# React Native Technical Challenge

## Introduction

The purpose of this challenge is to analyze the candidate's technical knowledge of the following skills:

- Library react-native
- TypeScript language
- Implementation of screens and components
- Access API services
- State management
- Import and configure third-party libraries
- Code organization (clean code)

## Task

The candidate must implement a simple project with 4 screens following the specifications below. **The candidate has 5 challenges and the candidate does not need to complete all the challenges to submit the project for evaluation**. However, the more challenges the candidate completes, the better they will be evaluated.

## Timeline

The candidate has up to two weeks to submit the challenge, do it in your time and if you have any questions please contact anderson@gooseinsurance.com.

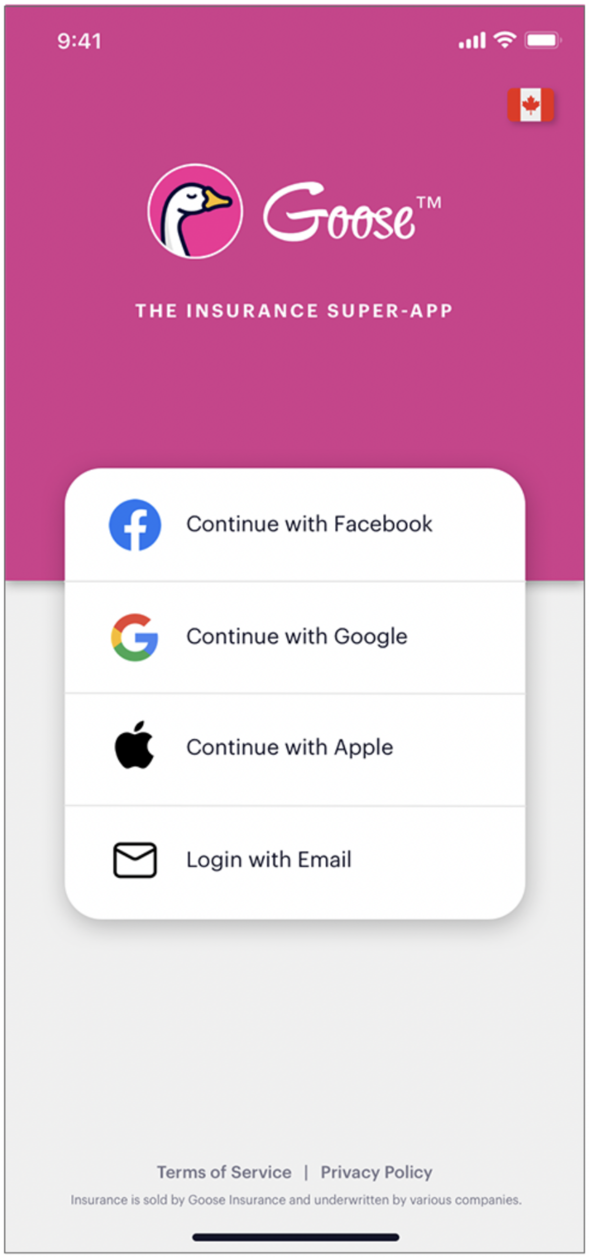
## Submission

Upon completing the challenge, the candidate must make the project available on GitHub and send the repository link to email anderson@gooseinsurance.com

Good Luck.

