Anlage A Pflichtenheft

Inhalt

1. Zielbestimmung

1.1 Musskriterien

1.2 Sollkriterien

1.3 Abgrenzungskriterien

1. Produkteinsatz

2.1 Anwendungsbereich

2.2 Betriebsbedingungen

1. Produktübersicht

3.1 Funktionsumfang

3.2 Benutzeroberfläche

1. Technische Produktumgebung
2. Glossar
3. **Zielbestimmung**

Als Ziel wurde die Entwicklung eines bis zum Prototypenstatus zu entwerfendes Programm definiert. Anhand dieses Prototypen soll aufgezeigt werden, ob sich die aktuell im produktiven Einsatz befindliche Technologie durch andere Ansätze ablösen bzw. für Projekte mit differenzierten Anforderungen ergänzen lässt. Um die Tauglichkeit bestimmen zu können, soll herausgefunden werden, ob die Technologie, die für dieses Projekt zum Einsatz kommt, eine ausreichende Funktionalität und Performancevorteile bietet. Zum einen soll die neue Software an eine bestehende Softwareschnittstelle angebunden werden, um so die Kompatibilität sicher zu stellen, zum anderen soll herausgefunden werden, mit welchem Ressourcenaufwand die Software später betrieben werden kann.

**1.1 Musskriterien**

Die zu erfüllenden Anforderungen sind sowohl die Implementierung eines eigenständigen Programmes, das ohne zusätzlich vorausgesetzte Programme laufen soll.

Diese Kernsoftware, muss auf Linux basierten Systemen lauffähig sein und möglichst Ressourcen schonend arbeiten.

Die Benutzerschnittstelle soll über die Anbindung an eine Schnittstelle des Backoffice des Unternehmens dynamisch erzeugt werden können. Dies bedingt nicht die vollständige Umsetzung der Anbindung an diese Schnittstelle, sondern ausschließlich den Nachweis der Machbarkeit.

**1.2 Sollkriterien**

Als weiteren Nachweis der Tauglichkeit bzw. der Vorteile der neuen Technologie soll die Benutzerschnittstelle zusätzlich zur Anbindung an das Backoffice auch über statisch deklarierte Oberflächen erfolgen. Dies Beinhaltet die Implementierung einer QML basierten Schnittstelle zum Benutzer. Für diese Schnittstelle wird ein eigens für diesen Prototypen entworfenes Layout beschreiben, welches die Vorzüge dieser Technik verdeutlichen soll.

**1.3 Abgrenzungskriterien**

Da diese Projekt ausschließlich den Entwurf und die Entwicklung eines Proof of Concept beinhaltet, wird darauf verzichtet die Anbindung der Software an die Schnittstelle für die Kommunikation mit dem Automaten zu realisieren. Auch das Reporting an das Backoffice, sowie der Komplette Zahlungsvorgang wird nicht mit eingeschlossen.

Diese Punkte bedürfen eines nicht unerheblichen Entwicklungsaufwandes und würden nur in dem Fall berücksichtigt werden, dass dieses Projekt für eine Weiterentwicklung in Frage käme.

1. **Produkteinsatz**

**2.1 Anwendungsbereich**

Die zu entwickelnde Software muss darauf ausgelegt sein später auf minimalen Rechensystemen lauffähig zu sein. Die Anforderung der Lauffähigkeit auf sogenannten Embeddedsystemen ist ein unabdingbarer Bestandteil der Anforderung, da diese Eigenschaft später zu erheblichen Kostenersparnissen führen soll.

Die Software soll zudem später als Bedienoberfläche eines Automaten dienen. Das setzt eine stabile Laufzeitumgebung voraus, die von möglichst wenigen Faktoren abhängig sein sollte. Weshalb als eines der Hauptkriterien die Entwicklung einer speziellen Software im Mittelpunkt steht. Die das bisherige Mehrkomponenten System ablösen können sollte.

**2.2 Betriebsbedingungen**

Aufgrund der Anwendervielfalt, die an einem Verkaufsautomaten auftritt müssen bestimmte Punkte berücksichtigt werden. Zum einem muss die Benutzerführung intuitiv gestaltet werden. So das auch unerfahrene Benutzer eine hohe Akzeptanz für das System aufbringen. Die Bedienung sollte nicht unnötig kompliziert oder Langwierig gestaltet werden. Auch für den Langzeitbetrieb muss die zugrundeliegende Technologie geeignet sein, da die Anforderungen an die Laufzeitstabilität des Produktes erheblich höher sind als bei normaler Software.

1. **Produktübersicht**

**3.1 Funktionsumfang**

Die Software wird ausschließlich als Proof of Concept implementiert, was jegliche Ansprüche auf Vollständigkeit ausschließt. Dennoch sind bestimmte Grundfunktionen gefordert.

