

FINAL YEAR PROJECT



trickll

Uzair Majid (u.majid@khi.iba.edu.pk)
Syed Moaz Hasan (s.hasan@khi.iba.edu.pk)
Syed Muhammad Talha (s.talha@khi.iba.edu.pk)

Introduction:

A two-way communication mobile app that lets famous personalities/celebrities inspire their followers/fans by directly interacting with them. The main features of this app includes the following things:

1. A live question answer session.

How it works?

Celebs will start a question&answer session for a certain amount of time. The participation of fans will depend on the following

- Fans will be given some prior time to register for that session.
- Once registered, fans will be picked randomly to interact with the celebrity.

The session will end at the prescribed time by the celebrity.

App Platform: IOS

This app will be designed with the focus on both technical (backend) and visible (frontend) aspects. The technical aspects will allow the features to be implemented in a correct manner with the primary focus on ensuring an effective communication. The visible aspects will allow the maintenance of a consistent and simple design which will make it easily understandable and usable.

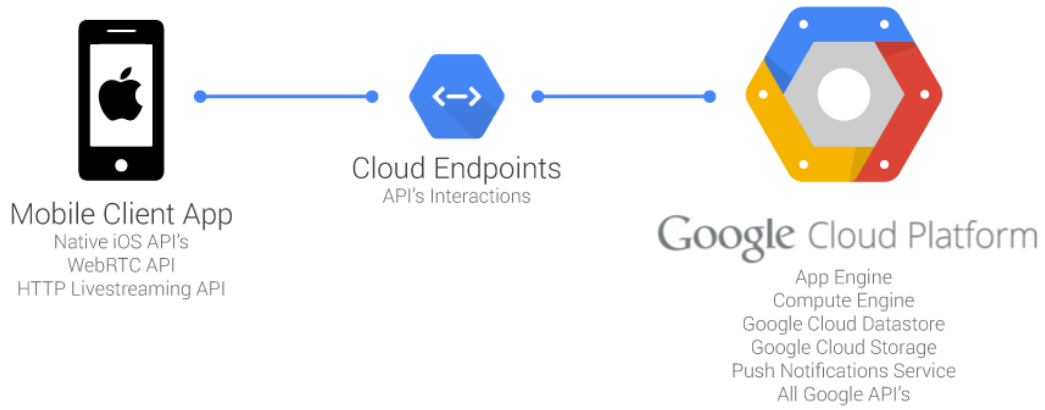
Watch your favorite celebrities
and get a chance to talk to them
LIVE.



High Level Design:


