

QSE-ASE-08
Tenez
Projektauftrag

Alexander Bohn
Manuel Djalili
Dominik Gruber
Manuel Mayrhofer
Philipp Naderer

31. Oktober 2009

Inhaltsverzeichnis

1	Projektbeschreibung	2
1.1	Projektbezeichnung & Projektpartner	2
1.1.1	Projektbezeichnung	2
1.1.2	Projektteam	2
1.1.3	Auftraggeber	2
1.2	Ausgangssituation	2
1.3	Projektkurzbeschreibung	3
2	Zielgruppen	4
2.1	Vereine	4
2.2	Kunden & Mitglieder eines Vereins	4
2.3	Systembetreiber	4
3	Domänenmodell	5
4	Komponentendiagramm	6
5	Abgrenzungen	7
6	Funktionale Anforderungen	8
7	Use Cases	10
7.1	Übersicht	10
7.2	Vereinverwaltung	12
7.3	Kundenverwaltung	19
7.4	Reportverwaltung	26
7.5	Buchungsverwaltung	29
7.6	Actor hierarchy	31
8	Nichtfunktionale Anforderungen	32
9	Lieferumfang und Abnahme	33
9.1	Lieferumfang	33
9.2	Abnahme	33
10	Arbeitsstruktur & Rollenverteilung	34
11	Projektplan	35
11.1	Iceberg List	36
11.2	Projektstrukturplan	37

12 Risikoanalyse	38
12.1 Übersicht	38
12.2 Beschreibung	39
13 Informationswesen	40
13.1 Teaminterne Kommunikation	40
13.2 SVN-Repository	40
13.3 Projektdokumentation	40
13.4 Quelltexte	40

Kapitel 1

Projektbeschreibung

1.1 Projektbezeichnung & Projektpartner

1.1.1 Projektbezeichnung

Tenez - Open Source Reservierungssystem

1.1.2 Projektteam

Dominik Gruber	Teamkoordinator	dominik.gruber@tenez.at
Philipp Naderer	Software Architekt, TK. Stv.	philipp.naderer@tenez.at
Manuel Mayrhofer	Software Architekt Stv.	manuel.mayrhofer@tenez.at
Alexander Bohn	Tester	jagang@gmx.net
Manuel Djalili	Dokumentbeauftragter	manuel.djalili@gmail.com

Der Auftraggeber kann außerdem unter der E-Mail-Adresse projektleitung@tenez.at die beiden Teamkoordinatoren erreichen. Weiters werden im Verlauf des Projekts immer wieder aktuelle Informationen zur Entwicklung auf der Webseite www.tenez.at veröffentlicht.

1.1.3 Auftraggeber

TC Ulmerfeld-Hausmening
ZVR 885557744
Stadionstraße 6
3363 Ulmerfeld-Hausmening

Ansprechpartner

Horst Haider (Hallenwart)
horst.haider@ktvam.at
+43 664 111 80 10

1.2 Ausgangssituation

Im Moment existieren diverse Reservierungssysteme für Tennisplätze, die jedoch alle große Schwächen im Hinblick auf Administration und Benutzerführung aufweisen. Zumeist ist auch keine direkte Bezahlung durch den Kunden möglich, was darin resultiert, dass weiterhin ein Hallenwart benötigt wird, der die Abrechnung vornimmt. Viele Vereine finden

jedoch keine geeignete Person, die diese Funktion ausüben will. Deswegen ist eine möglichst wartungsarme und einfache Administration eine der wichtigsten Anforderungen.

Weiters bieten bestehende Lösungen keine umfassenden Auswertungen. Excel und selbstgeführte Listen sind bei vielen Vereinen noch immer das einzige Mittel zur Datenauswertung bzw. -erfassung.

Dabei ist die Halle für jeden Verein überlebenswichtig, da zwischen 60 bis 75 Prozent des Jahresumsatzes durch die Hallenvermietung eingenommen werden. Gerade hier will das Projekt ansetzen und eine einfache, schnelle und übersichtliche Auswertung ermöglichen, die allen Anforderungen an ein modernes Reporting gerecht wird. Das ermöglicht es Vereinen beispielsweise die Preise an die Nutzungsfrequenz anzupassen.

