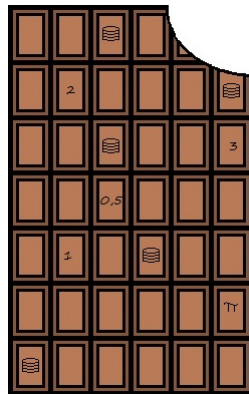


iFeeD

Interaktives Feedbacksystem zur Datenanalyse

Felix Bening, Erik Borker, Joshua Brutscher, Nico Denner, Robert Krause

03-02-2019



Inhaltsverzeichnis

1	Einleitung	4
2	Änderungen Pflichtenheft	5
3	Änderungen Entwurf	5
3.1	Backend	5
3.1.1	Erweiterbarkeit durch neue Klassen	5
3.1.2	Mechanismus zum Manipulieren der Daten	6
3.1.3	Bestimmung des Rangs	7
3.1.4	Attribute der Klasse Session	7
3.1.5	Neue Gestaltung der OcalAPI-Anfrage	8
3.2	Frontend	9
3.2.1	Implementierung des Interface OnInit	9
3.2.2	Zusätzliche Liste in der AdminMainOverviewComponent	9
3.2.3	Zusätzliche Methoden zum Erstellen der Setup/Session Informa- tionen	9
3.2.4	Auslagerung der clone, delete, download Methoden	10
3.2.5	Setupeditierung und Beendigung der Setuperstellung	10
3.2.6	Dialogfenster zur Setup- und Sessionlöschung	11
3.2.7	Setup Konfigurationsansicht	11
3.2.8	Einschränkung der Funktionalität der Serivces	11
3.2.9	Download der Labelung in einer Session	12
3.2.10	Hinzufügen der Component DeletionDialog	12
3.2.11	Auslagerung des Fortschrittsbalkens	13
3.2.12	Hinzufügen des MixedGuard	13
3.2.13	Hinzufügen der DisplayList Klasse	14
3.2.14	Entfernung der Methode getIteration	14

3.2.15	Rückgabewert compareLabeledData	14
3.2.16	Parameterliste und Rückgabewert von compareSessions	15
3.2.17	Methode für Cohens-Kappa-Wert	15
3.2.18	Parameterliste für deleteSession	15
3.2.19	Parameteränderung AdminUserManagement	16
3.2.20	createUser() initiiert nur Routing	16
3.2.21	openUser() in UserManagmenet entfernt	17
3.2.22	checkUsername() entfernt	17
3.2.23	Parameteränderung checkPassword()	17
3.2.24	Parameteränderung createSetup()	18
3.2.25	Änderung in den Routes	18
3.2.26	Parameteränderung in CSVtoJSON	18
3.2.27	Methode in JSONHandlerService entfernt	19
3.2.28	Darstellung subspaces geändert	19
3.2.29	Singleton EnumService	19
3.2.30	Mehr Methoden für PlotService	20
4	Implementierung der Wunsch- und Kann-Kriterien	21
5	Übersicht Bibliotheken und Frameworks	22
5.1	Backend	22
5.2	Frontend	22
5.2.1	Angular Forms	22
5.2.2	Angular Routing	23
5.2.3	ngx-spinner	23
5.2.4	Angular Material	23
5.2.5	Plotly.js	23
6	Planung und Verlauf	24

1 Einleitung

Diese Dokumentation ist im Kontext des PSE am Lehrstuhl IPD Böhm während der Implementierungsphase entstanden. Darunter ist eine grobe Übersicht über die Änderungen bezüglich des Pflichtenhefts und des Entwurfs, außerdem eine Vorstellung der verwendeten Frameworks und der implementierten Tests. Im letzten Kapitel findet sich die Planung der Phase als Gantt Diagramm und die im Laufe der Phase aufgetretenen Veränderungen wieder.

