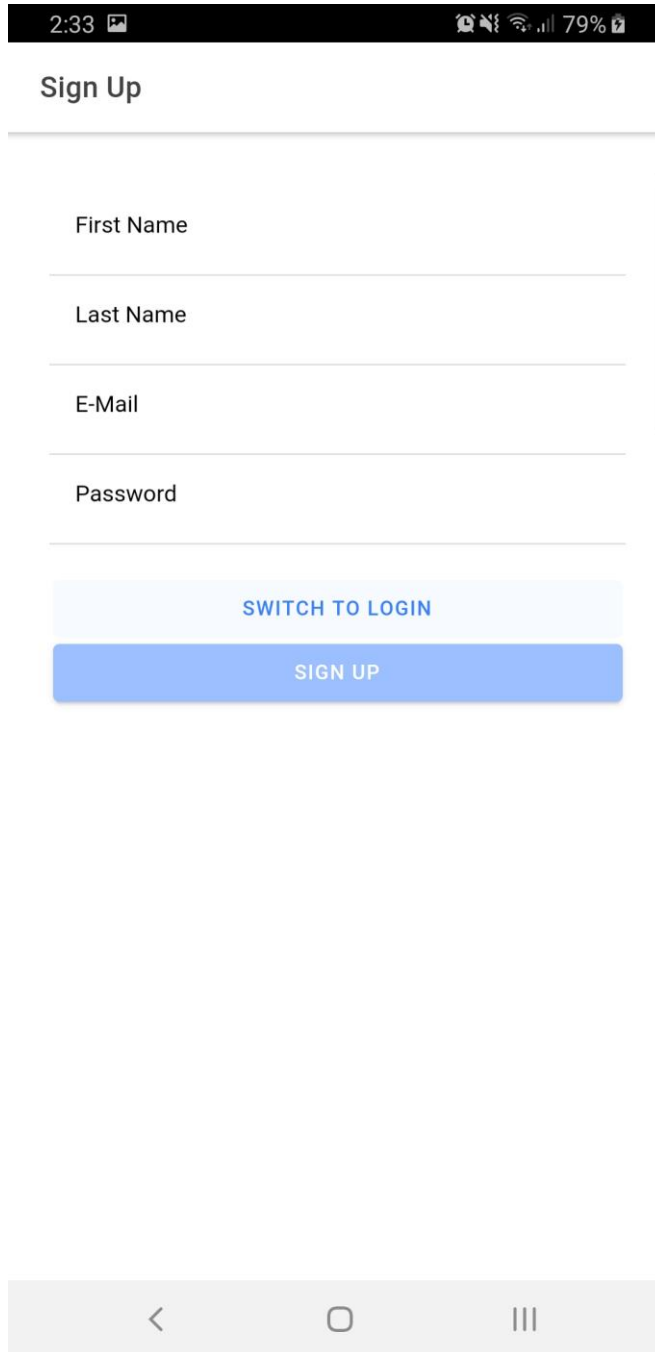
7.1.3. Navigation Menu

The Navigation menu is displayed in the form of a 'hamburger' menu. This is the user's main tool for navigating throughout the application. Each element of the menu is represented with a corresponding relative icon.
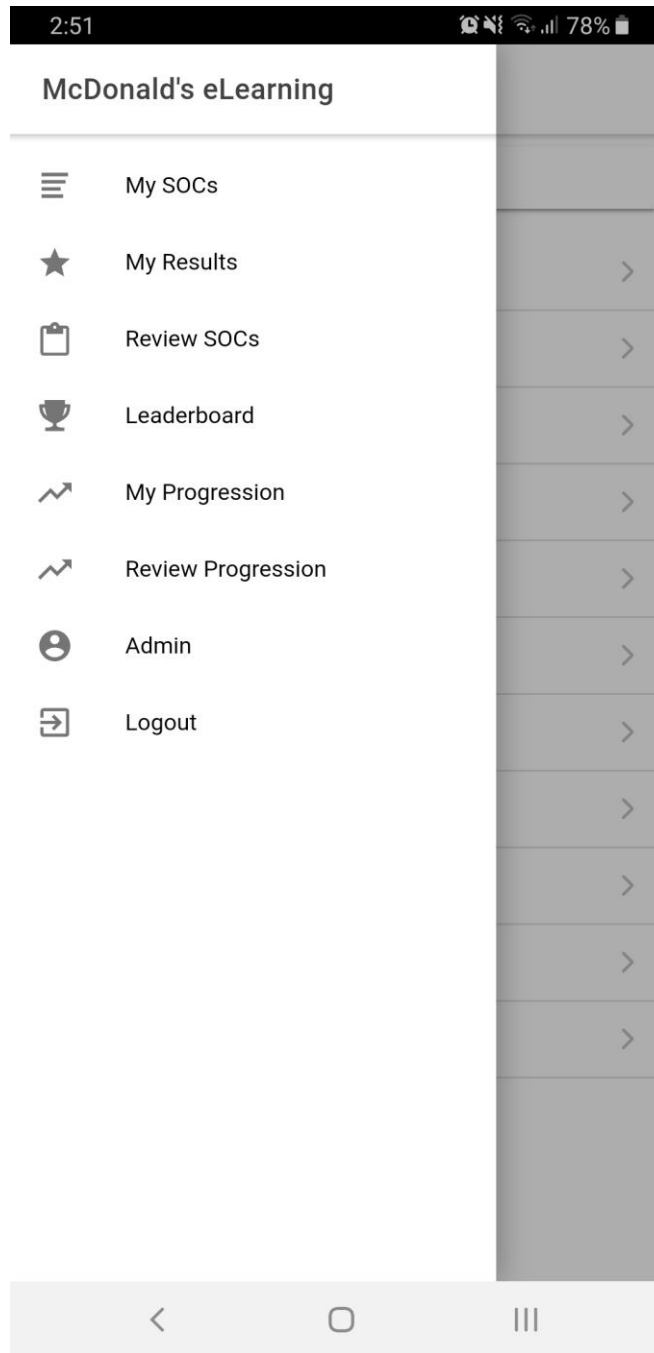
7.1.4. View Pending SOCs

