SRS Document April 4, 2022

1. Introduction

1.1 Purpose

The application being developed is a driving evaluation system that caters to trained DriveQuest driving instructors who will be teaching people how to drive. Instructors will be able to keep track of multiple students and their corresponding driving tests, both in progress and completed. The results will then be emailed out to students after the training lesson with their instructor.

1.2 Intended Audience

The main users of this system are:

1) **Business owner:** Individual who owns and oversees the business.
2) **Driving instructors**: Individuals who give students the driving test and grade them on their performance.

The related users of this system are:

1) **Parents**: Individuals who will receive the email about the test result.
2) **Students**: Individuals who will receive the email about the test result.

This is a California driving simulation grading system for the company DriveQuest. It is only meant for use by trained driving instructors.

1.3 Business Rules

Workflow	Business Rules and Constraints
Freeway	The student must control their speed and show control before, during and after the lane change
Parking lot	The student must be able to signal correctly in the parking lot
	The student must be able to position their car correctly in the parking lot
	The student must be able to check their mirrors in the parking lot
	The student must be able to go the correct speed in the parking lot
Post-Test	Parents and Students can view feedback **ALMOST DONE**
	The test results will be created upon completion of the session **MOSTLY DONE**
	Test results will be stored in our database **ASYNC STORAGE - PASS**
	The user shall be able to send feedback via email right after the test to the student and parents **FAIL - STILL WORKING ON IT**

	The user shall be able to view and edit the test results before finalizing them
	The number of errors on the pre-drive check items 9-14 must be calculated.
	Instructor needs to be able to upload all feedback online.
	Parents/students must be able to receive test feedback.
	The instructor can generate test results and send it to the parent and user.
	The app has error handling for sending emails (ex. The email is wrong, there is no connection).
	With the email result, email will include a google form to rate the app. **WORKING ON IT**
Pre-drive Check	On the pre-drive checklist, "not correct" is assumed by default.
	The student must complete the pre-drive checklist items correctly.
	If the student misses a turn signal, then the instructor must provide detailed feedback to the student on which turn signal(s) is (are) missed on the pre-drive check.
	If the student misses an arm signal, then the instructor must provide detailed feedback to the student on which arm signal(s) is (are) missed on the pre-drive check.
Pre-Test	The user is able to play a predictable sound at will
	The user can demo sounds
Residential	Student must reverse at a safe, slow speed
	Student must back up in a straight line for 3 cars length
	Student must not hit the curb
	Students should maintain a safe distance to the front and sides of your vehicle
	Students need to maintain the vehicle in the center of the lane
	Students need to observe traffic for hazard that's ahead and behind and to the left or right
	The speed of the car need to maintain the appropriate speed limit without exceeding the speed limit
	Students should maintain a safe distance to the front and sides of your vehicle
	Students need to maintain the vehicle in the center of the lane
	Students need to observe traffic for hazard that's ahead and behind and to the left or right
	The speed of the car need to maintain the appropriate speed limit without exceeding the speed limit
	Students should follow the proper procedure to reverse such as checking over their right shoulder, maintaining control of the vehicle, back up parallel to the curb, etc.
	The instructor writes down a diagram of where the student failed at an intersection.

	The student will maintain the speed limit throughout the test.
	Student must pull up to the curb at a proper speed
Setup	The app needs a way to authenticate and keep track of instructors
	The app needs a way to know which student is taking the test
	If the instructor inputs the practice route # in the app, the app should display the appropriate evaluation screens to ensure smooth and efficient evaluation.
	User must enter their login id and password to enter the app
	User must enter their first name, last name, and password to signup and receive their user ID
	User must be an admin to edit a user
Test	All sections of the test shall save their data in a way that the test results can access
Throughout	The student needs to receive feedback while driving
	The map feedback must be accurate and detailed
	Live Saving all input from the instructor
	The ability to save different ratings via notes, true/pass or false/fail, or rating (1-5)
	The student should remain the correct lane when turning, without going too wide
	The student must not make unnecessary stops when completing right turns
	The student must yield correctly when making right turns
	The student must check their mirrors before turning
	The student must check their mirrors properly before changing lanes
	The device will provide haptic feedback whenever an item is checked on the checklist
	The test must feature a save button
	The test will be formatted in a checklist, similar to the paper test
	The user must be able to scale text and images as needed to make the app more accessible for individuals with visual impairments.
	The app must be able to run smoothly without distracting the instructor from evaluating the student.
	The student must signal within 100ft of a right turn
	The student must signal before changing lanes
	The system will keep track of the total infraction made by the student
	Intervention by examiner

	Striking any object is against the rule
	Disobeys traffic sign and signal
	Disobeys examiner
	Dangerous maneuver
	Driving 10 mph below/above speed limit
	Auxiliary equipment use
	Lane violation
	The student will have control of their steering when making a lane change
	The student will exhibit a safe speed when changing lanes
	The student must check traffic before changing lanes
	When the instructor marks down a student they must be able to specify which subsection it is referring to
	Instructor must be able to navigate to the Lane Change page at any point once the driving test has begun
	Instructor must be able to mark down error and edit error counter
	Instructors must be able to access the home page in order to switch sections or use the DQ button. Instructor should also have access to change settings **(P.H)**
	Instructor must differentiate left and right lane change errors
	Instructor must be have quick access to frequently used pages
	Instructors need to be able to react in an emergency
	determining the effectiveness of the app
	Website accessibility
	The database saves the results.
	The feedback gets saved via the app cache and does not get deleted if the app crashes or there is no internet connection.
	The app provides the ability to edit comments by adding them and viewing them in the feedback section.
	The user will be able to play a sound on demand

1.4 Systems Overview

Login System:

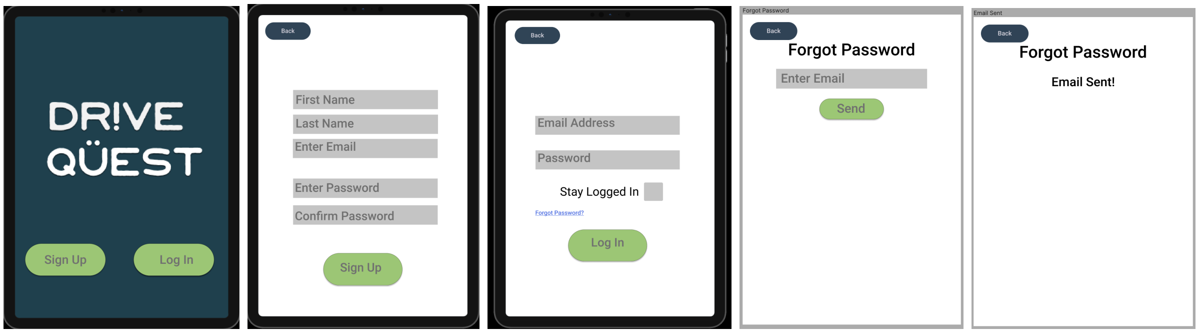
▸ Pages - Drive Quest (figma.com)

Description:
Allows for the creation of new instructor accounts and the logging in of already existing instructor accounts using an email and password. The system will also include a forgot password feature to allow instructors to reset their password in the event they forget it.

Features:

- Log in: The user can enter their email address and password to log in. They can also select the “stay logged in” option so that they do not have to enter this information each time they open the app.
- Sign up: The user can enter their name, last name, email address and password to sign up for the app by selecting the “sign up" button.
- Forgot password: The user can request for their password to be reset by providing their email. The system will then send the user a temporary password to allow the user to login and change their password.

