

Diplomarbeit

Winetastic



**Höhere Technische Lehranstalt Perg
für Informatik**

Fertigstellung am 05.04.2017

Projektauftraggeber:

Sabine Stumvoll

Projektteam:

Felix Stumvoll

Ronald Peroutka

Danksagung

Wir bedanken uns bei all jenen, die uns im Laufe der Diplomarbeit unterstützt haben, sehr herzlich.

Einerseits bedanken wir uns bei Familie Stumvoll, genauer gesagt bei Frau Dr. Sabine Stumvoll, da ohne sie, diese Arbeit nie zustande gekommen wäre.

Besonderen Dank möchten wir außerdem gegenüber Frau Prof. Mag. Gabriela Danner kundtun, die uns während des ganzen Prozesses organisatorisch beistand. Ein weiterer Dank geht an Herrn Prof. Dipl.-Ing. Helmut Otto, der uns bei technischen Fragen immer gut beraten hat.

Herzlichen Dank!

Eigenständigkeitserklärung

Hiermit versichern wir, die vorliegende Arbeit selbständig, ohne fremde Hilfe und ohne Benutzung anderer als der von uns angegebenen Quellen angefertigt zu haben. Alle Stellen, die wörtlich oder sinngemäß aus fremden Quellen direkt oder indirekt übernommen wurden, sind als solche gekennzeichnet.

.....
Felix Stumvoll

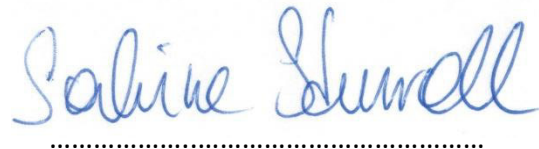
Projektleiter

.....
Ronald Peroutka

Projektmitglied

Einverständniserklärung

Ich, Dr. Sabine Stumvoll, Veranstalterin der zweimal im Jahr stattfindenden Weinverkostung, erkläre mich hiermit einverstanden, dass die Schüler der HTL Perg eine Diplomarbeit über die Veranstaltung erstellen dürfen und ich, die dafür benötigten Daten und Informationen zur Verfügung stellen werde.

A handwritten signature in blue ink that reads 'Sabine Stumvoll'. Below the signature is a horizontal dotted line.

Dr. Sabine Stumvoll

Auftraggeber

Kurzbeschreibung

Die Diplomarbeit „Winetastic“ dient als Zusatz und Unterstützung der zweimal jährlich stattfindenden Weinverkostungsveranstaltung, welche von der Familie Stumvoll organisiert wird. Dort treten mehrere Winzer auf und präsentieren ihre Weine.

Derzeit werden die Winzer und deren Produkte in einer Excel Datei gespeichert und in einem ausgedruckten Katalog präsentiert.

Winetastic soll diese Daten jetzt auch elektronisch zur Verfügung stellen. Die Diplomarbeit besteht aus drei Teilen:

- Admin Tool
- Datenbank
- App

Das Admin Tool wird den Veranstalter unterstützen, indem das Programm die bestehenden Daten von einem Excel File einliest und die CRUD Operationen (Create, Read, Update, Delete) implementiert. Der Anwender kann dabei Veranstaltungen erstellen und diesen die dort auftretenden Winzer und deren Weine zuordnen.

Diese Daten werden anschließend in eine Azure Mobile Service Datenbank persistiert. Diese Datenbank dient als Datenquelle für die Smartphone Clients.

In einer Hybrid-App werden die Winzer und Weine, mit all deren Details präsentiert. Darüber hinaus können Anwender eigene Notizen zu den Weinen speichern und diese auch bewerten.

Die abgegebenen Bewertungen werden ebenfalls in der Datenbank persistiert, damit diese auch von anderen Benutzern gesehen werden können.

Das Ziel der Diplomarbeit ist eine Vereinfachung der Datenverwaltung für die Veranstaltungen und ein Zuwachs an jüngeren Teilnehmern an der Veranstaltung.

Abstract

The diploma thesis *Winetastic* is an addition and support of the wine tasting event, which takes place twice a year. The family Stumvoll organizes this event, where several wineries show their wines. Currently all existing winery and their products are stored in an Excel File and presented in a catalog. Every time when a new event takes place, the organizer has to create a new catalog and has to update the old Excel file.

Winetastic should provide these data electronically. The diploma thesis consists of three parts:

- Admin Tool
- Database
- App

The Admin Tool should read the existing data from an Excel File and implements the CRUD (Create, Read, Update, and Delete) operations.

The data is stored in an Azure database.

Winery and their products are presented in an App. Users can save their own information about products and are able to rate the wines.

The aim of the diploma thesis is to simplify data management and to increase the number of younger members at the event.

Inhaltsverzeichnis

1	Einleitung.....	16
1.1	Auftraggeber.....	16
1.2	Projektteam	16
1.3	Ist Zustand & Ziel Zustand	16
1.3.1	Ist Zustand	16
1.3.2	Ziel Zustand	16
2	Die Veranstaltung.....	16
2.1	Winzer.....	16
2.2	Weine.....	16
2.3	Katalog.....	17
3	Funktionalität	18
3.1	Admin Tool.....	18
3.1.1	Veranstaltungsfenster	18
3.1.2	Datenseite	19
3.1.3	Winzer hinzufügen Fenster	25
3.1.4	Weine hinzufügen Fenster	26
3.1.5	Weinsorten Fenster.....	27
3.2	App.....	28
3.2.1	Startseite	28
3.2.2	Veranstaltungsfenster	29
3.2.3	Sidebar.....	29
3.2.4	Wein Fenster	30

3.2.5	Weindetail Fenster	31
3.2.6	Winzer Fenster	33
3.2.7	Winzerdetails Fenster.....	34
3.2.8	Bestenliste	35
3.2.9	Meine Notizen	35
3.2.10	Meine Bewertungen.....	36
3.2.11	Einstellungen	37
3.2.12	Impressum.....	37
3.2.13	Bewertungssystem	37
4	Grundlagen.....	38
4.1	Anfängliche Überlegungen	38
4.1.1	Datenbank	38
4.1.2	App Entwicklung.....	39
4.2	Aufbau und Kommunikation	40
4.2.1	Allgemein.....	40
4.2.2	Veranstalteransicht	41
4.2.3	Anwenderansicht	42
4.3	Benötigte Ressourcen.....	43
4.3.1	Microsoft Visual Studio	43
4.3.2	Android Smartphone.....	43
4.3.3	SmartGit	43
4.4	Verwendete Technologien	44
4.4.1	Mobile Applikation	44
4.4.2	Admin Tool	53

4.4.3	Datenbank	53
4.5	Datenbankmodell	54
4.6	Native Apps vs. Hybrid-Apps	57
4.6.1	Native Apps	58
4.6.2	Hybrid-Apps.....	60
4.6.3	Warum Hybrid-App	62
5	Organisation	63
5.1	Kommunikation im Team	63
5.2	Meilensteine.....	64
5.3	Risiken.....	65
5.3.1	Neue Technologie Cordova	65
5.3.2	Excel Import nicht möglich.....	66
5.3.3	Schlechte Kommunikation zum Auftraggeber	66
5.4	Probleme	67
5.4.1	App	67
5.5	Maßnahmen zur Qualitätssicherung.....	68
5.5.1	Review mit dem Auftraggeber	68
5.5.2	Rücksprache unter Teammitgliedern.....	68
6	Persönliches Resümee.....	69
6.1	Stumvoll Felix – Admin Tool	69
6.2	Ronald Peroutka – App.....	70
7	Literaturverzeichnis.....	71
8	Abbildungsverzeichnis.....	72
9	Abkürzungsverzeichnis.....	74

1 Einleitung

2 Die Veranstaltung

Die Veranstaltung findet 2-mal jährlich statt, einmal im Frühling und einmal im Herbst. Auf dieser Veranstaltung können Weine von bis zu 40 Winzern verkostet werden. Alle Winzer und Weine sind in einem Katalog für die Besucher festgehalten.

2.1 Winzer

Auf einer Veranstaltung sind meist bis zu 40 Winzer anwesend und präsentieren ihre Weine für die Besucher. Durchschnittlich werden von jedem Winzer circa acht Weine zur Verkostung bereitgestellt. Jeder Winzer hat dabei eine Tischnummer.

Jeder Winzer besitzt folgende Daten:

- Tischnummer
- Name des Weinguts
- Anbauregion
- Adresse
 - Postleitzahl
 - Straße
 - Ort
- Telefon und Fax
- E-Mail-Adresse
- Homepage

2.2 Weine

Ein Wein wird von einem Winzer präsentiert. Ein Wein kann von einem Winzer auch auf mehreren Veranstaltungen präsentiert werden.

Ein Wein wird durch folgende Daten definiert:

- Jahrgang
- Rebsorte
- Lage/Bezeichnung
- Qualitätsstufe

2.3 Katalog

Der Katalog steht für Besucher der Veranstaltung kostenlos zur Verfügung. Im Katalog sind alle Winzer und deren Weine verzeichnet. Weiters gibt es genügend Platz, um sich Notizen zu machen.

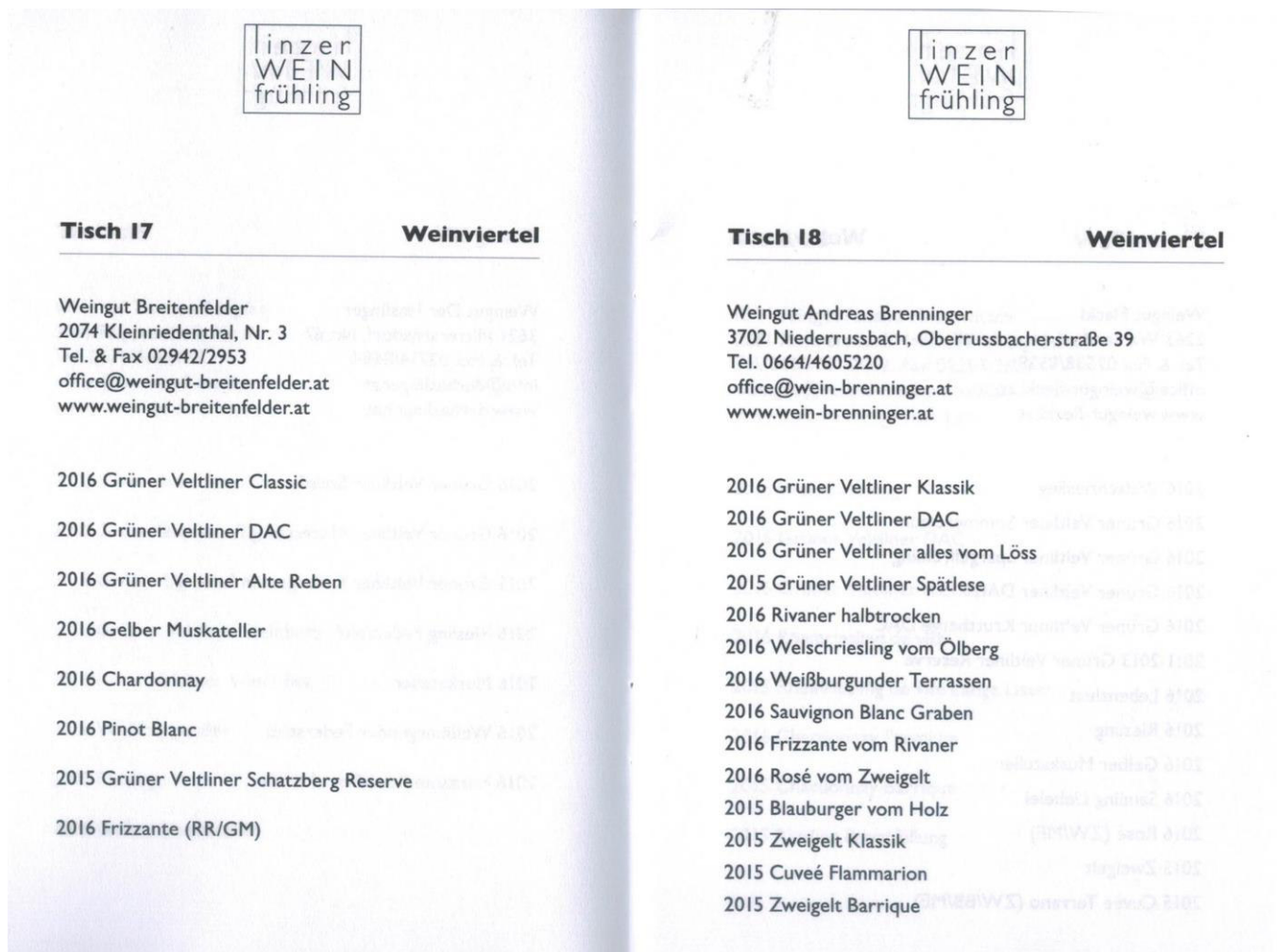


Abbildung 1 – derzeitiger Katalog

3 Funktionalität

3.1 Admin Tool

3.1.1 Veranstaltungsfenster

In diesem Fenster werden alle Veranstaltungen mit Name, Datum, Uhrzeit und Veranstaltungsort in einem DataGrid angezeigt. Mit einem Klick auf die gewünschte Veranstaltung kann man die einzelnen Felder dieser Veranstaltung bearbeiten.



