Alarm Clock

Jonathan Hyams
Pascal Schmalz

21. März 2017

Inhaltsverzeichnis

0.1 Zweck des Dokument

TODO this is new text

0.2 Kurzbeschreibung

Das Ziel des Projektes ist einen Ersatz zum Programm kAlarm zu entwickeln. Das Program kAlarm erlaubt es den User Benutzerdefinierte Erinnerungen zu erstellen. Mittels Pop-Up Windows wird der User dann zur gegebenen Zeit daran erinnert. Im gegensatz zu kAlarm soll das zu erstellende Produkt Platformübergreifend verfügbar sein. Wie kAlarm soll dieses Produkt unter einer Open Source Lizenz entwickelt werden.

0.3 Projektziele

Unser Auftraggeber ist kein Unternehmen. Es werden somit auch keine Ziele innerhalb eines Unternehmens verfolgt. Ziele welche ein Benutzer unsere Aplikation verfolgen könnte wären beispielsweise:

- Wiederkehrende Events nicht zu vergessen.
- An den Wäschetag erinnert weden.
- Die Wäsche rechtzeitig aus de Wäscheküche holen.
- Den Kuchen nicht im Backofen verbrennen lassen.
- Der Sekretärin zum Geburtstag Blumen schicken.

0.4 Stakeholders

- Auftragsgeber: Prof. Claude Furrer
- Auftragsnehmer: Jonathan Hyams, Pascal Schmalz
- Benutzer: FOSS Communitiy

0.5 Systemabgrenzung

0.5.1 Geschäftsprozesse

Es existieren keine vor oder nachgelagerten Prozesse. Unsere Lösung lässt sich in beliebige viele Prozesse integrieren.

0.5.2 Systeme

Unser System wirkt mit Datebanken zusammen, um die Events persistent zu speichern. Unser System beihnaltet die komponente bis und mit dem JPA, die konfiguration der Datenbank gehört somit nicht mehr zum Projekt.

Das Program baut auf Betriebsystemskomponenten auf. Namentlich auf Cronjob /Schtasks um zur gegebenen Zeit eine Notification zu erzeugen.

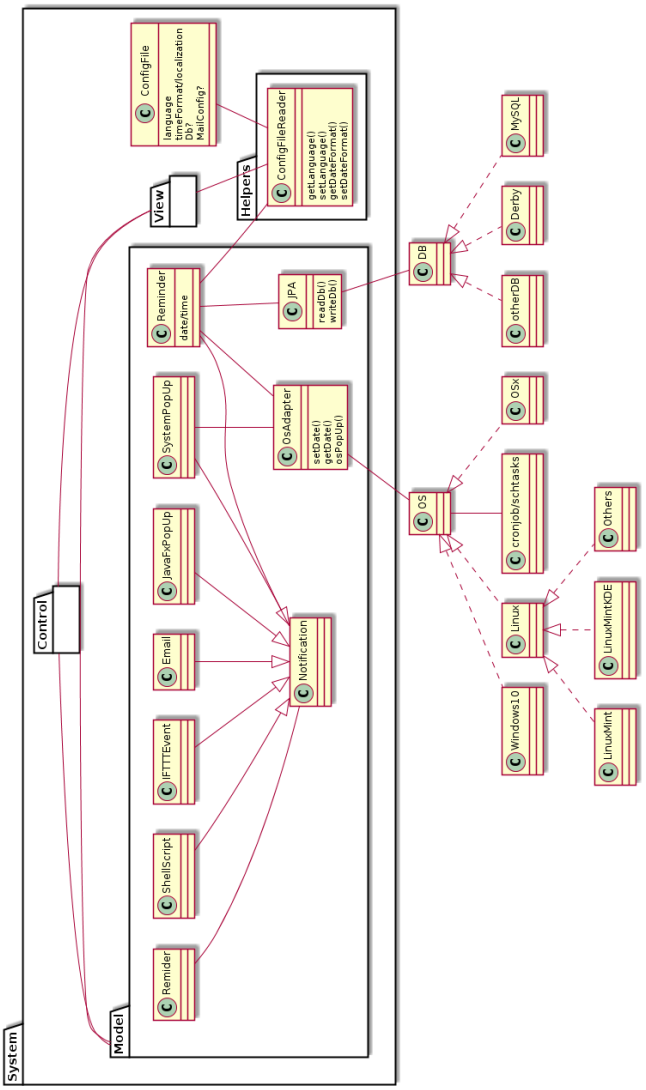

Abbildung 0.1: Systemübersicht und Abgrenzung

Die Systemübersicht und Abgrenzung Abb. 0.1 Seite 6 bedient sich der UML Notation, ohne die UML Spezifikationvollständig einzuhalten. Die Komponenten innerhalb des Kasten“System”, insbesondere ihre Beziehung zueinander, sind nicht Teil der Anforderungen, und somit nicht bindend sie veranschaulichen lediglich was noch innerhalb des Systems liegt und was ausserhalb angesiedelt ist. 6

Falls die Funktionalität vom Auftraggeber verlangt wird, muss das Programm den Nutzer mit E-Mails, SMS und Systemnotifications erinnern. Dies Bedingt, dass entsprechende Services / Server eingerichtet und über eine definierte Schnittstelle erreichbar sind, was nicht mehr teil des Projektes ist.