Wireframe:



Driving Test:

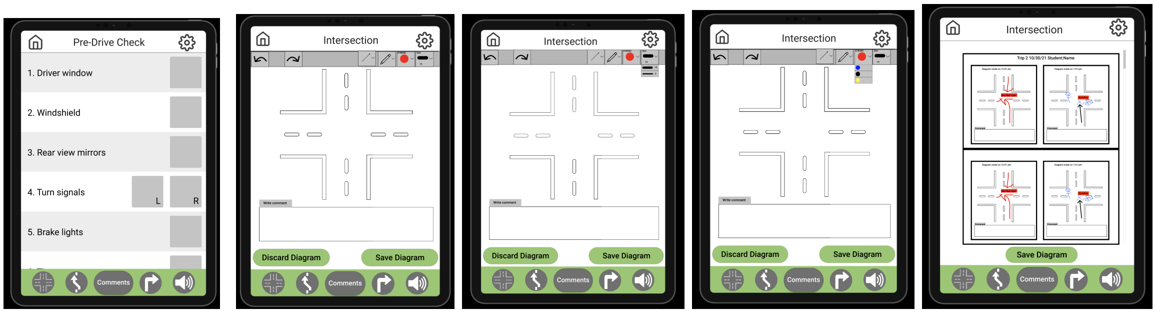
▸ Pages - Drive Quest (figma.com)

Description: The user will be able to score the student based on their performance on pre-drive check, parking lot, residential/business, freeway, turns, lane changes, and intersections.

Features:

- Pre-drive check
- Parking Lot
- Residential/Business
- Freeway
- Turns
- Lane Changes
- Intersections

Wireframe:



Pre-Drive Check

1. Driver window
2. Windshield
3. Rear view mirrors
4. Turn signals L R
5. Brake lights

Comments

Intersection

Write comment

Discard Diagram

Save Diagram

Comments

Intersection

Trip 2 10/30/21 Student Name

Save Diagram

Comments

[Post-Test](https://www.figma.com):

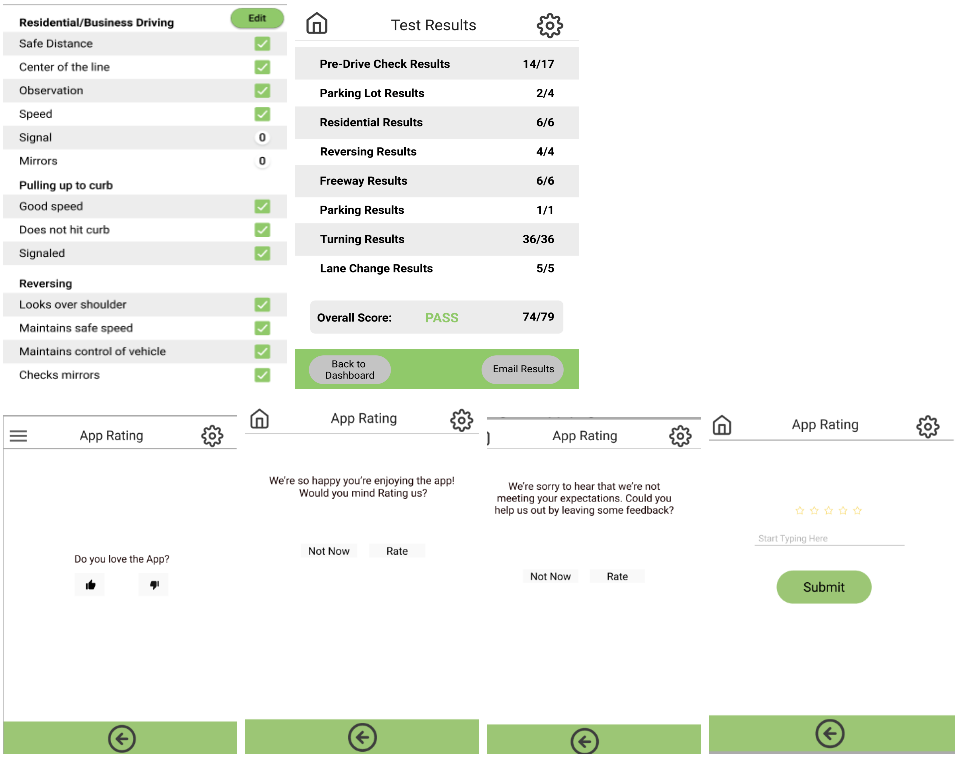
▸ [Pages - Drive Quest (figma.com)](https://www.figma.com)

Description: The user will be able to see their test results and receive a copy of them to their email. The results are organized into parking lot, residential/business, and freeways.

Features:

- Generated results **M.R.**
- Email Results **M.R.**
- Rate app

Wireframe:



Residential/Business Driving	Edit
Safe Distance	✓
Center of the line	✓
Observation	✓
Speed	✓
Signal	0
Mirrors	0
Pulling up to curb	
Good speed	✓
Does not hit curb	✓
Signaled	✓
Reversing	
Looks over shoulder	✓
Maintains safe speed	✓
Maintains control of vehicle	✓
Checks mirrors	✓

Test Results

Pre-Drive Check Results	14/17
Parking Lot Results	2/4
Residential Results	6/6
Reversing Results	4/4
Freeway Results	6/6
Parking Results	1/1
Turning Results	36/36
Lane Change Results	5/5

Overall Score: PASS 74/79

Back to Dashboard

Email Results

App Rating

Do you love the App?

App Rating

We're so happy you're enjoying the app! Would you mind Rating us?

Not Now Rate

App Rating

We're sorry to hear that we're not meeting your expectations. Could you help us out by leaving some feedback?

Not Now Rate

App Rating

☆ ☆ ☆ ☆ ☆

Start Typing Here

Submit

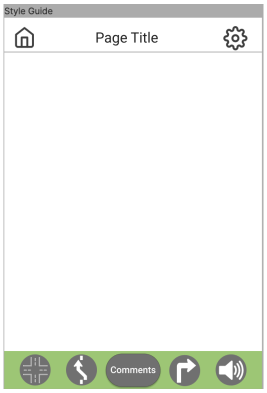
Throughout Test:

Description: During the driving test, a uniform page format will be used.

