IPA Formatvorlage

autor : Pascal Honegger

version : 0.1

status : Draft

quelle : Atos

dokumentendatum : 31 Oktober 2017

anzahl der seiten : 16

Änderungshistorie

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Datum | Beschreibung | Autor |
| 0.1 | 31.10.2017 | Initiale Version erstellt | Pascal Honegger |
|  |  |  |  |
|  |  |  |  |
| 1.0 | xx.yy.2018 | Erster Release | Pascal Honegger |

# Inhaltsverzeichnis

**Teil 1**

1 Aufgabenstellung 4

2 Projektorganisation 5

2.1 Beteiligte Personen 5

2.2 Projektmanagement 5

3 Ausgangslage 6

3.1 Vorkenntnisse 6

3.2 Vorarbeit 6

3.3 Firmenstandards 6

4 Zeitplanung 7

4.1 Meilensteine 7

4.2 Gantt-Diagramm 8

5 Arbeitsprotokoll 9

5.1 Tag 1 (Datum) 9

**Teil 2**

6 Management Summary 10

6.1 Projektbeschreibung 10

6.2 System-Beschreibung 10

7 Informieren 11

7.1 Ist-Analyse 11

7.2 Soll-Analyse 11

7.3 Use-Case 11

8 Planen 12

8.1 Detailplanung 12

8.2 Architektur 12

8.3 Datenbankdesign 12

9 Entscheiden 13

9.1 Planung finalisieren 13

9.2 Verwendete Technologien und Methoden 13

9.3 Projektstart 13

10 Realisieren 14

10.1 Erweiterte Architektur 14

10.2 Abweichung Planung 14

11 Kontrollieren 15

11.1 Zeitmanagement 15

11.2 Funktionalität 15

11.3 Codequalität 15

11.4 Fertigstellung 15

12 Auswerten 16

12.1 Status Ziele 16

12.2 Verbesserungen 16

12.3 Erfahrungen 16

12.4 Fazit 16

**Anhang**

13 Verzeichnisse 17

13.1 Quellenverzeichnis 17

13.2 Abbildverzeichnis 17

13.3 Tabellenverzeichnis 17

14 Quellcode 18

# Aufgabenstellung

# Aufgabenstellung

## Hintergrund

Der Projektleiter und C#-Teamleiter Christoph Fauti bemüht sich stets um die Zufriedenheit seiner Arbeitskollegen. Deshalb wird das AHM (Atos Happy Meter - Gefühlslage-Messer) als Website realisiert. Die Idee hierbei ist, dass ein Mitarbeiter schnell seine jetzige Gefühlslage festhalten kann und sieht, wie es den Arbeitskollegen geht.

## Funktionale Anforderungen

### Gefühlslage Erfassen

Das Ziel der Website liegt darin, dass ein Benutzer schnell und anonym seine Gefühlslage eintragen kann. Auf der Startseite sieht man eine Auswahl von Smileys (von sehr glücklich bis zu sehr unglücklich). Hier kann man basierend auf seiner jetzigen Gefühlslage einen Smiley drücken. Zusätzlich kann optional ein Kommentar verfasst werden, welcher die eigene Gefühlslage beschreibt. Nach dem Absenden seiner Gefühlslage wird der ausgewählte Smiley und optionale Kommentar zurückgesetzt. Dies erlaubt beispielsweise das Aufsetzen eines dedizierten Laptops im Pausenbereich, auf welchem jeder seine Gefühlslage eintragen kann, ohne die vorherige Auswahl zu sehen. Als ein simpler Spam-Schutz wird das erneute Absenden einer Gefühlslage für eine Minute blockiert.

### Gefühlslage auswerten

Durch herunterscrollen findet man unterhalb der Erfassung eine Übersicht der allgemeinen Gefühlslage. Man sieht die gleichen Smileys, welche man beim Erfassen auswählen kann. Die Grösse der Smileys variiert je nachdem, wie oft dieser an diesem Tag bereits ausgewählt wurde. Diese Daten werden automatisch aktualisiert, ohne dass ein Benutzer die Seite neu laden muss. Zusätzlich zu jedem Smiley wird ein zufälliger Kommentar angezeigt, falls einer für diesen Smiley an diesem Tag existiert.

