



# Patienten-Verwaltung

## Projekt Dokumentation

Praxis „Lustigmann“

Jhoni Dewa und Vincent Knapp

## Inhalt

Einleitung.....	2
Arbeitsauftrag.....	2
Voraussetzung.....	2
Projektdurchführung.....	3
Funktionsanalyse.....	3
Klassen.....	5
UML-Diagramm.....	5
Projektmanagement.....	6
Projektablauf.....	6
Aufteilung der Teilprojekte.....	6
Ablaufdiagramm.....	7
Projektergebnisse.....	8
Probleme die entstanden sind.....	10
Technische Seite.....	10
Organisatorische Seite.....	10
Team.....	10
Ausblick Erweiterungsmöglichkeiten.....	11
Fazit, Reflexion.....	11

## Einleitung

### Arbeitsauftrag

Für die Praxis Lustigmann galt es eine Softwarelösung für die Verwaltung und Registrierung von Patienten und Neupatienten zu entwickeln. Diese Entwicklung hat einen Zeitraum von vier Wochen mit je fünf Schulstunden beansprucht.

Fokus lag bei der Programmierung auf der Realisierung des Projekts mit sauber getrennten Klassen.

Ziel war das Anlegen und Bearbeiten von Patienten-Informationen im Verwaltungs-System. Die Praxis Lustigmann hat seither ihre Patienten mit Excel-Tabellen geführt und erhoffen sich durch die Entwicklung der Software eine effizientere und zentralisierte Datengrundlage für alle Kollegen in der Praxis (als auch extern arbeitende Kollegen). Besonderen Wert legt die Praxis Lustigmann auf ein einfaches Handling mit der Software, da dadurch intensive Schulungen im Zuge des Projekts entfallen und die Nutzer direkt mit der Software arbeiten können.

### Voraussetzung

Für die Entwicklung, Implementierung und einen dauerhaften Betrieb der Software im Praxis-Umfeld sind folgende Punkte als Voraussetzung gegeben:

1. Ein bereits eingesetzter Server mit einer SQLite Datenbank für das Abspeichern aller Informationen.
2. Die Rechner, auf denen die Software ausgeführt werden soll, müssen eine Verbindung zum Server aufbauen können. Ob das durch die lokale Netzkonfiguration oder durch das Arbeiten mit einer VPN erfolgt ist in diesem Fall egal. Die Praxis Lustigmann hat hierfür schon Vorkehrungen für frühere Situationen ausgerollt.
3. Die Software wurde in der Programmiersprache „C#“ entwickelt. Somit ist die Software primär für Windows-Betriebssysteme ausgelegt. Eine Ausführung der Software in Linux, auf einem Mac o.ä. ist aber über Umwege ebenfalls möglich. Die Kompatibilitäts-Diskrepanzen wird die Praxis Lustigmann auf eigenem Wege klären.

# Projektdurchführung

## Funktionsanalyse

Die Software ist in drei Seiten aufgeteilt.

### Login-Screen:

Die erste Ansicht, die zu sehen ist, sobald die Software startet ist der Login-Bereich. Dieser wird passiert durch die Eingabe des Passwortes, das an alle Kollegen ausgehändigt wird. Das Passwort wird von der Software verifiziert und - sollte es richtig sein – gelangt der Nutzer automatisch auf Seite Zwei. Sollte das Passwort falsch sein, so wird eine Fehlermeldung ausgegeben.