2 Änderungen Pflichtenheft

- Veränderung von F13 Anzeigen von Statistiken zu einem Setup:

Das System zeigt nur an ob eine Session gestartet, beendet oder in Bearbeitung ist. Ist keine Anzahl an maximalen Iterationen eingestellt, so wird nach Start der Session ausschließlich der Feedbackmodus angezeigt.

3 Änderungen Entwurf

3.1 Backend

3.1.1 Erweiterbarkeit durch neue Klassen

Änderungen

Die Klassen QueryStrategies und Classifier wurden hinzugefügt. Dafür wurden das Enum QueryStrategies entfernt.

Die verweisenden Attribute wurden dementsprechend aktualisiert.

Ein Skript Setup.py wurde erstellt mit welchem sich die Defaultobjekte, welche zuvor als Enum vorlagen, generieren lassen.

Gewonnene Vorzüge

Durch das Transformieren zu Klassen ist die Erweiterbarkeit erhöht. Dem System können leichter neue QueryStrategien und Classifier hinzugefügt werden. Außerdem können die dazugehörigen Parameter dynamischer verändert werden. Nicht jedes Setup hat nun die gleichen Parameter sondern diese sind von dem Classifier abhängig.

3.1.2 Mechanismus zum Manipulieren der Daten

Ursprüngliche Planung

Sollen, durch eine REST-Anfrage, Daten im Backend manipuliert werden so sollte der Request mit der Methode `get(r: Request):Response` ausgewertet und durchgeführt werden.

Änderungen

Während der Implementierung sind wir auf ein eleganteres Verfahren gestoßen. Dieses fordert pro models-Klasse eine neue Klasse welche von `RetrieveUpdateDestroyAPIView` erbt. Diese Klassen fordern die Attribute

- `serializer_class`
Klasse des Serializers
- `lookup_field`
Attribut über welches das zu manipulierende Objekt angesprochen wird
- `queryset`
Liste der Objekte aus welcher das zu manipulierende Objekt stammt.

Für jede dieser Klassen wurde ein eigener URL-Pfad definiert über welchen man das Objekt anfordert. Das REST-Framework erlaubt uns nun über diesen Pfad

- `delete-`
- `put- und`
- `get-Requests`

durchzuführen.

Als Gegenzug wurde die get-Methode entfernt.

Gewonnene Vorzüge

Dieses Verfahren sorgt für lesbareren und übersichtlicheren Code. Außerdem erspart es, durch noch intensivere Nutzung des Frameworks, Implementierungsarbeit. Des weiteren erhöht es die Erweiterbarkeit da nicht für jede neue Datentabelle in der Datenbank eine eigene get-Methode geschrieben werden muss sondern diese durch Erben direkt vorliegen.

3.1.3 Bestimmung des Rangs

Ursprüngliche Planung

Um den Rang einer Person zu bestimmen wurde ein Rang vom REST-Server ausgeliefert.

Änderungen

Dieser Funktionalität wurde entfernt.

Begründung

Dadurch, dass der Rang schon durch die Implementierung in zwei unterschiedlichen Klassen festgelegt ist, ist dieses Implementierungsdetail redundant.

3.1.4 Attribute der Klasse Session

Änderungen

Attribut labelHistory wurde entfernt. Attribute labels und history wurden hinzugefügt. Die Deklaration des Attributes finalLabels wurde spezifiziert.

```

labels = ArrayField(models.CharField(
    choices=[(tag.value['label'], tag.name)
              for tag in Labels], max_length=30))
finalLabels = ArrayField(ArrayField(models.CharField(
    choices=[(tag.value['finalLabel'], tag.name)
              for tag in Labels], max_length=30)))
history = ArrayField(ArrayField(models.IntegerField()))

```

Gewonnene Vorzüge

Die Anfrage an die OcalAPI lässt sich schneller generieren. Im Frontend entstehen keine Nachteile.

