|  |
| --- |
| Elmer Lukas, Heidt Christina, Treichler Delia  22. Dezember 2011 |

|  |
| --- |
| Studienarbeit |
| Anforderungen |
|  |

****

# Dokumentinformationen

## Änderungsgeschichte

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Änderung | Autor |
| 26.10.2011 | 1.0 | Erste Version des Dokuments | cheidt |
| 28.10.2011 | 1.2 | Ergänzungen | dtreichl |
| 02.11.2011 | 1.3 | Review | dtreichl |
| 15.11.2011 | 1.4 | Review | lelmer |
| 25.11.2011 | 1.5 | Review | dtreichl |
| 28.11.2011 | 1.6 | Kleine Korrekturen | dtreichl |
| 11.12.2011 | 1.7 | Funktionale Anforderungen, NF Anforderungen | lelmer |
| 12.12.2011 | 1.8 | Kapitel Grundsätzl. Angaben und kleine Korrekturen | dtreichl |
| 16.12.2011 | 1.9 | Tools angepasst | lelmer |

## Inhaltsverzeichnis

[1 Dokumentinformationen 1](#_Toc311821264)

[1.1 Änderungsgeschichte 1](#_Toc311821265)

[1.2 Inhaltsverzeichnis 1](#_Toc311821266)

[1.3 Tabellenverzeichnis 1](#_Toc311821267)

[2 Tools 2](#_Toc311821268)

[3 Funktionale Anforderungen 3](#_Toc311821269)

[4 Nichtfunktionale Anforderungen 7](#_Toc311821270)

[4.1 Funktionalität 7](#_Toc311821271)

[4.1.1 Angemessenheit 7](#_Toc311821272)

[4.2 Zuverlässigkeit 7](#_Toc311821273)

[4.2.1 Fehlertoleranz 7](#_Toc311821274)

[4.3 Benutzbarkeit 7](#_Toc311821275)

[4.3.1 Verständlichkeit & Erlernbarkeit 7](#_Toc311821276)

[4.3.2 Bedienbarkeit 7](#_Toc311821277)

[4.3.3 Attraktivität 7](#_Toc311821278)

[4.4 Effizienz 8](#_Toc311821279)

[4.4.1 Zeitverhalten 8](#_Toc311821280)

[4.5 Änderbarkeit & Wartbarkeit 8](#_Toc311821281)

[5 Design Constraints 9](#_Toc311821282)

[6 Zugänglichkeit (Accessibility) 10](#_Toc311821283)

## Abbildungsverzeichnis

[Abbildung 1 - Startscreen, original 11](#_Toc312321242)

[Abbildung 2 - Startscreen, Deuteranopie 12](#_Toc312321243)

[Abbildung 3 - Startscreen, Protanopie 12](#_Toc312321244)

[Abbildung 4 - Startscreen, Tritanopie 13](#_Toc312321245)

## Tabellenverzeichnis

[Tabelle 1 - User Stories 6](#_Toc311819805)

# Tools

Zur Durchführung des Projektes und Entwicklung der Software wurden die nachfolgend aufgelisteten Werkzeuge verwendet.

* Windows 7 SP1
* SVN Client (Tortoise SVN, <http://tortoisesvn.net/>)
* Adobe Reader X (v10)
* .NET (v4.0.30319 RTMRel)
* Surface 2 SDK (<http://blogs.msdn.com/b/surface/archive/2011/07/12/links-to-sdk-and-resources.aspx>)
* Visual Studio 2010 Ultimate (v10.0.30319.1 RTMRel) mit Power-Tools
  + ReSharper 6.0 und dotCover 1.1.1 (<http://www.jetbrains.com>)
  + GhostDoc 3.0 (<http://submain.com/download/ghostdoc/>)
  + NDepend Trial (3.9, <http://www.ndepend.com/>)
* Expression Blend (v4.0.20525.0, <http://www.microsoft.com/expression/products/blend_overview.aspx>)
* WPF Inspector (v0.9.9, <http://www.wpftutorial.net/Inspector.html>)
* Adobe Photoshop CS4 Extended (<http://www.adobe.com/de/products/photoshop.html>)
* Microsoft Office 2010 (<http://office.microsoft.com/de-ch/>)
* Redmine (<http://redmine.org>)

# Funktionale Anforderungen

Um die funktionalen Anforderungen möglichst effizient und trotzdem exakt zu definieren, wurden User Stories als Teil von Scrum verwendet. Nachfolgend sind die User Stories nach Sprint gruppiert. Die User Stories sind mit dem jeweiligen Sprint, in welchem sie umgesetzt wurden, gekennzeichnet. Nicht umgesetzte User Stories sind mit „U“ markiert. Solche, die durch die fehlende Hardware nicht umgesetzt werden konnten mit „F". Weitere Details sind dem Todo: Excel Dokument „User Stories Project Flip 2.0 Zühlke.xls“ im Anhang zu entnehmen.

