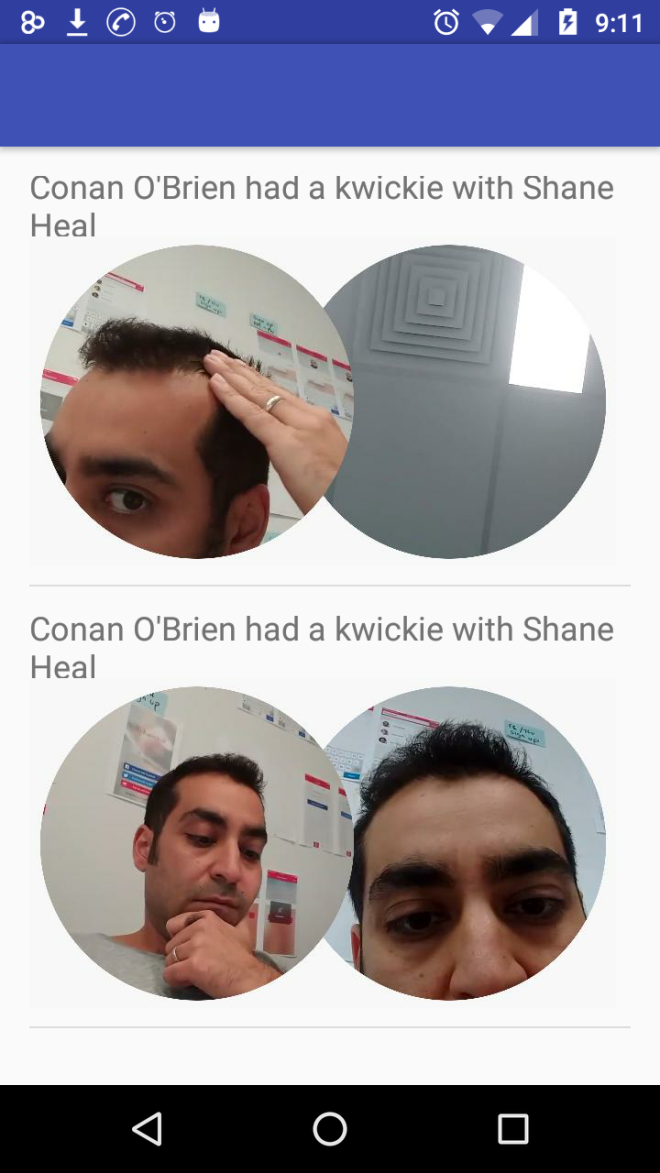
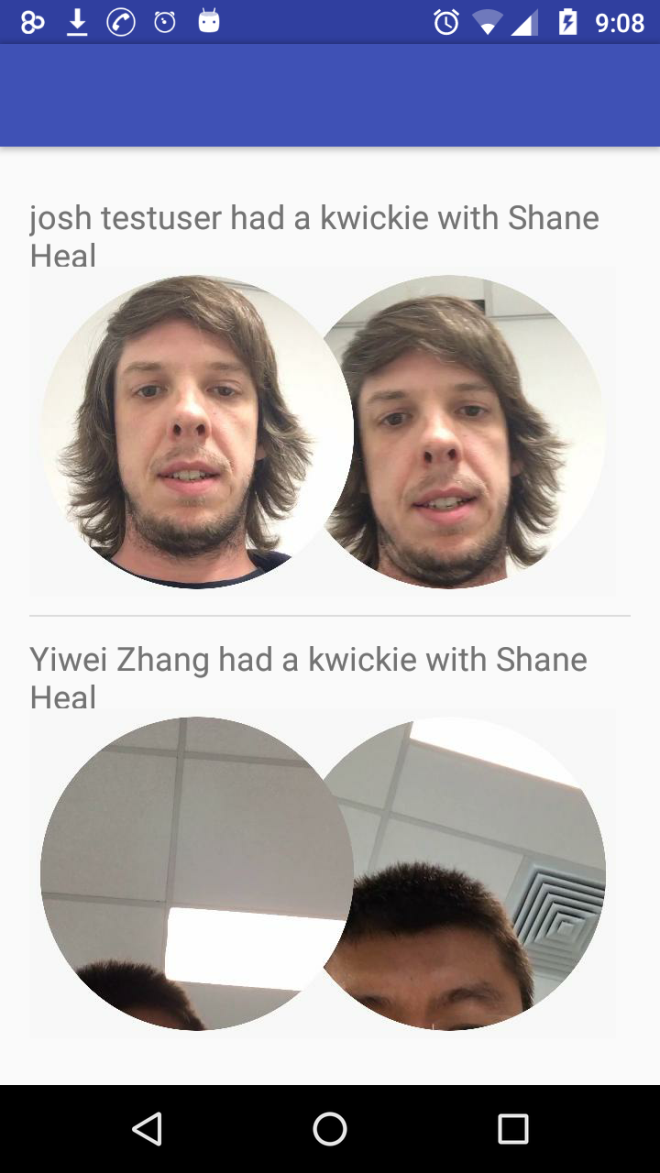
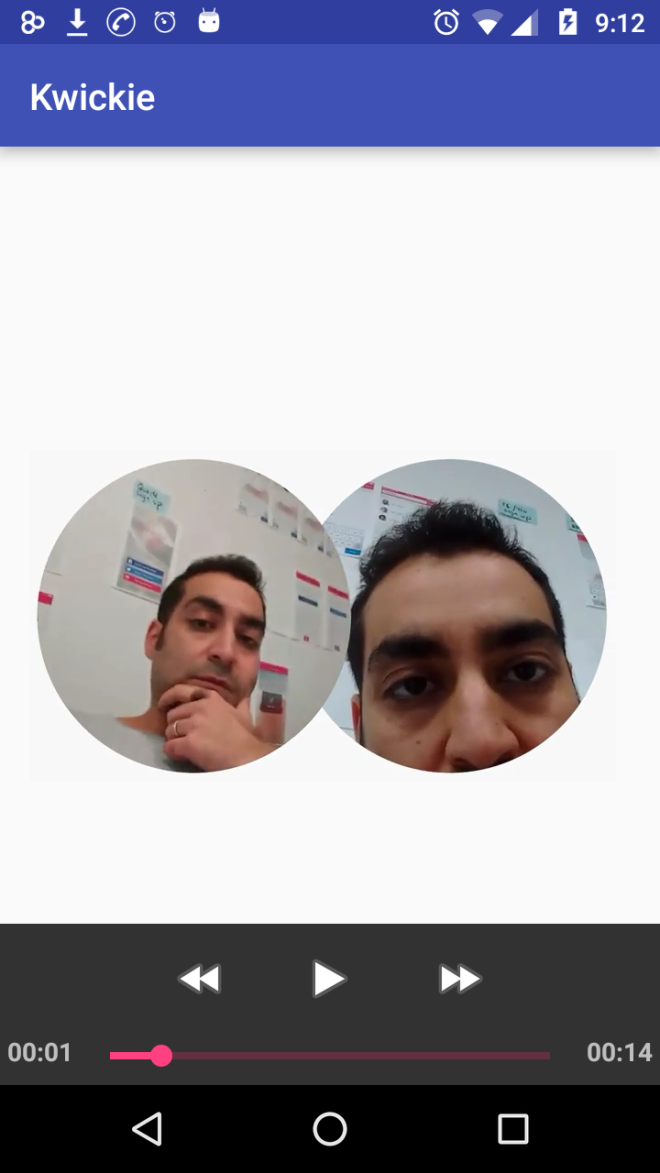
**Task 1: Kwickies pulled using server APIs**