3.1.5 Neue Gestaltung der OcalAPI-Anfrage

Änderungen

Anstelle die Anfrage über ein Skript in der url.py laufen zu lassen wurde hierfür ein eigener Serializer (OcalAPISerializer) erstellt. Dieser Serialisiert das zu Anfrage gehörende Session-Objekt und hat als Attribut das Ergebnis der OcalAPI-Anfrage. Dementsprechend wurde in der view.py eine Klasse (OcalAPI) erstellt welche das JSON Objekt ausgeliefert. Dieses wird wiederum in der url.py eingebunden.

Gewonnene Vorzüge

Die Anfrage ist ausgelagert. Das führt zu lesbarerem und übersichtlicherem Code.

3.2 Frontend

3.2.1 Implementierung des Interface OnInit

Änderungen

Components implementieren das Interface OnInit respektive die Methode ngOnInit.

Gewonnene Vorzüge

Die Methode ngOnInit wird ausgeführt sobald auf die entsprechende Component navigiert wird. Somit eignet sie sich sehr gut um Daten (beispielsweise vom REST Server) zu laden, sowie benötigte Attribute zu initialisieren.

3.2.2 Zusätzliche Liste in der AdminMainOverviewComponent

Änderungen

Attribut userList sowie die Methode openUser.

Gewonnene Vorzüge

Zusätzlich zu den Setups, Sessions und Datensätzen wird eine Liste der zuletzt hinzugefügten Persons angezeigt.

3.2.3 Zusätzliche Methoden zum Erstellen der Setup/Session Informationen

Änderungen

Methoden zum Erstellen der Setup/Session Informationen wurden in den entsprechenden Components hinzugefügt.

Gewonnene Vorzüge

Um die direkte Abhängigkeit von Controller (CalcService mit beispielsweise der Methode calcSessionStatus) zur View (html der AdminMainOverviewComponent) zu vermeiden, wurde eine Methode im Model hinzugefügt (hier: createSessionStatusInfo). Diese geht einerseits mit den berechneten Werten des Services um und bereitet diese für die Anzeige in der View vor.

3.2.4 Auslagerung der clone, delete, download Methoden

Änderungen

Auslagerung der clone, delete, download Methoden von der AdminSetupOverviewComponent in die AdminSetupDetailComponent.

Gewonnene Vorzüge

Verminderung der Komplexität in der AdminSetupOverviewComponent.

3.2.5 Setupeditierung und Beendigung der Setuperstellung

Änderungen

Methoden editSetup und finishedCreation wurden in der AdminSetupOverviewComponent hinzugefügt.

Gewonnene Vorzüge

Der Admin ist in der Lage eine noch nicht beendete Setuperstellung erneut zu verändern. Durch Klick auf den finish creation Button ist die Erstellung final beendet.

3.2.6 Dialogfenster zur Setup- und Sessionlöschung

Änderungen

Dialogfenster bei delete Methode in der AdminSetupDetailComponent.

Gewonnene Vorzüge

Mittels des Dialogfensters wird abgefragt, ob der Löschvorgang einer Session beziehungsweise des Setups wirklich durchgeführt werden soll. Dem Admin wird somit die Möglichkeit gegeben, den Vorgang abubrechen und zuerst ein Download der Ergebnisse durchzuführen.

3.2.7 Setup Konfigurationsansicht

Änderungen

Methode openSetupConfig wurde in der AdminSetupOverviewComponent hinzugefügt.

Gewonnene Vorzüge

Ermöglicht es die eingestellten Parameter des ausgewählten Setups anzusehen, ohne sie erneut ändern zu können.

3.2.8 Einschränkung der Funktionalität der Services

Änderungen

Services wie der RESTService fragen die Daten vom Server an und leiten sie mittels einer asynchronen Anfrage an die jeweilige Component weiter.

Begründung

Implementierungstechnisch ist es nicht möglich in den Services REST Anfragen zu stellen und diese direkt zu verarbeiten. Das liegt daran, dass die Anfragen asynchron sind und der Service nicht wartet bis er die nötigen Daten erhalten hat.

3.2.9 Download der Labelung in einer Session

Änderungen

Methode für den ausschließlichen Export der Labels wurde zum ExportService hinzugefügt.