1.3 Projektkurzbeschreibung

Das Projekt Tenez soll ein Open Source Hallen-Reservierungssystem erstellen, welches von Vereinen kostengünstig eingesetzt werden kann und einfach in der Benutzung ist. Dabei soll es mehrere Vereine mit unterschiedlichen Hallenarten und auch Buchungsbelegungen verwalten können. Somit ist Tenez als Grundlage für ein zentrales Reservierungssystem für Vereine ausgelegt.

Ein wichtiger Punkt ist die zielgruppengerechte Ausführung der Benutzerschnittstelle. Gerade Personen ab 50 Jahren nehmen häufig den größten Anteil an Stundenreservierungen in Anspruch und somit muss auf ihre Anforderungen eingegangen werden. Ein umständlicher Reservierungsprozess ist unbedingt zu vermeiden.

Kapitel 2

Zielgruppen

2.1 Vereine

Vereine, die nach einem kostengünstigen und einfachen Reservierungssystem suchen. Sie wollen die Administration möglichst einfach halten und somit auch die anfallenden Kosten. Sie sollen Standorte mit Plätzen anlegen können. Weiters soll das System genaue Auflistungen von verkauften bzw. gebuchten Stunden liefern und auf Abruf immer die aktuelle Auslastung anzeigen. Abonnements müssen schnell und einfach eingetragen werden können. Zu unterschiedlichen Tageszeiten bzw. an speziellen Tagen können auch unterschiedliche Preise gelten.

2.2 Kunden & Mitglieder eines Vereins

Kunden & Mitglieder eines Vereins, die Einheiten von einem Platz buchen und sich über ihr Guthaben informieren wollen. Durch ein Reservierungssystem können sie unabhängig von einem Hallenwart einen Überblick über die momentane Buchungssituation erhalten. Dies kann auch über ein mobiles Gerät mit Internetzugang erfolgen.

Kunden sind Personen, die bei einem Verein ein Guthaben besitzen, aber keine eingetragenen Mitglieder sind.

