Funktionale Anforderungen

Das System muss die Möglichkeit anbieten, sich ein Konto anzulegen.

Das System muss die Möglichkeit anbieten, die Detailen seines Kontos zu bearbeiten.

Das System muss die Möglichkeit anbieten, sein Konto selbst zu löschen.

Das System muss die Möglichkeit anbieten, sich an seinem Konto anzumelden.

Das System muss die Möglichkeit anbieten, angemeldeten Benutzern SSH-Sitzungaufnahmen hochzuladen.

Das System muss die Möglichkeit anbieten, angemeldeten Benutzern den Titel, die Beschreibung und Kategorien ihrer SSH-Sitzungaufnahmen zu bearbeiten.

Das System muss die Möglichkeit anbieten, angemeldeten Benutzern ihre Videos zu löschen.

Das System muss fähig sein, UML-Kompatibel SSH-Sitzungsaufnahmen in ein asciinema-kompatibel-Format umzuwandeln.

Das System muss die Möglichkeit anbieten, SSH-Sitzungaufnahmen anzuschauen.

Das System muss die Möglichkeit anbieten, SSH-Sitzungaufnahmen nach Schlusswörter zu suchen.

Das System muss fähig sein, SSH-Sitzungsaufnahmen nach dem Datum, der Bewertung oder Kategorien zu sortieren.

Das System muss die Möglichkeit anbieten, angemeldeten Benutzern hochgeladene SSH-Sitzungsaufnahmen zu bewerten.

Das System muss die Möglichkeit anbieten, vertrauenswürdigen Benutzern unangemessene Videos oder Kommentaren zu moderieren.

Das System muss eine typische Privacy-policy-seite zeigen.

Das System muss eine typische About-seite zeigen.

Das System muss einfache Gebrauchsanweisungen für die Honeypot-App zeigen.

Das System soll die Möglichkeit anbieten, angemeldeten Benutzern hochgeladene SSH-Sitzungsaufnahmen zu kommentieren.

Das System soll die Möglichkeit anbieten, unangemessen Videos oder Kommentaren zu melden.

Das System soll die Möglichkeit anbieten, angemeldeten Benutzern ihre Kommentaren zu löschen.

Das System soll die Möglichkeit anbieten, sein Kennwort zurückzusetzen.

Das System soll die Möglichkeit anbieten, vertrauenswürdigen Benutzern Blogposts über die Webseite hochzuladen.

Das System soll die Möglichkeit anbieten, vertrauenswürdigen Benutzern neue Kategorien zu schaffen.

Nicht funktionale Anforderungen

Das System muss Kennworte als Hash speichern.

Das System muss wenige Daten über die Benutzer speichern, damit die Privacy Policy einfach bleibt.

Das System soll die Sicherheitsrisiken in Bezug auf die Umwandlung der UML-Dateien mindern.

Das System soll in der Lage sein, die meisten Bots zu verhindern.

Vorgehensweise

Wie sind wir dazu gekommen, eine sammelnde Honeypotaufnahmenwebapp zu implementieren?

Es stand mehrere Möglichkeiten zur Wahl:

* Die Landingpage einer Firma
* Honeypotaufnahmenwebapp

Die Umsetzung der erste Möglichkeit wäre zwar relativ einfach gewesen, aber wäre vielleicht nicht ehrgeizig genug für eine Gruppe von drei Leuten.

Die zweite Wahl klang gut, ein Element musste schon geprüft werden:

Der Entwurf der Webseite hat dazu gedient, sowohl sich einen Überblick zu schaffen über wie es aussehen sollte, als auch grobe Ziele zu definieren.

Dank parallelen Klassen haben wir uns auch auf ein schon existierendes Datenbankmodell basieren können. Tatsächlich kam die Idee aus diesem anderen Kurs.

Zuerst ginge es darum, sich für eine Liste von Technologien zu entscheiden. Diese Auswahl stütze sich auf keine besondere Methodologie. Diese geschah nach den folgenden Prinzipien: was schon etwa gemeistert ist, sollte verwendet werden. Da Kaytano wenig Erfahrung mit Webdev hatte, konnte er schon viel davon lernen, und sich geläufige Werkzeugen und Software-Pattern aneignen.

Docker wurde schon extensiv bei den Kursen benutzt, weswegen wurde es benutzt, um eine gemeinsame Entwicklungsumgebung zu definieren, auf der PHP, Apache und PostgreSQL laufen würde.

Was die Datenbank angeht, haben wir uns auf PostgreSQL verlassen: tatsächlich ist OracleXE nicht besonders gut unterstütz von PHP-Bibliotheken und Richard hatte auch Erfahrung damit.