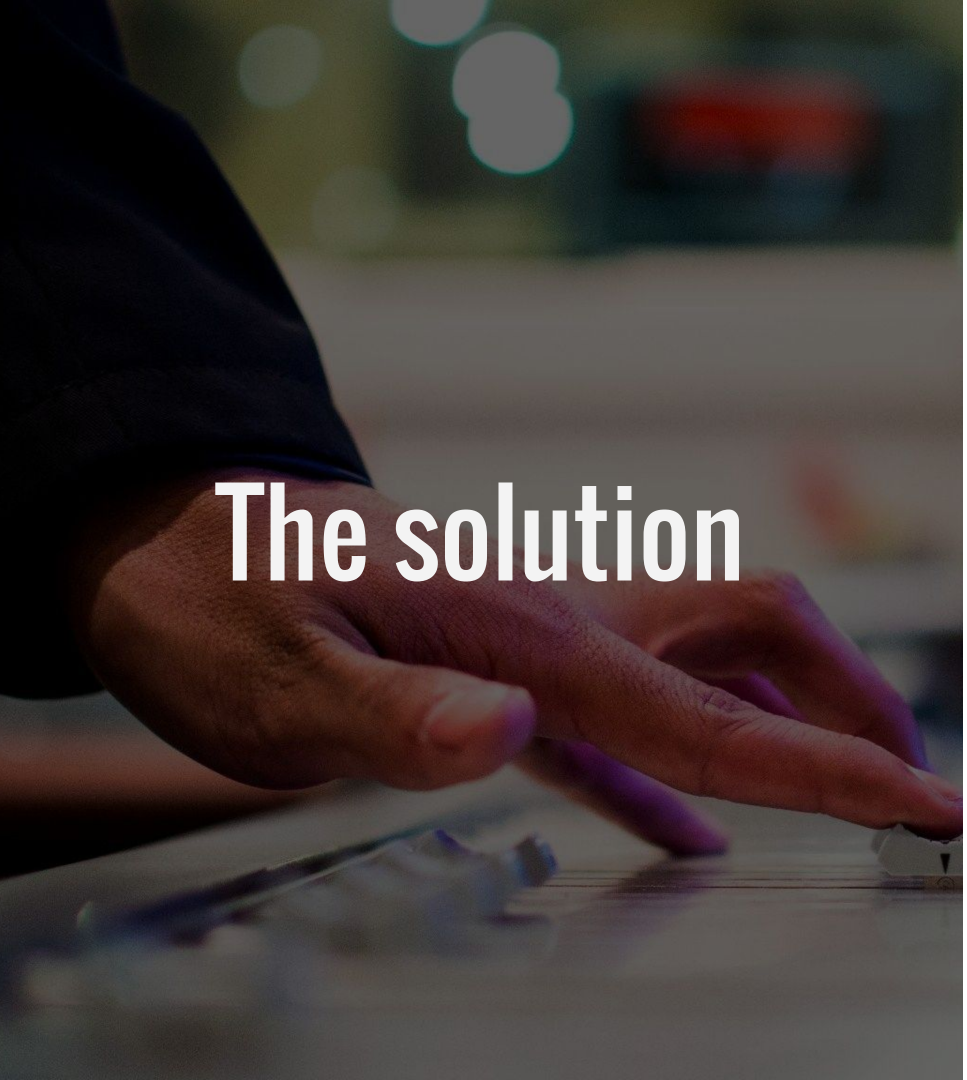
TuneIn

•••

CS263 Final Project Demo
Nimisha Srinivasa
UCSB Winter 2016

Motivation: Create and share audio for free!