Features:

- Top Bar
  - Home Button
  - Page Title
  - Settings Button
- Bottom Bar
  - Intersections Button
  - Lane Change Button
  - Comments Button
  - Turns Button
  - Mute Button

Wireframe:



Settings & Preferences:

▸ Pages - Drive Quest (figma.com)

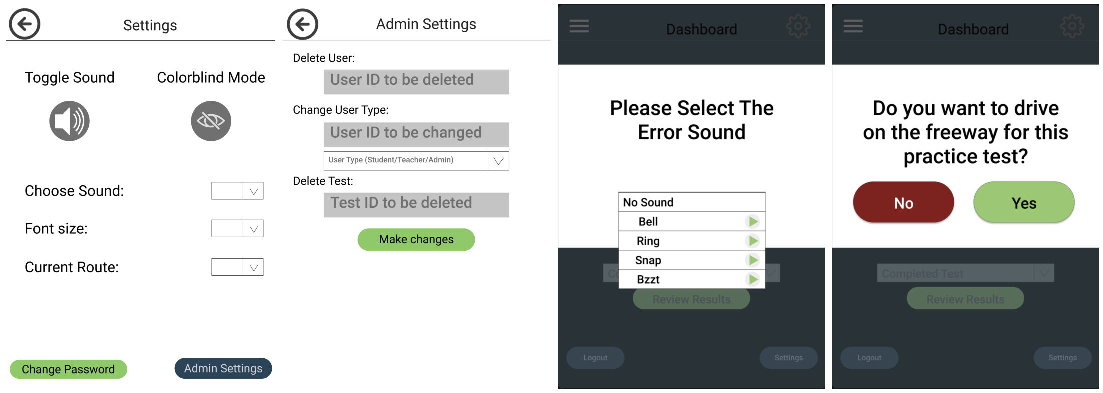
Description: The user will be able to access a settings menu to customize the app to their needs and perform administrative tasks. Among other features, the user will be able to toggle error sounds on and off, change the text size for readability, and manage users. The user will access the settings menu by clicking a gear icon in the top right of the screen. To access the admin settings menu, the user can click on the “admin settings" button in the bottom right of the regular settings menu. Finally, the user will be able to exit both of these menus by clicking the back button in the top left of the screen.

Features:

- The user will be able to select a sound to play on demand.
- The user will be able to demo a sound before selecting the sound.
- The user will be able to mute and unmute the error sound function.

- The user will be able to modify the font size throughout the app by accessing the settings menu and changing the font size there.
- The user will be able to toggle a colorblind mode which will change the colors of the app as needed to provide a better and more easily differentiated UI for colorblind users.
- The user will be able to modify their chosen practice route on demand and the subsequent screens will reflect the new practice route.
- ~~If the user is an admin, they will be able to delete users from the database.~~
- ~~If the user is an admin, they will be able to change a user's user type in the database.~~
- ~~If the user is an admin, they will be able to delete tests from the database.~~

Wireframe:



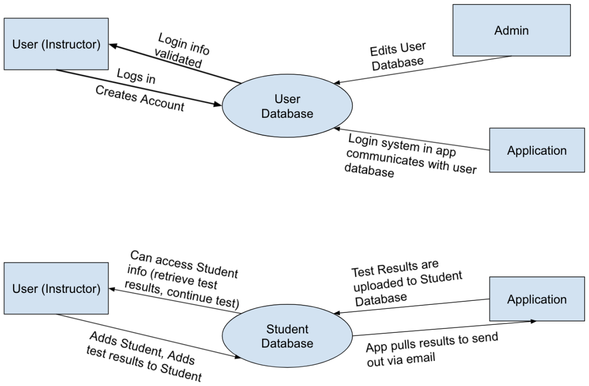
Database:

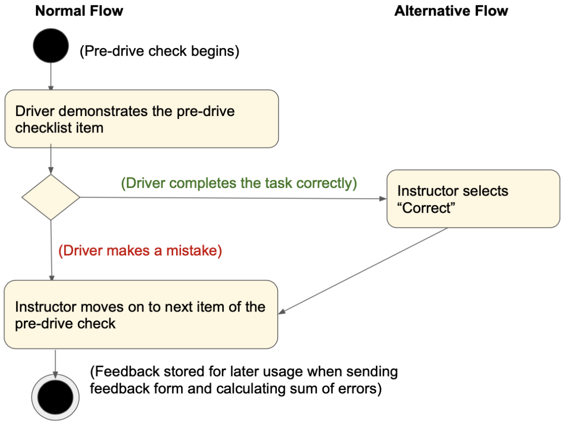
Description: The backend will hold all of the appropriate data and be able to access, delete, change, read and write. The database architecture will be based on a Firebase application hosting the REST API application to create the functionality backend, which will hold the data in a firebase database. Firebase is Google's easy and intuitive database platform that will allow us to store data in the cloud.

https://firebase.google.com/docs/reference/rest/database

Features:

- Stores the results for each portion of the test **M.R.**
- Generates a summary of the results at the end of the test **M.R.**
  - Previews the results after creation
- ~~Handles user accounts (Creation, deletion, storage, management)~~
- ~~Handles student profiles (Creation, deletion, storage, management)~~
- ~~Handles admin accounts (Creation, deletion, storage, management)~~


User (Instructor)
Login info validated
Logs in
Creates Account
User Database
Edits User Database
Admin
Application
Login system in app communicates with user database
User (Instructor)
Can access Student info (retrieve test results, continue test)
Adds Student, Adds test results to Student
Student Database
Test Results are uploaded to Student Database
Application
App pulls results to send out via email

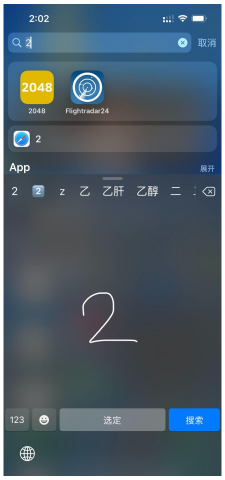

Normal Flow
Alternative Flow
(Pre-drive check begins)
Driver demonstrates the pre-drive checklist item
(Driver completes the task correctly)
Instructor selects "Correct"
(Driver makes a mistake)
Instructor moves on to next item of the pre-drive check
(Feedback stored for later usage when sending feedback form and calculating sum of errors)

1.5 References to Existing Systems

This application will have some similar functionality to that of Google Forms, since users will be able to tap on certain icons or checkboxes to grade students on their driving performance. Users will also be able to access a comment function to provide more in depth feedback.

This application also has a feature similar to Markup (seen in Notes and Photos apps) which will allow users to draw diagrams to describe actions taken by students during the test.

The Notes app, having the Markup drawing feature.



The Chinese handwriting keyboard, having the similar drawing feature as using your hand to draw on the screen.

2. Product Description

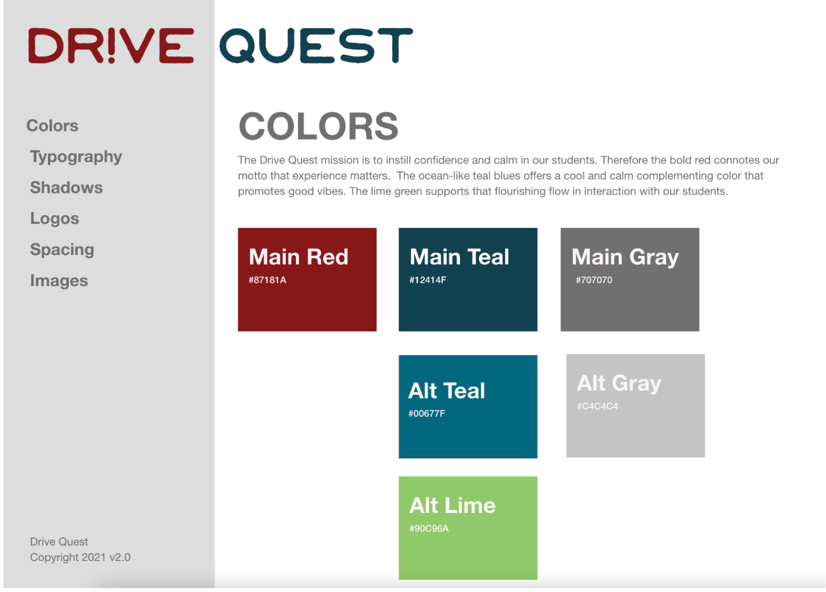
2.1 System Architecture

This system contains multiple components:

- Mobile application for interacting with the system
- Login system for driving instructors to access the system
- UI system for designing the visuals of the frontend
- User/Results Database system for storing user information and results to be sent out

2.2 Design Constraints

- The app shall function on both iOS and Android
- The app shall follow a consistent UI design style provided to us by the business owner.



