# 

# 

# 

# **Human Computer Interaction**

**Project Deliverables**



**your cute meet story…**

**Team Members:**

Suresh Kumar Tathari

Vinay Kumar Calastry Ramesh

Shekhar Singh

Tasmin Chowdhury

Contents

[1. Project overview, purpose, and motivation 1](#_Toc499915550)

[2. Existing system analysis and evaluation: 2](#_Toc499915551)

[3. Methods employed to identify target and potential users: 3](#_Toc499915552)

[4. Methods employed to identify functionality and features: 4](#_Toc499915553)

[5. Usability evaluations and results 5](#_Toc499915554)

[6. Final design features and functionality 7](#_Toc499915555)

[7. Technical and Social Implications 15](#_Toc499915561)

[8. Challenges encountered 16](#_Toc499915564)

[9. Project timeline and team member contributions 18](#_Toc499915565)

[10. Future work 19](#_Toc499915569)

[11. Summary of research 20](#_Toc499915570)

## **Project overview, purpose, and motivation**

Our app aims to follow rules of good, simple design for the Android Operating system following the design guidelines put forth by Google™ popularly called the Material Design guidelines. Following the principles of Material Design by adopting accepted styles in iconography, color palettes will ensure that the idea of consistency will be maintained not only within our app but also with the general design of the Operating System and many first and third-party apps available on the Play Store as well as providing a unified design with consistent sizes and shapes across devices and platforms. The typography, grids, space, scale, color, and use of imagery, more than just pleasing the eye, establish a hierarchy and helps the user maintain focus. This conscious effort on detailed design does a good job at providing a logical flow and help the user find their way around the app much faster than usual and helps prevent errors in using features or navigating around the app. Another effect of following these guidelines is that it helps to implement a clear feedback mechanism with subtle animations.

URL for the guidelines of Material design:<https://material.io/guidelines/>

The second aim is to develop a secure messaging platform as well as maintain a strict privacy profile of every user and allow only the information they want to make public as criteria for the matching algorithm. We will allow the user to report a profile in case of any wrongdoing and handout bans on sufficient reports.

By following a microservice architecture in favor of a monolithic one, we ensure that new features can be developed and maintained independently of existing features as well as removing the burden of learning a new programming language/framework like Node.js or Python(Flask) when starting to develop new features. The developer will be free to choose any programming environment as long as a RESTful API architecture is followed. As mentioned before, using such an architecture will aid us in selectively scaling heavily used features and not have to add that extra memory, storage or processing power unnecessarily. This will also ensure high availability as backend maintenance can also be done selectively. Only those Web Services controlling the features to be maintained/ updated or removed will be brought down and this will ensure that the rest of the app does not collapse and provide nearly complete or partial usability during the maintenance window.

If allowed to develop the ultimate app, we will aim to develop a policy on the lines of Quora’s BNBR (Be Nice Be Respectful) policy so that users on the platform are not harassed and in case of misuse, the system would hand out appropriate bans commensurate with the seriousness of the felony. The ultimate design for our app would follow all the material design guidelines to the book without sacrificing its individuality.

## **Existing system analysis and evaluation:**

Tinder is among the first "swiping apps", whose users will use swiping option to choose photos of other users, swiping right for potentially good matches and swiping left on a photo to move to the next one. It is registering more one billion "swipes" per day. Seeing its popular in the masses we choose this application as the base case for studying the implementation and user experience.

The features which made tinder rule the dating over internet are pretty simple theory called gamification, the application gives a twist of a gaming experience choosing other profiles through swipe action either to left or right making the search a fun activity and keeping the users engaged. This is one of the first application to give a new dimension to the dating application.

Tinder has a provision to connect with other social networks and can build a user profile with photos that have already been uploaded. Basic information is fetched and is used to build the profile using the authentication provided from social networking sites after seeking user’s consent for sharing his / her information to the application.

The prominent features of Tinder is its match-making algorithm, it has evolved drastically in catching the nerve of finding people of liked interests. It also has the feature of analysis the social graph. It helps find users who are most likely to be compatible based on geographic location, number of mutual friends, and common interests are then streamed into a list of matches.

Tinder is also good at generating its revenue through advertising and subscription. It has a subscription model of providing services to its users maintaining the course of features under different categories, basic and premium giving some additional features on the premium scale.

Besides Tinder was great going with the current user, it faced few concerns from the public on the privacy matters regarding the access of location without the user permission violating the privacy policies. Later it was resolved to capture the location only on user’s consent.

All these cool features make Tinder one among the top dating application with the user base in millions. The app is most commonly used as a dating app but it also has services, making it a social media application too. Though it is already loaded with all the functionality and it will be added with few latest technologies. We observed that the look and feel of the app is flawless, but there is always the scope for improvement in terms of user experience making it further more accessible. Our effort is to improvise over some aspects with respect to User interface following the guidelines and principles of Design.

We took most of our features from tinder and tweaked to have our touch in User Interface improving user experience and also few features which we felt will be essential for a contemporary dating application.

## **Methods employed to identify target and potential users:**

The **Amante** application will provide a virtual platform for dating, allowing users to connect and communicate with other users of similar interests and likes, to take further from acquaintance to a relationship. It will provide best of service for the users with easing the usage with no compromise of functionality, making an effort to give the best user experience.

The application is intended to service the people of age from 18 to 40 years in finding their love of life and help them engage in relationship with person of similar interests. We feel that people in this age group is quite a norm for finding relationships. And our technology curve will always try to introduce advanced technologies into this app making it astound and fun to use.

This document is intended to give the overview of the Dating application right from its ideation to completion for the technical members. This document contains conception of the idea of Dating application including its various aspects from the design and implementation perspective giving an outline of what technologies are used and help in knowing all the things implemented as part of this project accomplishment. After going through this document, you will have a good idea of all the requirements, technologies considered and the functionality of the application features.

## **Methods employed to identify functionality and features:**

Functionality and Features were identified by analyzing the golden rules for Interface design.

* The UI was purposefully kept plain and simple to ensure reduction in **short-term memory load** and seek **universal usability**.
* Visual elements for buttons and text boxes and colors used were similar in all the screens to ensure **Consistency**.
* When errors occurred, we have tried our best to offer **informative feedback** to the users.
* The Profile section was developed to make sure that it is easy to go back and change details like About Me, Interests and the profile image itself for **easy reversal of actions**.
* The Workflow of the app follows a logical sequence and is kept similar to highly rated apps so that **users feel in control**.

## **Usability evaluations and results**

The process is applied to the six external quality characteristics: functionality, reliability, usability, efficiency, maintainability, and portability.

1. Reliability is influenced by the following mobile limitations:
2. frequent disconnection, variable bandwidth, and limited energy autonomy. Therefore, during the evaluation of this characteristic, these three mobile limitations must be supported using the ISO 9126 measures or by providing other ones specific to mobile environments.
3. The same goes for the Efficiency characteristic (lower bandwidth and limited storage capacity), and the usability characteristic (limited user interface).
4. Android OS provides the reliable layer for interacting the app with the internet, and the app is designed following the guidelines shared by android for user interface design and development.
5. **Usability** is influenced just by limited user interface, since we have observed that most of usability external metrics are influenced by limited user interface.
6. Usability testing was targeted to complete few tasks like
7. Registering
8. Sign-in and Sign-in google
9. Viewing matched Profiles.
10. We tested our application with 4 users from different education backgrounds of engineering and given the above tasks with no training. The users were able to do all the tasks with ease.

We evaluated our application in terms of the following and addressed all the applicable design principles:

* + - 1. **Understandability:**

The User interface was minimal with no redundant activities planned on the application, making the user interface intuitive for users to see and understand where to look for getting the desired tasks accomplished.

* + - 1. **Learnability:**

The capability of the user to learn its application and usage was improvised in two iterations, by adding icons and detailing, By seeing these the user can make out the place holders or information asked for in the application and can look for options going across the application with no memory overhead.

1. **Operability:**

The operability of the application was given the first preference by placing the interactive elements, components of data entry close to the fingers position on the screen while holding the screen.

1. **Attractiveness:** The user interface appearance was built using the common theme shared across all the screens of the application, keeping the application appealing and attractive to the user.

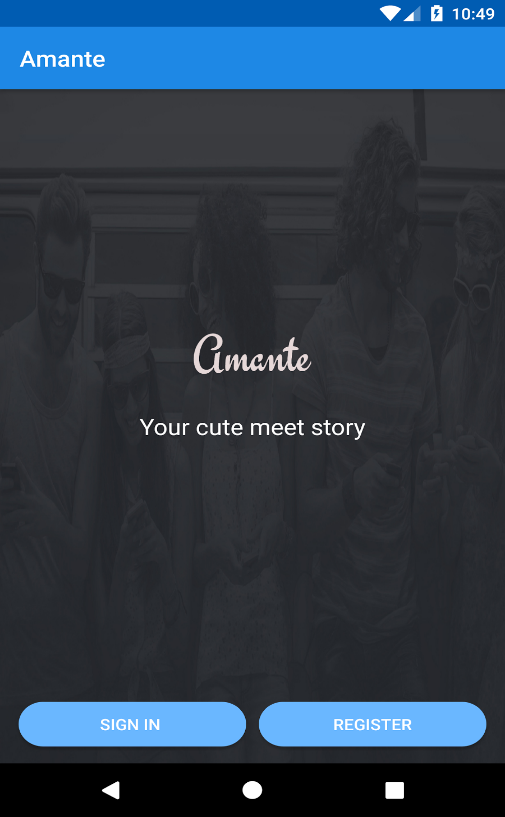
## **Final design features and functionality**

#### **Design Features**

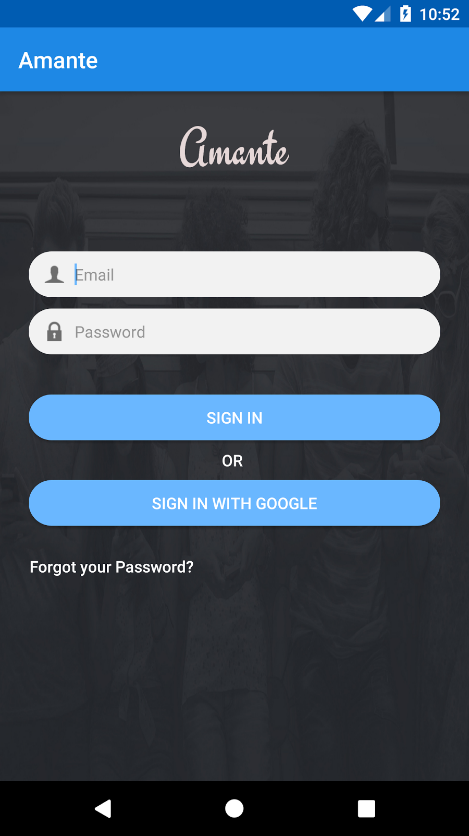
1. Simple Design: Clean and Intuitive design needing no tutorial or user docs to offer explanation.
2. Consistent Interface: Visual Elements used throughout the app are same. For eg: Colors are kept light blue (Reason being blue looks good on mobile phones and is a neutral color - neither liked or disliked by many people and therefore, it is used heavily in social networking sites like Facebook and Twitter. The buttons are of the same size and shape throughout the app.
3. Easy Navigation: A navigation drawer allows the user to switch between different screens in the app without requiring much help. Also, a logical flow is maintained from when the user opens and registers on the app to when the user fills out the Profile details and then gets into the Matches found screen.
4. Login using multiple sources: Everyone today has at least one social networking account and since our app is Android based, we have chosen to include Google SignOn (SSO) and the platform keeps the user signed on until they decide to sign out. The traditional method of using a email/password combination is also provided.

#### **Functionality** Screens

1. Splash Screen: When a user first opens the app, the splash screen is displayed. This provides users with an option to:
   1. Login
   2. Register



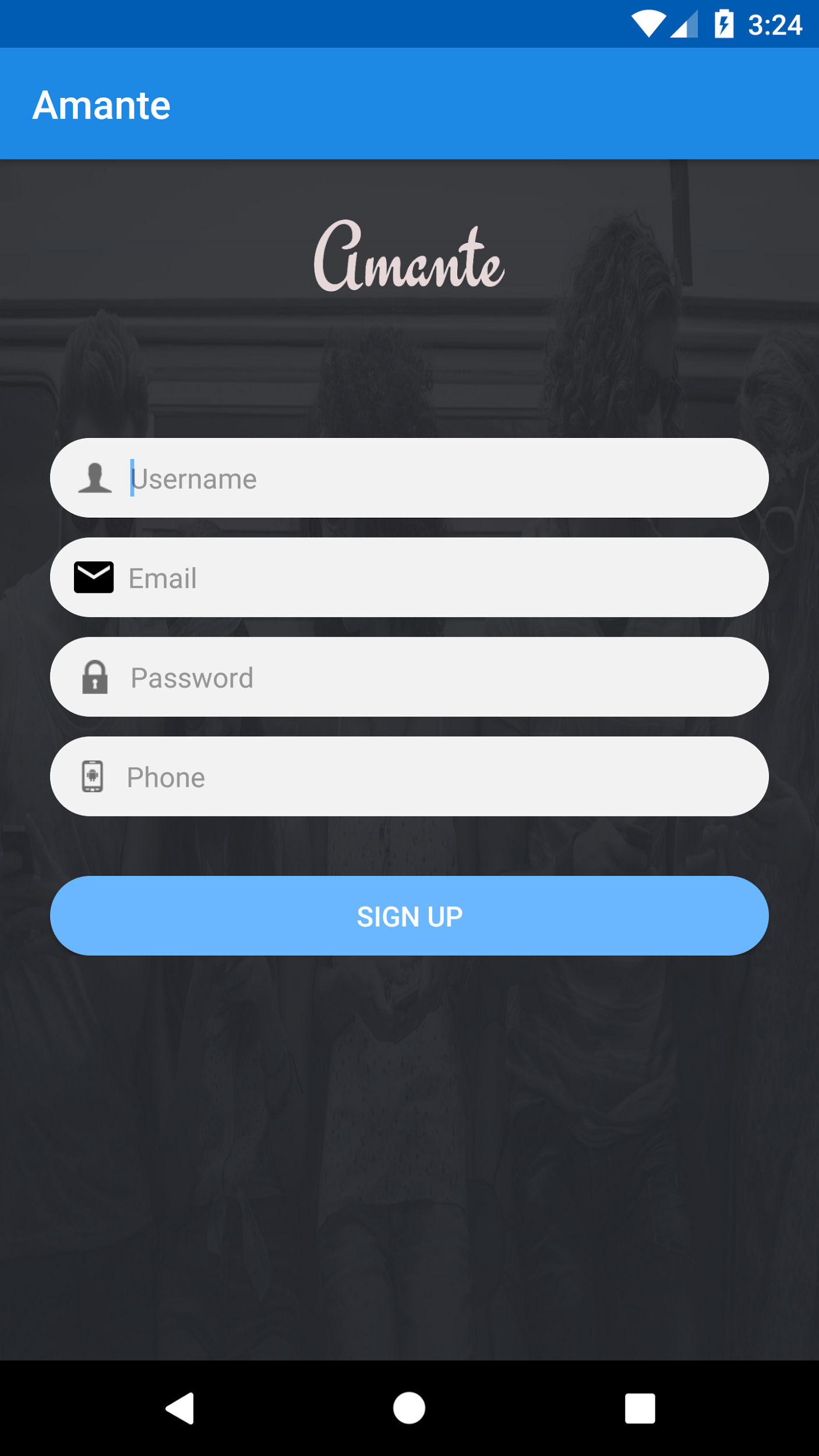
1. Login: The Login page allows the user to login into the app using either a email/password combination or a Google account.



1. Registration: Registration screen allows the user to register using a email/password mechanism.

Details requested from User:

1. Username
2. Email
3. Password
4. Phone

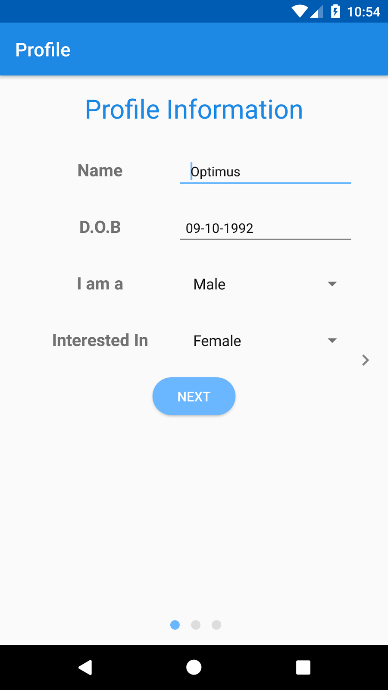
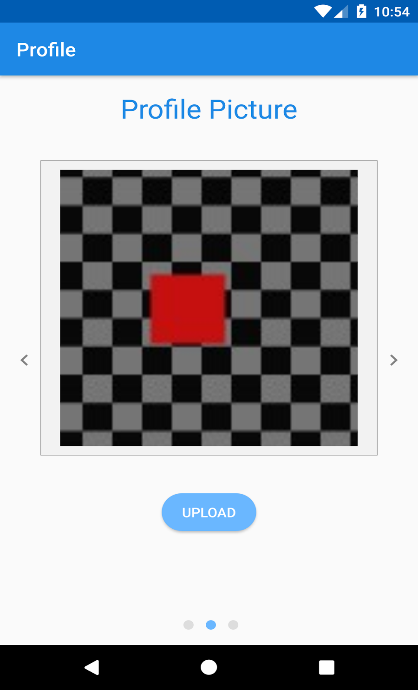


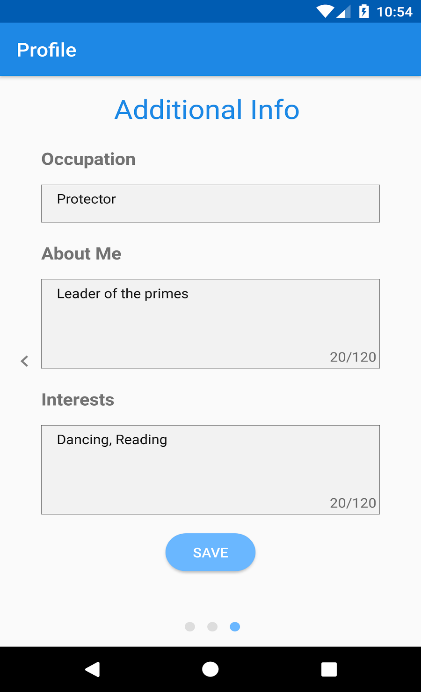
1. Profile Creation: This screen consists of 3 different sub-panes to declutter the interface.
   1. First Screen: Name, DOB, Gender and Interested In.
   2. Second Screen: Profile Picture Upload.

Profile Picture can be uploaded by taking a picture on the phone camera or using an existing picture in the phone.

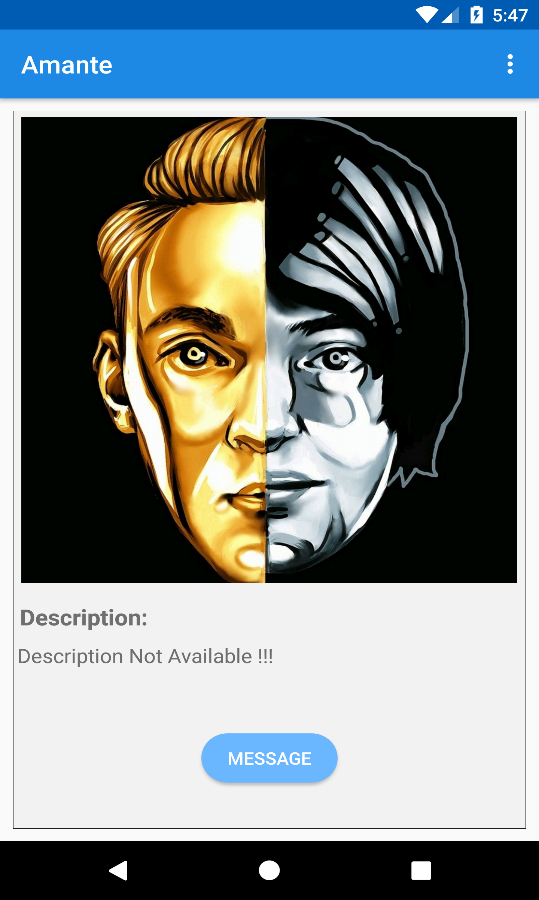
When the user is satisfied with the profile image, it can be uploaded or cancelled otherwise.

* 1. Third Screen: Occupation, About me, Topic of Interests.



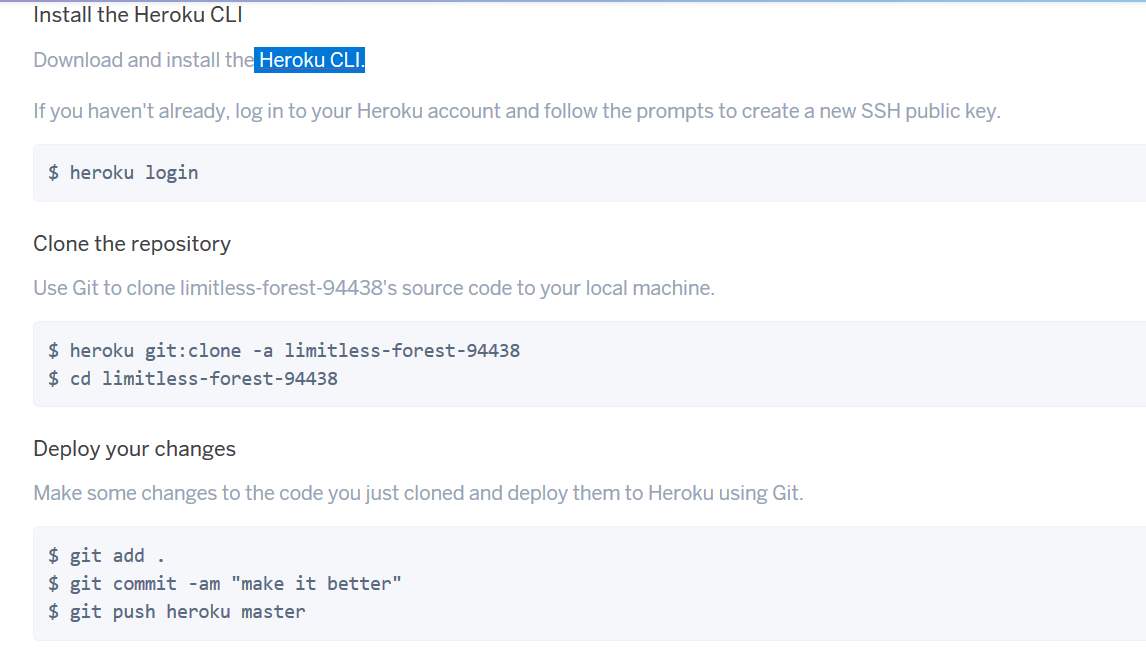
1. Matched Profiles Display: This screen consists of matched profiles with a button to message the matched user.



