NoteWriter+

Anforderungsanalyse

|  |  |
| --- | --- |
| **Autor(en):** | Dawid Kapka |
| **Version:** | 1.0 |
| **Erstellt am:** | 3. Dezember 2020 |
| **Letzte Änderung:** | 03.12.2020 |
| **Speicherort:** | Document1 |

Inhalt

[1.1 Systemidee 3](#_Toc57899562)

[Die wichtigsten Funktionen sind: 3](#_Toc57899563)

[1.2 Management Summary 3](#_Toc57899564)

[1.3 Projektteam 3](#_Toc57899565)

[1.4 Glossar 3](#_Toc57899566)

[2 Ausgangslage (IST), Problembereiche 4](#_Toc57899567)

[2.1 Beschreibung der Ausgangslage 4](#_Toc57899568)

[2.2 Problembereiche und Schwachstellen 4](#_Toc57899569)

[3 Ziele (SOLL) 5](#_Toc57899570)

[3.1 Beschreibung der Ziele 5](#_Toc57899571)

[3.2 Produktperspektive, Nutzen 5](#_Toc57899572)

[3.3 Zielkonflikte 5](#_Toc57899573)

[3.4 Abgrenzung 5](#_Toc57899574)

[4 Anforderungsanalyse 6](#_Toc57899575)

[4.1 Identifizierung der Akteure 6](#_Toc57899576)

[4.2 Anforderungskatalog 7](#_Toc57899577)

[A: Funktionale Anforderungen 7](#_Toc57899578)

[NF: Nichtfunktionale Anforderungen 9](#_Toc57899579)

[5 Systemablaufmodelle (Aktivitäten) 10](#_Toc57899580)

[5.1 Aktivität "Text auslesen" 10](#_Toc57899581)

[6 Anhang 11](#_Toc57899582)

[6.1 Termine 11](#_Toc57899583)

Einleitung

## Systemidee

Die Systemidee ist eine Mobile Applikation, welche mithilfe von «CRAFT» (Character Region Awareness for Text Detection) und eines Neuralen Netzes, welches Buchstaben erkennt, ein mit der Smartphone-Kamera gemachtes Bild mit Text drauf zu reinem Text umwandeln kann, damit man so z.B. verschiedene Notizen einlesen und auf dem Smartphone speichern kann. Die Applikation soll in Flutter entwickelt werden, während das Neuronale Netz in Python mithilfe von der PyTorch Library erstellt wird.

### Die wichtigsten Funktionen sind:

* Funktion 1: Möglichkeit, durch die App direkt ein Bild zu machen
* Funktion 2: Möglichkeit, ein bereits erstelltes Bild aus der Galerie auszuwählen
* Funktion 3: Einlesen von Text anhand eines Bildes
* Funktion 4: Speicherung von Text in der App

## Management Summary

Es ist eine Mobile Applikation, welche den Text von einem Bild, welches mit der Smartphone-Kamera erstellt wurde, ausliest und als reiner Text in der App speichert, z.B. als Notizen.

## Projektteam

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Kürzel** | **Funktion** | **Kontakt (Telefon, Mail...)** |
| Urs Nussbaumer | nuu | Projektleiter | 041 371 24 28 urs.nussbaumer@ict-bz.ch |
| Dawid Kapka |  | Developer | 078 402 70 93  dawid\_kapka@sluz.ch |
|  |  |  |  |
|  |  |  |  |

## Glossar

Die im Projekt verwendeten Fachbegriffe und Abkürzungen in alphabetischer Reihenfolge.

|  |  |
| --- | --- |
| **Begriff** | **Beschreibung** |
| CRAFT | Character Region Awareness for Text Detection |

# Ausgangslage (IST), Problembereiche

In diesem Kapitel wird die Ausgangslage beschrieben und es werden die Problembereiche identifiziert, die sich in der heutigen Situation zeigen.

## Beschreibung der Ausgangslage

Ich habe am 03.Dez.2020 mit dem Projekt angefangen und habe somit als erstes etwas Recherche gemacht, um zu entscheiden, wie das Endprodukt aussehen soll.

## Problembereiche und Schwachstellen

Momentan bestehen keine Probleme oder Schwachstellen, da das Projekt noch nicht implementiert wurde.

# Ziele (SOLL)

In diesem Kapitel werden die übergeordneten Ziele beschrieben, die mit dem zu entwickelnden System erreicht werden sollen.

## Beschreibung der Ziele

Das übergeordnete Ziel ist es, eine Software mit Python (Mithilfe von Pytorch und CRAFT) zu erstellen, welche ein Bild als Input erwartet und aus dem Bild den Text auslesen kann. Das Bild soll zuerst in der Mobile App mit der Kamera gemacht werden und so als Input für das Netz weitergegeben werden. Als Output gibt das Netz den Text aus, welcher danach wieder in die App überwiesen wird und dort gespeichert werden kann.