1. System Level Overview

System Design



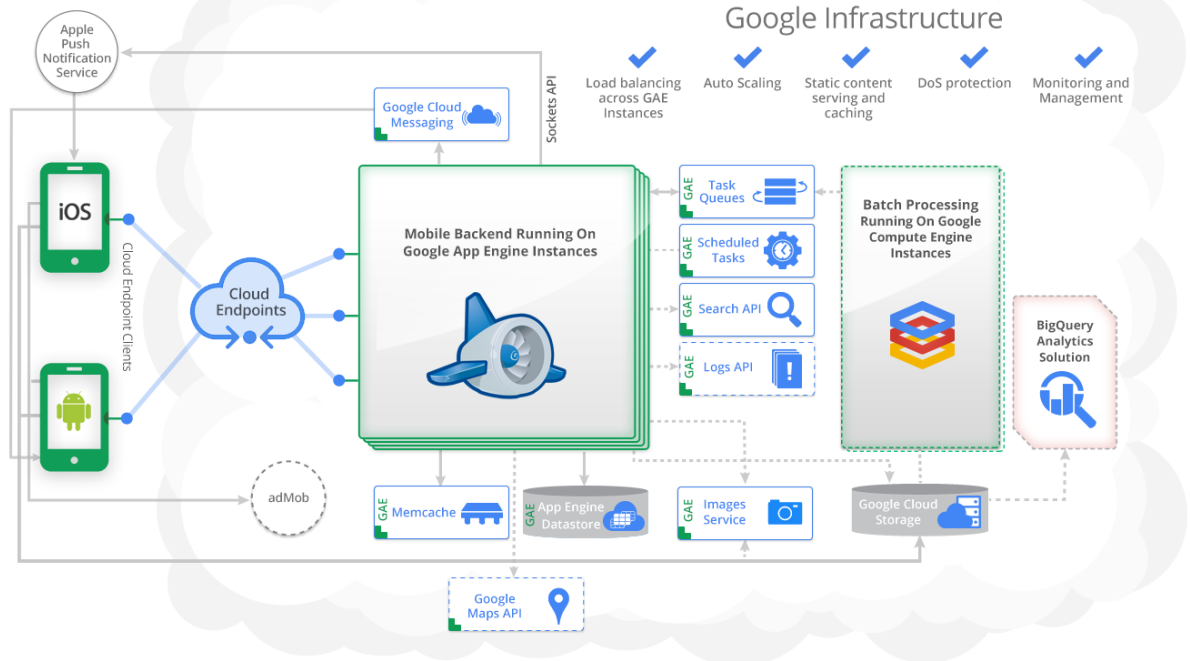
trickll

Mobile Solutions on the Google Cloud Platform

 Your Application Code running on Google App Engine (GAE), Google Compute Engine (GCE), and Client Devices

 Google Cloud Platform Services  Optional components

 Capabilities Included



Software / Hardware Design:

1. Overview

- Backend: Google App Engine
- Communication: WebRTC Platform
- Frontend: Native iOS App

