

ENTWICKLER DOKUMENTATION BTS SCHEDULINGVERFAHREN PROJEKT

4. Juni 2011

Marcel Schwarzmann

Patrick Müller

Markus Zapf

Norman Schütt

Markus Klumpp

Inhaltsverzeichnis

1	HERAUSFORDERUNGEN BEI DER VERWENDUNG VON HTML UND PHP	3
1.1	VORTEILE VON PHP	3
1.2	NACHTEILE VON PHP	4
2	GRUNDGERÜST	4
2.1	VISUELLE SCHICHT	4
2.2	LOGISCHE SCHICHT	5
2.3	ORGANISATION DES TEAMS	5
3	ABLAUF DES PROGRAMMES	6
4	VERFAHRENS IMPLEMENTIERUNG	8
5	ANHANG	10
5.1	KLASSENDIAGRAMM	10
5.2	HOMEPAGE AUFBAU	11
5.3	ABLAUFPLAN	12

1 HERAUSFORDERUNGEN BEI DER VERWENDUNG VON HTML UND PHP

Da wir mit HTML auf die Browserumgebung angewiesen sind, war einer unserer ersten Diskussionen wie wir die erzeugten Prozesse eines Benutzers speichern können. Mit HTML5 Local Storage hätte uns zwar ein reiner clientseitiger Speichermechanismus zur Verfügung gestanden, dieser ist jedoch erst seit Internet Explorer 8 in allen größeren Browsern implementiert. Außerdem wollten wir den Prozessablauf auf der Serverseite berechnen, weshalb wir uns am Ende dafür entschieden haben die Daten in einer SQL-Tabelle abzulegen.

Damit nun aber nicht jeder Benutzer die gleichen Daten zu Gesicht bekommt, ist es notwendig Benutzer auf irgendeine Art und Weise zu unterscheiden. Die offensichtliche Idee einer Benutzeranmeldung mit einem Benutzernamen und einem Passwort schied für uns aus, weil wir nicht auch noch eine entsprechende Benutzerregistrierung programmieren wollten. Wir haben uns deshalb dafür entschieden, die Speicherung der Prozesse pro Benutzer anhand eines Cookies durchzuführen, d.h. jedem Benutzer wird eine eindeutige ID zugewiesen und in einem Cookie gespeichert. Diese ID befindet sich auch in der Datenbank für die Prozesse des Benutzers. Dieses Vorgehen hat natürlich den Nachteil, dass der Benutzer Cookies aktiviert haben muss. Wir denken jedoch davon ausgehen zu können, dass die meisten Benutzer heutzutage diese Einstellung aktiviert lassen, so dass wir nicht mit Problemen aus diesem Bereich rechnen. Nichtsdestotrotz fangen wir den Fall nicht aktivierter Cookies mit einer Fehlermeldung ab, die Webseite ist dann jedoch für die Benutzer nicht funktionstüchtig.

1.1 VORTEILE VON PHP

Der größte Vorteil von PHP ist zweifelsohne dessen übergreifende Verfügbarkeit auf praktisch allen Systemen, so dass keines der Teammitglieder mit Problemen bezüglich der Mitarbeit am Projekt rechnen musste. Als freie, quelloffene und kostenlos verfügbare Software ist PHP mit verschiedenen Implementierungen auf unterschiedlichsten Systemen einsetzbar.

Aufgrund der Vielzahl von Open-Source-Projekten, die PHP einsetzen ist es sehr wahrscheinlich, dass man bei der Anpassung eines Webprojektes auf eigene Bedürfnisse schon einmal mit PHP in Kontakt gekommen ist. So war es auch bei unserem Team, wo jeder zumindest schon Einsteigererfahrungen mit PHP vorweisen konnte.