Name	Datum	Uhrzeit	Veranstaltungsort
Linzer Weinfrühling 2017	21.04.2017	14:00 bis 21:00	Linzer Vereinshaus
Linzer Weinherbst 2017	10.11.2017	14:00 bis 21:00	Linzer Vereinshaus
Linzer Weinfrühling 2018	06.04.2018	14:00 bis 21:00	Linzer Vereinshaus

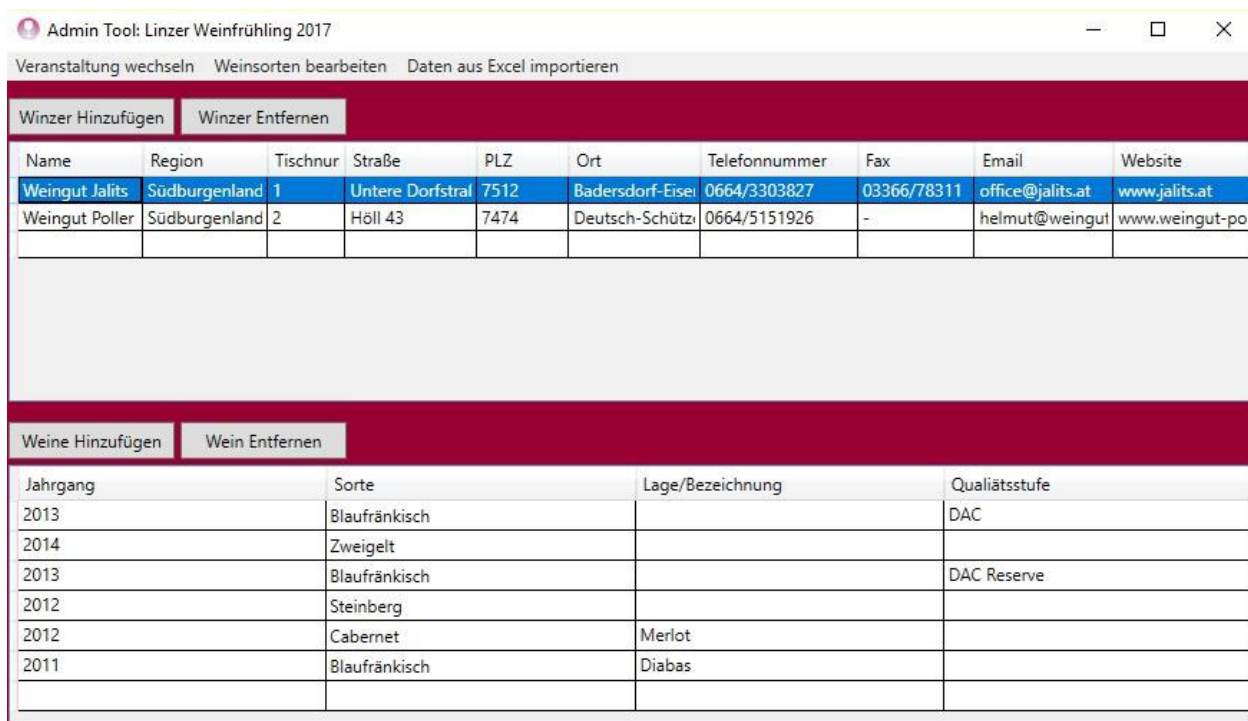
Abbildung 2 – Admin Tool - Veranstaltungsfenster

Das Datum lässt sich über eine Datumsauswahl festlegen. Dabei können nur Daten ab dem aktuellen ausgewählt werden. Dies verhindert das Anlegen von Veranstaltungen in der Vergangenheit. Möchte man eine Veranstaltung bearbeiten, welche bereits stattgefunden hat, öffnet sich ein Pop-Up Fenster. Dieses Fenster fragt den Benutzer, ob er sich sicher ist, diese Veranstaltung auch wirklich zu bearbeiten. Dies verhindert, dass vergangene Veranstaltungen unabsichtlich bearbeitet werden.

Mit der leeren Zeile am Ende des DataGrids kann eine neue Veranstaltung hinzugefügt werden. Über einen „Entfernen Button“ können Events wieder entfernt werden. Alternativ kann ein Event auch über „Rechtsklick“ > „Entfernen“ gelöscht werden. Dabei wird überprüft, ob Winzer in der Veranstaltung eingetragen sind, da eine Veranstaltung nur entfernt werden darf, wenn diese leer ist.

Hat man eine Veranstaltung ausgewählt, wird man über den „Auswählen Button“ auf die Daten-seite der Veranstaltung geleitet. Hat man keine Veranstaltung ausgewählt, bleibt der „Auswählen Button“ ausgegraut.

3.1.2 Datenseite



Name	Region	Tischnummer	Straße	PLZ	Ort	Telefonnummer	Fax	Email	Website
Weingut Jalits	Südburgenland	1	Untere Dorfstral	7512	Badersdorf-Eisei	0664/3303827	03366/78311	office@jalits.at	www.jalits.at
Weingut Poller	Südburgenland	2	Höll 43	7474	Deutsch-Schützi	0664/5151926	-	helmut@weingut	www.weingut-po

Jahrgang	Sorte	Lage/Bezeichnung	Qualitätsstufe
2013	Blaufränkisch		DAC
2014	Zweigelt		
2013	Blaufränkisch		DAC Reserve
2012	Steinberg		
2012	Cabernet	Merlot	
2011	Blaufränkisch	Diabas	

Abbildung 3 – Admin Tool - Datenseite

Die Datenseite ist in 3 Abschnitte unterteilt:

- Menüleiste
- Winzeransicht
- Weinansicht

3.1.2.1 Menüleiste

Die Menüleiste bietet drei Funktionen: das Wechseln der Veranstaltung, das Bearbeiten der Weinsorten und das Importieren von Daten aus Excel Dateien.

Mit dem Menüpunkt „Veranstaltung wechseln“ kommt man auf die Veranstaltungsübersicht zurück und kann dort andere Veranstaltungen bearbeiten.

Die Funktion „Weinsorten bearbeiten“ wird in [Abschnitt 3.1.3](#) näher erklärt.

Das Formular dient dazu, die Daten aller Winzer zu sammeln, welche auf der jeweiligen Veranstaltung anwesend sind. Die gesammelten Daten werden automatisch in ein Excel File gespeichert, welches als Grundlage für die jeweilige Veranstaltung dient.

Mit dem Menüpunkt „Daten aus Excel importieren“ können Daten aus einem Excel File importiert werden. Das Excel File wird durch ein Online Formular erstellt, in welchem die Winzer Ihre Grunddaten, sowie Ihre Weinliste eingeben.

Name des Betriebes	Anbauregion	Adresse	Ort	Postleitzahl	Telefon	Fax	E-Mail	Homepage
Weingut Jalits	Südburgenland-Eisenberg	Untere Dorfstraße 16	Badersdorf-Eisenberg	7512	0664/3303827	03366/78311	office@jalits.at	www.jalits.at
Weingut Poller	Südburgenland - Eisenberg	Höll 43	Deutsch-Schützen	7474	0664/5151926	-	!mut@weingut-poller.	www.weingut-poller.at

Abbildung 5 - Excel – Winzerdaten

Weinliste
Jahrgang: 2013, Sorte: Blaufränkisch, Lage/Bezeichnung: , Qualitätsstufe: DAC
Jahrgang: 2014, Sorte: Zweigelt, Lage/Bezeichnung: , Qualitätsstufe:
Jahrgang: 2013, Sorte: Blaufränkisch, Lage/Bezeichnung: , Qualitätsstufe: DAC Reserve
Jahrgang: 2012, Sorte: Steinberg, Lage/Bezeichnung: , Qualitätsstufe:
Jahrgang: 2012, Sorte: Cabernet, Lage/Bezeichnung: Merlot, Qualitätsstufe:
Jahrgang: 2011, Sorte: Blaufränkisch, Lage/Bezeichnung: Diabas, Qualitätsstufe:
Jahrgang: 2013, Sorte: Blaufränkisch, Lage/Bezeichnung: Steinried, Qualitätsstufe: DAC
Jahrgang: 2012, Sorte: Blaufränkisch, Lage/Bezeichnung: Kokomandl, Qualitätsstufe: DAC Reserve
Jahrgang: 2013, Sorte: Blaufränkisch, Lage/Bezeichnung: Höll, Qualitätsstufe: DAC Reserve
Jahrgang: 2009, Sorte: Blaufränkisch, Lage/Bezeichnung: Pie5, Qualitätsstufe:
Jahrgang: 2012, Sorte: Cabernet, Lage/Bezeichnung: Sauvignon, Qualitätsstufe:

Abbildung 6 - Excel – Weinliste

Beim Einfügen der Daten wird darauf geachtet, dass bereits existierende Daten nicht doppelt in die Datenbank eingetragen werden. Beim Winzer wird der Name überprüft. Existiert bereits ein Winzer mit diesem Namen, werden die anderen Daten automatisch aktualisiert. Falls kein Winzer mit gleichem Namen gefunden wird, wird dieser neu angelegt. Der Winzer wird anschließend noch der Veranstaltung hinzugefügt.

Bei den Weinen muss der ganze Name überprüft werden. Wird ein Wein mit gleichem Namen in der Datenbank gefunden, wird er zur Veranstaltung hinzugefügt, ansonsten wird er zuerst neu angelegt und dann hinzugefügt. Die Weinsorte des Weines wird ebenfalls überprüft, falls sie noch nicht existiert, wird sie automatisch hinzugefügt.

3.1.2.2 Winzeransicht

Winzer Hinzufügen		Winzer Entfernen							
Name	Region	Tischnr	Straße	PLZ	Ort	Telefonnummer	Fax	Email	Website
Weingut Jalits	Südburgenland	1	Untere Dorfstral	7512	Badersdorf-Eisel	0664/3303827	03366/78311	office@jalits.at	www.jalits.at
Weingut Poller	Südburgenland	2	Höll 43	7474	Deutsch-Schützi	0664/5151926	-	helmut@weingut	www.weingut-po

Abbildung 7 – Admin Tool - Winzeransicht

Im oberen Teil der Seite befindet sich der Winzerbereich. Im Winzerbereich befindet sich ein „Hinzufügen Button“, ein „Entfernen Button“ und ein DataGrid, in dem alle Winzer angezeigt werden, die auf der ausgewählten Veranstaltung präsent sind.

Über den „Hinzufügen Button“ gelangt man zum „Winzer hinzufügen Fenster“, näheres dazu in [Abschnitt 4.1.3](#).

Mit dem „Entfernen Button“ lassen sich Winzer wieder von der Veranstaltung entfernen. Alternativ ist dies auch über „Rechtsklick“ > „Entfernen“ möglich. Beim Entfernen werden auch automatisch alle Weine aus der ausgewählten Veranstaltung entfernt. Die Attribute des Winzers können mit einem Klick bearbeitet werden.

Wählt man einen Winzer aus dem DataGrid aus, werden automatisch im unteren Teil die Weine des Winzers angezeigt, die er auf der ausgewählten Veranstaltung präsentiert.

3.1.2.3 Weinansicht

Weine Hinzufügen		Wein Entfernen	
Jahrgang	Sorte	Lage/Bezeichnung	Qualitätsstufe
2013	Blafränkisch		DAC
2014	Zweigelt		
2013	Blafränkisch		DAC Reserve
2012	Steinberg		
2012	Cabernet	Merlot	
2011	Blafränkisch	Diabas	

Abbildung 8 – Admin Tool - Weinansicht

Wie bei den Winzern gibt es auch bei den Weinen einen „Hinzufügen Button“, einen „Entfernen Button“ und ein DataGrid. In der Suchleiste können Weine des ausgewählten Winzers gesucht werden, dabei kann man nach allen Attributen des Weines suchen.

Über den „Hinzufügen Button“ gelangt man zum „Weine hinzufügen Fenster“. Dazu näheres in [Abschnitt 4.1.4.](#)

Im DataGrid, können die Attribute des Weines mit einem Klick bearbeitet werden. Die Weinsorte des Weines kann über ein Dropdownmenü ausgewählt werden.

Jahrgang	Sorte	Lage/Bezeichnung	Qualitätsstufe
2013	Steinberg	Steinried	DAC
2012	Steinberg	Kokomandl	DAC Reserve
2013	Cabernet	Höll	DAC Reserve
2009	Blafränkisch	Pie5	
2012	Zweigelt	Sauvignon	

Abbildung 9 – Admin Tool – Weinsortenauswahl

3.1.3 Winzer hinzufügen Fenster



Abbildung 10 – Admin Tool – Winzer hinzufügen

In diesem Fenster lassen sich Winzer, die bis zu diesem Zeitpunkt noch nicht eingetragen sind, zur aktuellen Veranstaltung hinzufügen. Im DataGrid werden die Namen der Winzer angezeigt, welche hinzugefügt werden können. Mit der Checkbox neben dem Namen kann man die Winzer auswählen, welche hinzugefügt werden sollen.