DRIVE QUEST

Colors
Typography
Shadows
Logos
Spacing
Images

COLORS

The Drive Quest mission is to instill confidence and calm in our students. Therefore the bold red connotes our motto that experience matters. The ocean-like teal blues offers a cool and calm complementing color that promotes good vibes. The lime green supports that flourishing flow in interaction with our students.

Main Red #87181A
Main Teal #12414F
Main Gray #707070
Alt Teal #00677F
Alt Gray #C4C4C4
Alt Lime #90C96A

Drive Quest
Copyright 2021 v2.0

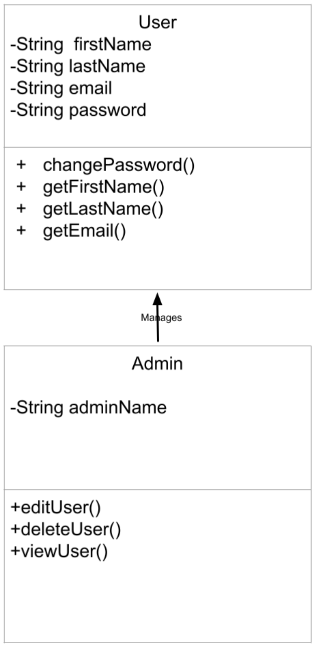
- **Interface.Clarity.Clutter:** The app shall follow a non-cluttered UI design to promote focus, performance, and usability.
- **Interface.Clarity.Contrast:** The app's colors and UI should have enough contrast to promote readability and usability.

3. System Requirements

Interface Class Diagram

3.1 Login Requirements

- **Login.login (online)** ~Trashed~: ~~The user shall be able to enter the application if they enter a valid email address and corresponding password.~~
- **Login.login (offline):** The user shall be able to enter the application if they have the valid login key. **(L.T.)**
- **Login.signup** ~Trashed~: ~~The user shall be able to enter their first and last name along with their email address and a password to sign up as a user.~~
- **Login.signup.validPassword** ~Trashed~: ~~The user's password must contain 8 characters and a number. No blanks, spaces or special characters.~~
- **Login.forgotPassword** ~Trashed~: ~~The user shall be able to hit the "Forgot Password" button to open the Forgot Password page, which will ask for the user's email address to send an email with a temporary password.~~
- **Login.stayLoggedIn**: The user shall be able to stay logged in after entering credentials and would not have to log in again even after application is closed/exited **M.R.**



User and Admin Classes

3.2 Driving Test Requirements

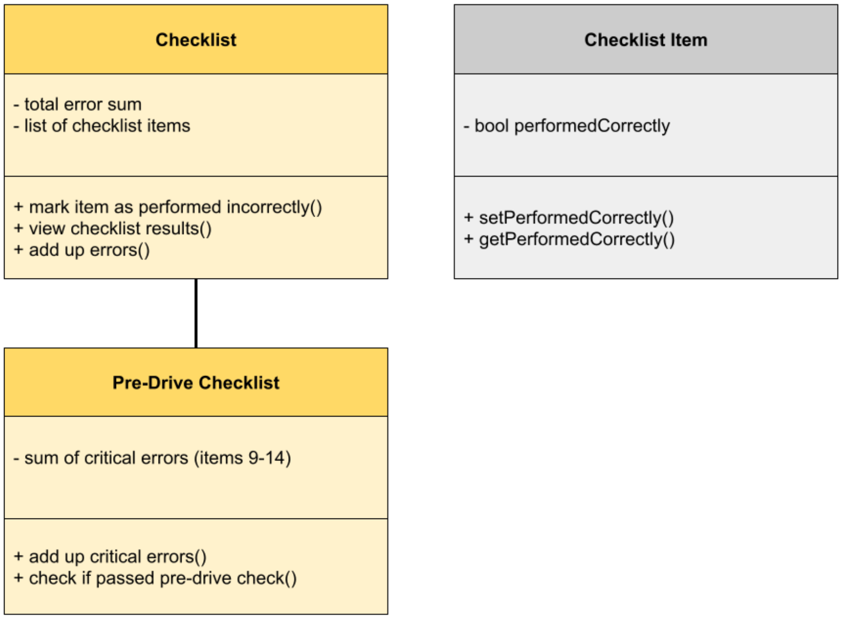
- **Interface.DrivingTest.TopBar:** During the test, the user will have access to a top bar containing the home button, the page title, and the settings button **(P.H)**
- **Interface.DrivingTest.BottomBar:** During the test, the user will have access to a bottom bar containing the intersection button, the lane change button, the comments button, the turns button, and the mute button **(P.H)**

  3.2.1 Pre-Drive

  - **User.FreewayDrivingPopUp.NavigateToPre-DriveCheck**: The user shall be able to access the pre-drive check page after selecting whether or not they would like to use the freeway for the practice test. **(KL)**
  - **User.HomePage.Navigation.Pre-DriveCheck**: The user shall be able to navigate to the pre-drive check page from the home page.
  - **Interface.RouteNum:** The system shall provide a pop-up that asks the instructor which practice route will be used for each practice test. The user will be able to select a practice route from a dropdown menu and the subsequent evaluation pages will change accordingly. **(KL)**
  - **Interface.RouteNum.Description:** The user shall be able to press and hold down on a route number option to gain a brief description about the practice route.
  - **Functional.Pre-Drive.Sounds.Select:** The user will be able to select the sound to play on demand. The user will be directed by a popup on the “start test” screen **(KL)**
  - **Functional.Pre-Drive.Sounds.Demo:** The user will be able to demo a sound before selecting the sound. The user will be directed by a popup on the “start test” screen **(Not done yet)**
  - **Business.Pre-DriveCheck.Default**: On the pre-drive checklist, “not correct” is assumed by default. **SC**
  - **User.Pre-DriveCheck.Checklist.Correct**: The user shall be able to place a checkmark next to each pre-drive checklist item to indicate that it has been completed correctly. **SC/KO**

    1. TC_User.Pre-driveCheck.Checklist
       a. Precondition – User begins test and access Pre-Drive Checklist page
       b. Input – Steps required:
          i. User Clicks on checkboxes next to each item
       c. Validation point/trigger/action – A green checkmark appears in the box
       d. Output – The checkbox next to each item can be checked or unchecked
       e. Postcondition – Checks are saved and the user can continue to access other pages

- **User.Pre-DriveCheck.Checklist.NotCorrect**: The user shall be able to leave a checkbox blank to indicate that the pre-drive check item has not been completed correctly. **SC/KO**
- **User.Pre-DriveCheck.TopBar**: The user shall be able to access the top bar from the pre-drive check page. **(P.H)**
- **User.Pre-DriveCheck.BottomBar**: The user shall be able to access the bottom navigation bar from the pre-drive check page.
- **Function.Button.OnTap:** Instructor should be able to tap the buttons to indicate whether something has been done **SC/KO**
    1. TC_Functional.Button.OnTap
        a. Precondition - User starts the test and access the residential and business driving page
        b. Input- Steps required:
            i. User clicks on the image icon on each of those
        c. Validation point/trigger/action - by clicking on the image this triggers the output
        d. Output - The image icon is able to turn red as the icon is clicked
        e. Postcondition - The clicking information is able to be accessed in the database and other pages
- **Function.Counter.OnTap:** Instructor should be able to click on a counter to indicate the number of mistakes the person has made
- **Function.Bar.OnTap:** Instructor should be able to tap on the bar/screen to create check marks **KO/SC**


