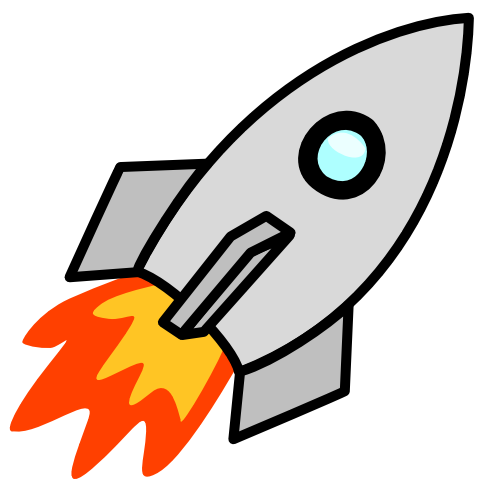
**Projektdokumentation**

Ticketverwaltungssystem Rocket



In

PERL

Datenbanken

Von

Thomas Dorsch und Matthias Nagel

Inhaltsverzeichnis

**(wird später automatisch erstellt)**

# Aufgaben- und Problemstellung

Wir hatten es uns zur Aufgabe gemacht, ein webbasiertes Ticketverwaltungssystem zu erstellen, das beispielsweise in einem IT- oder Softwareunternehmen für den Kundensupport eingesetzt werden kann.

Dabei sollte es einen separaten, passwortgeschützten Nutzer- und einen Mitarbeiter-Bereich mit Rechtverwaltung geben.

Da das Projekt den im Unterricht durchgenommenen Stoff beziehungsweise dass daraus gewonnene Wissen sowie Kenntnisse wiederspiegeln soll, haben wir es uns als Ziel gesetzt, die Programmier-Aufgaben mit reinem Perl-CGI und etwas HTML-Code zu realisieren. Dieser sollte einen strukturierten und modularen Aufbau besitzen.

## Ausbauphasen

Da wir bisher nur wenig Erfahrungen mit CGI gesammelt hatten, haben wir unsere Aufgaben und deren Umsetzung nach Wichtigkeit und funktionaler Notwendigkeit der einzelnen Punkte in mehrere Ausbaustufen unterteilt, um zum Ablauf der Projektfrist ein voll funktionsfähiges Ticketverwaltungssystem präsentieren zu können, das im Nachgang noch um weitere Zusatzfunktionen ergänzt werden kann.

### Phase 1: Sollvorgaben

* Reines Perl-CGI, HTML, CSS sowie MySQL Projekt
* Zugangsschutz für die „internen Bereiche“
* Gehashte Passwörter zur besseren Sicherheit
* Registration neuer Benutzer
* Aufteilung in verschiedene Bereiche mit unterschiedlichen Rechten
  + Benutzerbereich:
    - Erstellen neuer Tickets
    - Übersicht aller eigenen erstellten Tickets
    - Antwortfunktion auf Tickets
  + Mitarbeiterbereich mit verschiedenen Zuständigkeiten
    - Übersicht der neuen Tickets
    - Bearbeiten der Tickets
    - Effiziente Arbeitsverteilung
    - Ticketauswertung in Graphen
  + Administrationsbereich
    - Userverwaltung
    - Rechtegabe und -verwaltung
* Gemeinsame Login-Seite für alle Bereiche
* Logout-Funktion

### Phase 2: Kannvorgaben mit höherer Priorität

* Bearbeitung eines Tickets durch nur einen Mitarbeiter zur selben Zeit
* Möglichkeit der Ticket-Priorisierung
* Einheitliches, übersichtliches Layout in allen Bereichen
* Zuordnung des Tickets zu einer Kategorie
* User können Account selbst bearbeiten und löschen
* Änderung der Zugriffsrechte

### Phase 3: Kannvorgaben als Zusatzfunktionen

* Benachrichtigungen per Email oder Instant Messanger
* Maximale Anzahl der angezeigten Tickets pro Seite
  + Erst im späteren Betriebsverlauf nötig
* Vorhandene Statistik-Grafiken ausweiten
* Suchfunktion für Tickets / Problemkategorie

## Aufgabenverteilung

Um eine genauere Vorstellung über unsere Ziele und Vorgaben zu bekommen und deren Umfang besser einschätzen zu können, haben wir uns zu Anfang des Projekts gemeinsam an die Konzentration der Ideen und deren Einbringung und Umsetzung gemacht, um ein erstes Konzept ausarbeiten zu können.

### Gemeinsame Planung

* Zielsetzung unseres Projekts
* Erstellen erster Layout-Mockups
* Durchdenken der relationalen Datenbankstruktur
* Konzept zur Namensgebung der Module und Funktionen
* Aufteilung von Test- und Serverumgebung

### Aufgaben von Thomas Dorsch

* Erstellen eines Entscheidungsbaums zur Abfrage
* Ausarbeitung Webseiten-Layouts mit CSS
* Programmierung des Userbereichs mit Perl-CGI
* Programmierung der Login-, Logout- und Registrierungsfunktion in Perl-CGI
* Sitzungsverwaltung mit Sessions

### Aufgaben von Matthias Nagel

* Aufsetzen der Serverumgebung mit Webserver, Datenbankserver
* Einrichten des GIT-Repositories zur gemeinsamen Versionsverwaltung
* Erstellung der Datenbankstruktur
* Bash-Script zur Datenübermittlung an Testumgebung
* Programmierung des Mitarbeiter- und Administratorbereichs mit Perl-CGI

# Schnittstellen

Um unser Projekt planen, durchführen und dokumentieren zu können, haben wir folgende Fremdsoftware im Einsatz gehabt.

Hierbei unterscheiden wir die Programme nach Verwendungszweck. Um gemeinsam zur selben Zeit am Projekt arbeiten zu können, nutzen wir GIT zur Versionsverwaltung und Sicherung unserer Dateien.

## Perl inklusive verwendeter Module

* Perl 5.10
* CGI
* DBI
* DBD::mysql
* HTML::Table

## Server- bzw. Einsatzumgebung

* Virtueller Privater Server (VPS) mit Debian 6 als Betriebssystem
* Datenbankserver: MySQL mit PHPMyAdmin
* Webserver: Apache2
* Apache-Perl-Modul

## Testumgebung

* Betriebssystem: Windows 7
* Perl-Umgebung: ActivePerl 5.16.1
* Entwicklungsumgebung: Eclipse SDK 4.2.1, EPIC-Plug-In für Eclipse, Notepad++
* Webserver: XAMPP
* Datenbankserver: MySQL mit PHPMyAdmin
* Bash-Script zur Datenübermittlung (siehe Anlage)

## Arbeitsbereich (zusätzlich zur Testumgebung)

* Versionskontrolle: Github for Windows 1.0.41.2
* Windows-Console
* MySQL Workbench 5.2.44