

Dokumentation IPA Wochenplaner

IPA in Applikationsentwicklung

Auszubildender - Marius Küng

Auftraggeber - allink.creative GmbH

Projektleiter - Silvan Spross

Chefexperte - Hans Riesenmann

Experte - Antonio Di Luzio

Durchführungsort - allink.creative GmbH

Informatikmittelschule Basel

13.03. - 26.03.2012

Inhaltsverzeichnis

I. Umfeld und Ablauf	4
1. Aufgabenstellung	5
1.1. Titel der Facharbeit	5
1.2. Thematik	5
1.3. Klassierung	5
1.4. Durchführungsblock	5
1.5. Ausgangslage	6
1.6. Detaillierte Aufgabenstellung	6
1.7. Mittel und Methoden	6
1.8. Vorkenntnisse	7
1.9. Vorarbeiten	7
1.10. Neue Lerninhalte	7
1.11. Arbeiten in den letzten 6 Monaten	7
2. Einleitung	8
2.1. Projektorganisation	8
2.2. Vorkenntnisse	8
2.3. Vorarbeiten	9
2.4. Firmenstandards	9
II. Projekt	11
3. Projektbeschreibung	12
3.1. Umfeld	12
3.2. Präzisierung der Aufgabenstellung	12
3.3. Analyse Wochenplaner (IST-Zustand)	12
3.3.1. Modellierung	12
3.3.2. Darstellung	13
3.3.3. Funktionsumfang	13

3.3.4. Defizite	13
3.4. Definition Ziele	14
3.4.1. MUSS-Ziele	14
3.4.2. KANN-Ziele	15
4. Realisierung	16
4.1. ERM's	16
4.1.1. allink.planer	16
4.1.2. Wochenplaner	17
4.2. Design	18
4.2.1. Altes Design	18
4.2.2. Neues Design	19
4.3. Umsetzung	19
4.4. Resultat	20
4.5. Modellierung (Django)	21
4.6. Layout	21
4.7. Entwicklungsumgebung	21
4.8. Herausforderungen & Entscheidungen	21
4.8.1. JSON	21
4.8.2. Queries für Wochenansicht	22
4.8.3. JS Funktionalität (CRUD mit jQueryUI & AJAX)	22
4.8.4. Sperrtage	22
4.8.5. Task duplizieren per jQueryUI	22
4.9. Semantische Codebehebungen	23
4.10. Implementierung	23
5. Testing	24
5.1. Testfälle erfassen	24
5.2. Testmethoden	24
5.2.1. Erfolge Testing	37
5.2.2. Field-Test	38
6. Konklusion	39
7. Literaturverzeichnis	40
III. Arbeitsjournal	41
8. Arbeitsjournal	42
8.1. 13.03.2012	42

8.2. 14.03.2012	42
8.3. 15.03.2012	43
8.4. 16.03.2012	43
8.5. 19.03.2012	44
8.6. 20.03.2012	45
8.7. 21.03.2012	45
8.8. 22.03.2012	45
8.9. 23.03.2012	46
9. Abbildungsverzeichnis	47
10. Tabellenverzeichnis	48

Teil I.

Umfeld und Ablauf

1. Aufgabenstellung

1.1. Titel der Facharbeit

Webapplikation zur Ressourcenplanung von allink.creative

1.2. Thematik

Es soll eine Webapplikation mit Django erstellt werden, mit welcher die Geschäftsleitung die Ressourcenplanung der Mitarbeiter vornehmen kann. Damit soll eine ältere Webapplikation abgelöst werden.

1.3. Klassierung

- Applikationsentwicklung OO
- UNIX / Linux
- andere Programmiersprache

1.4. Durchführungsblock

Startblock 1: 12.03.2012 - 23.04.2012

IPA-Durchführung: 12.03.2012 - 26.03.2012

Einreichung bis: Montag, 30.01.2012

1.5. Ausgangslage

Bei allink besteht seit Mitte 2010 ein rudimentäres Ressourcenplanungstool. Zur Entwicklung wurden damals jedoch Technologien verwendet, die heute nicht mehr zur Kernkompetenz von allink zählen. Da dieses Tool jedoch jeden Freitag zur Planung der nächsten Woche verwendet wird, ist es seit längerem überfällig es in die bestehende Managementapplikation zu integrieren.

Dank der übermässig langen Testphase des Prototypen sind nun die Anforderungen an das definitive Tool gut bekannt. Daher soll eine Webapplikation mit Django erstellt werden, mit welcher die Geschäftsleitung von allink.creative die Ressourcenplanung der Mitarbeiter vornehmen kann. Damit soll eine ältere Webapplikation abgelöst werden.

1.6. Detaillierte Aufgabenstellung

Das bestehende Tool namens “Yatplaner” soll als Modul im bestehenden Management Tool namens “allink.planer” reimplementiert werden. Dabei soll die Bedienbarkeit verbessert werden. Das Ziel ist es das neue Tool so intuitiv bedienen zu können, dass für die Geschäftsleitung keine Schulung nötig ist. Der Praxistest wird voraussichtlich in der letzten IPA Woche an der Wochenplansitzung durchgeführt.

Nebst der begleitenden IPA Dokumentation, wo unter anderem der Funktionsumfang des bestehenden Tools analysiert wird, wird keine zusätzliche Dokumentation gefordert. Der Funktionsumfang des bestehenden Tools soll vom Lernenden in einer Analysephase aufgenommen werden. Dabei sollen die bestehenden Features als Muss- und mögliche neue Features als Kann-Ziele ausformuliert werden.

Der Quellcode des bestehenden Tools ist unter folgender Adresse einsehbar:

<https://github.com/sspross/yatplaner/tree/rails>

1.7. Mittel und Methoden

Folgende Technologien sind zwingend zu verwenden:

- Python 2.6
- Django 1.3
- Piston 2.3
- jQuery 1.8

- HTML5
- CSS3

Das Tool soll in folgenden Browsern fehlerfrei funktionieren:

- Firefox ≥ 8
- Chrome ≥ 10
- Safari ≥ 5

Der Internet Explorer muss explizit nicht unterstützt werden.

1.8. Vorkenntnisse

Dem Lernenden sind alle genannten Technologien bereits bekannt. Seit Beginn seines Praktikums im August 2011 setzt er sich damit auseinander. Gewisse Kombinationen wie z.B. mit jQuery einen AJAX Request zu erstellen, sind jedoch Neuland.

1.9. Vorarbeiten

Es findet keine explizite Vorarbeit statt.

1.10. Neue Lerninhalte

Wie bereits erwähnt sind dem Lernenden alle Technologien bereits bekannt. Jedoch sind gewisse Kombinationen noch nie vom Lernenden selbst angewandt worden. Das Know-how ist bei allink ausreichend vorhanden. Der Lernende kann zudem auf eine Vielzahl von bestehenden Projekten zurückgreifen, wo er unzählige Beispiele studieren kann.

1.11. Arbeiten in den letzten 6 Monaten

Der Lernende hat überwiegend Webseiten mit den oben genannten Technologien umgesetzt. Zu seinen umfassendsten Arbeiten zählen bis jetzt eine Webseite eines Immobilienunternehmens und einer Eventplattform eines Finanzkonzerns. Für ersteres arbeitete der Lernende rund 250 Stunden daran.

2. Einleitung

2.1. Projekorganisation

Projekorganisation

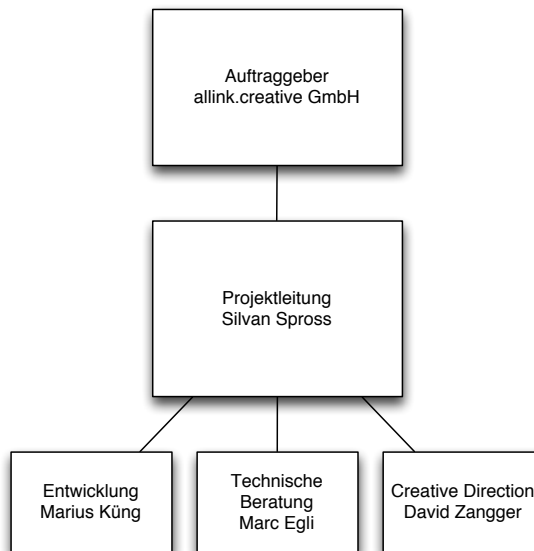


Abbildung 2.1.: Projekorganisation¹

2.2. Vorkenntnisse

Technologien

- Python Grundkenntnisse
- Django Fortgeschrittene Kenntnisse

¹Eigene Darstellung

- Piston Grundkenntnisse
- jQuery Fortgeschrittene Kenntnisse
- HTML5 Gute Kenntnisse
- CSS3 Gute Kenntnisse
- AJAX Fortgeschrittene Kenntnisse

Anwendungen

- Mehrere komplette Webauftritte realisiert
- Applikationen in Django erstellt (News, Blog, Produkteübersicht)
- Schnittstellen programmiert (XML, JSON)
- Per AJAX dynamische Inhalte laden und einfügen
- Dynamische Formulare abschicken und Request entgegennehmen
- DOM-Elemente manipulieren, per Events steuern

2.3. Vorarbeiten

Ich habe, um einen ersten Eindruck über die Funktionalität zu erhalten, den bestehenden yatplaner ausprobiert. Ansonsten fand keine explizite Vorarbeit statt.