Über die Suchleiste im oberen Teil kann man nach Winzern suchen. Dabei wird nur nach dem Winzernamen gesucht.

Mit dem „Fertig Button“ werden die ausgewählten Winzer zur Veranstaltung hinzugefügt.

3.1.4 Weine hinzufügen Fenster



Abbildung 11 – Admin Tool – Weine hinzufügen

In diesem Fenster lassen sich Weine eines bestimmten Winzers, welche noch nicht in der ausgewählten Veranstaltung eingetragen sind, zur aktuellen Veranstaltung hinzufügen.

Im DataGrid werden alle Daten der verfügbaren Weine angezeigt. Mit einer Checkbox können die Weine ausgewählt werden, welche hinzugefügt werden sollen.

Über eine Suchleiste im oberen Teil des Fensters können Weine gesucht werden. Dabei werden alle Attribute des Weines überprüft.

Mit dem „Fertig Button“ werden die ausgewählten Weine zur Veranstaltung hinzugefügt.

3.1.5 Weinsorten Fenster

Im Weinsorten Fenster kann man alle Weinsorten bearbeiten und bei Bedarf neue hinzufügen.

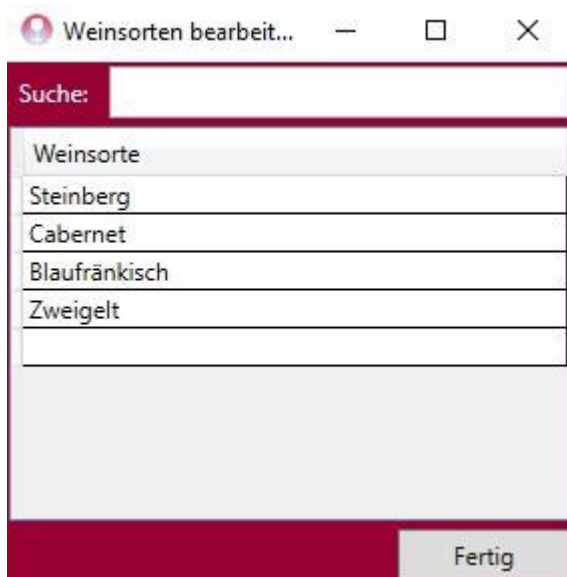


Abbildung 12 – Admin Tool – Weinsorten bearbeiten

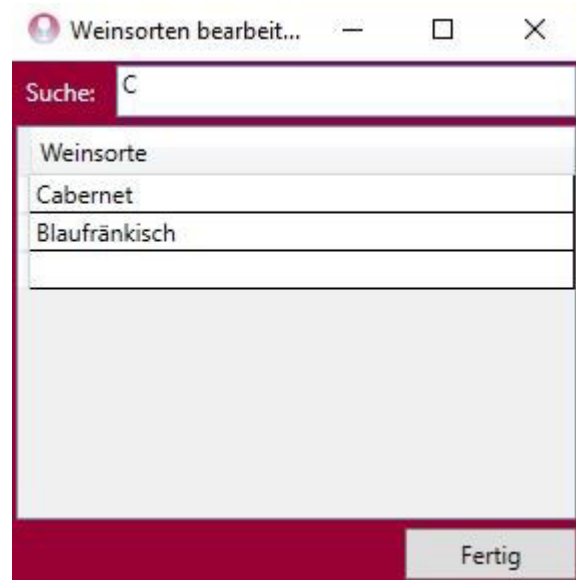


Abbildung 13 – Admin Tool – Weinsorten bearbeiten – mit Suchbegriff

Über eine Suchleiste kann man nach bestimmten Weinsorten suchen. Beim Suchen wird nicht unterschieden, ob die gesuchte Zeichenkette am Anfang oder in der Mitte der Weinsorte steht. Ob die Eingabe in Groß- oder Kleinbuchstaben erfolgt, ist dabei gleichgültig.

Hat man eine Weinsorte bearbeitet, werden die Änderungen automatisch in allen Weinen übernommen, ohne dass die Daten neu geladen werden müssen.

3.2 App

3.2.1 Startseite

Öffnet der Benutzer die Applikation auf seinem Smartphone, so wird zuerst überprüft, ob eine aktive Internetverbindung besteht. Dabei ist nicht von Bedeutung, ob es sich um eine WLAN Verbindung oder um die mobilen Daten handelt.

Besteht keine Internetverbindung, wird der User mit einer Alert-Box darüber informiert, dass die Applikation eine Verbindung benötigt, damit die App die aktuellen Daten aus der Datenbank laden kann. Bestätigt der User diese Meldung, wird die Applikation automatisch wieder geschlossen.

Besitzt der Anwender jedoch eine aktive Internetverbindung, so erscheint die Startseite der Applikation. Diese besteht aus dem Titel der Diplomarbeit, dem Logo, welches beim Start eine kurze Animation ausführt, und einem „Veranstaltung wählen Button“. Klickt der Benutzer auf diesen Button, so navigiert die App zum Veranstaltungsüberblick, wo alle aktiven Veranstaltungen aufgelistet sind.

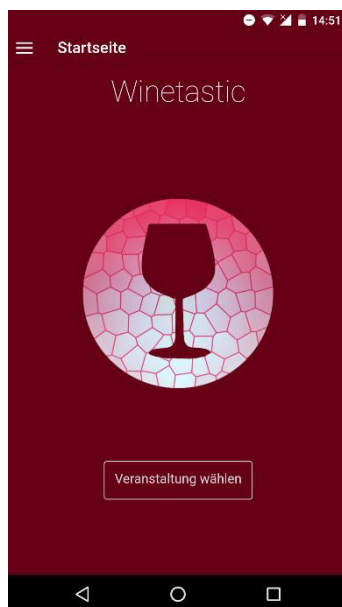


Abbildung 14 - App Startseite

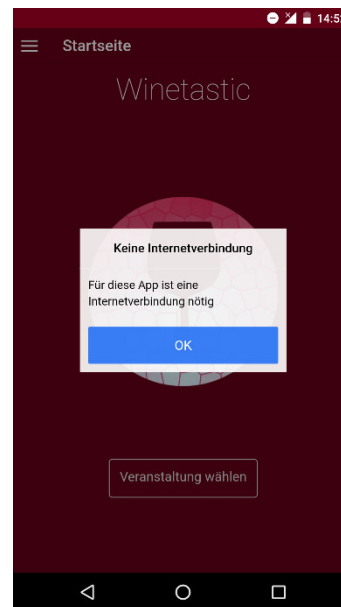


Abbildung 15 - App Startseite ohne Internetverbindung

3.2.2 Veranstaltungsfenster

Hier bekommt der User einen Überblick über alle Veranstaltungen. Dabei werden sowohl die Aktuellste, als auch Vergangene, sofern sich diese noch in der Datenbank befinden, angezeigt. Der Benutzer erfährt auf einen Blick, ob es sich um eine Frühlings-, oder eine Herbstveranstaltung handelt. Unter anderem werden Informationen, wie Datum, Uhrzeit und Ort der Veranstaltung angezeigt. Der Anwender wählt als Nächstes ein Event aus. Dieses wird daraufhin mit einem Hacken an der Seite als markiert dargestellt. Gleich danach wird automatisch die Sidebar der mobilen Applikation ausgeklappt, welche dem User einen groben Überblick über den Aufbau der App gibt.

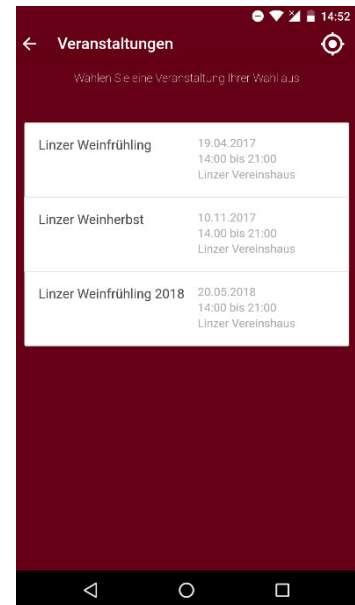


Abbildung 16 - App Veranstaltungsüberblick

3.2.3 Sidebar

Die Applikation unterteilt sich in folgende Bereiche:

- Startseite
- Weine
- Winzer
- Bestenliste
- Veranstaltungen
- Meine Notizen
- Meine Bewertungen
- Einstellungen
- Impressum

Der User kann nun frei entscheiden, welchen Bereich er zuerst besuchen möchte.

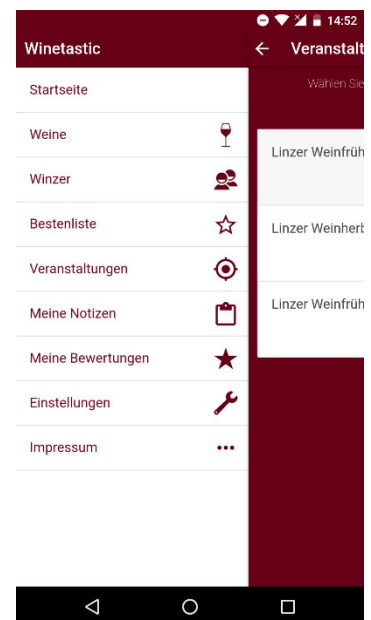


Abbildung 17 - App Sidebar

3.2.4 Wein Fenster

Das Wein Fenster besitzt eine Auflistung aller Weine der zuvor ausgewählten Veranstaltung. Zusätzlich hat der Benutzer die Möglichkeit, nach einem Wein in der verfügbaren Filterleiste zu suchen. Die Filterleiste hat sofort eine Auswirkung auf die darunterliegende Weinliste, wenn ein Buchstabe oder eine Zahl eingetragen wird. Klickt der User nun auf einen Wein, wird er zum Weindetail Fenster weitergeleitet.

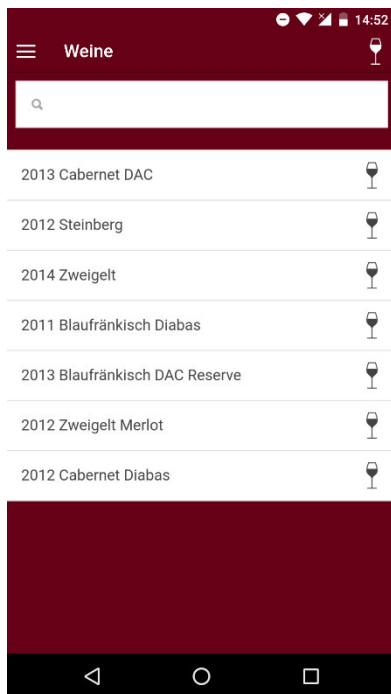


Abbildung 18 - App Weinübersicht

3.2.5 Weindetail Fenster

Das Weindetail Fenster besitzt drei Bereiche, dargestellt als Tabs.

- Allgemein
- Notizen
- Bewerten

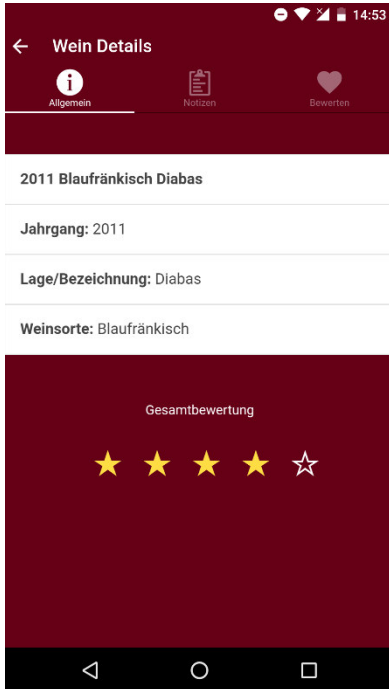


Abbildung 19 - App Weindetails Allgemein

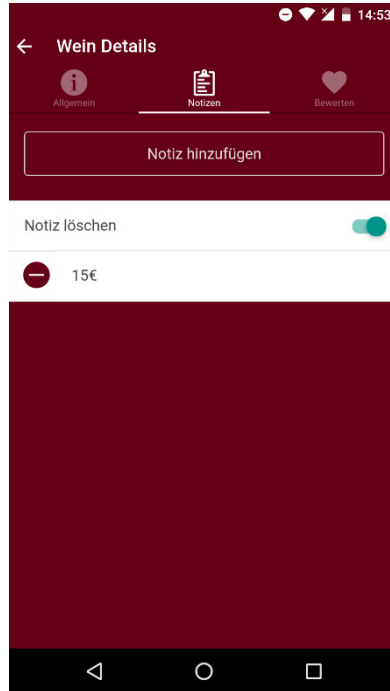


Abbildung 20 - App Weindetails Notizen



Abbildung 21 - App Weindetails Bewertung

3.2.5.1 Allgemein:

Im „Allgemein“ Bereich werden alle wichtigen Informationen über den Wein angezeigt. Die vollständige Informationsliste eines Weines besteht aus:

- Name
- Jahrgang
- Lage/Bezeichnung
- Qualität
- Weinsorte

Besitzt ein Wein beispielsweise keine Bezeichnung, wird das Feld nicht einfach leer gelassen, sondern es wird erst gar nicht dargestellt.