##### Backend

Get Matches Service: Webservice to find matches and send them to the app was developed on Node.js and deployed as a Heroku instance.

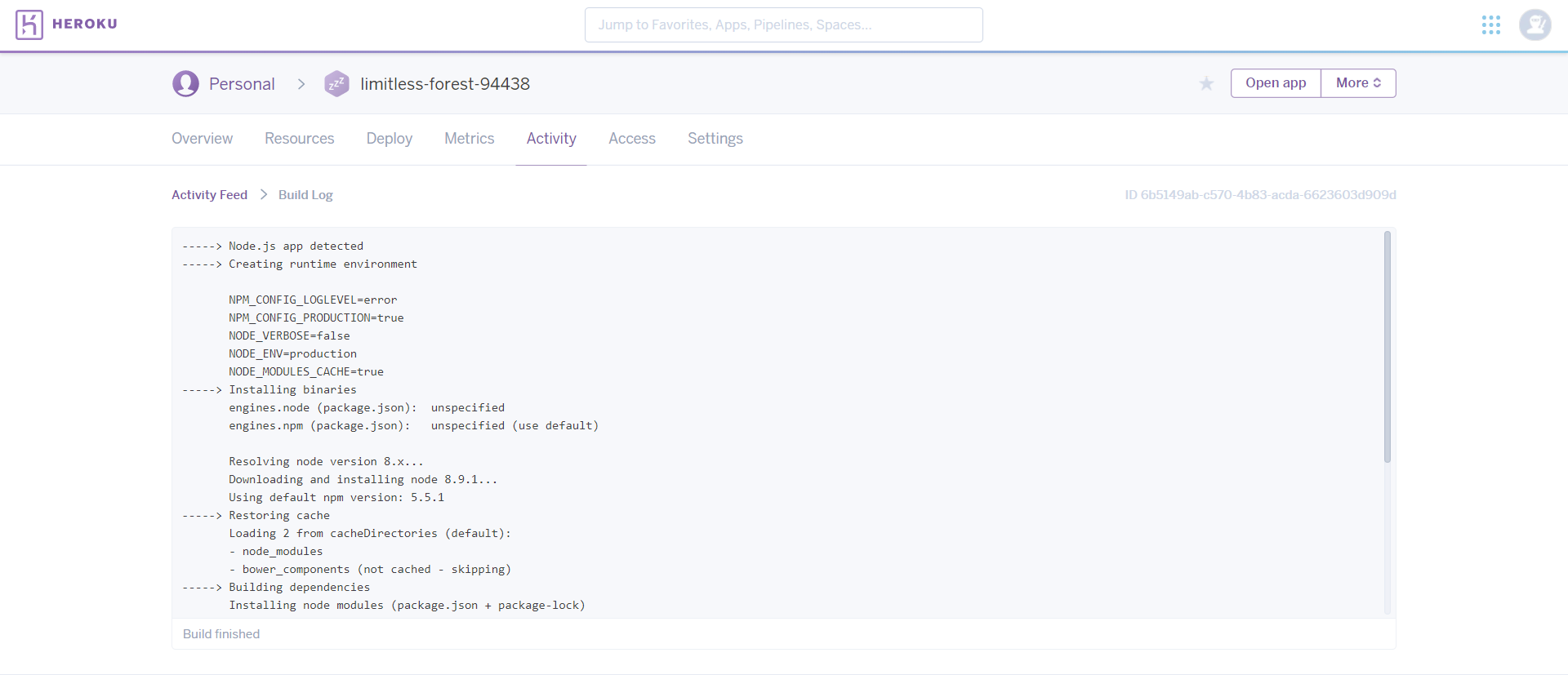
Node.js Application can be deployed onto the heroku server using the below commands:



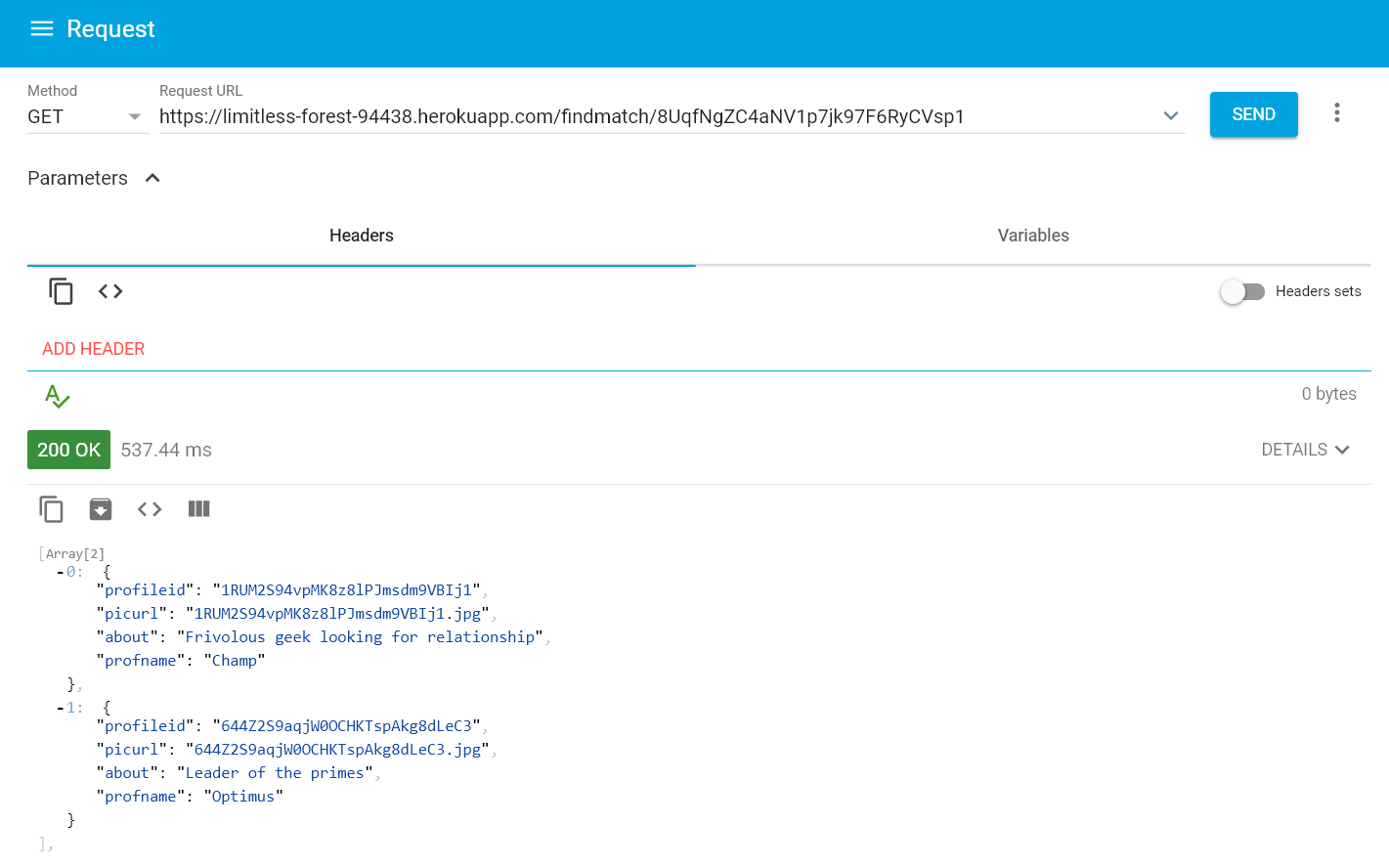
The git repository on the heroku is setup with CD/CI which will take care of building the deploying the server on to the heroku server.

