TrackHub
Requirements Analysis

www.trackhub.com

Clement Garcia - c_garcia30@u.pacific.edu

Edgar Oregel - e_oregel@u.pacific.edu

Drew Overgaard - d_overgaard1@u.pacific.edu

Kyle Hartman - k_hartman@u.pacific.edu

Ryan Espinosa - r_espinosa1@u.pacific.edu

Last Revision: Feb. 9, 2018

System Description and Project Overview

This section should give an initial introduction to the system, the project stakeholders and potential system users.

It should include the following information:

- Motivation for the system.
    - Our motivation came from existing music players, such as SoundCloud, Itunes Music, and Spotify, and their limitations. As nice as the they are, they have their faults and features that limit users to perform simple actions. This is where we want to jump in with our implementation of a media player. We want to address these small features that are ignored by other media players as well as give our media player a lot more features to make it easier and intriguing to use it as a Social media platform.
- The nature of the problem the system will solve.
    - What our media player will solve is small features that are ignored by existing media players that are simply annoying that other people see as a something that would be “cool” or helpful if it existed. These features include a chat feature when sharing music/playlists, allowing people to not only collaborate on playlists, but be able to clone or fork a personal version of the playlist to only affect changes in your playlist and not the other person’s as well, and easier ways of sharing music/playlists to external social media platforms.
- How this problem is solved in the current existing systems (current state of the art).
    - There are several media players that exist that all have some form of implementations of our ideas, however, none of them have them all. Things like making it easy to share to several external social medias and a more social media player exists more on SoundCloud than Spotify. Spotify allows you to collaborate on playlists, but if someone makes a change, it will reflect for both sides. We want to take a look at current implementations and ideas that were addressed or attempted in current media players and change it to our view and ideas of what a true “Social” media player should look like.
- Project stakeholders (managers, sponsors, user, customers, etc.).
    - Music enthusiasts who want to listen to music
    - Curious users who want to know what music is popular
    - Curious users who want to know more info about music
    - Music inclined individuals who want to mix music from existing songs
    - Social media users who want to share their preference of music
    - Music critics who want to share their opinions on music
    - Users who want to be kept in the loop of the current musical trends
- A description of typical system users.

- This media player will be usable by anyone. The typical user will be one that is into music a lot as well as collaborating with others on the top soundtracks and trending albums. The platform will make it easier for these types of users to share their ideas, favorite songs/playlists/albums/artists/etc. to several external social medias and other users within the application.

Use the application domain language and terminology, (the language familiar to your target users), and avoid technical jargon. Utilize charts, illustrations, and screen mock-ups to make it easier for the reader to understand the problem. Provide references to additional information or similar existing systems, if available.

Non-Functional Requirements

Non-functional requirements are requirements that cannot be expressed in terms of use cases (see next section). Non-functional requirements may include things such as system up-time/availability requirements, implementation restrictions, or security requirements. Non-functional requirements are usually best presented as a bulleted list.

- Trackhub user accounts must be securely stored in a DataBase, so that their information is kept private.
- We are aiming for a 24/7 uptime requirement for the TrackHub website. We plan to roll out updates without affecting site availability.
- We want to ensure that Track Hubs user interface is intuitive.

