

**Projekt Softwaretechnik Sommersemester 2017  
BlitzEdit**



|  |  |  |
| --- | --- | --- |
| **Christian Gärtner**  Matrikelnr.:  @hs-esslingen.de |  | **David Schick**  Matrikelnr.: 750838  dascit12@hs-esslingen.de |
| **Nico Pfaff**  Matrikelnr.: 751854  nipfit00@hs-esslingen.de |  | **Marcel Weller**  Matrikelnr.: 752369  maweit06@hs-esslingen.de |

# **Inhaltsverzeichnis**

[Inhaltsverzeichnis 2](#_Toc487884026)

[Abbildungsverzeichnis 3](#_Toc487884027)

[1 Team und Projekt 4](#_Toc487884028)

[2 Marktanalyse 5](#_Toc487884029)

[3 Anforderungen 6](#_Toc487884030)

[3.1 Funktionale Anforderungen 6](#_Toc487884031)

[3.1.1 Schaltplan 6](#_Toc487884032)

[3.1.2 Bauteile 6](#_Toc487884033)

[3.1.3 Speichern 7](#_Toc487884034)

[3.1.4 Laden 7](#_Toc487884035)

[3.2 Nicht-Funktionale Anforderungen 8](#_Toc487884036)

[3.2.1 Design der Anwendung 8](#_Toc487884037)

[3.2.2 Bedienung 8](#_Toc487884038)

[3.2.3 Code 8](#_Toc487884039)

[3.2.4 Bedienen der Bauteile 9](#_Toc487884040)

[3.2.5 Beschriftung 9](#_Toc487884041)

[3.2.6 Bibliothek 9](#_Toc487884042)

[3.2.7 Sonstiges 9](#_Toc487884043)

[4 Ausblick 10](#_Toc487884044)

[4.1 Ideen für Erweiterungen 10](#_Toc487884045)

[4.1.1 Komponenten 10](#_Toc487884046)

[4.1.2 Eigenschaften 10](#_Toc487884047)

[4.1.3 Simulation 10](#_Toc487884048)

[4.1.4 Design 11](#_Toc487884049)

**Einleitung:**

* **Teamvorstellung**
* **Projektvorstellung**

**Marktanalyse**

**Anforderungen**

**Lösungen**

* **Techniken**
  + **Java**
  + **JavaFX (Scene Builder)**
  + **Komponenten als SVGs**
  + **Speichern in XML**
  + **Maven**
  + **Corepackage**
* **Projektmanagement**

**Ausblick**

# **Abbildungsverzeichnis**

[Abbildung 1 - Beispiel eines einfachen Schaltplans (Erstellt in BlitzEdit) 5](#_Toc487882863)

# **Team und Projekt**

* 1. **Das Team**

Das Team besteht aus 4 Mitgliedern aus dem Studiengang Softwaretechnik und Medieninformatik an der Hochschule Esslingen:

Christian Gärtner, Medieninformatik

Nico Pfaff, Medieninformatik

David Schick, Softwaretechnik

Marcel Weller, Softwaretechnik

* 1. **Das Projekt**

Das Projekt ist Inhalt der Veranstaltung „Projekt Softwaretechnik“ und entsteht im Auftrag und unter der Betreuung der IT-Designers GmbH.

Gegenstand des Projekts ist die Entwicklung eines Editors für elektrische Schaltpläne.

Darunter wird eine grafische Darstellung verstanden, die komplexe Schaltungen vereinfacht und somit zur Planung und Wartung einer solchen dient.  
Elektrische Bauelemente, wie zum Beispiel Spannungsquellen oder Widerstände, werden durch vordefinierte abstrakte Symbole dargestellt und durch Leitungen logisch miteinander verbunden (s. Abb. 1).

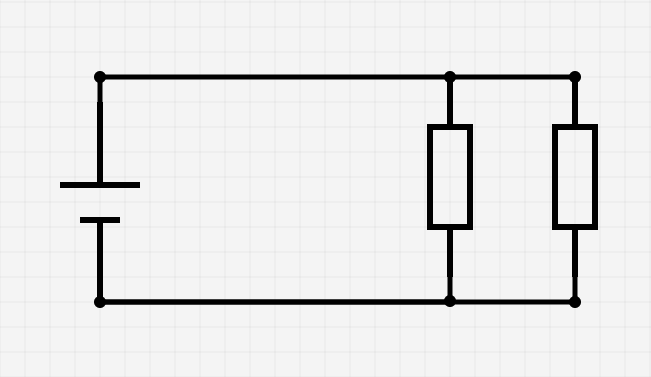


Abbildung 1 - Beispiel eines einfachen Schaltplans (Erstellt in BlitzEdit)

# **Marktanalyse**

# **Anforderungen**

Aus Kundengesprächen und der Marktanalyse ergeben sich folgende Anforderungen an das Produkt:

* 1. **Funktionale Anforderungen**

Die Funktionalen Anforderungen beschreiben, was das Produkt bewerkstelligen muss.

