**Project Netlabel**  
By Dominic Wanner & Dennis Rösner

Inhalt

[1. What is „Netlabel“ 4](#_Toc362034489)

[2. Languages 5](#_Toc362034490)

[2.1 Javascript 5](#_Toc362034491)

[2.2 Jade 6](#_Toc362034492)

[2.3 Stylus 6](#_Toc362034493)

[3. Technology overview 7](#_Toc362034494)

[3.1 Server Side 7](#_Toc362034495)

[3.1.1 Node.js 7](#_Toc362034496)

[3.1.2 Express.js 8](#_Toc362034497)

[3.1.3 Jade 9](#_Toc362034498)

[3.1.4 Stylus 9](#_Toc362034499)

[3.1.5 MongoDB 10](#_Toc362034500)

[3.2 Client Side 11](#_Toc362034501)

[3.2.1 Twitter Bootstrap 11](#_Toc362034502)

[3.2.2 JQuery 12](#_Toc362034503)

[4. Workflow 13](#_Toc362034504)

[4.1 Scrum 13](#_Toc362034505)

[4.1.1 Sprint 1 13](#_Toc362034506)

[4.1.2 Sprint 2 14](#_Toc362034507)

[4.1.3 Sprint 3 14](#_Toc362034508)

[4.2 Pair Programming 14](#_Toc362034509)

[4.3 Github 14](#_Toc362034510)

[5. The Project itself 15](#_Toc362034511)

[5.1 Server architecture 15](#_Toc362034512)

[5.2 Application architecture 15](#_Toc362034513)

[5.2.1 Model-View Controller 16](#_Toc362034514)

[5.3 Pages 16](#_Toc362034515)

[6. Future Aspects (what needs to be done) 17](#_Toc362034516)

[7. What we’ve learned 18](#_Toc362034517)

[7.1 Bootstrap 18](#_Toc362034518)

[7.2 Node.js / Express.js 18](#_Toc362034519)

[7.3 Stylus 18](#_Toc362034520)

[7.4 Jade 18](#_Toc362034521)

[7.5 MongoDB 18](#_Toc362034522)

[7.5 Other Technologies 18](#_Toc362034523)

# 1. What is „Netlabel“

The Netlabel is, like the name says, a label for musicans in the internet where everyone can register for free. The musicans can upload their songs on our page by accepting our agbs and contract.

At this point the musicans album is running under our label and all rights of it belongs to our company for a limited time.  
We send the songs to „Kontor“ a „content aggregator“ which spreads the songs tp all their music reseller (for example: itunes).

After the upload process succeeded we gain billings by kontor which were spreaded between the musicans (minus a specified percentage amount)

The Netlabel wants to specify on metal/gothic music because of the trend to specified markets that know the special needs of the customers

In future The Netlabel could be extended by its own mp3 shop and community to perfectly sell advertise, and specified shopping.

# 2. Languages

## 2.1 Javascript

Javascript is thought as a client sided scripting language. It was developed in 1995 by „Brendan Eich“, it is not to confound with Java. It is normally used for client sided validation, animations, ajax calls. It is possible to program javascript object oriented but also procedural or functional. It is made for changing html and css attributes dynamically, it is able to address each html and css object individually or as group. The more usage of javascipt, the less performance of the server is used. The Syntax reminds on normal c-descendants.

Weakness:

* Browser compatibility problems
* Can be modified by user
* Garbage collector can’t be modified
* User can deaktivate js
* Variables casts very fast automatically

Strength:

* Easy to learn
* Easy Syntax
* Script/programlanguage
* Linkable through the web
* Event based
* Asynchrounous executions

**Javascript as Serversided Language:**

Javascript was chosen for node.js because it is well known by many developers, it is the best programming language for eventbased programming and it is perfectly working asynchronous.

We chose javascript as serversided Language because we both know well how to use it and because we were convinced of the speed of node.js

## 2.2 Jade

Jade is our replacement for html, we use it because of the save variable handling, because of the clean code and fast programming.

