Werbebudgetoptimierung – Erweiterung

Bericht der Erweiterung des Werbebudgetoptimierung

Fach: Angewandte Betrieblich Systemforschung

Professor: Prof. Dr. Michael Grütz

Autor: Stefan Brecht

Mat-Nr.: 277590

E-Mail: [Stefan.Brecht@htw-konstanz.de](mailto:Stefan.Brecht@htw-konstanz.de)

Semester: 8. Semester Wirtschaftsinformatik, Fachrichtung Software-Engineering

Datum:

Inhaltsverzeichnis

[1 Stand vor der Überarbeitung 3](#_Toc199514501)

[1.1 Features Realisiert 3](#_Toc199514502)

[1.2 Features noch nicht Realisiert 3](#_Toc199514503)

[2 Stand nach der Überarbeitung 4](#_Toc199514504)

[2.1 Features Realisiert 4](#_Toc199514505)

[2.2 Features noch nicht Realisiert 4](#_Toc199514506)

[3 Zeitaufwand für die Realisierung 5](#_Toc199514507)

[3.1 Zeitaufwand 5](#_Toc199514508)

# Stand vor der Überarbeitung

In diesem Kapitel wird der Stand vor der Überarbeitung aufgelistet.

## Features Realisiert

Vor der Überarbeitung waren folgende Features Realisiert:

* Daten können eingegeben werden
  + Über das GUI können alle nötigen Daten eingegeben werden
* LP-Ansatz wird erzeugt
  + Durch bestätigen der Eingaben wird der LP-Ansatz erzeugt
  + Der LP-Ansatz wird an den Solver geschickt
  + Das Ergebnis des Solver wird eingelesen
  + Solver Ergebnis wird interpretiert
* Ergebnis wird ausgegeben
  + Das Interpretierte Ergebnis wird über das GUI ausgegebenen

Diese Features haben ohne größere Fehler funktioniert, wobei der LP-Ansatz von mir nicht überprüft wurde.

## Features noch nicht Realisiert

* Neu Funktion nicht vorhanden (Zurücksetzen aller Felder)
* Speichern Funktion nicht vorhanden
* Öffnen Funktion nicht vorhanden

Diese Features waren nicht Implementiert.

# 

# Stand nach der Überarbeitung

In diesem Kapitel wird der Stand nach der Überarbeitung aufgelistet und die neuen Features kurz erklärt.

## Features Realisiert

* Speichern Funktion hinzugefügt
  + Das Speichern aller eingegebenen Daten ist möglich und funktioniert
  + Es wird die Dateiendung \*.wbo genutzt
  + Es wurde ein Filter implementiert damit nur Dateien des mit der Endung \*.wbo angezeigt werden
* Öffnen Funktion hinzugefügt
  + Die gespeicherten Daten können geöffnet werden
  + Hier wurde der selbe Filter wie für die Speichern-Funktion genutzt
* Neu Funktion zum Zurücksetzten der Daten
  + Über die Neu Funktion können alle Daten zurückgesetzt werden

Die Implementierten Features wurden von mir implementiert und getestet und sind funktionsfähig so wie hier beschrieben. Es wurde falsche Nutzereingaben abzufangen.

## Features noch nicht Realisiert

Es sind alle Features realisiert welche zur Verbesserung der Handhabung dienen.

# Zeitaufwand für die Realisierung

Während der Realisierung der habe ich meine Arbeiten dokumentiert und zeitlich erfasst. Diese Dokumentation wird in diesem Kapitel genauer erläutert.

Es wurden in der Vorlesung folgende Features festgelegt welche von mit Implementiert werden sollten:

* Speichern/Öffnen Funktion
* Neu Funktion zum Zurücksetzten aller Felder

Es war mir möglich in der vorgegebenen Zeit alle Features zu realisieren.

## Zeitaufwand

|  |  |  |
| --- | --- | --- |
| **Stunden** | **Typ** | **Beschreibung** |
| 1,5 | Selbststudium | Einarbeiten in Programm-Code |
| 2 | Selbststudium | GUI Änderungen durchgeführt:   |  |  | | --- | --- | | **Stunden** | **Beschreibung** | | 0.25 | „Neu“-Button zu „Leeren“ umbenannt | | 0,25 | Radio-Button „Männlich“ voreingestellt. | | 1,0 | Reiter wechseln ohne Datenverlust | | 0,5 | Entfernen aller unnötigen Button für eine Bessere Navigation | |
| 2 | Selbststudium | **Problem:**  Einfaches speichern der Daten ohne viel Aufwand  **Vorgehen:**   |  |  | | --- | --- | | **Stunden** | **Beschreibung** | | 0,5 | Recherche über Speichern von Daten und Zugriff auf das Dateisystem in Java. | | 1 | Testen der Verschiedenen Möglichkeiten:   * Schreiben einer TXT-Datei (0,5) * Schreiben einer XML-Datei (0,25) * Objectserialisierung (0,25) | | 0,5 | Objectserialisierung über die SolverDaten Klassen eingebaut |   **Lösung:**  Objektserialisierung und eine Klasse die alle Daten enthält. |
| 1 | Selbststudium | Implementieren eines Speichern-Fensters:   |  |  | | --- | --- | | **Stunden** | **Beschreibung** | | 0.25 | Recherche über Speichern-Fenster über Google | | 0,25 | FileChooser für das Speichern eingebaut und MenuPunkt Speichern hinzugefügt. | | 0,25 | Recherche über Filter Funktion im FileChooser | | 0,25 | Implementieren des Filter mit der Dateiendung \*.wbo | |
| 2,5 | Selbststudium | **Problem:**  Die späteren Eingabefelder hängen von den anfänglich eingegeben Daten ab.  **Vorgehen:**   |  |  | | --- | --- | | **Stunden** | **Beschreibung** | | 0,5 | Verstehen des Codes wo und wie die einzelnen Felder und Tabellen erzeugt werden | | 0,5 | Versuchen diese Felder über diese Funktionen zu erzeugen (War nicht möglich da ich nicht alle Abhängigkeiten auflösen konnte) | | 1,5 | Simulieren der Gesamten Nutzereingaben mit befüllen der Felder und klicken der Button per DoClick-Funktion. |   **Lösung:**  Simulieren der Nutzer Eingaben beim öffnen der gespeicherten Datei. |
| 0,5 | Selbststudium | Testen der implementierten Funktionen |
| 0,5 | Selbststudium | Erstellen von Beispiel-Daten (3 Beispiele) |
| 2 | Selbststudium | Schreiben der Dokumentationen für die Änderungen   |  |  | | --- | --- | | **Stunden** | **Beschreibung** | | 0,5 | Struktur und Template Dokument erstellt | | 1 | Schreiben der Dokumentation | | 0,5 | Überarbeiten der Dokumentation nach Besprechung in der Vorlesung | |
| 12,0 |  | ***Summe Gesamt*** |