Projektlaborbericht

Mit dem Thema:

Programm für die Zeiterfassung von Kleinunternehmen

eingereicht durch: Name: David Kempf, Sean Woods, Lauritz Abel

Matrikelnummer: 207815, ---------, 207804

Studiengang: AE

Hochschule Heilbronn – Campus Künzelsau

betreut von: Prof. Dr.-Ing. Marcus Stolz

Hochschule Heilbronn – Campus Künzelsau

Künzelsau, xx.xx.xxxx

Ehrenwörtliche Erklärung

Hiermit erkläre wir, David Kempf, Sean Woods und Lauritz Abel dass wir die vorliegende Arbeit selbstständig angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Es wurden keine anderen als die angegebenen Quellen und Hinweise verwendet.

Die vorliegende Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Künzelsau xx.xx.xxxx David Kempf, Sean Woods, Lauritz Abel

Inhaltsverzeichnis

[1. Ist-Zustand: 1](#_Toc139352792)

[2. Kurzfassung 1](#_Toc139352793)

[3. Aktuell Verfügbare Systeme 2](#_Toc139352794)

[TOGGL 2](#_Toc139352795)

[AtWork 2](#_Toc139352796)

[Factro 2](#_Toc139352797)

[4. Entwicklung 3](#_Toc139352798)

[GitHub 3](#_Toc139352799)

[JetBrain Rider 3](#_Toc139352800)

[Fazit und Ausblick 4](#_Toc139352801)

[1. Abbildungsverzeichnis 5](#_Toc139352802)

# Ist-Zustand:

Für kleine Handwerks- und Landschaftspflegebetriebe besteht aktuell das Problem, es gibt keine digitale Lösung für die Zeitdokumentation. Momentan dokumentieren viele kleine Betriebe ihre Zeiten in kleinen Notizbüchern. Maschinenstunden werden wegen des Aufwandes oft nicht richtig dokumentiert und die Zeiten geschätzt. Es gibt die Möglichkeit mittels Produkte wie zum Beispiel von Stihl (Stihl Connector) Maschinenzeiten aufzunehmen und zu dokumentieren. Jedoch müssen diese wieder händisch in ein Rechnungstool übertragen werden. Ein weiteres Problem ist das oft von Handwerker die Büroarbeit als lästig und unnötig betrachtet wird und somit wieder zu kurz kommt. Jedoch verdient ein Betrieb sein Geld nicht mit der getanen Arbeit, sondern mit dem Ausstellen von korrekten Rechnungen.

# Kurzfassung

Um den aktuellen Zustand zu verbessern ist das Ziel dieser Arbeit, mit Hilfe einer App die Zeiterfassung und Rechnungsstellung zu automatisieren. Das Programm soll in zwei Bereiche aufgeteilt werden. Eine Desktopanwendung, welche zur Aufgabenplanung und Controlling genutzt werden soll und eine mobile App welche als Anwenderapp gedacht ist, für die Zeiterfassung. Das Programm soll es dem Anwender ermöglichen leicht seine Arbeitszeiten und Maschinenzeiten zu dokumentieren und zu Verwalten. Nachdem ein Auftrag abgearbeitet ist, soll mittels eines klickes möglich sein, eine Rechnung erstellen zu lassen. Somit wird die Zeit im Büro minimiert und die effektive Arbeitszeit erhöht.

# Aktuell Verfügbare Systeme

## TOGGL

Toggl ist ein großes Zeitverwaltungssystem welches von Unternehmen wie Amazon, SAP, Linkedin usw. genutzt wird. Das Programm besticht mit einer einfachen Bedienung, somit kann auch ohne eine Schulung gearbeitet werden. Die Software ist sehr offen aufgebaut und lässt sich mit andere Softwaren wie Jira, Salesforce usw. verbinden.

## AtWork

AtWork ist eine kostenpflichtiges Zeiterfassungssystem. In diesem lässt sich das ganze Mitarbeiterteam verwalten. Das Programm sticht vor allem mit der mit der Funktion der Überstundenberechnung und Auszahlung und der Funktion mit der Schichtplanug heraus

## Factro

Fracto ist eine Cloud basierte App. Diese speichert die Arbeitsstunden Projekt bezogen ab. In der App ist es möglich ein ganzes Team zu verwalten. Mithilfe der Controlling Funktion ist es möglich die Auslastung des Teams auszuwerten und zu bewerten. Eine kleine Basis-Version ist kostenlos danach müssen kostenpflichtige Lizenzen gekauft werden.

Die bestehenden Anwendungen sind oft sehr teuer und somit nicht für kleine Betriebe rentabel oder geeignet. Die kostenlosen Apps sind oft unflexibel und eigenen sich nicht für den kommerziellen Gebrauch. Momentan fehlt auf dem Markt eine einfache und flexible Anwendung, welche sich auf die relevanten Funktionen begrenzt.

# Entwicklung

## GitHub

Was ist GitHub? GitHub ist eine Cloudbasiert Software, welche die Datenverwaltung und den Projektablauf vereinfacht. Somit ist es möglich das mehrere Entwickler gleichzeitig an einem Sourcecode schrieben könne oder Daten gemeinsam geteilt werden. Die neu entwickelten Inhalte können mit allen geteilt werden und werden versioniert abgespeichert. Somit ist für alle ersichtlich was geändert oder hinzugefügt wurde. Nicht nur Sourcecode ist möglich, sondern auch Word, Excel, Powerpoint Dateien und vieles mehr.