Checklist
- total error sum
- list of checklist items
+ mark item as performed incorrectly()
+ view checklist results()
+ add up errors()
Checklist Item
- bool performedCorrectly
+ setPerformedCorrectly()
+ getPerformedCorrectly()
Pre-Drive Checklist
- sum of critical errors (items 9-14)
+ add up critical errors()
+ check if passed pre-drive check()

Pre-drive Check Class Diagram

3.2.2 Parking Lot K.I

- **Functional.ParkingLot.Signal**: The user shall be able to increment a counter button when the student does not signal correctly. K.I
- **Functional.ParkingLot.Mirrors**: The user shall be able to increment a counter button when the student does not check their mirrors. K.I
    1. TC_Functional.ParkingLot.Mirrors K.I
        a. Precondition – The user has access to the parking lot driving page, and the user observed an error the student made with mirrors
        b. Input – Steps required:
            i. The user taps on the icon
            ii. The '+' of the counter
        c. Validation point/trigger/action – The counter will be incremented by 1
        d. Output – The counter has been incremented
        e. Postcondition – The user can continue to mark errors on the test
- **Functional.ParkingLot.Positioning**: The user shall be able to increment a counter button when the student is not in the correct lane position. K.I
- **Functional.ParkingLot.Speed**: The user shall be able to increment a counter button when the student does not drive the correct speed. K.I

3.2.3 Residential SM

- **Functional.Residential.safeDistance**: The user shall be able to access the bottom to mark off when the student's vehicle is too close or too far from the car near them in residential driving.
- **Functional.Residential.centerOfLine**: The user shall be able to access the bottom to mark off when the student's vehicle is driving on the line or being off the position of the lane in residential driving.
- **Functional.Residential.speed**: The user shall be able to access the bottom to mark off when the speed of the car is too slow or too fast in residential driving.
- **Functional.Residential.observation**: The user shall be able to access the bottom to mark off when student does not observe the traffic for hazard in residential driving.

3.2.3.1 Business SM

- **Function.Residential.Business.safeDistance**: The user shall be able to access the bottom to mark off when the student's vehicle is too close or too far from the car near them in business driving.
- **Function.Residential.Business.goodPositioning**: The user shall be able to access the bottom to mark off when the student's vehicle is driving on the line or being off the position of the lane in business driving.
- **Function.Residential.Business.observation**: The user shall be able to access the bottom to mark off when the speed of the car is too slow or too fast in business driving.

- **Function.Residential.Business.safeSpeed:** The user shall be able to access the bottom to mark off when student does not observe the traffic for hazard in business driving.

3.2.3.2 Pulling up to the curb SM

- **Functional.Residential.parkingAtCurb:** Student must pull up to the curb at a proper speed
- **Function.Residential.hitCurb:** Student must not hit the curb

3.2.3.3 Reversing SM

- **Function.Residential.Reversing.rightShoulderCheck:** The instructor should be able to click the right shoulder icon to indicate an error if the student does not look over the right shoulder when reversing.
- **Function.Residential.Reversing.maintainControl:** The instructor should be able to click the maintain control icon to indicate an error if the student does not maintain control of the vehicle when reversing.
- **Function.Residential.Reversing.checkMirror:** The instructor should be able to click the mirror icon to indicate an error if the student does not check their mirrors when reversing.
- **Function.Residential.Reversing.safeSpeed:** The instructor should be able to click the speed icon to indicate an error if the student does not reverse at a safe speed.

3.2.4 Freeway

3.2.4.1 Entering ML

- ~~**Function.Freeway.scanning:** The user shall be able to mark whether the student scanned while entering the freeway.~~
- ~~**Function.Freeway.enterSpeed:** The user shall be able to mark the enter speed of the student on the freeway.~~
- ~~**Funciton.Freeway.trafficCheck** The user shall be able to increment a counter button when the student does not perform a traffic check on the freeway.~~
- ~~**Function.Freeway.positioning:** The user shall be able to increment a counter button when the student does not have proper lane positioning on the freeway.~~
- ~~**Function.Freeway.signal:** The user shall be able to increment a counter button when the student does not utilize their signal when changing lanes on the freeway.~~

3.2.4.2 Freeway Driving ML

- ~~**Function.Freeway.speed:** The user shall be able to mark the speed of the student on the freeway.~~
- ~~**Funciton.Freeway.trafficCheck** The user shall be able to increment a counter button when the student does not perform a traffic check on the freeway.~~

- ~~**Function.Freeway.positioning:** The user shall be able to increment a counter button when the student does not have proper lane positioning on the freeway.~~
- ~~**Function.Freeway.signal:** The user shall be able to increment a counter button when the student does not utilize their signal when changing lanes on the freeway.~~

3.2.4.3 Exiting ML

- ~~**Function.Freeway.exitSpeed:** The user shall be able to mark the exit speed of the student on the freeway.~~
- ~~**Funciton.Freeway.trafficCheck** The user shall be able to increment a counter button when the student does not perform a traffic check on the freeway.~~
- ~~**Function.Freeway.positioning:** The user shall be able to increment a counter button when the student does not have proper lane positioning while exiting the freeway.~~
- ~~**Function.Freeway.signal:** The user shall be able to increment a counter button when the student does not utilize their signal when exiting the freeway.~~
- ~~**Function.Freeway.postTest:** The user shall be able to move onto the post-test results screen.~~

3.2.5 Turns CA

- **Interface.Turns.Nagivation:** The user shall navigate between right and left turn pages by clicking on the associated text at the top of the page
  1. Test Case: TC_Interface.turns.navigation
     a. Precondition – The user has access to turns page
     b. Input – Steps required:
        i. The user navigates to turns page
        ii. The user clicks on the “left” button
        iii. The user clicks on the “right” button
     c. Validation point/trigger/action – User clicks the button
     d. Output – User is navigated to the associated turns page associated to which page they want to go on
     e. Postcondition – The user is able to use each of the turns page
- **Interface.Turns.ExitIntersection:** The user shall be able to exit the turns page by clicking on the intersection button
- **Interface.Turns.Back:** The system shall have a back button that returns the user to the residential driving section
- **Interface.Turns.ExitBack:** The user shall be able to exit the turns page by clicking on the back button
  1. TC_Interface.turns.ExitPage
     a. Precondition - Navigate to turns page
     b. Input- Steps required:
        i. Tap the back button on the bottom bar

        c. Validation point/trigger/action - User clicks back button
        d. Output - User is taken back to the residential page
        e. Postcondition - User is back on the residential page and can mark errors there or navigate to other pages using the bottom bar

- **Interface.BottomBar.RightTurn:** The user shall be able to reach the right turns page from any page during the driving test by clicking on the turns icon on the bottom bar
- ~~**Interface.Turns.ExitHome:** The user shall be able to exit the turns page by clicking on the home button~~
    1. ~~TC_Interface.Turns.ExitHome~~
        a. ~~Precondition – The user has access to the turn page~~
        b. ~~Input – Steps required:~~
            i. ~~The user taps on the home button~~
        c. ~~Validation point/trigger/action – The home button is tapped by the user~~
        d. ~~Output – "Home screen appears" or "Home screen does not appear"~~
        e. ~~Postcondition – The user is on the home page and can access other pages of the app~~ Didn't do