Nachfolgend eine Übersicht über die User Stories: Legende: F -> Future, U -> Unplanned

|  |  |  |  |
| --- | --- | --- | --- |
| Titel | User Story | Definition of Done | Sprint |
| Übersicht für PN | Als Surface Benutzer möchte ich die PN in einer Übersicht anzeigen lassen, damit ich mir einen Überblick über die verschiedenen PN verschaffen kann. | Alle PNs werden in einem Gitter dargestellt. | 3 |
| Detailansicht PN | Als Surface Benutzer möchte ich eine Detailansicht der PN sehen, damit ich die einzelnen PN lesen kann und genauere Infos erhalte. | Eine bestimmte PN wird in einem separaten Container dargestellt, wobei die XPS Datei geöffnet sein soll. | 3 |
| Navigation “Übersicht -> Detail” | Als Surface Benutzer möchte ich von der Übersicht über alle PN in eine Einzelansicht wechseln, damit ich die einzelnen PN lesen kann und genauere Infos erhalte. | Durch das Anklicken einer PN in der Übersicht wird die Detailansicht mit genau dieser PN geöffnet. | 3 |
| Navigation “Detail -> Übersicht” | Als Surface Benutzer möchte ich von der Detailansicht zur Übersicht navigieren, damit ich mir einen Überblick über die aktuelle Detailansicht verschaffen kann. | Durch das Anklicken eines zurück-Buttons wird die Detailansicht geschlossen und die Übersicht wird wieder angezeigt. | 3 |
| Darstellung der verkleinerten PN mit Bild | Als Surface Benutzer möchte ich die PN mithilfe eines Bildes unterscheiden können, wenn sie verkleinert sind, um einfacher eine Auswahl zu treffen. | Die PNs werden in einem Gitter mit je einem Bild dargestellt. | 3 |
| Navigation "Detailansicht -> Detailansicht" | Als Surface Benutzer möchte ich bei der Detailansicht einer PN zur nächsten oder vorherigen PN wechseln können, damit ich alle PN nacheinander anschauen kann. | In der Detailansicht kann auf einen Button "nach rechts" oder "nach links" geklickt werden, wobei die aktuelle PN mit der rechten bzw. linken PN ausgetauscht wird. | 3 |
| Tags zu PN angezeigt | Als Surface Benutzer möchte ich zu einer PN zusätzliche Begriffe anzeigen können, damit ich die PN einem bestimmten Gebiet zuordnen kann. | Tags, die zu einer PN gehören, werden in der Detailansicht zur PN angezeigt. | 4 |
| Tags aggregiert | Als Surface Benutzer möchte ich bei den Tags statt vielen spezifischen Tags eine kleinere Auswahl von aggregierten Tags sehen, um die Übersicht zu behalten und nicht zu viele Elemente auf dem Bildschirm zu sehen. (Bsp: Java ME, Java EE, Java Beans -> wird aggregiert zu Java) | Die Tags werden aggregiert angezeigt. | 4 |
| Aggregierte Tags anpassbar | Als Surface Administrator möchte ich die Zuordnungen der Tags bearbeiten können, damit ich neue Tags in eine Oberkategorie aggregieren kann und damit ich die aggregierten Elemente ändern kann. | Config File kann von einem Admin bearbeitet werden. | 4 |
| Filter in Übersicht setzen | Als Surface Benutzer möchte ich einen Filter in der Übersicht setzen können, damit die angezeigten PN eingeschränkt werden. | In der Übersicht kann ein Filter gesetzt werden, sodass alle PN nach dem gesetzten Filter gefiltert werden. Die tatsächliche Auswahl des Filters ist hier nicht inbegriffen. | 4 |
| Filterkriterium auswählen | Als Surface Benutzer möchte ich das Filterkriterium aus einer nach Kategorien gruppierten Liste auswählen können, damit ich den gesuchten Begriff einfach finden kann. | Ein Filterkriterium kann aus einer Menge von Filterkriterien ausgewählt werden. | 4 |
| Filter in Übersicht entfernen | Als Surface Benutzer möchte ich einen Filter in der Übersicht entfernen können, damit mir mehr PN angezeigt werden. | Ein zuvor gewähltes Filterkriterium kann wieder entfernt werden, wobei die PN nicht mehr nach diesem Filterkriterium gefiltert werden. | 4 |
| Filter in Detailansicht entfernen | Als Surface Benutzer möchte ich einen Filter in der Detailansicht entfernen können, damit ich dazu nicht zuerst zur Übersicht wechseln muss. | In der Detailansicht kann der Filter entfernt werden. | 4 |
| Externes Design festgelegt und validiert | Als Surface Benutzer möchte ich Als Entwickler möchte ich für die Design User Stories eine "Definition of Done" festlegen können, damit der Abschluss der User Stories validiert werden kann. | Das externe Design wurde im Photoshop erstellt und dem Kunden gezeigt, der damit zufrieden war. | 4 |
| Filter in Detailansicht setzen | Als Surface Benutzer möchte ich einen Filter in der Detailansicht setzen können, damit ich dazu nicht zuerst zur Übersicht wechseln muss. | In der Detailansicht kann ein Filter gesetzt werden, wobei nach dem Setzen des Filters zur Übersicht gewechselt wird. | 5 |