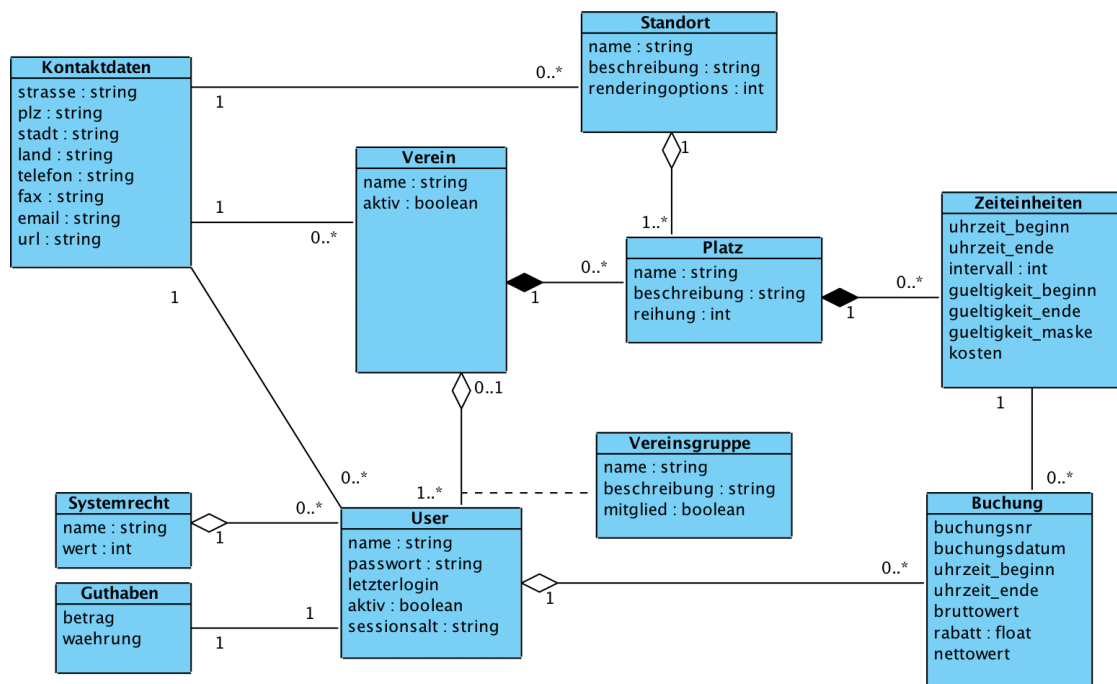
2.3 Systembetreiber

Systembetreiber, der Vereine anlegen kann. Zu diesen Vereinen kann er User und Plätze (mit Standorten) anlegen. Er kann User zu Verwalter des Vereins ernennen.

Er agiert als Anbieter des Reservierungssystem und hat für den reibungslosen Betrieb zu sorgen bzw. eventuelle Probleme und Anfragen von Vereinen und Usern zu lösen.

