Match Planner

Anforderungsanalyse

|  |  |
| --- | --- |
| **Autor:** | Björn Gedeon |
| **Version:** | 1.0 |
| **Erstellt am:** | 3. Dezember 2021 |
| **Letzte Änderung:** | 9. Dezember 2021 |
| **Speicherort:** |  |

Inhalt

[1 Einleitung 3](#_Toc89934821)

[1.1 Systemidee 3](#_Toc89934822)

[Die wichtigsten Funktionen sind: 3](#_Toc89934823)

[1.2 Management Summary 3](#_Toc89934824)

[1.3 Projektteam 3](#_Toc89934825)

[2 Ausgangslage (IST) 3](#_Toc89934826)

[2.1 Beschreibung der Ausgangslage 3](#_Toc89934827)

[2.2 Problembereiche und Schwachstellen 3](#_Toc89934828)

[3 Ziele (SOLL) 4](#_Toc89934829)

[3.1 Beschreibung der Ziele 4](#_Toc89934830)

[3.2 Produktperspektive, Nutzen 4](#_Toc89934831)

[4 Anforderungsanalyse 5](#_Toc89934832)

[4.1 Identifizierung der Akteure 5](#_Toc89934833)

[4.2 Kontextdiagramm 5](#_Toc89934834)

[4.3 Anforderungskatalog 6](#_Toc89934835)

[A: Funktionale Anforderungen 6](#_Toc89934836)

[NF: Nichtfunktionale Anforderungen 7](#_Toc89934837)

[5 Systemablaufmodelle (Aktivitäten) 8](#_Toc89934838)

[5.1 Aktivität "DVD ausleihen" 8](#_Toc89934839)

[5.2 Aktivität «XY» 8](#_Toc89934840)

# Einleitung

## Systemidee

Match Planner ist eine einfache Möglichkeit sich für ein Turnier anzumelden. In meinen Fall mach ich dies für das Schülerturnier in Ebikon. Dieses Programm ersetzt die Anmeldung per Zettel und spart so Zeit für die Organisation.

### Die wichtigsten Funktionen sind:

* Funktion 1: Man soll sich anmelden können.
* Funktion 2: Der Spielplan soll man auf einer anderen Seite auffüllen können.
* Funktion 3: Man soll die anmelde Daten auslesen können.
* Funktion 4: Man soll die Tabelle sehen können.

## Management Summary

## Projektteam

In meinem Projekt erstelle ich eine Website für das das Schülerturnier Ebikon. Man soll sich online al Klasse anmelden können und auch den Spielplan sehen können, sobald sich Teams angemeldet haben.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Kürzel** | **Funktion** | **Kontakt (Telefon, Mail...)** |
| Urs Nussbaumer | nuu | Projektleiter | 041 371 24 28 urs.nussbaumer@ict-bz.ch |
| Björn Gedeon | bjg | Entwickler | 077 457 88 87  bjoern.gedeon@gmail.com |

Die im Projekt verwendeten Fachbegriffe und Abkürzungen in alphabetischer Reihenfolge.

# Ausgangslage (IST)

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

## Beschreibung der Ausgangslage

Ausgangslage beschreiben (IST-Zustand). Wo stehen wir jetzt?

Zurzeit habe ich nur die Anforderungsanalyse geschrieben.

## Problembereiche und Schwachstellen

Welches sind die Schwachstellen der jetzigen Lösung? Welche Probleme bestehen?

# 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, dass man sich online für das Turnier anmelden kann.

Weitere Ziele sind:

* Man soll den Spielplan erstellen können
* Das GUI soll verständlich sein.
* Ich will, dass man eine Tabelle hat, in welcher man die Punkte und Tore sieht die ein Team erstellt hat. Die Tore und Punkte sollen nach jedem Spiel aktualisiert werden.

## Produktperspektive, Nutzen

Man soll sich so einfach anmelden können. Man soll die Resultate und die aktuelen Zwischenstände online ansehen können. Zudem soll auch der Spielplan ersichtlich sein.

# 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 erstellenden Produkt(en).

## Identifizierung der Akteure

Ein Akteur bezeichnet eine Rolle, die mit dem System interagiert. Es werden 2 Akteure unterschieden:

* **Angemeldeter Organisator:** Als angemeldeter Organisator, will ich die Anmeldedaten entgegennehmen können und den Spielplan ausfüllen können. Ich soll den Spielplan und die Rangliste sehen können.
* **registrierter Benutzer:** Der Benutzer soll sich als angemeldeter Benutzer für das Turnier anmelden können, er soll den Spielplan und die Rangliste sehen.
* **nicht registrierter Benutzer:** Als Benutzer soll ich den Spielplan und die Rangliste sehen können.

## Kontextdiagramm

Die angemeldeten Benutzer sind die, welche sich für das Turnier anmelden. Die Benutzer sind die, die den Spielplan und die Rangliste anschauen wollen. Der angemeldete Organisator will die Anmeldedaten auslesen können und Spielplan und Rangliste auffüllen können.

## 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 will ich mich registrieren können. | 1 |
| A002 | Als registrierter Benutzer möchte ich mich für das Turnier anmelden können. | 1 |
| A003 | Als Organisator möchte ich mich anmelden können, um die Anmeldedaten auslesen zu können. | 1 |
| A004 | Als angemeldeter Organisator möchte ich die Anmeldedaten auslesen können. | 1 |
| A005 | Als angemeldeter Organisator möchte den Spielplan erstellen können | 11 |
| A006 | Als Benutzer möchte ich den Spielplan sehen können. | 1 |
| A007 | Als Benutzer möchte ich die Rangliste sehen können. | 2 |
| A008 | Als Organisator möchte ich eine Ausschreibung starten und beenden können. [Z# Ich möchte die Ausschreibung an einem bestimmten Datum starten können. Und die Ausschreibung an einem bestimmten Datum beenden können.] | 2 |

### NF: Nichtfunktionale Anforderungen

|  |  |  |
| --- | --- | --- |
| **ID** | **Anforderung** | **Prio** |
| NF001 | Man darf keinen Code einschliessen können. | 1 |
| NF002 | Personen, die nicht zu den Organisatoren gehören, dürfen keinen Zugriff auf die Namen der Teilnehmer haben. | 1 |
| NF003 | Der Code soll strukturiert und dokumentiert sein. | 2 |
| NF004 | Das Programm soll nicht zulassen, dass sich die gleiche Klasse mehrfach anmelden kann. | 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 es für das Verständnis eines Prozesses/Vorgangs hilfeich: Aktivitägendigramm erstellen und beschreiben. Hinweis: Es muss nicht UML sein, Flussdiagramme sind auch OK. Sonst Kapitel löschen.

## Aktivität "DVD ausleihen"

Das folgende Aktivitätsdiagramm zeigt die einzelnen Aktivitäten, die sich aus der Anforderung F.REQ.002 (siehe Kapitel "‎0") ergeben.

## Aktivität «XY»

todo