
Grafischer Darstellungsrechner

Authors: Priller Patrick, Mairhofer David, Pernthaler Daniel

stpripat@bx.fallmerayer.it, stmaidav@bx.fallmerayer.it, stperdan@bx.fallmerayer.it



Dokumentation

Informatik
Klasse 4AT

Professor: Rainer Ulrich

Brixen, den 03. Mai 2023

Inhaltsverzeichnis

1	Themenbeschreibung	3
1.1	Projektziele	3
1.2	Unsere Vision	3
1.3	Potenziell nutzbare APIs	3
1.3.1	mxParser	3
1.3.2	JScience	4
1.4	Technologien und Tools	4
2	Vorgehensmodell: Kanban	5
2.1	Grundlagen von Kanban	5
2.2	Anwendung von Kanban im Projekt	6
2.3	Vorteile und Erfahrungen	6
3	Anforderungsanalyse	6
3.1	Funktionale Anforderungen	6
3.2	Nicht funktionale Anforderungen	6
3.3	Aktivitätsdiagramm	6
4	Use-Case-Diagramm	7
4.1	Erklärung	7
4.2	Diagramm	7
5	Systemarchitektur	8
5.1	Übersicht	8
5.2	Frontend	8
5.3	Backend	8
5.4	Klassendiagramm	8
6	Umsetzung (Programmierung)	8
6.1	Backend-Entwicklung	8
6.2	Frontend-Entwicklung	8
6.2.1	UI-Prototyp	8
6.3	Integration von APIs	8
6.4	Teststrategie	8
7	Benutzung (User Guide)	8
7.1	Installation und Einrichtung	8
7.2	Bedienung der Anwendung	8
7.3	Beispiele	8
8	Fazit	8
8.1	Zusammenfassung	8
8.2	Erreichte Ziele	8
8.3	Zukünftige Erweiterungen	8

9	Anhang	8
9.1	Arbeitstagebuch (Planung)	8
9.2	Arbeitstagebuch (Umsetzung)	8
9.3	Quellcode	8
9.4	Screenshots	8
9.5	Lizenzinformationen	8

1 Themenbeschreibung

1.1 Projektziele

1.2 Unsere Vision

Unsere Vision für das Plotter-Projekt war es, eine benutzerfreundliche und leistungsfähige Java-basierte grafische Funktionen-Plotting-Anwendung zu entwickeln, die es Benutzern ermöglicht, mathematische Funktionen und ihre Ableitungen zu visualisieren. Wir wollten eine breite Palette von Funktionen unterstützen, einschließlich linearer, quadratischer, exponentieller und trigonometrischer Funktionen. Dabei sollte die Anwendung die mxParser-Bibliothek verwenden, um mathematische Ausdrücke zu analysieren und ihre Werte zu berechnen.

Zu Beginn des Projekts haben wir uns darauf konzentriert, die Kernfunktionen der Anwendung zu entwickeln, wie das Plotten von Funktionen in einer benutzerfreundlichen Oberfläche, das Berechnen und Anzeigen von Ableitungen von Funktionen, das Anwenden von Konstanten und Variablen auf Funktionen, das Ein- und Ausblenden von Funktionen im Graphen und das Anpassen des Erscheinungsbilds des Plotbereichs. Zudem haben wir die Möglichkeit geschaffen, die Ansicht des Plotbereichs zu vergrößern und zu verkleinern.

Wir wollten sicherstellen, dass die Anwendung einfach zu installieren und auszuführen ist, weshalb wir uns für den Einsatz von Maven zur Erstellung des Projekts entschieden haben. Um das Projekt schnell und einfach zu nutzen, haben wir detaillierte Anweisungen zur Installation der erforderlichen Abhängigkeiten, zum Erstellen des Projekts mit Maven und zum Ausführen der Anwendung bereitgestellt.

Insgesamt war unsere Vision, ein leistungsstarkes und benutzerfreundliches Plotter-Tool zu entwickeln, das sowohl für den Bildungsbereich als auch für professionelle Anwender wertvoll ist und die Visualisierung und Analyse von mathematischen Funktionen vereinfacht.

