System and Unit Test Report

RememberMe

6/1/2021

System Test Scenarios

Sprint 1:

A. As a user, I would like to be able to keep a journal of memories about friends, family and myself.
B. As a user, I'd like to store a multitude of data types such as text, contact information, photos, etc.

Scenario:

1. Have Main Page for contact information:
    a. Header and side menu page
    b. Add "plus" button to add new contact profile
2. Side Menu page
    a. Setting button
    b. Logout button
3. Have Create Contact page
    a. Make a text box and enter a text entry.
        i. <Text Input>
4. Created contact saves on main page,

Sprint 2:

A. As a user, I would like to efficiently catalog my data.
B. As a user, I would like to categorize data entries based on the person they are related to.

Scenario:

1. Update Create Contact page
    a. Default empty profile picture icon
    b. “Choose profile photo: button activates Raw Bottom
        i. “Take from camera” button
            1. Opens up camera and image cropper
        ii. “Choose from gallery” button
            1. Opens up android camera roll, google drive, or google photo
    c. First Name = <First>
    d. Last Name = <Last>
    e. Relationship = <Test>
    f. Birthday = <YYYY-MM-DD>
    g. Phone Number = <Select Country Button> <XXX-XXXX>
    h. Address = <Test>
    i. Text Box = <Test>
    j. Press Submit button
        i. Saves onto the main page list
2. Store Data into Firebase

Sprint 3:

A. As a user, when I change devices I would like to be able to transfer data easily.
B. As a user, I would like my data stored reliably in an account.

Scenario:

1. Create Login Page
    a. Email = <test@test.com>
    b. Password = <Test>
        i. Has to be up to 6 characters long
    c. Submit Button
        i. If valid, will bring to main page
        ii. If not valid, error message of no contact
    d. Registration button

            i. Leads to registration page

2. Create Registration Page
    a. First Name = <Test>
    b. Last Name = <Test>
    c. Email/username = <test@test.com/test>
    d. Password = <Test>
    e. Click submit button
    f. Receive alert that sign up was successful

Sprint 4:

A. As a user, I would like assistance in recognizing the faces and names of recorded individuals.
B. As a user, I would like to pull up their contact information through facial scanning.

Scenario:

1. Show image library
    a. Click button leading to phone gallery
2. Select from Camera
    a. Access to camera
3. Save New Face
    a. Alert to confirm if user want to add new face
        i. User selects yes button, message send to server
            1. Receive message update successful
            2. Clicking button will add add new name to server
        ii. User selects cancel button, process stops
4. Typing in new name in textbox
    a. Enter new face name here: <First Last>
5. Select photo and select button to recognize faces in image
    a. Message “Recognized status is ‘loading’ “
    b. Server processes image and request
    c. Response given and Recognized status displays name(s) of people recognized in photo
        i. Yellow box displays name identifying face <First Last>
    d. Secondary image container displays photo with boxes and names drawn across recognized people
6. Typing name and choosing button to remove from server
    a. Alert to confirm if user wants to delete person from server

i. User selects yes and message sends to server to process
    1. Receive message that update successful

ii. User cancels, process stops

Unit Tests

Sarah Liang

Contact List UI

Lists all the contact profiles that the user can add with profile photo.

Manual Testing

- Open up a mobile app.
- Scroll through the main page if the number of contacts exceeds page size.
- Click + button on the bottom right side to create contact, directed to create contact page.
- Open modal page with privacy policy and terms of service.
- Click on the side menu icon to prompt the side menu drawer.

Sidebar Menu UI

Displays icon, settings page, and logout buttons.

Manual Testing

- Open up a mobile app.
- Click the setting button and the icon to prompt the setting page.
- Click on the logout button prompts alert to logout. Selecting yes leads to a login page with sign-in information.

Create Contact UI

Create a new contact profile with personal information such as first name, last name, relationship to the user, birthday date, phone number, address, and any other additional information.

Manual Testing

- Open up a mobile app.
- Click + button on the bottom right side to create contact, directed to create contact page.
- Click choose photo button
- Click take from camera button
- Take photo with photo camera
- Adjust crop sizer
- Click choose from gallery button
- Either choose existing photo from android phone gallery or from google photo/google drive

- Click on the First name text input box and typed test text.
- Click on the Last name text input box and typed test text.
- Click on the Relationship text input box and typed test text.
- Click on the Birthday date picker and choose the date button on the calendar.
- Click on the Phone number text input box, first the country picker, then enter in digits.
- Click on the Address text input box and typed test text.
- Click on the extra text input box and typed test text.
- Click on submit button and redirect to contact list page.

UI / UX Design

1. Styling of Main contact page, side menu and create contact
   Manual Testing:
    - Resized buttons and positions
    - Text sizing and font
    - Added icons, logo and their size and position
    - When buttons clicked, leads to respective pages/drawers

Dillon Lee

New Contact Page UI

Choosing Profile Photo

- Implemented the feature for a user to upload a profile picture from the camera or by the local storage on their phone
- Manual Testing
  - Console logs a message when a user clicks on “Choose Profile Photo.”
  - Console logs a return message when a user clicks onto either “Take From Camera” or “Choose from Gallery.”
  - Downloaded an image from Google into the emulator to check if it saves locally. Once the image is saved, I used the image to upload it onto the entry
  - Opening the mobile application

Settings Page UI

Creating a Settings Page

- Implemented the settings page for the user to adjust the users’ personal settings. The settings page includes their information, account, display contacts, sorting, name format, export and import profile, and blocked numbers.
- Manual Testing
  - Opening up the mobile application
  - Console logs a message when the user presses “settings.”
  - Console logs a message when the user presses the different buttons in the settings page.