Gewonnene Vorzüge

Ermöglicht es dem Admin ausschließlich die Labelung zu betrachten, wenn die restlichen Informationen zu der Session nicht benötigt werden.

3.2.10 Hinzufügen der Component DeletionDialog

Änderungen

Neue Component, die ein Dialogfenster für das Löschen von Daten aller Art modelliert.

Gewonnene Vorzüge

Der Admin muss den Löschvorgang bestätigen, somit wird versehentliches Löschen verhindert.

3.2.11 Auslagerung des Fortschrittsbalkens

Änderungen

Der Fortschrittsbalken der zur Visualisierung vergangener Heatmaps verwendet wird, wurde in eine separate Komponente ausgelagert.

Gewonnene Vorzüge

Durch das Auslagern werden die Komponenten, die diesen verwenden, übersichtlicher und weniger überladen. Außerdem wird die Funktionalität des Balkens in seiner eigenen Klasse gekapselt, was die Wiederverwendbarkeit erhöht. Die übergeordneten Komponenten können über die definierte Schnittstelle mit dem Balken kommunizieren.

3.2.12 Hinzufügen des MixedGuard

Änderungen

Ein weiterer Guard wurde hinzugefügt. Dieser ist dazu da um den Zugriff auf bestimmte Seiten auf angemeldete Benutzer zu beschränken.

Gewonnene Vorzüge

Durch diesen Guard wird der Zugriff auf die Sessiondetail-Seite von sowohl Admins als auch Usern ermöglicht. Gleichzeitig wird verhindert, dass dem System fremde, also nicht angemeldete, Personen auf die Seite einsehen können.

3.2.13 Hinzufügen der DisplayList Klasse

Änderung

Eine neue Klasse DisplayList wurde hinzugefügt. Diese repräsentiert die Angezeigte Liste an Rohdaten.

Gewonnene Vorzüge

Das Heraustrennen der Klasse aus der UserIterationComponent, vereinfacht die UserIterationComponent und macht sie dadurch übersichtlicher und besser wartbar. Desweiteren sind dadurch die Informationen die diese Liste hält separat gekapselt und somit auch wiederverwendbar. Dementsprechend erhöht dies die Objektorientierung wodurch die Erweiterbarkeit gefördert wird.

3.2.14 Entfernung der Methode getIteration

Änderungen

Die Methode getIteration aus SessionDetailComponent wurde entfernt. Die Methode sollte benutzt werden um die Progressbar der Heatmaps auch in der SessionDetailComponent anzeigen zu können, diese Funktionalität wurde allerdings in die neu hinzugefügte Component ProgressbarComponent ausgelagert.

Gewonnene Vorzüge

Bessere Code-Struktur, da der selbe Progressbar in zwei Components benötigt wird.

3.2.15 Rückgabewert compareLabeledData

Änderungen

Ursprünglich gab die Methode compareLabeledData des StatisticsService ein Array mit

drei Einträgen zurück. Der letzte Eintrag war dabei die Anzahl der Punkte die bei beiden Label-Arrays gleich waren. Dieser Eintrag wurde aufgeteilt in zwei Werte, die die Anzahl der inlier- bzw. outlier-Punkte die bei beiden Label-Arrays gleich waren repräsentieren. Diese Änderung wurde vorgenommen, da zur Berechnung des Cohen-Kappa-Wertes diese Unterteilung nötig war.

3.2.16 Parameterliste und Rückgabewert von compareSessions

Änderungen

Die Methode compareSessions des StatisticsService bekommt nun zwei zu vergleichende Sessions gegeben und gibt einen number Wert zurück.

Gewonnene Vorzüge

Bessere Erweiterbarkeit, da nun mehr Components Nutzen aus dieser Methode ziehen könnten und nicht nur die SetupOverviewComponent.

3.2.17 Methode für Cohens-Kappa-Wert

Änderungen

Es wurde die Methode cohensKappa zum StatisticsService hinzugefügt.