3.2.5.1 Left turn CA

- ○ **Functional.Turns.Left.Approach.TrafficCheck:** If the student fails to perform a traffic check when approaching a left turn, the instructor will mark the error in the system.
- ○ **Functional.Turns.Left.Approach.Signal:** If the student fails to signal when approaching a left turn, the instructor will mark the error in the system.
- ○ **Functional.Turns.Left.Approach.Braking:** If the student fails to brake when approaching a left turn, the instructor will mark the error in the system.
- ○ **Functional.Turns.Left.Approach.Yield:** If the student fails to yield when approaching a left turn, the instructor will mark the error in the system.
- ○ **Functional.Turns.Left.Approach.LaneUse:** If the student has incorrect lane use when approaching a left turn, the instructor will mark the error in the system.
- ○ **Functional.Turns.Left.Approach.UnnecessaryStop:** If the student performs an unnecessary stop when approaching a left turn, the instructor will mark the error in the system.
- ○ **Functional.Turns.Left.Stop.LimitLine:** If the student fails to stop at the limit line when stopping before a left turn, the instructor will mark the error in the system.

- **Functional.Turns.Left.Stop.TrafficCheck:** If the student fails to perform a traffic check when stopping before a left turn, the instructor will mark the error in the system.
- **Functional.Turns.Left.Stop.WheelsStraight:** If the student fails to have the wheels straight when stopping before a left turn, the instructor will mark the error in the system.
- **Functional.Turns.Left.Stop.FullStop:** If the student fails to come to a full stop when stopping before a left turn, the instructor will mark the error in the system.
- **Functional.Turns.Left.During.TrafficCheck:** If the student fails to perform a traffic check during a left turn, the instructor will mark the error in the system.
- **Functional.Turns.Left.During.SteeringControl:** If the student fails to show adequate steering control during a left turn, the instructor will mark the error in the system.
- **Functional.Turns.Left.During.TooWide:** If the student turns too wide during a left turn, the instructor will mark the error in the system.
- **Functional.Turns.Left.During.TooShort:** If the student turns too short during a left turn, the instructor will mark the error in the system.
- **Functional.Turns.Left.During.Yield:** If the student fails to yield during a left turn, the instructor will mark the error in the system.
- **Functional.Turns.Left.During.CorrectLane:** If the student fails to turn into the correct lane during a left turn, the instructor will mark the error in the system.
- **Functional.Turns.Left.During.Speed:** If the student uses an incorrect speed during a left turn, the instructor will mark the error in the system.
- **Functional.Turns.Left.During.Signal:** If the student makes signaling errors during a left turn, the instructor will mark the error in the system.

3.2.5.2 Right turn CA

- **Functional.Turns.Right.Approach.TrafficCheck:** If the student fails to perform a traffic check when approaching a right turn, the instructor will mark the error in the system.
- **Functional.Turns.Right.Approach.Signal:** If the student fails to signal when approaching a right turn, the instructor will mark the error in the system.
- **Functional.Turns.Right.Approach.Braking:** If the student fails to brake when approaching a right turn, the instructor will mark the error in the system.

- **Functional.Turns.Right.Approach.Yield:** If the student fails to yield when approaching a right turn, the instructor will mark the error in the system.
- **Functional.Turns.Right.Approach.LaneUse:** If the student has incorrect lane use when approaching a right turn, the instructor will mark the error in the system.
- **Functional.Turns.Right.Approach.UnnecessaryStop:** If the student performs an unnecessary stop when approaching a right turn, the instructor will mark the error in the system.
- **Functional.Turns.Right.Stop.LimitLine:** If the student fails to stop at the limit line when stopping before a right turn, the instructor will mark the error in the system.
- **Functional.Turns.Right.Stop.TrafficCheck:** If the student fails to perform a traffic check when stopping before a right turn, the instructor will mark the error in the system.
- **Functional.Turns.Right.Stop.WheelsStraight:** If the student fails to have the wheels straight when stopping before a right turn, the instructor will mark the error in the system.
- **Functional.Turns.Right.Stop.FullStop:** If the student fails to come to a full stop when stopping before a right turn, the instructor will mark the error in the system.
- **Functional.Turns.Right.During.TrafficCheck:** If the student fails to perform a traffic check during a right turn, the instructor will mark the error in the system.
- **Functional.Turns.Right.During.SteeringControl:** If the student fails to show adequate steering control during a right turn, the instructor will mark the error in the system.
- **Functional.Turns.Right.During.TooWide:** If the student turns too wide during a right turn, the instructor will mark the error in the system.
- **Functional.Turns.Right.During.TooShort:** If the student turns too short during a right turn, the instructor will mark the error in the system.
- **Functional.Turns.Right.During.Yield:** If the student fails to yield during a right turn, the instructor will mark the error in the system.
- **Functional.Turns.Right.During.CorrectLane:** If the student fails to turn into the correct lane during a right turn, the instructor will mark the error in the system.
- **Functional.Turns.Right.During.Speed:** If the student uses an incorrect speed during a right turn, the instructor will mark the error in the system.
- **Functional.Turns.Right.During.Signal:** If the student makes signaling errors during a right turn, the instructor will mark the error in the system.

3.2.6 Lane Changes CL

The following requirements apply to both the left and right Lane Change page:

- **Interface.LaneChange.Icon:** The user shall be able to access the lane change page at any time during the drive by selecting the lane change icon from the bottom bar
    1. TC_Interface.LaneChange.Icon
        a. Precondition – User has access to the pre-drive check page to view lane change button
        b. Input – Steps required:
            i. Click the lane change icon
        c. Validation point/trigger/action – Icon is clicked by user
        d. Output – The user is able to navigate to the lane change page at any time in the drive.
        e. Postcondition – After accessing the lane change page, the user is able to then mark any necessary errors in regard to changing lanes. They are able to navigate to intersections, turns, or the home page from their corresponding buttons.
- **Interface.LaneChange.Counter:** The user shall be able to reduce and increment the error counter
- **Interface.LaneChange.AccessHomeSettings:** The user shall be able to access home and settings page from the lane change page by selecting the appropriate icon
- **Interface.LaneChange.LeftRightOption:** The user shall be able to select from a left and right lane change page
- **Interface.LaneChange.AccessIntersectionTurns:** The user shall be able to access the intersection and turns page from the lane change page
- **Functional.LaneChange.Subsection:** The user shall be able to mark down an error for each subsection within the field
- **Functional.LaneChange.GradeSteering:** The instructor shall be able to mark down an error when the driver makes a lane change and does meet steering control guidelines
- **Functional.LaneChange.SafeSpeed:** The instructor shall be able to mark down an error when the driver makes a lane change and does not exhibit a safe speed
- **Functional.LaneChange.VerifyTrafficCheck:** The instructor shall be able to mark down an error when the driver makes a lane change and does not meet traffic check guidelines

3.2.7 Intersections ~Trashed~

- ~~**Function.Intersections.DiagramFeedback:** The user shall be able to make a mark on the diagram to give feedback on how the point was lost.~~

3.3 Post-Test Requirements

3.3.1 Test results summary

- **Functional.Post-Test.Feedback.Pre-CheckEdit:** The user shall be able to review the test results and make edits to the test score before submitting~Trashed~

