

Zema:

Applikation zum strukturierten  
Alltags- Zeitmanagement

ARCHITEKTURENTWURF,  
SOFTWAREENGINEERING

für die Prüfung zum  
Bachelor of Science  
des Studienganges Informatik  
an der  
Dualen Hochschule Baden-Württemberg Karlsruhe  
von  
**Alica Penndorf**

Abgabedatum 15. Juni 2020

Matrikelnummer	4245158
Kurs	tin18B5
Ausbildungsfirma	Fraport AG Frankfurt am Main
Gutachter der Studienakademie	Dr.-Ing. Sascha Alpers

<i>Version</i>	<i>Änderungsstand</i>	<i>Autor</i>
1.0	initiale Fassung	Alica Penndorf

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung und Glossar</b>	<b>3</b>
1.1	Hintergrund der Applikation . . . . .	3
1.2	Glossar . . . . .	4
<b>2</b>	<b>Statische Sichten</b>	<b>5</b>
2.1	Statische Sicht- zu persistierende Daten . . . . .	5
2.2	Statische Sicht- Objekttypen zur Laufzeit . . . . .	8
<b>3</b>	<b>Dynamische Sichten</b>	<b>10</b>
3.1	Funktionale Einheiten der Anforderungen . . . . .	10
3.2	Exemplarische Prozessbeschreibung für eine Funktionseinheit . . . . .	14
<b>4</b>	<b>Technologiestack</b>	<b>16</b>
4.1	Auswahl des Technologiestack . . . . .	16
<b>5</b>	<b>Fazit</b>	<b>18</b>

# Kapitel 1

## Einleitung und Glossar

### 1.1 Hintergrund der Applikation

Die Motivation, die hinter Zema steht, ist bereits in dem zuvor verfassten Anforderungsdokument dargestellt, wird hier aber dennoch in geringem Umfang erläutert.

Eine Schülerin der deutschen Oberstufe äußerte den Wunsch einer funktionsfähigen Applikation zur unterstützenden Planung des Schulalltages. Diese Schülerin forderte gleichermaßen, eine solche Applikation auch für das englische Schulsystem nutzen zu können. Aufgrund dieser Kriterien wurden die Anforderungen erhoben (siehe Anforderungsdokument) und festgestellt, dass die Anwendergruppe ein System wünscht, welches dem Nutzer Gestaltungsfreiheit lässt. Das bedeutet, dass die Anwendung keinen Bezug zu einem Schulsystem nimmt und die Übersicht allgemein gestaltet wird.

Demnach soll Zema die Planung und Organisation eines Tagesablaufes digitalisieren und dem Anwender die Möglichkeit, des ständigen Zugriffs auf die eingetragenen Daten geben. Dies soll mit verschiedenen Funktionen umgesetzt werden; das Menü soll übergreifend aufzeigen, welche verschiedenen Funktionen für den Nutzer zur Verfügung stehen. Diese bestehen bei Erfüllung aller Anforderungen, aus einer Kalenderfunktion, einer To-Do-Listen-Funktion und einer Zeiterfassungsfunktion ("Zeta"). Der Anwender soll so die Möglichkeit bekommen, in verschiedenen Ansichten seinen bevorstehenden Tag zu planen, sowie den vergangenen Tagesablauf zu dokumentieren und eigenständig zu reflektieren.

In diesem Kontext soll "Zema" als Android-App bereitgestellt werden. Dieses Dokument dient der Beschreibung des Architekturentwurfes von "Zema".

## 1.2 Glossar

allgemein

innerhalb der Anwendung verwendete Bezeichnungen, sowie das Layout, geben keinen Aufschluss darüber, für welches Anwendungsszenario diese gedacht sind. Der Nutzer kann sich die Inhalte individuell gestalten.

Rubrik

Themenseiten zur unterschiedlichen Eintragserfassung, sowie die Einstellungen

Applikationsgerüst

Hiermit sind die Funktionalen Einheiten beschrieben, die im Wesentlichen, das Konstrukt der Applikation abbilden.

# Kapitel 2

## Statische Sichten

### 2.1 Statische Sicht- zu persistierende Daten

Zu persistierende Daten bei Zema spielen für weitere Anwendungsversionen und Weiterentwicklungen der Anwendung eine bedeutende Rolle. Schließlich sollen diese eine Möglichkeit der Datensicherung bekommen, um auch zu einem späterem Zeitpunkt genutzt werden zu können und Fortschritte an den letzten Datenstand anzupassen und zu integrieren.

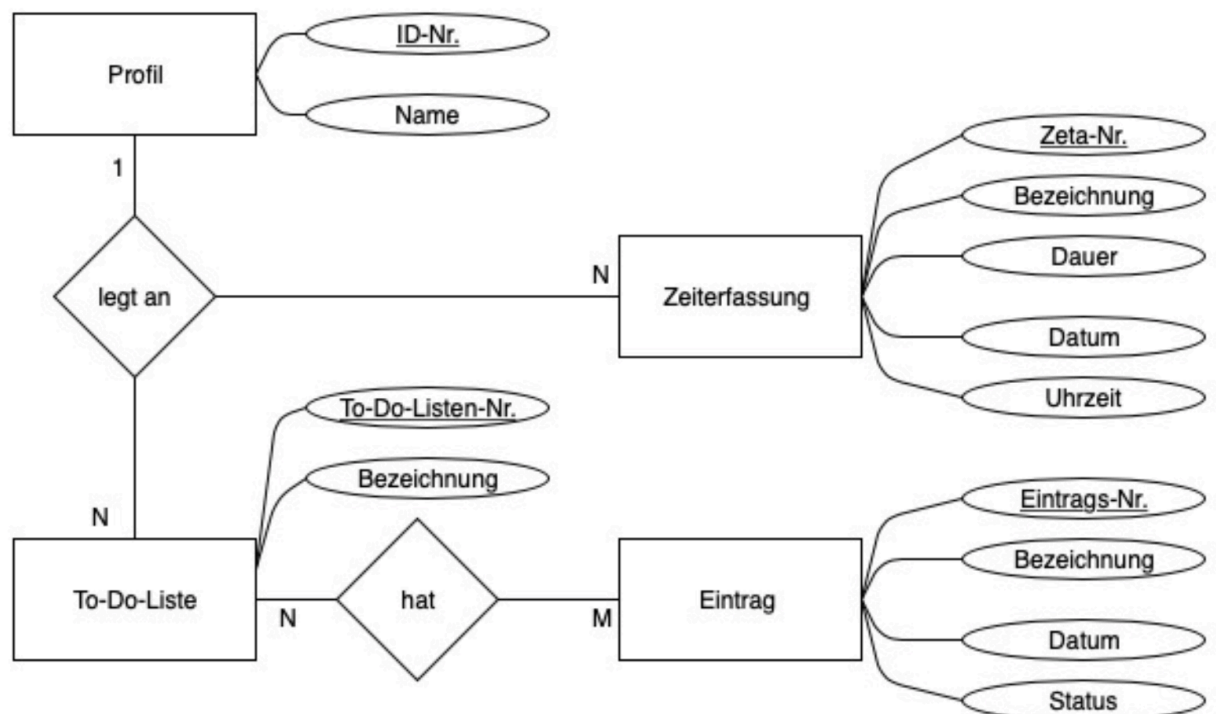


Abbildung 2.1: Zema: ERM

Das oben abgebildete Entity- Relationship- Modell (2.1) zeigt die Daten, die im Falle einer Weiterentwicklung von “Zema” und somit einem Update für den Endnutzer, nicht verloren gehen dürfen. Das Datenmodell soll folglich auch über den ersten Entwicklungsstand verfügbar bleiben.

Der Nutzer wird, beim ersten Start von Zema, aufgefordert, einen Profilnamen anzugeben. Dieser Profilname soll dazu dienen, den Anwender, während der Nutzung von verschiedenen Funktionen, persönlich anzusprechen. Das “Profil“, welches nur lokal genutzt werden soll, hat eine ID- Nummer sowie einen Namen. Der Name wird in einem String gespeichert. Die ID- Nummer in einem Integer- Wert.

Hat der Anwender eine “To-Do-Liste“, und möglicherweise auch Einträge, angelegt, und noch nicht als bearbeitet gekennzeichnet, so sind diese Daten, bei einem Versionsupdate auch weiterhin zu Verfügung zu stellen. Dies gilt auch für die Einträge in die (“Zeta“)Zeiterfassungsfunktion. Diese Daten bleiben bestehen, da der Endnutzer sie nur durch aktives Löschen wieder aus der Ansicht entnehmen kann. Jede “Zeiterfassung“ benötigt einen eindeutigen Indikator, welcher in der “Zeta- Nr. gespeichert wird. Auch hier ist der Datentyp ein Integer. Zusätzlich soll die Dauer als Integer, das Datum und die Uhrzeit jeweils als date und die Bezeichnung als String abgespeichert werden. Die Dauer wird durch einen Integer abgebildet, da der Nutzer in der Graphischen Oberfläche ein Eingabefeld, sowie “min“ (Minuten) angezeigt bekommen soll.

Die “To-Do-Liste“ hat ebenso eine ID vom Datentyp Integer, sowie einen Namen, welcher als String abgespeichert wird. Dies stellt für den Anwender lediglich die Bezeichnung seiner Liste(-n) dar. Die Einträge selber sind jeweils mit einer ID: der Eintrags- Nr. als Integer, einer Bezeichnung als String, und einem (Anlege-)Datum als date versehen. Zusätzlich hat ein Eintrag noch einen Status. Dieser wird als boolean erfasst, da der Anwender entscheiden können soll, ob er den Eintrag bereits erledigt hat, oder ob er noch aussteht. Dabei wird ein Status, der als erledigt gilt auf “true” und ein ausstehender Eintrag auf “false“ gesetzt.

Im Zuge der Datenerhebung ist ein relationales Datenbanksystem gewählt worden, da die Daten strukturiert sind.

Die hier aufgezeigten Entitäten sind für die Umsetzung der Anwendung alle relevant, da diese die essentiellen (sowie einzigen) Funktionen, aus welchen das Gesamtsystem resultiert, möglich machen werden.

Im Folgenden sind zu jeder Entität Beispieldaten aufgelistet.

	Integer	String
• Profil:	ID-Nr.	Name
	0001	Alex

Tabelle 2.1: Zema: Beispieldaten Profil

	Integer	String
	To-Do-Listen-Nr.	Bezeichnung
	00011	Hausaufgaben
• To-Do-Liste:	00012	Putzen
	00013	Einkauf
	00014	Zu Lernen
	00015	Anrufen
	00016	Bewerbungen

Tabelle 2.2: Zema: Beispieldaten To-Do-Liste

	Integer	String	Date	Boolean
	Eintrags-Nr.	Bezeichnung	Datum	Status
	00111	SE Wiederholung	10.02.2020	erledigt
• Eintrag:	00112	Wohnung staubsaugen	13.05.2020	erledigt
	00113	Mehl einkaufen	19.05.2020	ausstehend
	00114	Waschmittel auffüllen	01.02.2020	erledigt
	00115	Georg anrufen	14.02.2020	ausstehend
	00116	Lebenslauf updaten	20.05.2020	ausstehend

Tabelle 2.3: Zema: Beispieldaten Eintrag

	Integer	String	Date	Date	Integer
	Zeta-Nr.	Bezeichnung	Datum	Uhrzeit	Dauer
	01111	Investierte Zeit in HA's	28.02.2020	19:30	120
• Zeiterfassung:	01112	Wohnung geputzt	14.05.2020	22:40	100
	01113	Shopping	17.05.2020	17:15	45
	01114	Gaming	08.02.2020	20:45	15
	01115	Familytime	13.04.2020	15:30	130
	01116	Ehrenamt	26.02.2020	18:00	80