## Screens to Implement

**SCREEN A**


9:41

THE INSURANCE SUPER-APP

Continue with Facebook

Continue with Google

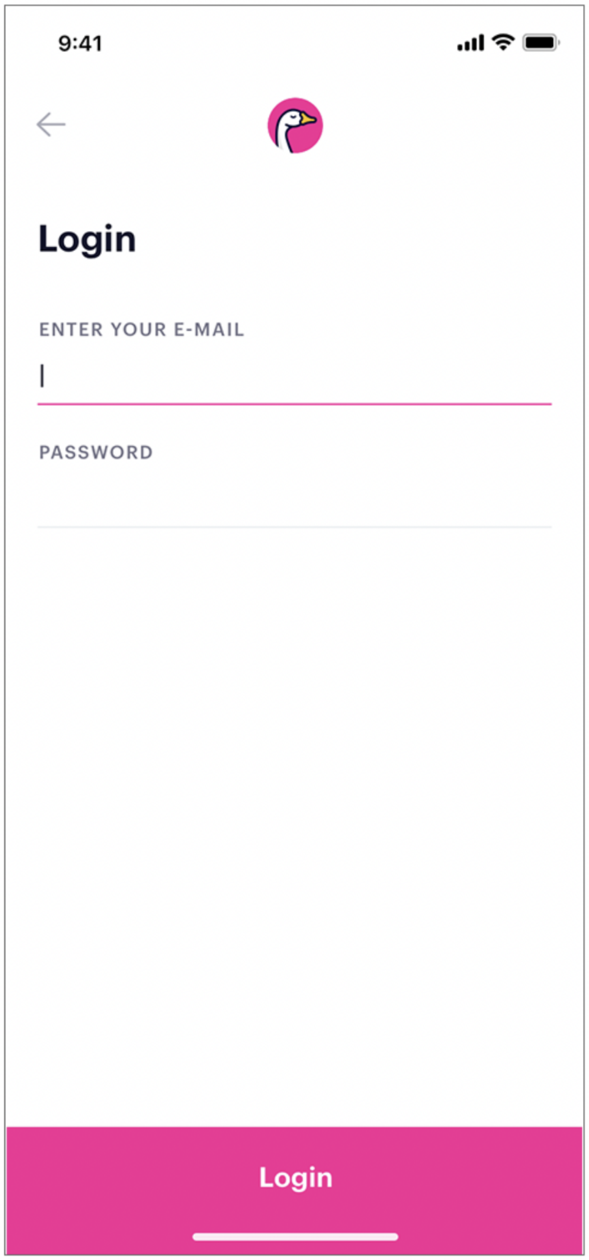
Continue with Apple

Login with Email

Terms of Service | Privacy Policy

Insurance is sold by Goose Insurance and underwritten by various companies.