Weakness:

* Needs compiling

Strength:

* Tab based
* No bracket chaos
* Fast to learn
* Fast to programm
* Safe use of variables
* More clear code
* Jade documents stack into each other

## 2.3 Stylus

Stylus is our replacement for css, we use it because its clean code, predefined variables, and faster programming

Weakness

* Needs to compile

Strength

* Faster developement
* No brackets/semi-colons/colons
* Possability of variables
* Fast learning
* Compiles for looking equal on every browser
* Ability to use functions

# 3. Technology overview

## 3.1 Server Side

## 3.1.1 Node.js

The web is changing – users want to interact with the web evermore, they want to use realtime technology like chatting, gaming, socialmedia updates, etc. simultaneously to that the number of users is growing this means we need to handle massive real time requests between client, server, and other servers.

Http wasn’t made for that, many clients polling one server at one time is very inefficient.  
So we needed a server-model that uses lightweight communication which is easily handleable as software developer. This is where Node.js comes in.

Node.js is an event-driven, fast, server-sided Javascript Enviroment using Googles V8-Engine for javascript compiling. V8 directly compiles JS into machine-code, what makes it so fast.  
Additionally to the V8 Node.js uses the „Event-loop“ to handle large numbers of clients.  
the „Event-loop“ is based on „callbacks“ this means, that every step the programm should run through should be placed in the callback oft he step before. This enables asynchronous operation handling. This minimizes the memory usage per client, by not using a new thread per client. Also if something blocks the actual task, node.js starts another task. This makes Node.js „Unblockable“

Node.js automatically manages, handles and optimizes the Event-loop, what makes it easy to handle. Node.js has a fast growing community, which offers modules for nearly everything. For example Express.js: for a responsable design architecture and a more easy handling of node.js MongoDB(or other db drivers): as driver for mongoDB SocketIO: for easily opening sockets between server and client the actual number of all modules for node.js is over 35.000. All modules are download and installable using „NPM“ – NPM(Node Package Manager) is installed with node.js and an easy to use, linux package manager similar download manager. NPM can download everything within its community, which is open for everyone.

### 3.1.2 Express.js

Express.js is the most popular web framework, inspired by „Sinatra“. It is a thin layer with has much effect on the work with node.js.  
With its powerful api, it makes starting with node easy. But also great web applications benefit of ist power.

If you start building an Express app, Express creates you a responsable folder structure containing a „hello world“ app. The architecture is easily understand- and modifyable.  
Also routing is made easy by express each route is stored in the „app.js“ file and redirects to the specific operation in other files.

Applications powered by Express:

Myspace.com – social Network platform  
LearnBoost – free online education suite  
Storify – social Network for stories  
Geekli.st – social network for geeks  
Klout – influence comparison platform  
Prismatic – social network which trys to fit for your needs  
Clipboard – platform to share anything online  
Mozilla Persona – identification system  
Countly – real-time analytics service  
Balloons.io – realtime chat  
Segment.io – analytic distributor  
Yummly – world largest recipe search site  
Koding – Realtime software developement in browser  
Apiary.io – REST API documentation  
and many more…

### 3.1.3 Jade

Jade is a template engine specificly developed for node.js. It simplifies html with an high performance compiling engine. Jade got influenced by Haml with an Javascript extension.  
The advantage of using Jade with Node.js is that all variables can be passed safely to the jade compiler (middleware), after placing the variables in the right place, jade renders out clean html with all used variables as static text. This makes it impossible that variables can be changed by users during runtime.  
Also the user can’t see the difference to any other html page because no jade document is passed to the client.  
Jade is using no brackets at all, every element which is more intended than its parent is automatically in its brackets. All html tags can be easily written as one word, classes can be added by a dot operator and ids are just defined by using an hashtag.  
Jade also enables the usage of standart comments within the code using „ // „ or „ /\* \*/ “.  
All attributes, which were normally written into the tag are just added in normal brackets  
for example: div.class(href=“#“)  
By using node.js variables the possibility of if/for/while statements is given, directly in jade. Jade is easy to learn, because of the similarity to html what makes it the perfect language for fast programming.