Upon successful login, the user is directed to the Pending SOCs page. This page displays the SOCs the user has not completed within the last six months.
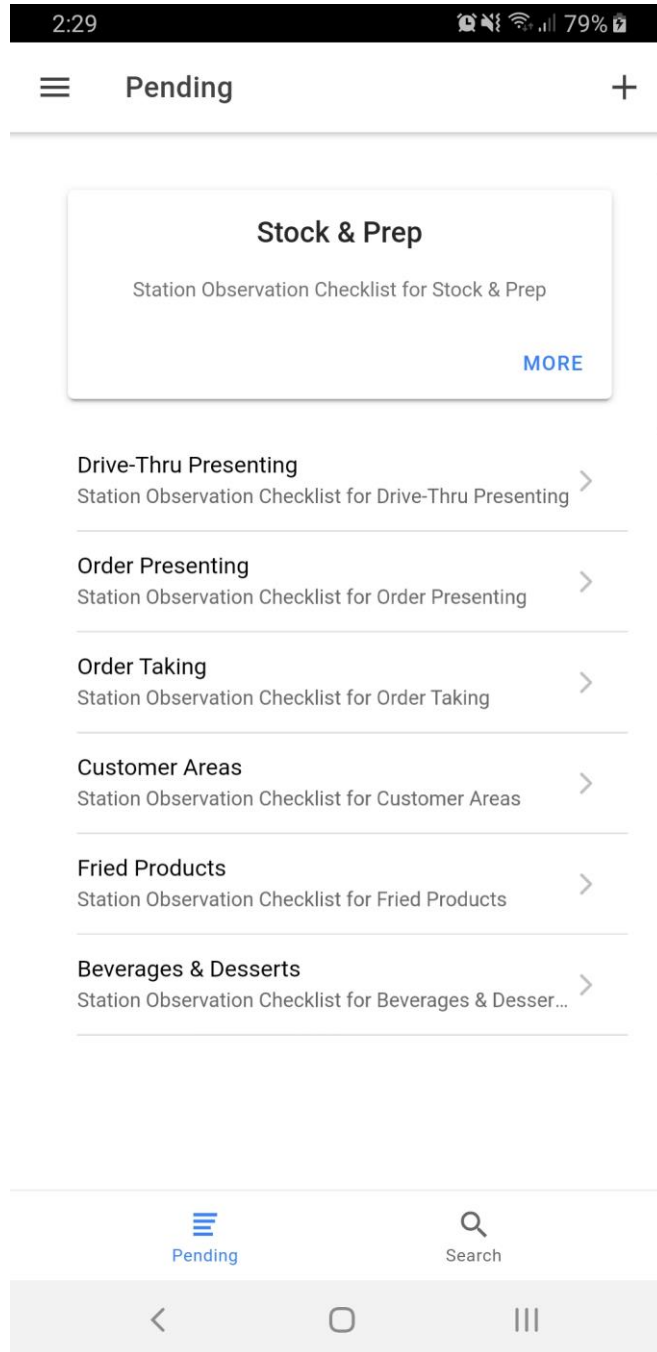
7.1.5. Search SOCs

The Search SOCs page displays all available SOCs. The search bar allows users to search for a specific SOC.
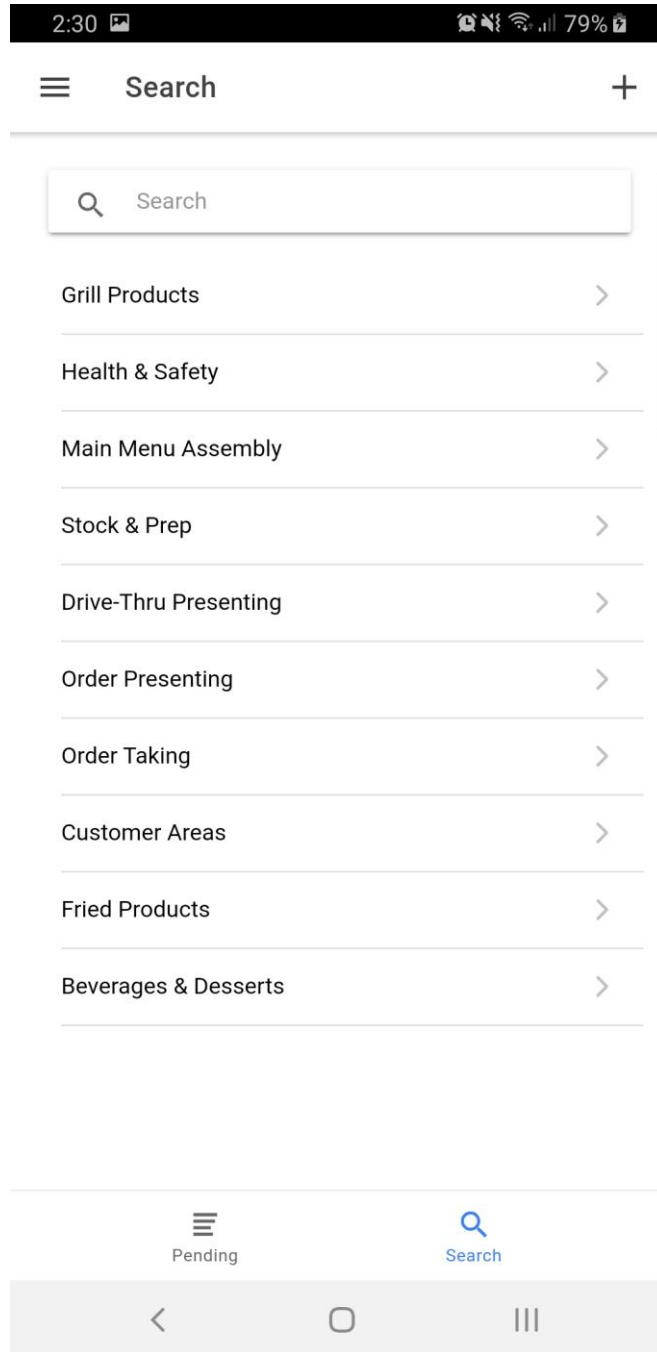
7.1.6. Create SOC (Managers ONLY)

The Create SOC page is accessible by Managers only through the '+' button at the top right of the Pending SOCs and Search SOCs page. The Create SOC page contains a dynamic form that allows the user to add additional questions and answers as they see fit. The form is submitted by clicking the 'checkmark' button at the top right of the screen.

7.1.7. View SOC

The View SOC page is accessed by clicking on the desired SOC from the Pending SOCs or Search SOCs page. This page displays more information about the SOC such as its related questions.
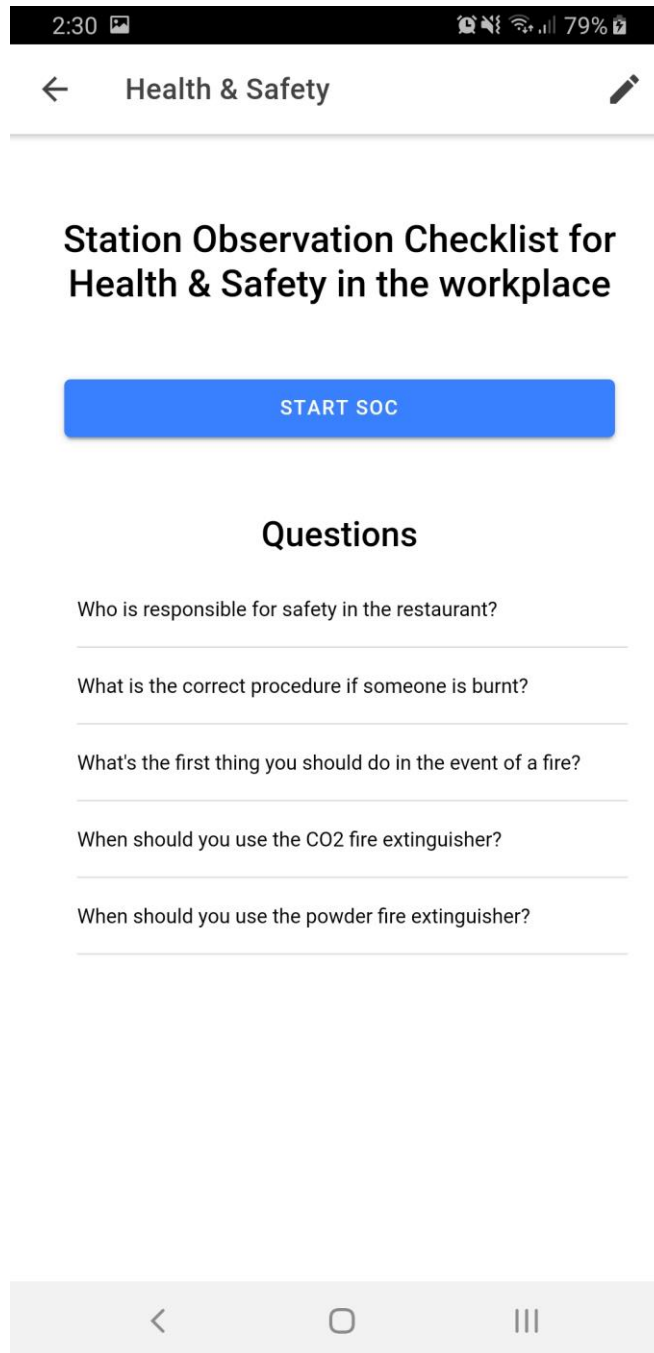
7.1.8. Update/Delete SOC (Managers ONLY)