| Filtern nach mehreren Kriterien | Als Surface Benutzer möchte ich nach mehreren bestimmten Kriterien filtern können, um eine begrenzte Auswahl zu erhalten. | Es können mehrere Filter gleichzeitig eingegeben werden, wobei nach allen gefiltert wird (AND). | 5 |
| Animierte Navigation “Übersicht -> Detail” | Als Surface Benutzer möchte ich durch eine Animation erkennen können, dass von der Übersicht in die Detailansicht (oder in die Gegenrichtung) gewechselt wurde, damit der Programmfluss für mich verständlicher ist. | Der abgedunkelte Hintergrund kann angeklickt werden, wodurch zur Übersicht gewechselt wird. | 5 |
| Scrollerkennung in Übersicht dargestellt | Als Surface Benutzer möchte ich an der Darstellung der Liste erkennen können, dass durch die PN gescrollt werden kann. | In der Übersicht existiert ein grafisches Element, welches auf mehr Project Notes hindeutet. | 5 |
| Schöne Darstellung der Tags | Als Surface Benutzer möchte ich eine ansprechende Darstellung der Begriffe, damit mir die Applikation besser gefällt und die Bedienung mehr Spass macht. | Die Tags werden gemäss externem Design dargestellt. | 5 |
| Schöne Darstellung des Filters | Als Surface Benutzer möchte ich eine ansprechende Darstellung des Filters, damit mir die Applikation besser gefällt und die Bedienung mehr Spass macht. | Der Filter wird gemäss dem externen Design dargestellt | 5 |
| Lesemodus PN anzeigen | Als Surface Benutzer möchte ich eine PN vergrössern können und einen abgedunkelten Hintergrund sehen, damit ich sie besser lesen kann. | Die PN kann so weit verbreitert werden, wie neben dem Filter und den Tags noch Platz ist. | 6 |
| Animation für Navigation "Detailansicht -> Detailansicht" | Als Surface Benutzer möchte ich eine Animation sehen können, wenn ich durch die PNs navigiere, damit es für mich besser ersichtlich ist, dass die Project Note gewechselt hat. | Es existiert eine Animation, wenn von der einen Detailansicht in die andere Detailansicht gewechselt wird. | 6 |
| Anzahl PN bei Übersicht anzeigen | Als Surface Benutzer möchte ich die Anzahl PN in der Übersicht sehen können, damit ich weiss, wie gross die Liste ist. | Oberhalb des Filters wird dargestellt, wie viele von der maximalen Anzahl an Project Notes aktuell in der Übersicht angezeigt werden. | 6 |
| Easteregg | Als Entwickler oder Zühlke Engineering AG Mitarbeiter möchte ich ein Easteregg ausführen können, damit ich Spass haben und Kunden beeindrucken kann. | Info View für Beteiligte Parteien & Personen erscheint beim Klicken des Info Buttons. | 6 |
| Animation Filter -> Übersicht | Als Surface Benutzer möchte ich eine Animation sehen, wenn die Ansicht geschlossen und der Filter gesetzt wird, damit ich verstehe, dass dieser nun gesetzt ist und sich die Übersicht dementsprechend angepasst hat. | Wenn ein Filterkriterium gesetzt wird, so wird es eine Animation ausgelöst. | 6 |
| Animation Filterkriterium | Als Surface Benutzer möchte ich, dass sich die Unterkategorie einblendet, wenn ich eine Oberkategorie auswähle, damit mir die Applikation besser gefällt und die Bedienung mehr Spass macht. |  | F |
| PN ausgedruckt | Als Surface Benutzer möchte ich eine PN ausdrucken können, um sie mitzunehmen und später genau lesen zu können. |  | F |
| Badge erkennen | Als Surface Benutzer möchte ich einen Besucher Badge auf den Surface legen können, welcher dann automatisch erkannt wird und mir entsprechende Informationen anzeigt. |  | F |
| PN verschickt | Als Surface Benutzer möchte ich eine PN an eine Emailadresse schicken können, um sie später genau lesen zu können. |  | F |
| Animation für aktionslose Beschriftung | Als Surface Benutzer möchte ich beim Drücken einer aktionslosen Beschriftung Feedback erhalten, damit ich erkenne, dass keine Aktion hinter der Beschriftung steht. |  | U |
| Volltextsuche | Als Surface Benutzer möchte ich mithilfe einer Volltextsuche PNs filtern können, damit ich nur diese anzeigen lassen kann, die mich wirklich interessieren. |  | U |
| Demomodus erstellt | Als Surface Benutzer möchte ich durch einen Demomodus auf die Applikation aufmerksam gemacht werden, damit ich animiert werde, den Surface zu bedienen. |  | U |
| Demomodus erkennen und verlassen | Als Surface Benutzer möchte ich den Demo Modus erkennen, damit ich diesen verlassen kann, um selbst navigieren zu können. |  | U |
| Animation für Lesemodus | Als Surface Benutzer möchte ich eine Animation sehen, wenn ich in den Lesemodus wechsle, damit mir die Applikation mehr Spass macht. |  | U |
| Animation Übersicht -> Filter | Als Surface Benutzer möchte ich eine Animation sehen können, wenn die Kriterienliste für das Filtern geöffnet wird, damit ich den Filtermodus besser erkenne. |  | U |