### 3.1.4 Stylus

Stylus is a dynamic stylesheet language for css. It is influenced by Sass and LESS and is the third most used css preprocessor syntax language.  
The most obvious advantage is the shorter syntax because no brackets, semicolons and colons are needed but you can use them if you want.  
Additional advantages of Stylus is the dynamic aspect of it. You can define global variables, for example the colors you want to use like: blue = #451577. If u want to change the blue color in your page you only need to change one line. Also dynamic functions can be implemented so your „css“ accommodates itself perfectly to your demands.

### 3.1.5 MongoDB

MongoDB is a modern noSQL Database. It is mainly used for saving all types of files in their natural form. The main advantages of MongoDB are its Agility and Scalablility. MongoDB can be used in various programming languages thanks to its great collection of drivers.  
Also its big community supports your project by every problem.  
Another strength of MongoDB is „GridFS“. GridFS stores files in little chunks, for example you upload a video via gridFS into MongoDB, this video gets stored in little chunkfiles and if you try to watch the video only the part you want to watch is getting loaded. By skipping through the video it doesnt need to load the whole video to the part you skipped to, it just needs to load the nearest part. For finding the files in the DB you are able to add metadata to the files.  
By Using MongoDB the normal way, data is saved in a file format called BSON (spoken bison) which is nearly similar to JSON.  
This makes it perfekt for working with node.js, using simple commands for storing/reading files.  
You can store files in mongoDB without any limitation, it is possible to store an object completely with five attributes in the same collection (table) as an object with seven attributes.  
MongoDB and Node.js fits perfectly because of the asynchronous architecture of it like in Node.js. This makes them work perfectly together.

## 3.2 Client Side

### 3.2.1 Twitter Bootstrap

Bootstrap is a front-end framework for html. It works sleek, powerful and intuitive and makes modern webdeveloping more easy.

Bootstrap consists of several different parts:

1. Responsive Design: Bootstrap is designed completely responsive, this means that a designed layout can be used on pc, tablet and phone, without producing any bugs.  
   This is Possible because of bootstraps „grid system“ every div placed in a row, which defines the top position oft he element. For the width attribute you can put every file in a span. Spans can go from span1 to span12 whereby span12 is the max with of divs in one row.
2. Css Classes: Bootstrap contains many css classes for nearly every html element. For example special styled headings, tables, forms, buttons, images. But also Buttons, Dropdowns, Navigations, and more complex css builts.
3. Icons: Bootstrap offers 140 icons by „glyphicons“ which are limited to a very small size
4. Javascript Animations: Additionally to the classes, many of the the elements like input fields are standartized minimalistic animated. But also many additional javascript animations like fade ins are predefined in bootstrap in compound with jQuer

Bootstrap has many additional modules or other projects which are thought to use with bootstrap.

For example: Font Awesome  
Font Awesome includes a set of 361 Icons which replaces the bootstrap standart icons. But the Icons aren’t saved as png, icon or jpg, they are built as a font. This makes them completely free scalable and colourable.  
Also Font Awesome provides a little repertoire of css classes and css animations for larger icons in relation oft he text or animations for example loading.

### 3.2.2 JQuery

JQuery ist he most well known javascript library. It is fully compatible with all mayor browsers, uses only very little ram space and is css3 complient.  
It consists of many features:

* Javascript animations for many cases for example: .animate()
* A jquery object selector for example: $(„#id“) instead of document.findElementById(„id“)
* Nearly every javascript function working with the object selector, for example: $(„#id“).val() instead of document.findElementById(„id“).value
* Simpler listeners for every possable event .on(„click“, function{}); or .click(function{});

In summary: jquery makes javascript shorter and more functionable.