Unter den Informationen wird die Gesamtbewertung des Weines in Form von fünf Sternen abgebildet.

3.2.5.2 Notizen

In diesem Bereich hat der Benutzer die Möglichkeit, mehrere Notizen zum jeweiligen Wein zu verfassen. Klickt der Anwender auf „Notiz hinzufügen“, so erscheint ein kleines Fenster, in dem er seine Nachrichten aufzeichnen und speichern kann.

Es gibt keine Begrenzung, wie viele Notizen ein Wein besitzen kann. Nachdem der User auf den „Speichern Button“ gedrückt hat, erscheint der zuvor eingegebene Text in einer Liste. Neue Notizen werden an diese Liste angehängt. Bleibt das Feld jedoch leer und der Benutzer drückt trotzdem auf „Speichern“, schließt sich das Fenster, ohne der Liste einen neuen Eintrag hinzuzufügen. Mit dem „Schließen Button“ verschwindet das Fenster wieder. Dank dem „Notizen löschen Button“ kann der User die zuvor gespeicherten Notizen wieder löschen. Die Notizen werden lokal gespeichert und sind nur für den Benutzer ersichtlich.

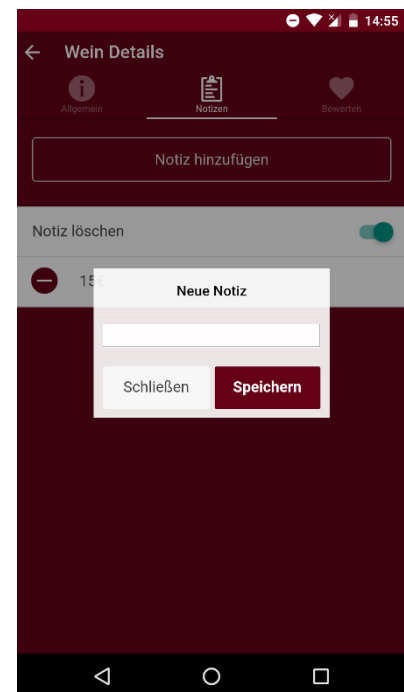


Abbildung 22 - App Weindetails Notizen

3.2.5.3 Bewerten

Wie der Name es schon vermuten lässt, bietet der Bewerten-Tab die Option an, den Wein zu bewerten. Der Benutzer kann dem Wein bis zu fünf Sterne geben. Ein nachträgliches Ändern der Bewertung ist jederzeit möglich. So wie die Notizen, werden auch die Bewertungen lokal gespeichert. Nur die Anzahl der vergebenen Sterne in der Online-Datenbank festgehalten, um die Gesamtbewertung von jedem Wein aus dem Durchschnitt der User-Bewertungen zu errechnen.

3.2.6 Winzer Fenster

Das Winzerfenster ist ähnlich aufgebaut wie das Weinfenster. In einer Liste werden alle Winzer der zuvor ausgewählten Veranstaltung dargestellt. Zusätzlich wird die Tischnummer jedes einzel-

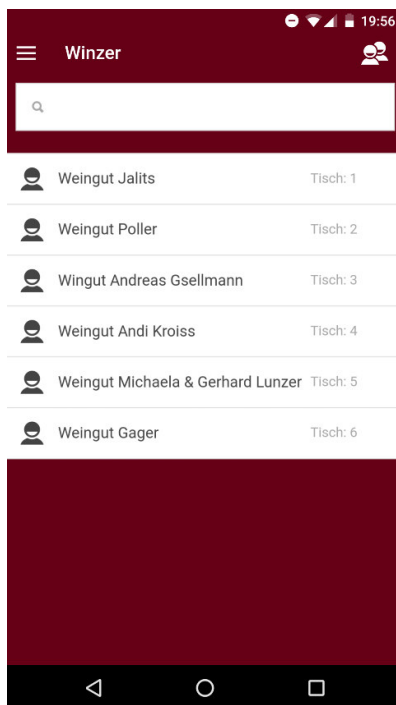


Abbildung 24 - App Winzerübersicht



Abbildung 23 - App Winzerdetails Weine

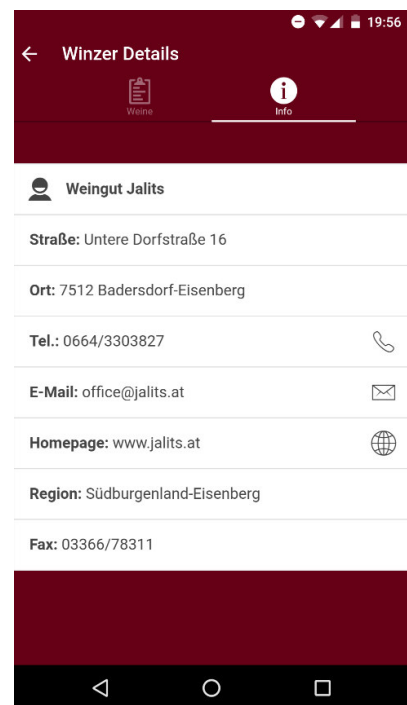


Abbildung 25 - App Winzerdetails Info

nen Winzers nebenbei angezeigt, um ihn auf der Veranstaltung schneller zu finden. Wie im Weinfenster, hat der Anwender auch hier die Möglichkeit nach einem Wein zu suchen. Die Filterleiste reagiert auch dieses Mal direkt nach einem Input. Wählt der Benutzer nun einen Winzer aus, so gelangt er zum Winzerdetails Fenster.

3.2.7 Winzerdetails Fenster

Das Winzerdetails Fenster ist in folgende zwei Bereiche eingeteilt:

- Weine
- Allgemein

Im „Weine“ Bereich werden alle Weine des Winzers bei dieser Veranstaltung aufgelistet. Diese Weine sind ebenfalls wieder auswählbar und leiten den Anwender zum Weindetails Fenster des jeweiligen Weines.

Im zweiten Bereich *Allgemein* werden alle allgemeinen Informationen über den Winzer preisgegeben. Eine vollständige Liste eines Winzers beinhaltet:

- Name
- Straße
- Ort
- Telefonnummer
- E-Mail
- Homepage
- Region
- Fax

Auch in dieser Liste werden nicht angegebene Attribute, nicht einfach leer gelassen, sondern erst gar nicht angezeigt.

Die Felder „Telefonnummer“, „E-Mail“ und „Homepage“ sind noch einmal gesondert mittels eines Icons rechts davon gekennzeichnet.

Wählt der User das Feld „Telefonnummer“ aus, so leitet ihn die App zur Stock-Telefon-Applikation, wo der Benutzer die Option hat, den Winzer anzurufen oder seine Nummer zu speichern.

Klickt er jedoch auf das „E-Mail“ Feld, öffnet sich die Stock-E-Mail-Applikation. Die E-Mail-Adresse des Winzers wird direkt in das Feld „An“ eingetragen und so muss der Benutzer nur mehr den Text verfassen.

Zu guter Letzt hat der Anwender auch die Möglichkeit, direkt zur Homepage des ausgewählten Winzers zu navigieren, falls er diese angegeben hat.

3.2.8 Bestenliste

In der Bestenliste werden alle bewerteten Weine der Veranstaltung aufgelistet. Neben dem Namen des Weines wird eine Punkteanzahl angezeigt. Je höher diese Punkteanzahl, desto weiter oben ist der Wein in der Liste zu finden. Wie sich der Punktestand ergibt, wird im Abschnitt „Bewertungssystem“ näher erklärt.

3.2.9 Meine Notizen

Wird zu einem Wein eine Notiz verfasst, befindet sich der Wein anschließend in diesem Bereich. Neben dem Namen wird die letzte Notiz des Weines angezeigt.

Auf dem Smartphone wird, nachdem der User dem Wein eine Notiz hinzugefügt hat, eine Datei erstellt. In dieser Datei befinden sich alle WineIDs mit der dazugehörigen Notiz. Der Name der Datei setzt sich aus der EventID und dem Wort „Notiz“ zusammen. So können die Notizen zu verschiedenen Veranstaltungen unterschieden werden.

Viele Benutzer notieren sich auf dem ausgedruckten Katalog zum Beispiel den Preis des Weines. Die Applikation bietet den Vorteil, dass man die Notizen von verschiedenen Veranstaltungen vergleichen kann, um so beispielsweise zu sehen, ob sich der Preis eines Weines verändert hat, wenn dieser wieder auf der Veranstaltung präsentiert wird.

Ein weiterer Vorteil dabei ist, dass die erstellte Notiz nicht mittels Explorer auffindbar ist. Die Datei wird in einem privaten Bereich auf dem Smartphone gespeichert. So wird verhindert, dass Notizen manipuliert werden können.



Abbildung 26 - App Meine Notizen

3.2.10 Meine Bewertungen

Ähnlich wird es im „Meine Bewertungen Fenster“ gehandhabt. Nachdem der Anwender einen Wein bewertet hat, wird eine Datei erstellt, wobei auch hier wieder, dank der EventID, zwischen verschiedenen Veranstaltungen unterschieden werden kann. Auch diese Datei ist mittels Explorer

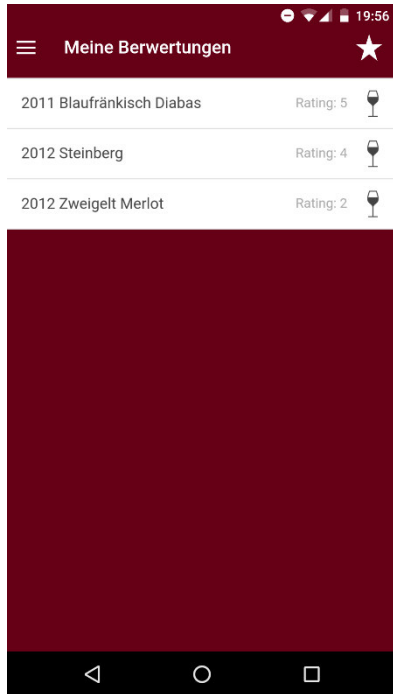


Abbildung 27 - App Meine Bewertungen

nicht auffindbar. Die bewerteten Weine werden in diesem Bereich mit den vergebenen Sternen aufgelistet.

3.2.11 Einstellungen

Dank unseres Bewertungssystems, kann sich die Bestenliste während der Veranstaltung stark ändern. Um den Datenverkehr jedoch gering zu halten, verzichteten wir auf eine Funktion die alle paar Minuten oder Sekunden die Daten neu von der Datenbank herunterlädt. Im „Einstellungen“ Bereich klären wir den Nutzer darüber auf. Zusätzlich gibt es einen Button, mit dem der User die Daten manuell aus der Datenbank laden kann.

Wählt der Benutzer diesen Button aus, so werden auch alle Daten, nicht nur die Bestenliste, aktualisiert.

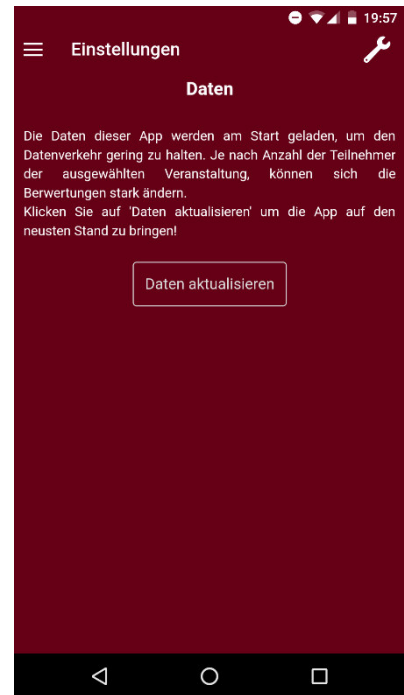


Abbildung 28 - App Einstellungen

3.2.12 Impressum

Im Impressum befinden sich die Kontaktdaten der Veranstalter.

3.2.13 Bewertungssystem

Wird ein Wein von einem Benutzer bewertet, so wird die Anzahl der vergebenen Sterne an die Datenbank gesendet. Dort werden alle vergebenen Sterne durch die Anzahl, wie oft der Wein bereits bewertet wurde, dividiert. In der App wird die errechnete Gesamtbewertung bei jedem Wein angezeigt. Das Bewertungssystem arbeitet mit ganzen Sternen, so wird ab einem Wert von 0.5 aufgerundet. Hat ein Benutzer bereits einen Wein bewertet und bewertet diesen erneut, so wird nur die Anzahl der Sterne in der Datenbank korrigiert, nicht jedoch die Anzahl, wie oft der Wein bewertet wurde.

4 Grundlagen

4.1 Anfängliche Überlegungen

4.1.1 Datenbank

Zu Beginn der Diplomarbeit war geplant, dass wir den Datenaustausch zwischen der Mobilien Applikation, beziehungsweise dem Admin Tool und der Datenbank über einen WCF Service abwickeln. Zur Erklärung, ein WCF Service (Windows Communication Foundation Service) ist ein Service, welcher für eine Anwendung beziehungsweise eine Applikation als Schnittstelle zu einer Datenbank fungiert. Dabei muss der Client nicht direkt auf die Datenbank zugreifen, stattdessen greift die Anwendung auf den WCF Service zu, um Daten zu konsumieren.