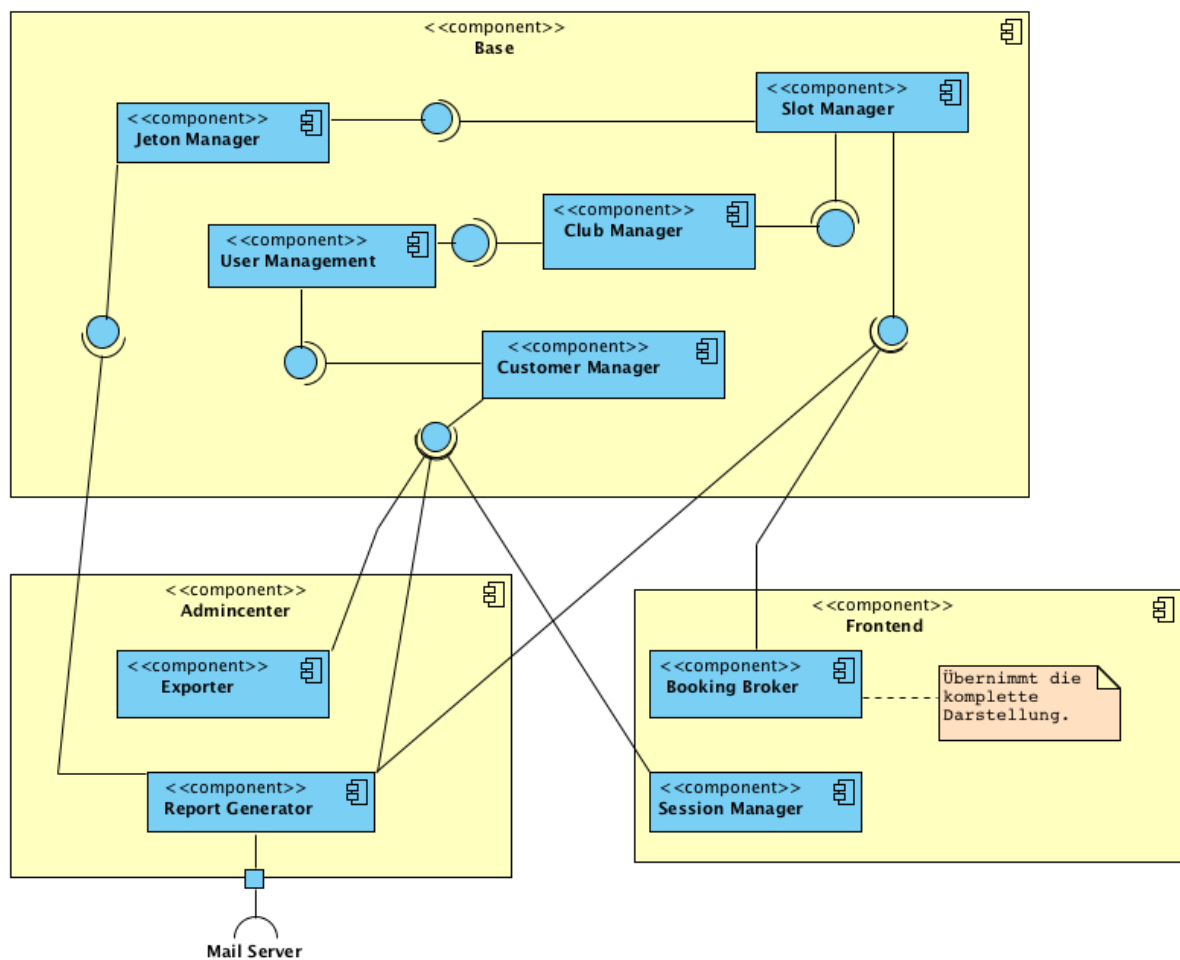
Kapitel 3

Domänenmodell



Kapitel 4

Komponentendiagramm



Kapitel 5

Abgrenzungen

Das Reservierungssystem soll keine Zeitschaltuhr umfassen, welche die Beleuchtung in einer Halle ein- oder ausschaltet. Chipkarten, Jetons und andere Zutrittsberechtigungen können nicht mit dem System verwaltet, bearbeitet oder ausgegeben werden.

Zwar wird ein Kunde grundsätzlich einem Verein als Mitglied zugeordnet, jedoch ist damit keine Vereinsverwaltung angedacht, sondern nur eine für Auswertungen und Rabatten wichtige Trennung zwischen auswärtigen Spielern und Mitgliedern.

Das System ist nicht für die Verwaltung und Planung von Tennisturnieren bzw. deren Spielreihenfolge gedacht.

Eine möglichst störungsfreie Internetanbindung ist ein wichtiges Kriterium zum reibungslosen Ablauf, im Projekt wird darauf aber keine Rücksicht genommen und fällt in die Zuständigkeit des lokalen Systembetreuers. Das System ist weiters nicht für die Erstellung von Backups verantwortlich.

Für eine korrekte Darstellung der Templates muss auf Standgeräten zumindest eine Auflösung von 1024x768 Pixel eingestellt sein, für die mobile Version wird zumindest die Auflösung 480x320 Pixel benötigt.

Kapitel 6

Funktionale Anforderungen

- **Vereinsverwaltung**

Der Systemadministrator muss Vereine anlegen, bearbeiten und löschen können.

- **Benutzerverwaltung**

Jeder muss sich bei einem Verein als Benutzer registrieren und einloggen können. Eine E-Mail Adresse ist dafür keine Voraussetzung. Ein alphanumerisches Passwort mit mindestens fünf Stellen wird dabei automatisch generiert.

- **Vereinsgruppen verwalten**

Vereinsverwalter können für ihre Vereine Gruppierungen der Benutzer anlegen (z.B. Senioren, Kinder, Mitglied).

- **Benutzerstammdaten verwalten**

Benutzer können ihre persönlichen Daten und ihr Passwort bearbeiten. Der Systemadministrator kann User zu Vereinsverwaltern ernennen. Der Vereinsverwalter kann User bestimmten internen Vereinsgruppen zuordnen. Weiters kann er für jeden User ein Guthaben verwalten, mit dem dieser seine Reservierungen vornehmen kann.

- **Platzverwaltung**

Der Systemadministrator muss für einen Verein Plätze anlegen können. Der Vereinsverwalter kann Zeiteinheiten bestimmen, zu denen der Platz für Mitglieder zur Reservierung offen steht. Dabei soll es möglich sein, dass jede Benutzergruppe ihren eigenen Preis erhält.

- **Stammdaten exportieren**

Vereinsverwalter können die Stammdaten ihrer Benutzer als CVS exportieren um diese so in externe Textverarbeitungsprogramme zu importieren. Beim Exportvorgang können Filterkriterien, z.B. nur Mitglieder, angegeben werden.