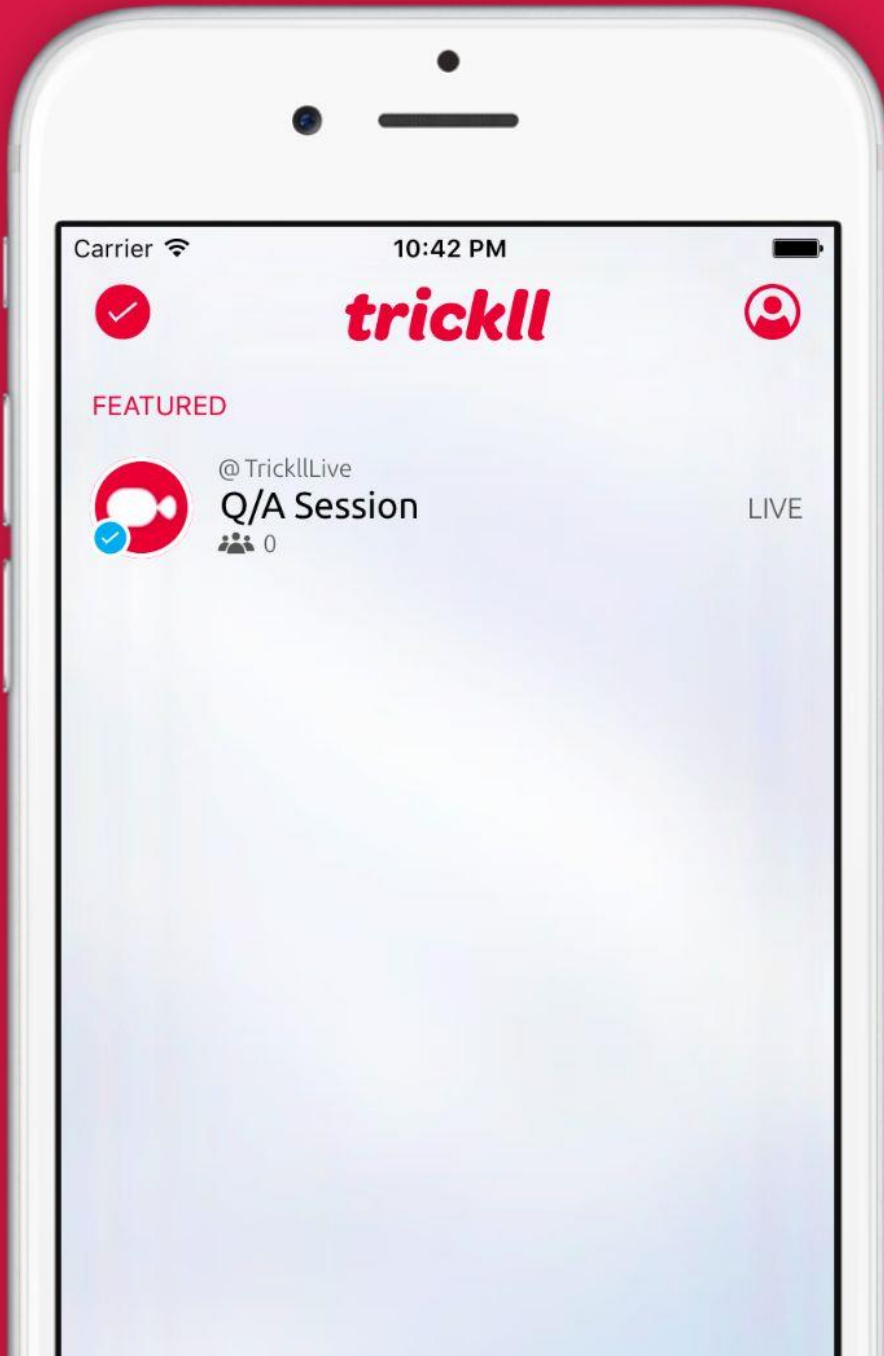
2. Program Details

a. Modules

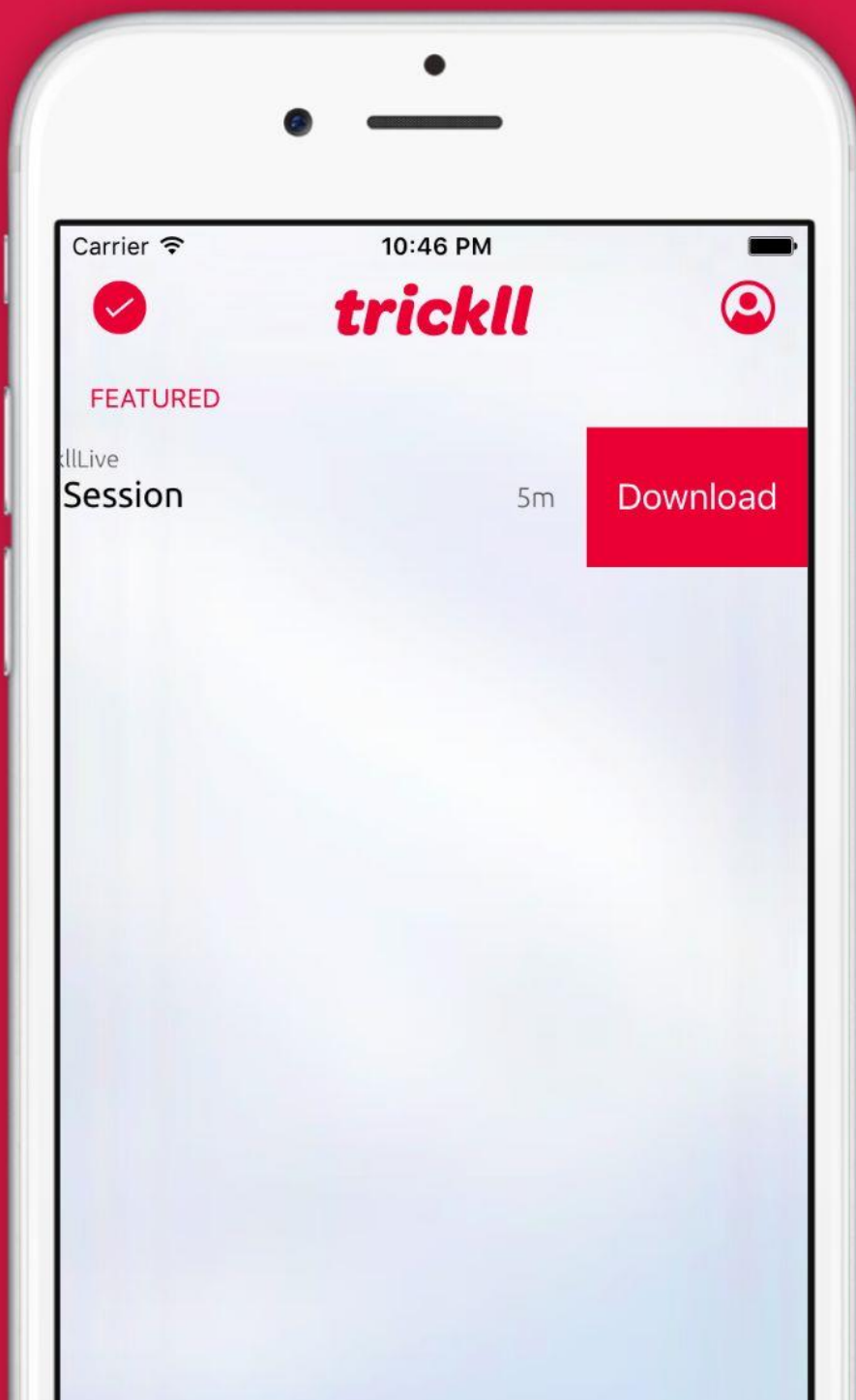
- Cloud Endpoints API (GAE) using Java
- NoSQL Datastore (GAE)
- Push Notifications using GAE Sockets with APNS
- Real-time video communication using WebRTC Platform
- Native iOS App Client using Objective C