Jedoch haben wir uns schließlich doch für eine Azure Mobile Service Datenbank entschieden. Wir trafen diese Entscheidung, da die Mobile Service Datenbank speziell für die Nutzung im Zusammenhang mit mobilen Applikationen gedacht ist. Weiters war die Datenbank sehr einfach einzurichten und zu verwalten.

4.1.2 App Entwicklung

Eigentlich war von Beginn an geplant, dass wir zur Entwicklung der mobilen Applikation Microsoft Xamarin verwenden, da uns Xamarin ermöglicht eine Applikation zu schreiben, welche auf allen Plattformen funktioniert. Da Xamarin noch sehr neu und noch in den Kinderschuhen steckt, mussten wir uns entscheiden, ob wir das Risiko eingehen ein Framework zu verwenden, welches noch nicht ganz fehlerfrei ist.

Xamarin bietet einige Vorteile, zum Beispiel, dass Anwendungen das native User Interface des jeweiligen Betriebssystems verwenden, welches je nach OS variiert. So hat es den Anschein, als wurde die Applikation nativ für das System entwickelt. Die Geschwindigkeit einer mit Xamarin entwickelten Applikation kann man auch mit einer nativ geschriebenen App vergleichen. Weiters sind alle .NET Funktionen in einer Xamarin App verfügbar.

Unsere Entscheidung fiel jedoch auf Apache Cordova, früher als PhoneGap bekannt. Einerseits, weil Cordova schon als Multi-Platform Framework etabliert ist. So mussten wir nicht mehr mit etwaigen Fehlern und Kompatibilitätsproblemen rechnen. Andererseits, weil Cordova HTML, CSS und JavaScript verwendet. Dadurch fiel uns die Entwicklungsarbeit um vieles leichter, da wir aufgrund unserer Ausbildung an der HTL mit dieser Technologie in den letzten Jahren schon viel Erfahrung sammeln konnten. Apache Cordova unterstützt zusätzlich alle gängigen HTML Frameworks. Weiters ist es mit Apache Cordova möglich, ein und dieselbe Applikation auf verschiedenen Betriebssystemen auszuführen. Ein kleiner Nachteil besteht jedoch, denn die mobile Applikation unterstützt nicht das native User Interface eines Betriebssystems, sondern bringt sein eigenes, das je nach OS unterschiedlich ist, mit. Auch die Performance ist nicht mit einer nativen Applikation zu vergleichen. Dank der Framework Unterstützung von Cordova, verwenden wir das Framework Ionic, welches das Design der App, an das Design des Betriebssystems angleicht. Des Weiteren werden die Smartphones immer schneller, somit verringert sich der Performanceunterschied.

4.2 Aufbau und Kommunikation

4.2.1 Allgemein

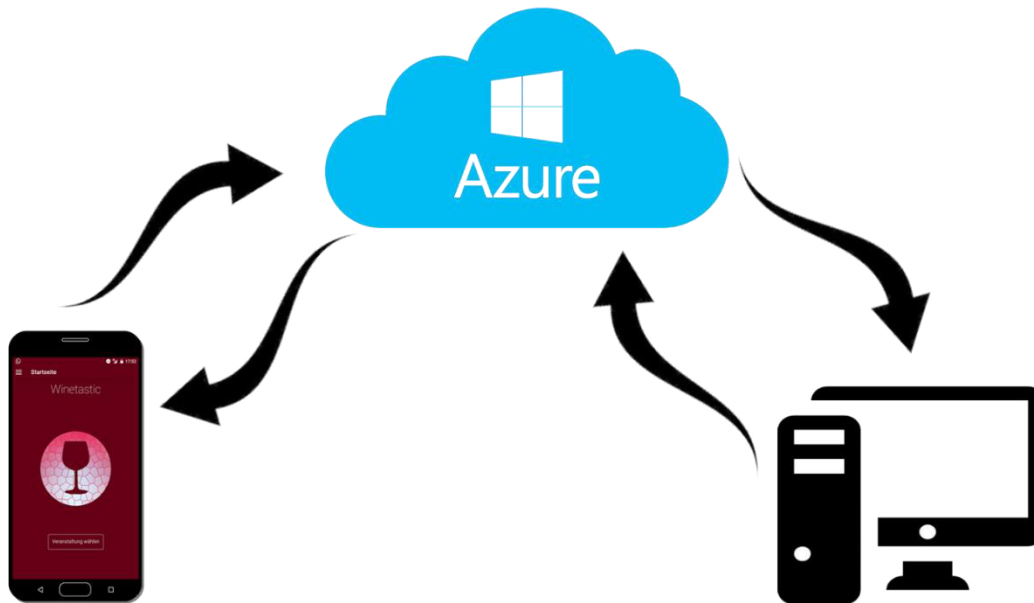


Abbildung 29 - Aufbau Allgemein

Im Zentrum der beiden Endgeräte steht die Easy Tables Datenbank, welche auf Azure gehostet ist. Sowohl die Applikation als auch das Admin Tool auf dem PC greifen darauf zu und holen sich die Daten davon.

4.2.2 Veranstalteransicht

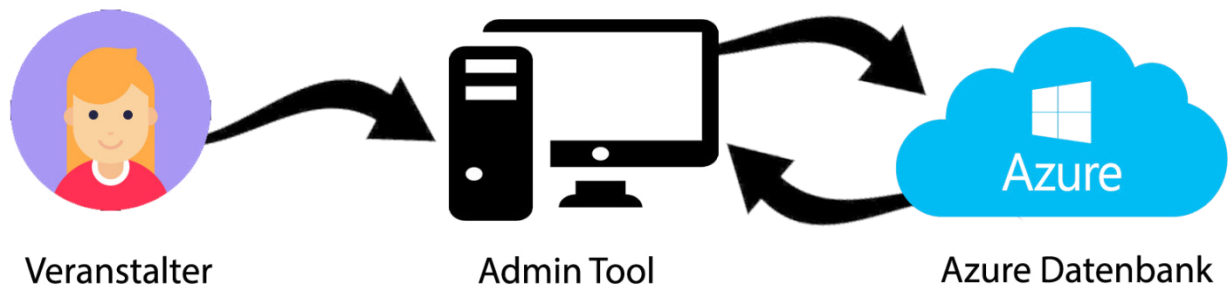


Abbildung 30 - Kommunikation Veranstalteransicht

Der Veranstalter öffnet das Admin Tool auf seinem PC, um beispielsweise eine neue Veranstaltung mit all den Daten zu erstellen. Das Admin Tool holt sich beim Start alle verfügbaren Daten aus der Datenbank und bringt diese Daten in ein übersichtliches Format. Erstellt der Veranstalter nun eine neue Veranstaltung, so wird diese in der Datenbank gespeichert.

4.2.3 Anwenderansicht



Abbildung 31 - Kommunikation Anwenderansicht

Der Anwender lädt die App aus dem Play Store herunter und startet sie auf seinem Smartphone. Beim Start werden alle Daten aus der Datenbank in die Applikation geladen und in verschiedenen Bereichen präsentiert. Bewertet der Anwender einen Wein, so wird die Bewertung an die Datenbank gesendet.

4.3 Benötigte Ressourcen

4.3.1 Microsoft Visual Studio

Das wichtigste Tool für unsere Diplomarbeit war die Entwicklungsumgebung Visual Studio. Mittels Visual Studio wurde sowohl das Admin Tool als auch die mobile Applikation entwickelt.

4.3.2 Android Smartphone

Um die mobile Applikation auch ausführlich zu testen, wurde ein Android Smartphone genutzt. Als Testgrundlage diente das Nexus 5X mit der aktuellsten Android Version 7.1.1. Nexus Geräte eignen sich ausgezeichnet zum Testen einer neuen Applikation. Sie werden in Kooperation mit Google hergestellt und besitzen die neueste Android Version, ohne jegliche Softwareänderung, wie es bei anderen Herstellern oft der Fall ist.

4.3.3 SmartGit

Ein sehr wichtiges Tool bei der Entwicklung der Diplomarbeit war SmartGit. SmartGit ist ein verteiltes Versionierungssystem. Es ist vor allem dann sehr hilfreich, wenn mehrere Teammitglieder an einem Projekt gleichzeitig arbeiten.

4.4 Verwendete Technologien

Folgende Technologien wurden für die Entwicklung der Diplomarbeit verwendet:

4.4.1 Mobile Applikation

4.4.1.1 HTML

Hypertext Markup Language, oder auch besser bekannt unter der Abkürzung HTML, ist eine XML-Sprache, die es ermöglicht Texte, Bilder bis hin zu Videos darzustellen. Die textbasierte Auszeichnungssprache ist die Grundlage des World Wide Webs. (selfhtml, 2015)

HTML verwendet sogenannte „Tags“, die einerseits den Aufbau des Dokuments beschreiben und andererseits Inhalte einbinden.

Ein HTML Dokument kann jeder beliebige Browser lesen und den damit verbundenen Inhalt anzeigen. Somit kann jedes Gerät, welches einen Browser besitzt, HTML Dokumente parsen. (Pasqua, 2011)

Apache Cordova, der Grundstein für die mobile Applikation, verwendet HTML, um das grafische Grundgerüst der App zu realisieren.

Die derzeit aktuellste Version, die auch für die Diplomarbeit verwendet wurde, ist HTML 5.



Abbildung 32 - HTML Logo

4.4.1.2 CSS

Cascading Style Sheets sind eine Erweiterung von HTML. Mittels CSS-Attributen ist es möglich, Webseiten individuell zu gestalten, beispielsweise Farbcodes zu hinterlegen, Form und Farbe von Items zu ändern und vieles mehr. Diese Attribute können im HTML Dokument direkt eingefügt oder in eigenständigen Dateien definiert und im HTML Dokumente eingebunden werden.

(Astrup, kein Datum)

Dies hat den Vorteil, dass bei größeren Webseiten oder auch Webprojekten, die Ordnerstruktur übersichtlicher wird und der HTML Code besser gelesen werden kann.

Definiert man zuvor ein CSS-Attribut mit eindeutigem Namen, so ist es möglich, gewisse Abschnitte einer Webseite mit dem vorher definierten Attribut „class = Name“ auszustatten.

Große Vorteile durch CSS ergeben sich durch den zuvor erwähnten besseren Strukturüberblick, sowohl im Code selber, als auch in der Ordnerstruktur. Des Weiteren können Attribute immer wieder verwendet werden und ein weiterer großer Vorteil ist, dass HTML Seiten dadurch schneller laden. (Apsel, kein Datum)

CSS wurde in unserer App dazu verwendet, um die verschiedenen Grafikelemente der mobilen Applikation in eine einheitliche Farbe zu bringen.



Abbildung 33 - Cascading Style Sheet Logo

4.4.1.3 JavaScript

JavaScript ist eine Script-Sprache, welche HTML um dynamische Funktionen bereichert. Die Sprache ermöglicht es statische Webseiten dynamisch zu gestalten. Mittels JS können Benutzerinteraktionen ausgeführt und Funktionen implementieren werden. So ist es beispielsweise möglich, „Benutzerinteraktionen auszuwerten, Inhalte zu verändern, nachzuladen oder zu generieren und so die Möglichkeiten von HTML und CSS erweitert.“ (Koch, 2009)



Abbildung 34 - JavaScript Logo

JavaScript wurde dazu verwendet, um das Bewertungssystem der Applikation zu realisieren.

4.4.1.4 Ionic

Ionic ist ein Framework für Hybrid-Apps und basiert auf HTML und CSS. Es wurde von dem Unternehmen „Drifty“ entwickelt und legt seinen Fokus auf das Designen von User Interfaces. Das Framework wird oft mit AngularJS verwendet und besitzt mehrere grafische Elemente, welche sich sehr gut für Touch Systeme eignen. Je nachdem, auf welchem System die Applikation ausgeführt wird, ändert sich auch das Design der Applikation. So sieht ein und dieselbe App auf zwei verschiedenen Systemen anders aus, obwohl diese den gleichen Ionic Code beinhaltet. Dies soll das „Look and Feel“ des Smartphones beibehalten, damit die App nicht aus der Masse sticht. (Kluge, kein Datum)



Abbildung 35 - Ionic Logo

Das Framework wird oft mit AngularJS verwendet und

besitzt mehrere grafische Elemente, welche sich sehr gut für Touch Systeme eignen. Je nachdem, auf welchem System die Applikation ausgeführt wird, ändert sich auch das Design der Applikation. So sieht ein und dieselbe App auf zwei verschiedenen Systemen anders aus, obwohl diese den gleichen Ionic Code beinhaltet. Dies soll das „Look and Feel“ des Smartphones beibehalten, damit die App nicht aus der Masse sticht. (Kluge, kein Datum)

Hier ein Beispiel, wie eine App, die mit Ionic design wurde, auf verschiedenen Betriebssystemen aussieht.