Tabelle 2.4: Zema: Beispieldaten Zeiterfassung, "Zeta"



## 2.2 Statische Sicht- Objekttypen zur Laufzeit

Auch in diesem Kapitel soll der Zusammenhang zwischen einzelnen Eigenschaften der Anwendung verdeutlicht werden. Dabei handelt es sich hier auch um solche Daten, die zwar nicht zu persistieren sind, aber dennoch zur Laufzeit benötigt werden. Solche sind beispielsweise Daten, die für die graphische Nutzeroberfläche verwendet werden müssen. Um diese darzustellen wird ein Klassenmodell aufgeführt. Abbildung 2.2 bildet alle bislang bekannten Objekttypen ab.

*Anmerkung: Da die Entwicklung des Programmes noch nicht weiter fortgeschritten ist, hat dieses Modell keine Vollständigkeit zum Zeitpunkt dieser Abgabe.*

Das unten dargestellte Klassendiagramm zeigt zudem auf, dass es in der (lokalen) Anwendung *Zema* nur ein Profil zu führen gibt. Diesem Nutzerprofil ist es allerdings möglich, beliebig viele Zeiterfassungen, sowie beliebig viele To-Do-Listen anzulegen.

Nur nach Erstellung einer To-Do-Liste, kann ein Eintrag gemacht werden. Dabei gilt: Eine To-Do-Liste hat beliebig viele Einträge. Ein Eintrag hat mindestens eine To-Do-Liste.