b. User interface

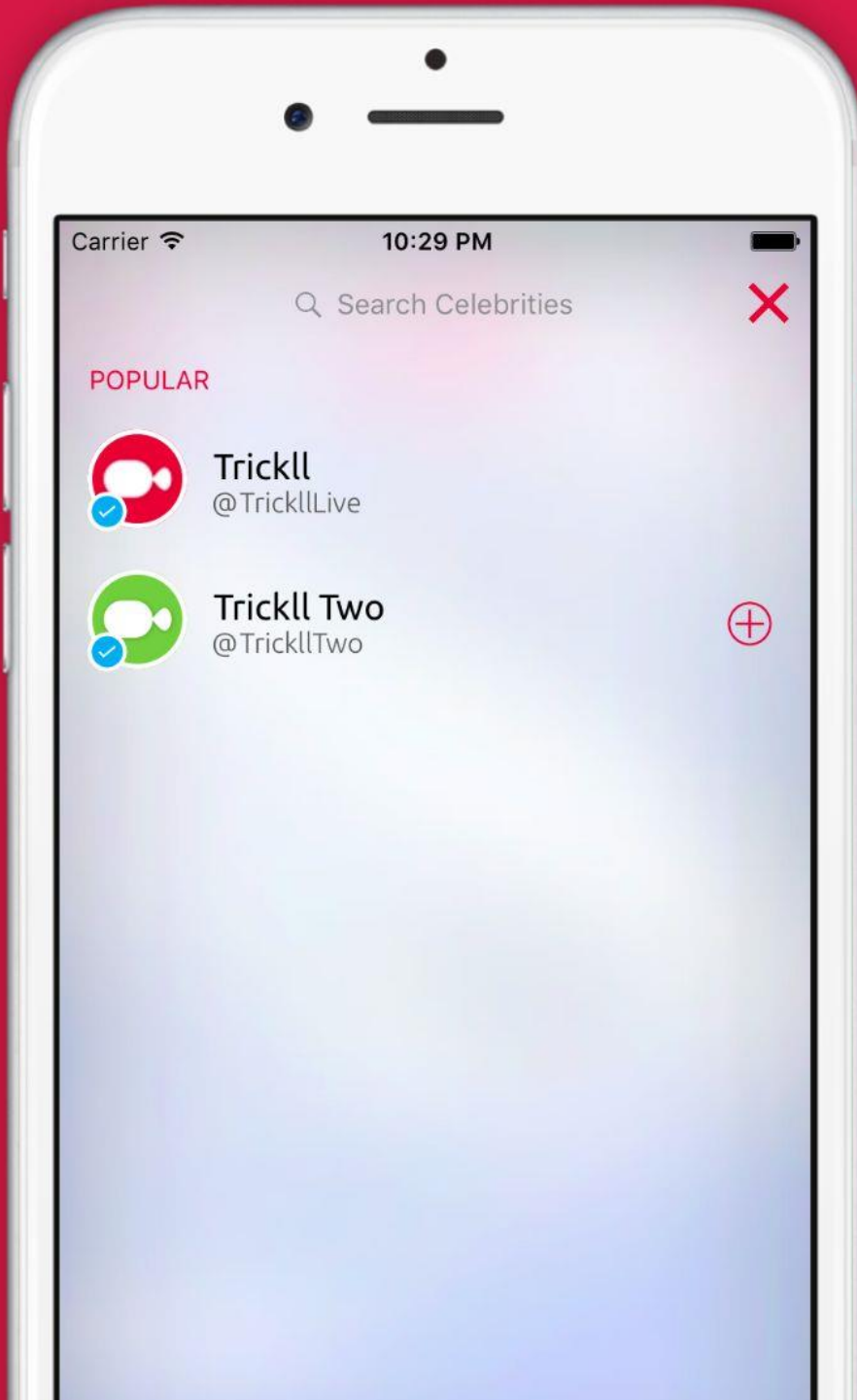
Watch your favorite celebrities
and get a chance to talk to them
LIVE.