**SCREEN B**


9:41

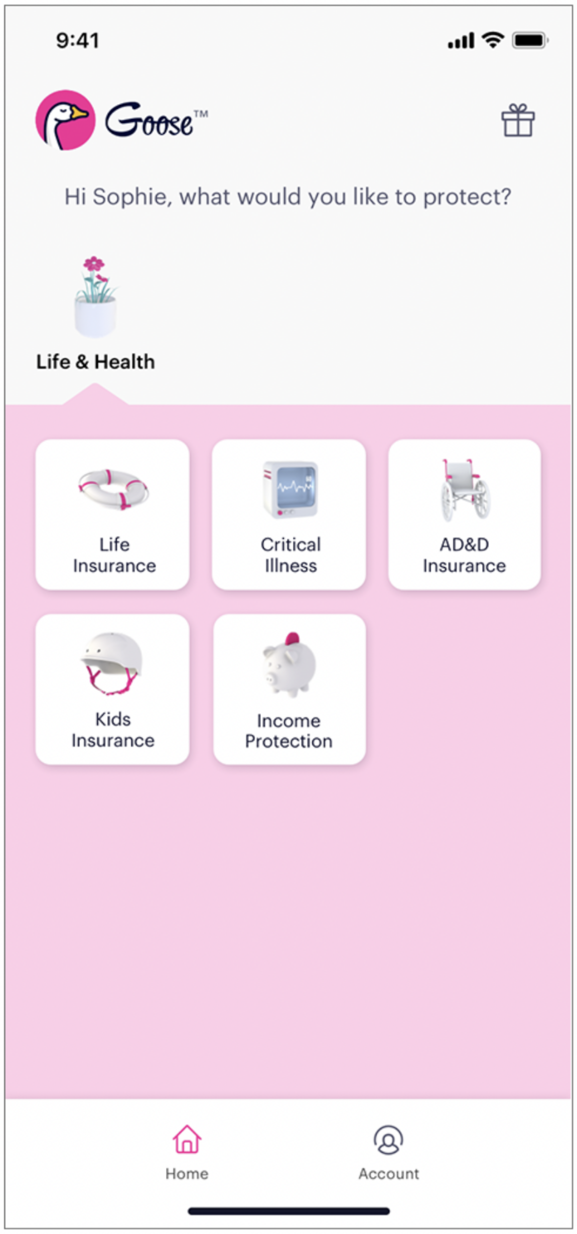
Login

ENTER YOUR E-MAIL

PASSWORD

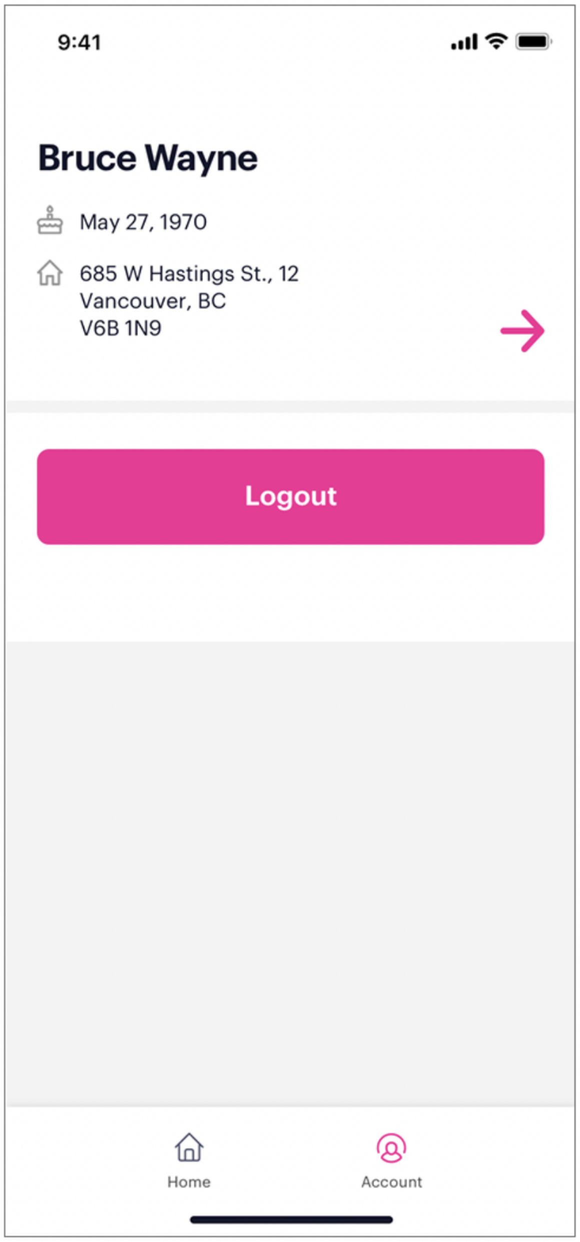
Login

## SCREEN C



Hi Sophie, what would you like to protect?

## SCREEN D



Bruce Wayne

May 27, 1970

685 W Hastings St., 12
Vancouver, BC
V6B 1N9

# List of Challenges

## Challenge 1: Initialize Project

- Initialize the project [ `npx react-native init gooseinsurance` ]
- Import third-party libraries, if needed. Axios, reactnavigation, redux, styled-components etc

## Challenge 2: Screen A

- Implement **Screen A**
- Implement components, if needed
- On this screen, only the "Login with Email" button will have an action. Clicking on this button should navigate to Screen B
- Implement navigation between screens (StackNavigator)

### Extra info:

- The images for the implementation of this screen are in the screen_A folder of the attached file
- The Facebook, Google and Apple buttons will have no action

## Challenge 3: Screen B

- Implement **screen B**
- Implement components, if needed
- Clicking on the arrow image "<-", you should navigate back to Screen A
- Clicking on the "Login" button, you must call the API of the login service
- If the service response was successful (status 200) the service information should persist in the global state (redux)
- Navigate to Screen C

### Extra info:

- The images for the implementation of this screen are in the screen_B folder of the attached file
- Login service specification:

- **API Endpoint: POST** https://gslwn81z5i.execute-api.us-east-2.amazonaws.com/goose/technical-challenge/login

- Body:

```
   {
      "email": "candidate@gooseinsurance.com",
      "password": "gooseinsurance"
   }
```

- **Success response (status 200):**

```
{
    user: {
       name: "Bruce Wayne",
       birthday: 1990-12-12,
       address: "1281 West Georgia St, 800, Vancouver, BC
V6B 5N6"
    },
    products: [{
       id: 1,
       title: "Life Insurance"
    },
    …….]
}
```

- **Error response (status 400):**

```
    {
        error: "unregistered user"
    }
```

## Challenge 4: Screen C

- Implement **Screen C**
- Implement components, if needed
- Implement TabNavigator navigation (home and account)
- Retrieve the user name from the global state and insert in the text "Hi **<USER_NAME>**, what would you like to protect?"
- Retrieve the list of products from the global state and implement their rendering
- Clicking on the "account" button, Screen D should be displayed

### Extra info:

- The images for the implementation of this screen are in the screen_B folder of the attached file
- On this screen, only the StackNavigator buttons will have action (navigate between the home and account screen)

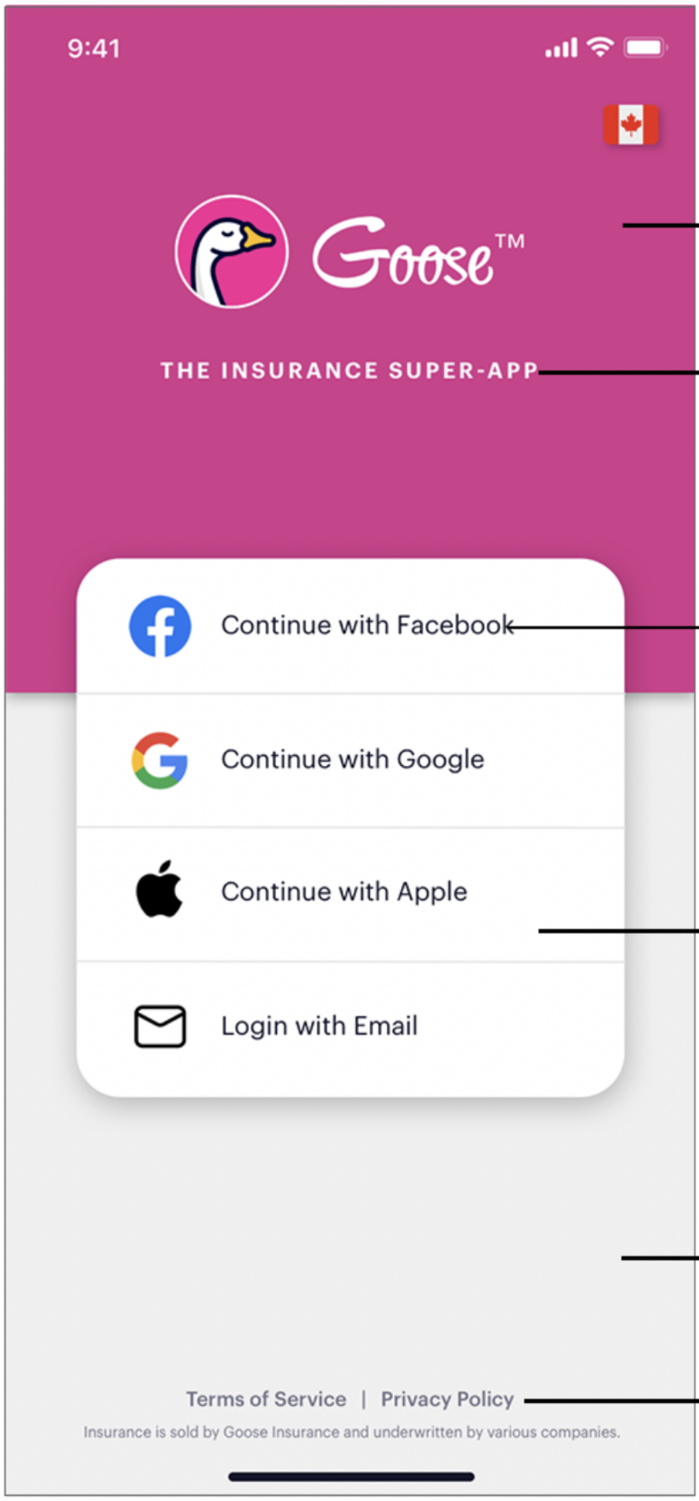
## Challenge 5: Screen D

- Implement **Screen D**
- Implement components, if needed
- Retrieve user information from the global state: name, birthday and address
- On this screen, the "Logout" button will have action and when clicked, it should take the user to the initial screen (Screen A)
- Clicking the "logout" button, screen A should be shown

### Extra info:

- The images for the implementation of this screen are in the screen_B folder of the attached file

## Screen Specifications



Insurance is sold by Goose Insurance and underwritten by various companies.

