

Webcam installieren und konfigurieren



	Konzept und Abschlussbericht
	Mauchle Nicolas Augsburger Denis

**Lehrer
Schaub Rolf**

Pratteln, Mai 2011

Versionierung

Vers.	Datum	Autoren	Status	Änderungen
1.0	16.03.2011	DA & NM	Konzept erstellt	
1.1	17.05.2011	DA & NM	Abschluss Bericht	
1.2	24.05.2011	DA & NM	Abschluss Bericht	

Inhalt

Versionierung	2
Konzept.....	3
Analyse	3
Panorama erstellen	4
Archiv aufräumen	4
Abschluss	5
Voraussetzungen	5
Client.....	5
Server	5
Webcam	5
Umsetzung.....	5
Panoarma erstellen	5
Archiv aufräumen	5
Datenbank Schema.....	6
Installation.....	6
Manuelle Installation.....	6
Test	7

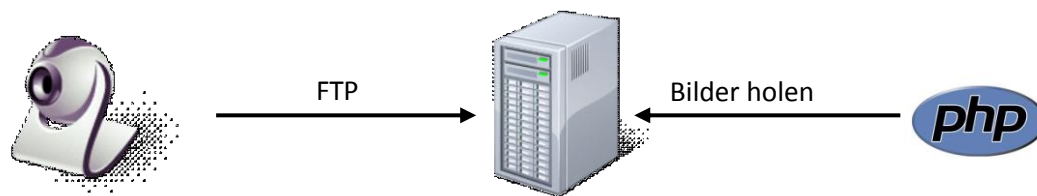
Konzept

Analyse

Wir haben eine Webcam von Level One. Sie verfügt über einen Netzwerkanschluss und kann somit direkt in das lokale Netzwerk angeschlossen werden. Es wird eine Software mitgeliefert, über welche die Kamera gesteuert werden kann. Leider funktioniert dies nur mit dem IE richtig, da ein Active-X Steuerelement gebraucht wird.

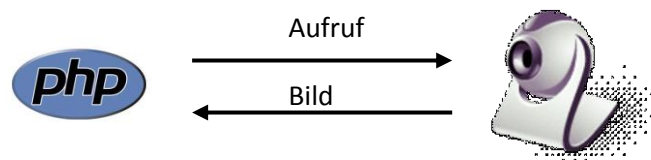
Wir möchten mit dieser Kamera Panorama Bilder erstellen und auf unserer Webseite anzeigen. Dies soll zu bestimmten Zeiten geschehen. Für die Ansteuerung haben wir uns über verschiedene Möglichkeiten informiert. Hier sind die 2 Hauptvarianten, welche sich aus unseren Forschungen herauskristalisiert haben.

Variante 1:



Die Kamera wird so konfiguriert, dass zu bestimmten Zeiten die Bilder erstellt und per FTP auf einen Server hochgeladen werden. Das PHP Script holt sich die Bilder im FTP-Verzeichnis und generiert daraus ein Panorama.

Variante 2:



Das PHP Script steuert die Kamera direkt über die mitgelieferte CGI-Schnittstelle. Die zurückgegebenen Bilder werden zu einem Panorama erstellt.