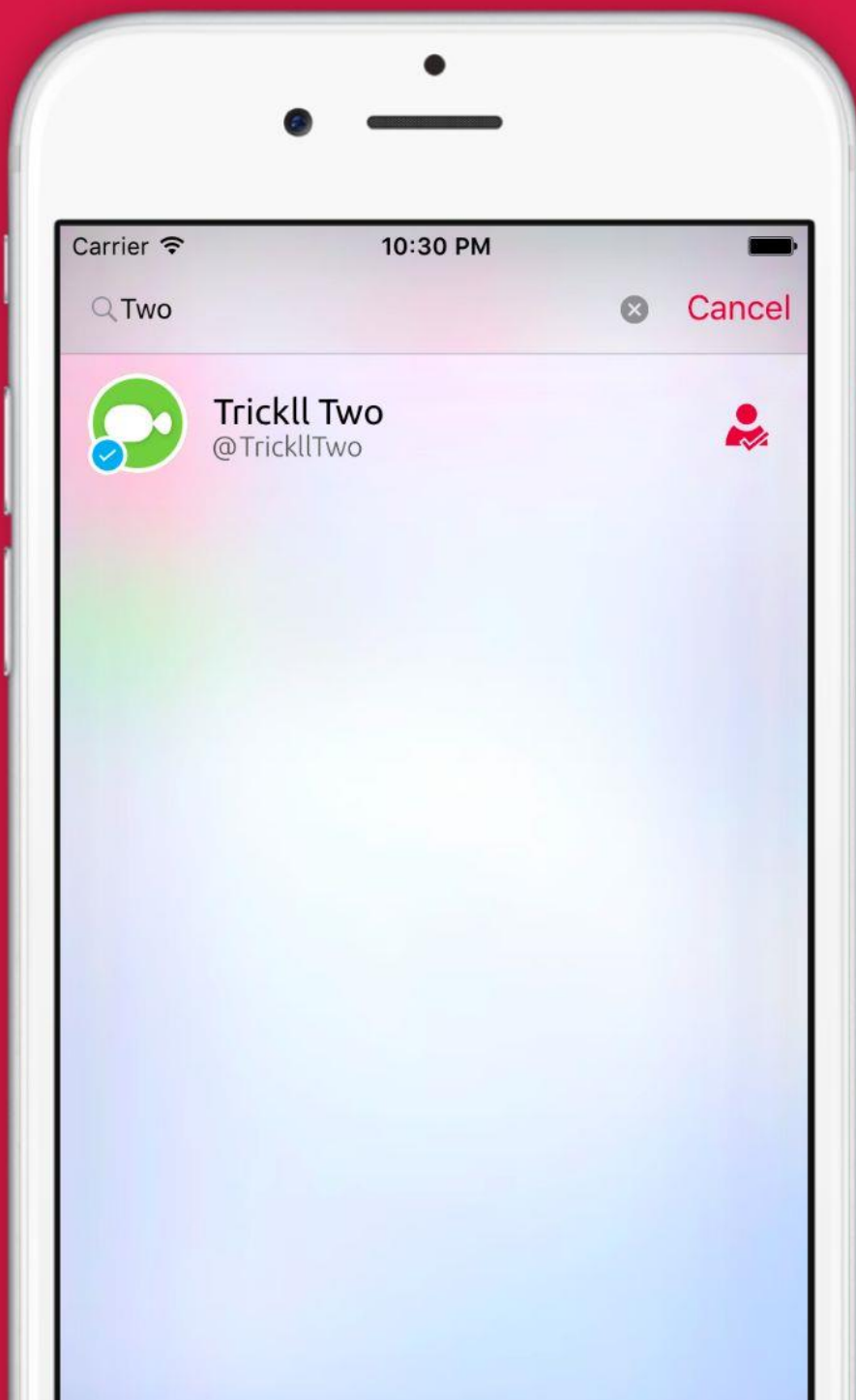
Playback or download recent sessions.



