

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

GitHub Username: [finnishslavic](#)

Attached

Description

This is a life-saver or eye-opener for many complicated relationships. Based on a [book](#) "Attached - The New Science of Adult Attachment and How It Can Help You Find - and Keep - Love" by psychiatrist and neuroscientist Dr. Amir Levine and Rachel Heller.

This app provides quizzes and condensed information described in the book, allows you to quickly determine your attachment style and find out easy steps to avoid being driven by automatic protection behavior.

Intended User

This app will help anyone who has difficulties in relationships. It might also give a valuable insight into origins of destructive behaviours in a relationship with your significant other. It might be of great help for those looking for a partner - you'll better know what to look for and who to avoid.

Features


This app has the following features:

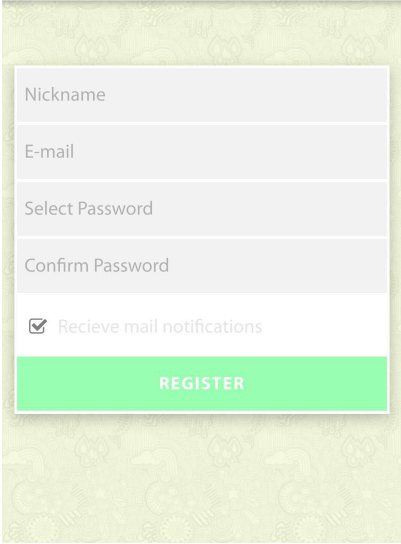
- Authorizes user using Google account
- Provides quizzes to determine attachment styles
- Saves user answers
- Allows see suggestions to given user actions

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1


Attached



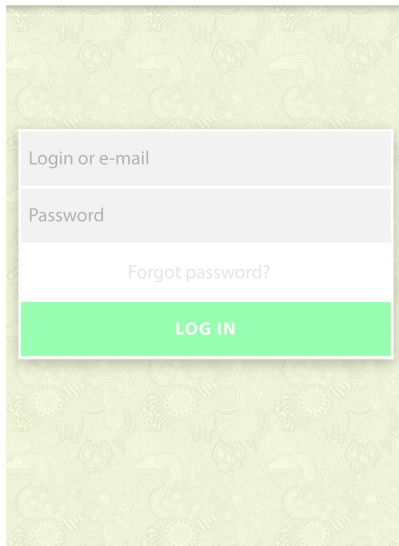
The mockup shows a registration form with the following fields and elements:

- Nickname
- E-mail
- Select Password
- Confirm Password
- ☒ Recieve mail notifications
- REGISTER** button

Signup screen

Screen 2

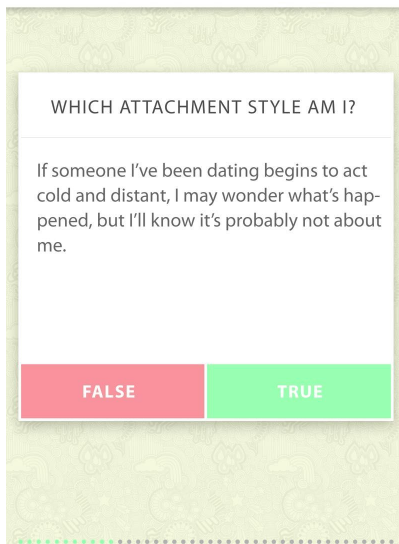

Attached

A login screen mockup with a light green background featuring a repeating pattern of small icons. The login form is a white rectangle with a light gray border. It contains two input fields: 'Login or e-mail' and 'Password'. Below these is a link 'Forgot password?'. At the bottom is a green button with the text 'LOG IN' in white capital letters.

Login screen

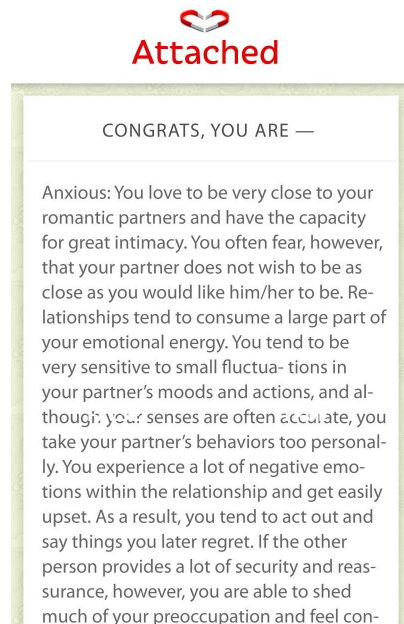
Screen 3


Attached

A quiz screen mockup with a light green background featuring a repeating pattern of small icons. The quiz card is a white rectangle with a light gray border. It has a title 'WHICH ATTACHMENT STYLE AM I?' in bold. Below the title is a paragraph of text: 'If someone I've been dating begins to act cold and distant, I may wonder what's happened, but I'll know it's probably not about me.' At the bottom are two buttons: a red button labeled 'FALSE' and a green button labeled 'TRUE'. A progress bar is visible at the very bottom of the screen, consisting of a series of small colored dots.

This is the quiz question and progress bar on the bottom

Screen 4



The screen with results of the app

Key Considerations

How will your app handle data persistence?

The app will use Realm database to save/restore data. In the first version this app won't rely on having to many entries - it will only store user answers.

Describe any corner cases in the UX.

The main concern is that user leaves quiz unfinished and closes the app. However, I will use Scoop library and hope to utilize its functionality to preserve the state of the app.

Describe any libraries you'll be using and share your reasoning for including them.

Realm, RxJava, Scoop, something for google authorization

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 0: Project architecture

- Outline package structure and architecture to follow (MVP probably)
- Make a very rough estimate on how modules will interact
- Identify problematic places - whether to use dependency injection etc.
- Quiz UI - how to make it easily extendable with new questions?

Task 1: Project Setup

- Setup a github repo
- Create empty project
- Add libraries

Task 2: Implement UI for Each Activity and Fragment

Implement UI using Scoop

- Create MainActivity as a container
- Add main scoop view
- Add login/authorization screen
- Implement template for quiz UI

Task 3: Persistence layer

Add persistence layer:

- Create models to store user answers
- Add quiz data to strings

Task 4: Implement quiz logic

- Implement service to figure out logic for quiz answers and producing the results
- Implement Quiz 1, 2, 3 UI
- Add results screen

Task 5: Add suggestions and history

- Create layout and implement suggestions screen
- Add ability to review answers and see results

Task 5: Get in touch with book authors

When app is ready:

- Get in touch with book authors/publishers in order to get permission for publishing

Task 5: Publish in Google Play Store

If got permission - publish application in Google Play store.

Add as many tasks as you need to complete your app.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"