The Update/Delete SOC page is accessible by Managers only through the 'pencil' icon at the top right of the View SOC page. The page displays all the information about the SOC in a form format. When the page is initially loaded all fields are read only to prevent any accidental edits the user may make. To toggle editing, the user can click the 'pencil' icon at the top right of the page. The page then resembles the Create SOC page and allows the user to edit existing information or add additional questions and answers. To confirm any changes made the user can click the 'checkmark' button at the top right of the screen. The user also has the ability to delete the SOC by clicking the 'trash bin' icon at the top right of the screen.
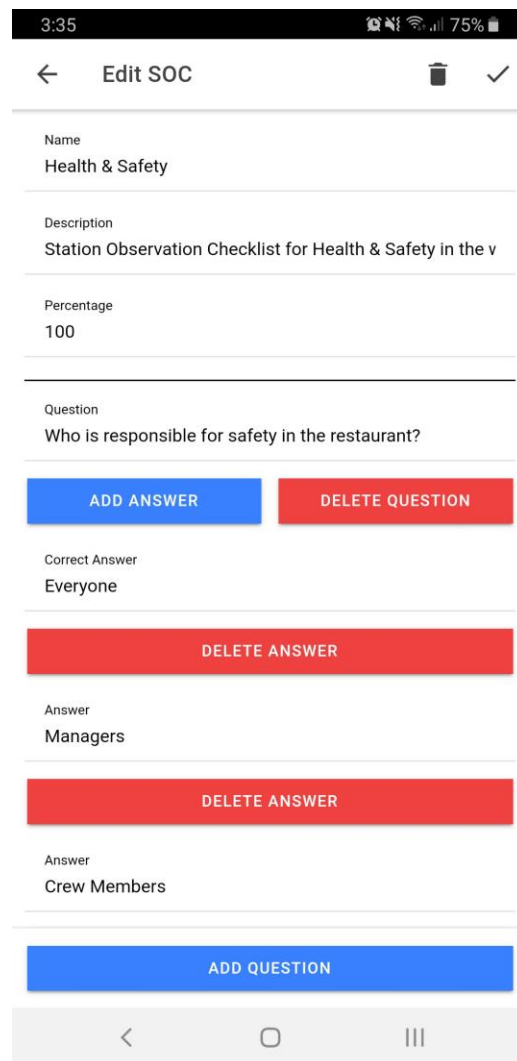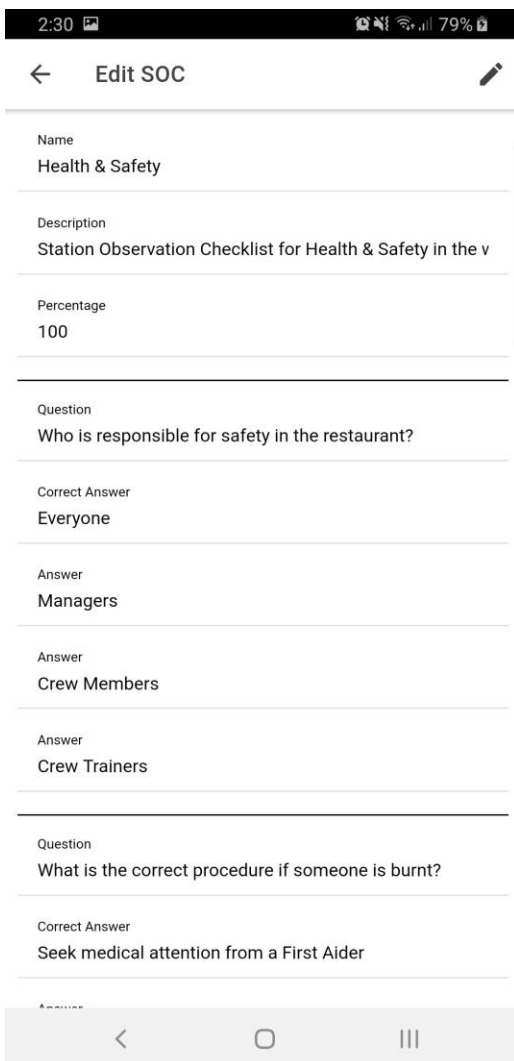
7.1.9. Take SOC

The Take SOC page is accessed by clicking the 'Start SOC' button on the View SOC page. This page represents the Take SOC process. The page displays the SOC question, its answers and SOC progression. Upon clicking the correct answer, the button turns green and a pop-up shows at the bottom of the screen allowing the user to continue. The progression bar is also incremented. Upon clicking the incorrect answer, the button turns red and a pop-up shows at the bottom of the screen informing the user of the correct answer and allowing them to continue. The progression bar does not increment.
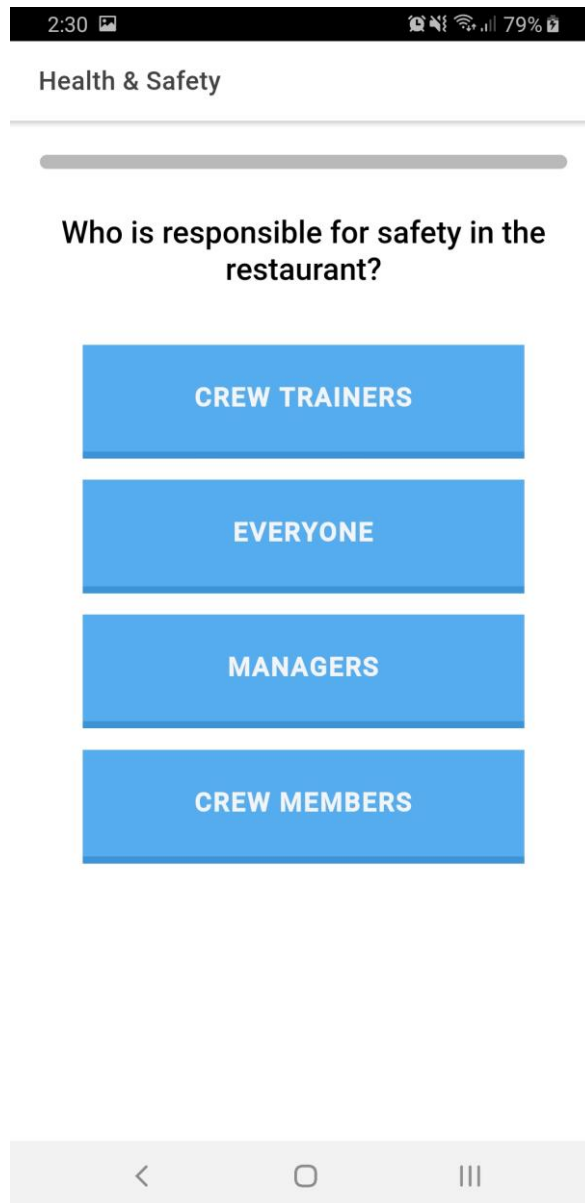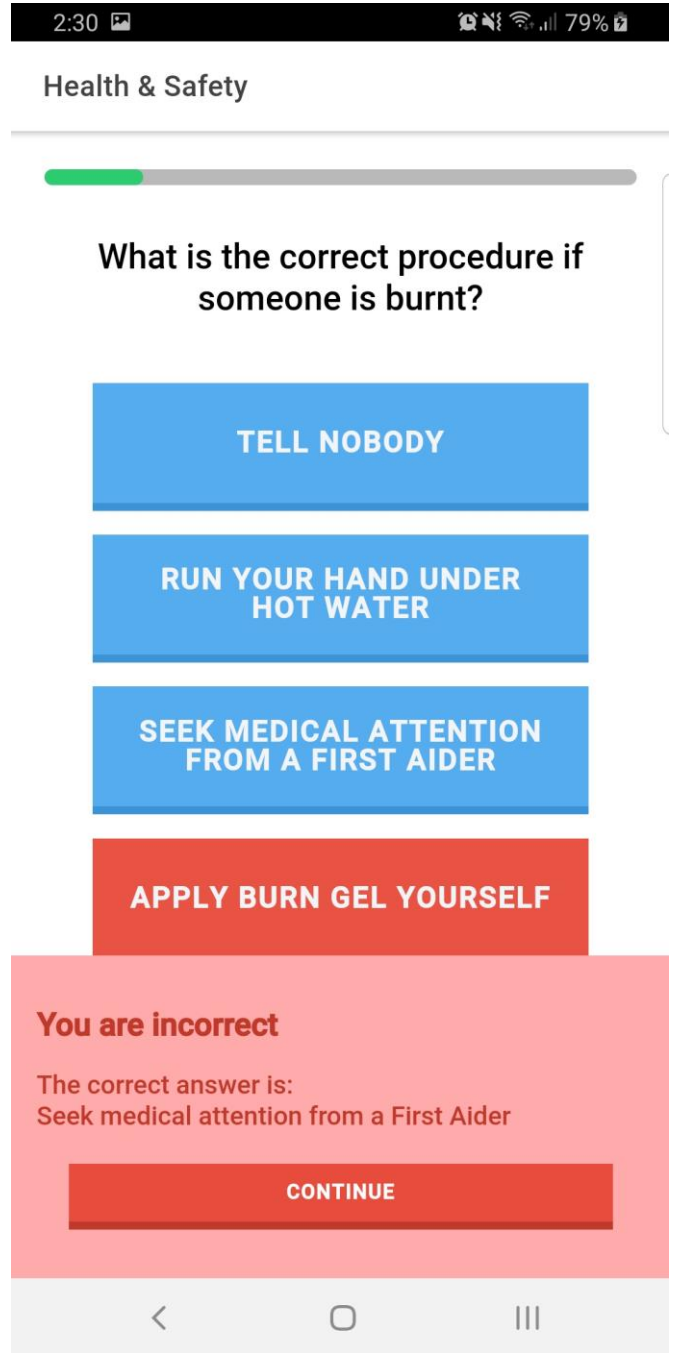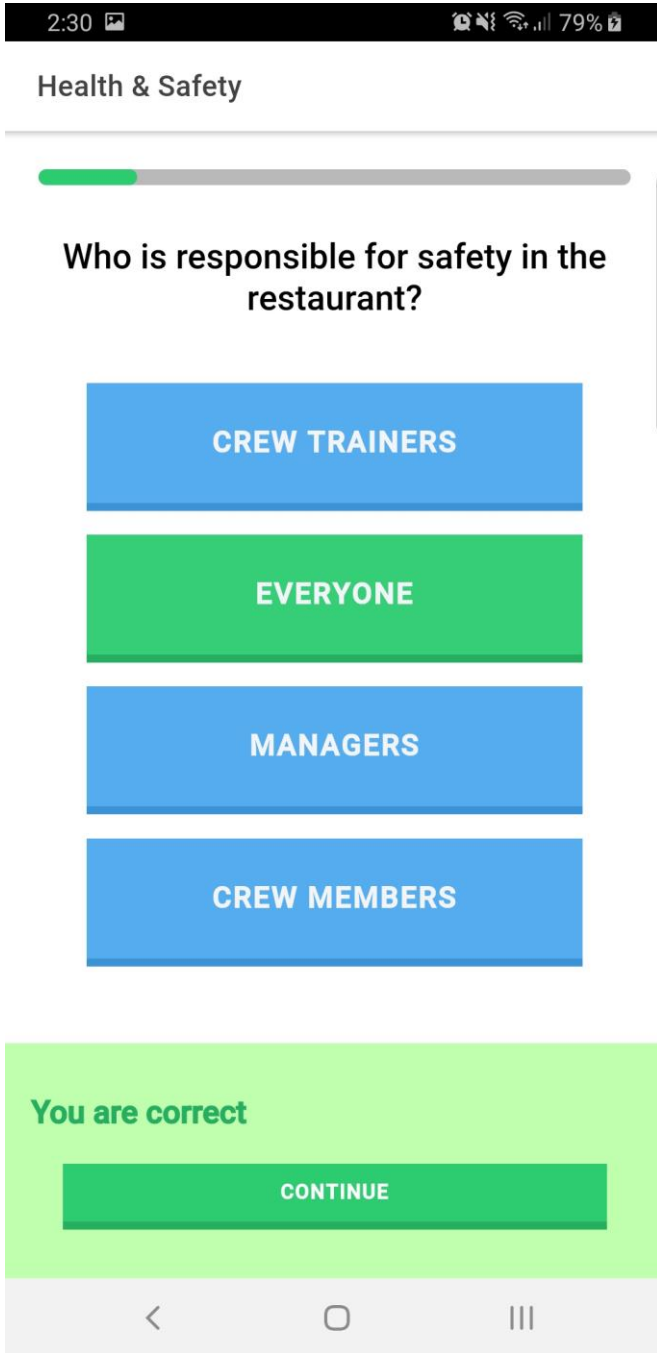
7.1.10. SOC Results

