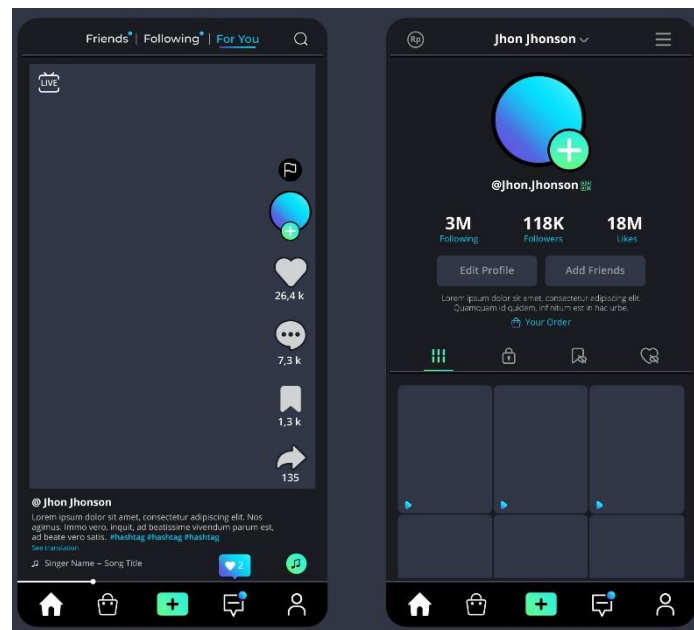


MOBILE PROGRAMMING

Tutorial 06

Activity 01: TikTok Clone

For this exercise, students will embark on creating a simplified version of a TikTok app, focusing primarily on implementing navigation functionalities using React Navigation. The exercise aims to familiarize students with the process of structuring and navigating between different screens within a mobile application.



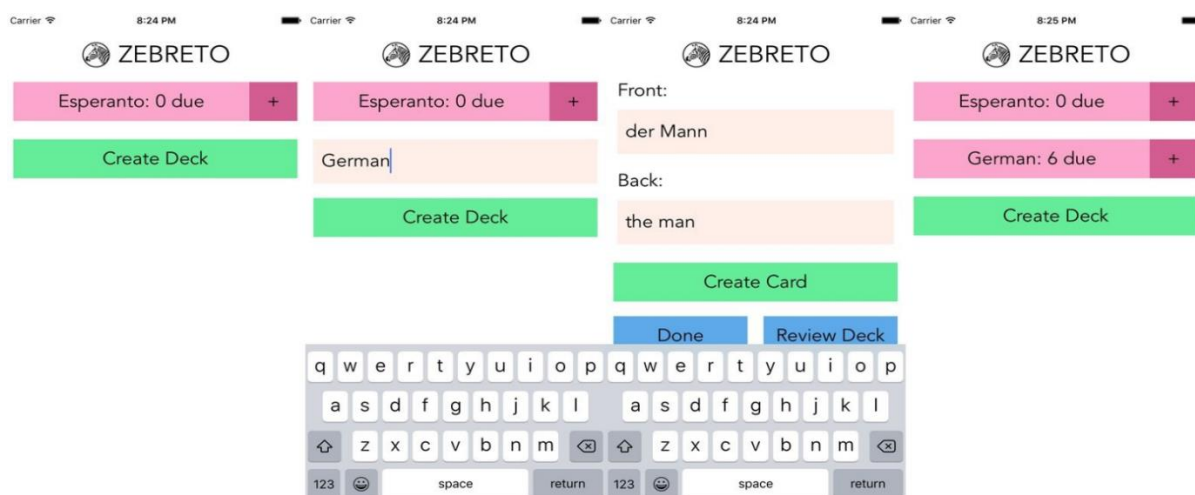
Initially, students will design and create the foundational components of the app, including the **Home**, **Inbox**, **Inbox Details**, **Profile**, and **Profile Details** screens. These screens will serve as the main navigation points within the application, allowing users to navigate seamlessly through various sections.

To facilitate the exercise, students may use demo data provided in pre-loaded files within a designated '**data**' folder. This data will simulate user interactions and content within the application, allowing students to focus solely on implementing navigation functionalities without the complexity of integrating real-time data sources.

During the development, students should design the screens creatively and efficiently, leveraging reusable components and loops for generating duplicated components where necessary. Additionally, it's imperative to guarantee seamless adaptation to both portrait and landscape orientation, ensuring a responsive user interface that caters to diverse device orientations.

Activity 02: Zebreto

Zebreto is a flashcard application based on the Spaced Repetition System (SRS), a learning strategy for effective memorization. With SRS, the goal is to review information just before you would otherwise forget it. If you do any foreign language study, you may be familiar with SRS systems; they allow you to memorize large amounts of data more quickly, focusing on long-term retention. A common approach is to start with a small interval between reviews, such as an hour, and to slowly scale up as you get cards correct: first an hour, then a day, then three days, then a week. Intervals can gradually increase to as much as a year, or five years. Tracking these intervals is impractical with pencil-and-paper flashcards, so we'll build an app instead.



The app begins with setting up the project structure and installing necessary dependencies, including React Navigation. Students will then create the basic layout and components for each screen, focusing on designing intuitive user interfaces that facilitate efficient navigation and interaction. Utilizing React Navigation's stack navigator, students will establish navigation routes between the screens, ensuring smooth transitions and consistent user experience throughout the app.

In the **Deck** and **Deck Creation** screens, students will implement functionalities for managing flashcard decks, including viewing existing decks and creating new ones. The **Card Deck** and **Card Creation** screens will enable users to manage individual flashcards within a deck, allowing for the creation, editing, and deletion of cards. The **Review** screen will simulate the spaced repetition algorithm, presenting flashcards to users based on their scheduled review intervals. Finally, the **Result** screen will provide feedback on users' performance during the review session, displaying metrics such as the number of correct and incorrect responses.







Throughout the exercise, students will have the option to use demo data loaded from the **'data'** folder, streamlining the development process, and focusing on implementing navigation features effectively.

During the development, students should design the screens creatively and efficiently, leveraging reusable components and loops for generating duplicated components where necessary. Additionally, it's imperative to guarantee seamless adaptation to both portrait and landscape orientation, ensuring a responsive user interface that caters to diverse device orientations.

Hint: *moment* is used for date objects, and *md5* is used for calculating card and deck IDs. In addition, *lodash* gives us some nice utility functions, and we use it for shuffling card reviews.

Submission

-  Create a new project folder named 61fit3mpr_tutorial_06_<your student ID>, this folder should have all files and sub-folders (except for folder 'node_modules') related to your project, for example: assets, components, etc.
-  Create new folders named: **DESIGN** (.png design files), **GUIDE** (How to install and use), **RUN** (Sample run images), **Presentation slides** (How did you built its components, screens, etc. steps by steps).
-  Compress all those folders into a .zip file named: 61fit3mpr_tutorial_06_<your student ID>.zip
-  Submit the .zip file on LMS using your own account.

Note: If you use some graphics program such as Adobe Photoshop or Illustrator to create your images, you don't have to include the design files (.psd, .ai, ...) into your folder.