### Administration

Zusätzlich zu dieser Funktionalität gibt es einen Administrator-Modus. Dieser ist durch ein Login geschützt und für die normalen Benutzer nicht sichtbar. Der Administrator bleibt während der Browser-Session angemeldet und kann sich bei Bedarf abmelden.

Der Administrator kann einen Smiley-Verlauf ansehen. Er kann einen Datumsbereich auswählen und sieht in Form eines Liniendiagramms an welchem Tag welche Smileys wie oft gewählt wurden.

Der Administrator kann die verfügbaren Smileys verwalten. Man kann neue hinzufügen, bestehende bearbeiten und ungebrauchte deaktivieren. Durch das deaktivieren können keine neuen Gefühlslagen mit diesem Smiley erstellt werden, die Auswertungen bleiben trotzdem in der Administration erhalten.

Der Administrator kann seine eigenen Anmeldedaten (Benutzername & Passwort) bearbeiten.

## Technische Anforderungen

* Frontend als Website
  + Unterstützt werden folgende Browser
    - Muss: Google Chrome
    - Muss: Mozilla Firefox
    - Optional: Microsoft Internet Explorer 11
  + Läuft unter Windows 7
* Backend-Server
  + Windows-basierter Server
  + Datenspeicherung in einer relationalen Datenbank
* Admin Login
  + Passwort sicher gespeichert (nicht Klartext)
  + Standard Benutzer ‘admin’ mit dem Passwort ‘admin’

## Erweiterungen

Die Installation auf dem Testsystem ist nicht teil der IPA. Falls die Rückmeldungen nach den ersten Tests positive ausfallen wird das System um eine geografische Gruppierung (Bsp. Stockwerke, Gebäude) erweitert. Diese Funktionalität sollte beim Erstellen der Datenstruktur berücksichtigt werden. Während der Testphase kann dieser Ort für alle Einträge auf einen Test-Wert gesetzt werden.

# Projektorganisation

## Beteiligte Personen

|  |  |  |  |
| --- | --- | --- | --- |
| Position | Name | E-Mail | Telefonnummer |
| Auftraggeber | Christoph Fauti | [christoph.fauto@atos.net](mailto:christoph.fauto@atos.net) | - |
| Fachexperte | Rolf Fux | [rolf.fux@atos.net](mailto:rolf.fux@atos.net) | - |
| Kandidat | Pascal Honegger | [pascal.honegger@siemens.com](mailto:pascal.honegger@siemens.com) | 077 421 51 28 |
| Hauptexperte | Remo Steinmann | [remo.steinmann@siemens.com](mailto:remo.steinmann@siemens.com) | - |
| Zweit-Experte | - | - | - |

Tabelle : Projektteilnehmer

## Projektmanagement

### Gewählte Projektmethode IPERKA

Eigene Grafik zu IPERKA

Struktur vorgegeben, gut für IPA geeignet.

Viel Erfahrungen von der Schule.

### Alternative Scrum

Im Betrieb immer verwendet

Zu hoher Zeitaufwand mit kleinem Mehrwert da feste Meilensteine gesetzt sind & keine Teamkommunikation nötig ist.

# Ausgangslage

Beschreibung Umfeld / Projekt.

## Vorkenntnisse

Pascal Honegger arbeitet seit zwei Jahren mit C# und seit einem Jahr im Web-Bereich.

## Vorarbeit

Grobe Word-Vorlage für die Dokumentation.

## Firmenstandards

Frameworks, OR-Mapper

# Zeitplanung

## Meilensteine

|  |  |  |
| --- | --- | --- |
| Meilenstein | Beschreibung | Datum |
| Projektstart | Start des Projektes | 01.11.2017 |
|  |  |  |
|  |  |  |
|  |  |  |

## Gantt-Diagramm



Tabelle : Zeitplan

# Arbeitsprotokoll

## Tag 1 (Datum)

### Planung

|  |  |  |
| --- | --- | --- |
| Was | Soll-Dauer | Ist-Dauer |
| Ist-Analyse | 2h | 3h |
| Zeitplanung | 2h | 4h |

### Getane Arbeit

Heute waren die Tests zu 50% Rot, was aufgrund von Fehler Y nicht korrigiert wurden. Morgen werde ich Fehler Y und Z korrigieren und wieder im Zeitplan liegen.