● #D5398D
background

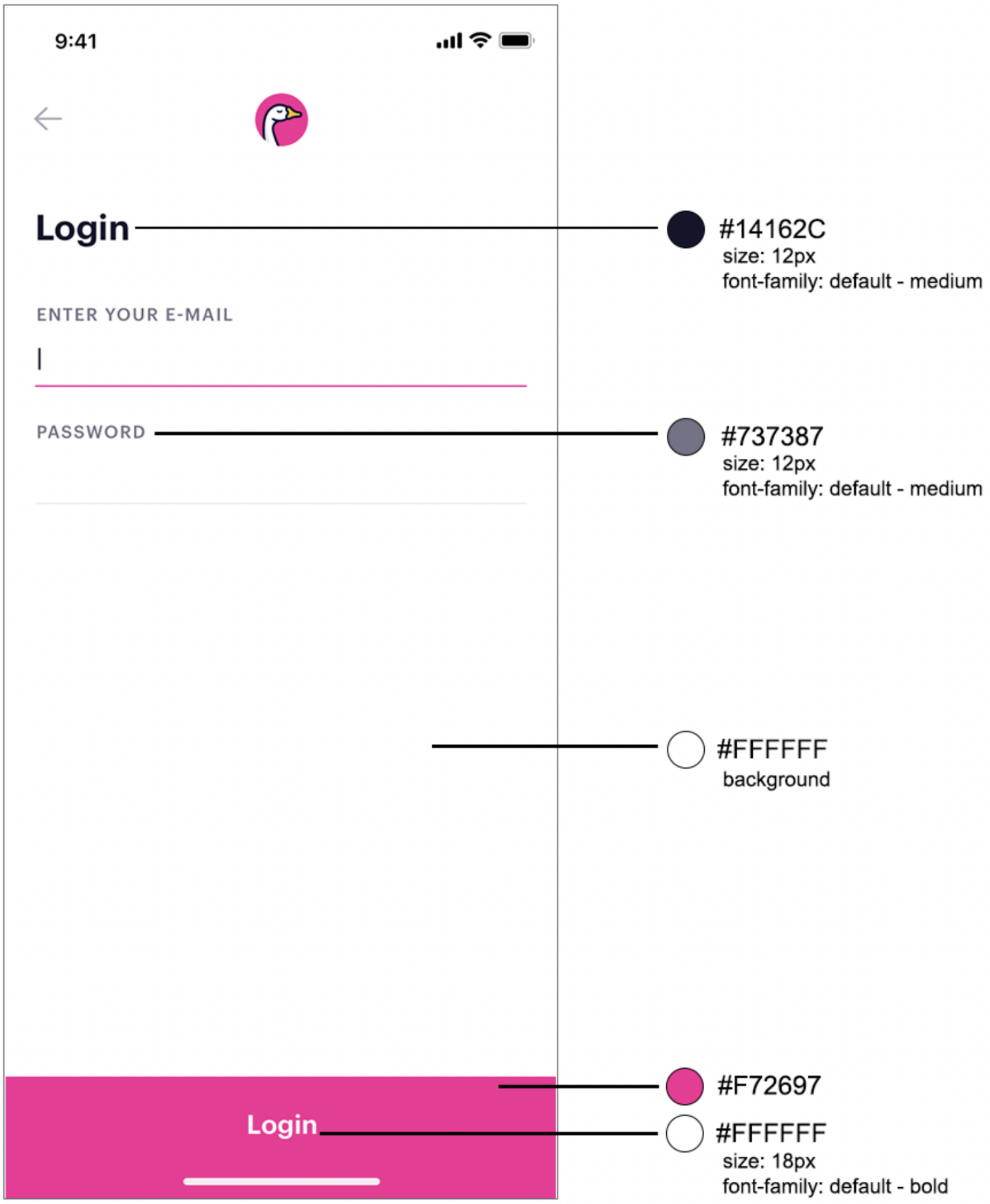
○ #FFFFFF
size: 18px
font-family: default - bold