3.3.2 Generated Results M.R. P.T.

- **Functional.Post-Test.Feedback.PassResults:** The system shall present an overall summary of the scores received on each main section of the test, then have the overall score and the “Pass” or “Fail” visible immediately at the bottom of the page.
- **Functional.Post-Test.Feedback.Pre-CheckOperationalErrors**: The system shall sum the number of operational pre-drive check items (numbered 9-16) not completed correctly.
- **Functional.Post-Test.Feedback.Pre-CheckOperationalDQResults**: The system shall include in the results page that the driver has been automatically disqualified if they miss 4 or more operational pre-drive check items (numbered 9-16).
- **Functional.Post-Test.Feedback.Pre-CheckMechanicalErrors**: The system shall sum the number of mechanical pre-drive check items (numbered 1-8 & 17-20) not completed correctly.
- **Functional.Post-Test.Feedback.Pre-CheckMechanicalDQResults**: The system shall include in the results page that the driver has been automatically disqualified if they miss any mechanical pre-drive check items (numbered 1-8 & 17-20).
- **Functional.Post-Test.Feedback.DriveTestErrors**: The system shall sum the number of errors in each category of the driving test (Parking Lot, Residential, Reversing, Freeway, Parking, Turning, Lane Change)
- **Functional.Post-Test.Feedback.DriveTestErrors.Expansion:** The user shall be able to select a section of the test in the generated results page that will expand that section and show the errors the driver received in that specific section.Not done yet, but ideal
- **Functional.Post-Test.Feedback.DriveTestDQResults**: The system shall include in the results page that the driver has been automatically disqualified if they made an automatic disqualification error at any point during the test.
- **Functional.Post-Test.Feedback.Upload**: The system shall include cloud functionality allowing the user to upload test results to the database from anywhere with cloud access.~Trashed~

3.3.3 Email Results MR

- **Functional.Post-Test.Feedback.Send:** The user shall send the finalized PDF test results via email in the app RR
- **Functional.Post-Test.Feedback.Send.EmailUI:** The system shall automatically fill in the email box with the student’s email that is saved on file and send out the PDF to the student first. The system shall also allow for multiple emails to be entered in for the PDF to be sent to.~Altered~~

3.3.4 Rating the App P.T. ~WORKING ON IT~

- **Functional.Post-Test.Feedback.RatingTheApp:** The user shall be able to access to a google form link to rate the app.

~Trashed~

- ~~**Functional.Rate.Student/Instructor:** Users of the app will be able to choose whether they want to rate or not.~~
    1. ~~TC_Functional.Rate.Student/Instructor~~
        a. ~~Precondition – The user hits the logout button and the user hits the logout button~~
        b. ~~Input –~~
            i. ~~User presses thumbs up~~
            ii. ~~User presses Thumbs down~~
        c. ~~User presses rate 2b. User presses do not rate~~
        d. ~~Validation point/trigger/action –~~
            i. ~~1a-> Screen where user can rate the app or not -> (2a or 2b)~~
            ii. ~~1b -> Goes to sorry screen where user can still leave feedback -> (2a or 2b)~~
            iii. ~~2a -> Goes to rate screen and user is able to rate the app then -> return to login page~~
            iv. ~~2b -> User clicks not now and is -> returned to the login page~~
        e. ~~Output – Goes to either positive rate prompt or negative rate prompt, then could go to rate screen or login page~~

        f. ~~Postcondition – "Returns to the login page of the app."~~

- **Functional.5StarRate.Student/Instructor:** Users of the app will be able to rate the app with a 5-star rating system.~Trashed~
- **Functional.RateComment.Student/Instructor:** Users will be able to give their comments about the app. ~Trashed~
- **Functional.Rate.Student/Instructor/SubmitUI:** The system shall allow for users to submit their rating to the Admin. ~Trashed~

3.3.5 Comment Written Feedback M.R.

- **Functional.Post-Test.Feedback.Comments:** The user shall be able to save written feedback comments.

3.4 Throughout Test Requirements

3.4.1 Sounds (Not done yet)

- **Functional.Throughout.Sounds:** The system will allow the user to play a sound on demand. When the user presses a button, a sound is played

1. TC_Interface.Throughout.Sounds
    a. Precondition – User has selected a sound in the pop-up menu when prompted, prior to starting the test.
    b. Input – Steps required:
        i. Log in
        ii. Select student and press “Start test”
        iii. Select sound and continue to begin the actual test
    c. Validation point/trigger/action – The user prompts the app to play a sound by pressing the sound icon in the bottom right corner of any evaluation screen.
    d. Output – The app should play a sound upon the user pressing the sound icon.
    e. Postcondition – The sound is played and can be played again at any time on demand.

3.4.2 Interface

- **Interface.Clarity.Scaling.Image:** The user will be able to tap on an image to focus it and the pan around the image using two fingers as well as zoom into the image using two fingers.

3.5 Settings & Preferences Requirements

3.5.1 Accessibility

- **Interface.Clarity.Scaling.Text:** The user will be able to modify the font size throughout the app by accessing the settings menu and changing the font size there.
- **Interface.Clarity.ColorbindMode:** The user will be able to toggle a colorblind mode which will change the colors of the app as needed to provide a better and more easily differentiated UI for colorblind users.

3.5.2 App Settings (Not done yet)

- **Functional.App-Settings.Sounds.Select:** The user will be able to select the sound to play on demand.
- **Functional.App-Settings.Sounds.Demo:** The user will be able to demo a sound before selecting the sound
- **Functional.App-Settings.RouteNum.Modify:** The user will be able to modify their chosen practice route on demand and the subsequent screens will reflect the new practice route.

3.5.3 Admin Settings

- ~~**Admin-Settings.admin.edituser**: Only an admin should be able to change who is a user.~~
- ~~**Admin-Settings.admin.deleteuser**: Only an admin will be able to delete the users.~~

3.6 Database Requirements

3.6.1 Database Functionality Requirements

- **Functional.Database.AppConnection:** The system shall connect the app to the database and the system shall be able to send and retrieve information to and from the database through the application.~Altered~

3.6.2 Database System Requirements

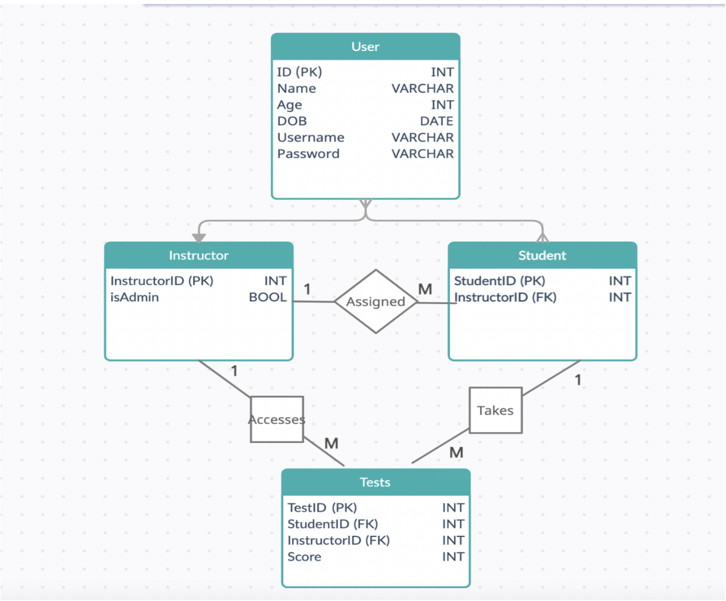
- ~~**System.Database.Instructor.CacheSaving:** Live Saving all input from the instructor (in a local database, to later be uploaded to a server database) PASS~~
- ~~**System.Database.Instructor.Ratings:** The ability to save different ratings via notes, true/pass or false/fail, or rating (1-5)~~
- **System.Database.Instructor.Comments:** The system shall store comments made by the instructor along with the results

