Mandatory Assignment - UX design

The deadline for this assignment is Friday, February 22, 08:59.

For this **mandatory exercise** you should work on **master branch only**.

Preparation

1. Create a new repository on GitHub called **mandatory-uxdesign2**.
2. Follow the instructions that GitHub gives you; Create a local repository and add a remote or clone the newly created repository.

Submission

When you submit the exercise in PingPong, before the deadline, you will enter a link to your repository, such as:

https://github.com/mygithubusername/mandatory-uxdesign2

The teacher will look in the **master branch**. If any commits are done to the branch after the deadline, the teacher will look at the last commit before the deadline.

Included in this assignment is an oral presentation on **Friday, February 22**. A schedule will be available in PingPong.

You will get one of the grades **G**, **VG** or **IG**.

Instructions

In this assignment you will create a quiz application using the **Open Trivia DB**, a database of trivia question with an accompanying API that can be used to generate random quiz questions.

The URL to the API with documentation is
https://opentdb.com/api_config.php

Requirements

The requirements for the application are different for the grade VG and G. Requirements marked with (VG) are required to achieve the grade VG but not required for the grade G. The oral presentation will also affect the final grade.

A prototype is available on
https://projects.invisionapp.com/share/R7QA2LBTVMB#/screens
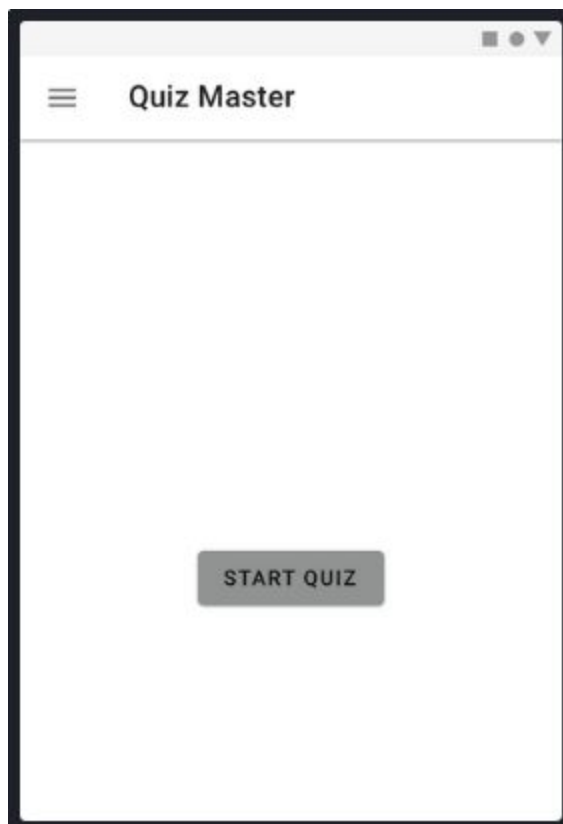
The following should be implemented

- Main screen
- Quiz screen
- Modal dialog when the quiz is finished
- Drawer menu (VG)
- Stats screen (VG)
- About screen (VG)

Main screen

The main screen should contain a button that is used to start the quiz. When the button is pressed a set of trivia questions are fetched from the API and the user is shown the Quiz screen.