0.5.3 Randbedingungen

Die Entwicklung soll mittels Java und JavaFX erfolgen. Das Programm soll Betriebssystem und Datenbank agnostisch sein. Das Projekt muss unter einer Opensource Lizenz veröffentlicht werden. Die Dokumentation muss in LaTeX erstellt werden.

0.5.4 Prozessumfeld

0.5.5 Systemumfeld

0.5.6 Randbedingungen

0.6 Anforderungen

0.6.1 Basisfaktoren

- Die Schnittstelle zwischen dem Programmcode und der Datenbank soll so gestaltet sein, dass mit einem minimalen Aufwand eine beliebige gängige Datenabank genutzt werden kann.
- Die Schnittstelle zwischen dem Programmcode und der Datenbank soll so gestaltet sein, dass mit einem minimalen Aufwand eine beliebige Standard Datenbank genutzt werden kann.
- Die Bereitstellung eines JPA Drivers der Datenbank ist dabei nicht mehr im Scope des Projektes, sondern wird von den Datenbankherstellern oder von dritten vorgenommen.
- Das System muss dem Benutzer die Möglichkeit bieten, in dem GUI ein Einzelevent zu erfassen. Das System sollte dem Benutzer die Möglichkeit bieten, über das GUI ein Einzelevent zu erfassen.
- Das System muss dem Benutzer die Möglichkeit bieten, in dem GUI wiederkehrende Events zu erfassen. Das System sollte dem Benutzer die Möglichkeit bieten, über das CLI wiederkehrende Events zu erfassen.

- Das System kann anderen Programmen über eine API ermöglichen, Einzelevents zu erfassen. Das System kann anderen Programmen über eine API ermöglichen, wiederkehrende Events zu erfassen.
- Das löschen und anpassen von Events soll wie oben beschrieben, ebenfalls möglich sein.
- Das System muss den Benutzer mittels einer Notification rechtzeitig, zur voreingestellten Zeit, auf ein Event hinweisen.
- Das Programm muss beim Programmstart seine Konfiguration mittels eines Configsfiles vorunehmen.
- Das Programm kann dem Benutzer die Möglichkeit bieten, die Konfiguration über die GUI vorzunehmen und im Configfile zu speichern.

0.6.2 Leistungsfaktoren

- Der Benutzer soll neue Kategorien hinzufügen, anpassen und löschen können.
- Events sollen durch den Benutzer in Kategorien eingeteilt werden können.
- Der Benutzer soll Events nach Kategorien gruppieren und filtern können.
- Das Programm soll vorkonfigurierte Dienste nutzen können um seine Notifications als SMS und Emails zu versenden.
- Das Programm soll vorkonfigurierte Dienste nutzen können um anstelle einer Notification ein Script laufen zu lassen.
- Die GUI soll Benutzerzentriert und Ergonomisch sein (Wie kann man dies sinvoll als Anforderung formulieren?)

0.6.3 Begeisterungsfaktoren

- Das Programm soll vorkonfigurierte Dienste nutzen können um anstelle einer Notification ein IFTTT Event auszulösen.
- Das Programm soll zusätzlich zu den gängigen Computer Betriebssystem auch auf Android lauffähig sein.
- Events sollen zwischen allen registrieren Geräten eines Registrierten Benutzers synchronisiert werden. Dazu muss der Benutzer in der Lage sein sich und seine Geräte,welche das Programm installiert haben, zu Registrieren.
- Das Programm soll auf den Betriebssystemen, welche eine Notificationinfrastruktur bereitstellen in der Lage sein die Notifications über diese Notificationinfrastruktur vorzunehmen.(BSP KDE)
- Das Programm soll sich vom Loock and Feel in die jeweilige Desktop Environment anpassen.