- **Reservierung vornehmen**

Benutzer reservieren über eine HTML-Maske im Browser einen vom System als frei markierten Eintrag in der Stundenübersicht. Die Stundenübersicht ist eine tabellarische Ansicht für den vorher ausgewählten Platz des Vereins. Alternativ können auch der Systemadministrator und der Vereinsverwalter Reservierungen von Kunden eintragen. Diese können außerdem Abonnements mit variabler Laufzeit für Kunden eintragen und selbst Reservierungen ohne Kosten vornehmen (z.B. für Mannschaftstraining, Turniere oder Wartungsarbeiten).

- **Skins für Vereine**
Für jeden Verein kann vom Systemadministrator ein individueller Skin ausgewählt werden, um das Layout an die Vereinsfarben anzupassen.
- **Buchungsübersicht**
Der Administrator und Vereinsverwalter kann eine Übersicht der letzten Buchungen abrufen, mit genauem Datum und dem Kundennamen. Er kann einzelne Buchungen stornieren und neu anlegen.
- **Auswertungen vornehmen**
Auswertungen sind eine der zentralen und innovativen Funktionen, die im Projekt umgesetzt werden. Ein Beispiel dafür ist die Anzeige der Auslastung einzelner Zeiteinheiten über einen gewissen Zeitraum. Dieser Wert ist ein Indikator dafür ob der Preis zu bestimmten Zeiten evtl. erhöht bzw. gesenkt werden sollte um sowohl die Auslastung als auch die Einnahmen möglichst hoch zu halten. Weiters denkbar ist eine Auswertung der Einnahmen zu einem definierbaren Zeitraum (Tag/Monat/Jahr) oder durch bestimmte Kunden.
- **Guthaben abfragen**
Jeder Benutzer kann sich am System anmelden und sein Guthaben abfragen. Weiters sieht er seine letzten Buchungen aufgeschlüsselt, damit er missbräuchlichen Zugriff schneller erkennen kann.
- **Mobile Version**
Durch die Bereitstellung einer Layoutvariante, speziell angepasst an die Displaygröße von mobilen Geräten mit Internetanschluss (z.B. iPhone/iPod Touch), soll für Benutzer eine Abfrage des Buchungsstandes sowie eine Reservierung möglich sein.
- **Import von bestehenden Daten**
Die bereits bestehende PHP-Anwendung des TC Ulmerfeld-Hausmenings enthält zahlreiche Kundendaten. Es sind Name, Zugangs-PIN, Adresse und E-Mail-Adresse als Felder in der Datenbank vorhanden. Diese sollen vollständig in die neue Applikation übernommen werden bzw. gegebenenfalls mit geringem Aufwand importiert werden können.
- **Newsletter**
Dem Systemadministrator und den Vereinsverwaltern wird eine Maske bereitgestellt, um den Kunden einen Newsletter schicken zu können.
- **Kontakt**
Der Vereinsverwalter kann direkt über ein Kontaktformular kontaktiert werden.
- **Validatoren**
Die Applikation hat gegen gezielte bzw. auch nicht beabsichtigte XSS-Angriffe geschützt zu sein. Hierfür müssen alle Benutzereingaben einer Validierung unterzogen oder vor der Ausgabe richtig gefiltert werden.
- **Online-Überweisung (Optional)**
Um neues Guthaben zu erhalten, könnte der Kunde mit einer Online-Überweisung bezahlen. Varianten wären EPS oder PayPal.