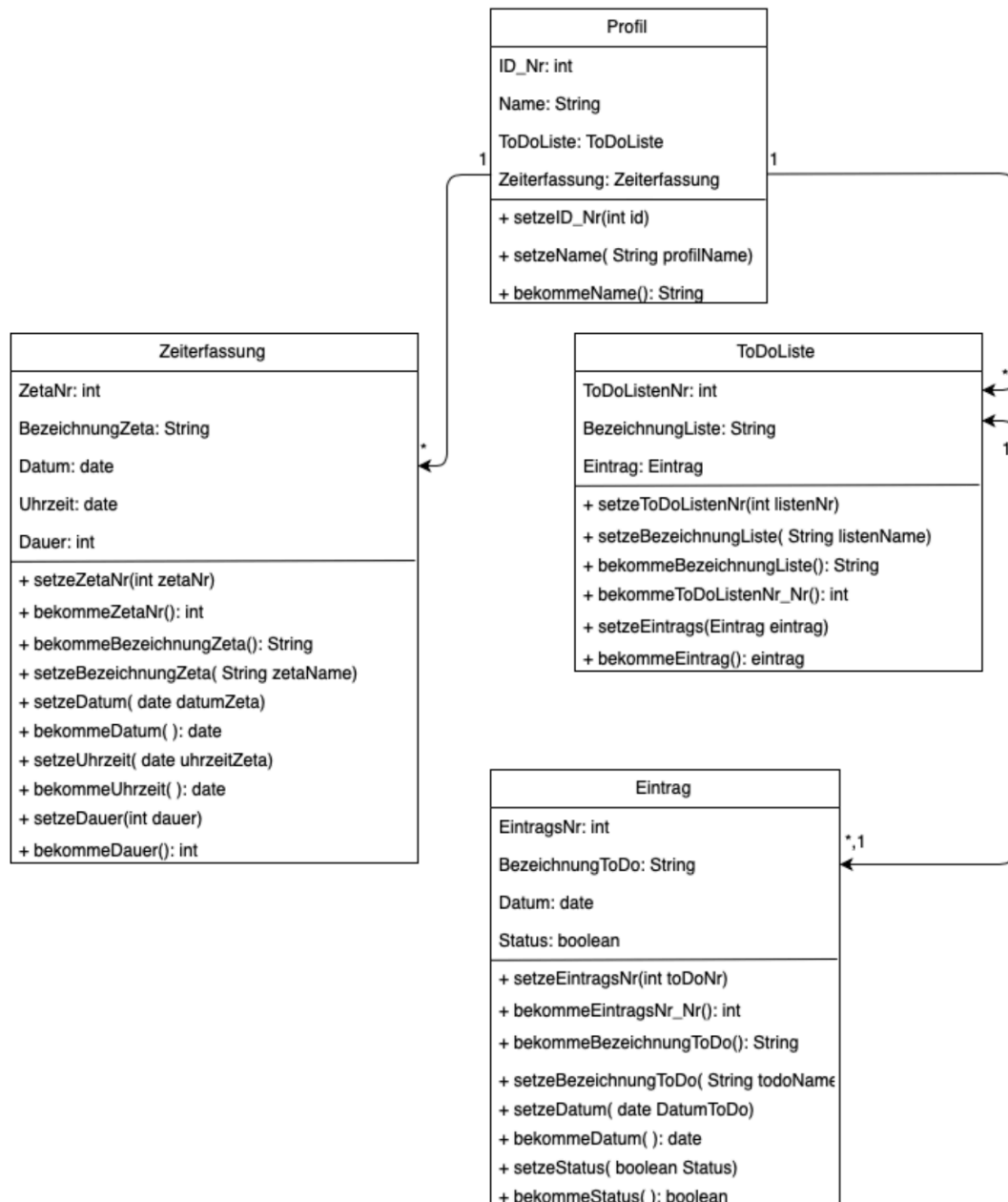


Abbildung 2.2: Zema: Klassendiagramm

# Kapitel 3

## Dynamische Sichten

### 3.1 Funktionale Einheiten der Anforderungen

Im Folgenden sind die Anforderungen, die im Anforderungsdokument dargelegt wurden, in Funktionseinheiten aufgeteilt. Die Referenzen beziehen sich somit auf das vorherige Dokument. Diese Funktionseinheiten sind tabellarsich aufgelistet und weisen auf deren Priorisierung. Diese lautet wie folgt:

#### **FE 1- MUST**

Die Anforderungen, die unter FE 1 aufgelistet sind, sind essentiell für den Erfolg der Anwendung. Das bedeutet, sie sind von sehr hoher Wichtigkeit und verdienen ein hohes Maß an Beachtung während der Realisierung dieser Funktionen.

#### **FE 2- SHOULD**

Die Anforderungen, die unter FE 2 aufgelistet sind, sind notwendig für die Anwendung, aber nicht grundsätzlich zwingend zu erfüllen. An diesen, lassen sich Änderungswünsche, Abwandlungen oder Vereinfachungen erneut diskutieren.

#### **FE 3- COULD**

Die Anforderungen, die unter FE 3 aufgelistet sind, sind für die Umsetzung erbeten. Kommt es aber, während der Entwicklung, zu einem Zeitmangel oder anderen Komplikationen, so sind nur die Anforderungen erstrebenswert, die den größten Nutzen für den Anwender mit sich bringen.

**FE 4- WON'T** Die Anforderungen, die unter FE 4 aufgelistet sind, sind bei ausge-

schöpftem Leistungsumfang, für eine spätere Version zu dokumentieren und vorerst nicht umzusetzen.