2.4. Firmenstandards

- Betriebssystem: Mac OS X
- Editor: Textmate
- Entwicklungsumgebung: Python, Django (Python-Framework für Webapplikationen)
- Virtuelle Testumgebung: Django Serversimulation (per Terminal steuerbar)
- Deployment: per fabric-script auf Apache-Server mit WSGI-Protokoll
- Versionierung: Git, auf Github gehostet

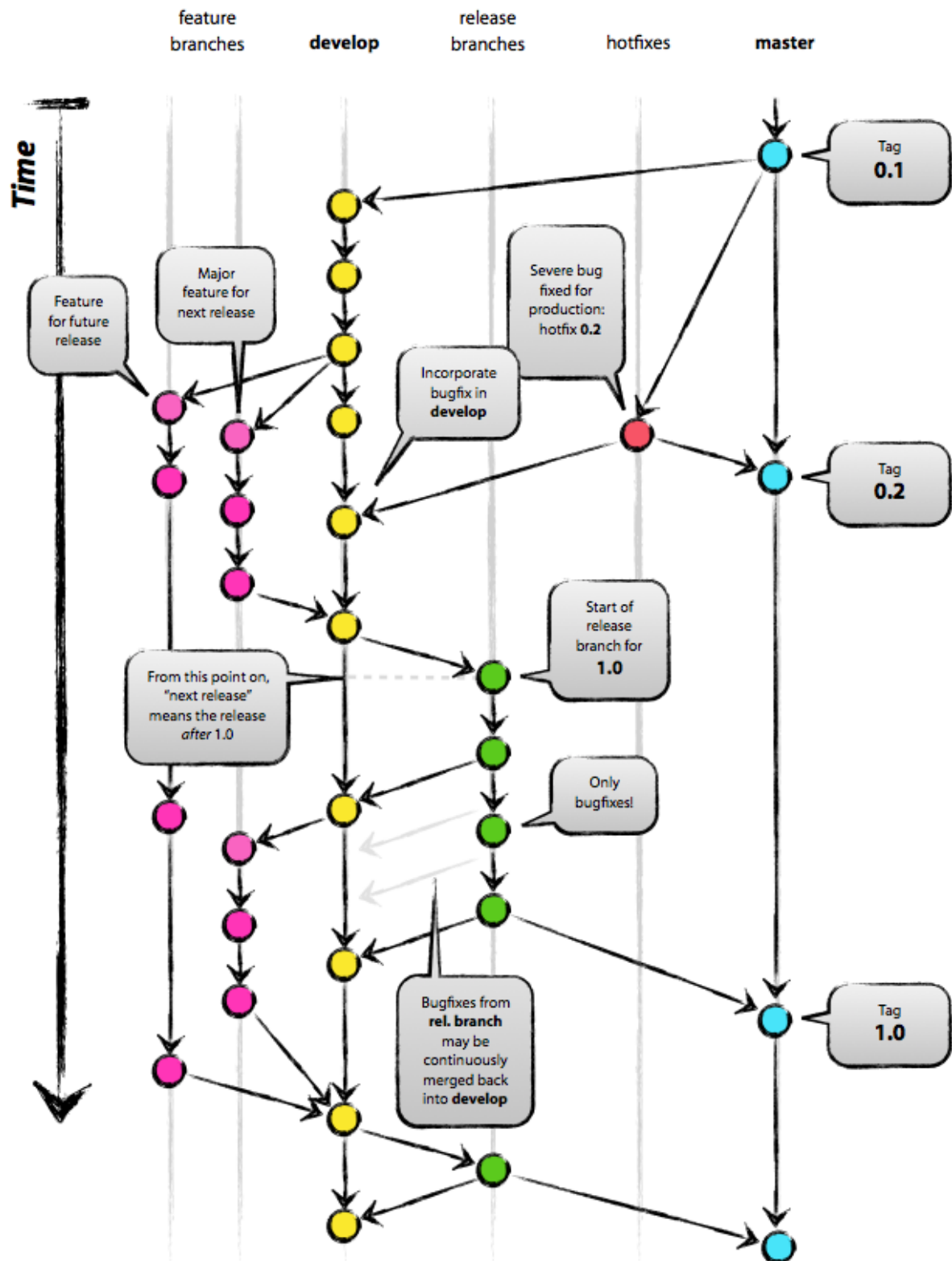


Abbildung 2.2.: A successful Git branching model von Vincent Driessen

Teil II.

Projekt

3. Projektbeschreibung

3.1. Umfeld

Momentan existiert der Wochenplaner als eigenständige Applikation. Das Tool fungiert mehr als eine Tafel auf die Post-It's geklebt werden. D. h. die Aufgaben sind nicht direkt im Projektmanagementtool mit der Person verbunden. Somit bietet das Tool keinen Mehrwert bei der Auswertung des Projektablaufs.

Das Tool ist in Ruby on Rails programmiert, was nicht dem Firmenstandard entspricht, und kann deshalb nur mühsam gewartet und erweitert werden.

3.2. Präzisierung der Aufgabenstellung

Die Aufgabenstellung sieht vor, dass die mindestens bestehende Funktionalität vorhanden und die intuitive Bedienung verbessert wird. Zur weiteren Präzisierung, werde ich eine detaillierte IST-Analyse vornehmen. Diese wird mit dem Auftraggeber besprochen, damit die MUSS- und KANN-Ziele erfasst werden können und ersichtlich wird, welche Funktionalitäten überhaupt noch Sinn machen und was weggelassen werden kann.

3.3. Analyse Wochenplaner (IST-Zustand)

3.3.1. Modellierung

Analysiert anhand der Rails-Modelle

Person

- Name
- Locked Days (gesperrte Tage, Auswahl von Montag bis Freitag)
- Beschreibungstext für jeden gesperrten Tag

3.3. Analyse Wochenplaner (IST-Zustand)

Task

- Name
- Datum (Fälligkeitsdatum, DATE)
- Person (Fremdschlüssel)
- Dauer (Stundenanzahl)

3.3.2. Darstellung

Der Wochenplaner verfügt über mehrere Darstellungen:

- Kalenderansicht der aktuellen Woche (Screenshot)
- Übersicht aller Angestellten
- Monatsansicht einer Person
- Übersicht aller erfassten Tasks

3.3.3. Funktionsumfang

- Neue Task durch Doppelklick auf Personentag per Dialog erfassen (Screenshot)
- Klick auf Task öffnet die Detailansicht (um bspw. Änderungen vorzunehmen)
- Verschiebung einer Task per Drag & Drop
 - Task in anderen Tag verschieben
 - Task einer anderen Person zuweisen
- In der Kalenderwoche per Link vor- oder zurückspringen
- weitere 7 Tage in der Ansicht anzeigen (funktioniert auch umgekehrt)
- Wochentage können für wiederkehrende Events (Schule, Frei) markiert werden

3.3.4. Defizite

- Die Sperrtage haben kein Startdatum und sind für jeden Wochentag (Vergangenheit & Zukunft) erfasst
- Die Ajax-Funktionalität beschränkt sich auf Task erstellen und verschieben
- Die Bearbeitung der Tasks ist mühsam

3.4. Definition Ziele

In einer Sitzung mit der Projektleitung wurde der IST-Zustand besprochen und die Ziele definiert.

