

README

Project Name: RateMate

Synopsis

RateMate is an app that allows UT students to create their own profile and write reviews/rate their fellow peers on a scale of 1 star to 5 stars. If the student to be reviewed is not registered, registered students can create a placeholder account to add reviews to until said student registers and decides to merge their accounts.

Dependencies

Apple Swift version 6.0

Swift Language Version: Swift 5

XCode Version 16.0

Package Dependencies: Firebase (11.5.0)

Note: If there is an alert that says Missing Packages for Firebase, this is how I resolved that issue:

File > Packages > Reset Package Caches

File > Packages > Update to Latest Package Versions

Running the App

The simulator that should be used when running this app is the iPhone 16. Other simulators with different iPhone dimensions will mess with the constraints of the labels/text fields/etc. The app was intended to function in portrait mode and should not be rotated landscape or upside down. Some dummy accounts have been created (using a few SpongeBob characters) and have reviews and ratings for them already for login purposes. I have also created my own personal account for fun which can be found by typing "Tracey Le" in the search bar.

Emails

Sandy Cheeks: sandy@my.utexas.edu

Eugene Krabs: mrkrabs@my.utexas.edu

SpongeBob Squarepants: spongebob@my.utexas.edu

Squidward Tentacles: squidward@my.utexas.edu

Patrick Star: patrick@my.utexas.edu

Gary the Snail: gary@my.utexas.edu

Sheldon Plankton: plankton@my.utexas.edu

Passwords

All passwords for these accounts are the same: 123456

(Note: Originally I used the default password requirements which was 6 characters when creating these accounts but have since added the restriction that the password must be at least 8 characters.)

Additional Notes (Optional)

Adding Reviews

When adding reviews, there are two ways: directly and indirectly.

Directly adding reviews means using the search tab to find a specific user, navigating to their page, and pressing the 'leave review' button under their profile image. When segued to the review VC, the name at the top will already be filled in as a label, and users will be able to leave the reviews for that specific user until they navigate to another user's page.

Indirectly adding reviews is clicking the 'Add Review' tab at the bottom of the screen. Then, the user will be prompted to enter in the first and last name of the user they wish to review. Upon completion, the new account will be created with the user's review, and others can then *directly* add reviews onto the placeholder user's account by navigating to their page.

Deleting Reviews

Only the original poster can delete reviews from other people's pages.

Merging Accounts

As a basic security measure, the name on the merging account must match the name on the user's account. Merging account moves all the reviews from the merging account to the user's account and afterwards, deletes the merger account

Profile Images

Profile pictures were stored in Core Data as Firebase Storage (which can store files and images) required payment to use. That means profile images will only be saved to the user's device and won't appear on other devices. However, all other elements have been saved to Firebase Firestore, so they will appear across launches.

Search Function

Sometimes the profile images in each cell will not appear circular at first launch. However, interacting with the search bar or clicking on a profile and going back will correct this.

The text entered in the search bar and the search results are unfortunately case sensitive. As Apple automatically does title case when entering text and the profiles provided are title cased (aside from SpongeBob), this shouldn't be a major problem.

To dismiss the keyboard, press the "Search" key

Themes

Even though the user can change the theme color across the app, the signup and login page theme will remain orange. This is because this is an app aimed at UT students, so I wanted to keep a semblance of that somewhere. The text fields will change between white and black based on the user's phone setting for dark mode (though the black text boxes look a bit strange against the orange background). This is because I wanted to preserve the placeholder text.

Required Feature Checklist

☒ "Settings" screen. The two behaviors implemented are:

(fill in) dark mode & color theme

☒ Non-default fonts and colors used

Two major elements used:

☒ Login/register path with Firebase

☐ Core Data

☒ User Profile path using camera and photo library

☐ Multithreading

☐ SwiftUI

Minor Elements used:

☒ Two additional view types such as sliders, segmented controllers, etc. The two implemented are: (fill in) segmented controllers, text views, switches, search fields, bars and bar buttons

At least one of the following:

☒ Table View (*search results*)

☒ Collection View (*review display*)

☒ Tab VC

☐ Page VC

At least one of the following:

☒ Alerts

☐ Popovers

☐ Stack Views

☐ Scroll Views

☐ Haptics

☒ User Defaults

At least one of the following:

☐ Local notifications

☐ Core Graphics

☐ Gesture Recognition

☐ Animation

☐ Calendar

☐ Core Motion

☐ Core Location / MapKit

☐ Core Audio

☐ Firebase (if not used to fulfill Major Element requirement)

☒ Core Data (if not used to fulfill Major Element requirement)

☐ Other (such as QR code, Koloda, etc.) with approval from the instructor – list them