Functional Requirements

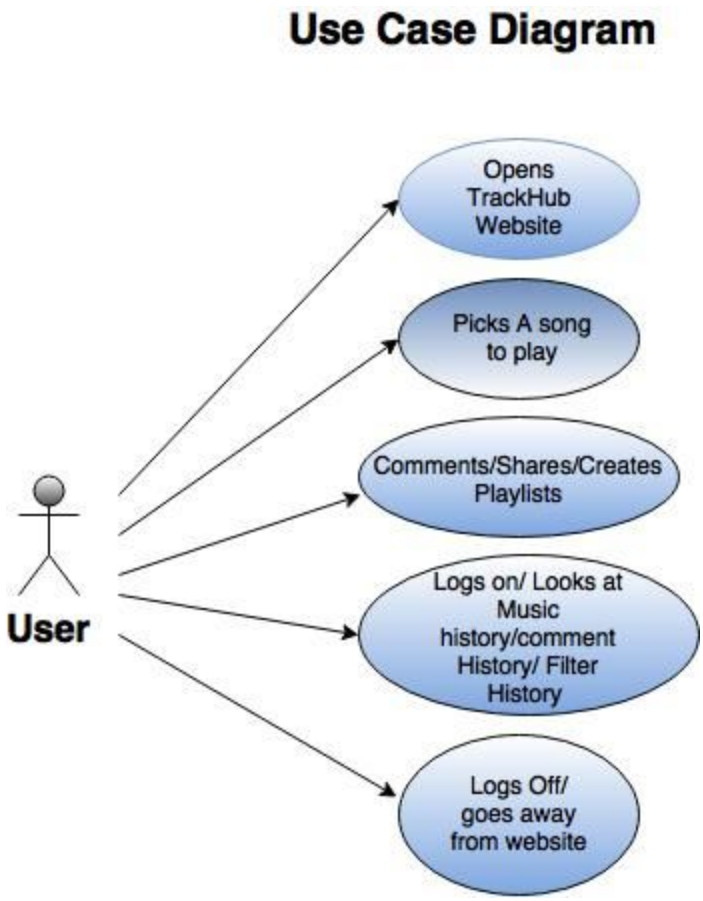
Functional requirements are the core of the analysis document since they define what the system must do. Functional requirements are best defined through use cases (also known as user stories), that explain how the system responds to the actions of entities (humans or systems) external to the system. It is essential to view the system as a black-box, since the internal operation of the system is defined later, during system design.

Use Case Diagram

The first component of the functional requirements is the use case diagram. The use case diagram starts with a box, representing the system boundary (an important part of limiting project scope). External to the system boundary are actors, which represent humans or other systems that interact with the system being defined. Inside the system boundary are bubbles labelled with use case names. The actors are connected to the use cases that represent units of their interaction with the system. Defining an appropriate set of use

cases can be one of the most challenging tasks in the analysis process. Look to the reference textbooks for examples of good use case definitions.

A common mistake in developing the use case diagram is to attempt to indicate flow of control within the system through direct connections between the use cases. Flow of control is not something that use cases are designed to capture. The flow of control will be defined by the pre-conditions and post-conditions defined in the fully-dressed use case descriptions (below).



Casual Use Case Descriptions

The casual use case descriptions are a small bit of informal text (a short paragraph) that describes what the actors do to the system and how the system responds to the actors' actions. Descriptions of internal state changes in the system should be limited to what is apparent or assumed by the actors.

1) The user goes to a website

2) The user sees trending songs in a list with options sorted by the most favorited and options to change the sorting order.
3) The user sees a login or an option to log in when he tries to comment and is not currently logged in.
4) The user creates an account using the account creation page. They enter name, email, and password information.
5) The user presses a play button on a specific song. The song begins to play.
6) If the shuffle button is pressed a next song would play depending on options.
7) The user sees options to see subtitles and music info
8) The user sees an option to share the music.
9) The user presses a button next to a song. They are given the option to add the song to an existing playlist or create a new playlist starting with that song.
10) The user has an option to view their account and sees their song history, comments, playlist, used filters.
11) The user quits with the option to log out before leaving the website.

Fully-Dressed Use Case Descriptions

The fully-dressed use case descriptions define pre-conditions, the main flow of events, exceptional flows of events, and post-conditions. The pre-conditions and post-conditions are defined in terms of the internal state of the system, which may be explicit or implicit. Since pre-conditions for one use case may be set as post-conditions of another use case, they provide a very abstract notion of flow of control within the system. Fully-dressed use case descriptions may be omitted for obvious or trivial use cases.

Format = precondition.postcondition ->

0) The user goes to the TrackHub website

1.1) The user sees trending songs in a list with an options menu sorted by the most favorited and options to change the sorting order.