* + 1. **Schaltplan**
* Das System muss es ermöglichen, Schaltpläne zu erstellen, verändern und zu speichern bzw. zu laden.
* Die Schaltpläne bestehen aus Leitungen und über diese verbundenen Bauteile.
* Die Leitungen müssen die Bauteile über rechtwinklig, wenn notwendig auch über 45 Grad, verlaufende, automatisch platzierte Leiterbahnen verbinden.
* Das System muss mehrere Schaltpläne parallel zueinander öffnen und bearbeiten können.
  + 1. **Bauteile**
* Sie müssen frei bewegbar sein, Leitungen müssen sinnvoll mitverschoben werden.
* Die Bauteile müssen in 45 Grad Winkeln drehbar sein.
* Man muss sie kopieren und einfügen können (auch über Tabs hinweg).
* Sie verfügen über beliebig viele Connectoren, an denen die Leitungen anknüpfen.
* Jeder Connector hat eine relative Position zum Bauteil.
* Jeder Connector kann verlängert und wieder verkleinert werden.
* Jedes Bauteil besitzt eine Grafik im SVG-Format.
  + 1. **Speichern**
* Sowohl die Schaltpläne als auch die Bauteile müssen im XML-Format gespeichert werden.
* Um einen Schaltplan zu speichern, muss:
  + die Position der Bauteile gespeichert werden,
  + die Rotation der Bauteile gespeichert werden,
  + die Verbindungen zwischen den Connectoren der Bauteile gespeichert werden.
* Um ein Bauteil zu speichern, muss:
  + die relative Position der Connectoren gespeichert werden,
  + der Typ des Bauteils benannt werden,
  + die Grafik, die dem Bauteil zugeordnet ist, gespeichert werden.
* Es muss feststellbar sein, ob ein Teil des Schaltplans seit der letzten Sitzung verändert wurde. Ist dies der Fall muss der Benutzer darüber in Kenntnis gesetzt werden.
  + 1. **Laden**
* Zur Laufzeit des Systems müssen neue Bauteile erstellt und geladen werden können.
* Bestehende Bauteile müssen durch Änderungen an den XML-Dateien verändert werden können.
  1. **Nicht-Funktionale Anforderungen**

Die Nicht-Funktionalen Anforderungen beschreiben, wie das Produkt funktionieren soll.