- Leicht einzusetzen
- Leicht zu erlernen

1.2 NACHTEILE VON PHP

Zu einem der größten Nachteile von PHP gehört die extreme Mischung von verschiedensten Programmiersprachen und Datenelementen. Dieses ist auch in unserem Projekt sehr gut zu sehen. So kann eine einzige PHP-Datei PHP-Code, HTML, CSS, Javascript, SQL in Strings uvm. in beliebiger Mischung enthalten. Dadurch findet keinerlei Trennung von Datenzugriff, Berechnungslogik und Anzeige statt. Außerdem kann die Mischung von verschiedensten Paradigmen wie z.B. dem eher funktionalen Ansatz Javascripts gegenüber dem imperativen Ansatz von PHP gerade für Einsteiger sehr verwirrend sein.

Weiterhin unterstützte PHP in den ersten Versionen keinen objektorientierten Ansatz, was dazu geführt hat, dass PHP heute zwar objektorientiertes Programmieren unterstützt, aber dabei ähnliche Inkonsistenzen wie C++ aufweist, d.h. viele Sprachkonstrukte wie zum Beispiel Konstruktoren wirken künstlich aufgesetzt.

Viele der von uns aufgezählten Nachteile kommen jedoch erst bei größeren Projekten oder Projekten mit hoher Benutzerzahl zum Tragen, weshalb wir uns trotzdem sehr schnell auf die Benutzung von PHP für die Lösung der Aufgabe einigen konnten.

Weitere mögliche Nachteile:

- Einladend für Sicherheitslücken (laut wikipedia 43% aller Sicherheitslücken in 2006 in PHP-Software)
- Langsam
- Dynamische Programmiersprachennachteile
- Keine Automatischen Tests möglich
- Kein Debugger vorhanden

2 GRUNDGERÜST

Wir können das Projekt im Groben in 3 Teile unterteilen einmal die Datenschicht die bei diesem Projekt eine SQL Datenbank ist, eine Visuelle Schicht in Form einer Weboberfläche diese besteht aus HTML 5, CSS 3 und Java Script und die Logische Schicht dies ist die PHP Klassen Schicht.

2.1 VISUELLE SCHICHT

Diese Schicht besteht aus 5 Webseiten. Jede dieser Seiten hat ein HTML Grundgerüst eine CSS Datei. Die Seiten Scheduler und Team haben zusätz-

zlich noch eine Java Script Datei die ein Akkordeon mit Funktionen ausstattet und eine Akkordeon CSS die dieses Akkordeon und die neu hinzugewonnen Funktionalitäten mit Visuellen Elementen Ausstattet.

An die Oberfläche haben wir folgende Ansprüche gestellt. Sie sollte zum einen Schlicht sein und ein natives Handling aufweisen. Des Weiteren wollten wir nur wenige bis gar keine Effekte und Optischen Reize einbinden die den Benutzer von der Hauptfunktion ablenken würden.

Homepage Aufbau 5.2 auf Seite 11

2.2 LOGISCHE SCHICHT

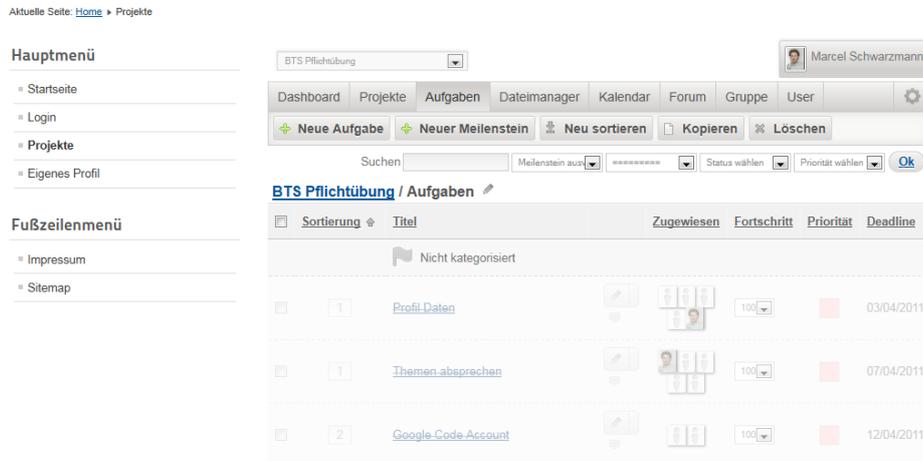
Die logische Schicht besteht aus 3 PHP Klassen und einem ADT. Es gibt einmal die Klasse DBHandler. Diese Klasse steuert das ein und auslesen der Dummy Prozesse sowie die Verwaltung der Cookies.