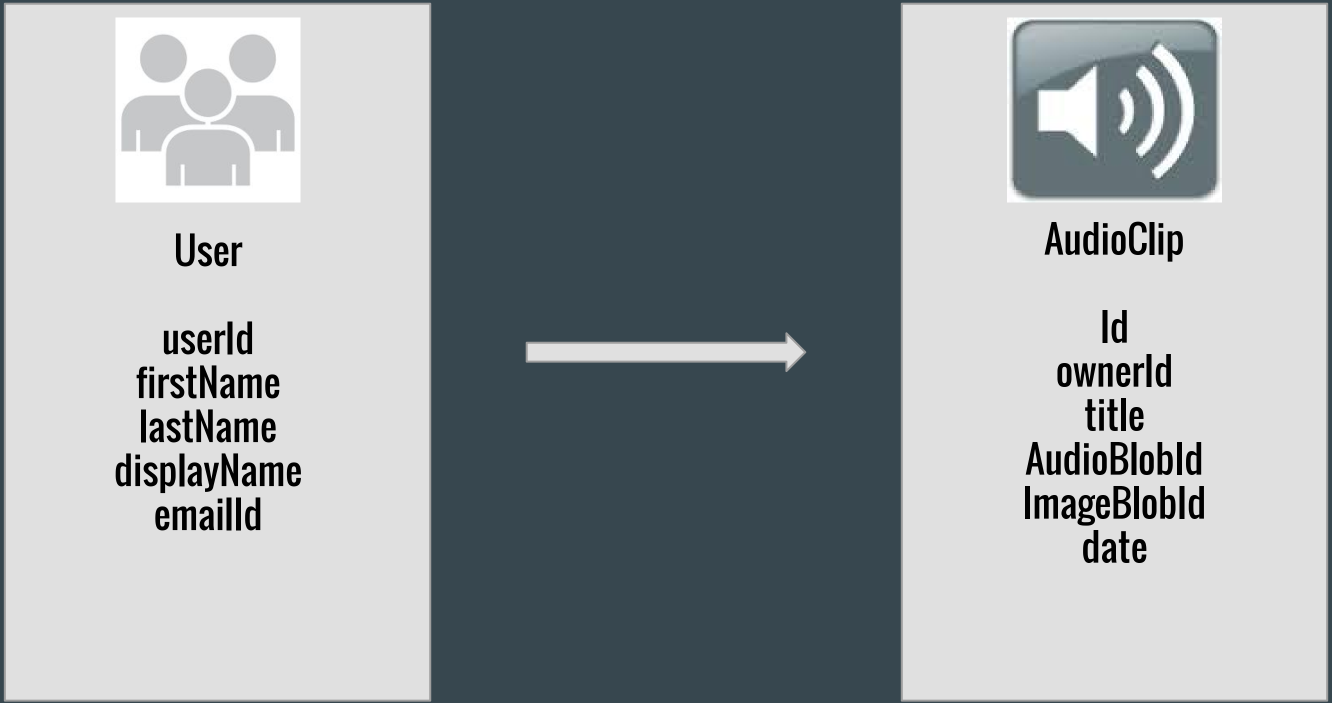


The solution

- A Google App Engine (GAE) Application
- Backend:
  - JAX-RS
  - API’s used:
    - Datastore
    - Memcache
    - Taskqueue
    - Blobstore
- Front-end:
  - HTML5 and Javascript frameworks
    - Hello.js
    - Bootstrap
    - Jquery plugins

DEMO

Model (DataStore)



User

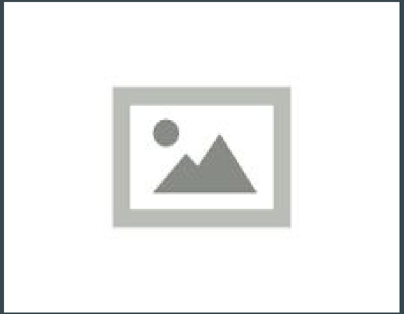
userId
firstName
lastName
displayName
emailId

AudioClip

Id
ownerId
title
AudioBlobId
ImageBlobId
date

BlobStore

- Upload audio clips (.mp3 support)
- Upload images for audioClips



Memcache

- In-memory **distributed** data cache
- Used to reduce the query load on server
- Data is **not** presistant
    - For caching the work of other users

Task Queues

- Perform work in the backend, where no user interaction is required.
- **Push Queue** to push pending tasks to the queue.
  - Accessible only within the app engine environment
- Usage:
  - Deleting the audio and image blobs in the background when an audio clip is deleted by user

Learnings & Experience

- Design of REST-ful API’s
- Good understanding of REST-ful services using Jersey.
- Blobstore is a little hard to deal with (deprecated).
- Little documentation available online for Google App Engine Development using Java when compared to the python support.

Novelty

- A dedicated web service for music lovers.
- Finding with like-minded people.

Future Work

- Integrating with facebook
- Friendship between Users.
- Uploading to Google Drive
- Likes and Shares

Performance Evaluation

- **Server-side Caching** using memcache. The Response Time decreased from 1.7s to 150ms
- Doing work in the background using Task Queues reduces **Response Time**.

Thank You