0.6.4 Quellen und Vorgehen

Unsere wichtigste Quelle ist unser Auftraggeber Prof. Furrer. Um die Fragen, welche bei der Erstellung der Anforderungen aufgetaucht sind zu klären, werden wir uns mit ihm treffen. Besondere folgende Fragen gilt es abzuklären:

- Entspricht die bisherige Ausarbeitung der Vorstellung des Auftraggebers.
- vollständigkei der Anforderungen.
- Kategoriserung der Anforderungen.
- Priorisierung der Anforderungen.
- Was soll mit der Eventkategorisierung erreicht werden, welche wieteren Anforderungen ergeben sich daraus.
- Abgrenzung des Programmes gegenüber eines kAlarm, Kalenders und gegenüber des Cronjobs.

Da unser Auftraggeber als Professor der Informatik, Als weitere Diskusionsgrundlage sind wir daran neben diesem Dokument einen Prototyp des GUI zu erstellen.

Die Angegebenen Informationen,insbesonder die abzuklärenden Punkte, sind dementsprechend lediglich eine Work in Progress.

Als weitere Qullen können wir die Dokumentation und Code Basis von kAlarm nutzen. Da wir selber der potentiellen Nutzergrupp des Programmes angehören, könneten wir auch uns selber als Quelle nutzen, als Entwickler hat man jedoch oft einen anderen Blickwinkel als ein normaler Benutzer.

Im sinne einer Agilen Entwicklung wollen wir möglichst rasch präsentierbare Artefakte generieren, damit der Auftraggeber frühstmöglich Einfluss nehmen kann und Fehlentwicklungen vermieden werden. Treffen werden bei bei bedarf mit dem Auftraggeber ausgemacht.

0.6.5 Technische Anforderungen

TODO redundanz wirklich gewollt?

- Das Programm muss auf Linux, Windows und OSx lauffähig sein.
- Das Programm muss Datenbankagnostisch sein.
- Das Programm muss auf dem Buildsystem kompiliert werden können.
- Das Programm muss auf dem Referenzsystem ohne weitere installationen kompiliert werden können.

0.6.6 Qualitätsanforderungen

- 80% der Benuter sollen in der Lage sein ohne vorgänige Schlung, bei der ersten Benutzung, folgend Aktionen, jeweils innert 5 Minuten nach erfolgtem Programmstart, auszuführen:
  - Erstellen eines Events.
  - Löschen eines vorhandenen Events.
  - Editieren eines Events (Zeit, Datum, Notificationtext).
- 60% der Benutzer soll in der Lage sein die Konfiguration des Programmes mittels des ConfigFiles Vorzunhemen.

0.7 Glossar

Abbildungsverzeichnis

Tabellenverzeichnis

Glossar

Android TODO. 12

API TODO Wikipedia. 12

Benutzer TODO Ein Benutzer ist ein Mensch, welcher unser Programm benutzt, falls nicht anders Erwähnt, muss dieser weder Registriert noch Angemeldet sein. 12

Buildystem Linux Mint 17,Java JDK 1.8.1, Intellij 2016.3.5, JavaFx Scene Builder 2.0. 12

CLI TODO Wikipedia. 12

Cronjob TODO. 12

Desktop Environment TODO. 12

Einzelevent TODO Ein Einzelevent, ist ein Event, welches nicht periodisch wiederkehrend stattfindet, Also können auch ein Treffen, weclhes unregelmässig stattfindet wird so auch zu einem Einzelevent. 12

Event TODO Ein Event ist ein Erreigniss, welches zu einem bestimmten Zeitpunkt stattfindet. 12

Filtern TODO. 12

Gruppieren TODO. 12

GUI TODO Wikipdeia. 12

IFTTT IFTTT (die Abkürzung von If This Then That, ausgesprochen „ift“ wie in „Gift“[1]) ist ein Dienstanbieter, der es Benutzern erlaubt, verschiedene Webanwendungen (zum Beispiel Facebook, Evernote, Dropbox usw.) mit einfachen bedingten Anweisungen zu verknüpfen. Wikipedia. 12

JPA Driver TODO. 12

Kategorien TODO . 12

Konfiguration Die Konfiguration kann ein Programm auf die Bedürfnisse des Nutzers anpassen. So kann man Beispielsweise die Spracheinstellung konfigurieren.. 12

Notification TODO Eine Notification ist eine Aktion des Programms, welche den Benutzer auf ein Event aufmerksam machen soll.. 12

Notification Infrastruktur TODO. 12

Programmstart TODO. 12

Referenzsystem Linux Mint 17,Java JRE 1.8.1,Derby10.13.1.1. 12

Registrierter Benutzer TODO. 12

schtask TODO. 12

Standarddatenbank TODO Derby,MySQL . 12

0.8 Anhang

0.8.1 Abstimmung der Anforderungen

0.8.2 Definition of Ready - Checklist

0.9 Versionskontrolle

Manuelle Version: 0.0.2

Automatische Versionierung:
Last compiled: Tue Mar 21 15:14:35 CET 2017
Git HEAD Version: 29