1.3 Potenziell nutzbare APIs

In unserem Plotter-Projekt haben wir verschiedene APIs in Betracht gezogen, um die Funktionalität und Benutzerfreundlichkeit der Anwendung zu erweitern. Zwei der vielversprechendsten APIs, die wir untersucht haben, sind mxParser und JScience.

1.3.1 mxParser

mxParser ist eine vielseitige und leistungsfähige Bibliothek zur Analyse und Berechnung mathematischer Ausdrücke. Die Bibliothek ist in Java geschrieben und unterstützt eine breite Palette von Funktionen, einschließlich algebraischer, trigonometrischer, exponentieller und logarithmischer Funktionen. Wir haben mxParser in unserem Plotter-Projekt verwendet, um mathematische Ausdrücke zu analysieren und ihre Werte zu berechnen. Die offizielle Website von mxParser bietet umfassende Dokumentation und Beispiele zur Verwendung der Bibliothek: <http://mathparser.org/>

1.3.2 JScience

JScience ist eine umfangreiche Java-Bibliothek, die verschiedene wissenschaftliche Disziplinen abdeckt, darunter Mathematik, Physik, Chemie und Informatik. Die Bibliothek bietet eine Reihe von Funktionen, die für unser Plotter-Projekt nützlich sein könnten, wie z.B. Unterstützung für komplexe Zahlen, Einheitenkonversion und lineare Algebra. JScience könnte verwendet werden, um die Funktionalität unserer Anwendung zu erweitern und die Unterstützung für zusätzliche mathematische Funktionen und Operationen zu verbessern. Weitere Informationen zu JScience finden Sie auf der offiziellen Website: <http://jscience.org/>

Durch die Integration dieser APIs in unser Plotter-Projekt können wir die Benutzererfahrung und die Funktionalität unserer Anwendung weiter verbessern und eine breitere Palette von mathematischen Funktionen und Operationen unterstützen.

1.4 Technologien und Tools

2 Vorgehensmodell: Kanban

2.1 Grundlagen von Kanban

Kanban ist ein visuelles Projektmanagement-System, das ursprünglich in der Automobilindustrie entwickelt wurde und auf der Kanban-Methode basiert. Für unser Plotter-Projekt wurde Kanban gewählt, da es eine hohe Flexibilität bietet und es uns ermöglicht, einen guten Überblick über den Fortschritt des Projekts zu behalten.

Kanban besteht aus einer Reihe von Spalten, die verschiedene Phasen des Projekts repräsentieren, z. B. "Zu erledigen", "In Arbeit" und "Fertig". Innerhalb dieser Spalten werden Aufgaben als Karten dargestellt, die Informationen zu Priorität, Verantwortlichen und Fälligkeitsdatum enthalten. Karten werden von einer Spalte zur nächsten verschoben, wenn sie von einer Phase zur nächsten fortschreiten.

Dieses System erleichtert die Verfolgung des Fortschritts und hilft dabei, Engpässe oder Verzögerungen schnell zu identifizieren. Es fördert außerdem eine kontinuierliche Verbesserung, indem es Teams dazu anregt, regelmäßig den Workflow zu überprüfen und anzupassen.

Die folgende Abbildung zeigt ein Beispiel für ein Kanban-Board:

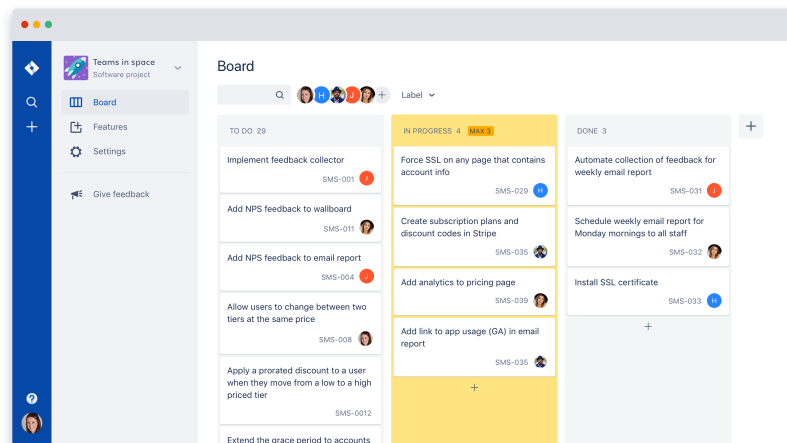


Abbildung 1: Beispiel eines Kanban-Boards

Insgesamt bietet Kanban eine einfache und effektive Möglichkeit, den Fortschritt unseres Plotter-Projekts zu verwalten und sicherzustellen, dass wir kontinuierlich Verbesserungen vornehmen und auf unsere Ziele hinarbeiten.