1.1.1) The user focuses on the menu and sees a play button, Shuffle Button, Share button, Next Song button, Previous Song, Add to playlist button, comment button.

1.1.1.1)The user presses a play button on a specific song. The song begins to play.

1.1.1.1.1)The user sees subtitles when the music is playing

1.1.1.2)If the shuffle button is pressed a next song would play depending on options.

1.1.1.3)The user sees an option to share the music.

1.1.1.3.1)The user shares the music on facebook,twitter, google plus or email.

1.1.1.4)The user presses a button next to a song.

1.1.1.4.1) The user goes to next song.

1.1.1.5) The user presses add to playlist button

1.1.1.5.1) List of playlists show up that the song can be added if logged in.

1.1.1.5.2) If user doesn't have a playlist, they go to the playlist creation menu.

1.1.1.5.2) User sees the playlist creation menu and has the ability to create playlists or delete playlists.

1.1.1.5.3) If user is not logged in go to 1.2.1)

1.1.1.6) The user sees a comment button

1.1.1.6.1) The user goes to 1.2.1) if not logged in

1.1.1.6.2) The user comments in a comments section

1.2)The user sees a login button and clicks it

1.2.1) The user sees a login option and enters information to login

1.2.1.1) The user gets redirected to step 1)

1.2.1.2) The user gets a message invalid login if invalid login. And gets create account button that redirects to 1.2.2 or try again button 1.2.1

1.2.2) The user clicks create account. The user creates an account using the account creation page. They enter name, email, and password information.

1.2.2.1) The user gets redirected to step 1)

1.2.3) The user sees account button in place of login button and clicks it.

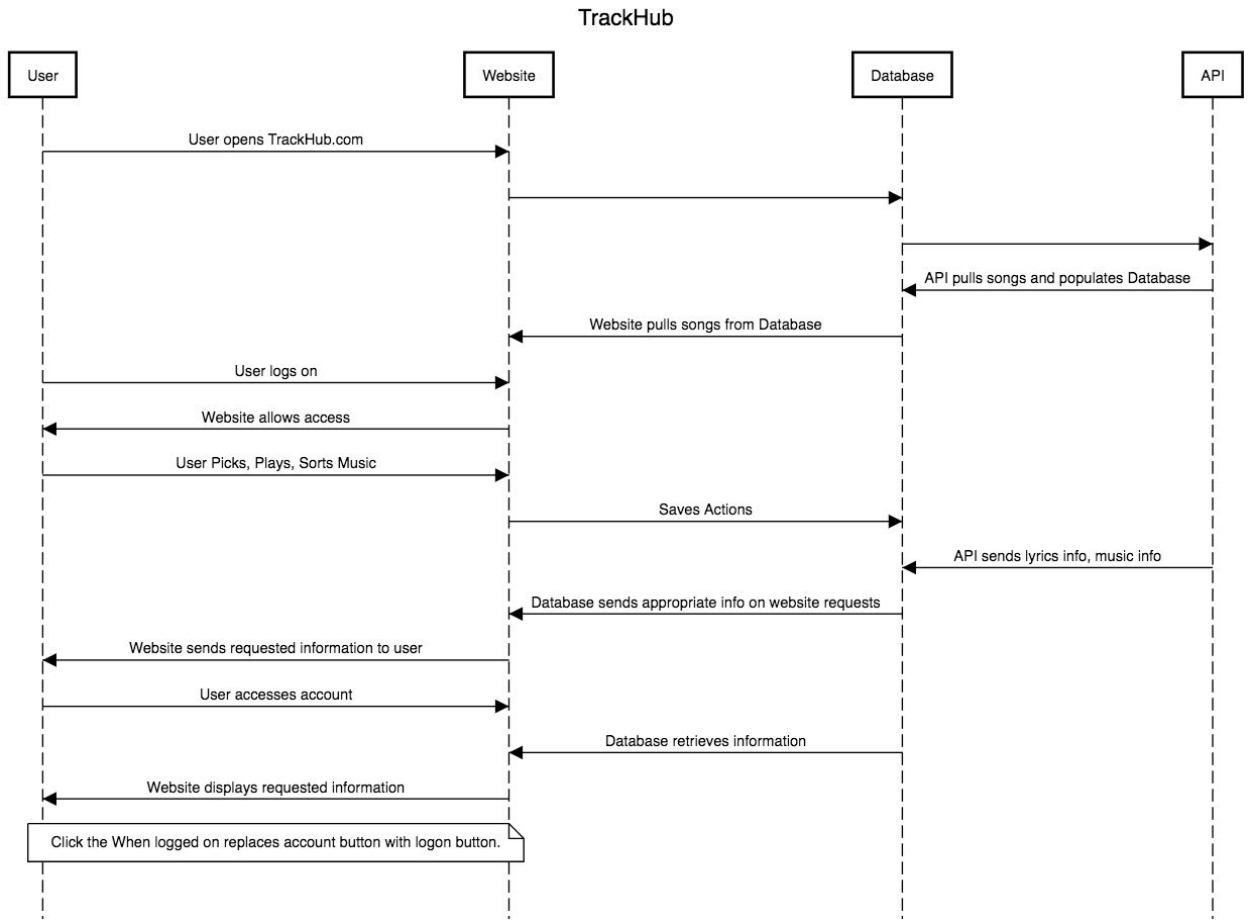
1.3.1)an option to view their account and sees their song history, comments, playlist, used filters.