Tabelle 1 - User Stories

# Nichtfunktionale Anforderungen

Die nichtfunktionalen Anforderungen lassen sich zum Teil aus den User Stories ableiten. Einige Anforderungen, wie z.B. die Wartbarkeit, können jedoch nicht daraus abgeleitet werden. Deshalb ist es notwendig, diese in diesem Dokument festzuhalten.

## Funktionalität

### Angemessenheit

Die Zeit, während der ein Anwender die Applikation benutzt, ist kurz. Die Software soll für ihren Zweck der Bereitstellung von Informationen über bisherige Projekte der Zühlke Engineering AG und auch zur Unterhaltung während einer kurzen Zeitdauer gut geeignet sein. Darum soll sie eine Funktionalität mit einer begrenzten Anzahl an Funktionen zur Verfügung stellen (browsen, filtern, lesen). Die Angemessenheit wird durch einen Usability Tests gemessen und verifiziert.

## Zuverlässigkeit

### Fehlertoleranz

Das System soll so aufgebaut sein, dass Fehleingaben nicht möglich sind. Wenn nach mehreren Kriterien gefiltert wird und keine Project Note diesen Kriterien entspricht, so soll der Benutzer auf diesen Umstand aufmerksam gemacht werden. Es muss zudem möglich sein, zur vorherigen Situation zurückzukehren. Dies wird durch heuristische Evaluation sichergestellt.

## Benutzbarkeit

### Verständlichkeit & Erlernbarkeit

Durch die begrenzte Zeit, die der Nutzer vor der Applikation verbringt, muss diese einfach benutzbar sein. Die Bedienung soll ohne Benutzerhandbuch oder anderweitige Hilfe möglich sein. Viel eher soll der Nutzer mit jeder Berührung des Tisches etwas Neues lernen und dieses sogleich anwenden können.