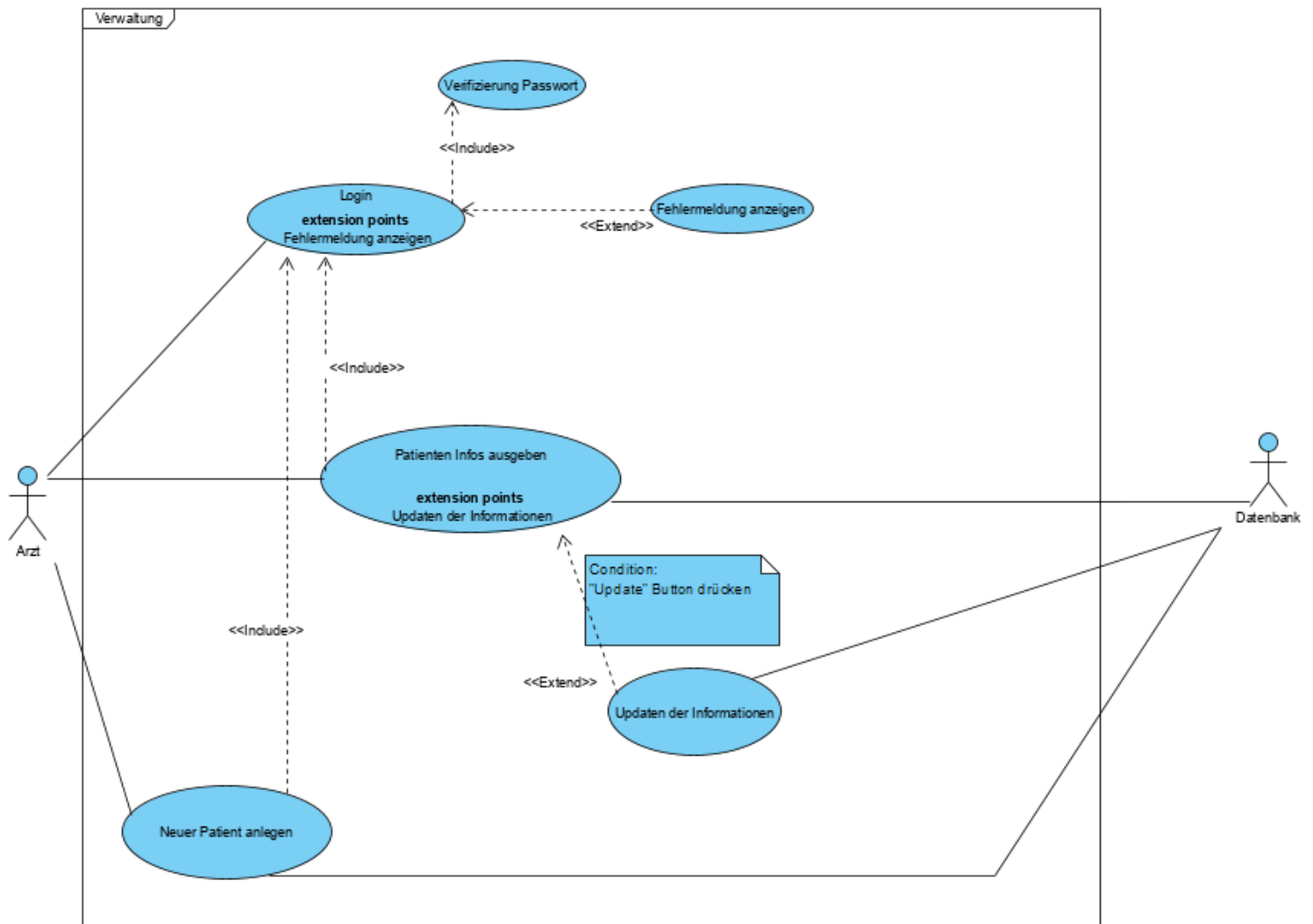
### Patienten-Ansicht:

Die zweite Ansicht zeigt die einzelnen Patienten und deren Informationen an. Dort gibt es ebenso die Möglichkeit die Informationen des ausgewählten Patienten zu bearbeiten und die Möglichkeit über die Suche Patienten in der Verwaltung zu finden. Die Informationen auf dieser Seite sind schreibgeschützt, solange der „Update“ Button nicht gedrückt werden. Wenn der „Update“ Button gedrückt werden, so kann der Nutzer zwar weiterhin die Buttons zum Wechseln der einzelnen Patienten sehen und betätigen, wechselt dadurch jedoch nicht automatisch zum nächsten Patienten, sondern springt aus dem „Bearbeiten“-Modus für den ausgewählten Patienten.

### Neuer-Patient Anlegen:

Auf Seite Drei ist der Workflow zur Erstellung neuer Patienten im System abgebildet. Vom Nutzer werden folgende Informationen abgefragt: Vorname, Nachname, Geburtsdatum, Telefonnummer, Email-Adresse, momentane Krankheit, Geschlecht. Die Eingabe wird auf Validität geprüft (z.B. Email enthält @-Zeichen und mindestens einen Punkt) und danach als neuer Datensatz in der SQLite Datenbank abgespeichert. Um die Datenbank jedoch weiterführen zu können, ergänzt das System die Eingabe des Nutzers um einen Index, der von Patient zu Patient jeweils um eins erhöht wird.

Zur Veranschaulichung haben wir ein **USE-CASE** Diagramm erstellt:



## Klassen

Die Software wurde in drei Klassen aufgeteilt:

### -Login

Diese Klasse beinhaltet den Login-Screen und sämtliche Prüfungs-Funktionen bei der Anmeldung des Nutzers.

### -TabForm

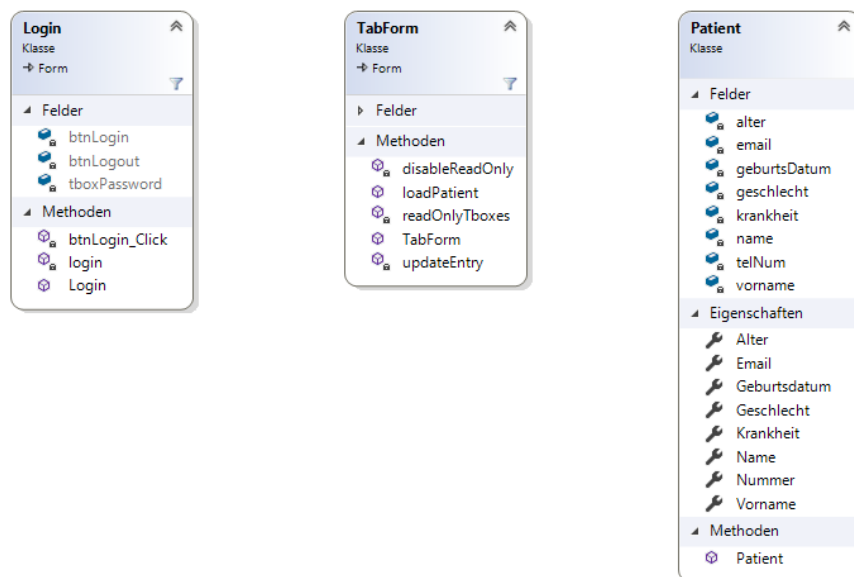
In der TabForm Klasse ist die grafische Oberfläche der Software abgebildet. Hinzu kommen die Funktionen, die auf diesen Seiten genutzt werden (z.B. Update-Button, Save-Patienten, Read-Write-Rights).

Über diese Form werden auch Objekte der dritte Klasse Patient erstellt.

### -Patient

In dieser Klasse stehen alle Werte des ausgewählten Patienten, so wie alle alle GET- und SET-Methoden, die der Klasse zugewiesen wurden. Im Gegensatz zu den ersten beiden Klassen beinhaltet die Patienten-Klasse keine grafische Oberfläche, da die Klasse als Datengrundlage für die anderen Klassen dient.

## UML-Diagramm



# Projektmanagement

## Projektablauf

Zu Beginn des Projekts haben wir uns Gedanken zur Umsetzung gemacht und diese auf einer Mind-Map visualisiert, bevor wir die Strukturellen Gegebenheiten klärten.

Im Anschluss richteten wir eine Git-Hub-Projekt ein, so dass wir beide an dem Projekt arbeiten konnten.

Johni erstellte einen Entwurf der Software in TP2, den er bereits in der nächsten Unterrichtsstunde vorzeigen konnte und auf dessen Basis wir dann noch Anpassungen und Weiterentwicklung vornahmen.

Über den kompletten Verlauf des Projekts waren wir regelmäßig mit Testing und Analyse beschäftigt, wofür wir jedoch nochmal zwei Tage nutzten um diese Analyse zu vertiefen und Probleme aufzuschreiben.

In TP4 wurden dann diese Fehler ausgebessert.

Die Dokumentation erfolgte bereits während der Projekterstellung durch die Notation von wichtigen Informationen, Probleme, Debuggings in Git-Hub. Diese Informationen wurden dann genutzt um sie in diesem Dokument zu beschreiben.