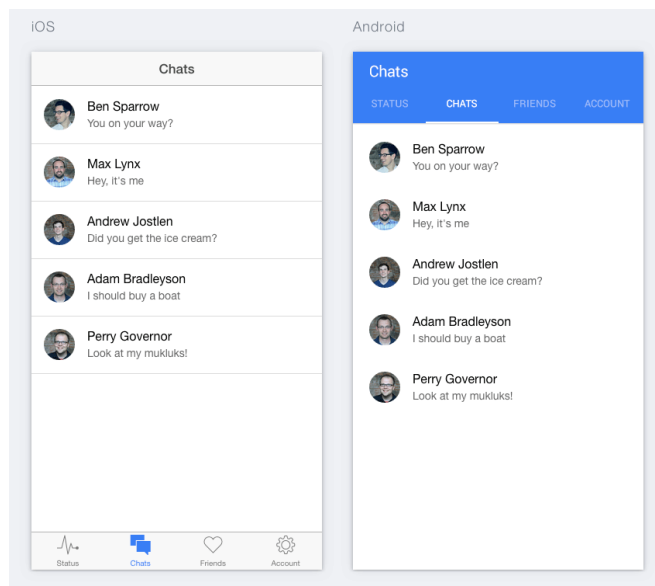


Abbildung 36 - Ionic Android und IOS

Unsere Applikation wurde mit Ionic 1 design. Während der Entwicklung wurde Ionic 2 veröffentlicht.

Das Framework wurde in der mobilen App genutzt um Smartphone relevante Elemente, wie die Sidebar oder On/Off Buttons darzustellen.

4.4.1.5 JQuery

JQuery ist ein externes Open-Source-Framework, welches JavaScript um eine Vielzahl von Bibliotheken erweitert. JQuery besitzt eine umfassende Klassenbibliothek, durch die der Zugriff auf das Document Object Model (DOM) realisiert wird.

„So müssen viele Funktionen nicht immer neu programmiert werden, sondern können einfach und komfortabel in den Quellcode eingebunden werden. Die Bibliothek besteht dabei aus einer JavaScript-Datei, die die grundlegenden Ajax-Funktionen und gängigen Effekte enthält.“ (Waschier, 2011)

„JQuery unterstützt eine sogenannte "Selektor" API (Application Programming Interface), die die Abfrage von HTML-Elementen mit Hilfe von Selektoren erlaubt, um dann entsprechende Befehle auf diese anzuwenden.“ (Lipinski, kein Datum)

Zusätzlich erlaubt es JQuery seinen Funktionsumfang mittels Plug-Ins zu erweitern.

JQuery wurde verwendet, um das Bewertungssystem, beziehungsweise das Anzeigen der Sterne, zu realisieren.



Abbildung 37 - JQuery Logo

4.4.1.6 AngularJS

AngularJS oder auch Angular genannt, ist ein JavaScript Framework, welches von Google in Verbindung mit der Online Community entwickelt wird. Angular baut auf dem Model-View-View Model Konzept auf und erweitert HTML um mehrere Attribute beziehungsweise Tags.

Das Framework wird in Kombination mit Apache Cordova verwendet, welches uns ermöglicht Hybrid-Apps zu entwickeln. (https://www.w3schools.com/angular/angular_intro.asp, kein Datum)



Abbildung 38 - AngularJS Logo

AngularJS erweitert HTML um

- Two-Way Binding
 - Beim Two-Way Binding wird eine Verknüpfung zwischen der repräsentierenden Schicht, der View, und der Business Logik, dem Modell, hergestellt. Wird beispielsweise ein Element aus einer List entfernt, welche im Programm dargestellt wird, verschwindet es nicht nur im Modell, sondern auch direkt grafisch auf der repräsentierenden Schicht.
- Direktiven
 - Sie sind spezielle Funktionen von AngularJS und fangen stets mit „ng“ an. Somit ist es möglich, gewisse Bereiche in der App auszublenden und erst ab einem gewissen Zeitpunkt wieder sichtbar zu machen. Direktiven kann man als eigene kleine Methoden interpretieren.
- NG-Repeat Schleifen
 - NG-Repeat Schleifen kann man mit „For-Schleifen“ vergleichen. Die Schleife geht eine Variable gefüllt mit JSON Daten durch. Dank dem JSON Format braucht die Schleife nicht wissen wie viele Daten vorhanden sind. Durch die Schleife lassen sich beispielsweise Listen aus Daten generieren.

- Filter
 - Mittels Filtern kann man aus einer Menge von Daten, gewisse Daten herausfiltern. Filter werden oft in Verbindung mit Suchleisten genutzt oder als Joins von verschiedenen Datenmengen
- Module
 - Ein Modul kann man mit Packages in Java vergleichen. Sie besitzen eine Ansammlung von Controllern, Services, Filtern und Direktiven. In einer Anwendung können mehrere Module erstellt werden.
- Scope
 - Ein Scope ist ein eigener Bereich in AngularJS. So gibt es bei der App Entwicklung für jede Seite einen eigenen Scope, in dem man Variablen und Funktionen deklarieren kann.
- RootScope
 - Ein RootScope ist ein Bereich für die ganze Applikation. Dort werden globale Variablen gespeichert.
- Controller
 - Ein Controller definiert einen Scope, somit hat jede Seite einen eigenen Controller. In diesem Controller wird die Funktionalität der Seite definiert.
- Expressions
 - Variablen, die im Controller deklariert werden, werden auf der grafischen Seite der Anwendung mit `{{variablen Name}}` sichtbar gemacht. Die geschwungenen Klammern werden Expressions genannt.
- Services
 - Viele wiederkehrende Aufgaben, wie zum Beispiel das Empfangen und Verarbeiten von JSON Daten, werden in sogenannte Services verpackt. AngularJS stellt mehrere Services zur Verfügung, falls nötig kann man auch eigene entwickeln. (Weiße, kein Datum)

Für Apache Cordova ist AngularJS notwendig, um der Applikation eine Hintergrundlogik zu geben. Die derzeit aktuelle Version, welche auch für die Diplomarbeit verwendet wird, ist 2.2.4.

4.4.1.7 Apache Cordova

Apache Cordova ist ein Framework zur Erstellung von mobilen Hybrid-Applikationen. Hybrid-Applikationen sind weder native noch webbasierte Applikationen, sondern eine Kombination daraus. Apps, die mit Apache Cordova entwickelt werden, sind plattformunabhängig und können so sowohl auf Android Systemen als auch auf IOS Systemen ausgeführt werden. (Michal Gralak)

Die Benutzeroberfläche wird mit HTML, CSS und etwaigen anderen Frameworks erstellt. Apache Cordova inkludiert des Weiteren eine Ansammlung an Geräte APIs. So haben die Applikationen Zugriff auf native Funktionen eines Gerätes, wie die Kamera, den Speicher oder den Fingerabdrucksensor, jedoch nicht auf alle Geräte APIs. (Foundation, 2015)

Der eigentliche Source Code wird mit JavaScript realisiert. Die entwickelte Applikation wird als vollwertige Applikation angesehen und kann in den diversen Applikation-Stores (Google Play Store, IOS App Store) zur Verfügung gestellt werden.

Die aktuelle Version, die wir auch zur Entwicklung verwendet haben, ist 6.1.



Abbildung 39 - Apache Cordova Logo

Hier eine Übersicht auf welchen Betriebssystemen man eine App, die mit Apache Cordova entwickelt wurde, ausführen kann und welche API Funktionen unterstützt werden.

	iPhone / iPhone 3G	iPhone 3GS and newer	Android	Blackberry OS 6.0+	Blackberry 10	Windows Phone 8	Ubuntu	Firefox OS
Accelerometer	✓	✓	✓	✓	✓	✓	✓	✓
Camera	✓	✓	✓	✓	✓	✓	✓	✓
Compass	X	✓	✓	X	✓	✓	✓	✓
Contacts	✓	✓	✓	✓	✓	✓	✓	✓
File	✓	✓	✓	✓	✓	✓	✓	X
Geolocation	✓	✓	✓	✓	✓	✓	✓	✓
Media	✓	✓	✓	X	✓	✓	✓	X
Network	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Alert)	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Sound)	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Vibration)	✓	✓	✓	✓	✓	✓	✓	✓
Storage	✓	✓	✓	✓	✓	✓	✓	✓

✓ - supported feature
 X - unsupported feature due to hardware or software restrictions

Abbildung 40 - Apache Cordova Funktionsumfang

4.4.2 Admin Tool

4.4.2.1 Windows Presentation Foundation Framework

Das Windows Presentation Foundation Framework (WPF) ist ein Grafik-Framework zum Erstellen von Windows Desktop Applikationen in einem einheitlichen Programmiermodell. Dabei werden die Geschäftslogik und das Benutzerinterface strikt getrennt. Dies wird dadurch unterstützt, dass die Benutzeroberfläche in XAML, einer auf XML basierenden Auszeichnungssprache, deklariert wird. Die Logik, wie die Oberfläche auf bestimmte Ereignisse reagiert, ist im Code Behind festgelegt. Durch die in WPF verfügbare Datenbindung können Objekte an Oberflächenelemente gebunden werden.

Das Windows Presentation Foundation Framework ist seit der .NET Version 3 verfügbar.

4.4.3 Datenbank

4.4.3.1 Azure Easy Tables

Die Easy Tables Datenbank funktioniert ähnlich wie eine SQL Datenbank. Sie bietet einige wesentliche Vorteile gegenüber SQL Datenbanken, bringt jedoch auch Nachteile mit sich.



Abbildung 41 – Microsoft Azure Logo

Ein großer Vorteil ist der, dass man von Mobilgeräten und auch Desktopanwendungen sehr einfach darauf zugreifen kann. Weiters ist es auch sehr einfach in einer Desktopanwendung auf die Daten zuzugreifen. Zudem bieten Easy Tables die Möglichkeit, den Zugriff auf bestimmte Funktionen, wie z.B., das Einfügen von Daten, nur für bestimmte Benutzer zu erlauben. Ein Nachteil von Azure Easy Tables ist, dass es keine Möglichkeit gibt Verbindungen, in Form von Fremdschlüsseln, zwischen den Tabellen herzustellen. Alle Verbindungen zwischen den Tabellen müssen somit in der Programmlogik beachtet werden.

4.5 Datenbankmodell

Unser Datenbankmodell besteht aus 7 Tabellen:

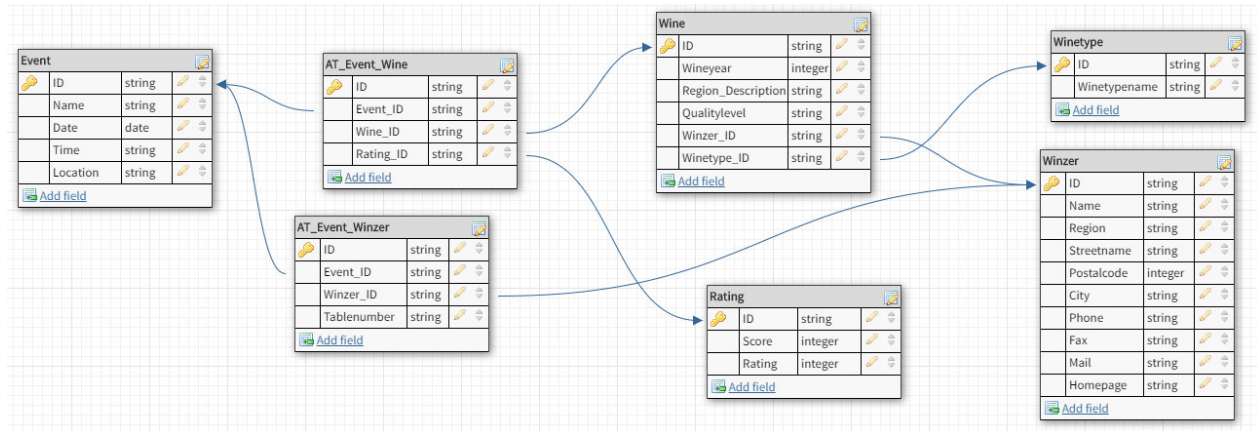


Abbildung 42 - Datenmodell

Tabelle	PK	Typ	Attribute	Typ	Bedeutung
Event	ID	String	Name	String	Name der Veranstaltung
			Date	Date	Datum der Veranstaltung
			Time	String	Uhrzeit der Veranstaltung
			Location	String	Veranstaltungsort
AT_Event_Wine	ID	String	Event_ID	String	Referenziert eine Veranstaltung, auf welcher ein bestimmter Wein präsent ist
			Wine_ID	String	Referenziert einen Wein, welcher auf einer bestimmten Veranstaltung präsentiert wird
			Rating_ID	String	Referenziert ein Rating für einen Wein auf einer bestimmten Veranstaltung

Tabelle	PK	Typ	Attribute	Typ	Bedeutung
AT_Event_Winzer	ID	String	Event_ID	String	Referenziert eine Veranstaltung, auf welcher ein bestimmter Winzer präsent ist
			Winzer_ID	String	Referenziert einen Winzer, welcher auf einer bestimmten Veranstaltung präsentiert wird
			Tablenumber	String	Gibt die Tischnummer an, welche der Winzer auf der Veranstaltung hat
Wine	ID	String	Wineyear	Number	Gibt den Jahrgang des Weines an
			Region_Description	String	Gibt die Anbauregion bzw. die Bezeichnung an
			Qualitylevel	String	Gibt die Qualitätsstufe des Weines an
			Winzer_ID	String	Referenziert den Winzer des Weines
			Winetype_ID	String	Referenziert die Weinsorte des Weines

Tabelle	PK	Typ	Attribute	Typ	Bedeutung
Winzer	ID	String	Name	String	Gibt den Namen des Weinguts an
			Region	String	Gibt die Anbauregion des Winzers an
			Streetname	String	Gibt die Straße des Winzers an
			Postalcode	Number	Gibt die Postleitzahl des Winzers an
			City	String	Gibt den Ort des Winzers an
			Phone	String	Gibt die Telefonnummer des Winzers an
			Fax	String	Gibt die Faxadresse des Winzers an
			Mail	String	Gibt die E-Mail Adresse des Winzers an
			Homepage	String	Gibt die Homepage des Winzers an

4.6 Native Apps vs. Hybrid-Apps

Ein großer Unterschied zwischen nativen und hybriden Applikationen ist, dass eine native App nur auf einem System installiert und ausgeführt werden kann und eine hybride Applikation auf mehreren Systemen. Derzeit befinden sich viele verschiedene Geräte mit unterschiedlichen Betriebssystemen auf dem Markt. Möchte man nun eine native Applikation schreiben, muss man sich dabei Gedanken machen, welche Betriebssysteme diese unterstützen soll. Bei einer hybriden App schreibt der Entwickler den Code für seine Applikation nur einmal und die App funktioniert auf verschiedenen Geräten. Dies hat den Vorteil, dass die Entwicklungszeit der Applikation sich um einiges verringert und die Wartung der Applikation um vieles leichter wird.