1.3) The user quits with the option to log out before leaving the website.

1.4) The user has the option to filter their search based on mood or personality by entering information in a filter search box and the list of songs changes accordingly.

System Sequence Diagrams

System sequence diagrams define the sequence of interactions between actors and the system, as well as a brief description of the information flow through each interaction. There should generally be one system sequence diagram for each fully-dressed use case.

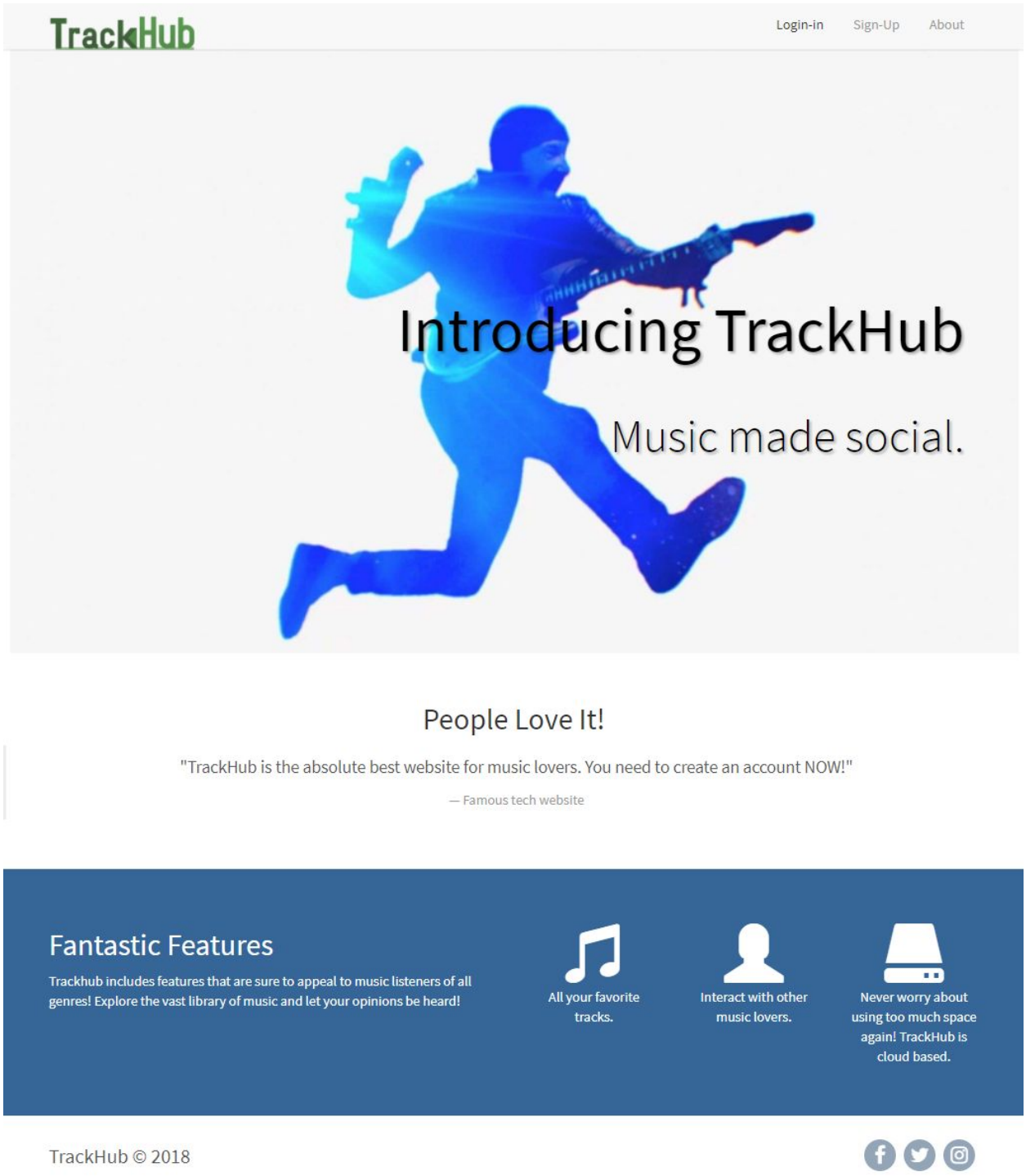


User Interface Design Sketches

Give sketches or mock-ups of the interfaces that human actors will use to interact with the system. If there is sequencing