The SOC Results page is accessed when the user completes the SOC process. This page displays the users final result and any incorrect questions they may have had.
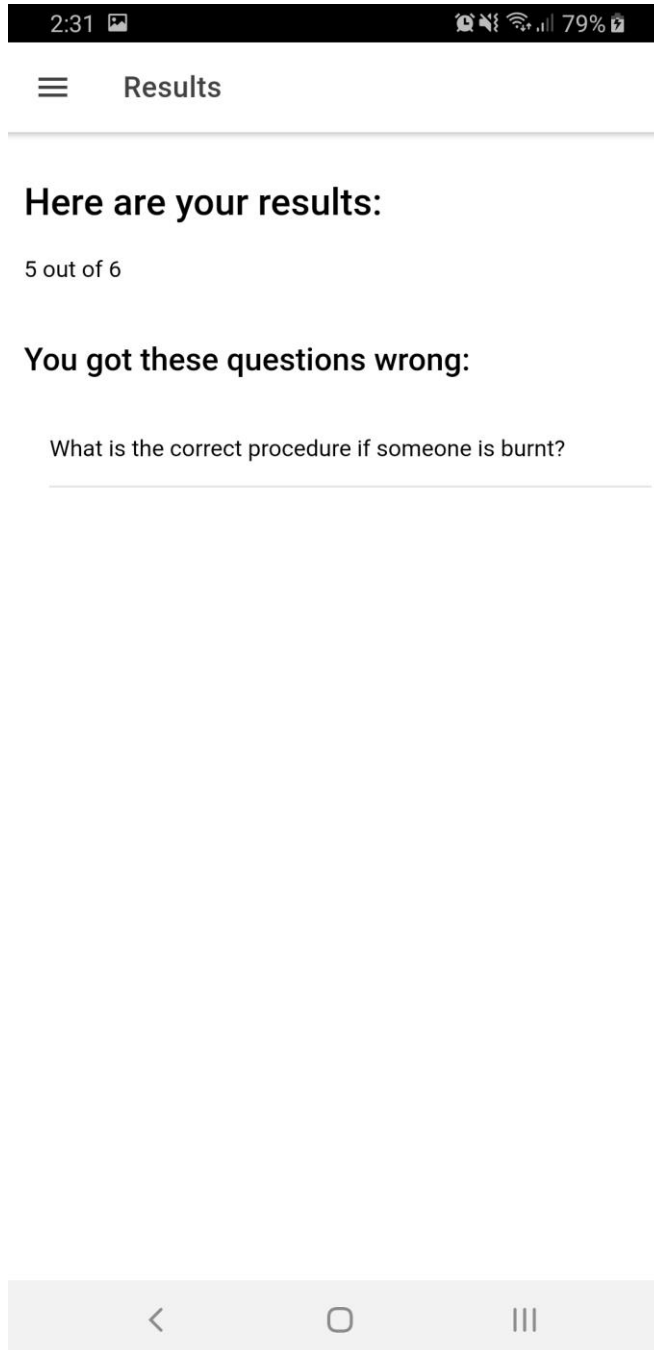
7.1.11. My Results

The My Results page is accessed through the 'hamburger' menu. It displays a list of SOCs that the user has results for. The user can view more results for a specific SOC by clicking on it. The results page displays all the user's results in a table format. The user can click on a result to view more information about it. The result details page displays the user's result in a donut graph along with any feedback they may have received on this result.
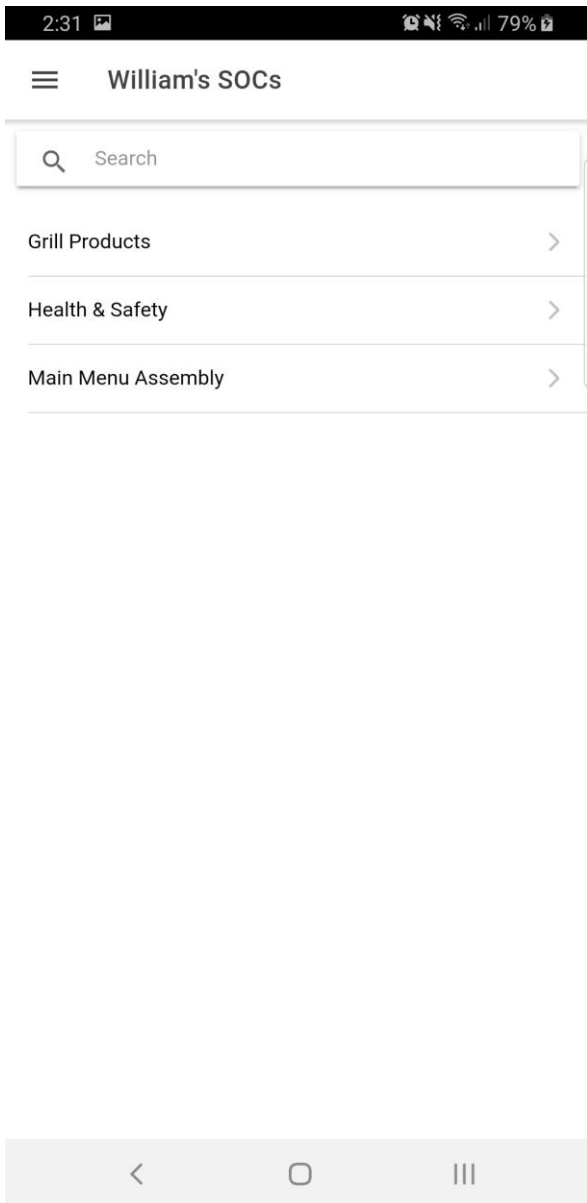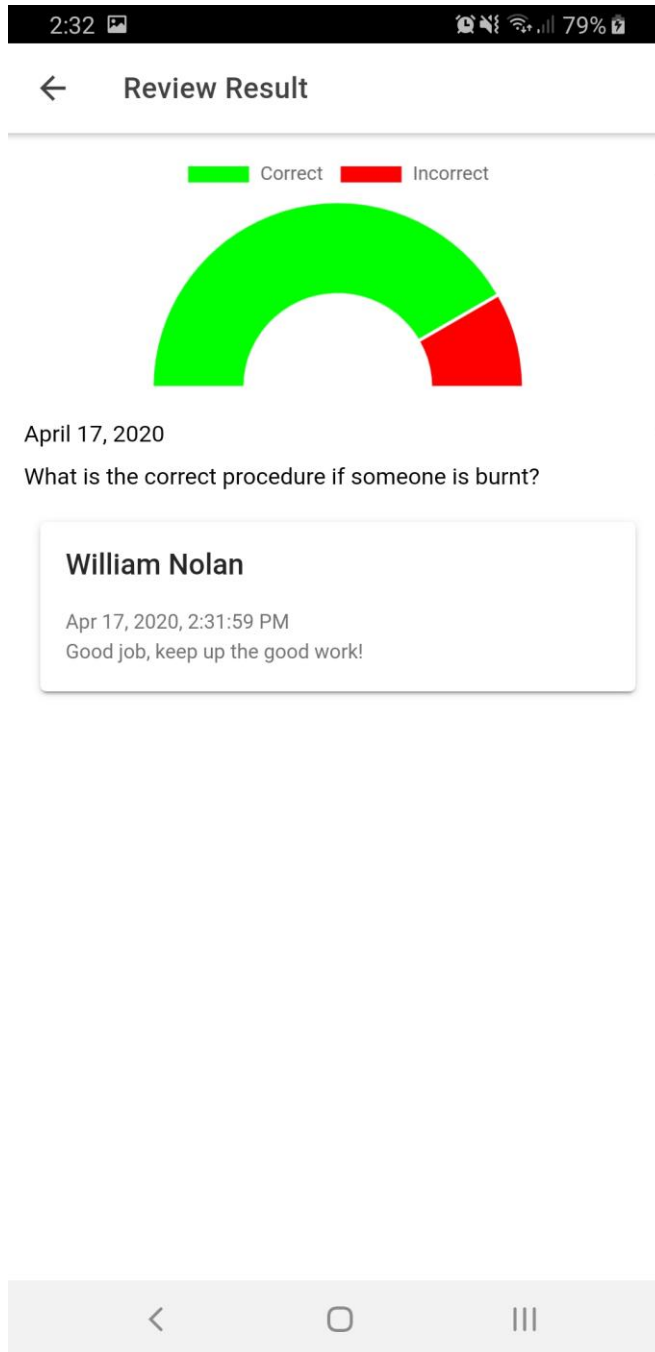
7.1.12. Review SOCs (Crew Trainers and Managers ONLY)