Ein technisch affiner Benutzer soll während des Usability-Tests ohne zusätzliche Hilfe die Bedienung der Software erlernen können.

### Bedienbarkeit

Die Bedienung soll einfach und intuitiv sein. Die einzelnen Komponenten sollen einfach erreichbar sein und über eine genügend grosse Fläche verfügen, damit sie anklickbar sind.

In einem Usability Test werden verschiedene Szenarien vorgegeben, die dann von Testpersonen ohne Erfahrung mit der Applikation und ohne ein Benutzerhandbuch durchgeführt werden. Es müssen alle Szenarien erreicht werden können, ohne dass die Testpersonen Hilfe beanspruchen müssen.

### Attraktivität

Durch das externe Design soll sofort ersichtlich sein, dass es sich um eine Applikation der Zühlke Engineering AG handelt. Dies wird durch das Übernehmen des Corporate Designs sichergestellt.  
Durch eine Fernwirkung, beispielsweise durch einen Demomodus, soll der Nutzer auf die Applikation aufmerksam gemacht und angelockt werden. Eine Nahwirkung wird durch verschiedene Animationen bei der Bedienung gewährleistet. Diese sollen auf den Nutzer ansprechend wirken und ihn dadurch dazu verführen, sich mit der Anwendung auseinanderzusetzen. Gerade bei kurzer Nutzung ist der erste Eindruck entscheidend.

Das Visuelle Design wurde von Zühlke validiert und akzeptiert.

## Effizienz

### Zeitverhalten

Durch den begrenzten Zeitrahmen soll es einem technisch affinen Benutzer innerhalb von drei Minuten möglich sein, ein gewünschtes Projekt oder einen gewünschten Themenbereich zu finden um sich mit diesem auseinandersetzen zu können.

## Änderbarkeit & Wartbarkeit

Die Software soll für die Zühlke Engineering AG möglichst einfach änderbar sein. Es wird deshalb auf die Codequalität geachtet, wobei ReSharper genutzt wird, um die Qualität zu prüfen (orange Markierungen auf der rechten Seite des Editors). Es sollen im Schnitt maximal drei solche orangen Markierungen pro C# Datei (.cs) erscheinen.

Zusätzlich sollen die Code Metriken beachtet werden. Ziel ist es, einen „Maintainability Index“ [microsoft11.2] von mindestens 50% zu erreichen, und zwar auf Ebene Projekt. Ausnahmen können mündlich begründet werden, speziell bei den Testprojekten.

Um die Software möglichst einfach warten zu können, sollen die Kategorien dynamisch anpassbar sein. Dafür soll ein Tab-getrenntes Textdokument zur Konfiguration der anzuzeigenden Tags bereitgestellt werden. Das Textfile soll in ein Tabellenkalkulationsprogramm (z.B. Excel) importiert und nach den Änderungen wieder in ein Textdokument abgespeichert werden können.

# Design Constraints

Vorgegeben ist als Hardware der Surface 2.0 Tisch der Microsoft Corporation. Durch die Verwendung dieser Hardware ist auch die Software-Plattform automatisch auf .Net beschränkt.

Die Zühlke Engineering AG wünscht sich, dass möglichst wenig Aufwand für den Unterhalt des Tisches nötig sein wird. Daher muss mit Project Notes im Originalformat PDF gearbeitet werden.

Die Kategorien und deren zugehörige Begriffe, die zu einer Project Note gehören, sind ebenfalls über die Metadaten vordefiniert. Diese können von der Applikation aggregiert oder gegebenenfalls ignoriert werden.

Das Corporate Design gibt zudem die Richtlinien für das externe Design vor.

# Zugänglichkeit (Accessibility)

Der Surface Tisch ist ein Multitouch Gerät und wird daher mit den Händen bedient. Die Applikation ist deshalb nicht darauf ausgerichtet, mit einer Tastatur oder sonstigen Eingabegeräten benutzbar zu sein.  
Eines der Hauptmerkmale von Project Flip 2.0 ist das Lesen einer Project Note. Durch diese zentrale Rolle kann die Applikation nicht von Menschen mit einer Sehbehinderung verwendet werden. Zwar könnte eine Project Note mit einer Vorlese-Anwendung vorgetragen werden, aber die Navigation kann damit nicht gelöst werden. Zudem würde dies andere Wartende oder das Empfangspersonal ablenken.