of interfaces implied by the use cases or sequence diagrams, describe that sequence graphically or through text. If the system produces reports or forms, give mock-ups illustrating how they might look. Depending on the nature of the project and

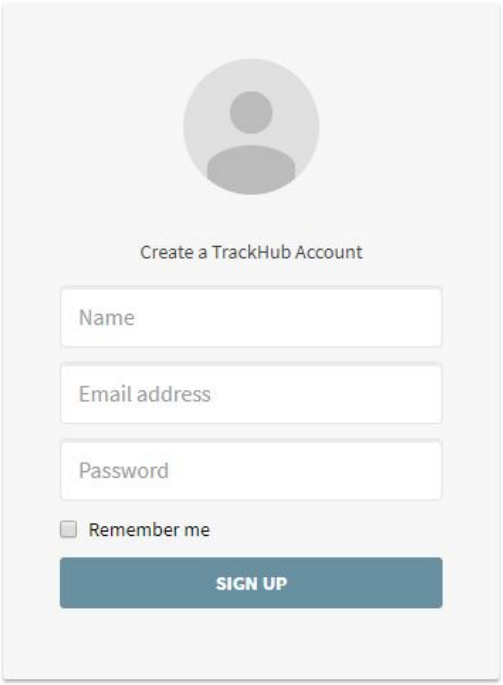
the interests of the project stakeholders, the user interface sketches may be used to specify the information content of interactions, the artistic style of the interfaces, or both.

A simple landing page design. The first page of the website a new or not signed-in user will see.

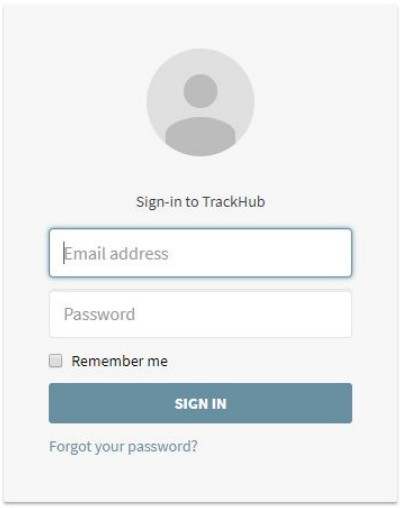


A create an account interface. Users will create their account here, entering in their name, email, and password.