The Review SOCs page is accessible by Crew Trainers and Managers through the 'hamburger' menu. This screen displays a list of all users with a search bar for the user's convenience. The user can click on a user to view their results. The results are displayed in a table format, similar to the My Results screen. Upon clicking on a specific result, the user is directed to the results detail page where they can view more information about the record and leave their feedback.

Date	Wrong Answers	Result(%)
April 17, 2020	1	83%
April 15, 2020	1	83%
March 27, 2020	0	100%
March 15, 2020	2	67%
January 15, 2020	2	67%
December 15, 2019	3	50%
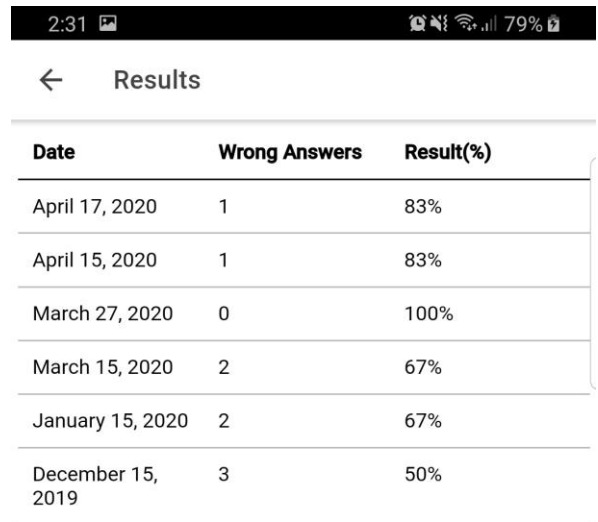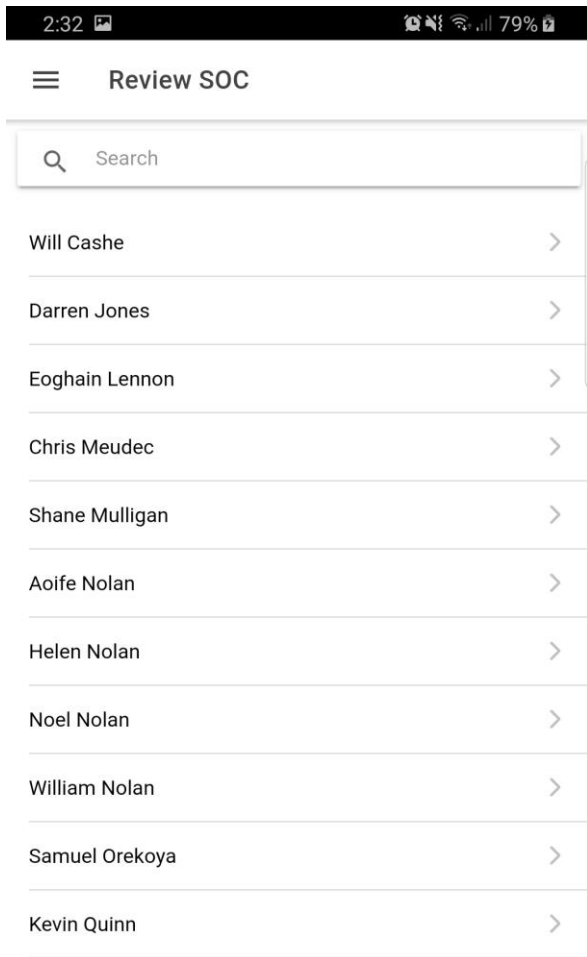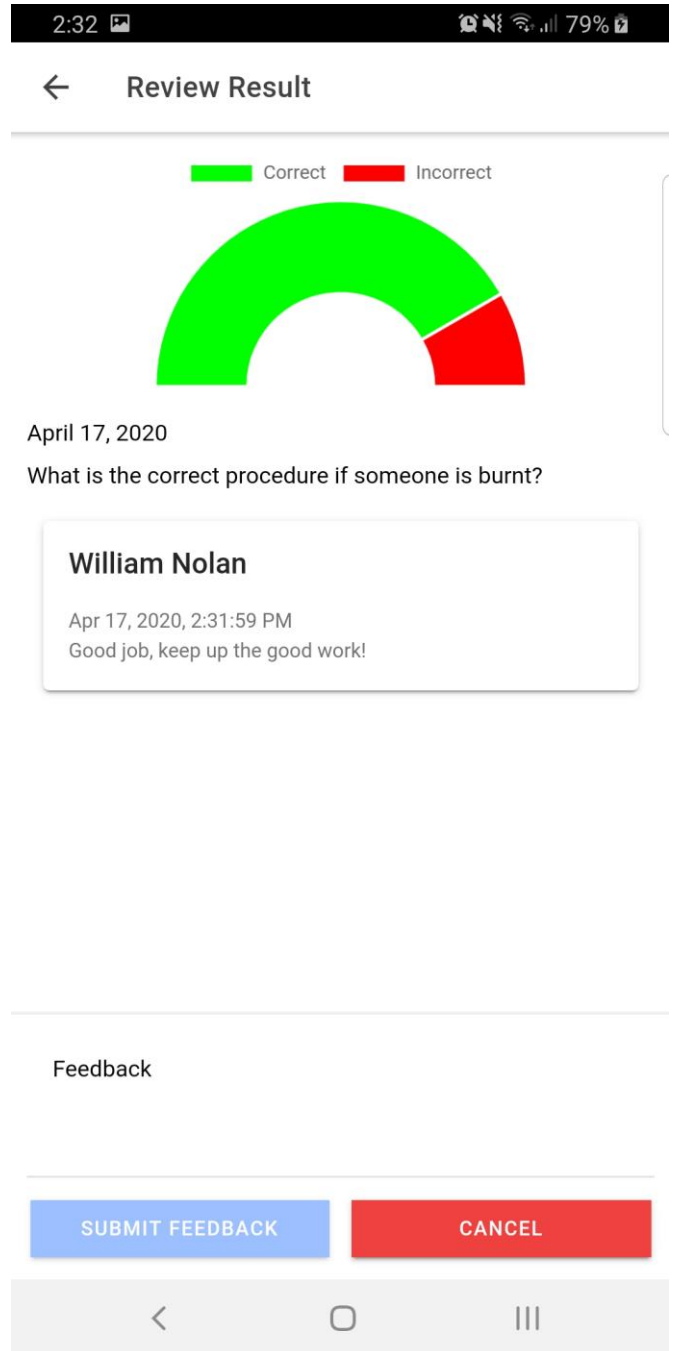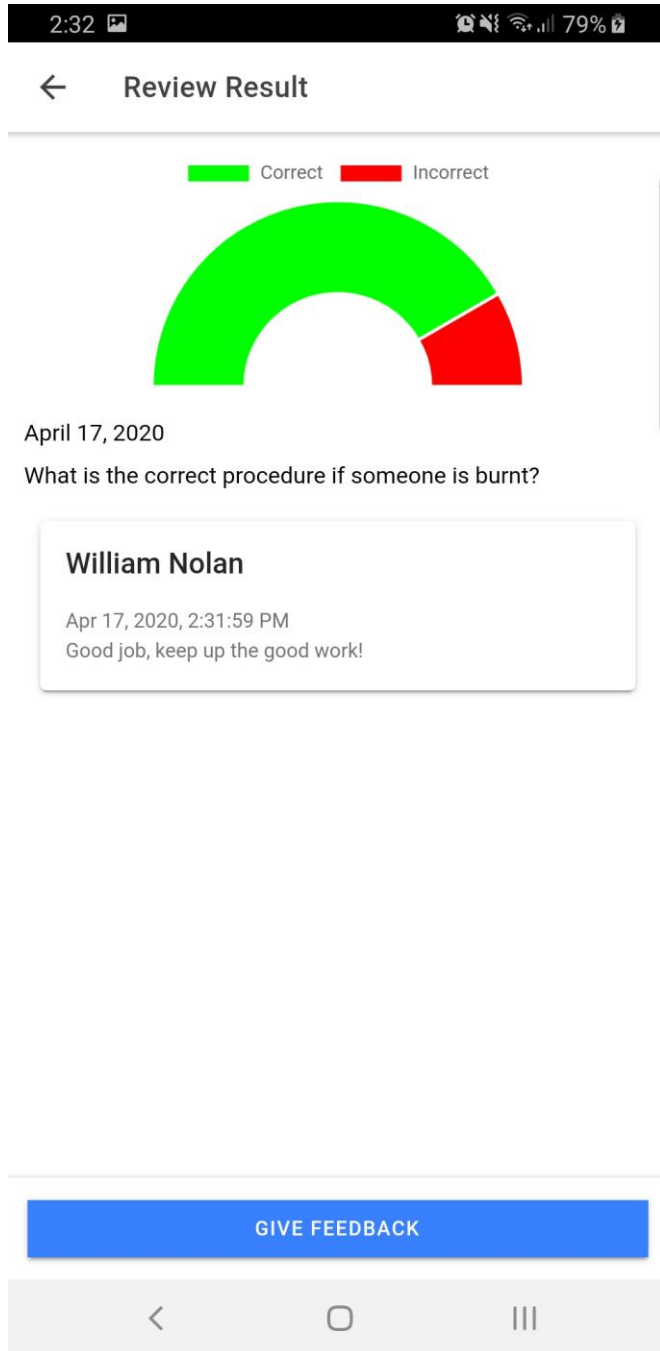
7.1.13. Leaderboard