Gewonnene Vorzüge

Ein besserer Vergleichswert kann Angezeigt werden.

3.2.18 Parameterliste für deleteSession

Änderungen

Die Methode deleteSession der AdminSessionOverviewComponent bekommt nun eine Session anstatt der Session-Id übergeben.

Gewonnene Vorzüge

Dieses kann direkt in der Methode des RestService übergeben werden, somit muss nicht erst die passende Session zur Id gesucht werden.

3.2.19 Parameteränderung AdminUserManagement

Änderungen

Sämtliche Methoden bekommen nun statt dem Namen eines Objekts (Datensatz, User, etc) dessen ID.

Gewonnene Vorzüge

Hierdurch kann ohne weiteren aufwand mit dem Backend kommuniziert werden, da dieses mit den IDs arbeitet.

3.2.20 createUser() initiiert nur Routing

Änderungen

Die Methode createUser() in AdminUserManagementComponent aktiviert nun lediglich das Routing zur AdminPersonCreateComponent und hat somit keinen Rückgabewert mehr.

Gewonnene Vorzüge

Das erstellen einer Person wird hierdurch komplett in einer Komponente gebündelt und lässt sich somit leichter austauschen/erweitern.

3.2.21 openUser() in UserManagmenet entfernt

Änderungen

Die Methode openUser() in AdminUserManagementComponent wurden komplett entfernt.

Gewonnene Vorzüge

Es ist nicht notwendig sich nähere Details zu einem User anzeigen zu lassen. Alle nötigen Informationen sind in der AdminUserManagementComponent zu finden.

3.2.22 checkUsername() entfernt

Änderungen

Die Methode checkUsername() in AdminPersonCreateComponent wurde entfernt.

Gewonnene Vorzüge

Das Backend übernimmt die Überprüfung. Hierdurch ist die Konsistenz der Datenbank unabhängig vom Frontend gewahrt.

3.2.23 Parameteränderung checkPassword()

Änderungen

Die Methode checkPassword() in AdminPersonCreateComponent bekommt nun einen string als Parameter übergeben.

Gewonnene Vorzüge

Das Password kann in Echtzeit auf syntaktische Richtigkeit überprüft werden und dem User können direkt Informationen hierzu angezeigt werden.

3.2.24 Parameteränderung createSetup()

Änderungen

Die Methode createSetup() bekommt nun keine Parameter mehr übergeben, weil alle notwendigen Information in einem Attribut JSON gespeichert werden.

Gewonnene Vorzüge

Die Werte müssen nicht vor dem Hochladen nochmal zusammengetragen werden, wodurch das Erstellen beschleunigt wurde.

3.2.25 Änderung in den Routes

Änderungen

Die Route 'setup/create' wurde zu 'setup/create/:id' geändert.

Gewonnene Vorzüge

Hierdurch ist es möglich die AdminSetupCreateComponent für mehrere Zwecke zu verwenden. Man kann nun ein neues Setup erzeugen, ein bestehendes noch nicht fertiges editieren und ein bestehendes fertiges sich anzeigen lassen.

3.2.26 Parameteränderung in CSVtoJSON

Änderungen

Die Methode CSVtoJSON() in JSONHandlerService bekommt nun einen zusätzlichen string als Parameter.

Gewonnene Vorzüge

Hierdurch kann der Methode mitgeteilt werden, ob der Aufrufer eine JSON mit Elementen vom Typ string oder number haben will.

3.2.27 Methode in JSONHandlerService entfernt

Änderungen

Die Methode JSONtoCSV() in JSONHandlerService wurde entfernt

Gewonnene Vorzüge

Es bestand keine Notwendigkeit für die Methode.

3.2.28 Darstellung subspaces geändert

Änderung

Die Darstellung der subspaces in der SetupCreateComponent wurde geändert. Dadurch sind die Methoden addSubspaceField und removeSubspaceField weggefallen.