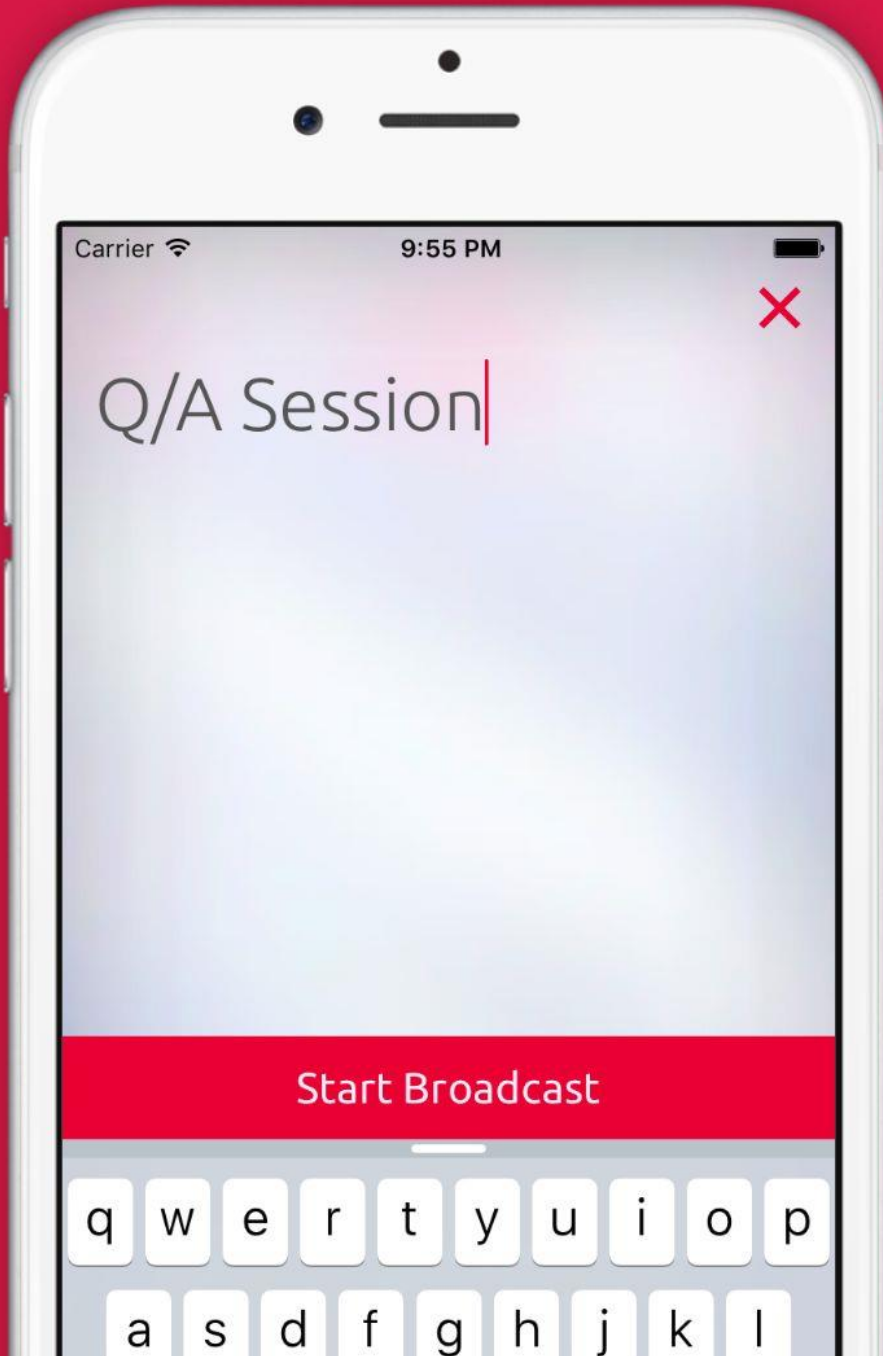
Discover popular and search for
your favorite celebrities.