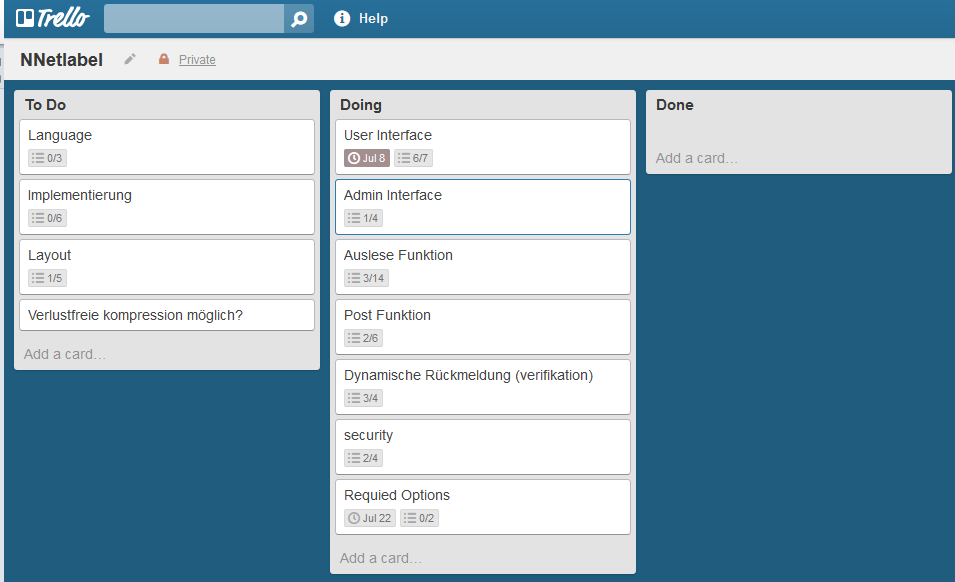
# 4. Workflow

## 4.1 Scrum

Scrum is an agile project management method, which tries to attempt empirical, incremental and iterative.  
Scrum tries to reduce the complexity of projects into three principles:

* Transpirancy: the advancement and obstacles will be revealed every day and set down, visible for everyone
* Review: the productfuntionality is getting delivered and approved by the costumer in certain intervals.
* Adjustment: the requirements oft he product will reviewed and readjusted on demand on every delivery

It aims on fast, economic and qualitativ completation of the product. Because of our expirience from other projects on this university, we won’t go deeper in this topic.



### **4.1.1 Sprint 1**

Concentrated on the Visual Part of Page (Gui)  
Mostly in Jade  
Cards:

* Navbar – for navigation
* Dashboard – for overview of the profil
* Uploader – for uploading songs
* Settings – for contact/banking/preferences
* landingpage
* Including fonts
* Basic route architecture

### 4.1.2 Sprint 2

Concentrated mostly on functional (background) part of the page

* Login/registration process
* MongoDB uplink
* Client animations
* Restyling of pages
* Including Footer
* Ajax page loads

### 4.1.3 Sprint 3

Including additional background functions and documentation/presentation.

* Ajax Function Calls
* Uploading Images direct in DB
* Streaming Images direct from DB
* Restructuring Database
* Writing of Documentation
* Making Presentation

## 4.2 Pair Programming

We used pair programming in our project in difficult coding parts. This enables the equality of knowledge oft he product and leads to a safer, better eloborated product.   
By using Pair Programming we saved much time and shared our knowledge about programming-languages together with the way of programming of every individual.  
Also we ensured that both developers know nearly every part of the project this way. This also prevented a specialization of the developers e.g one knows only the backend part, the other only the frontend part.

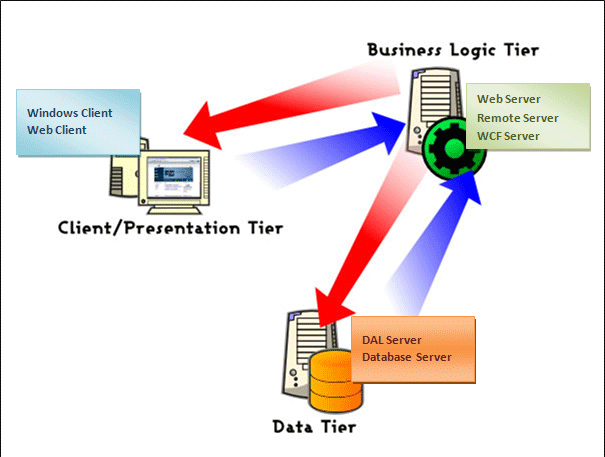
### 4.3 Github

Github is a free „gitserver“ we used to synchronize our work in not pair programming parts.  
Git is a versioning tool which saves each changes after every commit. For better usage of github we used „tortoise git“ in addition tot hat. „Tortoise git“ is a clientsided git tool what allowes to use git by GUI and has no need for console commands.