Gewonnene Vorzüge

Die Übernahme der Eingabe ist vereinfacht worden. Die Ansicht ist nun weniger komplex und übersichtlicher.

3.2.29 Singleton EnumService

Änderungen

Es wurde das Singleton EnumService hinzugefügt. Dieses enthält Enums für Historymode, Feedbackmode und Label. Die Enums werden im Backend gespeichert und durch eine REST-Anfrage an den EnumService übergeben.

Gewonnene Vorzüge

Bessere Erweiterbarkeit, da Enums nun nur an einer Stelle gespeichert werden müssen.

3.2.30 Mehr Methoden für Plotservice

Änderungen

Der PlotService hat nun die zusätzlichen Methoden makeHeatmapFromJson, visualizeRawData, generateGraph, generateImage und makeBarChart.

Gewonnene Vorzüge

Der Plotservice ist nun von mehreren Components zur Erstellung verschiedener Graphen nutzbar und nicht nur zur Generierung der Heatmaps.

4 Implementierung der Wunsch- und Kann-Kriterien

Es wurden alle Wunsch- und Kann-Kriterien erfüllt.

5 Übersicht Bibliotheken und Frameworks

5.1 Backend

Das Backend ist in Python mit dem Framework Django entstanden. Django bietet eine integrierte Datenbankverwaltung. Außerdem stehen weitere Bibliotheken zur Installation bereit. Von diesen sind folgende in iFeed integriert:

- djangorestframework und
- django-filter

5.2 Frontend

Das Frontend ist in TypeScript mit dem Framework Angular und den folgenden Bibliotheken entstanden.

5.2.1 Angular Forms

Diese Bibliothek ermöglicht es Usereingaben zu validieren.

5.2.2 Angular Routing

Diese Bibliothek wird verwendet, um zwischen den einzelnen Components zu navigieren.

5.2.3 ngx-spinner

Diese Bibliothek wird dazu verwendet, den Admin oder User über längere Ladezeiten zu informieren.

5.2.4 Angular Material

Diese Bibliothek bietet eine Vielzahl von html Komponenten zur Nutzereingabe (Dropdown Menü, Dialogfenster) welche direkt mit dem Angular Framework funktionieren und das Auslesen der Daten sehr vereinfachen.

5.2.5 Plotly.js

Plotly.js ist eine javascript Grafik-Bibliothek zur Generierung von Grafiken und Diagrammen. In iFeed wird plotly.js für die Visualisierung der Daten als Heatmap genutzt. Dabei wird die Methode `Plotly.newPlot()`, welche JSON-Dateien übergeben bekommt. Die JSON-Dateien wiederum enthalten die zu plottenden Daten zusammen mit Einstellungen für die jeweiligen Plots, wie zum Beispiel Farbe, Plot-Typ und anzuzeigender Text beim Hovern.

6 Planung und Verlauf

Auf den folgenden Seiten ist die Planung der Phase und die wirkliche Ausführung als Gantt-Diagramm aufgelistet.