Die Klasse ProcessHandler verwaltet das Handling der einzelnen Prozesse und stellt einige Funktionen zur Verfügung um einige Max und Min Werte zu ermitteln. Des Weiteren werden in dieser Klasse die Prozesse gezählt und die Kennzahlen wie Laufzeit, durchschnittliche Laufzeit, Wartezeit und durchschnittliche Wartezeit ermittelt. Der Scheduler ist die letzte Klasse im Bunde. Diese enthält die Methoden für die einzelnen Scheduling Verfahren und eine Hauptmethode Verfahren die diese Methoden dann anwendet in Abhängigkeit der Auswahl die der User vorgenommen hat. Aus Gründen der OOP haben wir einen Abstrakten Datentyp Prozess erstellt der die wichtigsten Eigenschaften aufnimmt. Zu diesen Zähler der Name des Prozesses, die Ankunftszeit, die Priorität, die CPU-Laufzeit, die CPU-Restlaufzeit, der Startzeitpunkt des Prozesses und der Endzeitpunkt des Prozesses. Dieser Datentyp wird in allen 3 PHP Klassen verwendet.

Klassendiagramm 5.1 auf Seite 10

2.3 ORGANISATION DES TEAMS

Wir haben zur Organisation des Teams eine Joomla Webseite mit der Komponente Projekt Frok genutzt. Durch diese Komponente wurde uns die Überwachung der einzelnen Aufgaben erleichtert. Des Weiteren konnte jeder zu Jeder Zeit seine Aufgaben Status updaten und die Anderen wurden per Mail über dieses Update benachrichtigt so Waren immer alle auf dem neusten Stand. Ein weitere Praktischer Vorteil dieser Komponente war es einzelnen Aufgaben mehrere Teammitglieder zu weisen zu können um die Zuständigkeiten klar zu definieren. Nachteile dieser Komponente war das das mitgelieferte Dateisystem umständlich war und nach einiger Zeit die Updates einem den Eindruck des Spammens vermittelt hat. Es wurde auch ein Kalender mitgeliefert. Dieser Enthielt alle Aufgaben und Meilensteine die mit einer Deadline versehen waren.



Für die Datenbank, Webspaces hat ein Team Mitglied einen Webserver zur Verfügung gestellt der gleichzeitig dann zum Datenaustausch genutzt wurde. Leider war auf diesem kein SVCS installiert womit die Versionierung und die Sicherung der Dateien manuell von statten gehen musste. Dieser Webspaces wurde dann auch für die Tests genutzt. Das Fehlen eines Automatischen Tests waren die Tests der Daten sehr aufwändig und umständlich da man immer nach einer Neu Implementierten Methode alles was bis dato implementiert wurde nochmal testen musste. Das Fehlen des Debuggers machte es dann schwer die Fehler eindeutig identifizieren zu können.

3 ABLAUF DES PROGRAMMES

Der Erstes Schritt ist auf die Homepage zu gehen. Die Startseite hat dann gleich ein Cookie für den Benutzer angelegt.