Kapitel 7

Use Cases

7.1 Übersicht

ID	Bezeichnung	Seitennummer
UC-01-01	Verein anlegen	12
UC-01-02	Vereine anzeigen	12
UC-01-03	Verein löschen	13
UC-01-04	Zeiteinheiten löschen	13
UC-01-05	Plätze löschen	13
UC-01-06	Platz anlegen	14
UC-01-07	Vereinsdaten bearbeiten	14
UC-01-08	Skin auswählen	14
UC-01-09	Usergruppe anlegen	15
UC-01-10	Plätze anzeigen	15
UC-01-11	Zeiteinheiten festlegen	15
UC-01-12	Platz reservieren	16
UC-01-13	Standort anzeigen	x
UC-01-14	Standort löschen	x
UC-01-15	Standort anlegen	x
UC-01-16	Standort zuweisen	x

ID	Bezeichnung	Seitennummer
UC-02-01	Kunden registrieren	17
UC-02-02	Kunden anzeigen	17
UC-02-03	Gruppe zuordnen	18
UC-02-04	Kunden löschen	18
UC-02-05	Guthaben verändern	18
UC-02-06	Kundendaten exportieren	19
UC-02-08	Newsletter verschicken	19
UC-02-09	registrieren	20
UC-02-10	anmelden	20
UC-02-11	Guthaben abfragen	20
UC-02-12	Buchungshistory anzeigen	21
UC-02-13	Platz reservieren	21
UC-02-14	löschen	21
UC-02-15	Daten bearbeiten	22
UC-02-16	Guthaben kaufen	22
UC-02-17	Guthaben aufladen	22
UC-02-18	Admin kontaktieren	23
UC-02-19	Kundenrating erstellen	23

ID	Bezeichnung	Seitennummer
UC-03-01	Auswertung wählen	24
UC-03-02	Reports on demand generieren	25
UC-03-03	Top Tageszeiten	25
UC-03-04	Auswärtige Kunden vs. Mitglieder	25
UC-03-05	Buchungsbelegung	25
UC-03-06	Meistgebuchte Wochen	26
UC-03-07	Kundenliste	x
UC-03-08	Zugriffsauswertung	x
UC-03-09	Filterung aktivieren	x
UC-03-10	periodische Reports generieren	x

ID	Bezeichnung	Seitennummer
UC-04-01	Buchungen abrufen	27
UC-04-02	Buchungen stornieren	27
UC-04-03	Buchung anlegen	28
UC-04-04	Abonnement anlegen	x

7.2 Vereinverwaltung

UC-01-01 Verein anlegen	
Ziel	Verein in der Datenbank abspeichern
Akteure	Admin
Vorbedingung	Admin muss angemeldet sein
Auslösendes Ereignis	-
Ablauf	1. Admin klickt auf den Button "Verein anlegen" 2. Admin gibt die Vereinsdaten in die Maske ein 3. Admin bestätigt die Eingabe in dem auf den Button Speichern geklickt wird
Alternativen	1. Admin klickt auf Abbrechen 2. Admin wird auf seine Benutzeroberfläche zurück verlinkt
Nachbedingung Erfolg	Verein wird in der Datenbank gespeichert
Nachbedingung Fehlschlag	1. Fehlermeldung bei falscher Eingabe von Daten
Stakeholder	Verein

UC-01-02 Vereine anzeigen	
Ziele	Alle oder nur einen bestimmten Verein anzeigen lassen
Akteure	Admin
Vorbedingung	Admin muss angemeldet sein
Auslösendes Ereignis	-
Ablauf	1. Admin tippt in das Suchformular Name des Vereins ein
Alternativen	1. Admin klickt auf den Button "Vereinsliste"
Nachbedingung Erfolg	Der gesuchte Verein wird angezeigt oder alle Vereine werden angezeigt
Nachbedingung Fehlschlag	1. Bei suche eines nicht vorhandenen Vereins wird eine Fehlermeldung angezeigt
Stakeholder	Admin, um so Vereine verwalten zu können

UC-01-03 Verein löschen	
Ziele	Verein aus der Datenbank entfernen
Akteure	Admin
Vorbedingung	1. Admin muss angemeldet sein 2. Admin muss sich den Verein anzeigen lassen
Auslösendes Ereignis	-
Ablauf	1. Admin klickt auf den Löschenbutton des Vereins 2. Meldung ob Verein wirklich gelöscht werden soll erscheint 3. Admin bestätigt
Alternativen	1. Admin bestätigt das Löschen nicht
Nachbedingung Erfolg	1. Der Verein wird aus der Datenbank entfernt 2. Platz/Plätze des Vereins werden gelöscht 3. Zeiteinheiten des Vereins werden gelöscht
Nachbedingung Fehlschlag	Fehlermeldung, Vorgang wird abgebrochen
Stakeholder	Admin, Verein