Anforderungs- Nummer	Anforderungsbeschreibung	Referenz	FE-Verweis
R1	Zema starten	Kapitel 3, S. 24	FE 1
R2	Profilnamen anlegen	Kapitel 3, S. 24	FE 1
R3	Frühzeitiges Beenden	Kapitel 3, S. 25	FE 1
R4	Erneutes Öffnen von Zema, ohne zuvor einen Profilna- men angelegt zu haben	Kapitel 3, S. 25	FE 1
R5	Anzahl Benutzer	Kapitel 3, S. 26	FE 1
R6	Menü	Kapitel 3, S. 26	FE 1
R7	Menüauswahl	Kapitel 3, S. 27	FE 1
R8	Weiterleiten zu Rubrik	Kapitel 3, S. 27	FE 1
R9	Menüauswahl beenden ohne Auswahl getroffen zu haben	Kapitel 3, S. 28	FE 3
R10	Anwählen des Kalenders	Kapitel 3, S. 29	FE 1
R11	Kalender	Kapitel 3, S. 28	FE 2
R12	Kalenderansicht	Kapitel 3, S. 29	FE 3
R13	Kalendereintrag anlegen	Kapitel 3, S. 30	FE 2
R14	Kalendereinträge in der Ge- samtansicht	Kapitel 3, S. 30	FE 3
R15	Kalendereinträge anzeigen	Kapitel 3, S. 31	FE 2
R16	Bestehenden Kalenderein- trag anwählen	Kapitel 3, S. 31	FE 2
R17	Bestehenden Kalenderein- trag bearbeiten	Kapitel 3, S. 32	FE 3
R18	Bestehenden Kalenderein- trag löschen	Kapitel 3, S. 32	FE 2
R19	Kalendereintrag speichern	Kapitel 3, S. 33	FE 2
R20	Kalendereintragserfassung verlassen	Kapitel 3, S. 33	FE 4
R21	Kalendertag wechseln	Kapitel 3, S. 34	FE 4
R22	Anwählen der Einstellungen	Kapitel 3, S. 34	FE 1
R23	Profilnamen ändern	Kapitel 3, S. 35	FE 3
R24	Anwählen der To-Do-Liste	Kapitel 3, S. 35	FE 1

R25	To-Do-Listenansicht	Kapitel 3, S. 36	FE 2
R26	Erstellung einer neuen To-Do-Liste	Kapitel 3, S. 36	FE 2
R27	Auswahl: To-Do-Liste löschen oder Namensbearbeitung	Kapitel 3, S. 37	FE 2
R28	Auswahl: To-Do-Liste löschen oder Namensbearbeitung	Kapitel 3, S. 37	FE 3
R29	Löschen einer bestehenden To-Do-Liste	Kapitel 3, S. 38	FE 2
R30	Bearbeitung eines Listennamens	Kapitel 3, S. 38	FE 3
R31	Abspeichern des neuen Listennamens	Kapitel 3, S. 39	FE 3
R32	Listeneintrag	Kapitel 3, S. 39	FE 2
R33	Listeneintragserfassung beenden	Kapitel 3, S. 40	FE 3
R33(a)	Listeneintrag als erledigt kennzeichnen	Kapitel 3, S. 40	FE 2
R34	Listeneintrag bearbeiten	Kapitel 3, S. 41	FE 4
R35	Listeneintrag löschen	Kapitel 3, S. 41	FE 2
R36	Entfernung eines Listeneintrages	Kapitel 3, S. 42	FE 2
R38	Anwählen der Zeiterfassung	Kapitel 3, S. 42	FE 1
R39	Zeiterfassung	Kapitel 3, S. 43	FE 2
R40	Ansicht: Zeiterfassung	Kapitel 3, S. 43	FE 4
R41	Eintrag zur Zeiterfassung hinzufügen	Kapitel 3, S. 44	FE 2
R42	Daten zur Zeiterfassung hinzufügen	Kapitel 3, S. 44	FE 4
R43	Daten in Zeiterfassung speichern	Kapitel 3, S. 44	FE 4
R44	Eine Zeiterfassung anwählen und optional löschen	Kapitel 3, S. 44	FE 4

R45	Zeiterfassung löschen	Kapitel 3, S. 45	FE 4
R46	Eine Zeiterfassung anwählen ohne diese ver- ändern zu wollen	Kapitel 3, S. 46	FE 4
R47	Zema beenden	Kapitel 3, S. 46	FE 1