TP6 und TP7 sind noch ausstehend: Die Implementierung der Software erfolgt beim Kunden nach der Abnahme der Entwicklung durch den Kunden, die eine Präsentation beinhaltet. Im Anschluss der Implementierung erfolgt die Schulung der Angestellten.

Der Finale Abschluss soll spätestens am 08.03.2020 stattfinden.

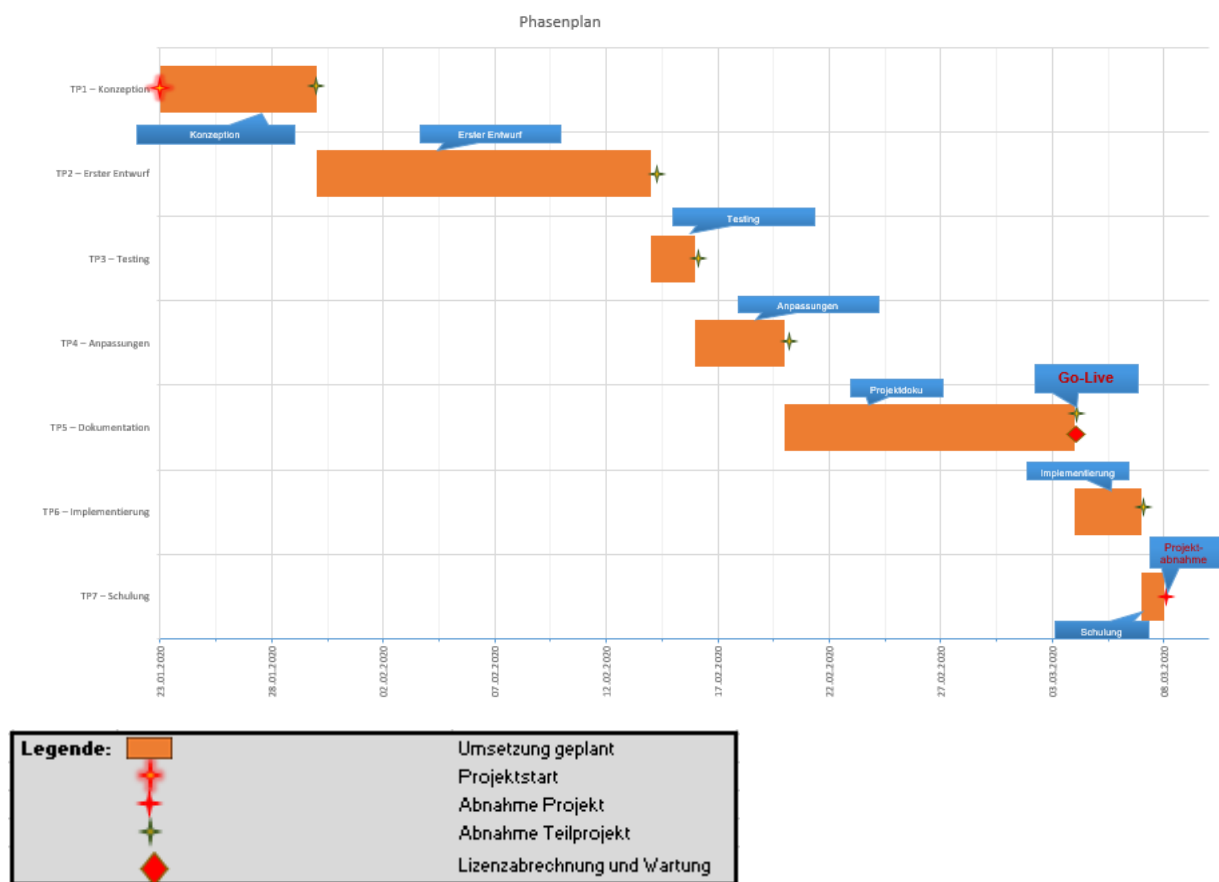
Dieser Termin ist wegen krankheitsbedingten Terminabsagen der Kunden nicht einzuhalten. Der Kunde hat den Abschluss des Projekts und die Schulung auf vorerst unbestimmt verschoben, da zum aktuellen Zeitpunkt nicht klar ist, wann diese TP's erfolgen können.

### Aufteilung der Teilprojekte:

Task	Start	Dauer zeitlich	So Ende
TP7 - Schulung	07.03.2020	1	08.03.2020
TP6 - Implementierung	04.03.2020	3	07.03.2020
TP5 - Dokumentation	20.02.2020	13	04.03.2020
TP4 - Anpassungen	16.02.2020	4	20.02.2020
TP3 - Testing	14.02.2020	2	16.02.2020
TP2 - Erster Entwurf	30.01.2020	15	14.02.2020
TP1 - Konzeption	23.01.2020	7	30.01.2020

Sämtliche Meilensteine sind der jeweilige Abschluss des Teilprojekts.

## Ablaufdiagramm:





## Projektergebnisse

Die Entwicklung der Software war ein gemischter Prozess aus Erfolg, Problemen und Weiterentwicklung der eigenen Fähigkeiten und Wissen. Am Ende sind wir als Entwickler doch sehr zufrieden mit der erbrachten Leistung und gleiche Feedback erhoffen wir uns bei der, noch anstehenden Präsentation vor dem Kunden.



The image shows a user interface mockup for a patient management system. It features a central graphic of a doctor in a blue coat with a red cross, set against a light gray background. Above the graphic is a red button labeled "LOGOUT". Below the graphic is a large orange banner with the text "Patientenverwaltung" in green. Underneath the banner is a password input field with the label "Password" and a blue border. At the bottom is a blue button labeled "Login".

Optisch haben wir uns selbst übertroffen und dadurch das Handling, die Übersichtlichkeit und die Bereitschaft mit der Software zu arbeiten enorm optimiert.

Durch die Reiter-Orientierte Lösung bleibt die Applikation stets übersichtlich und wirkt nicht überladen. (s. oben links)

The screenshot shows a web application for patient management. At the top, there are two tabs: 'Patient' and 'Neu Patient'. To the right of these tabs is a 'LOGOUT' button. Below the tabs, on the left, is a form for patient details. On the right, there is a search bar labeled 'Patientensuche'. Below the search bar is a large, stylized tree graphic with various medical icons as leaves. At the bottom left, there is an illustration of a group of healthcare professionals.

Vorname	Test
Nachname	Tester
Geburtsdatum	26.08.1984
Geschlecht	Männlich
Tel. Nummer	9141212
Email	test@tester.com
Angemeldet am	25.03.2020 09:48:20
Krankheit	too much testing..?

**UPDATE**

Einheitliches nutzen der CI erleichtert das Erkennen der Funktionen auf der Übersicht (s. Update)

Ebenso erleichtert die Such-Funktion die tägliche Nutzung der Software sehr. Nutzer müssen nicht mehr, wie es bei Lustigmann davor der Fall war, Excel-Tabellen hin und her schieben, sondern haben hier die Möglichkeit ein einheitliches System zu füllen, das einem auch das Scrollen durch lange Listen erspart.

Der Logout-Button bietet die Funktion die Software NICHT zu schließen, sondern nur durch zum Login-Screen zu gelangen. Nutzer, die sich „eben nur kurz einen Kaffee holen“ gehen müssen die Software nicht wieder erneut starten, laden und einloggen, sondern können sich direkt nach erfüllter Kaffee-Maschinen-Versorgung wieder anmelden.

Probleme die entstanden sind

#### Technische Seite

Aus technischer Sicht gab es nur ein Problem:

Die Verbindung zur Datenbank bleibt während der Laufzeit bestehen. Das ist sicherheitstechnisch erstmal kein Problem, solange Firewall und andere Komponenten des Netzwerks dem entsprechend eingerichtet sind. Das größere Problem ist eine dauerhafte Grund-Auslastung des Netzes.

Vorteil ist jedoch, dass Daten in Lifetime abgerufen werden können, ohne zuerst eine Verbindung aufbauen und danach wieder trennen zu müssen. Das ist zeitlich betrachtet ein Vorteil, jedoch nicht Grund genug, um diesen Fehler ins positive reden zu können.

