"MixList" - DJ Mix Sharing & Listening Application

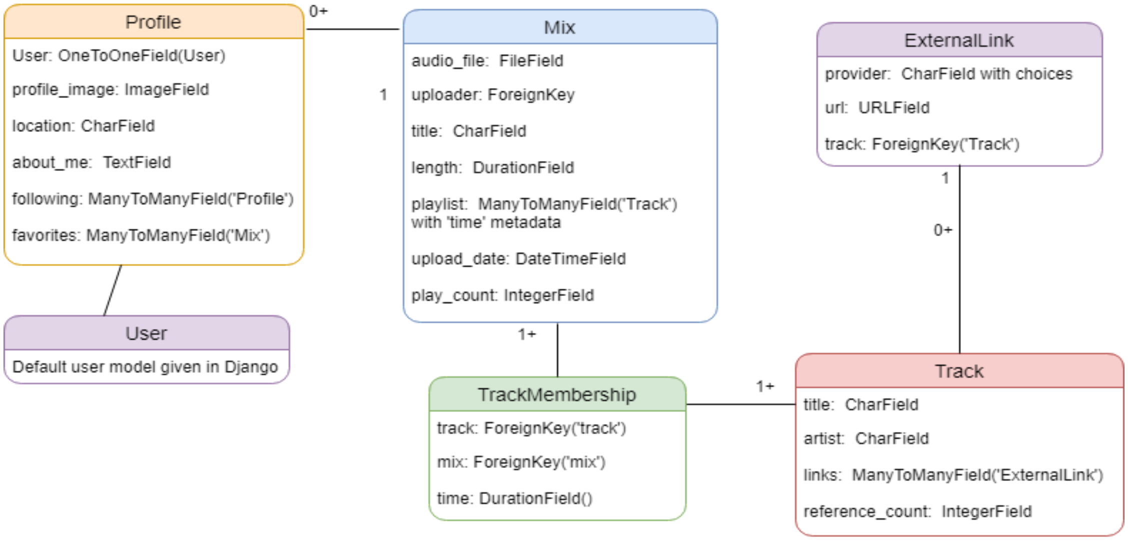
The Krusty Krab - Github Repository

Team Member Name	**Github Username**	**Team Member Name**	**Github Username**
Benjamin Murphy	*benmurphy7*	Matthew Robinson	*mattrobinson125*
Daniel Maryanski	*danielmaryanski*	Shipeng Yu	*GK67*
Daniel Szymanski	*DSSzymanski*	Yuzhuo Shi	*SuedeOO*

Overview / Changes:

Our project is a website that allows users to upload and listen to DJ mixes. It is designed for long-form audio files that have more than one track in them, like a DJ mix, a radio show, or a podcast. The application allows for users to see a tracklisting of a mix and access the individual track on their preferred platform (SoundCloud, Spotify, Apple Music, YouTube, etc).

Design Overview:


Profile
User: OneToOneField(User)
profile_image: ImageField
location: CharField
about_me: TextField
following: ManyToManyField('Profile')
favorites: ManyToManyField('Mix')
0+
1
Mix
audio_file: FileField
uploader: ForeignKey
title: CharField
length: DurationField
playlist: ManyToManyField('Track') with 'time' metadata
upload_date: DateTimeField
play_count: IntegerField
ExternalLink
provider: CharField with choices
url: URLField
track: ForeignKey('Track')
1
0+
User
Default user model given in Django
1+
TrackMembership
track: ForeignKey('track')
mix: ForeignKey('mix')
time: DurationField()
1+
Track
title: CharField
artist: CharField
links: ManyToManyField('ExternalLink')
reference_count: IntegerField

In addition to the homepage, there are routes `/mix/track_slug`, `/profile/profile_id`, `/edit/track_slug`, and `/upload`.

Problems / Successes:

In the upload and edit pages, the desired behavior of the tag system couldn't be accomplished with ordinary javascript. To implement our desired behavior, we had to incorporate jQuery so that we could add, remove, swap, and modify DOM elements. There are still some bugs, but they seem to require a complete overhaul of the editor system, which we plan to do anyway for the next project, as we will need a more comprehensive form to send data back to the server.

The profile page was an interesting process in learning how to extend the base user model to include other fields. Mapping at first was a struggle, but worked itself out with help from others.

Communication was lacking this project. Everyone seemed to be doing their own thing for the first few weeks which led to issues with people not knowing what needed to be done/what was already done. For the next project we will first make a complete plan for how to complete the project and who will do which parts before starting work on it.

Individual Write Ups:

Team Member Name	**Write Up**
Benjamin Murphy	I worked on creating a view for the Charts page and added some mock data to the database. The charts view displays a table of mixes sorted by play count and provides links to each mix page. This view uses the Mix model to display relevant attributes such as mix title, length and artist. To accomplish this, a simple for loop block is used to iterate through each mix in the associated object_list of the charts view. This object_list is initially sorted by the play_count attribute within the charts view class. Currently functionality does not exist to sort by other attributes however this will be implemented in the future. Overall I'd say I contributed 10% of the work done for this project.
Daniel Maryanski	I primarily worked on the Upload and Edit Mix views. The Edit Mix view populates a form with the data from a mix so that the user will be able to make changes to the Mix object. The Upload Mix view is similar to the one that we had in Project 1, because it doesn't need to pull any data from the database to be created. To accomplish this I created the three templates "editor_generic", "editor_edit", and "editor_upload" and the javascript file "upload.js". I also did a few small tasks such as editing models and adding to the navbar. Overall I would say I did about 20% of the work for this project. We got a lot done this time around and I'm confident that we will end up with a great final product after the next two projects.
Daniel Szymanski	I worked on the profile viewing pieces of this project, as well as going back afterwards and populating the actual URL's into the nav-bar, and some additional mapping throughout the site. The profile model has a 1-1 relationship with a User, so it extends into the profile class. The view generates a list of mixes that are filtered by whichever profile is being looked at to be put in it's content. Due to there not being a way to be mapped to a profile without it going through the primary key, there was no empty profile. During this process, I feel like I hit a few walls, that kept me stuck for a few hours, but with some help from groupmates, I eventually figured it out. The main struggles here were generating the mixes that belonged to the profile and mapping them backwards to the mix page they were associated with. After I was done with this part, I poked around in other areas, doing missing odds and ends such as adding the links to the nav-bar, adding in other mappings, and updating the data model diagram. 20% work.
Matthew Robinson	For this part, I worked on adding the mix detail view (/mix/<slug>) with the tracklist sidebar and audio player. I encountered challenges trying to get the data model accessible in the javascript code so that I could link up the audio player with the current track playing and display visual indicators. I ended up having to define a custom context object and serialize this to do so. I also connected our application to a cloud storage system, so we could add mixes to our database without adding the large files (100mb each) to source control. We discussed our data model design together in a shared document, but I implemented the mix and track models in python. I would say I contributed 25-30% of this project.
Shipeng Yu	I worked on the main page, instead post link view, I changed it into our mix style in a card-desk. Once you click images, it will get into right page of Mix-music. Mapping url, change files of views and models part.Same nav-bar as others, you can go any page with this nav-bar. The database works for my mainpage, it get randoms 3 mixes in our mainpage, and image links to url. I would say I did 10% work, teammates helped a lot, especially Matthew. I really love this team and feelings work with them.
Yuzhuo Shi	For this part, I added a new account setting page. I pretty much just add a new html template and css file and added some code in view.py and url.py. Even though the user information can be displayed in the editing page, we haven't figured out how to change data in database after user edit.