* + 1. **Design der Anwendung**
* Das Design des Produkts soll an das der Microsoft Office Produkte angelehnt sein.
* Die Menüleiste der Anwendung soll eine klare Struktur aufweisen.
* Das Design aller Fenster soll einheitlich gestaltet werden (zum Beispiel Popups).
* Das Anwendungsdesign soll dem Benutzer einen cleanen Look vermitteln.
* Im Editorfeld soll es möglich sein ein Gitternetz anzeigen zu lassen.
  + 1. **Bedienung**
* Das Produkt soll über Maus und Tastatur bedienbar sein.
* Die Bedienung der Oberfläche soll intuitiv geschehen.
* Der Benutzer soll die Anwendung aber auch lediglich mit der Maus steuern können.
* Das Produkt soll es dem Benutzer ermöglichen, die meisten bekannten Tastenkürzel benutzen zu können.
  + 1. **Code**
* Die Anwendung und der Schaltplan sollen sicher sein vor (unbeabsichtigter) Manipulation durch den Benutzer.
* Die Anwendung soll leicht erweiterbar sein, d.h. neue Bauteile können ohne großen Aufwand hinzugefügt werden.
* Der Programmcode soll verständlich geschrieben werden und gut kommentiert sein.
* Die Wartung von Programmcode soll möglichst einfach sein.
  + 1. **Bedienen der Bauteile**
* Die Grafiken von Bauteilen sollen stets gut erkennbar und genormten Vorgaben entsprechen.
* Bauteile sollen per Drag-and-Drop eingefügt werden können.
* Die Auswahl der Bauteile soll intuitiv erfolgen.
* Wählt der Benutzer ein Objekt aus, so wird dieses mittels einer Umrahmung markiert.
* Man soll mehrere Bauteile auf einmal markieren und bearbeiten können.
* Die Leitungen der markierten Bauteile sollen farblich hervorgehoben werden.
* Die Bauteile sollen sich auf Wunsch des Benutzers an einem Raster ausrichten.
* Beim Auswählen eines Connectors soll dieser sich farblich von allen anderen hervorheben. Bei Letzteren soll klar erkennbar sein, ob sie schon mit dem Ausgewählten verbunden sind oder nicht.
  + 1. **Beschriftung**
* Die Anwendung soll in deutscher Sprache entwickelt und designed werden.
* Sämtliche Texte der Anwendung sollen in der gleichen Schriftart verfasst sein.
* Im Falle von Fehlermeldungen sollen diese informativ und verständlich verfasst sein.
* Die Beschriftung von Bauteilen soll stets lesbar und deutlich sein. (z.B. nicht verpixelt)
  + 1. **Bibliothek**
* Das Programm soll einen Bereich haben, in dem die verfügbaren Bauteile mit Beschriftung dargestellt sind.
* Bauteile sollen in Gruppen, sprich Bibliotheken, organisiert sein.
* Der Benutzer soll Bauteile bestehenden Bibliotheken hinzufügen oder neue Bibliotheken importieren können.
  + 1. **Sonstiges**
* In einem Hilfefenster soll der Benutzer Informationen beziehungsweise eine Einführung zu der Bedienung des Programms erhalten.
* In einem About Fenster sollen Informationen über das Entwicklerteam und andere Mitwirkende bereitgestellt werden.

# **Ausblick**

Das Produkt ist darauf ausgelegt durch verschiedenste Funktionalitäten möglichst einfach erweitert werden zu können. Außerdem ist es als Open Source Projekt auf GitHub für jeden zugänglich und modifizierbar.

* 1. **Ideen für Erweiterungen**

Im Laufe des Projekts kamen immer wieder neue Ideen auf, wie das Produkt erweitert werden könnte, aber aus verschiedenen Gründen nicht in die nähere Planung einfließen konnten.

* + 1. **Komponenten**

Die naheliegendste Erweiterung ist das Hinzufügen von weiteren logischen Schaltplankomponenten zu den Bibliotheken wie zum Beispiel Dioden oder Schalter. Dazu müssen SVG Grafiken der jeweiligen Komponenten erstellt und in entsprechenden XML Dokumenten eingebunden werden. Es ist möglich die neuen Komponenten während der Laufzeit zu importieren oder schon zum Programmstart zu laden.

* + 1. **Eigenschaften**

Des Weiteren haben Komponenten oder auch Leitungen in echten elektrischen Schaltplänen bestimmte Eigenschaften. Über ein Eigenschaftenfenster könnten sich beispielsweise die Höhe des Widerstands oder die Stromstärke einer Stromquelle individuell einstellen lassen. Die Werte könnten auch auf dem Schaltplanfenster angezeigt werden und eine realistischere Darstellung ermöglichen. Ein erster Prototyp für das Eigenschaftenfenster ist bereits im Code vorhanden aber findet noch keine Verwendung.

* + 1. **Simulation**

Die Eigenschaften sind Voraussetzung für eine realistische Simulation des Schaltplans, mit der veranschaulicht werden soll, ob dieser funktionsfähig ist oder nicht. Benutzer sollen die Software dazu nutzen, Schaltpläne zu planen und zu testen, damit bei der späteren Umsetzung Komplikationen ausgeschlossen werden können. Durch eine Simulation könnte man zum Beispiel erkennen, ob ein Kurzschluss auftritt. Zur Umsetzung dieser Erweiterung ist allerdings ein tieferes elektrotechnisches Verständnis von Nöten.

* + 1. **Design**

Etwas einfacher ist die Umsetzung von verschiedenen Designs des Programms um die Nutzerfreundlichkeit zu steigern. Benutzer sollen das Produkt an ihre individuellen Bedürfnisse anpassen können, das Design steht dabei an oberster Stelle. Eine dunkle Benutzeroberfläche kann bei längerem Arbeiten angenehmer sein, als eine sehr helle und grelle. Da dies selbstverständlich eine Geschmacksfrage ist soll der Nutzer selbst entscheiden und jederzeit wechseln können. Für die Implementierung neuer Designs müssten nur CSS Dateien eingebunden werden.