**GitHub Username**: johntinashe

# Hooked

## Description

Dating app inspired by Tinder

## Intended User

For anyone who is looking for a date.

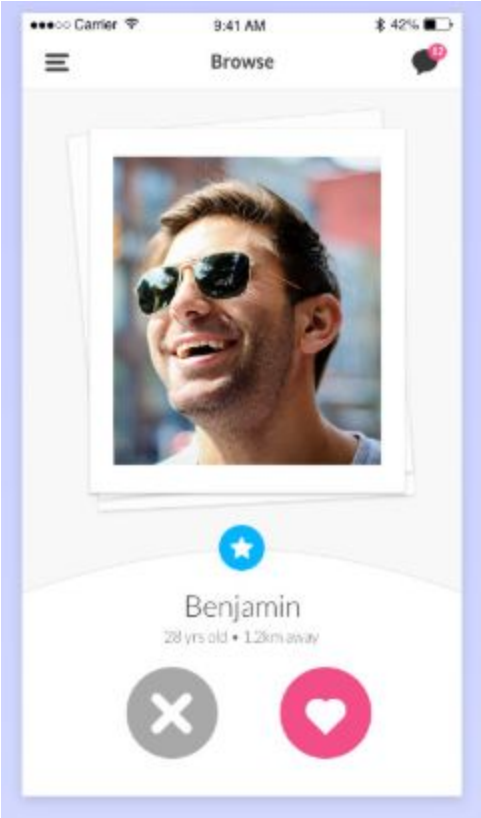
## Features

- Real Time chat
- Browse other users and  look for a match
- Add matches to favorites.

# User Interface Mocks

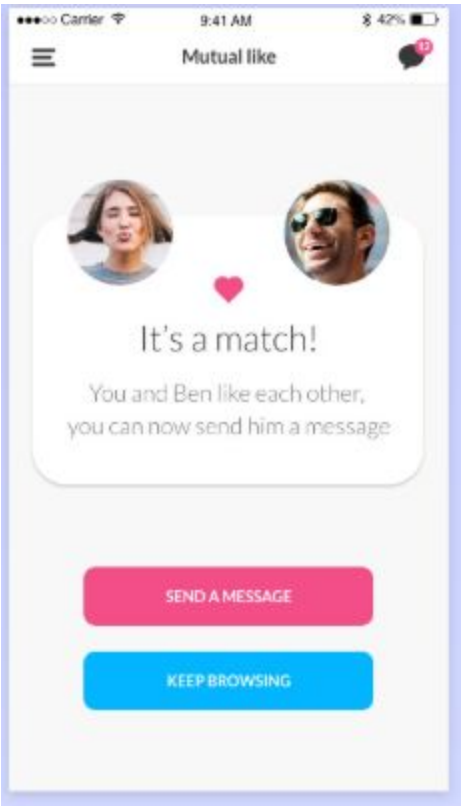
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