3.6.3 Database User Requirements

- **User.Database**
  - **User.Database.User:** The database should store admin users with the proper privilege values~Trashed~
  - ~~**User.Database.GetInfo:** The database should be able to retrieve information of a specific user~Trashed~~~
  - ~~**User.Database.Upload:** The User shall be able to submit the test results and it will upload the information to the database~~
  - ~~**User.Database.Retrieve:** The user shall be able to visit a previous test and the information shall be retrieved from the database and be available to the user.~Trashed~~~
- ~~**Admin.Database**~~
  - ~~**User.Database.Admin:** The database should store admin users with the proper privilege values~Trashed~~~
  - ~~**User.Database.GetInfo:** The database should be able to retrieve information about the admin and their privileges~Trashed~~~

~~3.6.5 Other Database Requirements~~

  - ~~**Database.Structure:** The database shall use Firebase for storing all app data on the cloud~Trashed~~~

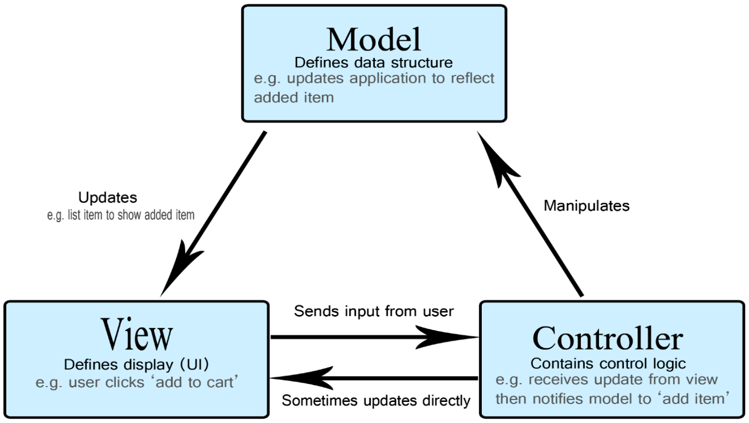


4. Design Patterns

~~**4.1 Account Validation - Layer Supertype:** Instructors and Admins will have overlapping and related functions. In order to create many Instructor objects without duplicating code, the Layer Supertype pattern is great for encapsulation. Common behaviors and attributes between Instructors and Admins like name, email, password can be moved into a common base case (also known as Layer Supertype).~~

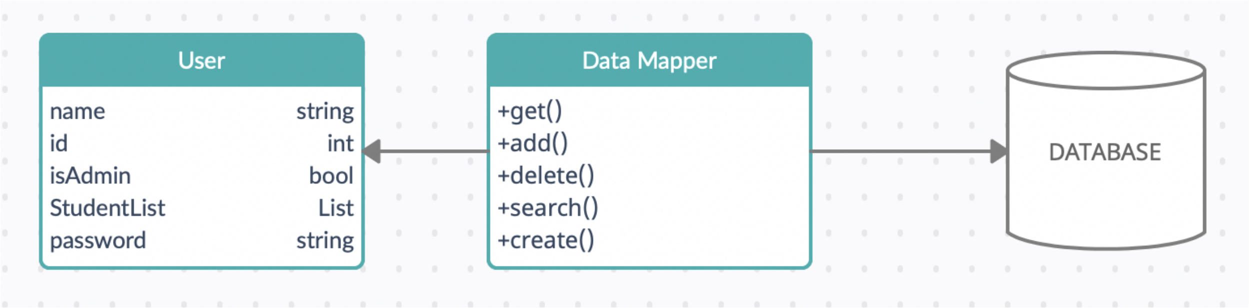
4.2 Test - possibilities: composite https://refactoring.guru/design-patterns/composite,

4.3 Generating Results - Application Controller
The application controller pattern is the pattern that permits the centralization of all view logic and promotes a unique process to define the flow of pages. It is responsible to handle inputs from the View components, manipulate data using the models from the Model component and then finally interact with the view components again to render the final output to the end user.


Model
Defines data structure
e.g. updates application to reflect added item
Updates
e.g. list item to show added item
Manipulates
View
Defines display (UI)
e.g. user clicks 'add to cart'
Sends input from user
Sometimes updates directly
Controller
Contains control logic
e.g. receives update from view then notifies model to 'add item'

~~4.4 Database - Data Mapper~~

~~The User class has the attributes of name, id, a check if they are an admin, a list of the students they oversee, and a password that is specific to them. The data mapper is an architectural design pattern named by Martin Fowler in 2003. The interface of the Data Mapper has functions such as get(), add(), delete(), and search() that interact with the relational databases which store data being objects of the User class. Stores and retrieves data from the database to store in the user class. Additionally, the Data Mapper also may serve as a means to create an account. Many of the same variables will be shared, such as name, id, and password. We also have a create() function that will add the user to the database with its respective variables.~~



4.5 Interface

4.5.1 Breadcrumbs

Use linked labels to provide secondary navigation that shows the path from the front to the current site page in the hierarchy. Applicable to the bottom bar when a button turns into a back arrow https://www.interaction-design.org/literature/topics/ui-design-patterns

4.5.2 Clear Primary Actions

Make buttons stand out with color so users know what to do (e.g., "Submit"). You may have to decide which actions take priority. This is used in the system during the driving test, when the error counters each have an associated icon that is tapped to increment an error. This is also used for the buttons that navigate through the system. https://www.interaction-design.org/literature/topics/ui-design-patterns

4.5.3 Progressive Disclosure

Show users only features relevant for the task at hand, one per screen. If you break input demands into sections, you'll reduce cognitive load (e.g., "Show More"). This is used on the test results page, where there is a score that summarizes each section. When the section header is tapped, more information is displayed about the results from that section. https://www.interaction-design.org/literature/topics/ui-design-patterns

4.5.4 Builder

Build individual parts of the interface (button, checkbox, text field, drawing feature, etc.) and then use these parts to build the overall interface. This would be used for the entire process of creating and then combining elements of the UI throughout the application. https://refactoring.guru/design-patterns/builder

4.6 Settings - Settings

The settings page will contain a variety of preferences that each user can modify to their own liking. These preferences will include whether or not a notification sound will play on a button press(and if so, what sound will play), whether or not a colorblind mode will be used, what the general font size is throughout the application, and which practice route the app is currently using to base subsequent test evaluation screens on. Initial values that are most likely to pose little risk are sound not being muted, colorblind mode being off, a bell sound being selected as the notification sound, 11 pt being chosen as the font size, and Route A being chosen as the practice route. Following this design pattern will help users dictate how the app should behave and increase accessibility. (Link to interface class diagrams)

4.7 Edit Results (Checkboxes) - Morphing Controls

Morphing Controls is a design pattern centered around making certain controls available depending on what mode the page or user is in. This applies to editing results because users are presented with edit functionality when they tap the edit button on the "Review Test Results" page. Checkboxes on this page morph into green checkmarks or red X's to signify passing or failing that particular part of the test respectively. (Link to interface class diagrams)