The Leaderboard page is accessed through the 'hamburger' menu. The page displays a list of all SOCs and a search bar for the user's convenience. Upon clicking a specific SOC, the user is redirected to the SOCs Leaderboard. The leaderboard is display in a table format.
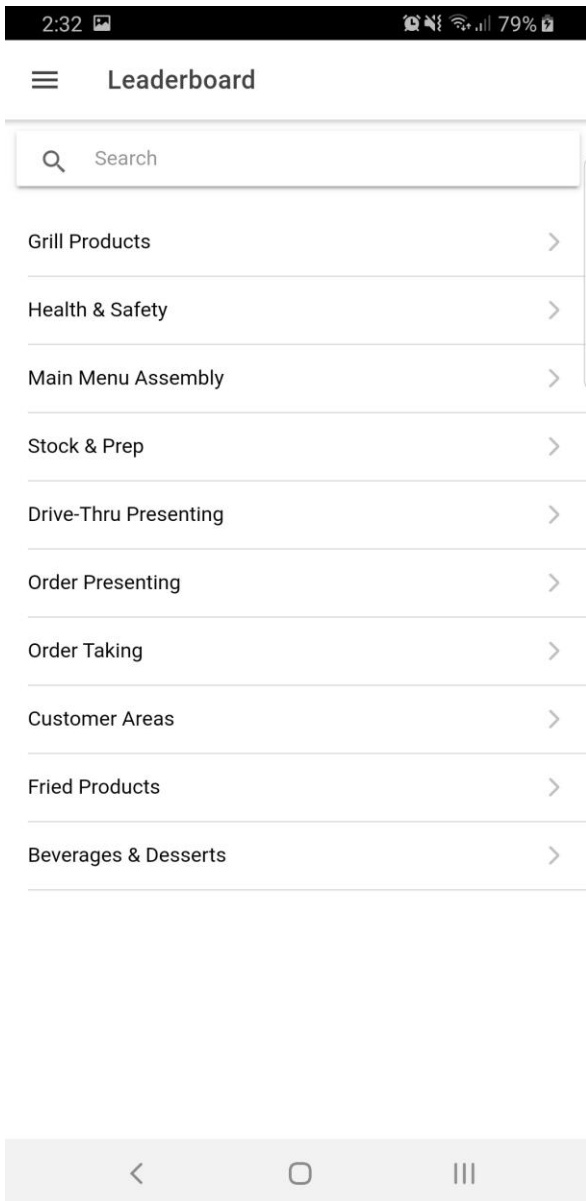
7.1.14. My Progression