Contacts Detail UI

Adding Contact Detail Header

- Implemented the contact detail header to display the users’ contact information. The contact header includes adding a trash and star icon.
- Manual Testing
  - Opening up the mobile application
  - Console log a message when the user presses the trash icon.

    - Console log a message when the user presses the star icon.

Gareth Samadhana

<u>User Authentication (Register, Login, Logout)</u>

Creating a Login/Register

- Implemented fully functioning user login, register and logout functions
- Connected each respective user account to Firebase
- Created inputs for first name, last name, email, and password for the Register along with a submit button
- Created inputs for email and password for the Login along with a submit button
- Created a place to input the logo for the login and register page at the top
- Manual Testing
    - Opened app to view the login and register page
    - Made sure that the links to go to the register page from the login page and the login page from the register page were working
    - Printed the forms that the user input to register/sign in for verification
    - Tried making edge cases for the register page by not submitting an email, an email that was already used, a fake email, an empty first name/last name/password
    - Tried making edge cases for the login by inputting a wrong email and password
    - Checked Firebase to see if when the user clicks “submit”, an account is created

Creating a logout function

- Implemented a logout function that logs them out of the app and back into the login screen
- Manual Testing
    - Clicked the logout button and said “Yes I want to logout” when given the question of “Are you sure you want to log out”
    - After getting logged out, I logged back in and logged out again to double check

<u>Contacts Details Page</u>

Accessing Contact Details

- Allowed for user to click on contacts on the contact list in order to access information regarding a contact

- Used <Touchable Opacity> and onPress{} specifically in order to allow this, as well as passing in the information regarding each contact
- Used <Swipeable> to allow user to swipe on the contact to favorite them
- Created the Contact Details page to have the title based on what the contact's name is
- Created an "add favorites" icon, a "delete contact" icon, and an "edit contact" button
- Manual Testing
  - Logged in and tested to see if contacts you created were clickable
  - Checked to see if the functions in the Contact Details were working as expected
  - Checked to see if you could swipe a contact
  - Checked to see if you could delete a contact
  - Checked to see if you could edit a contact you already made

Grant Fu

React Native Page handling image recognize/new/remove requests and response

- Implemented homepage with buttons for choosing photos from library and camera, removing a profile from server, adding a profile to server, and request to get recognized faces
- Manual Testing
    - Choosing photos form library and camera:
        - Use RN console.log() and physical display to 1. Save data from chosen photos to RN state and to 2. Display image on RN screen when uses
    - Removing a profile from server:
        - Observing Flask server behavior
        - Use console.log and physical display when tapping the save new face button for messages that show that profile/person has been removed.
        - Using face recognition function to make sure a person is no longer recognized
    - Adding a profile to the server:
        - Observing Flask server behavior
        - Use console.log and physical display to show what images chosen and names/status upload to server
        - Use face recognition function to make sure a person previously unknown is now recognized
    - Recognizing person(s) form a photo
        - Observing Flask server behavior
        - Use console.log and physical display to show images chosen and sent to flask server
        - Once response is given:
            - Use image container to display image of recognized faces
            - Used console.log and text display to show list of recognized people
    -

Python Flask server with face_recognition

- Implemented Python Flask server with routes to handle requests to add/delete profiles, recognize people from images, and return images per request
- Manual Testing:
    - Requests to add/delete profiles:
        - Observing directories and json files from vscode and windows file explorer
        - Using print statements from Python to display changing data from add/remove
    - Requests to recognize person(s) in profiles:
        - Use print statements to display passed data(s) and comparison functions inside flask server to add tweaks to numpy vector comparison
        - Use PIL module to draw recognition label boxes on images and display on desktop
    - Returning images from Server:
        - Using Flask status code eg 200, 404, 500, etc in terminal,
        - Webbrowser and React Native image container to make url requests of image and display

Gavril Tango

Login/Register Page

- Added Firebase connection to store users and log users in
- Register new users to application
- Fixed user login interface

Manual Testing

- Create new users in “Register” screen
- Check Firebase Firestore for newly created user
- Login to application through Firebase

Contacts UI

- Added form to create new contact
- Sent form information to Firebase backend
- Retrieved previously stored contacts from Firebase to display contacts
- Displayed contacts for a single user
- Added sorting by date, first name of contact and last name of contact
- Added ability to open contact profile and pull up their information
- Added ability to edit contact information

Manual Testing

- Login in with pre-existing account
- Click on adding new contact
- Fill in full information
- Submit form
- Pull up contact information again
- Edit contact information
- Check Firebase for updated information

App UI

- Changed app name on app store
- Changed logo for app store

Manual Testing

- Load phone emulator