# Pflichtenheft: Crash- Cam

Giorgio Groß, Christoph Hörtnagl, David Laubenstein, Josh Ramanowski, Fabian Wenzel

17. November 2016

## Inhaltsverzeichnis

# 1 Zielbestimmung

## 1.1 Musskriterien

## 1.2 Wunschkriterien

## 1.3 Abgrenzungskriterien

# 2 Produkteinsatz

## 2.1 Anwendungsbereiche

## 2.2 Zielgruppen

## 2.3 Betriebsbedingungen

# 3 Produktumgebung

## 3.1 Software

## 3.2 Hardware

## 3.3 Orgware (Netzwerkverbindung zum...)

## 3.4 Produktschnittstellen

# 4 Funktionale Anforderungen

## 4.1 Kundenverwaltung (Platzhalter)

## 4.2 Seminarverwaltung (Platzhalter)

## 4.3 Rechnung erstellen (Platzhalter)

# 5 Produktdaten

## 5.1 Kundendaten (Platzhalter)

## 5.2 Seminardaten (Platzhalter)

## 5.3 Buchungsdaten (Platzhalter)

# 6 Nichtfunktionale Anforderungen

# 7 Globale Testfälle

## 7.1 Erklärung zu den Testsuites der Qualitätssicherung

Um eine möglichst flächendeckende Qualitaätssicherung zu ermoöglichen, werden wir im Folgenden zwischen verschiedenen Arten von Testablaäufen unterscheiden. Diese sollen automatisierte Tests sein und das Backend und Frontend testen. Das Ziel ist, ca 70 bis 80 Prozent des geschriebenen Codes ausführhrlich zu testen.
Wir teilen das Testen in mehrere Testsuites auf. Die erste grosse Testsuite sind die **Komponententests**. Diese testen die einzelnen Bestandteile der Software auf deren Funktionalität.
Die zweite Testsuite ist die **Integration-Testsuite**. In dieser Phase wird das Zusammenspiel von Server, Webinterface und Android-App getestet.
Die dritte Testsuite sind die **Systemtests**, bei denen das komplette System auf deren Funktionalität überprüft wird.
Als vierte und letzte Testsuite führe wir den Abnahmetest durch. Dieser testet die Software unter realen Bedingungen.

### 7.1.1 Komponententests

Komponententests sind allgemein dazu da, um einzelne Komponenten der Software zu testen. In unserem Beispiel werden wir den Java-Server, die Android-App und das Webinterface auf deren Funktionalität testen.

### 7.1.2 Integration-Testsuite

In der Integration-Testsuite wird die Kommunikation der Komponenten getestet.

### 7.1.3 Systemtests

Die Software wird nun auf einer realen Umgebung installiert und mit Testdaten gefüllt. Dort wird die Software unter realen Bedingungen getestet.

## 7.2 Testzsenarien

### 7.2.1 REST-Request des Webinterfaces

Der Kunde führt auf der Anmeldeseite einen Log-In durch udn will nun seine gespeicherten Videos verwalten. Dabei wird ein REST-Request des Webinterface an den Server gesendet, dieser bearbeitet den Reqüst und liefert als Response eine Antwort, die dann vom Webinterface dargestellt wird.

### 7.2.2 Speichere gerade aufgenommenen Unfall

Der G-Sensor des Geräts löst aus, da gerade ein Unfall geschehen ist. Nun wird das Video verschlüsselt auf dem lokalen Speicherbereich abgelegt. Zudem wird es nun unter noch nicht hochgeladeneÜnfälle angezeigt. Die Möglichkeit zum hochladen, löschen und umbenennen des Unfalls wird angezeigt und jede der Möglichkeiten funktioniert.

### 7.2.3 Anonymisierung des Videos auf dem Java-Server

Der Kunde hat einen Unfall aufgenommen und diesen lokal gespeichert. Nun betätigt er die Funktion Ünfall hochladen". Das bereits lokal verschlüsselte Video wird zum Server gesendet und dieser legt die Datei in das "/tmpVerzeichnis ab, um die Daten nur temporär zu speichern. Das Video wird anonymisiert und richtig auf dem Server abgelegt. Dem Usereintrag in der Datenbank wird um den Pfad zum neu abgespeicherten Video ergänzt, damit der Kunde das Video nun auch auf seiner App/Webinterface sehen kann.

# 8 Systemmodelle

## 8.1 Szenarien

## 8.2 Anwendungsfälle

## 8.3 Objektmodelle

## 8.4 Dynamische Modelle

## 8.5 Benutzerschnittstellen - (Bildschirmskizzen, Navigationspfade, etc (Platzhalter))

# 9 Glossar