****

**App shows a scrollable list of all retrieved Kwickies.**

**Task 2: Playing one of the kwickie videos. App plays the video using Android components VideoView and MediaController.**

****

**User can tap on one of the kwickies to play it.**

**Task 3: Like functionality**

**API:** <https://bigdev.kwickie.com/api/kwickies/kwickieVideoId/:XXX/users/:YYY/like>

**Method: Post**

**Server data:**

Json array of Kwickie objects

[

{

….,

"kwickieVideoId": 12345,

"likesCount": 6,

….

},

{

….,

"kwickieVideoId": 23456,

"likesCount": 7,

….

},

{

….,

"kwickieVideoId": 34567,

"likesCount": 0,

….

}

]

Json Array of Registered users

[

{

"id": 1,

"myLikedVideos":

[

{

“kwickieVideoId” : 12345

},

{

“kwickieVideoId” : 23456

}

]

},

{

"id": 2,

"myLikedVideos":

[

{

“kwickieVideoId” : 12345

}

]

}

]

myLikedVideos contains all the videos liked by a registered user.

**Sequence Diagram**

Kwickie App Kwickie backend server

**Request**: Increment/Decrement likesCount

API endpoint: @serverurl/videos/:videoid/users/:userid/like

Method: HTTP Post

**Response**: Returns modified likesCount

Server modifies likesCount

and myLikedVideos object as a result of Post call

POST handler will determine whether to like or unlike this video.

If this kwickieVideo has been not found in myLikedVideos for this user, it will be liked. The kwickieVideoId will be added to myLikedVideos and likesCount for Kwickie will be incremented.

If the kwickieVideo was found in myLikedVideos, it will be disliked. The kwickieVideoId will be removed from myLikedVideos and likesCount for Kwickie will be decremented.

App will update likesCount in the UI with the response received from Post call.

**Task 4: Recommendation engine to send only appropriate videos**

Properties to consider:

* User profile
  + Gender
  + Age
    - Let users tag the videos during upload. Videos tagged with age-restricted should not be sent to minors.
  + Language
    - If the sending and receiving users do not speak the same language, videos may not be useful for receiving user and hence spoken language can be used as filtration mechanism.
* Video attributes
  + Content
  + Length
    - Kwickies provide a quick video chat option. Very long videos can be spam and hence can be filtered out.

Video filtering mechanisms:

* Content filtering:
  + Speech recognition
    - Recognise the spoken words in a video. Videos containing foul words should be marked as in-appropriate and shouldn’t be sent further. Build/grow a DB of foul words over time.
  + Face detection
    - Detect faces using face recognition. Block a video if user asking the question is found in the blocked list. Build/grow the block list DB over time.
  + Skin Detection
    - Adult content can be filtered out based on the degree of exposed skin detected in frames.
* Collaborative filtering:
  + Enable users to set their video preferences during app sign-up. During video upload, capture meta-data and compare meta-data against the receiving user’s preferences or with his/her reported videos history to eliminate non-appropriate videos.
  + Enable community ratings so that users can rate other users on the platform. Block users that go below a threshold score/rating.