The status of the deployment can be seen on the heroku dashboard.

Heroku Dashboard:



Sample Response from Webservice:



##### **Middleware**

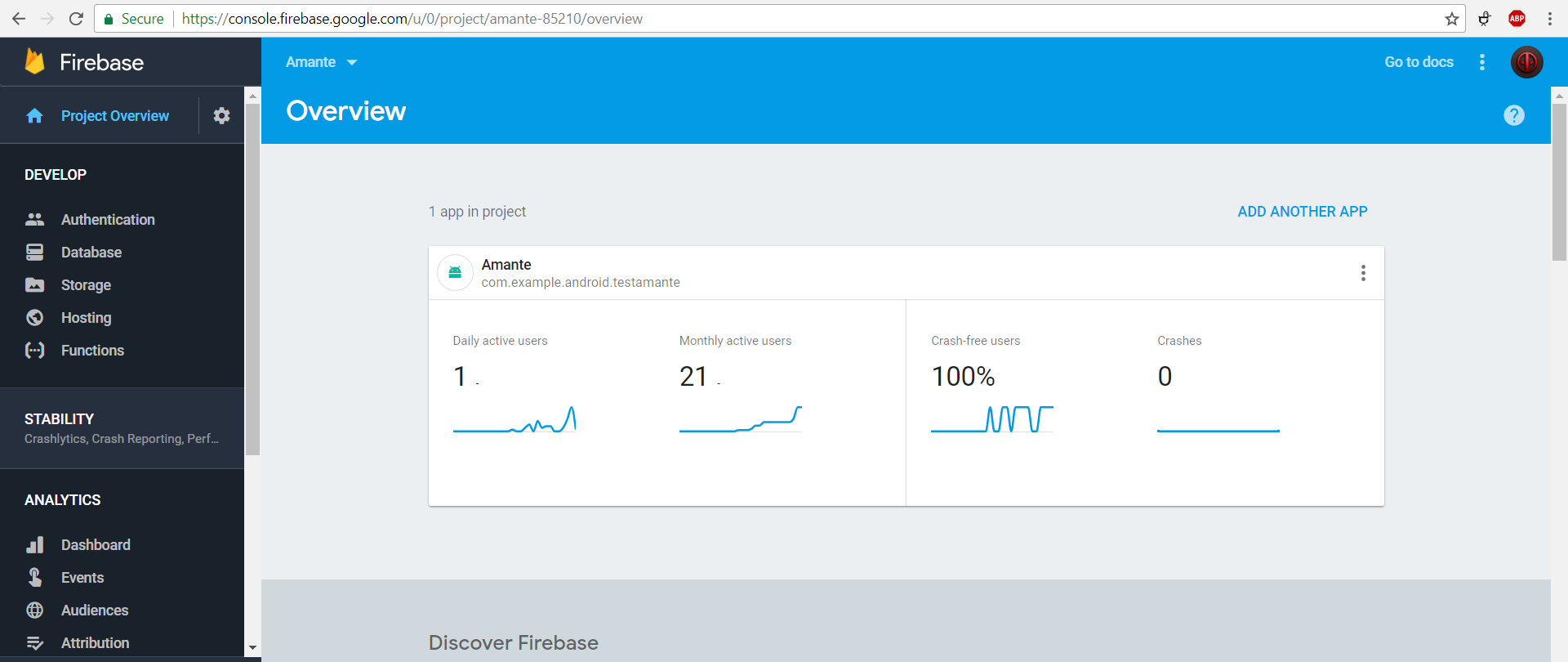
**Firebase Realtime Database:**

Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in real-time. Real-time syncing makes it easy for your users to access their data from any device: web or mobile, and it helps your users collaborate with one another.

