RentIt

Andreas Precht Poulsen, Morten F. Therkildsen, Mark Thorhauge, Toke Loke & Mikkel Funch

28-05-2013

1 Vision

RentIt is a radio hosting service, that enables users to host online radio stations and for other users to listen to them. It is a system intended to be used by the common user, without any special knowledge or traning.

2 Specification

This section contains the fucntional and non-functional requirements, as well as the use cases that they are based on.

2.1 Use Cases

Create account: The user navigates to the ”create account”/”register” page. He fills out required information and agrees on eventual conditions.
Delete account: The user does not want to use the service any longer, he has chosen to delete his account. All data about the user is deleted accordingly
Listen to channel: The user navigates to the channel, possibly by searching for it and selects listen to channel.
Subscribe to channel: The user has selected a channel and selects subscribe to channel. The subscription status changes accordingly.
Unsubscribe to channel: The user has selected a channel and selects unsubsribe to channel, if he/she is already subscribed. The subscription status changes accordingly.
Comment channel: The user is on a channel and want to comment on it. He writes his message and submits the comment. The comment is saved and the page is refreshed
Create channel: The user navigates to ”create channel” and fills out the required information as well as uploading tracks. When the user is done it is immediatly accessible by other users.
Delete channel: The creater of the channel does not want the channel anymore and he decides to delete it. All subscriptions to the channel is deleted, subscribers gets notified, and the channel is removed.
Upload track: The user has a channel and wants to upload a song. He selects the song through a filebrowser and selects upload.
Delete track: TODODODDODODODO
Edit channel: TODODODDODODODO
Upvote/downvote track: TODODODDODODODO

2.2 Functional requirements

The functional requirements describe the required functionality of the system and defines enumerations for reference.

R0: The system shall allow all users to create a channel, add attributes and edit it.
Domain concepts: The channel shall have its own collection of tracks that it plays and it is created and maintained by the channel host(the creator of the channel).
The channel shall have genres, a description and comments posted by users.
The channel must be visible to other users with prober search parameters.
Dependencies: The requirement is nessecary for upload track and listen to channel because a track is associated with a channel.
Priority: It is of essential importance, as the requirements depending on it is.

R1: The system must allow users to listen to online radio-channel that streams audio.
Domain concepts: A channel has a collection of tracks that can be listened to.
Priority: Highest as this is the primary service that the system provides.
Implementation notes: The system must be able to stream audio in a smooth and clutter-free manner. A radio-channel must be able to stream an endless stream of audio without pauses between audio tracks. The streaming must be implemented as a webservice(Windows Communication Foundation) and it must be easy to write a new clients for the service. The interface for listening to audio must be as user friendly as possible.

R2: The user must be able to rate a track associated with a channel.
Domain concepts: The rating of a track must affect the frequency with which it is played within its associated channel. A higher rating giving a proportionally higher chance of being played.
Dependencies: The listen to channel requirement are depended on this requirement because the track to be played is determined by the rating.
Priority: It is of essential importance, as the requirements depending on it is.

R3: It should be possible for users with a channel to upload a track to it.
Domain concepts: A channel has tracks that the creator manages.
Dependencies: The listen to channel requirement are depended on this requirement because the track to be played is determined by the rating.
Priority: It is of essential importance, as the requirements depending on it is.

2.3 Non-functional requirements

The non-functional requirements describes how the functionality shall work in multiple aspects, as well as defining success criteria for them. They are structured with FURPS+.

Functionality
Channels
Channel owners must be able to assign genre tags to the channels they control/own.
Two users listening to the same channel at the same time must hear the same song with same amount of elapsed time (+- 1s), in other words, a channel is "playing" the same for all listeners.
A function for searching for channels must be implemented.
A list of the most popular channels should be available to all users. Popularity must be based on an algorithm and it must be explained in the documentation.

Tracks
The service must support tracks in .mp3 format.

Logging and Error Handling
All exceptional states must be logged.
All state changes must be logged.

Security
All usage requires authentication. (usage meaning the use of any services related to the program)

Usability
The website should be intuitive to use for 90% of persons above the age of 12 with regard to the following:

- listening to a channel
- maintaining one.
- uploading songs
- voting
- subscribing

There must be help available for all functions in the form of text. This help must be accessible from within the client
Users must not be required to subscribe to a channel in order to listen to it.
Users must be required to be subscribed to a channel in order to vote on the channel.
If a user unsubscribes from a channel, all votes made by that user must be removed so as to not affect track ordering.
Users must be able to see all the songs contained in a channel.
Users must be able to upvote and downvote the last 5 songs that have been played on the channel while they were listening.
Users must not be required to download or install any third party software in order to listen to a channel.

Reliability
The server must not be affected by any errors clientside