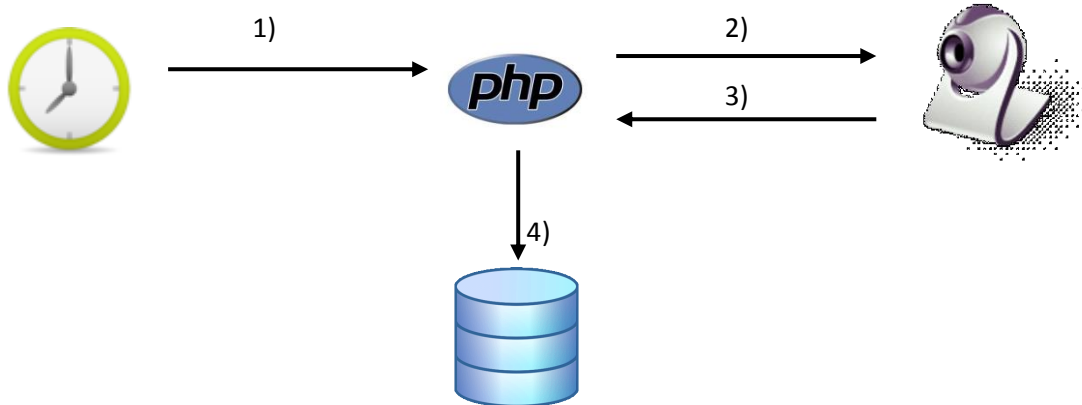
Entscheidung

Unsere Entscheidung ist auf die Variante 2 gefallen. Die beiden Systeme sind unabhängiger voneinander. Es ist die sichere Variante, da kein FTP Dienst gestartet werden muss. Es ist schneller, da die Kamera direkt angesprochen wird.

Panorama erstellen

Hier ist eine Gesamtübersicht wie das erstellen eines Panoramas funktioniert.

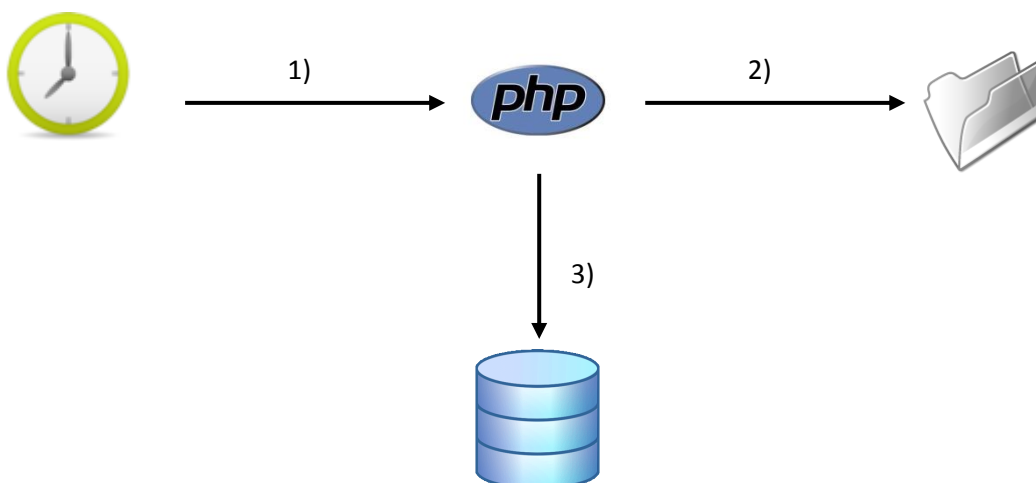
- 1) Ein Cronjob ruft zu definierten Zeiten das Script panorama.php auf
- 2) Das Script positioniert die Kamera und erstellt temporäre Bilder
- 3) Die zurückgegebenen Bilder werden in einem Panorama gespeichert und die entsprechenden Informationen in der Datenbank abgelegt



Archiv aufräumen

Das Archiv wird täglich aufgeräumt. Dabei werden die Fotos im Filesystem und in die Pfade darauf in der Datenbank gelöscht. Wir löschen zuerst das Filesystem, da dort ein Log, welcher zu einem Fehler führt eher auftritt.

- 1) Ein Cronjob ruft `deleteOldEntries.php` auf
- 2) Das Script löscht die Bilder und den Ordner der Einträge die älter als 14 Tage sind
- 3) Bei Erfolg werden die dazugehörigen Datensätze gelöscht



Abschluss

Voraussetzungen

Client

Der Client greift über einen Browser auf unsere Webseite zu. Mit aktiviertem Javascript, wechselt der Banner mit den letzten 3 erstellten Bildern. Zusätzlich wird es verwendet um die Bilder im Archiv vergrößert darzustellen. Ohne Javascript funktioniert die Webseite ebenfalls. Als Browser wurden Tests mit aktuellen Versionen von Firefox, Chrome und Internet Explorer durchgeführt. Dabei wurden keine Abweichungen festgestellt.