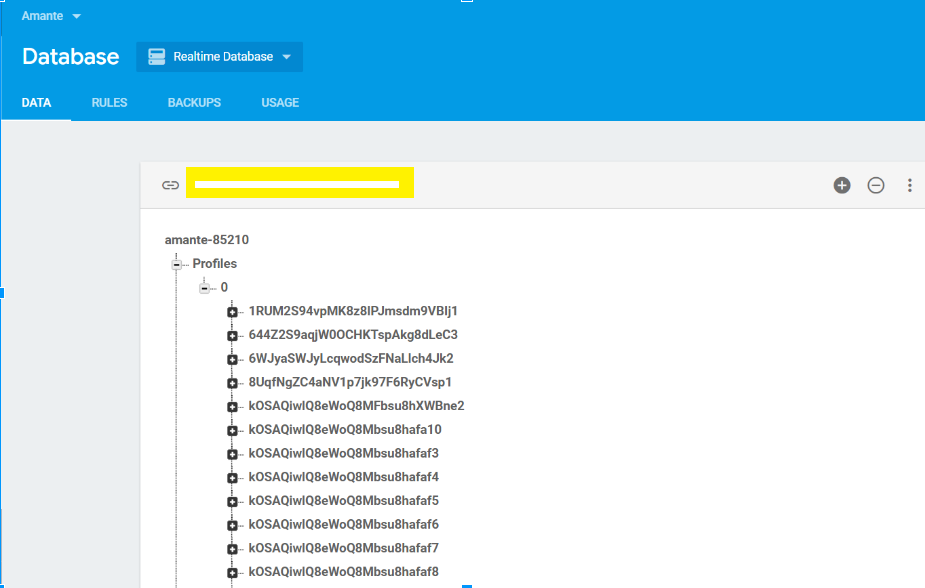
Real-time Database ships with mobile and web SDKs so you can build apps without the need of servers. You can also execute backend code that responds to events triggered by your database using cloud function for Firebase.

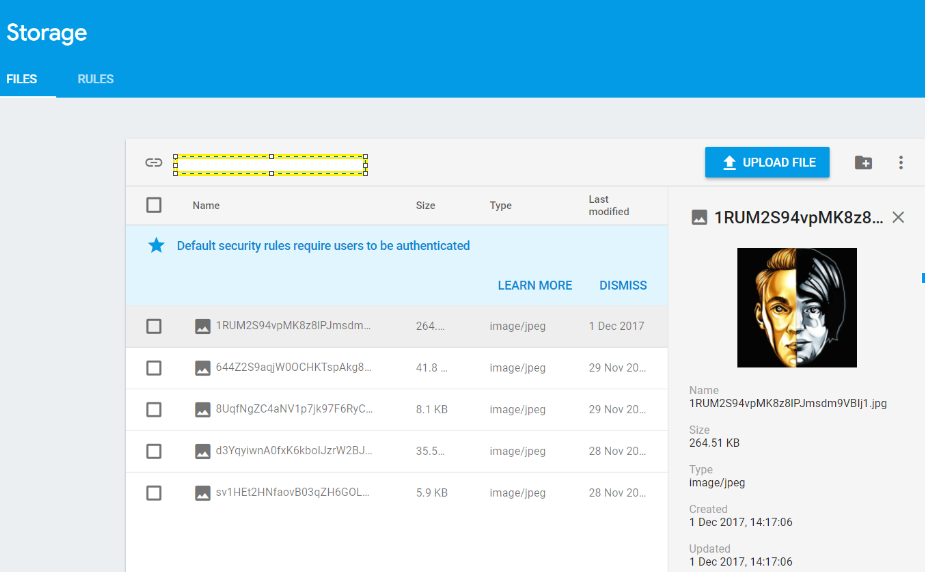
### Optimized for offline use, When your users go offline, the Realtime Database SDKs use local cache on the device to serve and store changes. When the device comes online, the local data is automatically synchronized. Finally Firebase helps you develop high-quality apps, grow your user base, and earn more money. Each feature works independently, and they work even better together.

Below is the screenshot of the Firebase admin console.



Below is the screenshot of the schema in use for Profiles.



Below is the screenshot of the storage on the Firebase server.

## 

## **Technical and Social Implications**

#### **Technical Implications**:

1. VR could be the next step for online dating: Dating app users talk to matches remotely with VR headsets and audio calls. “With VR people go on a first date where it looks like somewhat similar to what they look like in real life.

2. Implementing the best Match algorithm needs a deep dive into the existing algorithms and Working on advanced features like searching profile using images, we need to involve deep learning techniques.

#### **Social Implications:**

The impact of online dating on our daily behavior is pretty obvious: some lucky folks are finding the loves of their lives, sure, but many more are using it as an excuse to [behave creepily](http://www.refinery29.uk/2017/09/174467/tindstagramming-instagram-dms-not-matching-tinder-dating) towards others and [treat them like crap](http://www.refinery29.uk/2017/01/137937/what-is-breadcrumbing-dating-tactic). So, swings and roundabouts. But if we thought about the wider, big-picture impact of dating apps.

Well, a fascinating new piece of research highlights the influence of online dating on marriage rates and even the stability of marriage itself. By completely overhauling the way in which many of us meet our partners, online dating has made interracial marriage more commonplace. And not only that, but married couples who meet online tend to be stronger than those who meet in "traditional" ways.

The vast majority of people would meet their partners through loose social connections – people linked to their friendship group, through mutual friends, at church, through their families, etc. But this all changed with the advent of online dating.

## **Challenges encountered**

During the course of development, we faced several challenges - some of which are listed below.

* **Learning Curve:**

With the Android SDK going through several iterations and each one significantly improving over the previous APIs, implementing basic features has become incredibly complex and this has slightly hampered our ability to rapidly develop and test features and screens for our app.

* **Limitations with the basic version of Firebase:**

The free version of firebase which we have used in our project is a bit slow when compared to their other plans and the login and authentication times have taken a significant blow. Another problem encountered was in the Firebase Database where only one instance is provided and there have been challenges in storing chats sent to different matched profiles.

* **Maintaining visual consistency:**

Developing an app with consistent visual elements is incredibly hard, more so in a collaborative environment with 4 team members. Fortunately, establishing standards like which kind of buttons, colors, coding standards, dimensions to use helped alleviate this challenge to a certain extent.