Im Hauptmenü gibt es den Eintrag Scheduling. Diese Seite enthält dann die gesamte Logik. Die Seite ist in 3 Bereiche unterteilt Setp 1 enthält die Eingabe der Dummy Prozesse. Die Eingabe kann der Benutzer Manuell Vornehmen oder eine von dem Benutzer definierte Anzahl von Prozessen mit Zufallswerten erzeugen lassen. Die Erzeugten Prozesse kann der Benutzer dann unter der Eingabemaske sehen. Fehler beim Eingeben werden durch eine Rote Box gekennzeichnet. Des Weiteren Bietet der Bereich 1 die Möglichkeit die schon Vorhanden Prozesse zu löschen. Beim Start werden dem Benutzer auch schon Standard Prozesse angeboten mit denen der Benutzer schon Arbeiten kann. Der Step 1 ist verbunden mit der Klasse DBHandler der die Datenbank verwaltet. Die Ausgabe wird in der ProzessHandler Klasse vorbereitet.

Schritt 1: Prozesse erstellen

Prozess Dummys

Custom Add

Name: Bsp.: Prozess 1, (min. 1 max. 10 Zeichen)

CPU-Laufzeit: ms Bsp.: 25 ms, (min. 0 max. 500 ms)

Priorität: Bsp.: 15, (min. 1 max. 100)

Ankunftszeit: ms Bsp.: 50 ms, (min. 0 max. 100 ms)

Random Add

Anzahl: Prozesse Bsp.: 20 (max. 200)

Empfohlen: max. 20

#	Name	Ankunftszeit	CPU-Laufzeit	CPU-Restlaufzeit	Priorität
1	Prozess0	10 ms	10 ms	4	20
2	Prozess1	8 ms	8 ms	1	10
3	Prozess2	17 ms	17 ms	24	5
4	Prozess3	90 ms	90 ms	31	95
5	Prozess4	15 ms	15 ms	4	10
6	Prozess5	23 ms	23 ms	3	37
7	Prozess6	122 ms	122 ms	7	17
8	Prozess7	8 ms	8 ms	0	20
9	Prozess8	11 ms	11 ms	3	70
10	Prozess9	64 ms	64 ms	10	56

Im Zweiten Bereich Step 2 kann der Benutzer dann zwischen 8 Schedulingverfahren auswählen. Bis auf RoundRobin kann der Benutzer alle Verfahren ohne weitere Eingaben sofort Starten. Für RoundRobin wird dann noch ein Feld Quantum benötigt in dem der Benutzer die Zeitpanne der Prozesse angibt. Falls der Benutzer auch in diesem Bereich Fehlerhafte eingaben vornimmt wird er durch eine Fehlermeldung in einem Rot hinterlegten Kaste darauf hingewiesen. Durch das Auswählen eines Verfahrens wird die Klasse Scheduler angesteuert und dieser der ProzessHandler als Parameter übergeben. Wenn das Verfahren durchgelaufen ist und es keinen Fehler gab wird der Benutzer automatisch in den Bereich 3 Weitergeleitet.