### Reflexion

Planung hat wie immer länger gedauert als gedacht.

# Management Summary

## Projektbeschreibung

## System-Beschreibung

# Informieren

## Ist-Analyse

Analyse des vorhandenen Systems.

### Verwendete Software / Hardware für die Arbeit

Bsp. Server, Jenkins, Zugang zu GIT

## Soll-Analyse

Anforderungen genauer analysiert

## Use-Case

### Use-Case X: Beispiel kaufen

|  |  |
| --- | --- |
| Use-Case X.1: Produkt kaufen, welches ich mir leisten kann | |
| Beschreibung | Der Benutzer kann Produkte sofort kaufen, falls er genügend Geld besitzt |
| Ausführender | Normaler Benutzer |
| Vorbedingung | Benutzer ist angemeldet |
| Ergebnis / Nachbedingung | Produkt ist im Warenkorb |
| Ablauf | |  |  | | --- | --- | | Schritt |  | |  |  | |  |  | |

Neuer Raum erstellen

Beschreibung

Der User möchte einen neuen Raum erstellen können.

Vorbedingungen

Die Applikation ist gestartet, DB-Connection hergestellt.

Aktion

•Name eingeben

•Auf Plus-Button drücken

•Erstelle Raum in der DB

Nachbedingung

Erfolg:

Ein neuer Raum wurde auf der DB erstellt.

Fehlschlag:

Der Raum wurde nicht erstellt

Invarianten

Raumobjekte

Akteure

User und Server

Auslösendes Ereignis

User möchte Raum erstellen

# Planen

## Testkonzept

Das Testkonzept beschreibt wie nach der Implementation sichergestellt wird, dass die Funktionalität funktioniert. Dazu werden Testfälle definiert und später durchgeführt.

### Rahmenbedingungen

Verwendetes Gerät, Browser, Betriebssystem, Systemsprache, Internetverbindung

### Eingesetzte Testmittel und -Methoden

Wie werden die Blackbox-Tests durchgeführt? (Browser, Private-Browser, SoapUI)

### Testfälle

|  |  |
| --- | --- |
| Testfall X – Beispiel kaufen |  |
| Vorbedingungen | Der Benutzer ist angemeldet |
| Testmittel | Google Chrome |
| Testablauf | |  |  | | --- | --- | | Schritt | Beschreibung | | 1 | Shop öffnen | | 2 | Produkt «Banane» auswählen | | 3 | Auf den Knopf «Kaufen» drücken | |
| Erwartetes Resultat | Beispiel landet im Warenkorb |

## Architektur

UML? Übersicht über das System, keine Details zur Implementation.

## Datenbankdesign

ERM

# Entscheiden

## Planung finalisieren

Vorgehen sinnvoll? Zeitplan realistisch?

## Verwendete Technologien und Methoden

C#, ASP.NET (nicht Core-Variante), Angular 4 (5?) Frontend.

## Projektstart

Start Realisierungsphase

# Realisieren

## Backend

### Klassendiagramm

PlantUML erstelltes Diagramm

### Klassenübersicht

## Frontend

# Kontrollieren

## Zeitmanagement

Wurden Meilensteine erfüllt?

## Funktionalität

Sind Anforderungen vollständig erfüllt? Wurden Tests bestanden?

## Codequalität

Erfüllt die Codequalität die Anforderungen? Bin ich mit der Codequalität zufrieden? (Kommentare, Erweiterbarkeit, Clean-Code)

## Fertigstellung

Projekt finalisieren und abgeben.

# Auswerten

## Reflexion

Wurden alle Ziele erreicht?

## Verbesserungen / Erfahrungen

Welche Verbesserungen am Projekt gibt es? Was würde ich anders machen? Was habe ich gelernt?

## Einsetzbarkeit

Kann es Produktiv eingesetzt werden?

## Fazit

Wie verlief das Projekt?

# Verzeichnisse

## Quellenverzeichnis

**Im aktuellen Dokument sind keine Quellen vorhanden.**

## Abbildverzeichnis

**Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.**

## Tabellenverzeichnis

[Tabelle 1: Projektteilnehmer 4](#_Toc494382107)

[Tabelle 2: Zeitplan 6](#_Toc494382108)

# Quellcode