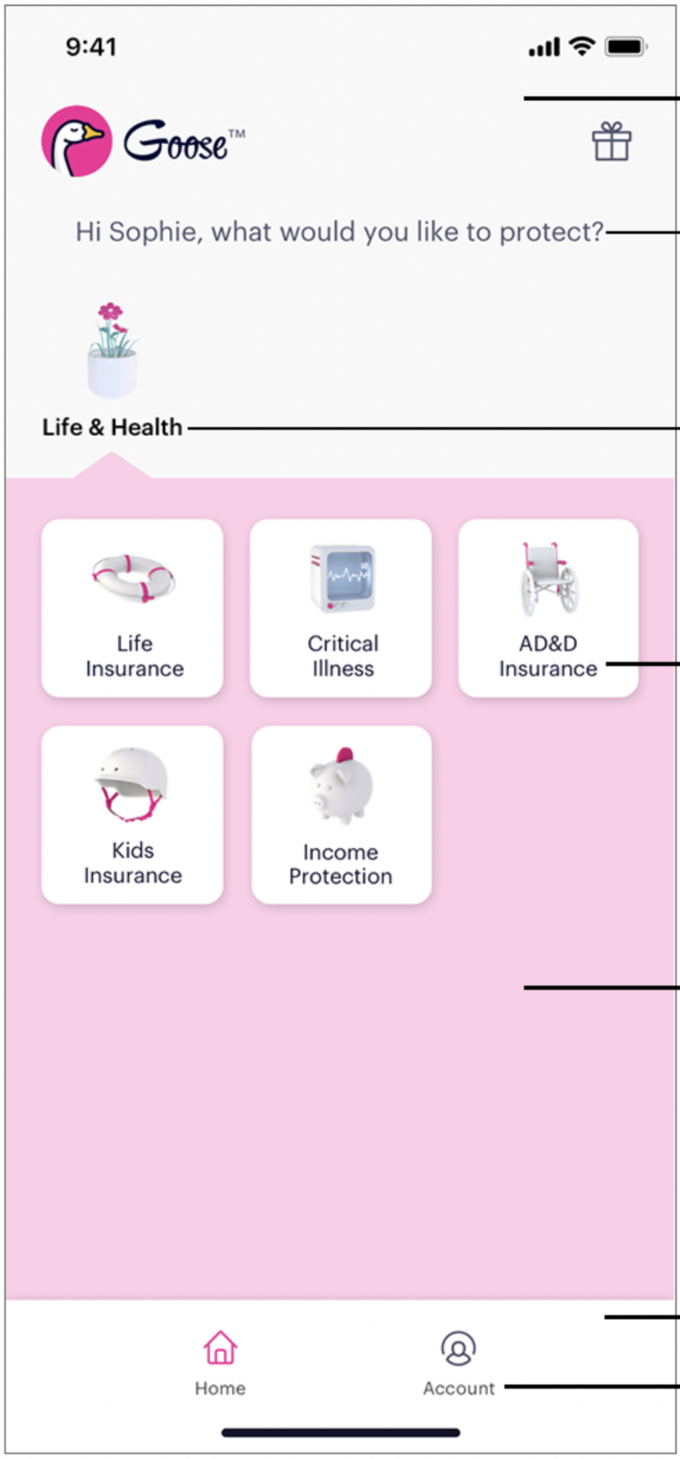
● #000000
size: 14px
font-family: default

○ #FFFFFF
background

○ #D5398D
background

● #6C6C81
size: 11px
size: 8px


9:41
Login
ENTER YOUR E-MAIL
PASSWORD
Login
#14162C
size: 12px
font-family: default - medium
#737387
size: 12px
font-family: default - medium
#FFFFFF
background
#F72697
#FFFFFF
size: 18px
font-family: default - bold