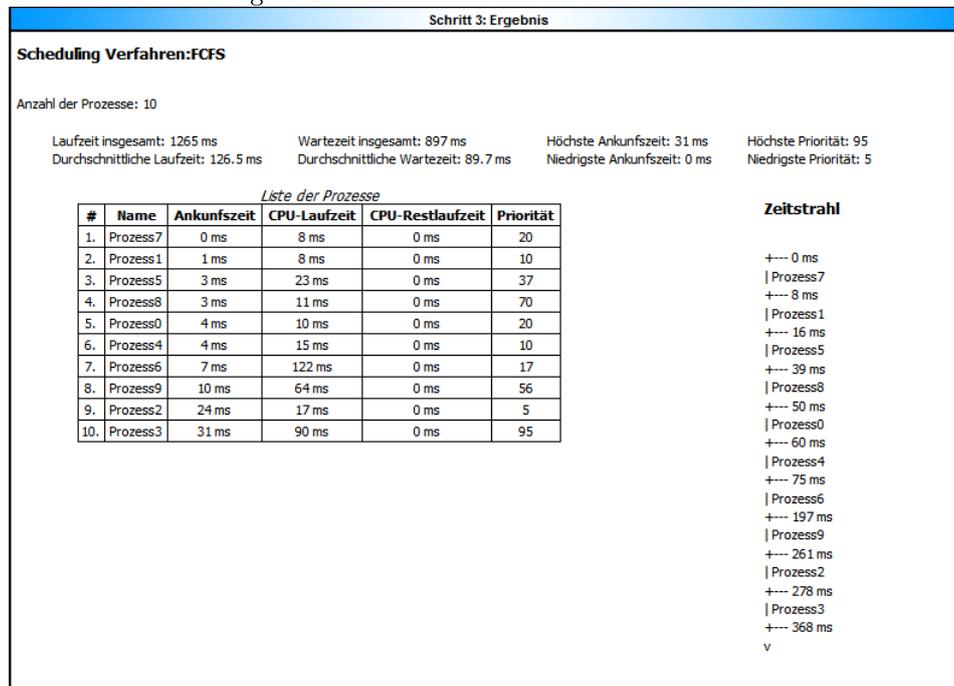
Schritt 2: Scheduling Verfahren wählen

Bitte Wählen Sie eine der Scheduling verfahren:

Quantum:

Im Bereich Step 3 wird dann das Ergebnis des Verfahrens angezeigt. Dies geschieht in Form einer Tabelle der die Aufrufe der Prozesse widerspiegelt in Zeitlich aufsteigender Reihenfolge. In der Tabelle werden dann auch alle Daten zu dem Prozess ausgegeben. Die für das Verfahren benötigten Daten sind weiter

Links zu finden und die nicht relevanten Daten auf der rechten Seite der Tabelle. Neben der Tabelle findet der Benutzer dann ein Zeitstrahl in dem die Prozesse dann nur noch Namentlich aufgeführt werden. Die Vorbereitung der Ausgabe und die dafür benötigten Daten kommen aus der ProzessHandler Klasse.



Ablaufplan 5.3 auf Seite 12

4 VERFAHRENS IMPLEMENTIERUNG

Alle Verfahren bis auf RoundRobin werden dadurch unterscheiden wie Sie den nächsten Prozess Wählen. Aus diesem Grund wurden für die Auswahl des Nächsten Prozesses separate Methoden geschrieben die dann immer den nächsten Prozess zurückliefern die dieses Verfahren nun als Nächsten Prozess sieht. Aufgrund des nur geringen Unterschiedes konnten deshalb alle Prozesse in einer Hauptmethode realisiert werden.

Diese Methode ist 3 wesentliche Bereiche unterteilt.

Der erste Bereich enthält eine Methode die in einem Zeitintervall (bei diesem Projekt jede ms) überprüft ob neue Prozesse anfragen auf erhalt der CPU gestellt haben.

Im zweiten Bereich wird dann je nach verfahren die Methode ausgewählt die den nächsten Prozess der für das jeweilige verfahren nun am besten Passt ausgewählt.