Im nächsten Abschnitt wird erklärt, warum wir uns für eine Hybrid-Applikation entschieden haben und welche Vor- und Nachteile sowohl native als auch hybride Apps besitzen.

4.6.1 Native Apps

Der Hauptgrund, warum noch viele Entwickler eine native Applikation entwickeln, ist wohl die Geschwindigkeit und Leistungsfähigkeit. Native Applikationen sind viel besser auf das Betriebssystem abgestimmt und besitzen somit bessere Responsezeiten. Darüber hinaus hat man vollen Zugriff auf alle Funktionen des Betriebssystems. Auch der Zugriff auf diese Funktionen ist um einiges einfacher als in Hybrid-Apps. (Korf, 2016)

4.6.1.1 Vorteile

- Leistung

Das Starten und Beenden der App, Animationen und Ladezeiten sind bei nativen Apps um vieles schneller und flüssiger, vor allem auf leistungsschwächeren Smartphones. Dies hat mit dem besseren Zugang zur Hardware zutun. Neue Smartphones bieten jedoch auch bei Hybrid-Apps schon ähnlich gute Performance und man wird, bis auf bei aufwendigen Animationen, kaum Leistungsunterschiede merken. Dies ist auch für unsere mobile Applikation wichtig, um bei den Endbenutzern einen positiven Eindruck zu hinterlassen.

- Material Design/User Interface

Native Applikationen greifen auf das interne User Interface des jeweiligen Betriebssystems zurück. Bei Android wird dieses ab Version 5.0 „Material Design“ genannt. Somit ist die App mit dem Betriebssystem designtechnisch abgestimmt und der Benutzer kennt sich beim ersten Start einer Applikation gleich aus, da wiederkehrende Designelemente benutzt werden. Bei Hybrid-Apps ist dies nicht so einfach, da die Applikation nur aus einem Code für alle Plattformen besteht, die verschiedenen Betriebssysteme jedoch auf unterschiedliche User Interfaces setzen. Abhilfe schaffen hierbei diverse Frameworks, die das Aussehen der App je nach System ändern, um dem nativen User Interface so nah wie möglich zu kommen.

- APIs

Die vielfältige Hardwareunterstützung eines Smartphones, wie die Kamera, verschiedene Sensoren oder Fingerabdruckscanner, bieten einem Entwickler von nativen Apps eine große Auswahl an Features, die seine zukünftige Applikation beinhalten könnte. Durch die bereitgestellten APIs des Betriebssystems, wird die Kommunikation mit den Hardwarekomponenten erleichtert. Für unsere App sind die verschiedenen Hardwarekomponenten nicht wichtig, somit ist dieser positive Aspekt einer nativen Applikation nicht relevant für uns.

- Vulkan/Grafik API

Möchte man eine Applikation mit vielen Grafikelementen schreiben, beispielsweise ein Spiel, oder ein Fotobearbeitungsprogramm, besitzen native Applikationen eine Grafikschnittstelle. Bei Android Applikationen wird sie Vulkan API genannt. Durch diese API wird die gesamte Grafikleistung eines Smartphones ausgeschöpft. Weiters soll die API die Entwicklung solcher Apps vereinfachen. Unsere App ist jedoch nicht grafikintensiv und somit stellt dies keinen Grund für die Entwicklung einer nativen Applikation dar.

4.6.1.2 Nachteile

- Entwicklungszeit

Möchte man seine Applikation jedoch für mehrere Systeme bereitstellen, so muss die Applikation doppelt oder dreifach entwickelt werden. Außerdem muss man auch die unterschiedlichen Programmiersprachen beherrschen. Somit erhöht sich die Entwicklungszeit um ein Vielfaches.

- Wartung

Wenn man nun die Applikation für mehrere Systeme entwickelt hat, bedeutet dies auch einen viel höheren Wartungsaufwand. Schwierig ist es auch, wenn man neue Features in die Applikation einfügen oder den kompletten Aufbau einer App nach gewisser Zeit wieder ändern möchte.

4.6.2 Hybrid-Apps

Hybride Apps haben den großen Vorteil, dass sich die Entwicklungszeit stark verringert, wenn die Applikation für mehrere Systeme entwickelt wird. Somit sind Hybrid-Apps gut geeignet, wenn man beispielsweise unter Zeitdruck steht. Auch nach der Entwicklung ist der Wartungsarbeit um einiges geringer.

4.6.2.1 Vorteile

- Implementierung

Der Code der Applikation wird mit HTML, JavaScript, CSS und anderen HTML-Frameworks realisiert. Somit muss man sich als Entwickler nur mit diesen Programmiersprachen auseinandersetzen und nicht mehrere native Sprachen beherrschen.

- Entwicklungszeit

Da die Applikation nur in einer Programmiersprache realisiert wird, welche auf mehreren Betriebssystemen ausgeführt werden kann, reduziert sich der Zeitaufwand für die Entwicklung der Applikation. Möchte man eine native App auf mehreren Plattformen bereitstellen, muss für jedes System eine eigene Version der Applikation entwickelt werden.

- Wartung

Daraus ergibt sich auch ein weiterer Vorteil für eine hybride Applikation. Die Wartung stellt weitaus weniger Aufwand dar. Müssen Fehler der App ausgebessert werden oder neue Features implementiert werden, so muss man sich als Entwickler nur um eine Version der Applikation kümmern. Bei nativen Apps hat man als Entwickler den großen Nachteil, dass man sich für jede einzelne Version der App Zeit nehmen muss.

4.6.2.2 Nachteile

- Langsame Performance

Hybride Apps werden nicht direkt auf der CPU ausgeführt, sondern benötigen einige Zwischenschritte. Es fehlen darüber hinaus Funktionen, wie der Garbage-Collector, diverse APIs zu der Hardware und die Unterstützung von Threads. Dies hat den Nachteil, dass diese Apps im Gegensatz zu nativen Apps langsamer sind. Durch die schnelle Hardware der heutigen Zeit wird der Performanceunterschied jedoch immer geringer.

- User Interface

Da eine Hybrid-App nur auf HTML und CSS aufbaut, besitzt die Applikation nicht die internen User Interface Elemente des Betriebssystems. Die Applikation sieht somit auf allen Systemen gleich aus. Dies kann zur Folge haben, dass die App designtechnisch aus dem System herausspringt und der Anwender die App nicht als benutzerfreundlich empfindet. Zum Glück wurden bereits einige Frameworks, wie zum Beispiel „Ionic“ entwickelt, welche genau diese Elemente für eine hybride Applikation bereitstellen und das Design automatisch auf das Betriebssystem abstimmen.

- APIs

Ein weiterer Nachteil ist die sperrige Handhabung mit den Hardwarezugriffen auf dem Smartphone. Kamera, Sensoren und weitere Hardwarekomponenten können zwar auch auf einer Hybrid-App angesprochen werden, jedoch ist der Umgang damit viel kompliziert als in einer nativen App. Cordova bietet dazu verschiedene Plug-Ins an, welche manuell installiert werden müssen, um einen etwas einfacheren Umgang mit den Komponenten zu ermöglichen.

4.6.3 Warum Hybrid-App

Wir haben uns dennoch für eine Hybrid-Applikation entschieden. Für Winetastic sind Hardwarekomponenten, wie die Kamera oder der Fingerabdrucksensor, auf dem Smartphone nicht von besonderer Bedeutung. Smartphones werden von Jahr zu Jahr schneller, somit war für uns die etwas geringere Performance kein stichhaltiger Punkt für eine native App. Des Weiteren standen wir zeittechnisch unter Druck. Ein enormer Vorteil für uns ist die Programmiersprache JavaScript. Dadurch, dass wir bei unserem parallel laufenden zweijährigen Projekt ebenfalls mit JavaScript, CSS und anderen Frameworks arbeiten, verkürzte dies die Einarbeitungsphase.

Dadurch, dass wir „Ionic“ benutzen, konnten wir unsere Applikation mit wenig Mühe mit einem ansprechenden Design ausstatten. Weiters können wir den Wartungsaufwand nach der Entwicklung durch die Verwendung einer Hybrid-App enorm reduzieren.

5 Organisation

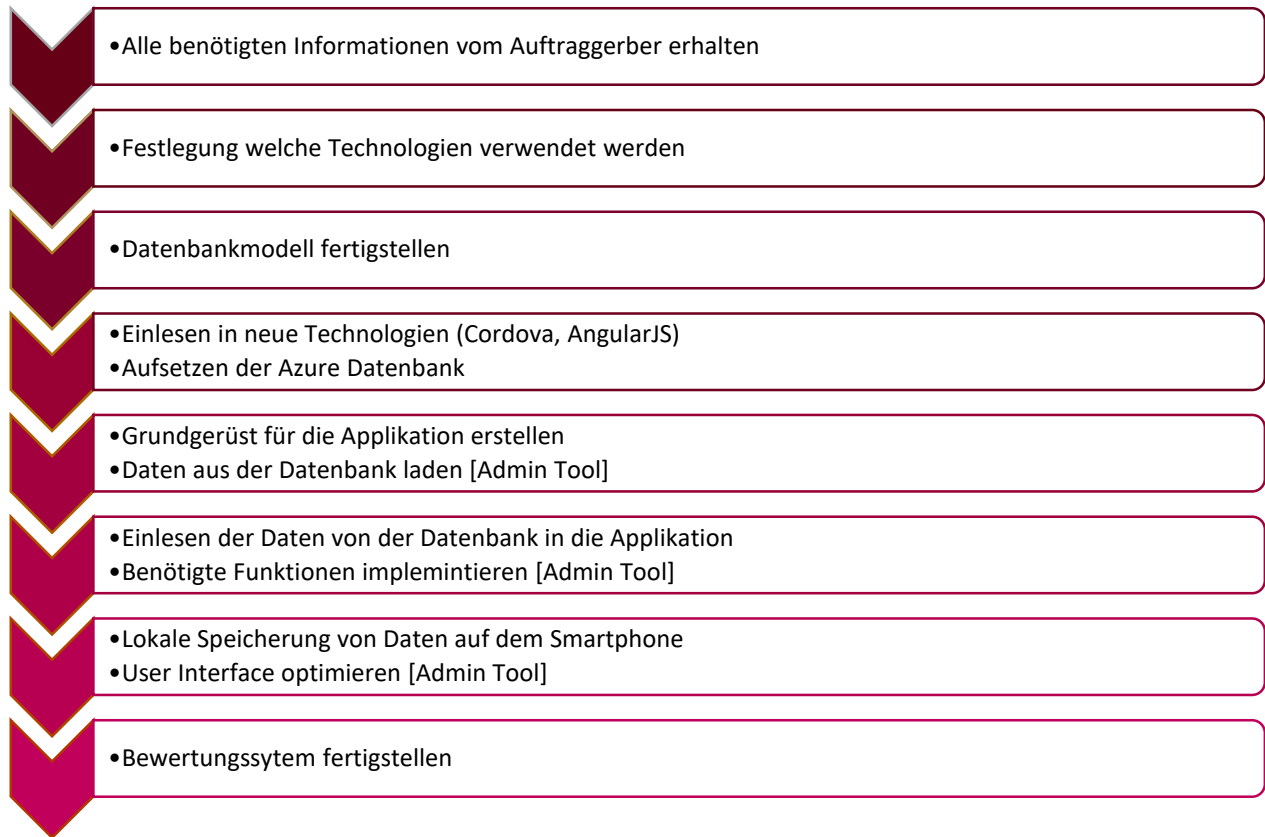
5.1 Kommunikation im Team

Wir, Felix Stumvoll und Ronald Peroutka, haben uns in der Schule über den aktuellen Stand der verschiedenen Teile ausgetauscht. Falls Probleme aufgetreten sind, konnten wir gemeinsam eine Lösung finden.

