

Members:

Rahshel Brown (RB)
Aalique Grahame (AG)
Jason Hugh Sam (JHS)
Jessica Shen (JS) <3
Christtia Zheng (CZ)

A complete list of all functionality and descriptions:**Social:**

Login - user logs in to web app

Register - new user registers for web app services

Home - shows your mixes and a feed of users that you're following

Edit Profile - edit user info

Profile - displays all of user's mixes and playlists

Follow - follow other users

Unfollow - unfollow other users

Submit Posts - submit posts containing music from studio

Delete Posts - remove post from studio

Edit Post -edit post text

Change Post Privacy - change the settings of the post from private (only the user can see it) to public (all users on Cloud DJ) or vice versa

Share Posts - Display another user's post on a user's own profile

Comment - comment on post

Rate - rate posts on a scale of 1 to 10

Create Competition - A user can start a musical contest open to all other cloud dj users if public and invite only, if private. The contest consists of creating songs or mixes with given elements and

Scoreboard!!! - know who's the best at cloud djing

Music related:

(CZ) Suggested songs - suggests songs based on genres or hashtags of posts that the users ranks highly as well as songs frequently listened to by users they're following

(AG) Studio- user edits and creates music on this page and can submit those songs in posts

(JHS) Create Playlist - make a list of songs to be played

(Group) Music editing:

- (CZ) Amplify - increases volume of music
- (CZ) Bass - increase volume of bass tones
- (CZ) Treble - increase volume of treble tones
- Change Pitch - change pitch by raising frequencies
- Change Speed - changes the speed (this also affects frequencies)
- Change Tempo - changes the speed while keeping frequencies the same
- Reverb - adds echo effect

- (AG) Fade in - fades music in (volume shift)
 - (AG) Fade out- fades music out (volume shift)
 - (AG) Repeat - play portion of music again
 - Reverse - plays the song in reverse
 - (RB)High Pass Filter - remove low frequencies
 - (RB)Low Pass Filter - remove high pitch frequencies
 - (RB)Band Pass Filter - allows a specific frequency range
 - Slice - users can select portions of the song that they want to keep or remove
- (RB) Record - convert (sound or a performance) into permanent form for later reproduction
- (JHS/RB) Upload music - transfer music from user computer and displays it on page with waveform

A complete list of the **functionality you will complete during your first sprint**, and how that work is allocated among your team members. This is your first sprint backlog.

Everything under music related (see above)

A complete implementation of the data models used by your application.

```
class Profile(models.Model):
    user = models.OneToOneField(User)
    followers = models.ManyToManyField('self', symmetrical=False, related_name="following")

    def __unicode__(self):
        return self.user.username

class Comment(models.Model):
    profile = models.ForeignKey(Profile)
    post = models.ForeignKey(Post)
    text = models.CharField(max_length=200)
    date = models.DateTimeField(auto_add_now=True)

    def __unicode__(self):
        return self.text

class Post(models.Model):
    profile = models.ForeignKey(profile)
    text = models.CharField(max_length = 255)
    date = models.DateTimeField(auto_add_now=True)
    photo = models.ImageField(upload_to='album-art', default='album-art/randompic.jpg')
    song = models.FileField(upload_to='music')
    plays = models.IntegerField(default = 0)
    genre = models.CharField(max_length = 255)
    hashtags = models.ManyToManyField(Hashtag, related_name='posts')
```

```
def __unicode__(self):  
    return self.profile.user.username + ': ' + self.text
```

```
class Rating(models.Model):  
    profile= models.ForeignKey(Profile)  
    post = models.ForeignKey(Post)  
    rating = models.IntegerField(min_value=1, max_value=5)
```

```
class Hashtag(models.Model):  
    text = models.CharField()
```

```
class Playlist(models.Model):  
    posts = models.ManyToManyField(Post, related_name='playlist')  
    profile = models.ForeignKey(Profile)
```