3.4.1. MUSS-Ziele

Nr	Funktion	Beschreibung
1	Task erfassen	Der Benutzer kann eine neue Task erfassen.
2	Task Namen geben	Der Benutzer kann einem Task einen Namen geben.
3	Task Datum geben	Der Benutzer kann einem Task ein Datum geben.
4	Task Person zuweisen	Der Benutzer kann einem Task eine Person zuweisen.
5	Task Dauer geben	Der Benutzer kann einem Task eine Dauer (in Stunden) geben.
6	Task bearbeiten	Der Benutzer kann einen Task bearbeiten.
7	Task löschen	Der Benutzer kann einen Task löschen.
8	Person hinzufügen	Der Benutzer kann eine Person dem Wochenplaner hinzufügen.
9	Sperntag Person zuweisen	Der Benutzer kann einer Person einen Sperntag zuweisen.
10	Sperntag Namen geben	Der Benutzer kann einem Sperntag einen Namen geben.
11	Sperntag bearbeiten	Der Benutzer kann einen Sperntag bearbeiten.
12	Sperntag löschen	Der Benutzer kann einen Sperntag löschen.
13	Task einem Projekt zuweisen	Der Benutzer kann einen Task einem Projekt zuweisen.
14	Person als Partner	Eine Person als Partner kennzeichnen.

Tabelle 3.1.: Zwingend umzusetzende Funktionen des Prototypen

3.4.2. KANN-Ziele

Nr	Funktion	Beschreibung	Priorität
15	Task duplizieren	Der Benutzer kann einen Task duplizieren.	1
16	Sperrtag Startdatum geben	Der Benutzer kann einem Sperrtag ein Start- & Enddatum geben.	3
17	Warnmeldung Max	Es erscheint eine Warnmeldung wenn die Summe aller Arbeitsstunden pro Tag 8h überschreiten.	4
18	Tasks sortieren	Tasks können innerhalb eines Tages sortiert werden.	2

Tabelle 3.2.: Nicht zwingend umzusetzende Funktionen des Prototypen

4. Realisierung

4.1. ERMs

4.1.1. allink.planer

Als Ausgangslage dient mir das ERM des allink.planer. Die Entitäten Person und Project werden mit dem Wochenplaner verknüpft.

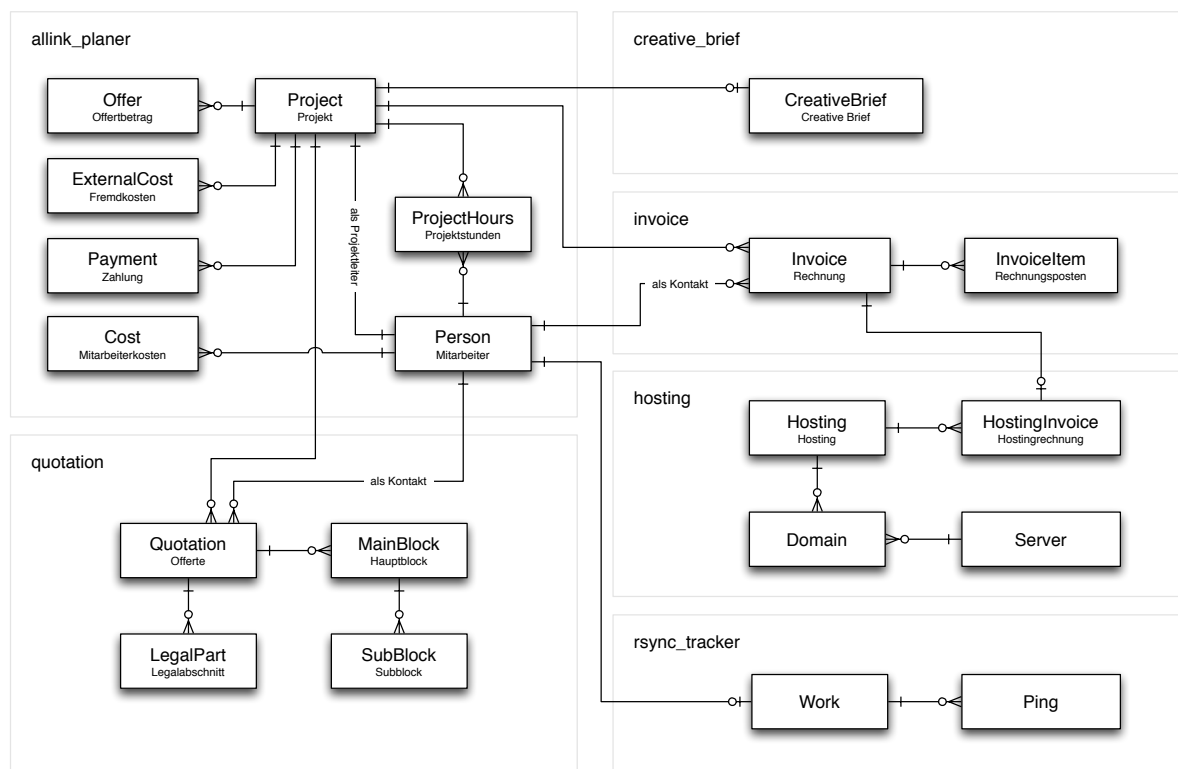


Abbildung 4.1.: ERM allink.planer¹

¹Entommen aus allink.planer

4.1.2. Wochenplaner

Wochenplaner

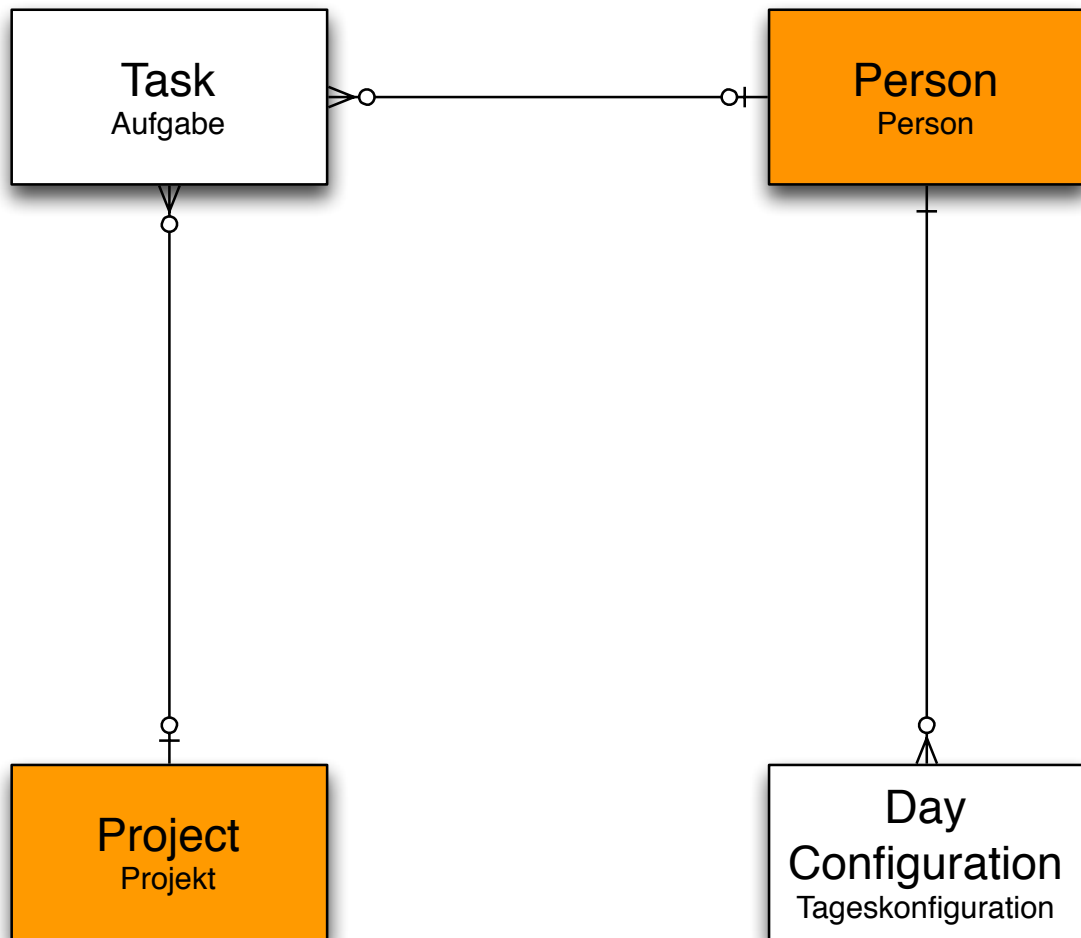


Abbildung 4.2.: ERM Wochenplaner (die Orange hinterlegten Entitäten stammen aus dem allink.planer)²