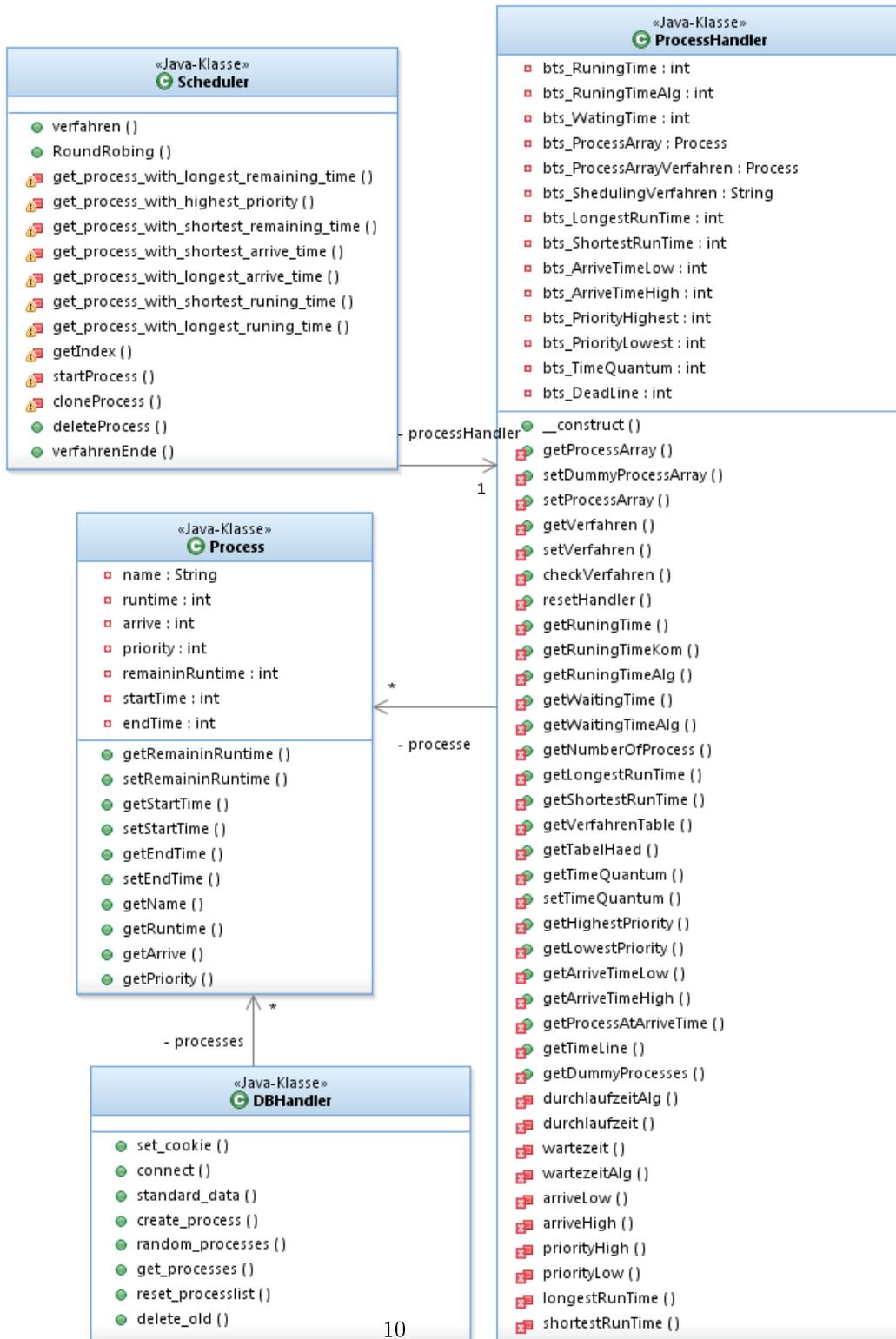
Bereich drei beschäftigt sich dann damit zu schauen ob der Prozess derzeit die

CPU Ressourcen hat oder ob es ein neuer Prozess ist. Wenn der Prozess die CPU derzeit hat kann er diese weiter nutzen. Wenn es aber ein neuer Prozess ist muss geschaut werden ob er schon einmal die CPU benutzt hat oder nicht. Wenn dieser die CPU schon einmal benutzt hat wird die Eigenschaft Restlaufzeit aus diesem Prozess gelesen und der neue Startzeitpunkt in den Prozess eingetragen. Hat der Prozess die CPU noch nicht gehabt so wird der Startzeitpunkt gesetzt und der Prozess bekommt die CPU. Der Alte Prozess bekommt dann den Wert für die Eigenschaft Endzeitpunkt eingetragen.