## JetBrain Rider

JetBrain Rider oder kurz nur Rider genannt, wurde als Entwicklungsumgebung gewählt da dieser einige Vorteile mit sich bringt. Rider ähnelt sehr stark der Entwicklungsumgebung Visual Studios von Microsoft. Hier liegt auch schon ein großer Unterschied, Rider ist nicht von Microsoft. Rider ist eine komplett offene Entwicklungsumgebung und nicht an ein Betriebssystem gebunden. Außerdem ist die Entwicklungsumgebung sehr Benutzerfreundlich da diese eine XAML Vorschau bietet welche in Echtzeit die Xamarin XAML Seite zeigt. Bei einem Fehler im Sourcecode kann mithilfe der Codeanalyse der Fehler schnell ausgemacht werden und behoben werden. Dies wird durch den Rider unterstützt da dieser sofort einen Lösungsvorschlag mitliefert. Tritt beim Ausführen ein Fehler auf unterstützt der Debugger den Entwickler. Dieser zeigt den Haltepunkt auf und den Grund, warum das Programm nicht ausgeführt werden konnte.

## .NET Framework

Das .NET Framework wurde Microsoft entwickelt und bedeutet übersetzt soviel wie „Rahmenarbeit“. Hierbei dient .Net Framework als Basisprogramm für die Programmierung von Desktopanwendungen und Mobilen Anwendungen. Es besteht aus einer Laufzeitumgebung, Klassenbibliothek und einem Compiler. Der geschriebene Sourcecode, welcher in der .Net Framework programmiert worden ist, wird in der Laufzeitumgebung ausgeführt. Die Klassenbibliothek dient als nützliche Quelle für jeglichen Anwendungen wie zum Beispiel für SQ Lite Datenbank, PDF Generator, Socket Server und so weiter. Um .Net Framework nutzen zu können wird ein Texteditor und ein Compiler benötigt. Hier bietet Microsoft ihr Visual Studios an oder wie in diesem Fall den Verwendeten JetBrain Rider.

## WPF .NET

Windows Presentation Foundation (WPF) ist eine Klassebibliothek von .NET, welche zu der Entwicklung von graphischen Oberflächen genutzt wird. WPF unterstützt verschiedenste Arten von GUIs: Typische Desktopanwendungen, 3D Grafiken, Dokumente, Browser-basierte Anwendungen und Videos. WPF ist von Microsoft als Nachfolger für das aus .NET bekannte Windows Forms eingeführt worden. Eine WPF-Oberfläche kann als Programmcode programmiert werden oder wie von Rider unterstütz durch XAML Language programmiert werden. Die Anwendung ist vektorbasiert aufgebaut. Daher ist die Auflösung unabhängig von der Bildschirmgröße. Diesen Vorteil wird richtig deutlich beim Verwenden eines responsiven Designs. Bei einem responsiven Design passt sich die grafische Oberfläche der Gesamtgröße an. Zieht man die Anwendung nun größer leidet die Auflösung nicht darunter.

## C#

C# wird in diesem Projekt als Programmiersprache verwendet. Diese ist eine objektorientierte Sprache. Ihren Ursprung hat diese auch bei Microsoft Anfang der 2000er Jahre. Die Sprache wurde Plattform unabhängig entwickelt im Rahmen der .NET-Strategie von Microsoft. Diese sollte das Erstellen von Windowsanwendungen mit der .NET-Framework vereinfachen. Hierbei greift die Sprache Konzepte von [Java](https://de.wikipedia.org/wiki/Java_(Programmiersprache)), [C++](https://de.wikipedia.org/wiki/C%2B%2B), [Haskell](https://de.wikipedia.org/wiki/Haskell_(Programmiersprache)), [C](https://de.wikipedia.org/wiki/C_(Programmiersprache)) sowie von [Delphi](https://de.wikipedia.org/wiki/Embarcadero_Delphi) auf.  
Zeiger welche in C++ als eher unsicher bekannt sind werden in C# nur für den sogenannten „unsicheren Code“ zugelassen zum Beispiel Programme welche von einer Website ausgeführt werden ohne erweiterte Rechte. Da C# als .NET-Sprache gilt, gibt es auch die Möglichkeit über eine Sprachunterstützung für Attribute und Delegaten. In Metadaten werden in C# Informationen über eine Klasse, ein Objekt oder eine Methode gespeichert. Diese werden zur Laufzeit ausgewertet.

* Planung
* Entwicklung
  + Technologiewahl
    - gitHub
    - Rider (xaml Vorschau Datenbank Vorschau mächtiges Programm integrierte Versionsverwaltung) unabhängig
    - .net Framework
    - Wpf visual (Desktopapp responsive design
    - C#
      * [Was ist C#? (dev-insider.de)](https://www.dev-insider.de/was-ist-c-a-846162/) Quelle
    - App IOS und Android nur mit einem Tool entwickeln
  + Software design
    - Wpf mvvm <https://www.youtube.com/watch?v=JWCudlf5c4Q>
    - Aufbau Datenbank (db\_Klassen rw für eine Tabelle zuständig)
    - Error-handling (Ausgabe Messagebox)
    - Socketserver
    - PDF
    - Benutzerverwaltung

# Fazit und Ausblick

# Abbildungsverzeichnis

1. Tabellenverzeichnis

1. Literaturverzeichnis

[Windows Presentation Foundation (WPF) - Begriffserklärung im Entwickler-Lexikon/Glossar auf www.IT-Visions.de](https://www.it-visions.de/glossar/alle/3718/Windows_Presentation_Foundation.aspx)