Supplementary Specification
Projekt BierIdee

Danilo Bargen, Christian Fässler, Jonas Furrer

20. März 2012

1 Einleitung

Der Sinn und Inhalt dieses Dokumentes ist es, hautpsächlich nicht funktionale Anforderungen an das Projekt gestellt sind festzuhalten. Diese beinhalten Qualitätskritereien nach FURPS+ aber auch Anforderungen an die Umgebung wie Hardware, Netzwerk, Internationalisierung, Dokumentation.

2 Usability

2.1 Efficiency

Die Benutzbarkeit der Client Software ist ein wichtiger Aspekt in diesem Projekt, da es auf mobilen Geräten, vielleicht ausschliesslich durch Toucheingaben, bedient wird. Die Applikation wird daher so entworfen, dass Sie sich mit reiner Toucheingabe bedienen lässt.

2.2 Learnability

Bei Applikationen für Mobiltelefonen ist es Erfolgs entscheidend, dass Sie intuitiv zu bedienen und zugleich lernfördernd sind. Es wird daher bewusst auf eine Hilfestellung in Form einer Bedienungsanleitung für die Clientseitige SW verzichtet. Die Sicherstellung dieser merkmale geschieht durch regelmässige Tests und Bewertungen durch Testpersonen. (siehe User Tests).

3 Performance

Im Rahmen dieses Projektes ist das System und die Serverseitige Umgebung für den Einsatz mit ¡50 Benutzern, 100 Bieren, 20 Brauereien ausgelegt.

3.1 Response Time

80% der Anfragen an das System sollen innerhalb von 3 Sekunden ausgeführt werden. Der durchschnitt soll bei 1.5 Sekunden liegen.

3.2 Computing Resource

Die Architektur wird so ausgelegt, dass so wenig wie möglich Rechenleistung auf Clientseite benötigt wird. D.h sämtliche Logik und Aufbereitung der Daten wird auf der Serverseite implementiert. Weil clientseitig die Unterschiede in der Verwendeten

Hardware sehr gross sein können (Smartphones, Tablets, PC's), können keine Einheitlichen Anforderungen zur Rechenkapazität auf den Geräten gestellt werden. Mit der zentralen Auführung Rechenintensiver Aufgaben, ist die Kapazität genaustens bekannt und kann adäquat ausgelegt und allenfalls angepasst werden.

4 Benutzer Tests

Nach jeder Constructionphase wird die Applikation durch einen definierten Benutzerkreis getestet. Die Benutzer sollen Einerseits anhand einer ausgehändigten Testspezifikation das System testen. Ebenso sollen die User das System nach belieben Testen können und Feedback direkt via Ticketsystem oder Email zu melden. Dieses Feedback und ev. daraus entstandene Tickets werden an den wöchentlichen Statusmeetings durch das Projektteam angeschaut und priorisiert.

5 Scalability

In Rahmen dieses Projektes kann die geforderte Last an das System gut eingeschätzt werden (nur Testuser und Entwickler). Falls das Projekt weitergeführt wird, muss das System möglicherweise skaliert werden. Die eingesetzten zentralen Komponenten wie Reslet und Postgres DB unterstützen eine Skalierung in hohem Masse.

6 Reliability

6.1 Security

Das System verfügt über Benutzerdaten wie Emailadresse, Konsumierte Getränke. Diese Benutzerdaten sind nur via Login berechtigen Benutzern zugänglich. Die gespeicherten Benutzerdaten sind aber nicht als "heikelänzusehen, da durch deren unabsichtliche Veröffentlichung keine finanzielle sowie massgebende persönliche Schäden zur Folge haben. Daher werden keine zu erfüllenden Anforderungen in Punkto Sicherheit (zu erfüllende Standards o.ä) spezifiziert.

6.2 Massnahmen

Passwort Policy Für Admin Passwörter werden Passwörter mit mindestens 8 Zeichen Alphanummerisch verwendet.

Backup Das Serversystem und die Datenbank wird täglich gesichert. (File Hot-Copy) Zusätzlich gibt es einen Snapshot der kompletten Serverinstanz, welches bei

einem kompletten Systemausfall auf einen anderen VM Host umgezogen werden kann.

6.3 Availability

Da es sich bei diesem Projekt um ein "Fun"Projekt handelt, welches bei nicht Verfügbarkeit keine finanziellen o.ä Schäden verursacht werden keine konkreten Anforderungen an die Systemverfügbarkeit gestellt. Grundsätzlich soll das System aber dennoch möglichst Hochverfübar sein, um kontinuierliches Testen zu ermöglichen (siehe auch Punkt Benutzer Tests). Es werden aber keine Massnahmen wie z.B der Einsatz von redundanten Komponenten vorgenommen.

6.3.1 Schwachstellen

Datenbank Server Das System besitzt eine zentrale Datenbank. Fällt diese Komponente aus, ist die Client Applikation nicht mehr benutzbar.

REST API Ist die Serverkomponente (REST API) nicht mehr verfügbar, ist die Client Applikation nicht mehr benutzbar.

6.4 Fehlererkennung

Das System wird mittels Integration- und Systemtests regelmäßig auf Funktionalität und damit auch auf Verfügbarkeit getestet.

7 Supporability

7.1 Internationalization

Das System soll verschiedene Sprachen unterstützen. Lokalisierung wird nicht unterstützt. D.h keine unterschiedlichen Masseinheiten, Währungen.

8 Design Requirements

9 Implementation Requirements

9.1 Database

Für das System wird eine Relationale Datenbank mit JDBC Kompatibilität benötigt. In diesem Projekt wird Postgres eingesetzt.

9.2 Programming Language

Als Programmiersprache wird für alle Teilsysteme (REST API, Client) Java Version 1.6 verwendet.

9.3 Frameworks

RESTlet	Version 2.0.11 (stable)
Android SDK	API Level 10, Android 2.3.3