# 5. The Project itself

## 5.1 Server architecture

As Server architechture we use a classic 3-layer model. This means that first of all, the client sends a request to the webserver, the webserver has now two options – 1. It sends a answere tot he client 2. It requests the database server and answeres the client afterwards.



## 5.2 Application architecture

Express.js is perfectly working in a model-view architecture, because of its native folder structure and the envisaged way of using functions.

The model-view architecture consists of three main parts:

* The controller: which is the backend part
* The view: which are the frontend parts
* The model: which stores attributes of an object in the database



### 5.2.1 Model-View Controller

In our project we use following model-view controller (at the moment)

* Login window
* Registration window
* Dashboard tab
* Upload songs tab
* Settings tab
* Admin user menu

In further developement every tab will be a model view controller.

## 5.3 Pages

In our current state of developement we have following pages

* Landing-page
  + Registration
  + Login
  + Info video
  + Explanation of what a netlabel is
  + Explanation how to work with the site
* Dashboard
  + Viewing state of uploaded albums
  + Viewing state of Balance of Netlabel „bank-account“
  + Viewing Incompleted Uploads
  + Viewing Account stat
* Profil
  + No function at the moment
* Upload Songs
  + Upload album cover
  + Upload album information
  + Upload tracks
  + Upload track information
  + Set the Payment options
  + View Uploaded Albums
* Statistic
  + View your sale amount and their revenue
  + View your „abstract of account“ (all banking transactions with the netlabel)
* Settings
  + Define your Contact information
  + Define your way you want to get payout
  + Define your preferences and passwords

# 6. Future Aspects (what needs to be done)

* User Interface
  + Design of landingpage
* Admin Interface
  + „Request“ (to handle tickets of users)
  + „Settings“ (to fast switch some options)
  + „Language“ (to make the page languages easily editable)
* Reading operations
  + Dashboard: Uploaded Albums
  + Dashboard: Incompleted Uploads
  + Upload: Uploaded Albums
  + Statistics: Selling amounts
  + Statistics: Albums
  + Statistics: abstract of account
* Post functions
  + Admin: User requests
  + Admin: language
  + Admin: settings
* Verification of inputs
  + Uploaded
* Security
  + Crypt passwords
  + Send registration Mail
* Language
  + Make all Words readable from db
  + Make db for german
  + Make db for english
* Implementation
  + Payment System
    - Paypal
    - CreditCard
    - Bank transaction
  + Ftp
  + Kontor Api
    - Converting to xml
  + Create Bills
  + Soundfile Detection
    - Detect if soundfile is really a wav file
* Endfinish
  + Color
  + Fonts
  + Logos

# 7. What we’ve learned

## 7.1 Bootstrap

We were totally happy with bootstrap, the great library of predefined css classes is amazing. It was so simple to use and made much effort with less work. Through the standartized responsive layout we had nearly no need to adjust the page look on smartphones or tablets.

## 7.2 Node.js / Express.js

Working with Node is very unorganized and confusing, but it’s very fast and has modules for nearly everyone. By the use of Express.js everything is clear. It made node intuitive and simple. The only negative thing in between this collaboration are the callback functions, which need to be stacked many times in each other, that the asynchronous power of node can be released.

## 7.3 Stylus

For every change in the bootstrap layout or for own defined classes we used stylus. Stylus made coding faster but we are concearned about server performance, because of rerendering the stylus document on every page access.

## 7.4 Jade

Jade was nice to use to avoid the bracket chaos of html, but like in stylus it must be rerendered all time. Jade was naturally built with an „block“ operator, which makes stacking blocks into each other easy. But every time you use the block operator – the page will be reloaded completely. Because of this we used ajax calls to modernize the page and minimize the server load.

## 7.5 MongoDB

MongoDB is very simple to use, but if an error occures you can’t rely on their documentation. The big community won’t help you in every case, but responding questions very fast.

## 7.5 Other Technologies

Were all well known and doesn’t offer any noteworthy expirience.