SOFTWARE REQUIREMENTS SPECIFICATION

for

MoodCat

Release 1.0

MoodCat.me

Functional requirements

#	Name	Description
	Website	
1.1	Front page	When the user is not logged in and visits the homepage, he will be redirected to the log-in screen, else the user is redirected to the mood-selection page.
1.2	Mood-selection page	On this page the user can select his mood. When the user selected his mood, he gets offered a list of compatible rooms.
1.3	Joining a room	The User can now pick a room and join it.
1.4	Room	When the user enters a room, he is served songs based on his chosen mood.
	Room	
2.1	Listening	The songs are streamed from SoundCloud.
2.2	Vote to skip	When a user doesn't like the song currently streamed in the room, the user can vote to skip the song. If enough users want to skip the current song, the next song is streamed to all users in the room.
2.3	User management	Users periodically receive suggestions of the system to join a different room.
	Learning Algorithm	
3.1	Perceptron	We use a multilayer perceptron to predict the mood of a song. The perceptron is trained on song metadata received from EEMCS.
3.2	Recommender system	A recommender system will select the next song that should be played in a room.
	Gamification	
4.1	Rewards	At a various levels for user points, the user receives rewards.
4.2	User contribution	The user can contribute to the recommender system and receive points in return.

Non-functional requirements

#	**Name**	**Description**
1.1	Versioning	Git will be used as a versioning system.
1.2	Pull Requests	For every completed feature a pull request is created. A pull request will be merged to the master branch if it fulfills all non-functional requirements below.
1.3	Test coverage	At all times, the test coverage on the core must be more than 80%. Every new feature needs manual testing.
1.4	Checkstyle warnings	Whenever a pull request is submitted, it is not allowed to add more than 5 Checkstyle warnings compared to the master branch.
1.5	Continuous Integration	Travis CI will be used for continuous integration. The master branch builds are never allowed to fail.
1.6	Documentation	Every package, class and method must have Javadoc. Angular controllers must have NgDoc.
1.7	Issues	Github issues must be used in combination with waffle.io.