* **Problem in User Interface:**

Android does not follow a common UI process during their application development. Google till date has not introduced a common User Interface rule that needs to be followed. This in turn has led to certain applications being incompatible with certain devices. So android developers and designers should make sure they must build applications with responsive layout

* **Software Fragmentation Issue:**

Several versions of Android operating system are available on different devices. Timely upgrades change the version of Android OS that runs on the device. This implies that the developers can’t just focus on the most recent versions of the OS as not everyone upgrades their devices. It is inconvenient for many users to upgrade their operating systems because manufacturers play a vital role in device configuration.

* **Hardware Fragmentation**:

This becomes a big Android app development challenge since there are nearly 170+ devices running the OS. Each device has different features with respect to keyboard forms, screen size, camera buttons, etc., making it a development nightmare.

* **Market Research Costs:**

One of the biggest Android app development challenges for developers is the cost behind market research. Understanding the end user is key to Android app development, but can require a lot of research, making it costly for developers.

* **Usage of Third Party API:**

Certain third party API is device dependent and this restricts the usage of many apps across different devices. The SDK and cross-platform IDE does not allow android developers to use a single API across the range of versions.

* **Security Issue:**

Recently Android mobile devices are getting affected by malware that is similar to PCs as a result of extensive internet usage. Lack of stringent rules in the area of software fragmentation in Android has led to proliferation of malware threats. To make things worse these security issues are quite difficult to fix, and it is often one of the biggest challenges faced by android developers.

## **Project timeline and team member contributions**

We are a team of 4 members and will split our work into 3 groups, front end development, Middleware and backend development group.

#### **Front End Team:**

● Shekhar Singh

● Suresh Kumar Tathari

The Front End team worked on tasks such as building the visual elements of the app such as screens, images, logos, buttons and tying them all together and developing a logical workflow from one screen to another.

#### **Back End Team:**

● Tasmin Chowdhury

● Vinay Kumar Calastry

Backend team implemented a lightweight Node.js app and exposed it as a web service for Matching various profiles and returning back matched profiles to the user. The matching algorithm profile details from the Firebase Realtime database and using those details, a list of matches is generated. This was deployed as a service on Heroku which automated most of the manual tasks involved with deployment and this resulted in much faster development and deployment cycles.

#### **MiddleWare Team:**

● Suresh Kumar Tathari

● Vinay Kumar Calastry

Middleware team was involved with all aspects of development related to Firebase. Many important functions of a standard app were implemented using the Firebase API.

* Authentication: Firebase Authentication service was leveraged to create a user directly on the platform and also use an existing Google account to login.
* Database: Firebase provides a nifty database to store information which can be added, updated or deleted in real time. The middleware team wrote the code to save or update a user’s profile in the database.
* Storage: Firebase also provides a storage facility for files and in our case, profile pictures. The tight integration between various features in the Firebase API ensured that we will be maintaining a strict correlation between the user’s profile picture in the storage and the profile details in the database and also prevented unauthorized access due to the fact that the Authentication service is tied to the database and storage. The team was involved in developing code to push an image to the Firebase storage which was either taken using the phone’s camera or was a picture selected from the phone’s memory.

## **Future work**

1. A complicated full-fledged Messaging feature implemented into the Dating app with specific features such as using dating related emojis and chat styles.
2. A visual overhaul of the app when adding more features which are easy to use and also prevent cluttering.
3. Working on advanced features like searching profile using images will be taken up later as it is an advanced future feature.
4. Introduce the message suggestions in the chat box using machine learning.
5. Better algorithm to detect fake profiles and introduce the second layer of security on top of the android environment in the application layer by end to end encryption.
6. Giving the users to handle what all information He or She can share on their profile for public access.
7. A future scope of this project will involve deep learning techniques to provide much better matches.
8. Other nice to have features would be to implement automatic Language translation and Video messaging.

## **Summary of research**

Our work on this project has widened our scope and perspective of the Human Computer Interaction. With the evolution of computers, HCI revolutionized the applications and now in this digital world it is playing a vital role and has been the spot on funda for making the applications ease to use.

Mobile applications are progressively more complex as our industry’s technologies and methodologies advance and things that used to be a one-way static medium has now evolved into a very rich and interactive experience.

We attempted to learn HCI principles utilizing the guidelines published in android mobile application, and has got to know the basic instincts from the scale of usability, understandability and utility. We tried to emphasize on the user-centric-design, making the user interface as simple and plain so that the user can do the tasks intuitively just by seeing the application screens.

Android’s user interface is based on direct manipulation, using touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching and reverse pinching to manipulate on-screen objects. The response to user input is designed to be immediate and provides a fluid touch interface.

**Working on this project exposed us to the following aspects of HCI:**

* + - 1. Our first and foremost accomplishment from the design perspective will be the implementation of user interfaces by optimizing a design for a desired property such as learnability or efficiency of use.
      2. The methods for implementing interfaces by means of software libraries some of which are part of the existing android environment and other third party libraries has given an extra learning curve posing some challenging in accomplishing the component design and implementation.
      3. Factors for evaluating and comparing interfaces with respect to their usability and other desirable properties by coming up with a novice design and functionality.
      4. Best practices for studying human computer use (mobile) and its sociocultural implications on a broader scale.
      5. How to choose on the best technologies in terms of ease of development, maintenance and advancements.

1. **References:**

* Android material design guidelines

<https://material.io/guidelines/usability/accessibility.html#accessibility-principles>

* Firebase guide, API for Android:

<https://firebase.google.com/docs/reference/android/packages>

* [The Split-Brain Human Computer User Interface](http://www.mitpressjournals.org/doi/abs/10.1162/002409402760105352) – Gregory P. Garvey

Leonardo, Vol. 35, No. 3: (35) 319-325

* [UI Design Principles for Interaction Design](http://www.mitpressjournals.org/doi/abs/10.1162/desi.2008.24.3.85) – Adream Blair-Early, Mike Zendor

Design Issues, Vol. 24, No. 3: 85-107

### 