2.2 Anwendung von Kanban im Projekt

2.3 Vorteile und Erfahrungen

3 Anforderungsanalyse

3.1 Funktionale Anforderungen

3.2 Nicht funktionale Anforderungen

3.3 Aktivitätsdiagramm

4 Use-Case-Diagramm

4.1 Erklärung

Das Use-Case-Diagramm beschreibt die Funktionalität des Plotters und zeigt, wie Benutzer mit der Anwendung interagieren können. Der Benutzer kann in ein Textfeld seine gewünschte Funktion eingeben und mithilfe von Buttons Sonderzeichen in das Textfeld schreiben. Der Benutzer hat auch die Möglichkeit, verschiedene Buttons zu verwenden, um die Funktion zu bearbeiten, wie zum Beispiel die Sichtbarkeit, Ableitung, Kurvendiskussion oder die Funktion zu löschen. Zusätzlich kann der Benutzer den Anzeigebereich der Funktion mithilfe von Zoomen oder Verschieben verändern.

4.2 Diagramm

Das folgende Diagramm zeigt das Use-Case-Diagramm für unseren Plotter:

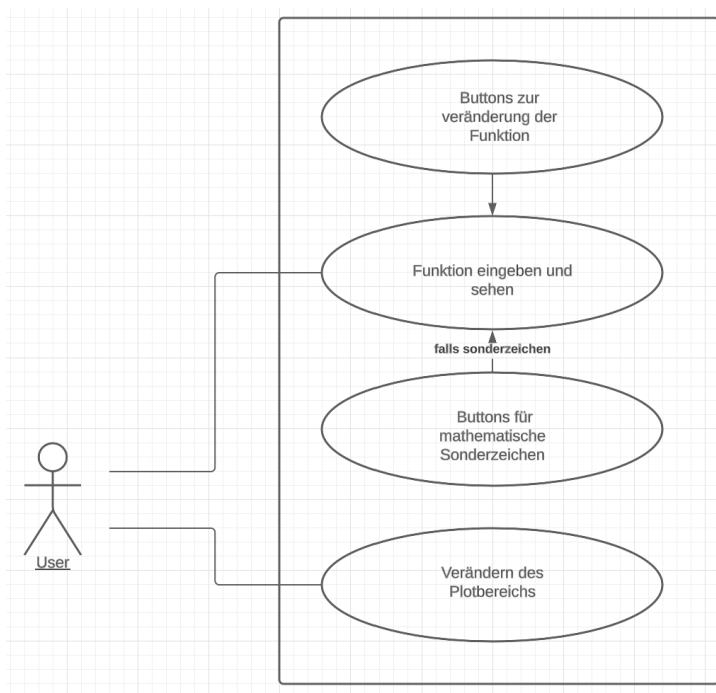


Abbildung 2: Use-Case-Diagramm des Plotters

Das Diagramm veranschaulicht die verschiedenen Aktionen, die ein Benutzer ausführen kann, um mit der Anwendung zu interagieren und mathematische Funktionen zu visualisieren und zu analysieren.

5 Systemarchitektur

5.1 Übersicht

5.2 Frontend

5.3 Backend

5.4 Klassendiagramm

6 Umsetzung (Programmierung)

6.1 Backend-Entwicklung

6.2 Frontend-Entwicklung

6.2.1 UI-Prototyp

6.3 Integration von APIs

6.4 Teststrategie

7 Benutzung (User Guide)

7.1 Installation und Einrichtung

7.2 Bedienung der Anwendung

7.3 Beispiele

8 Fazit

8.1 Zusammenfassung

8.2 Erreichte Ziele

8.3 Zukünftige Erweiterungen

9 Anhang

9.1 Arbeitstagebuch (Planung)

9.2 Arbeitstagebuch (Umsetzung)

9.3 Quellcode

9.4 Screenshots

9.5 Lizenzinformationen