The My Progression page is accessed through the 'hamburger' menu. This page displays the user's SOC results in a line graph format. The user can click on the SOC label to hide and unhide its respective line.
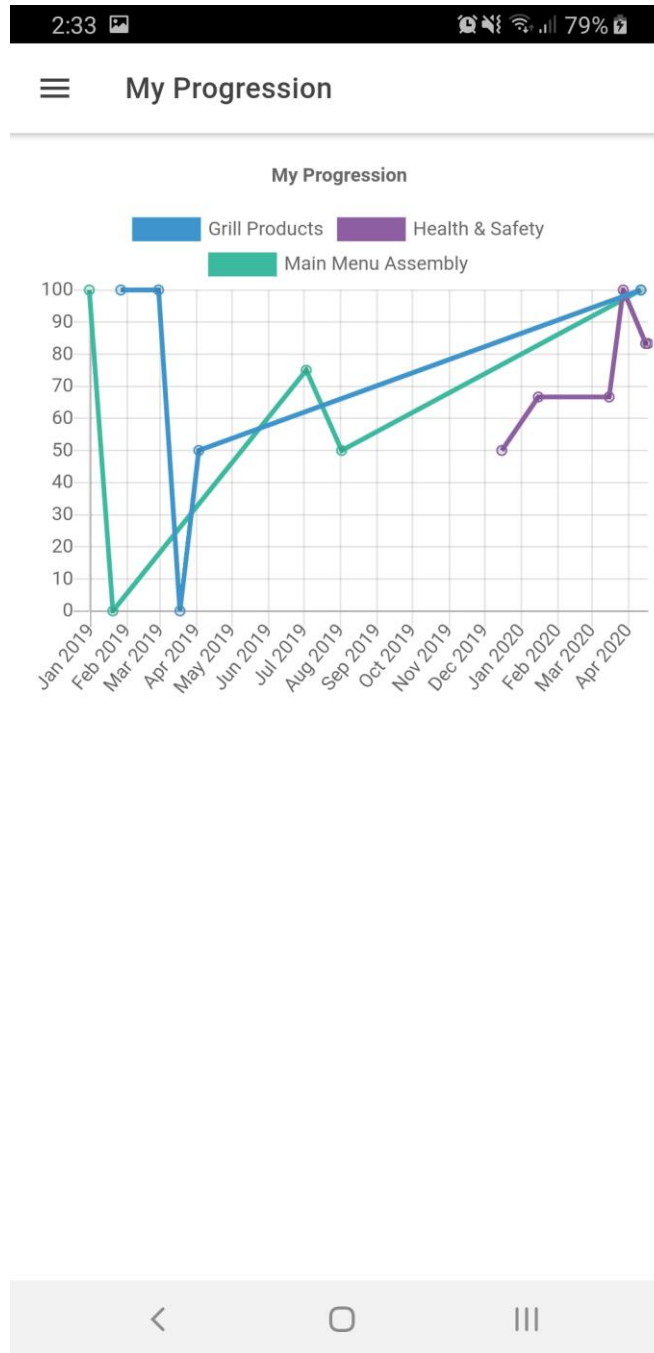
7.1.15. Review Progression (Crew Trainers and Managers ONLY)

The Review Progression page is accessible by Crew Trainers and Managers through the 'hamburger' menu. This screen displays a list of all users with a search bar for the user's convenience. The user can click on a user to view their progression. The page displays the user's SOC results in a line graph format. Similar to the My Progression page, the user can click on the SOC label to hide and unhide its respective line.
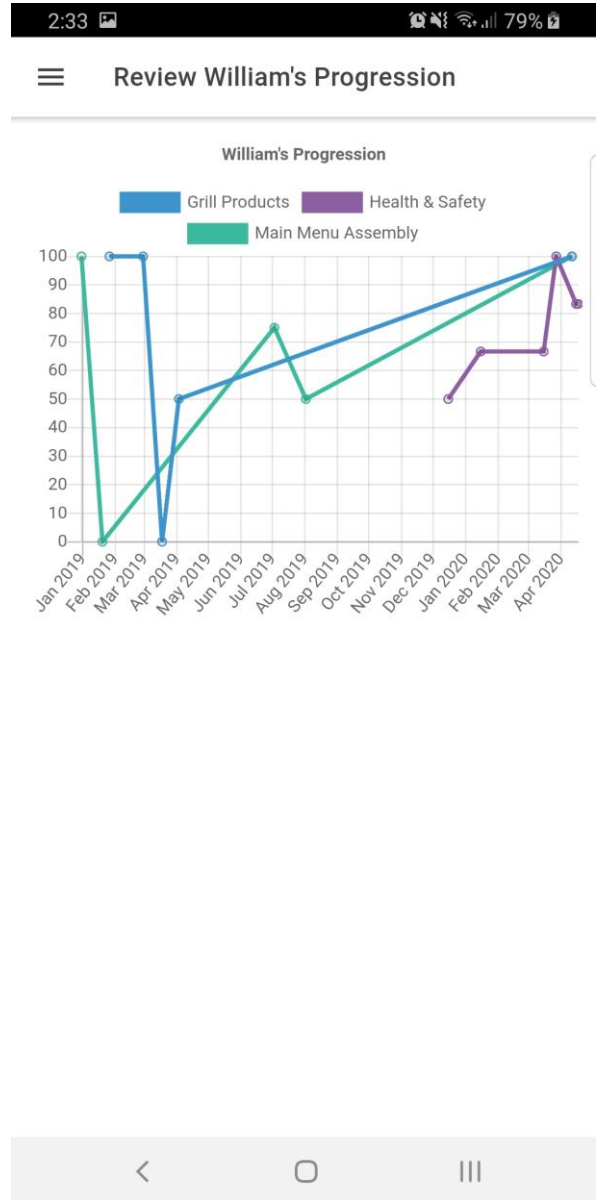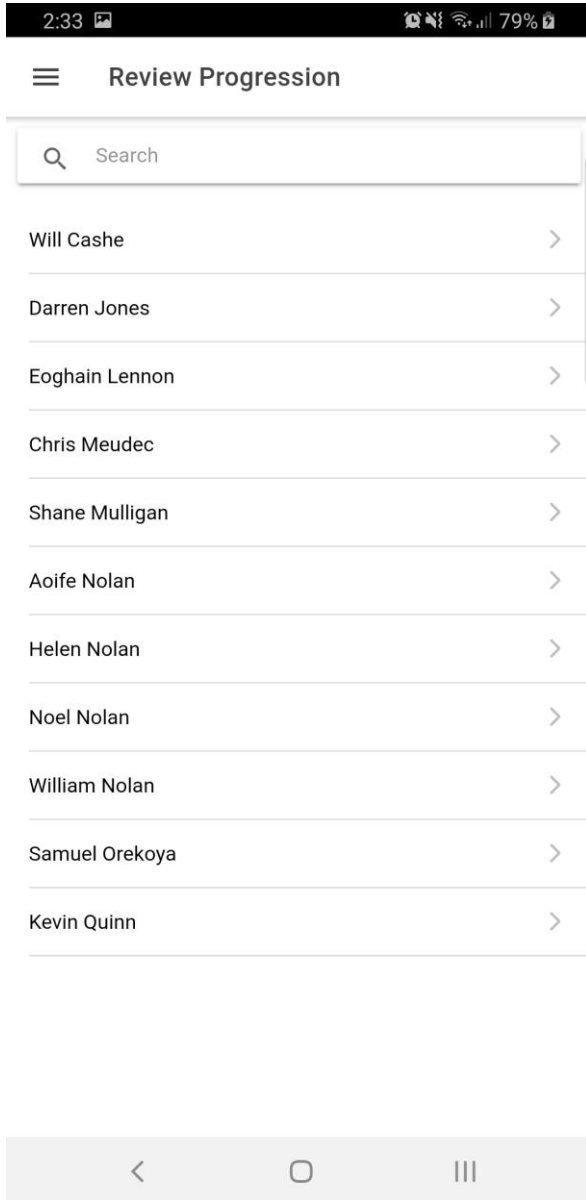
8. Appendix

8.1 DECLARATION

- I declare that all material in this submission e.g. thesis/essay/project/assignment is entirely my own work except where duly acknowledged.
- I have cited the sources of all quotations, paraphrases, summaries of information, tables, diagrams, or other material; including software and other electronic media in which intellectual property rights may reside.
- I have provided a complete bibliography of all works and sources used in the preparation of this submission.
- I understand that failure to comply with the Institute's regulations governing plagiarism constitutes a serious offense.

Student Name: William Nolan

Student Number: C00216986

Signature:

Date: 20/04/2020