Server

Der Server ist eine LAMP Konfiguration. Ein Ubuntu(Linux) mit einem Apache2 Server, Mysql 5.1 Datenbank und PHP 5 als Scriptsprache. Der Server muss unter Linux laufen, damit der Cronjob funktioniert. Ansonsten muss unter Windows ein Task eingerichtet werden. In der PHP-INI muss `allow_url_fopen` aktiviert sein. Zusätzlich muss Lynx installiert sein, damit der Cronjobs ordnungsgemäss funktionieren.

Webcam

Die Webcam muss richtig konfiguriert sein. Die URL zur Kamera muss unter `inc/config.php` inkl. User und Passwort angegeben werden.

Umsetzung

Panoarma erstellen

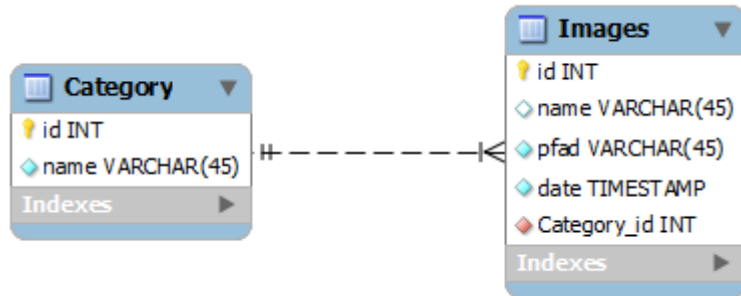
Das PHP Script wird von einem Cronjob jeweils um Acht, Zwölf und Sechs Uhr aufgerufen. Dieses steuert die Kamera über eine CGI-Schnittstelle an die entsprechenden Positionen und erstellt die Bilder. Das Panorama wird mit 3 Bildern erstellt und im Ordner `images/DATE` gespeichert. Ein Pfad auf das Bild wird in der Datenbank erstellt und das Panorama auf der Webseite angezeigt.

Archiv aufräumen

Das Script wird mit einem Cronjob um 2 Uhr Morgens aufgerufen, da um diese Zeit, wahrscheinlich niemand auf die Bilder zugreift. Es werden über ein SQL alle Datensätze ausgewählt, welche älter als 14 Tage sind. Diese werden dann auf dem Filesystem Rekursiv der Ordnerstruktur gelöscht und bei Erfolg auf der Datenbank nachgeführt.

Datenbank Schema

Die Tabelle Category speichert ob das Bild Morgens, Mittags oder Abends aufgenommen wurde. Zusätzlich wird mit einem Timestamp bei Images genau festgehalten, wann das Bild aufgenommen wurde. Der Pfad ist der Pfad zum Bild und der Name ist der Dateiname des Bildes inkl. Dateiendung.



Installation

Auf einem installierten Webserver müssen diese weiteren Schritte durchgeführt werden:

- Entpacken des Zip-Archivs
- Ausführen der Install.sh
- Anpassen von inc/config.php
- INI-Datei Einstellungen kontrollieren (allow_url_fopen=true)

Manuelle Installation

- Datenbank mountaincam erstellen
- User mountaincam mit allen Rechten auf DB mountaincam erstellen
- Schema.sql in DB importieren
- Mountaincam.tar.gz entpacken in Webroot
- Rechte setzen (Apache)
- Crontab einrichten
- Anpassen von inc/config.php
- INI-Datei Einstellungen kontrollieren (allow_url_fopen=true)

Test

Wir haben nach erfolgreicher Umsetzung des Projektes einen Integrationstest durchgeführt. Dabei konnten wir alle Anforderungen erfüllen.

Testfall	Erwartetes Ergebnis	Ergebnis
Panorama erstellen	Panorama wurde korrekt in DB und Filesystem erstellt	OK
Panorama auf Webseite anzeigen	Panorama wird auf Webseite mit korrekten Daten angezeigt	OK
Archiv aufräumen	Die Dateien und Datensätze, welche älter als 14 Tage sind, sind gelöscht worden	OK
Kamera Zugriff	Auf die Kamera, kann nur mit User und Passwort zugegriffen werden	OK
Seite mit verschiedenen Browser getestet	Die Webseite ist Funktionsfähig und weicht nur gering vom Design ab	OK
Plattform zügeln	Einfaches zügeln unter Linux servern möglich, dank Installationsscript	OK