Wir hatten versucht die Verbindung bei jeder Abfrage durch eine separate Methode zu öffnen und danach wieder zu schließen. Das ist uns aus zeitlichen und organisatorischen Gründen dann nicht mehr gelungen, da die Implementierung der neuen Methode ebenfalls wieder Fehler nach sich zog.

#### Organisatorische Seite

Aus organisatorischer Sicht war die Wahl der Abgleich-Plattform nicht gut getroffen. Git-Hub ist eine solide Plattform für das Übertragen und abgleichen von Daten, jedoch war das Einrichten auf die Clients ein nervenaufreibendes Unterfangen und hatte zu Folge, dass die ein oder anderen Verweise jedesmal erneut „aktualisiert“ werden mussten.

#### Team

Durch direkte und offene Kommunikation war die Zusammenarbeit als Team keinerlei Problem und trotz dem stetigen Ziel vor Augen ein gemütliches Zusammensein.

## Ausblick Erweiterungsmöglichkeiten

Nach der Einarbeitungs-Phase der Praxis Lustigmann werden Probleme, Erweiterungen und neue Anforderungen entstehen.

In diesem Zuge haben wir Techniker uns bereits überlegt, was für zukünftige Aspekte realisierbar sind und welchen Vorteil die Praxis daraus ziehen würde:

- Moment ist die Software eine reine „Kontakt-Bibliothek“. Patienten werden erfasst, momentane Beschwerden können in einem Feld eingetragen werden und neue Patienten angelegt. Das Prinzip einer **langanhaltenden und detaillierten Erfassung** der Patienten und deren Beschwerden ist nicht vorgesehen. In Zukünftigen Updates wäre das Aufbauen einer separaten Rubrik „Krankheiten und Probleme“ auf Patienten-Ebene sinnvoll, da Patienten in der Regel nicht nur einmal zum Arzt gehen und auch mehrere Krankheiten auftreten können. Hierfür müsste man die Datenbank um eine weitere Tabelle erweitern, in der als Sekundärschlüssel die Patientennummer angegeben wird. Dadurch lässt sich jeder Erkrankung einem Patienten zuordnen, der wiederum mehrere Krankheiten hat. Hinzu kommt die detailliertere Erfassung der Probleme, des Verlaufs und evtl. Medikamente. In der grafischen Oberfläche müsste man hierfür nur einen weiteren Reiter einbauen, der automatisch mit dem momentan ausgewählten Patienten aufgerufen wird.
- Die **Login-Funktion** dient momentan nur der Authentifizierung eines praxisinternen Mitarbeiters. Eine 1. Gliederung der Nutzer in verschiedene Gruppen und 2. Individualisierung der einzelnen Nutzern hilft bei der Erfassung, der Darstellung gewisser Bereiche für nur berechnigte Nutzer und langfristig sogar ein Individuelles Layout, da jeder Nutzer anders mit der Software arbeitet und für die unterschiedlichen Gruppen auch unterschiedliche Workflows vorgesehen sind (einem Arzt ist das „Eintritts-Datum“ des Patienten in die Praxis egal, der Empfangs-Dame jedoch nicht, da Sie evtl. noch Unterlagen benötigt, die der Patient beim letzten Besuch nicht zur Hand hatte).
- Die Vorbelegung bei der Neuanlegung von Patienten erfolgt noch manuell (bis auf das Datums-Feld). Hier wäre eine automatisierte Vorbelegung (z.B. Telefonnummer +49) durch die Konfiguration der Software denkbar. Ebenfalls wäre denkbar die, von Patienten vorzulegende Krankenkassen-Karte als Basis der Datenbefüllung zu nutzen, da dort ohnehin die entsprechenden Daten zum Teil hinterlegt sind.

## Fazit, Reflexion

Das Projekt war ein guter Ausgleich zum schulischen Alltag mit frontalem Unterricht, bot die Möglichkeit eigene Ideen umzusetzen und basierte auf einem bereits erfolgten Projekt, was die Erarbeitung und die Entwicklung enorm erleichterte.

Für zukünftige Projekte läge in unserem Team der Fokus auf ausgearbeiteten Konzepten vor Entwicklungs-Beginn, damit dieser besser von Stattem geht, Aufgaben klarer verteilt werden können und Aufgaben, die gleichzeitig erfolgen können auch den zeitlichen Faktor reduzieren.