## Screen 1



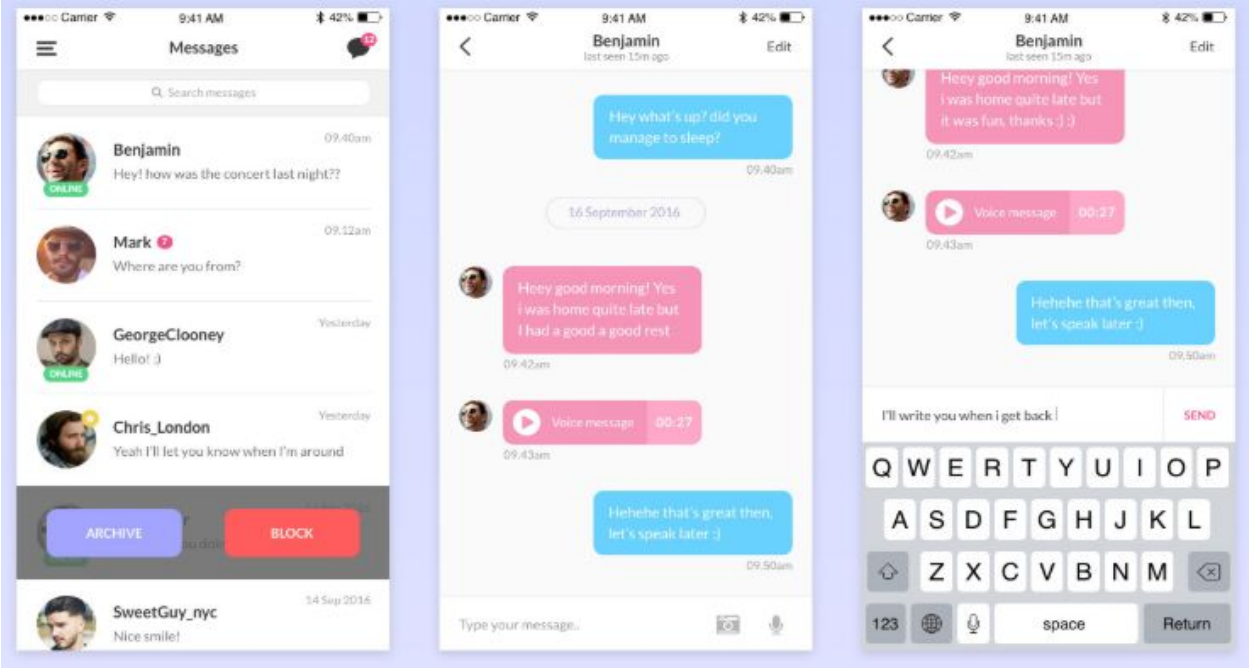

This will be the main activity where the user can browse and look for a potential match.

## Screen 2



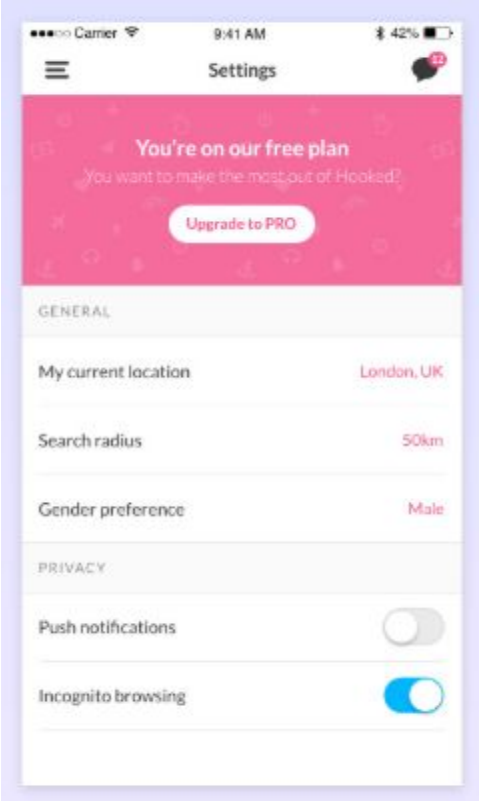

If there is a match both users can send each other messages.

## Screen 3




This will be the messaging interfaces one for the list of chats and the other for chatting.

## Screen 4



**Screen 5**


This will be the widget showing a list of conversations.

# Key Considerations

### How will your app handle data persistence?

The application will use Cloud Firestore to saved all the data.
The user preferences will be stored on the Cloud Firestore database using the Firebase authentication. Based on that information stored, the application's recommendation engine will then discover content and present it to the user.

### Describe any edge or corner cases in the UX.

- The user can also click the home screen widget to open messages activity

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

- ButterKnife (To avoid writing repetitive code just like `findViewById(R.id.yourview)`)
  - `implementation 'com.jakewharton:butterknife:9.0.0-rc3'`
- Picasso to handle the loading and caching of images.
  - `implementation 'com.squareup.picasso:picasso:2.71828'`

**Describe how you will implement Google Play Services or other external services.**

- The app uses google analytics for error reporting
- Uses google ads for showing ads(Test ad).

## Next Steps: Required Tasks

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

### Task 1: Project Setup

- Implement the application using Java
- Setup the project in Android Studio using only stable libraries.
- Configure libraries such as Picasso and ButterKnife

## Task 2: Implement UI for Each Activity and Fragment

- Build the on-boarding screen which explains how to use the application
- Build another Activity UI with fragments for Settings, Login, Register & Browse

## Task 3: Implement the widget

- Implement the widget which displays messages
- On clicking the widget, open the Messages Activity.

## Task 4: Recommendation Engine

- Implement the recommendation engine
- Test the recommendation engine results and optimize it
- The application will use the Firebase JobDispatcher as it will send notifications of new users at regular intervals.

## Task 5: Test the application

- Implement tests to test that recommendation engine
- Write UI tests to test the application

## Task 6: Add accessibility support

- Add support to accessibility by using contentDescription attribute
- Make sure the application uses strings.xml and removes any hard-coded values.

---

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"