17-437 Project Specification

Team 12: Aditya Ranade, Akshat Jain

Our web-application TrackMonger gives users the ability to stream, create and share user-curated playlists. The application has 3 main parts - Playlist Creation which allows users to search for songs and add them to a playlist, Home Feed which shows trending and recommended playlists, and Playlist Share which allows users to comment on and upvote/downvote playlists created by other users.

Product Backlog

Playlist Creation:

- Logged-in users will be able to create a custom playlist using the search and add functionality.
- The search bar will allow users to search for songs by album, artist and/or name and add them to his/her playlist. We will be using the Spotify API on the back-end to query the Spotify database for the user-inputted string.
- The Playlist page will have an embedded Spotify Audio Player allowing the user to click on a button and stream the music.
- Logged-in users can share their playlists with other users or make it public.

Trending/Recommended Playlists:

- The Home Feed will show the user the top trending playlists.
- These top trending playlists are dynamic depending on the number of listeners in the past week and the number of votes it currently has.
- The Home Feed will also showcase recommended playlists for the user. This recommendation system will be based on which artists and genres the user listens to. Upon login, the user will be asked for their most preferred genres and artists, which will be used to initialize recommendations for the user.

Playlist Sharing:

- Users can stream other users playlists as long as they are given access or the playlist is public.
- The playlist page will have a comment section allowing users to post comments.
- Users will also be able to upvote/downvote specific songs on a playlist as well as the playlist as a whole.

Description of individual web-pages:

Login/Register:

- In order to use most of the functionality of the web-application, users will have to create an account with a unique username and email id. Users will also be prompted to input their artist/genre preferences on this page.

Home Feed:

- This the first page the user will be redirected upon login/registration.
- The Home Feed will allow users to see playlists of users they currently follow.
- They can interact with the playlists by commenting on them.
- The Home Feed will additionally have all the functionality detailed above.

Create New Playlist:
- Users can click on the "+" button on the navigation bar to be redirected to the playlist creation page where they can search for songs and create custom playlists.
- This page will have a button to allow users to share their playlists.

Playlist Page:
- This page will show all content related to a specific playlist.
- It will contain a comment section, the name of the playlist, a description of the playlist.
- Users will also be able to upvote and downvote specific songs on the playlist on the this page.

User Playlists:
- Users can access this page through the navigation bar.
- It will show all the playlists a user has created and give them the ability to edit any specific playlist.

Profile Page:
- Every user will have a profile page containing a list of their playlists, who they follow and their top genres and artists.

Edit Playlist:
- Users who own the playlist will be able to edit its name, description and songs on this page.
- They can edit and add songs to this playlist based on upvotes and downvotes.

1st Sprint Backlog:

Our goal for the first sprint is to experiment with the Spotify API since it is a core technology to implement the embedded audio and search functionality.

Spotify API (Owner: Akshat Jain)
- Embed Spotify's audio player on our web-application.
- Test whether we can play any song given its uri using this embedded player.

Data Models (Owner: Aditya Ranade)
- Define the User, Playlist, and Song data models in Django. These models will most likely grow in complexity as we begin the implement other aspects of the project.

Data Models (Django)

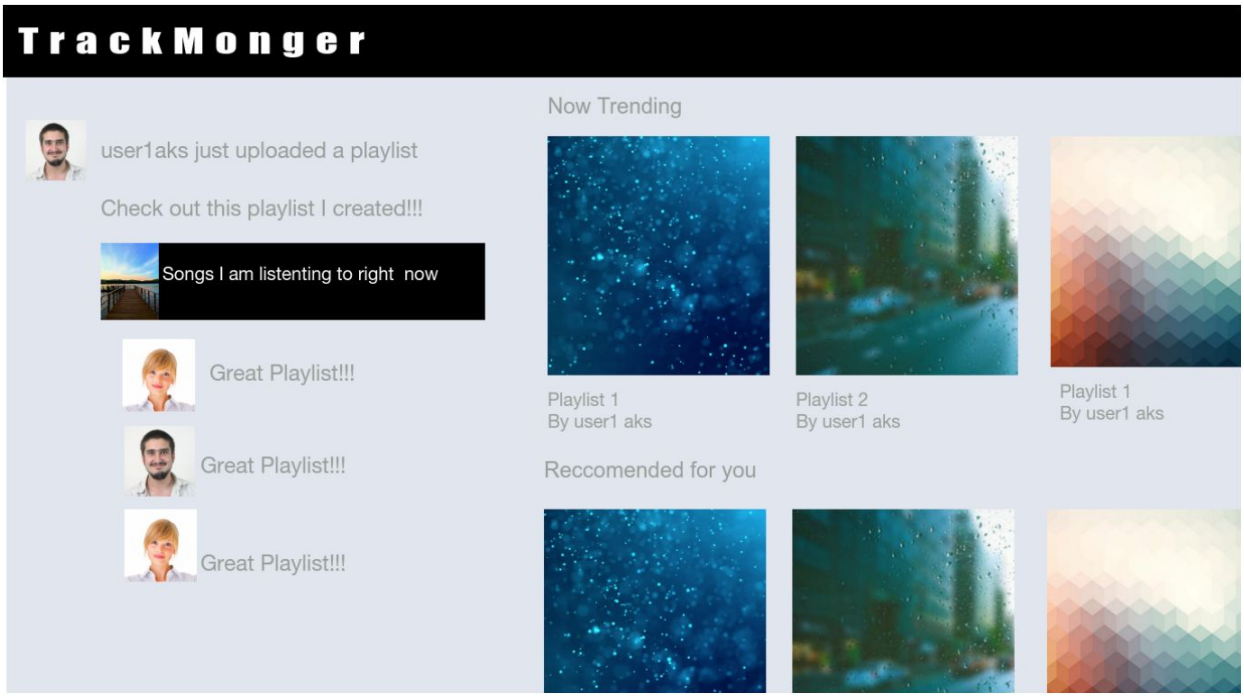
```python
class Profile(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    image = models.ImageField()
    playlists = models.ForeignKey(Playlist)
    # list of genres sorted by how many times a user listens to it
    topGenres = models.TextField()
    # list of artists a user most listens to
    topArtists = models.TextField()
    recommendedPlaylists = models.ForeignKey(Playlist)

class Playlist(models.Model):
    createdUser = models.ForeignKey(User, on_delete=models.PROTECT)
    name = models.CharField(max_length=100)
    upvotes = models.IntegerField(default=0)
    downvotes = models.IntegerField(default=0)
    description = models.TextField()

class Comment(models.Model):
    content = models.CharField(max_length = 100)
    created_by    = models.ForeignKey(User, on_delete=models.PROTECT, related_name="entry1")
    creation_time = models.DateTimeField()
    playlist = models.ForeignKey(Playlist)

class Song(models.Model):
    name = models.CharField(max_length = 100)
    url = models.TextField(validators=[URLValidator()])
    playlist = models.ForeignKey(Playlist)
    upvotes = models.IntegerField(default=0)
    downvotes = models.IntegerField(default=0)
    artistName = models.CharField(max_length=100)
    genre = models.TextField()
```

Mock-up Views:

Login/Register

Playlist Page:

Home Feed:


TrackMonger
user1aks just uploaded a playlist
Check out this playlist I created!!!
Songs I am listenting to right now
Great Playlist!!!
Great Playlist!!!
Great Playlist!!!
Now Trending
Playlist 1
By user1 aks
Playlist 2
By user1 aks
Playlist 1
By user1 aks
Reccomended for you