Quiz screen

The quiz screen should display the questions with the possible answers as a set of radio buttons. 10 questions should be displayed.
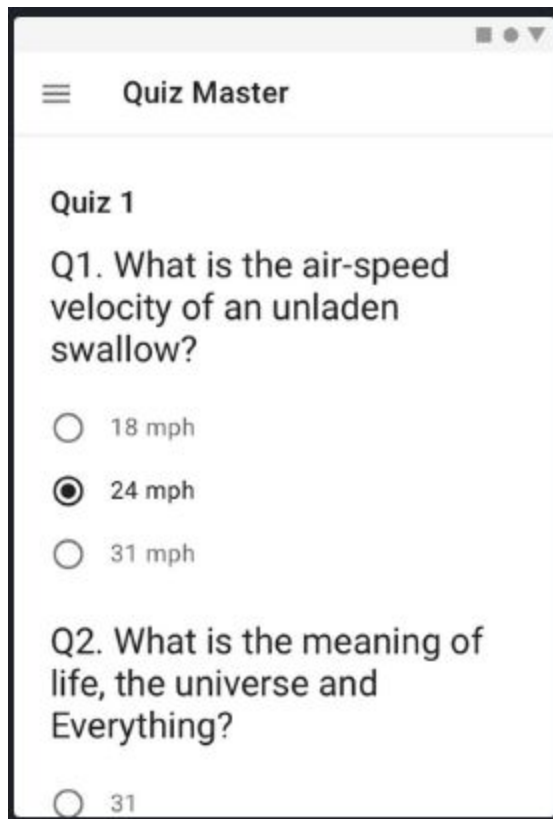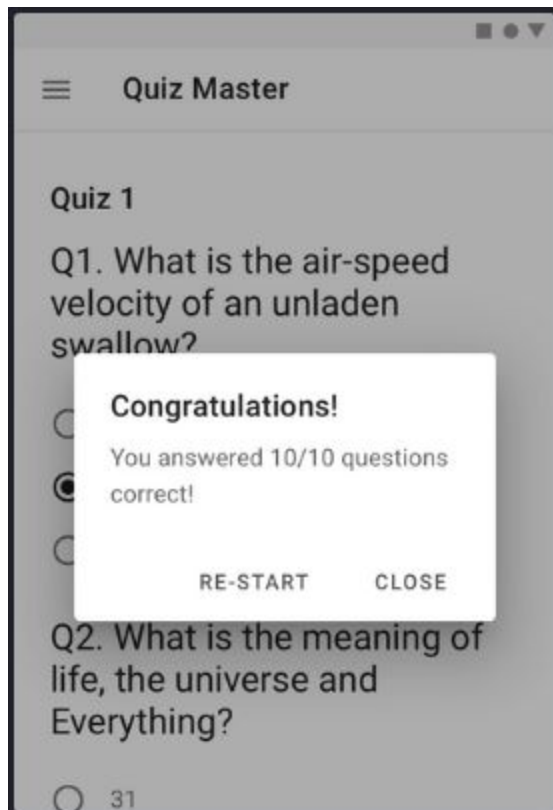

A button to submit the answers should also be included in the quiz screen.

Modal dialog

A modal dialog is a dialog that blocks all other interactions on a page and forces the user to interact with the dialog before continuing using the application

The dialog should display the score and ask if the user wants to try again or go back to the main screen.

When the dialog is visible, it should not be possible to interact with the background.

Drawer menu

A drawer menu is a navigation menu that slides in from either the left or right when pressing a button. The button is often a "hamburger" button in one of the top corners of the page.

The student should implement a drawer menu or use one included in a UI component library.

Quiz Master
Become the master of quizzes

○   Game screen

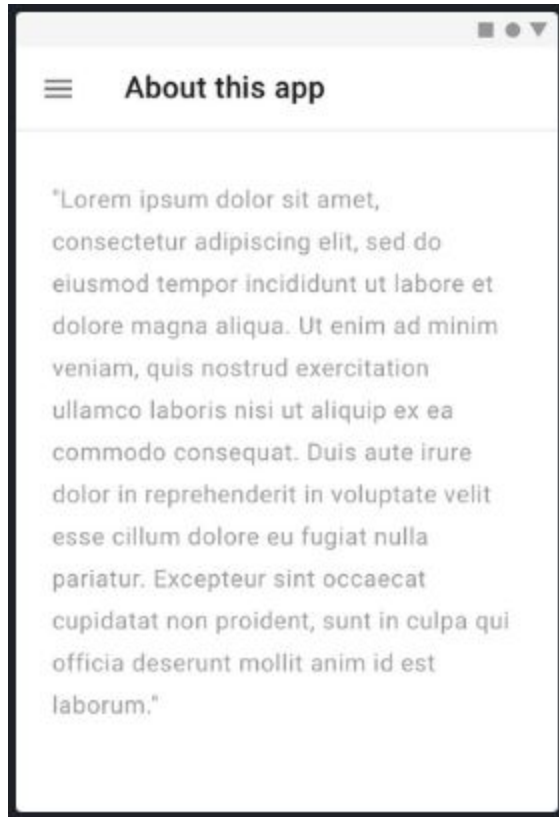○   Stats

○   About this app
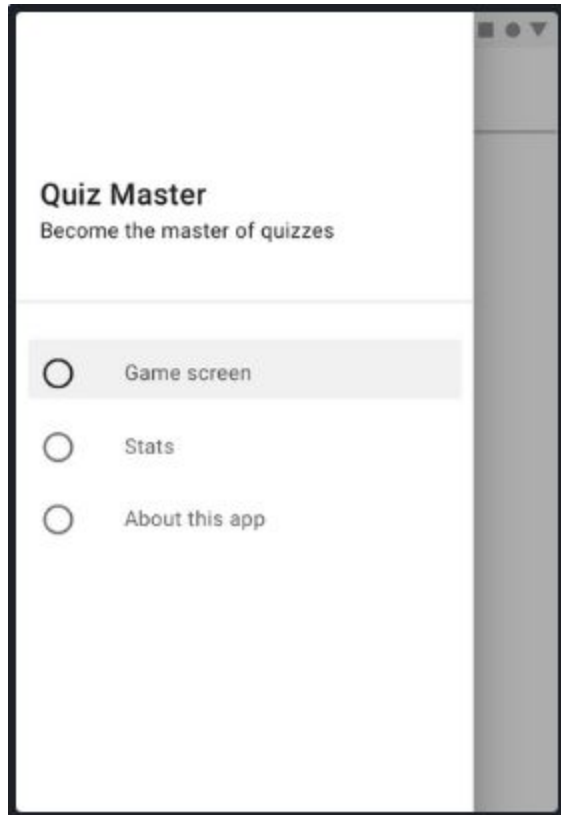
Stats screen

The application should record statistics and display them on the stats screen.
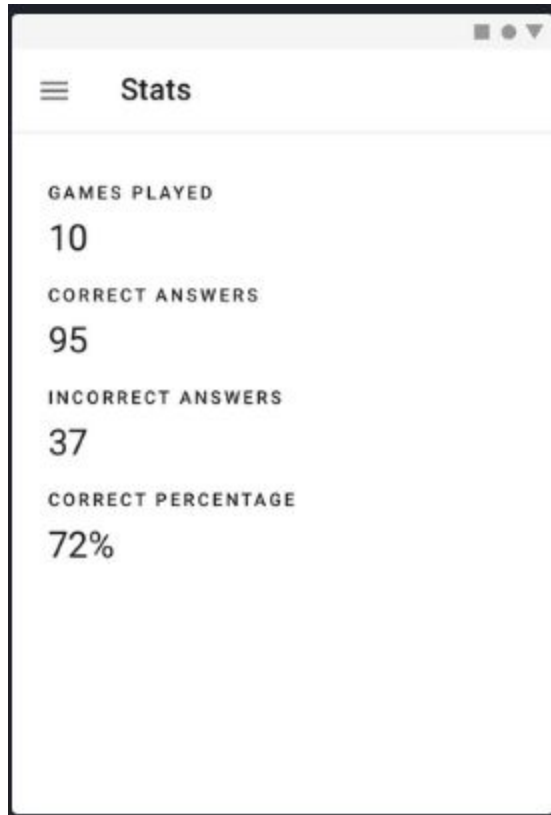
It is not required to save the stats between sessions. Storing the values in memory is sufficient.

About screen

The about screen shows some information about the application. Lorem ipsum is sufficient.

Design requirements

In the prototype, Material Design components have been used. However, Material Design is **not** required for this assignment.

The student is free to

- use a UI library (e.g. Bootstrap or Material Components for Web)
- implement components based on a design system, design specification or mimic some other application
- create an original design and implement the application based on that

In the case where the student chooses to create and implement a design from scratch it is important to describe the thought process that led to the design in the oral presentation.

If a library was used or the design was based on something else, the student should describe the choices made and why they were made during the oral presentation.

Responsive the design is **not** required for this assignment. The menu for the hamburger button does not have to automatically expand on larger screen sizes.

Accessibility requirements

To achieve the grade VG, the student is required to make an effort to make the application accessible and describe the steps that were taken to make the application accessible and how the accessibility was tested during the oral presentation.

For the grade VG the following is required

- The text is legible on small screens and in bad conditions
- The contrast between foreground (text) and the background should not be too low

  Contrast can be tested on https://dequeuniversity.com/rules/axe/2.2/color-contrast
- It should be possible to use the application with a screen reader
- Both the drawer menu and the modal dialog should be made accessible

Tips

- Try to use a UI library in case you feel implementing everything from scratch is too time consuming
- The best way to test accessibility for screen readers is actually using a screen reader
- Try to keep the code as simple as possible
- Try implementing the complex components (dialog and drawer menu) separately before you integrate them in the application