²Eigene Darstellung

4.2. Design

4.2. Design

4.2.1. Altes Design

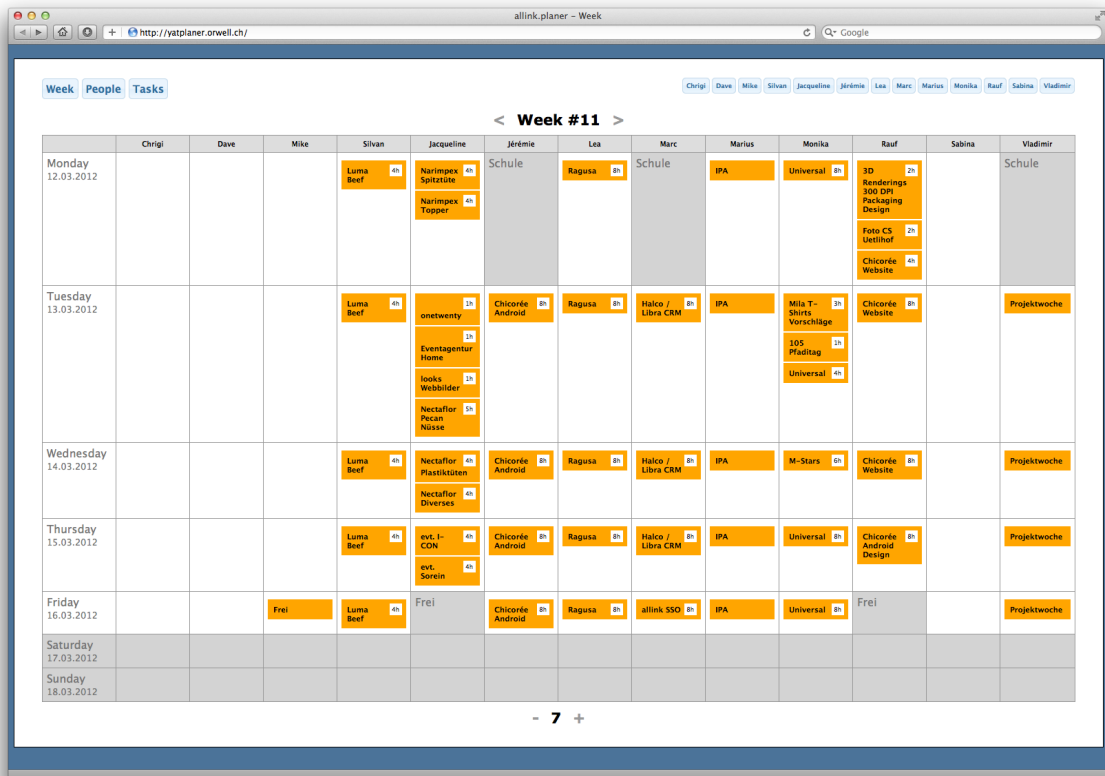


Abbildung 4.3.: Bisheriges Design des yatplaner

4.3. Umsetzung

4.2.2. Neues Design

	Christi	Dave	Mike	Silvan	Jacqueline	Jérémie	Lea	Marc	Marius	Monika	Rauf	Sabina	Vladimir
Monday 05.03.2012				Luma Beef 4h	JAM Anzeigen 1h Jam Kalender 1h Nectafloor Limited Topper 6h		Luma Beef 8h		Looks Website 8h	Telekurs Bilder 1h Luma Beef 1h Ragusa Bilder 1h	I AM Banner 1h Jam Entry Clip 2h Jobsch 1h	Conci 1h Kraft Kalender 1h	
Tuesday 06.03.2012				Luma Beef 4h	Limited Topper / film parkshot 4h Nectafloor Sticker Indonesien 2h Sorein Spenderflasche 2h	Orion News 4h Chicorée Android 4h	Sorein AD 2h Ragusa 8h	105 Apps 4h Halco / Libra CRM 4h	Looks Website 8h	Looks Flyer 8h	Chicorée Online 8h		Conci Seite 8h
Wednesday 07.03.2012				Luma Beef 4h	Sorein Spenderflasche 4h IAPN Roll-up 2h Nectafloor Spintain 1h	Chicorée Android 8h	Ragusa 8h	Halco / Libra CRM 8h	Looks Website 8h	M-Stars 4h Universal Nossal? 4h	Chicorée Online 8h		Conci Seite 8h
Thursday 08.03.2012				Zumstag 4h		Chicorée Android 8h	Ragusa 8h	Halco / Libra CRM 8h	Looks Website 8h	Universal Nossal? 8h	Chicorée Online 8h		Conci Seite 8h
Friday 09.03.2012				Zumstag 4h		Chicorée Android 8h	Ragusa 8h	allink SSO 8h	IPA Vorbereitung 8h	M-Stars 4h Camen 4h			Conci Seite 8h

Abbildung 4.4.: Neues Design

4.3. Umsetzung

Da ich nicht auf die gesamte Entstehung des Quellcodes eingehen kann und der Verlauf im git-repository einsehbar ist, führe ich die Umsetzung der Ziele auf.

Der Wochenplaner besteht aus 2 Bereichen in denen Daten manipuliert werden können:

- Administratorbereich
- Wochenansicht

Obwohl eine intuitive Bedienung gefordert ist, müssen nicht alle Ziele in der Wochenansicht manipulierbar sein.

Die folgende Aufstellung zeigt auf welche Ziele wie umgesetzt wurden.

Ziel-Nr (MUSS & KANN)	Admin	Wochenansicht
1	x	x
2	x	x
3	x	x
4	x	x
5	x	x
6	x	x
7	x	o
8	x	o
9	x	o
10	x	o
11	x	o
12	x	x
13	x	x
14	x	o
15	x	o
16	o	x
17	x	x
18	o	x
19	o	x

Tabelle 4.2.: Funktionsbereiche der Ziele

4.4. Resultat

Das Resultat meiner IPA ist einen Wochenplaner in das Projektmanagementtool allink.planer zu integrieren. Dieser Wochenplaner ist über eine HTML-Ansicht dynamisch bearbeitbar. Die

4.5. Modellierung (Django)

Tasks sind mit Personen und Projekten verknüpft. Dadurch können Analysen gemacht werden.

4.5. Modellierung (Django)

Über das Django-Framework kann man die Modelle der Klassen bzw. Entitäten erfassen. Alle Modelle können als Applikationen über einen Administratorbereich durch CRUD manipuliert werden.

4.6. Layout

Das Layout bleibt, bis auf einige kleine Anpassungen, dasselbe:

Im neuen Layout hat die Woche nur noch 5 anstatt 7 Tage und die Links zum admin-Bereich wurden versetzt.

4.7. Entwicklungsumgebung

Die Entwicklungsumgebung besteht aus den in den Firmenstandards genannten Technologien.

4.8. Herausforderungen & Entscheidungen

4.8.1. JSON

Um dynamisch die Datenbank zu manipulieren, kann eine Methode in Django verwendet werden, über die Requests im JSON-Format an den Server geschickt werden. Über den django-piston Handler können diese Requests entgegengenommen und weitergeleitet werden.

Bis jetzt habe ich lediglich über einen JSON-Handler Daten aus der Datenbank gelesen oder neue Datensätze erstellt.

Einen bestehenden Datensatz zu ändern und diesen dann korrekt im Frontend wieder einzusetzen, ist ein bisschen knifflig aber im Nachhinein doch sehr logisch zu verstehen. Durch eine Einführung von Silvan Spross konnte ich diesen Prozess verstehen und nachvollziehen.

4.8.2. Queries für Wochenansicht

Bisher habe ich alle Objekte die ich in der Wochenansicht benötigt habe über die view übergeben. Im Template habe ich über die Objekte iteriert und die entsprechenden Daten ausgegeben. Dies ist jedoch nicht 'schön'. Darum habe ich mit Unterstützung von Marc Egli, der mir die Verschachtlung von Tuples zeigte, die Objekte mit den korrekten Queries verschachtelt übergeben und dann im Template aufgelöst.