## Produktperspektive, Nutzen

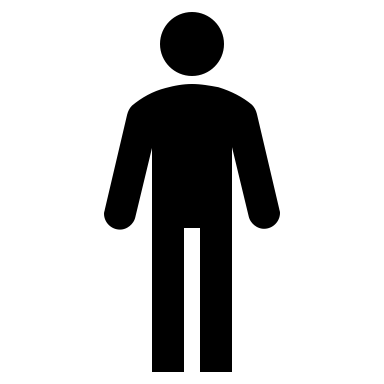
Wenn das Projekt beendet wird, soll der Benutzer die Möglichkeit haben, den Text aus einem beliebten Bild auszulesen und als reinen Text abzuspeichern. Es soll quasi eine «Scan-Funktion» für Text sein, der jedoch nicht nur aus einem Blatt Papier ausgelesen sein soll, sondern aus einer beliebten Fläche, indem sie nur fotografiert wird.

# Anforderungsanalyse

Die Anforderungen an das zu entwickelnde System definieren alle zu erfüllenden Eigenschaften oder die zu erbringende Leistung, sowie allfällige technische Vorgaben und weitere Forderungen des Kunden im Zusammenhang mit dem (den) zu erstellendem Produkt(en).

## Identifizierung der Akteure

* **Kunde:** Benutzer mit der Berechtigung, Notizen zu erfassen und ein Bild zu machen



Kunde

NoteWriter+

## Anforderungskatalog

Der Anforderungskatalog ist eine priorisierte Liste, die alles enthält, was im zu entwickelnden Produkt enthalten sein soll. Es wird unterschieden zwischen funktionalen und nicht-funktionalen Anforderungen.

Jede Anforderung ist in Alltagssprache, in der Form einer "User-Story", formuliert, mit einer ID zur späteren Identifizierung versehen sowie mittels 3 verschiedener Kategorien priorisiert: 1 = hohe Prorität, 2 = mittlere Priorität, 3 = keine Priorität. Diese drei Prioritäten repräsentieren die Verpflichtungen "must", "should" und "nice to have".

Im Verlauf der Zeit können neue Anforderungen hinzukommen und/oder bestehende Anforderungen können wegfallen. Anforderungen, die wegfallen, sind im Dokument zu belassen und als ~~gestrichen~~ zu markieren.

Hinter einer Anforderung kann ihn eckigen Klammern in der Form [Z#] eine Zusatzinformation hinterlegt werden und mit [F#] bzw. [F#, F#, …] können eine oder mehrere offene Fragen referenziert werden.

### A: Funktionale Anforderungen

|  |  |  |
| --- | --- | --- |
| **ID** | **Anforderung** | **Prio** |
| A001 | Als Benutzer möchte ich eine Mobile Applikation zur Verfügung haben. | 1 |
| A002 | Als Benutzer möchte ich die Möglichkeit haben, in der App ein Foto zu machen. | 1 |
| A003 | Als Benutzer will ich als Output den Text bekommen, der sich auf dem Bild befand. | 1 |
| A004 | Als Benutzer will ich, dass die App benutzerfreundlich und einfach zu bedienen ist. | 2 |
| A005 | Als Benutzer will ich, dass der Text in der App angezeigt wird und ich so Notizen erstellen kann. | 2 |
| A006 | Als Benutzer möchte ich die Möglichkeit haben, den Text nach dem Ablesen zu bearbeiten, oder weiteren Text dazuzuschreiben. | 2 |
| A007 | Als Benutzer will ich, dass alle meine Notizen auf meinem Smartphone abgespeichert werden. | 3 |
|  |  |  |
|  |  |  |

### NF: Nichtfunktionale Anforderungen

|  |  |  |
| --- | --- | --- |
| **ID** | **Anforderung** | **Prio** |
| NF001 | Es soll ein Neuronales Netz erstellt werden, welches Buchstaben auf Bildern erkennen kann | 1 |
| NF002 | Das Netz soll schlussendlich fähig sein, ganze Wörter zu erkennen und in eine .csv Datei abzuspeichern | 1 |
| NF003 | Das gemachte Foto soll mithilfe einer API als Input für die Schrifterkennung weitergegeben werden. | 1 |
| NF004 | Der Text auf dem Foto soll möglichst richtig erkannt werden. | 1 |
| NF005 | Die Schrifterkennung soll nicht mehr als 4 Sek. dauern | 2 |
| NF006 | Die Schrifterkennung soll eine Genauigkeit von min. 90% haben. | 2 |
|  |  |  |
|  |  |  |

# Systemablaufmodelle (Aktivitäten)

Dieses Kapitel zeigt die wichtigsten und/oder komplexesten funktionalen Anforderungen in ihrem Ablauf. Dazu werden die einzelnen Aktivitätsschritte detailliert analysiert. Für die Darstellung der einzelnen Aktivitätsschritte werden Aktivitätsdiagramme nach UML verwendet.

todo sofern hilfreich: Aktivitägendiagramme erstellen und hier hinterlegen (Hinweis: Es muss nicht UML sein, Flussdiagramme sind auch OK).

## Aktivität "Text auslesen"

Das folgende Aktivitätsdiagramm zeigt die einzelnen Aktivitäten, die sich aus der Anforderungen A002, A003, A005 und NF001 ergeben.

Diagram

Description automatically generated

Nein

Ja

# Anhang

## Termine

|  |  |  |
| --- | --- | --- |
| **Datum** | **Ziel, Beschreibung, Meilenstein** | **Verantwortlich** |
|  | Meilensteingespräch 1 |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |