Quizzy

Logic App Development Coursework

Name: ***Akshay Kumar Venkatesha Narla***
Matriculation Number: ***3571510***

Introduction: The developed application, known as “Quizzy”, shall provide a quiz based on propositional logic. The objective of the app is to run as a standalone app on a mobile device with the functional and non-functional requirements as specified below. The quiz app, which runs both offline and online, shall be of the multiple-choice format.

Requirements: This section shall provide the required app functionalities. The detailed requirements for each component of the app is provided below.

Functional Requirements:

Front-end:

- Flutter framework shall be used to develop the user interface(UI).
- The UI shall contain a “login” screen which allows users to log onto the quizzy app home screen.
- The home screen shall provide users with the option to start a new quiz session, look at the user statistics and app user guide.
- Upon “New quiz session” selection, a new screen with quiz shall be displayed. An individual session would provide 10 quiz questions/tasks.
- The app shall run on 2 modes: timed and non-timed quiz. The mode selection shall be possible after the new quiz session selection.
- The quiz screen shall provide the questions in a multiple choice format and shall be automatically generated from the back-end.
- A countdown timer shall start when a new question is displayed, if timed quiz mode is selected. If the question is not answered before timed out, a new question shall be generated.
- Once an answer is provided, the correct answer shall be highlighted in green while the incorrect options in red.
- Upon “User statistics” selection, quiz statistics of the user from previous sessions shall be displayed. This data shall be persistent.
- The app user guide shall provide users instructions for app usage.
- The app shall provide a different layout for portrait and landscape mode.
- To run the quiz offline, a set of 10 defualt quiz questions shall be used and built-in to the UI.

Back-end:

- The server shall provide the quiz questions or tasks to the UI using a Golang API server.
- JSON-based REST API shall be used for communication between UI and the server.
- The server shall also collect information from UI and provide it to the admin via a HTML interface.

- The server shall provide an interface to the admin to remove or add tasks/quiz questions without stopping the server.
- A local database based on sqflite package (SQLite) shall be used for storing and persisting user data.

User management:

- A new server backend based on REST API shall be implemented for login and registering users using Golang for synchronizing across devices.
- The admin shall be able to delete users and reset passwords.

Automatic Task/Quiz generation:

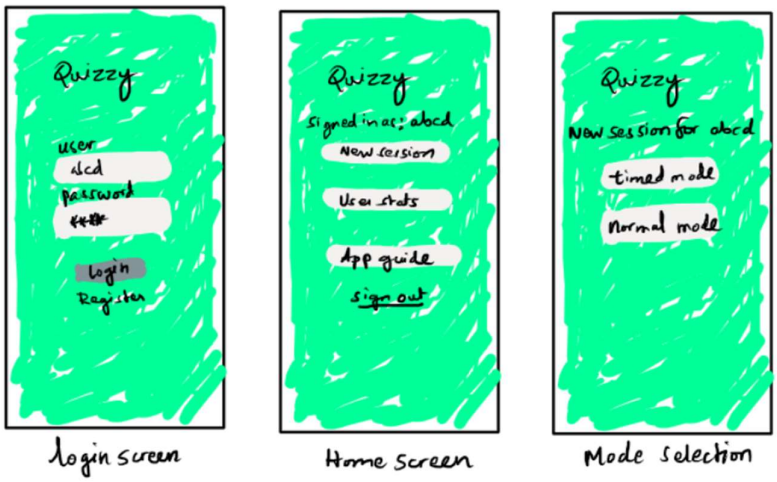
- The propositional logic quiz questions shall be generated automatically using Golang.
- The multiple choices shall also be automatically generated along with the questions.
- The generated tasks shall be fetched from the server to the UI using JSON-based APIs.

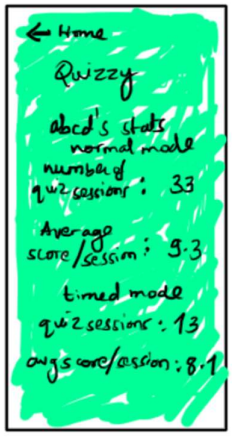
Non-functional Requirements:

- The developed app shall run both online as well as offline.
- The developed app shall be tested on android emulator.
- The validity of the generated quiz shall be ensured in the server-side.
- The app shall depend on “flutter_riverpod” for state management and “go_router” for navigation between screens.

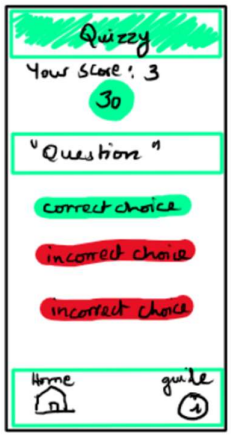
Sample App Layout Images: The following sample images are subject to minimal improvement as the app is developed, but the overall layout structures are expected to be similar.

Portrait mode:

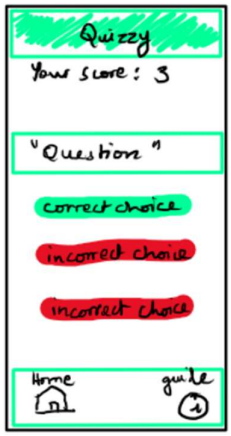




stats screen

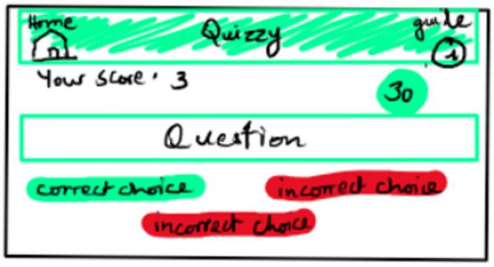


Quiz screen
"Timed mode"



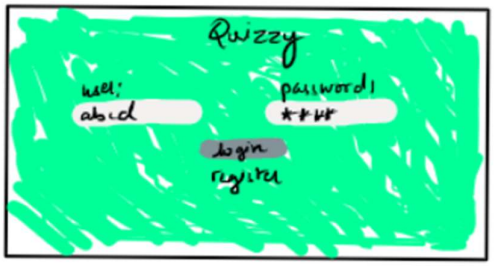
Quiz screen
"Normal mode"

Landscape mode:

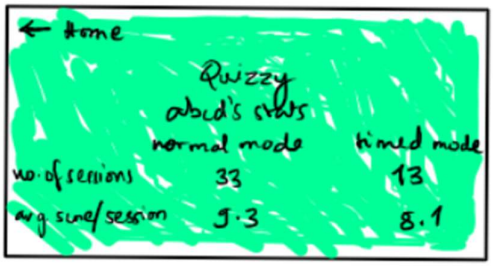


Quiz screen
"Timed mode"

Home screen



login screen



stats screen