Tabelle 3.1: Zema: Funktionale Einheiten und Verweise auf Priorisierung

## 3.2 Exemplarische Prozessbeschreibung für eine Funktionseinheit

### FE 1

FE 1 beschreibt nach Auswertung der oben aufgezeigten Tabelle, alle Funktionalen Einheiten, die unerlässlich für ein “Applikationsgerüst” sind. Bei Zema sind das die Anforderungen, die den Start und das Beenden der Anwendung, das Anlegen eines Nutzernamens, sowie das Öffnen des Menüs und der einzelnen Rubriken, ermöglichen. Die unten stehende Abbildung (3.1) verdeutlicht diesen Prozess in einem ERM.

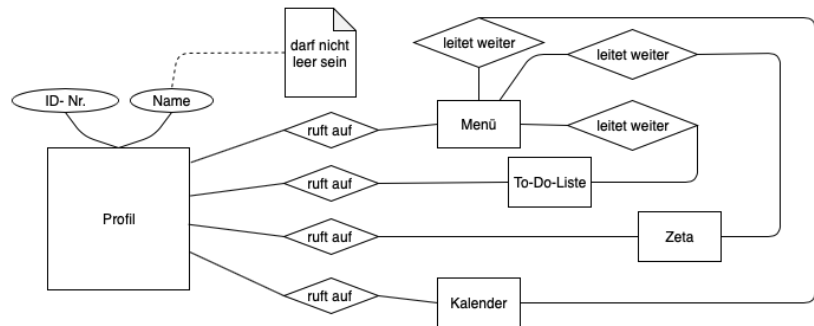


Abbildung 3.1: Zema: ERM



# Kapitel 4

## Technologiestack

### 4.1 Auswahl des Technologiestack

Da Zema Daten beinhalten wird, welche zu persitsieren gelten (siehe Kapitel 2.1), muss ein Datenbanksystem gewählt werden. Zema soll lokal als Android- Applikation bereitgestellt werden. Dabei wird auf einen webbasierten Zugang verzichtet. Entwickelt wird diese mit Java und Android Studio und MySQL als Datenbank.

#### Android Studio

Android Studio ist eine kostenfreie Entwicklungsumgebung von Google. Zudem ist es die offizielle Entwicklungsumgebung für Android- Apps.

#### Anforderungen an das Datenbanksystem

Anforderungs- Nr.	Kriterium
1	Kostengünstig
2	Zentrale Lauffähigkeit
3	Kompatibilität mit IOS
4	Java als Programmiersprache unterstützt
5	Relationale Datenbank
6	Erfahrung mit der Datenbank

Tabelle 4.1: Zema: Anforderungskriterien an das Datenbanksystem

All diese Anforderungskriterien sind im Kontext der Zema-Anwendungsentwicklung als K.O.-Kriterien eingestuft worden.

**Datenbanksysteme**

Da sich das Wissen und der Umgang mit einem Datenbanksystem lediglich auf MySQL beschränkt, wurde MySQL (DB1) mit der Oracle Database (DB2) verglichen.

K.O.- Kriterien	1	2	3	4	5	6
DB1	JA	JA	JA	JA	JA	JA
DB2	JA	JA	JA	JA	JA	NEIN

Tabelle 4.2: Zema: Entscheidungsmatrix

Beide Datenbanksysteme haben in Bezug auf die K.O.-Kriterien 1-5 gleichermaßen abschneiden können. Der Grund für die Wahl von MySQL bezieht sich auf Kriterium 6. Eine Einarbeitung in ein neues Datenbanksystem kann somit umgangen werden.

# Kapitel 5

## Fazit

In diesem Architekturdokument wurde die zu erstellende Softwarearchitektur für Zema diskutiert. Es wird sich vorbehalten, Änderungen und/ oder Ergänzungen während der nächsten Entwicklungsschritte vorzunehmen.