In den Ferien und an Wochenenden kommunizierten wir über WhatsApp und E-Mail.

Da Felix Stumvoll der Sohn der Auftraggeberin ist, haben wir immer aktuelles Feedback über die verschiedenen Teile der Diplomarbeit erhalten und designtechnische Fragen wurden immer schnell beantwortet.

5.2 Meilensteine



5.3 Risiken

Risiko	Risikohöhe	Eintrittswahrscheinlichkeit
Neue Technologie Cordova	5	5
Excel Import nicht möglich	3	2-3
Schlechte Kommunikation zum Auftraggeber	5	2

5.3.1 Neue Technologie Cordova

5.3.1.1 Beschreibung

Die Applikation kann nicht erstellt werden, da die Technologie Apache Cordova nur schlecht dokumentiert, zu kompliziert oder noch zu unausgereift ist.

5.3.1.2 Lösung

Bereits vor der Erstellung der Applikation wird die Dokumentation der Technologie studiert. Des Weiteren wird ein Professor aufgesucht, der Erfahrungen mit Apache Cordova hat, um aufkommende Fragen zu beantworten.

5.3.2 Excel Import nicht möglich

5.3.2.1 Beschreibung

Der Excel Import kann nicht implementiert werden, da das Excel File kein konkretes Format besitzt.

5.3.2.2 Lösung

Im Zuge der Diplomarbeit wurde ein Webformular erstellt, mit welchem die Winzer ihre Daten eingeben können. Alle Eingaben werden automatisch in ein strukturiertes Excel File gespeichert. Dieses File wird dann von unserem Programm eingelesen.

5.3.3 Schlechte Kommunikation zum Auftraggeber

5.3.3.1 Beschreibung

Durch schlechte Kommunikation mit dem Auftraggeber können wir nicht genau auf die Wünsche des Auftraggebers eingehen.

5.3.3.2 Lösung

Wir haben uns bei wichtigen Entscheidungen, z.B. Designentscheidungen, immer zuerst mit dem Auftraggeber zusammengesetzt, um die Änderung zu besprechen, beziehungsweise auf Verbesserungswünsche einzugehen.

5.4 Probleme

5.4.1 App

5.4.1.1 Lokale Speicherung

Das größte Problem beim Entwickeln der Applikation war die lokale Speicherung der Daten. Um Daten speichern zu können, war zuerst eine ausführliche Einarbeitung nötig. In der Dokumentation von Cordova wurde schließlich mit „Local Storage“ ein möglicher Lösungsansatz erwähnt. Dies ist ein kleiner Bereich in der Applikation, in dem man Daten ablegen und später wieder einlesen kann. Da dieser Bereich nur eine begrenzte Größe hat, war dieser Ansatz für unsere Diplomarbeit irrelevant. Ein anderer Lösungsansatz ist, die Daten lokal in eine Datei zu speichern. Dadurch, dass die Datei in einem separaten Bereich des Smartphones gespeichert wird war die Datei zunächst unauffindbar. Nach einiger Recherche stellte sich heraus, dass der Zugriff auf diesen Bereich nur mit Root-Rechten möglich ist. Das Lesen und Bearbeiten der Datei ist dennoch möglich. Nach erfolgreicher Implementierung dieses Lösungsansatzes entschieden wir uns endgültig für die Speicherung in einer Datei.

5.5 Maßnahmen zur Qualitätssicherung

5.5.1 Review mit dem Auftraggeber

Ein wichtiger Punkt zur Sicherstellung der vom Auftraggeber erwünschten Qualität war die sehr enge Zusammenarbeit mit dem Auftraggeber. Bei aufgetretenen Fragen, bzw. Designentscheidungen wurde immer zuerst Rücksprache mit dem Auftraggeber gehalten, um das Endprodukt möglichst den Wünschen des Kunden anzupassen.

5.5.2 Rücksprache unter Teammitgliedern

Damit bei heiklen Teilen der Diplomarbeit eine gute Lösung gefunden werden konnte, wurden mehrere Lösungsvorschläge unter den Teammitgliedern gesammelt und zusammen die perfekte Lösung ausgewählt. Dies hat nicht nur den Vorteil, dass man Ideen sammelt, auf die man alleine nie kommen würde, sondern auch, dass alle Teammitglieder wissen, wie Teile der Diplomarbeit funktionieren, um zukünftige Probleme eventuell alleine lösen zu können.

6 Persönliches Resümee

6.1 Stumvoll Felix – Admin Tool

Die Diplomarbeit zu entwickeln viel mir, bis auf ein paar Probleme, relativ leicht. Einerseits, da ich in meiner Schullaufbahn schon so manche Erfahrung mit dem .NET Framework sammeln konnte und andererseits da es online zahlreiche und sehr ausführliche Dokumentationen und Beispiele gab. Da es dennoch mein erstes größeres Projekt mit dem .NET Framework war, konnte ich vieles dazulernen. Es bereitete mir große Freude, das Admin Tool zu entwickeln, vor allem deshalb, weil es immer etwas zum Anpassen oder Verbessern gab, sei es ein Benutzerinterface, welches optimiert werden sollte oder der Code-Behind, der verbessert werden musste. Ich habe mich des Öfteren dabei ertappt wie ich Methoden mehrmals optimierte nur um den Code noch besser zu machen.

Organisatorisch gesehen bin ich sehr zufrieden, wie die Diplomarbeit abgelaufen ist. Dadurch, dass meine Mutter die Auftraggeberin war, konnte ich bei jeder Unklarheit direkt mit ihr kommunizieren und so schnell eine Lösung finden. Weiters empfand ich das Arbeiten mit meinem Teamkollegen sehr angenehm. Einerseits hielten wir uns immer an Deadlines und Abmachungen und andererseits konnten wir uns oft gegenseitig helfen bzw. einander Ratschläge geben.

Alles in allem hat mir die Arbeit an der Diplomarbeit großen Spaß gemacht. Außerdem konnte ich meine C#-Kenntnisse stark verbessern. Vor allem im Bereich von Azure Cloud Services konnte ich mein Wissen erweitern. Zum Schluss möchte ich noch sagen, dass ich froh bin, dass ich einerseits einen sehr guten Projektpartner an meiner Seite hatte und andererseits, dass die Diplomarbeit ohne große Schwierigkeiten ablief.

6.2 Ronald Peroutka – App

Die Diplomarbeit im Bereich App Entwicklung war für mich wie ein Sprung ins kalte Wasser. Zuvor hatte ich keine große Erfahrung mit App Entwicklung, Cordova, oder dergleichen gesammelt. Ich ging dabei also ein gewisses Risiko ein, dass ich mit meinem Teil der Diplomarbeit überfordert sein könnte. Ein kleiner Hoffnungsschimmer waren jedoch andere Mitschüler, die entweder bereits in vergangenen Projekten mit Cordova gearbeitet haben oder ebenfalls eine App für ihre eigene Diplomarbeit entwickelten. Somit hatte ich eine gewisse Absicherung, falls ich beim Entwickeln den Wald vor lauter Bäume nicht mehr sehen kann. Außerdem besitzt Cordova eine ausführliche Dokumentierung, was wiederum ein Vorteil für mich war.

Das Entwickeln der App ging mir trotz manch neuer Frameworks, relativ leicht von der Hand. Ich stieß dabei auf wenig technische Fehler und war überrascht, wie schnell ein Grundgerüst für eine App aufgebaut ist. Am meisten Spaß hat mir aber das Designen der App bereitet, da dort meine kreative Seite zum Einsatz kam und es eine Abwechslung vom andauernden systematischen Denken war. „Ionic“ bietet wirklich viele unterschiedliche Möglichkeiten, eine Applikation aufzubauen.

Organisatorisch empfand ich unser Team als sehr angenehm. Ich würde es sogar als „Dream Team“ bezeichnen. Wir waren uns recht schnell in Sachen Technologien oder Design einig. Bei Problemen wurde gemeinsam eine Lösung gefunden und Deadlines wurden stets eingehalten.

Zusammengefasst kann ich sagen, dass die Diplomarbeit mein Wissen im Bereich „App Entwicklung“ um einiges erweitert hat. Das selbständige Arbeiten, welches man in der Schule erlernt, kam hier voll und ganz zum Einsatz und ich bin froh darüber, dass die Diplomarbeit organisatorisch und technisch so einwandfrei verlief.

7 Literaturverzeichnis

- Apsel, M. (kein Datum). https://wiki.selfhtml.org/wiki/CSS/Anwendung_und_Praxis.
- Astrup, A. (kein Datum). <http://de.html.net/tutorials/css/lesson1.php>.
- Foundation, A. (2015). <https://cordova.apache.org/docs/en/latest/guide/support/index.html>.
- https://www.w3schools.com/angular/angular_intro.asp. (kein Datum).
- Kluge, E. (kein Datum). <http://neuwaerts.de/emerge/articles/2013/ionic-frontend-framework/>.
- Koch, S. (2009). *JavaScript. Einführung, Programmierung und Referenz*. Heidelberg.
- Korf, M. (2016). https://developer.salesforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options.
- Lipinski, K. (kein Datum). <http://www.itwissen.info/definition/lexikon/jquery.html>.
- Michal Gralak, T. S. (kein Datum). *Schnelleinstieg App Usability*.
- Pasqua, D. (2011). <http://daniel-pasqua.de/tuts/hypertext.php>.
- selfhtml. (2015). <https://wiki.selfhtml.org/wiki/Grundlagen/Technologien/HTML>.
- Waschier, G. (2011). <https://www.medani.at/was-ist-jquery/>.
- Weiß, B. (kein Datum). <https://angularjs.de/artikel/ionic-tutorial-deutsch>.

8 Abbildungsverzeichnis

Abbildung 1 – derzeitiger Katalog	17
Abbildung 2 – Admin Tool - Veranstaltungsfenster	18
Abbildung 3 – Admin Tool - Datenseite	19
Abbildung 4 - Formular für die Winzer.....	21
Abbildung 5 - Excel – Winzerdaten	22
Abbildung 6 - Excel – Weinliste	22
Abbildung 7 – Admin Tool - Winzeransicht.....	23
Abbildung 8 – Admin Tool - Weinansicht.....	24
Abbildung 9 – Admin Tool – Weinsortenauswahl.....	24
Abbildung 10 – Admin Tool – Winzer hinzufügen.....	25
Abbildung 11 – Admin Tool – Weine hinzufügen.....	26
Abbildung 12 – Admin Tool – Weinsorten bearbeiten	27
Abbildung 13 – Admin Tool – Weinsorten bearbeiten – mit Suchbegriff.....	27
Abbildung 14 - App Startseite	28
Abbildung 15 - App Startseite ohne Internetverbindung.....	28
Abbildung 16 - App Veranstaltungsüberblick.....	29
Abbildung 17 - App Sidebar.....	29
Abbildung 18 - App Weinübersicht	30
Abbildung 19 - App Weindetails Allgemein.....	31
Abbildung 20 - App Weindetails Notizen	31
Abbildung 21 - App Weindetails Bewertung.....	31
Abbildung 22 - App Weindetails Notizen	32
Abbildung 23 - App Winzerdetails Weine	33
Abbildung 24 - App Winzerübersicht	33
Abbildung 25 - App Winzerdetails Info	33
Abbildung 26 - App Meine Notizen	35
Abbildung 27 - App Meine Bewertungen.....	36
Abbildung 26 - App Einstellungen	37

Abbildung 27 - Aufbau Allgemein	40
Abbildung 28 - Kommunikation Veranstalteransicht	41
Abbildung 29 - Kommunikation Anwenderansicht	42
Abbildung 30 - HTML Logo	44
Abbildung 31 - Cascading Style Sheet Logo.....	45
Abbildung 32 - JavaScript Logo.....	46
Abbildung 33 - Ionic Logo	47
Abbildung 34 - Ionic Android und IOS	47
Abbildung 35 - JQuery Logo	48
Abbildung 36 - AngularJS Logo	49
Abbildung 37 - Apache Cordova Logo	51
Abbildung 38 - Apache Cordova Funktionsumfang.....	52
Abbildung 37 – Microsoft Azure Logo.....	53
Abbildung 37 - Datenmodell	54

9 Abkürzungsverzeichnis

API	Application Programming Interface
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheet
DOM	Document Object Model
HTML	Hypertext Markup Language
ID	Identifikation
JSON	JavaScript Object Notation
OS	Operating System
UI	User Interface
WCF	Windows Communication Foundation
WPF	Windows Presentation Foundation
XAML	Extensible Application Markup Language
XML	Extensible Markup Language