Follow celebrities to get notified when they are LIVE.



Celebrities can broadcast LIVE
and pick viewers to join in the
session.



c. Tests

- i. Unit Testing
- ii. System Testing

Results:

We have successfully achieved:

- Real-time one-to-many & two-to-many live video broadcasting
- Archiving of the broadcast sessions
- Simple and easy-to-use interface

Conclusions:

The acceptance test our app relates to the feedback received from both the fans and celebrities. Feedback in this case was taken via different surveys and meetups with celebrities. We have technically tested our app for almost all the real world scenarios. The app conforms to all the guidelines from the platform vendors. We expect a great response from the users once our app is in the AppStore.

Appendix:

Appendix 4: Software/parts list

- Cloud Endpoints API (GAE) using Java
- NoSQL Datastore (GAE)
- Push Notifications using GAE Sockets with APNS
- Real-time video communication using WebRTC Platform
- Native iOS App Client using Objective C

Appendix 5: Work distribution

- **Talha**
 - System Architecture
 - Cloud Endpoints API Design & Development (GAE)
 - Datastore Design (GAE)
 - iOS Client App Design & Development
- **Uzair**
 - WebRTC Platform Client Implementation
 - Push Notifications Implementation
 - Twitter API Implementation
 - Meetings with Celebrities
- **Moaz**
 - Datastore Implementation (GAE)
 - Code Review & Cleanup
 - System Testing

Appendix 6: Project timeline

Milestone	Duration
Idea refinement & user feedback	1 Weeks
Research on industry standard tools	1 Weeks
Scalable System Architecture Design	1 Month
Wireframes & Flow Diagrams	1 Week
Datastore Design	2 Weeks
Cloud Endpoints API Design	2 Weeks
Datastore Implementation	2 Weeks
Cloud Endpoints API Implementation	2 Months
iOS App Basic User Interface	1 Week
iOS Client App Logic	3 Months
Twitter API Integration	1 Week
Backend and Client integration	1 Week
WebRTC Client Implementation	2 Months
Push Notifications Implementation	2 Weeks
System Testing and Debugging	1 Month
Code Review & cleanup	1 Weeks
Refined User Interface	1 Weeks
Total Project Duration	12 Months

Acknowledgements:

We would like to thank our project supervisor Dr. Quratullain Rajput for her continuous support at every design and development step of our project.