MVC-Architektur

# Model:

Das “Model” des MVC-Prinzips wird in der “F-EAT-BACK” Applikation durch diverse PHP-Dateien realisiert. Zu diesen PHP-Dateien gehören die “Bewertung.php”, die “Details.php” und die “Overview.php”. Diese sind für die Bearbeitung und Verwaltung der im View präsentierten Daten verantwortlich. So kümmert sich die “Bewertung.php” um die Bewertungen, nimmt diese bei POST Anfragen des Controllers auf und speichert sie in der Datenbank ab. “Details.php” gibt auf die GET Anfrage des Controllers alle abgespeicherten Details zu dem ausgewählten Essen aus. Bei der “Overview.php” ist es sehr ähnlich, diese gibt die Titel aller Speisen des aktuellen Tages als Übersicht aus, wenn sie eine GET Anfrage des Controllers erhält. Somit sind die PHP-Dateien dafür zuständig, die Daten als Objekte zwischen Front- und Backend zu versenden. Diese Datenobjekte repräsentieren somit das Model.

# View:

Die View des MVC-Prinzips findet sich in diesem Projekt in der HTML-Seite, also die “index.html”, sowie in den CSS Dateien und dem Bootstrap-Framework welches in der Version 3.X genutzt wird. Die “index.html” ist der erste und direkte Berührungspunkt des Systems mit dem User und stellt als View sämtliche Informationen nur dar, ohne selbst einen eigenen Inhalt - also Daten - zu besitzen.

Das CSS sowie Bootstrap unterstützten die View-Schicht, indem sie die Usability und das Aussehen verbessern und die View mittels responsiven Design für Mobilgeräte optimiert. Die “index.html” ist somit der Hauptteil der View-Schicht. In ihr werden alle Daten abgebildet. Die von Bootstrap bereitgestellten CSS-Dateien und die selbst erstellte “template.css” greifen hierbei auf zahlreiche Klassen und Objekte zu, um die HTML-Seite gezielt zu modifizieren.

# Controller:

Der Controller ist in diesem Projekt differenziert für Client und Server zu betrachten. So ist der Client Controller in Form von Javascript-Dateien vertreten. Vor allem die “app.js”-Datei übernimmt dabei einen großen Teil der Funktionen. Zu Beginn lädt diese beispielsweise alle Details indem sie das Model mit einer GET Anfrage anstößt. Anschließend setzt sie den Anzeigestatus der Details auf unsichtbar. Je nach Anwahl bzw. Klick des Users werden dann zu der Auswahl entsprechende Details in Form eines Pop-Ups aufgerufen und präsentiert. Auch das Laden der Übersicht und die Logik hinter den einzelnen Buttons auf welchen die Übersicht abgebildet wird, übernimmt das Javascript. Will der Nutzer schließlich eine Bewertung abgeben, überprüft der Controller zunächst ob die Bewertungsfelder richtig ausgefüllt wurden und gibt dieses dann mit den entsprechenden Parametern an das Model weiter.

Serverseitig wird der Controller durch alle PHP-Dateien realisiert die den REST-Service darstellen (alle Dateien im “REST”-Ordner des Repositories). Diese PHP-Dateien nehmen die Anfragen des Javascripts entgegen. Anschließend geben sie entweder die zu speichernden Daten als ein Objekt (welches das Model ist) an die Datenbank weiter, oder aber sie nehmen die abgerufenen Dateien von der Datenbank entgegen und geben diese über das Javascript an das Frontend weiter.