Software Documentation

For

Receipt Reader

William Ent, Julio Espinola, Jacob Fisher, Brian Johnson, Mason Ringbom, Nathnael Seleba, Jatinder Singh, Ibrahim Sydock, Ichinnorov Tuguldur

Central Washington University

March 1, 2023

Module 1: SignInScreen.js

The first screen that is displayed on the application. Currently set to sign in, with the option for the user to either utilize the functionality of the first page, or request a password change, or create a new account.

Module 1.1 Sign In:

onSignInPressed = async (user, pass):

Checks the username and password fields to make sure something is written there and alerts the user to the missing data. Then it will pass the username and password variable to the database sign-in function.

Export async function signIn(username, password):

Takes in the username and password and checks the Supabase database. Checks for the username and password and if there is a match return the user_id or a -1.
Then it gets the returned user_id and put it into a variable and checks if the variable is -1 for an invalid user login and alerts the user otherwise it replaces their screen with the (module 2) homescreen.js and sends the user_id as a parameter.

onCreateAccountPressed ():

Replaces the current sign-in screen with CreateAccount.js.

const onForgotPasswordPressed ():

Replaces the current sign-in screen with ForgotPassword.js.

Module 1.2 Create Account:

const onSignUpPressed = async (user, pass1, pass2):

Checks to make sure that all three fields are populated and both the passwords match each other. Then it passes the username and password to the database function signUp.

Export async function signUp(username, password):

Hashes the password then inserts the username and hashed password into the database and throws an error if the username is already taken and returns the user_id and username.

Takes the returned user_id and username and puts it into a variable then alerts the user a welcome with their user_id and username and replaces their screen with the (module 2)Homescreen.js.

Module 1.3 Forgot Password:

onResetPressed = async (user, pass1, pass2):

Checks if all the fields are populated and that both the passwords match. Then it calls the database function forgotPassword.

Export async function forgotPassword(username, newPassword):

Searches the database to find the username specified and updates the password with the password provided. If there is an error it will throw the error and return 0 if the password was successfully changed and -1 for an error.

Takes what the forgotPassword function returns and puts it into a variable then if the variable is a 0 then replaces the screen with the original SignInScreen.js or alerts the user that it was not able to change the password.

Module 2: HomeScreen.js

The home screen is where all the navigation happens. After the user signs in it will default to the dashboard screen.

Module 2.1 Dashboard:

The dashboard screen shows the current budget and what has currently been spent. Then there is a button that allows the user to set a new budget amount. Underneath the budget information and budget, it displays the last five submitted receipt pictures. Then at the bottom of the page are the tab navigators to the other three pages.

Module 2.2 Expenses Screen:

The expense screen is going to be where the user can see how much they spent at the grocery store via graphs. These graphs currently come in weekly, monthly, quarterly, and yearly forms.

const updateWeekly = async ():

Updates the chart to show the expenses for the current week and the previous week's

const updateMonthly = async ():

Updates the chart to show the expenses for the current month and the previous 3 months.

const updateQuarter = async ():

Function that updates the chart to show the expenses for the current quarter and the previous 3 quarters.

const updateYearly = async ():

Function that updates the chart to show the expenses for the current year and the previous 3 years.

Module 2.3 Camera:

The camera screen is where users can take photos of their screen. On the camera screen there are three buttons: for taking a picture, uploading a picture, and for flashlight.

let takePic = async ():

Nothing gets passed, saves taken photo on gallery and to the database

const pickImage = async ():

Nothing gets passed, let's the user pick an image from their gallery.

const [flash, setFlash] = useState(Camera.Constants.FlashMode.off);

setFlash(): nothing gets passed, nothing gets returned. Ability to turn on the flashlight on and off.
When photo is taken user going to have 3 option: save, retake, and crop

Module 2.3.2 Save:

Let savePhoto = async():

Takes in no parameter and doesn't return anything but navigates the screen to editToDb screen after the save button is clicked.

const url = await uploadReceipt(photo, userParams.userid);

Pass to the OCR that returns receipt data based off url

let receiptData = await getDataFromOCR(url);

Calls the database function “getDataFromOCR” which passes to the OCR a URL and expects a receipt object in return. If there is no receipt object, error checking is resolved on “EditToDB.”

Module 2.3.3 Retake:

setPhoto(undefined)

Takes the parameter undefined so that users and retake a photo

Module 2.3.4 Crop:

const launchEditor = (imageObject):

Launches the image editor when this function is called and an image object is passed through. The Image editor is what allows us to crop in freeform on both Android and IOS.

function imageUriToBase64(uri, callback):

Since the image editor does not return the base64 of the image we use this function that takes a image uri and returns the base64 of the image based off its uri

onEditingComplete=(cropResult):

Once the photo is done being cropped we get its base64 and pass it as an image object to setPhoto

Module 2.4 Profile:

At the top of the page, it displays a welcome from us with the username and the number of receipts that the user has uploaded.

const onSignOutPressed = ():

Gets nothing to pass to it and doesn't return anything from it. It will replace the current profile screen with the Sign In page.

const onAboutPressed = ():

Gets nothing passed to the about page and nothing gets returned from the about page.

About.js

Displays that we are a group project from CWU and to meet our team. It shows all of our pictures with our names next to them. With a back button that just returns you to the previous page: profileScreen.

Module 3: EditToDB.js

The edit to database screen is going to be where the user can manipulate the data returned from the OCR and ensure that the data picked up is in the appropriate format. The most important qualities of the receipt currently is the "total" "store name" and "date." These aspects are "higher level" attributes of the receipt, isolated from the array of items and their corresponding prices.

const changeItemContent = (text, index):

Updates the item content within the provided array of items, based on the index.

const changePriceContent = (text, index):

Updates the price content within the provided array of items, based on the index.

const removeItem = (index):

Removes the item that the user selects, based on the index.

const addInput = ():

Adds an input at the bottom of the array of items, does not take in any arguments.

const updateReceipt = ():

Performs input validation to ensure proper data is provided, if proper data is not provided there is either automatic correction or the action to save the receipt is invalidated. Does not take in any arguments.

const postData = async ():

Utilizes the output from the updateReceipt function to post the data from the mobile app to the database. Navigates the user back to the dashboard clearing the stack navigation. Does not take in any arguments.