Planung Implementierungsphase									
			Mo	Di	Mi	Do	Fr	Sa	So
			7	8	9	10	11	12	13
Umgebung & Framework einrichten	J. Brutscher	B							
model.py implementieren	R. Krause	B							
serializer.py implementieren	R. Krause	B							
settings.py implementieren	R. Krause	B							
views.py implementieren	J. Brutscher	B							
RESTService implementieren	E. Borker	F							
RESTService implementieren	F. Bening	F							
PersonService implementieren	N. Denner	F							
			Mo	Di	Mi	Do	Fr	Sa	So
			14	15	16	17	18	19	20
OcalAPI Kontakt implementieren	R. Krause	B							
Getter Lösung	J. Brutscher	B							
OcalAPI in views.py einbinden	J. Brutscher	B							
LoginComponent implementieren	N. Denner	F							
RESTService implementieren	E. Borker	F							
SessionOverview	E. Borker	F							
Guards implementieren	F. Bening	F							
Routes implementieren	F. Bening	F							
AdminMainOverview	N. Denner	F							
AdminSetupOverview	N. Denner	F							
			Mo	Di	Mi	Do	Fr	Sa	So
			21	22	23	24	25	26	27
Änderungen Backend formulieren	J. Brutscher	B							
Implementierungsdokument	J. Brutscher	B							
Mock Objekte Backend	J. Brutscher	B							
IterationComponent	R. Krause	F							
PlotService	E. Borker	F							
AdminSetupDetail	N. Denner	F							
AdminSetupCreate	F. Bening	F							
ExportService	N. Denner	F							
AdminPersonCreate	F. Bening	F							
AdminDatasetOverview	F. Bening	F							
AdminUserManagement	F. Bening	F							
SessionDetail	E. Borker	F							
StatisticsService	E. Borker	F							
CSS	N. Denner	F							
CalcService	N. Denner	F							
			Mo	Di	Mi	Do	Fr	Sa	So
			28	29	30	31	1	2	3
Mock Objekte Backend	J. Brutscher	B							
IterationComponent & Tests	R. Krause	F							
CSS	N. Denner	F							
Tests	F. Bening	F							
Tests	E. Borker	F							
Dokument schreiben	Alle	A							

Abweichungen

Block hinzugefügt									
Block entfernt									
			Mo	Di	Mi	Do	Fr	Sa	So
			7	8	9	10	11	12	13
Umgebung & Framework einrichten	J. Brutscher	B							
model.py implementieren	R. Krause	B							
serializer.py implementieren	R. Krause	B							
settings.py implementieren	R. Krause	B							
views.py implementieren	J. Brutscher	B							
RESTService implementieren	E. Borker	F							
RESTService implementieren	F. Bening	F							
PersonService implementieren	N. Denner	F							
			Mo	Di	Mi	Do	Fr	Sa	So
			14	15	16	17	18	19	20
OcalAPI Kontakt implementieren	R. Krause	B							
UserOverview	R. Krause	F							
Getter Lösung	J. Brutscher	B							
Endpoint Lösung	J. Brutscher	B							
OcalAPI in views.py einbinden	J. Brutscher	B							
setup.py implementieren	J. Brutscher	B							
LoginComponent implementieren	N. Denner	F							
RESTService implementieren	E. Borker	F							
SessionOverview	E. Borker	F							
Guards implementieren	F. Bening	F							
Routes implementieren	F. Bening	F							
AdminMainOverview	N. Denner	F							
AdminSetupOverview	N. Denner	F							
			Mo	Di	Mi	Do	Fr	Sa	So
			21	22	23	24	25	26	27
Änderungen Backend formulieren	J. Brutscher	B							
Implementierungsdokument	J. Brutscher	B							
Mock Objekte Backend	J. Brutscher	B							
serializer.py & model.py ergänzen	J. Brutscher	B							
IterationComponent	R. Krause	F							
ProgressbarComponent	R. Krause	F							
Displaylist	R. Krause	F							
MixedGuard	R. Krause	F							
PlotService	E. Borker	F							
AdminSetupDetail	N. Denner	F							
AdminSetupCreate	F. Bening	F							
ExportService	N. Denner	F							
AdminPersonCreate	F. Bening	F							
AdminDatasetOverview	F. Bening	F							
AdminUserManagement	F. Bening	F							
SessionDetail	E. Borker	F							
StatisticsService	E. Borker	F							
CSS	N. Denner	F							
CalcService	N. Denner	F							
			Mo	Di	Mi	Do	Fr	Sa	So

Gantt

			28	29	30	31	1	2	3
Mock Objekte Backend	J. Brutscher	B							
CSS Iteration	J. Brutscher	F							
IterationComponent & Tests	R. Krause	F							
CSS	N. Denner	F							
Tests	F. Bening	F							
Tests	E. Borker	F							
Dokument schreiben	Alle	A							
AdminSetupCreate	F. Bening	F							
Error fixen	Alle	A							