4.8.3. JS Funktionalität (CRUD mit jQueryUI & AJAX)

Die Wochenansicht der Tasks ist eine rein statische Ansicht. Alle Tasks und Sperrtage können über den Django admin per CRUD manipuliert werden. Dies ist aber nur ein Teil der bestehenden Funktionalität. Es wird eine CRUD-Funktionalität im Frontend benötigt. Diese werde ich mit jQueryUI realisieren da ich mit jQuery am meisten Erfahrung habe. Sich in Alternativen einzuarbeiten benötigt zu viel Zeit und ist nicht mit jQuery kompatibel. Per jQueryUI können alle Task freis verschoben werden zu jedem Tag und/oder Person. Per AJAX werden neue Tasks automatisch in die DB geschrieben und danach ins HTML eingefügt.

4.8.4. Sperrtage

Zuerst habe ich Sperrtage nur mit einem Tag versehen, später noch mit einem Startdatum. Wenn man aber nachvollziehen will wer, wie lange, wann gefehlt hat, darf man einen Sperrtag nicht einfach überschreiben. Also sind die Sperrtage auch mit einem Enddatum versehen. In der Abfrage für die Wochenansicht wird für jeden Tag geprüft, ob in dem ein Sperrtag beginnt, endet oder kein Enddatum angegeben ist. Somit müssen keine Sperrtage gelöscht werden und es sind mehrere erfassbar, die sich automatisch ergänzen.

4.8.5. Task duplizieren per jQueryUI

Wie man eine Task per AJAX dupliziert ist auf verschiedenen Weisen möglich.

Zuerst habe ich es über ein sogenanntes Cloning versucht in dem eine Kopie der Task erstellt wird und dann als neue Task eingesetzt werden kann. Da ich wusste, dass dieses Feature umfangreich wird, habe ich es in einen separaten branch ausgelagert. Jedoch stiess ich dabei auf grosse Probleme in der Umsetzung, da das cloning zusätzlich mit einer gedrückten Tastenkombination erfolgen sollte. Nach einer Besprechung mit der Projektleitung wurde beschlossen, dass die Task über einen Button dupliziert wird und unten eingesetzt wird.

4.9. Semantische Codebehebungen

Am Ende des Testings habe ich zur Erhöhung der Codequalität eine Codesäuberung mit pep8, pyflakes und jshint vorgenommen. Dabei wird falsche Syntax und nicht verwendete Imports hervorgehoben.

4.10. Implementierung

Der Wochenplaner kann als App mit Abhängigkeiten (Person, Projekt, piston, jQueryUI) in den allink.planer direkt integriert werden.

5. Testing

5.1. Testfälle erfassen

Die Testfälle beziehen sich auf die Ausführbarkeit der MUSS- und KANN-Ziele. Einige Ziele sind jeweils im Admin und in der Wochenansicht ausführbar. Ziele die auch in der Wochenansicht vorhanden sind, werde ich nicht im Admin testen, da es dieselben Prozesse sind.

5.2. Testmethoden

Ich werde die definierten Testfälle durchführen und mit Bildern dokumentieren ob wie sie ausgeführt werden. Auf den Bildern ist (sofern vorhanden) der REST-Call aufgelistet, der die Kommunikation mit dem Server darstellt.

⁰Eigene Darstellung

Nr	Ziel-Funktion	Testfrage
1	1	Kann der Benutzer eine neue Task erfassen?
2	2	Kann der Benutzer einer Task einen Namen geben?
3	3	Kann der Benutzer einer Task ein Datum geben?
4	4	Kann der Benutzer einer Task eine Person zuweisen?
5	5	Kann der Benutzer einer Task eine Dauer geben?
6	6	Kann der Benutzer eine Task bearbeiten?
7	7	Kann der Benutzer eine Task löschen?
8	8	Kann der Benutzer einer Person einen Sperrtag zuweisen?
9	9	Kann der Benutzer einem Sperrtag einen Namen geben?
10	10	Kann der Benutzer einen Sperrtag bearbeiten?
11	11	Kann der Benutzer einen Sperrtag löschen?
12	12	Kann der Benutzer eine Task einem Projekt zuweisen?
13	13	Kann der Benutzer eine Person zum Wochenplaner hinzufügen?
14	14	Kann der Benutzer eine Person als Partner kennzeichnen?
15	15	Kann der Benutzer eine Task duplizieren?
16	16	Kann der Benutzer einem Sperrtag ein Start- & Enddatum geben?
17	17	Erscheint eine Warnmeldung wenn Taskdauer grösser als 8h
18	18	Kann der Benutzer Tasks in einem Tag sortieren?