○ #F9F9F9
background

● #54566F
size: 17px
font-family: default

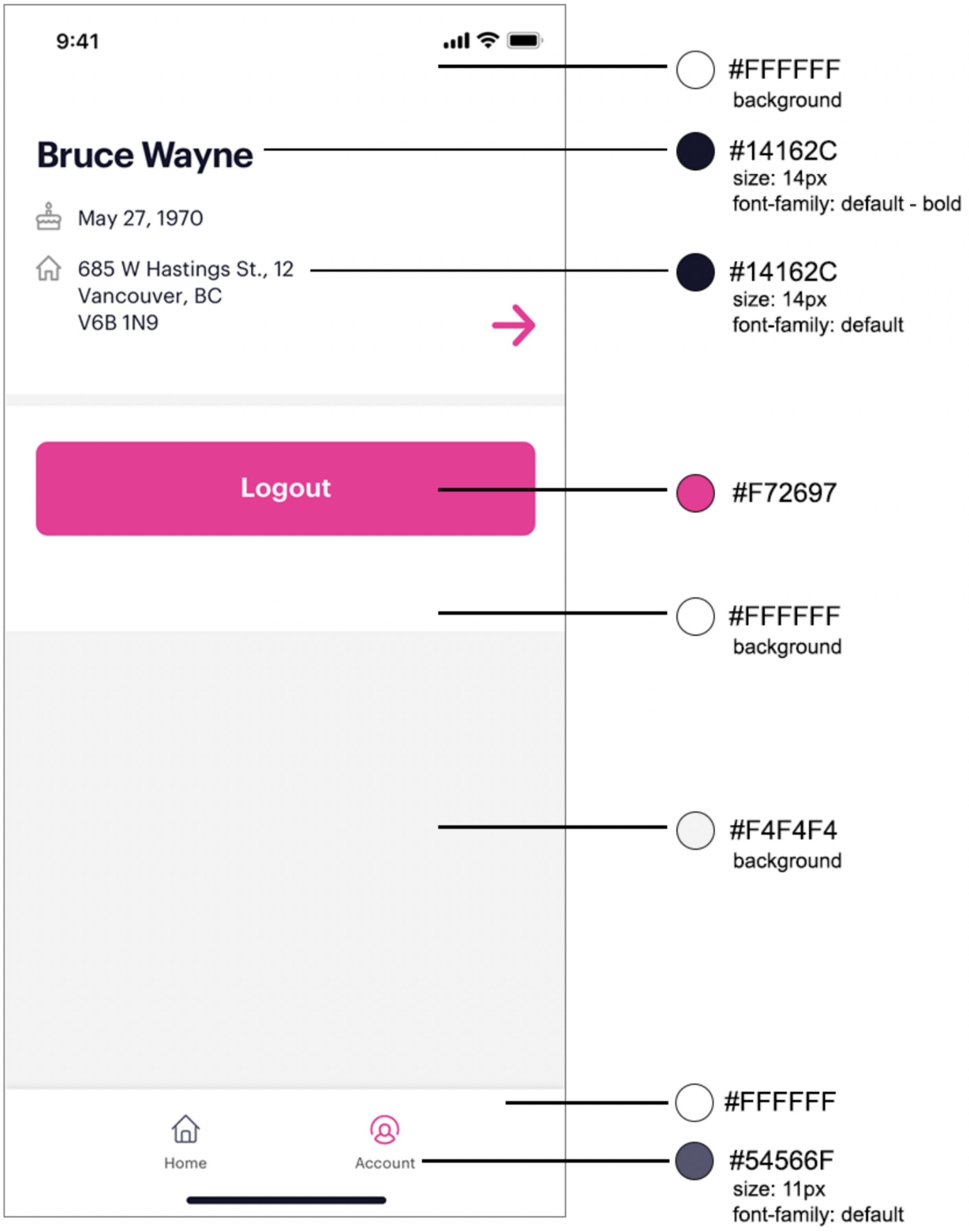
● #14162C
size: 12px
font-family: default - medium

● #737387
size: 12px
font-family: default - medium

● #FFCEE9
background

○ #FFFFFF

● #54566F
size: 11px
font-family: default


9:41
Bruce Wayne
May 27, 1970
685 W Hastings St., 12
Vancouver, BC
V6B 1N9
Logout
Home
Account
#FFFFFF
background
#14162C
size: 14px
font-family: default - bold
#14162C
size: 14px
font-family: default
#F72697
#FFFFFF
background
#F4F4F4
background
#FFFFFF
#54566F
size: 11px
font-family: default