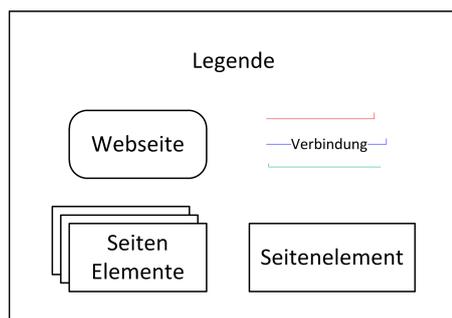
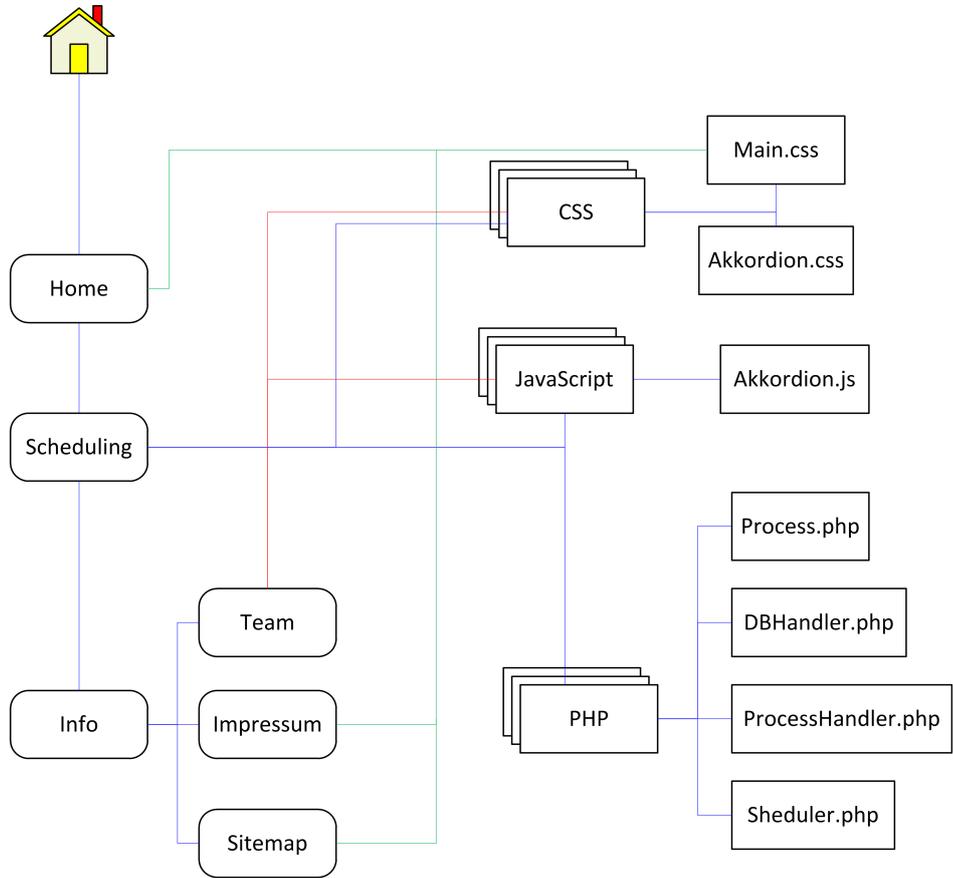
Diese drei Bereiche werden in einer Schleife nun solange durchgelaufen bis die Warteschlange der Prozesse Leer ist. Ist dies der Fall werden die Prozesse an die Klasse ProzessHandler übergeben und der Name des Verfahrens. Round Robin unterscheidet sich zu den Anderen in einer leider Komplizierteren Auswahl der Prozesse die nicht in einer Separaten Methode erfolgen kann. Aus diesem Grund hat das Verfahren eine eigene Methode bekommen die für die Auswahl der Verdrängung der erneuten Zuweisung und der Zuweisung der Prozesse andere Bedingungen benötigt.

5 ANHANG

5.1 KLASSENDIAGRAMM



5.2 HOMEPAGE AUFBAU



5.3 ABLAUFPLAN