Last Revision: Feb. 9, 2018

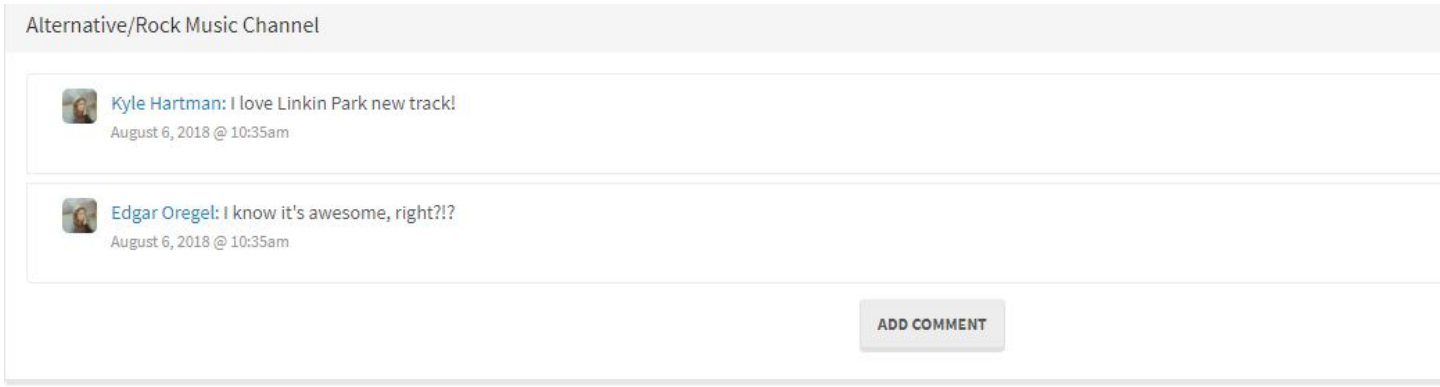


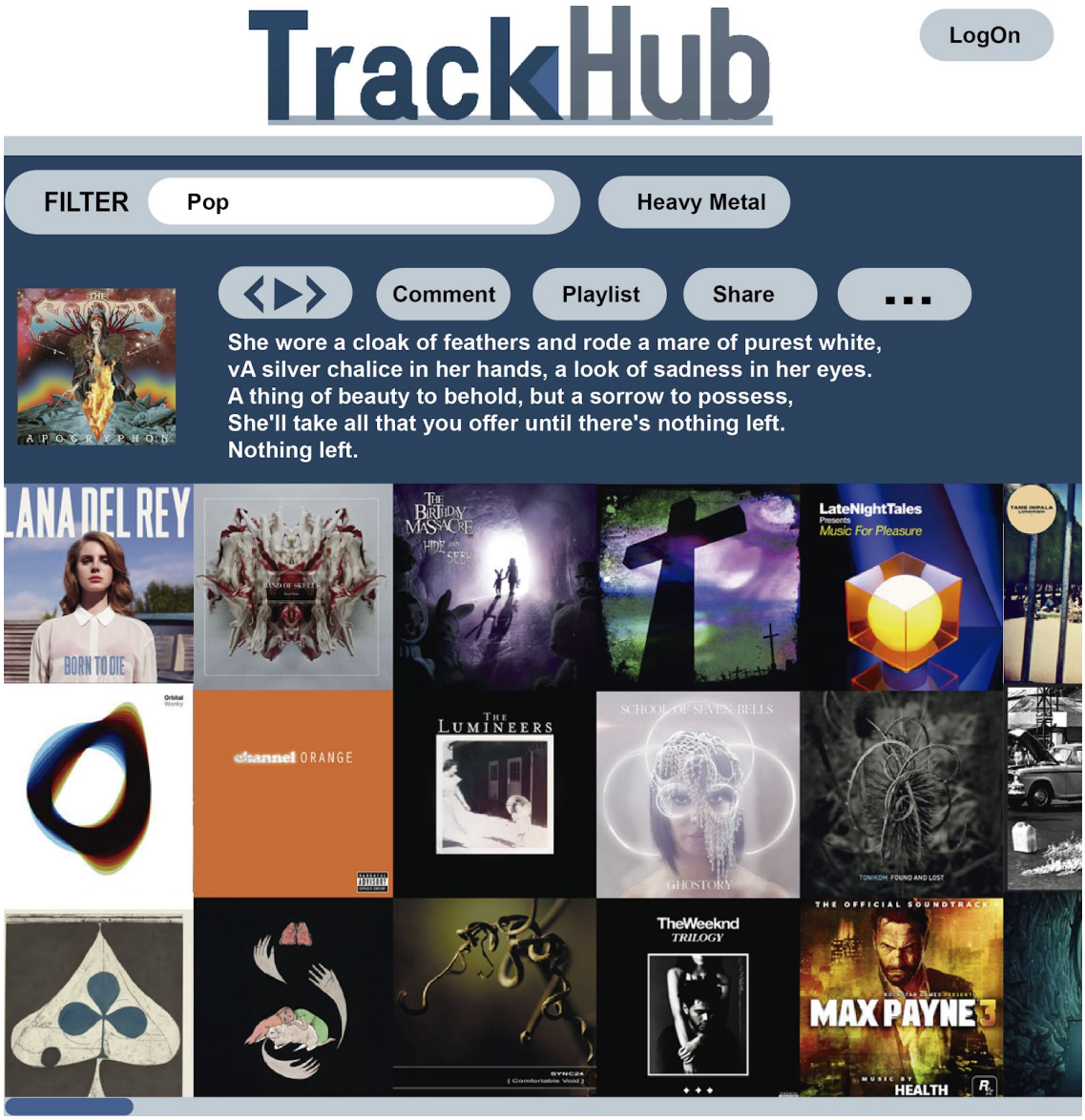
A Sign-in page for existing users.



A simple chat channel/forum mock-up. This where users can share music and post comments.

Last Revision: Feb. 9, 2018


Alternative/Rock Music Channel
Kyle Hartman: I love Linkin Park new track!
August 6, 2018 @ 10:35am
Edgar Oregel: I know it's awesome, right?!?
August 6, 2018 @ 10:35am
ADD COMMENT


TrackHub
LogOn
FILTER
Pop
Heavy Metal
Comment
Playlist
Share
...
She wore a cloak of feathers and rode a mare of purest white,
vA silver chalice in her hands, a look of sadness in her eyes.
A thing of beauty to behold, but a sorrow to possess,
She'll take all that you offer until there's nothing left.
Nothing left.
LANA DEL REY
BORN TO DIE
LateNightTales
Presents
Music For Pleasure
channel ORANGE
THE LUMINEERS
TheWeeknd
TRILOGY
MAX PAYNE 3
HEALTH

Glossary of Terms

List important terms and their definitions used in the RAD, to ensure consistency and avoid ambiguity in the system specification. Use the language of the application domain and avoid uncommon terms or define these as well.

Crossfade - Tracks seamlessly playing one after the other.
API - Application Programming Interface; allows developers to integrate software from outside sources into their own applications.

References

The list of references should contain exact references and/or URLs of any material that is cited in the
analysis document. The references should be formatted consistently using ACM, IEEE or APA style. Do
not mix citation styles. The following sites may be helpful for formatting your references.

Personality types. (n.d.). Retrieved February 08, 2018, from https://www.16personalities.com/personality-types
Find personality traits to sort Music

Myers Briggs (MBTI) Types As Musical Genres. (n.d.). Retrieved February 08, 2018, from http://personalitygrowth.com/myers-briggs-mbti-types-as-musical-genres/
Music by personality