Die Anbindung an das Backoffice für die Generierung dynamischer Layouts muss soweit implementiert werden, dass ein bestimmter Minimalsatz an Informationen interpretiert und visualisiert werden kann.

Gefordert sind hier:

* + Interpretation und Darstellung von „Screen“ Objekten
  + Interpretation und Darstellung von „Multi Purpose Button“ Objekten
  + Interpretation vom Modulen
  + Ausführung der Events „Show“ und „Hide“

Zu Test- und Validierungszwecken wurde zuvor eine bestimmte Menge an Objekten und deren Events definiert.

Zusammenfassung:

* + 2 Screen Objekte
  + mit je 4 Button Objekten
  + beim Click auf einen der Button soll wischen den Screens

hin und her gewechselt werden können

* + die anderen drei Buttons dienen lediglich dem Umschalten der

Sichtbarkeit der Buttons

Das alternative auf QML basierte Layout soll eine möglichst vollständige Benutzerführung beinhalten. Dazu gehören mehrere Bildschirme, die Verschiedenartige Informationen an den Benutzer weiter reichen. Nicht Bestandteil dieser Implementierung ist hingegen der Zahlungsvorgang, da dieser Prototyp ausschließlich die Aspekte der Visualisierung berücksichtigen soll.

Bestandteile dieses Layouts sind:

* + Begrüßungsbildschirm
  + Produktauswahl
  + Produktbeschreibung ( Verkaufsbildschirm )
  + Produktausgabe
  + Mediabildschirm

Diese Definitionen dient mehr der Vollständigkeit der Technologiepräsentation, um Stärken und Schwächen gegenüber der derzeitigen Technologie hervorzuheben. Sie wurden nicht genau definiert und unterliegen dem Ermessen des Entwicklers bezüglich des Designs und der Informationsdichte.

**3.2 Benutzeroberfläche**

Bezüglich der Benutzeroberfläche gibt es derzeit keine weiteren Anforderungsprofile, da sich die Entwicklung des Prototypen schwerpunkmäßig auf die technische Realisierung einer Verkaufsoberfläche beschränkt.

Deshalb wurde keine Benutzer oder genaue Daten definiert, mit denen das System umgehen können muss. Aufgrund der Tatsache, dass das System keine variablen Benutzereingaben vorsieht, erübrigen sich Tests diesbezüglich. Da ein Benutzer nur eine strikt definierte Anzahl an Aktionen auslösen kann, ist eine Eingabevalidierung vorerst nicht von Nöten sein.

Die einzigen Anforderungen an die Benutzeroberfläche sind Lesbarkeit, Verständlichkeit, Skalierbarkeit (bezüglich der Lauffähigkeit auf verschiedenen Ausgabemedien), Stabilität und Geschwindigkeit (soll unnötige Wartezeiten aufgrund von Softwareschwächen ausschließen).

1. **Technische Produktumgebung**

Produktentwicklungen dieser Art sollen später auf Linux basierten Rechnersystemen zum Einsatz kommen. Eine Anforderung an die Software war deshalb die Lauffähigkeit auf Linux Betriebssystemen.

Hier wurden keine Vorgaben gemacht auf welcher Distribution oder Version das System später laufen muss.

Ausgegangen wird derzeit von einer openSuse 11.3 Distribution, sowie einer Linux Angström Distribution für Tests an Embeddedsystemen. Die Lauffähigkeit auf verschiedenartigen Betriebssystemen wird durch die Wahl des Entwicklungsframeworks sichergestellt. Das verwendete Qt Framework wurde so entwickelt, dass es auf allen relevanten Betriebssystemen eingesetzt werden kann. So ist die Portierbarkeit der Software sichergestellt.

Es lag auch keine Definition einer Mindestvoraussetzung oder das Rechensystem vor. Deshalb werden keine Tests auf Erfüllung solch eines Anspruchs durchgeführt.

Für die Belegbarkeit der Vorteile bezüglich der Systemvoraussetzungen werden Tests an verschiedenen Entwicklungssystemen durchgeführt.

Zum einen am Entwicklungssystem selbst:

**Prozessor (CPU): Intel(R) Core(TM)2 Quad CPU Q9400**

**Geschwindigkeit: 2.660,32 Mhz**

**Kerne: 4**

**Arbeitsspeicher: 3,8 GiB**

**Kernel: Linux 2.6.34.7-0.7-desktop x86\_64**

**Distribution: openSUSE 11.3 (x86\_64)**

Zum anderen an einem Embeddedsystem:

**Prozessor (CPU): Super-scalar ARM Cortex -A8**

**Geschwindigkeit: 1 GHz**

**Arbeitsspeicher: 512-MB LPDDR RAM**

**Kernel: Linux 2.6.24**

**Distribution: Angstrom 2007.12**

1. **Glossar**

siehe Glossar Projektdokumentation