Personen mit partiell-funktionalen Sehbehinderungen wie Farbenblindheit soll es hingegen möglich sein, die Applikation zu verwenden.

Als Ausgangslage wurde der Startscreen, welcher im externen Design definiert wurde (Siehe TODO: Externes Design), verwendet.

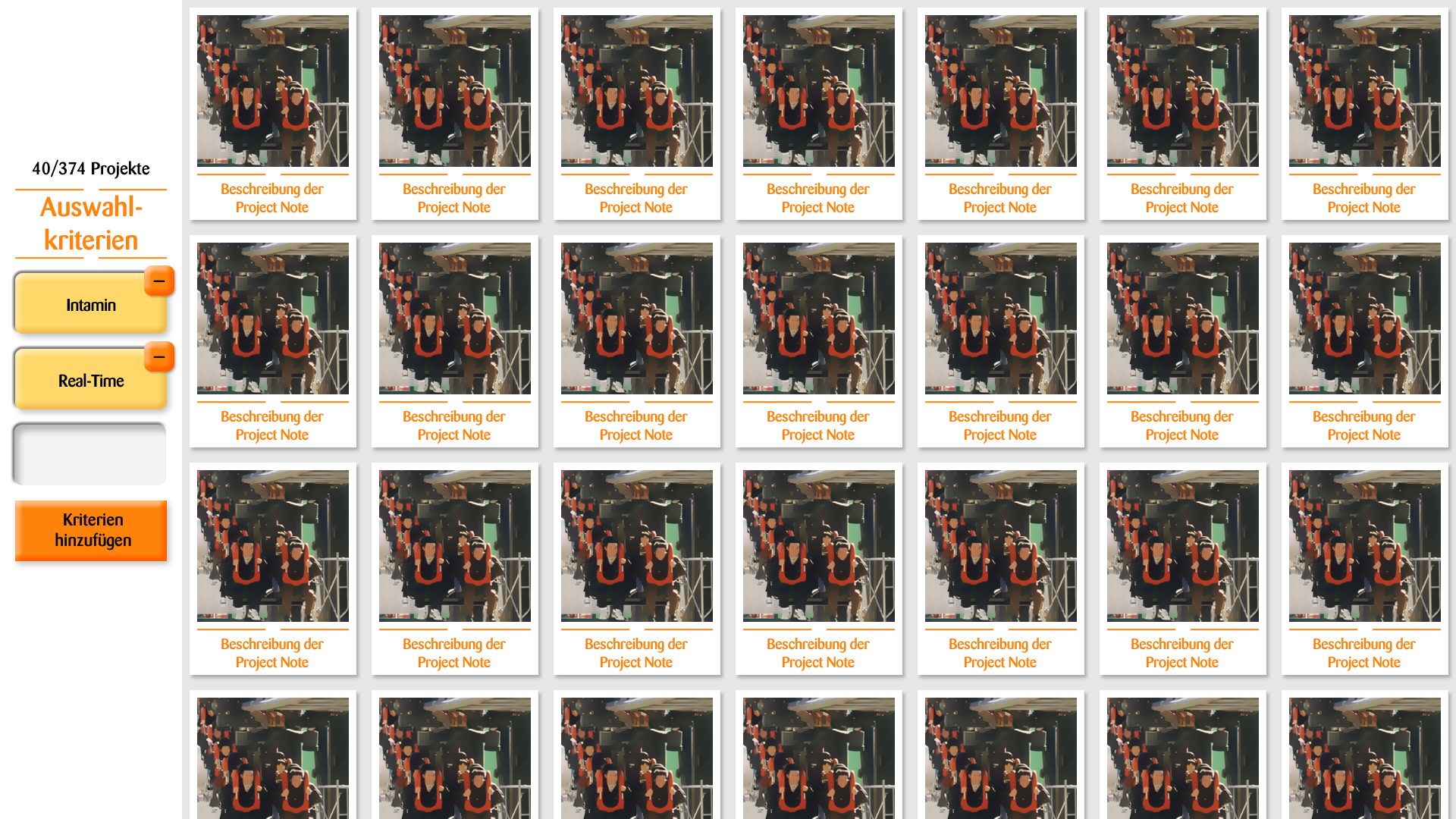


Abbildung 1 - Startscreen, original

Dieser wurde nachfolgend mit Vischeck[[1]](#footnote-1) geprüft. Die Webseite bietet ein Plugin an, welches drei Arten von Farbenblindheit simuliert:

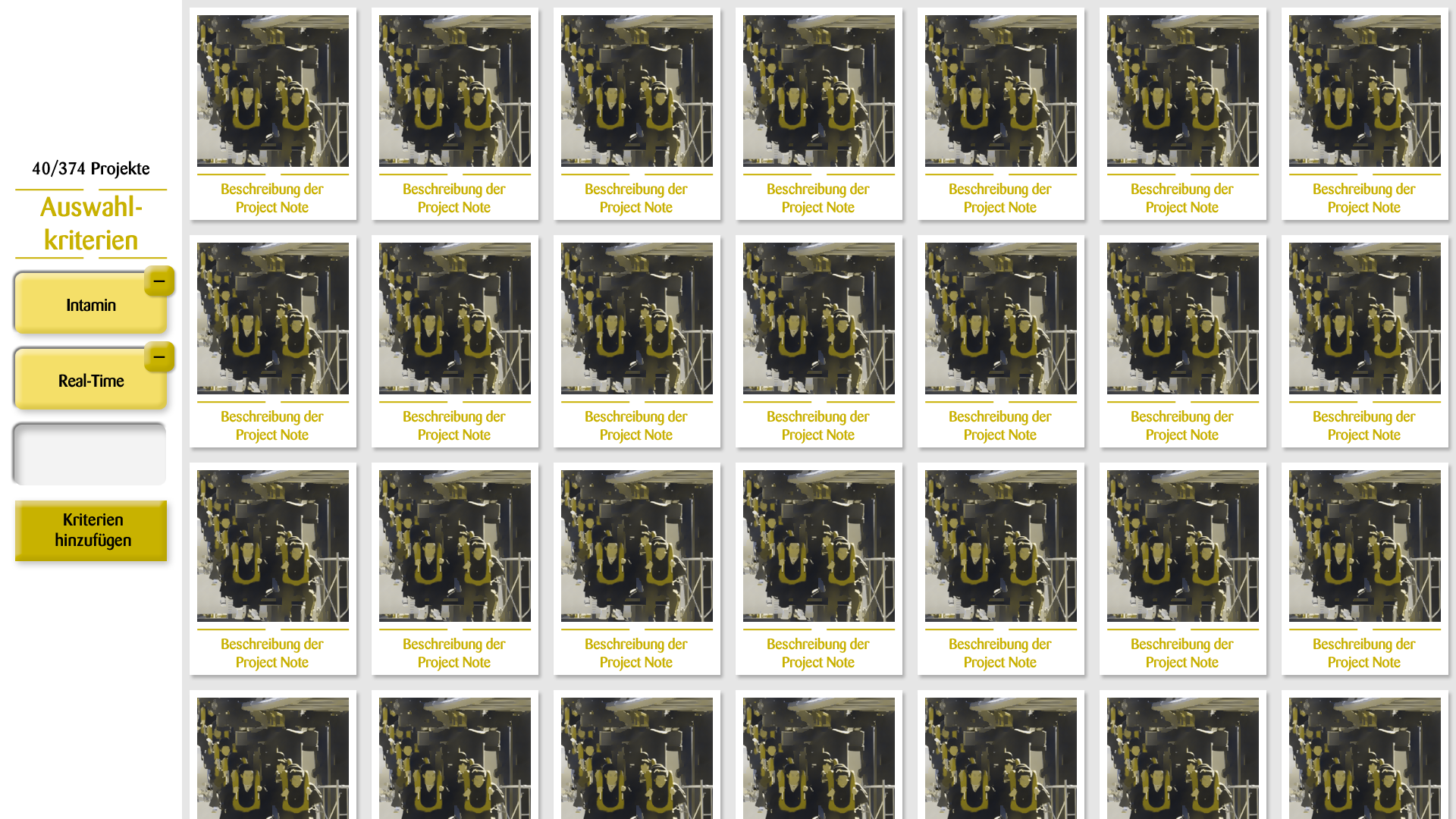
1. Deuteranopie oder Grünblindheit  
   

Abbildung 2 - Startscreen, Deuteranopie

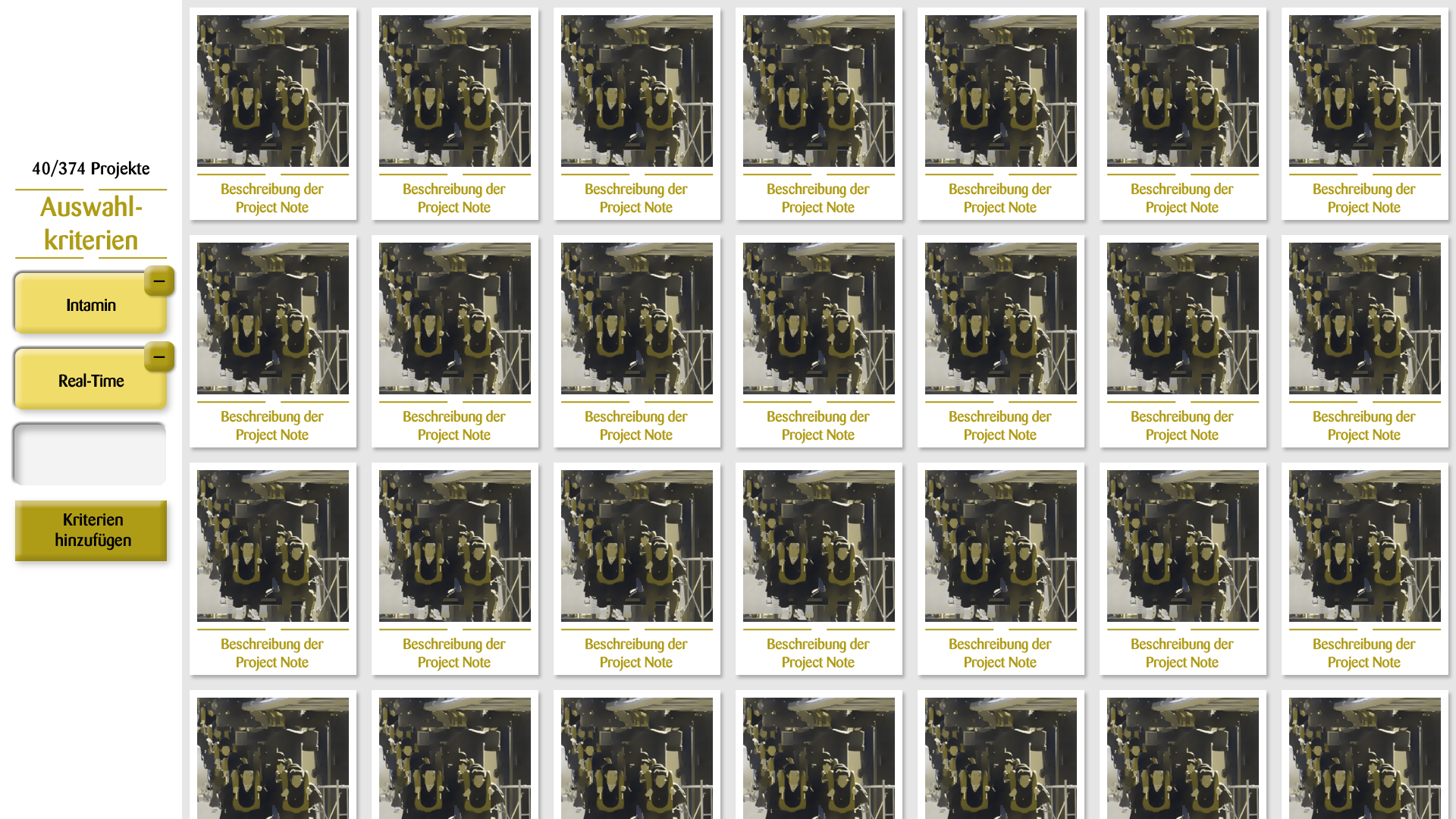
1. Protanopie oder Rotblindheit. Diese Form ist auch bekannt als Rotgrünblindheit erster Form  
   

Abbildung 3 - Startscreen, Protanopie

1. Tritanopie oder Blaublindheit.  
   

Abbildung 4 - Startscreen, Tritanopie

Wie aus den Abbildungen ersichtlich, ist der Kontrast immer noch genug hoch, so dass Texte lesbar und die einzelnen Elemente weiterhin erkennbar sind.

1. Quelle: http://www.vischeck.com/ [↑](#footnote-ref-1)