Tabelle 5.1.: Zu erfüllende Funktionen des Prototypen

5.2. Testmethoden

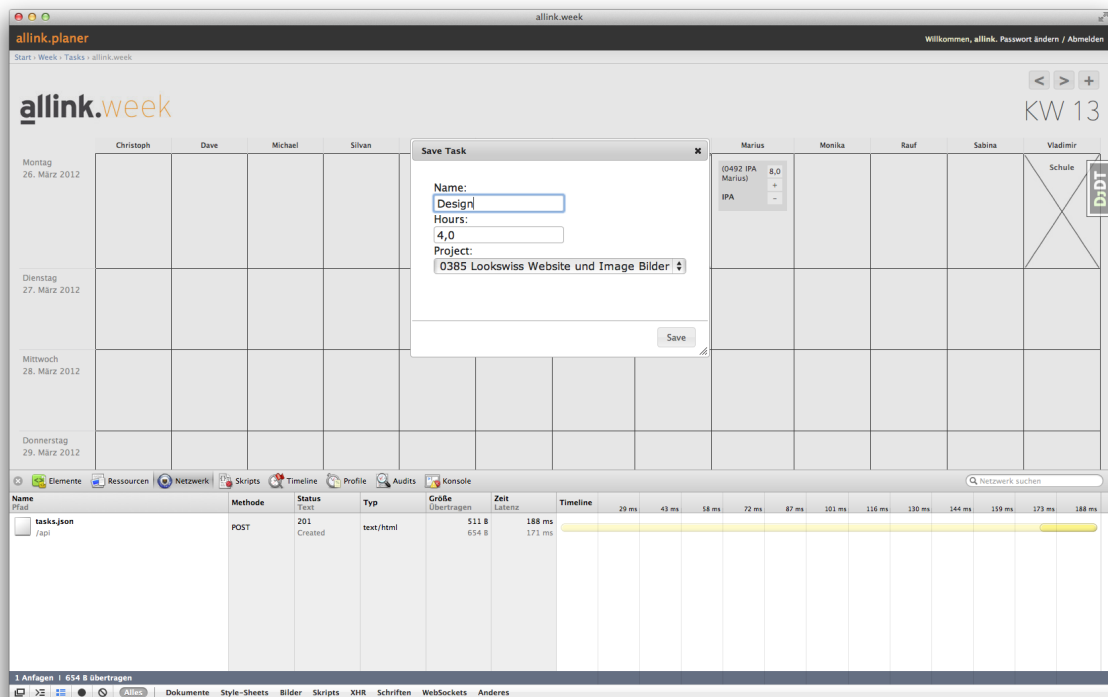


Abbildung 5.1.: Nr. 1, 2, 3, 4, 5, 12

5.2. Testmethoden

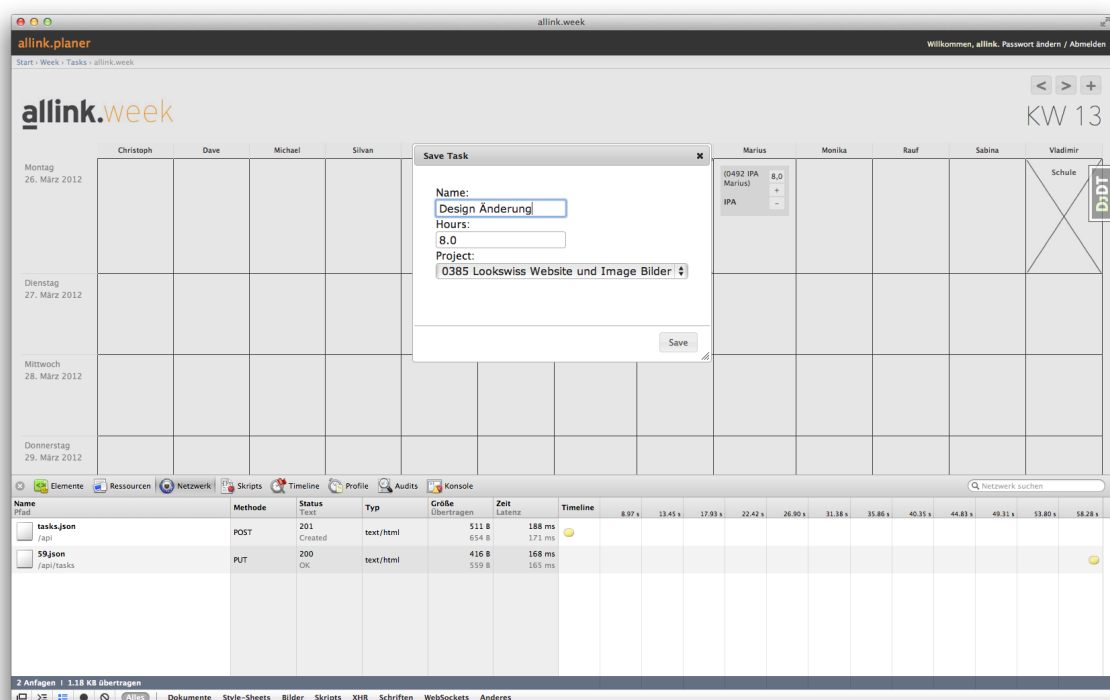


Abbildung 5.2.: Nr. 6

5.2. Testmethoden

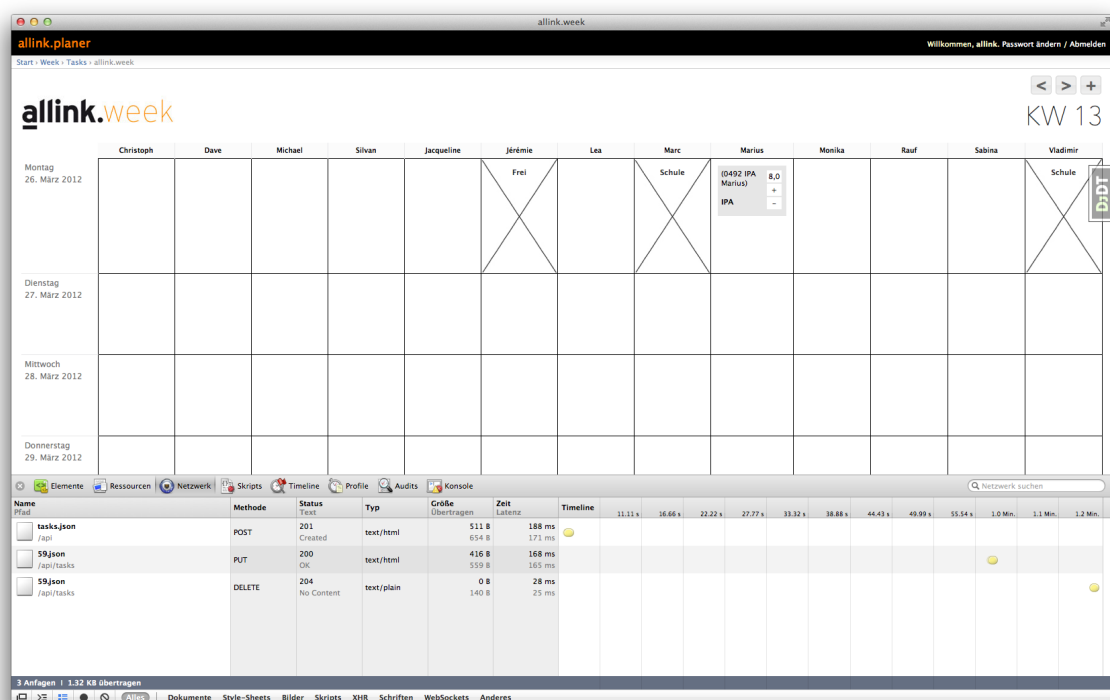


Abbildung 5.3.: Nr. 7

5.2. Testmethoden

The screenshot shows the 'allink.planer' web application interface. The title bar indicates 'Person ändern | allink.planer'. The top right corner has a 'Willkommen, allink. Passwort ändern / Abmelden' link and a 'Geschichte' button. The main content area is titled 'Person ändern' and includes a 'Geschichte' button and a 'DBT' logo. The form contains the following fields and sections:

- ☐ Partner
- ☐ Projektleiter
- ☐ Freelancer
- E-Mail:
- Phone:
- ☒ im Wochenplaner
- Basecamp id: S048306
- Firstname: Marc
- Lastname: Egli
- Last basecamp update: 15. März 2012 11:53:25

Below the form is a table for 'Relative Mitarbeiterkosten' with columns 'Valid from', 'Amount', and 'Löschen?'. Below that is a table for 'Tageskonfigurationen' with columns 'Name', 'Tag', 'Startdatum', 'Enddatum', and 'Löschen?'. The 'Tageskonfigurationen' table has two rows: one for 'Schule' on 'monday' starting '13.02.2012' and ending 'Heute', and another row with empty fields. At the bottom, there are buttons for 'Löschen', 'Sichern und neu hinzufügen', 'Sichern und weiter bearbeiten', and 'Sichern'.

Abbildung 5.4.: Nr. 8, 9

	Jérémie	Lea	Marc
	<div>Frei</div>		<div>Schule</div>

Abbildung 5.5.: Nr. 8, 9

The screenshot shows a web browser window with the title 'Person ändern | allink.planer'. The browser's address bar shows 'Start > Allink.planer > Personen > Marc Egli'. The page header includes 'Willkommen, allink. Passwort ändern / Abmelden' and a 'Geschichte' button. The main content area is titled 'Person ändern' and contains the following form fields:

- ☐ Partner
- ☐ Projektleiter
- ☐ Freelancer
- E-Mail:
- Phone:
- ☒ Im Wochenplaner
- Basecamp id: 5048306
- Firstname: Marc
- Lastname: Egli
- Last basecamp update: 23. März 2012 14:42:56

Below the form fields are two tables:

Effektive Mitarbeiterkosten			
Valid from	Amount	Löschen?	
+ Effektive Mitarbeiterkosten hinzufügen			

Tageskonfigurationen				
Name	Tag	Startdatum	Enddatum	Löschen?
Frei	monday	26.03.2012 Heute	23.04.2012 Heute	
+ Tageskonfiguration hinzufügen				

At the bottom of the form, there is a red 'Löschen' button and three buttons: 'Sichern und neu hinzufügen', 'Sichern und weiter bearbeiten', and 'Sichern'.

Abbildung 5.6.: Nr. 10

The screenshot displays the 'allink.planer' web application interface. The main heading is 'Person ändern | allink.planer'. The user is logged in as 'Willkommen, allink. Passwort ändern / Abmelden'. The breadcrumb trail is 'Start > Allink.planer > Personen > Silvan Spross'. The 'Person ändern' form includes the following fields and options:

- ☒ Partner
- ☒ Projektleiter
- ☐ Freelancer
- E-Mail:
- Phone:
- ☒ Im Wochenplaner
- Basecamp id: 4859111
- Firstname: Silvan
- Lastname: Spross
- Last basecamp update: 15. März 2012 11:34:46

Below the form, there are two tables:

Effektive Mitarbeiterkosten		
Valid from	Amount	Löschen?
+ Effektive Mitarbeiterkosten hinzufügen		

Tageskonfigurationen				
Name	Tag	Startdatum	Enddatum	Löschen?
<input type="text"/>	<input type="text"/>	<input type="text"/> Heute <input type="text"/>	<input type="text"/> Heute <input type="text"/>	<input type="text"/>
+ Tageskonfiguration hinzufügen				

At the bottom left, there is a red 'Löschen' button. At the bottom right, there are three buttons: 'Sichern und neu hinzufügen', 'Sichern und weiter bearbeiten', and 'Sichern'.

Abbildung 5.7.: Nr. 11

5.2. Testmethoden

allink.planer

Person ändern | allink.planer

Start > Allink.planer > Personen > Silvan Spross

Willkommen, allink. Passwort ändern / Abmelden

Person ändern

☒ Partner

☒ Projektleiter

☐ Freelancer

E-Mail:

Phone:

☒ im Wochenplaner

Basecamp id: 4859111

Firstname: Silvan

Lastname: Spross

Last basecamp update: 15. März 2012 11:34:46

Effektive Mitarbeiterkosten		
Valid from	Amount	Löschen?
+ Effektive Mitarbeiterkosten hinzufügen		

Tageskonfigurationen				
Name	Tag	Startdatum	Enddatum	Löschen?
<input type="text"/>	<input type="text"/>	<input type="text"/> Heute <input type="text"/>	<input type="text"/> Heute <input type="text"/>	<input type="text"/>
+ Tageskonfiguration hinzufügen				

☒ Löschen

Sichern und neu hinzufügen | Sichern und weiter bearbeiten | Sichern

Abbildung 5.8.: Nr. 13, 14

5.2. Testmethoden

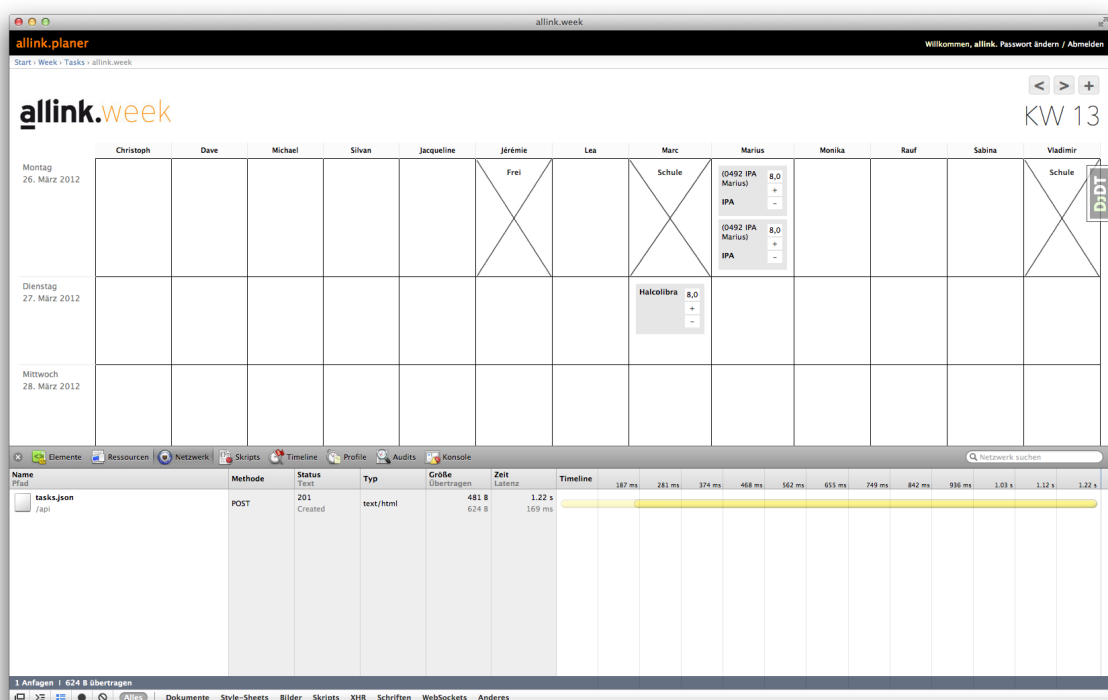


Abbildung 5.9.: Nr. 15

5.2. Testmethoden

The screenshot shows the 'allink.planer' web application interface. The browser tabs indicate the current page is 'Person ändern | allink.planer'. The page header includes the logo 'allink.planer' and a user greeting 'Willkommen, allink. Passwort ändern / Abmelden'. A success message at the top states: 'Person "Marc Egli" wurde erfolgreich geändert. Das Element kann jetzt erneut bearbeitet werden.' Below this is the 'Person ändern' form. It features radio buttons for 'Partner', 'Projektleiter', and 'Freelancer'. There are input fields for 'E-Mail:' and 'Phone:'. A checked checkbox 'In Wochenplaner' is present. The form also displays 'Basecamp id: 5048306', 'Firstname: Marc', and 'Lastname: Egli'. The 'Last basecamp update:' is dated '23. März 2012 15:37:41'. Below the form is a table for 'Effektive Mitarbeiterkosten' with columns for 'Valid from', 'Amount', and 'Löschen?'. At the bottom, there is a table for 'Tageskonfigurationen' with columns for 'Name', 'Tag', 'Startdatum', 'Enddatum', and 'Löschen?'. The 'Tageskonfigurationen' table has two rows: one for 'Schule' on 'monday' from '19.03.2012' to '23.07.2012', and another row with empty fields. At the very bottom, there are buttons for 'Löschen', 'Sichern und neu hinzufügen', 'Sichern und weiter bearbeiten', and 'Sichern'.

allink.planer

Start: allink.planer / Personen / Marc Egli

Person "Marc Egli" wurde erfolgreich geändert. Das Element kann jetzt erneut bearbeitet werden.

Person ändern

☐ Partner

☐ Projektleiter

☐ Freelancer

E-Mail:

Phone:

☒ In Wochenplaner

Basecamp id: 5048306

Firstname: Marc

Lastname: Egli

Last basecamp update: 23. März 2012 15:37:41

Valid from	Amount	Löschen?
Effektive Mitarbeiterkosten hinzufügen		

Name	Tag	Startdatum	Enddatum	Löschen?
Schule	monday	19.03.2012	23.07.2012	<input type="checkbox"/>

Tageskonfiguration hinzufügen

Abbildung 5.10.: Nr. 16

5.2. Testmethoden

The screenshot shows a web browser window with the address bar displaying 'allink.week'. The page title is 'allink.planner'. The navigation bar includes 'Start', 'Week', 'Tasks', and 'allink.week'. The main content area displays a weekly calendar for 'KW 13' (Week 13). The calendar is organized by days of the week (Montag to Freitag) and team members (Christoph to Vladimir). Tasks are entered into the calendar cells. A red task box for 'Test 0385: Lookwiss Website und Image Bilder' is highlighted on Monday, March 26, 2012, for Marius, with a duration of 9h. Other tasks include 'Schule' for Jérémie, Lea, and Vladimir, and 'Frei' (Free) for Jacqueline and Rauf.

	Christoph	Dave	Michael	Silvan	Jacqueline	Jérémie	Lea	Marc	Marius	Monika	Rauf	Sabina	Vladimir
Montag 26. März 2012						Schule		Schule	Test 0385: Lookwiss Website und Image Bilder 9h				Schule
Dienstag 27. März 2012													
Mittwoch 28. März 2012													
Donnerstag 29. März 2012													
Freitag 30. März 2012					Frei						Frei		

Abbildung 5.11.: Nr. 17

5.2. Testmethoden

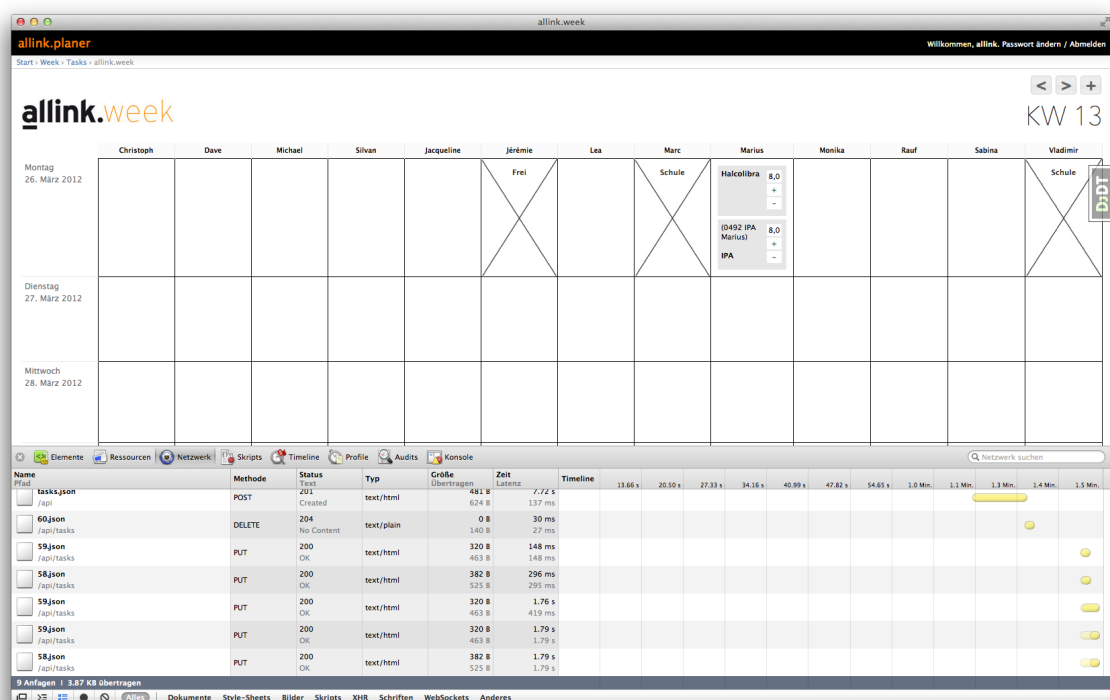


Abbildung 5.12.: Nr. 18

5.2.1. Erfolge Testing

Hier liste ich alle Tests auf und zeige ob sie erfolgreich waren oder nicht.

Nr	MUSS-Funktion	erfolgreich?
1	1	ja
2	2	ja
3	3	ja
4	4	ja
5	5	ja
6	6	ja
7	7	ja
8	8	ja
9	9	ja
10	10	ja
11	11	ja
12	12	ja
13	13	ja
14	14	ja
15	15	ja
16	16	ja
17	17	ja
18	18	ja

Tabelle 5.2.: Zu erfüllende Funktionen des Prototypen

⁰Eigene Darstellung

5.2.2. Field-Test

Am 24.03. wird die Geschäftsführung an ihrer wöchentlichen Planungssitzung das Tool das erste Mal verwenden. Dabei soll das Tool auf sein Intuitivität geprüft werden. Am Montag werde ich dann Feedback erhalten und anfallende Anpassungen vornehmen.

6. Konklusion

kommt noch

7. Literaturverzeichnis

Python Dokumentation:

<http://docs.python.org/>

Django Dokumentation:

<https://docs.djangoproject.com/en/1.3/>

Javascript Dokumentation:

<http://www.w3schools.com/jsref/default.asp>

jQuery Dokumentation:

http://docs.jquery.com/Main_Page

Teil III.

Arbeitsjournal

8. Arbeitsjournal

8.1. 13.03.2012

Beginn IPA yatplaner

- Dokumentation auf github hosten und versionieren
- Kapitelstruktur aufbauen Teil 1
- Zeitplan erstellen (Planung)
- Aufgabenstellung erfassen
- Projektorganisation grafisch erfassen
- Designbesprechung mit Dave
- Vorkenntnisse erfassen
- Vorarbeiten erfassen
- Firmenstandards erfassen
- Ziel erreicht: Teil 1 Der Doku erfasst Teil 2
- Umfeld erfassen
- IST-Analyse vorerfassen (Ansichten, Funktionalität)

8.2. 14.03.2012

- IST-Analyse vorerfassen (Modellierung der Rails App)
- Sitzungen mit Silvan Spross und Marc Egli zur Definition der MUSS- und KANN-Ziele
- MUSS-Ziele erfasst

- KANN-Ziele erfasst
- Planung abgeschlossen
- ERM für Wochenplaner erstellt
- allink.planer github repository gecloned
- Entwicklungsumgebung aufgesetzt
- Ziel erreicht: Anfang Teil 2 der Dokumentation erfasst und mit Realisierung begonnen
- mit Django-Modellierung begonnen (Task, DayConfiguration)

8.3. 15.03.2012

- Kapitel Realisierung weiterführen
- Django admin einrichten für week
- Testdaten (Personen) in die Entwicklungsumgebung importieren
- Problem: Die Wochentage zusammenzufassen war relativ knifflig damit man durch jeden Tag iterieren kann
- View erstellen für alle Tasks, Personen und Tage
- Template erstellt und Tabelle so aufgesetzt damit Daten korrekt eingefügt werden
- Template angefangen zu stylen (CSS)
- Ziel erreicht: Daten korrekt aus Modell laden und in Template einsetzen, Template funktioniert
- (Bin im Zeitplan mit Realisierung)

8.4. 16.03.2012

- Ansicht funktioniert für erste Versuche mit jQueryUI
- Task sind verschiebbar und sortierbar in andere Tage (ohne callback/Änderungen werden nicht gespeichert)
- alle Tasks werden zu javascript Objekten der Klasse Task
- jQuery Methoden hinzugefügt (Click-Events)

- Task-Dialog hinzugefügt um neue Task zu erfassen
- Task Objekt erweitert mit save Methode wenn eine Task gespeichert wird
- Django Task Formular
- Django Task Piston Handler (json)
- AJAX POST schicken
- piston nimmt json-POST entgegen und führt das Formular, aus welches die Daten in die Datenbank speichert
- wenn eine neue Task erstellt wurde, wird die neue task.id zurückgeschickt um das neue Javascript Task Objekt abzuspeichern
- Ziel erreicht: jQueryUI einrichten, mind. POST per Ajax möglich und dynamisches einfügen in Tabelle

8.5. 19.03.2012

- Taskobjekte der aktuellen Woche in javascript dictionary speichern
- Hilfestellung Silvan: PUT Funktionalität in Django abhandeln bei existierendem Objekt
- Verschieben von Task (PUT Funktionalität)
- Klick auf eine Task öffnet Dialog und man kann die Task bearbeiten und abspeichern
- Bugfix: wenn man eine Task verschoben wird, öffnet es danach den Bearbeiten-Dialog. Per deaktivieren des click events kann dies behoben werden.
- Bugfix: man konnte eine neu erstellte Task nicht verschieben
- Projekte werden dynamisch in den Dialog eingefügt und können ausgewählt werden
- Task einem Projekt zuweisen
- Ziel: PUT Funktionalität erstellen für Task verschieben und ändern
- Dokumentation weiter führen
- Besprechung mit Silvan Stand IPA

8.6. 20.03.2012

- Umsetzung MUSS-Ziele erreicht
- Dokumentation weiter geschrieben
- feature task duplizieren ausprobiert (Kann-Ziel)
- Bugfix: Taskzelle ist zu wenig hoch um problemlos neue Task in Zelle zu verschieben
- Bugfix: Bei Bearbeitung einer Task wird aktuelles Projekt nicht mitgeschickt

8.7. 21.03.2012

- Sperrtag bild ausgewechselt
- Bugfix: wenn man im Template die Woche wechselt kann das Datumsformat nicht verarbeitet werden
- Queries optimieren mit Marc, verschachtelte Tuples für queries
- Dokumentation weiter geschrieben
- Kleinere Optimierungen
- Cloning weiter versuchen

8.8. 22.03.2012

- Fehlerbehebungen im Template: Kalenderwoche wird nicht aktualisiert
- KANN-Ziel Tasks sortieren
- KANN-Ziel Warnung wenn Tasks in Tag mehr als 8h Stunden betragen
- KANN-Ziel Einer Tageskonfiguration ein Startdatum setzen
- Bugfix wenn neue Task erstellt wurde konnte sie nicht direkt bearbeitet werden.
- Besprechung Sperrtage habe ein Start- und Enddatum und können von weiteren Sperrtagen ersetzt ohne aus der DB gelöscht zu werden
- Mit Unterstützung von Marc die beste Lösung finden die Sperrtage auszulesen (Bei jedem Tag abfragen)
- Bugfix man kann nicht weniger als 1 Woche anzeigen

- Bugfix wenn man die Tagesanzahl erhöht wird aktuelles Datum nicht berücksichtigt
- Besprechung wie das Task-cloning am besten implementiert wird

8.9. 23.03.2012

- Task kann per Klick dupliziert werden (es wird eine neue mit gleichem Inhalt erzeugt und unten eingefügt)
- Task kann per Klick gelöscht werden
- Doku schreiben
- Mit Silvan Wochenplaner angeschaut und Darstellungsanpassungen vorgenommen
- Abnahme Wochenplaner Silvan

9. Abbildungsverzeichnis

2.1. Projekorganisation	8
2.2. A successful Git branching model von Vincent Driessen	10
4.1.	16
4.2.	17
4.3. Bisheriges Design des yatplaner	18
4.4. Neues Design	19
5.1. Nr. 1, 2, 3, 4, 5, 12	26
5.2. Nr. 6	27
5.3. Nr. 7	28
5.4. Nr. 8, 9	29
5.5. Nr. 8, 9	29
5.6. Nr. 10	30
5.7. Nr. 11	31
5.8. Nr. 13, 14	32
5.9. Nr. 15	33
5.10. Nr. 16	34
5.11. Nr. 17	35
5.12. Nr. 18	36

10. Tabellenverzeichnis

3.1. Zwingend umzusetzende Funktionen des Prototypen	14
3.2. Nicht zwingend umzusetzende Funktionen des Prototypen	15
4.2. Funktionsbereiche der Ziele	20
5.1. Zu erfüllende Funktionen des Prototypen	25
5.2. Zu erfüllende Funktionen des Prototypen	37