UC-01-04 Zeiteinheiten löschen	
Ziele	Zeiteinheiten des Vereins werden gelöscht
Akteure	-
Vorbedingung	1. Admin muss angemeldet sein 2. Admin muss das Löschen eines Vereins bestätigen
Auslösendes Ereignis	Verein löschen
Ablauf	1. Automatisches löschen der Zeiteinheiten
Alternativen	-
Nachbedingung Erfolg	1. Die Zeiteinheiten werden aus der Datenbank entfernt
Nachbedingung Fehlschlag	-
Stakeholder	Admin, Verein

UC-01-05 Plätze löschen	
Ziele	Plätze des Vereins werden gelöscht
Akteure	-
Vorbedingung	1. Admin muss angemeldet sein 2. Admin muss das Löschen eines Vereins bestätigen
Auslösendes Ereignis	Verein löschen
Ablauf	1. Automatisches löschen der Plätze
Alternativen	-
Nachbedingung Erfolg	1. Plätze sind nicht mehr in der Datenbank vorhanden
Nachbedingung Fehlschlag	-
Stakeholder	Admin, Verein

UC-01-06 Platz anlegen	
Ziele	Für einen Verein einen Platz anlegen
Akteure	-Admin
Vorbedingung	1. Admin muss angemeldet sein 2. Admin muss sich den Verein anzeigen lassen
Auslösendes Ereignis	-
Ablauf	1. Admin klickt auf den entsprechenden Button um einen Platz für diesen Verein anzulegen 2. Eine neue Maske öffnet sich in der alle Platzrelevanten Daten eingegeben werden 3. Nach Eingabe klickt der Admin auf speichern
Alternativen	Admin klickt auf Abbrechen
Nachbedingung Erfolg	Platz wird für den Verein in der Datenbank angelegt
Nachbedingung Fehlschlag	-
Stakeholder	Admin, Verein

UC-01-07 Vereinsdaten bearbeiten	
Ziele	Daten eines Vereins erfolgreich verändern
Akteure	Admin, Vereinsverwalter
Vorbedingung	1. Admin muss angemeldet sein 2. Admin muss sich den Verein anzeigen lassen
Auslösendes Ereignis	-
Ablauf	1. Admin klickt auf den entsprechenden Button um Daten des Vereins bearbeiten zu können 2. Eine neue Maske öffnet sich in der die aktuellen Vereinsdaten in Inputfeldern als Defaultwerte ersichtlich sind 3. Admin verändert Daten 4. Admin speichert die Änderung
Alternativen	Admin klickt auf Abbrechen
Nachbedingung Erfolg	Die bearbeiteten Daten werden gespeichert
Nachbedingung Fehlschlag	-
Stakeholder	Admin, Vereinsverwalter

UC-01-08 Skin auswählen	
Ziele	Für einen Verein ein Skin hochladen
Akteure	- Admin, Vereinsverwalter
Vorbedingung	1. Akteur muss angemeldet sein 2. Admin muss sich den Verein anzeigen lassen
Auslösendes Ereignis	-
Ablauf	1. Akteur klickt auf den entsprechenden Button um ein Skin für einen Verein hochladen zu können 2. Eine neue Maske öffnet sich in der Akteur ein lokalgespeichertes Bild als Skin suchen und uploaden kann 3. Akteur klickt auf ok
Alternativen	Akteur klickt auf Abbrechen
Nachbedingung Erfolg	Skin für Verein wurde erstellt
Nachbedingung Fehlschlag	Bild zu groß, falsches Format führt zu einer Fehlermeldung und Abbruch des Vorganges
Stakeholder	Admin, Verein, Kunde

UC-01-09 Usergruppe anlegen	
Ziele	Anlegen einer Usergruppe
Akteure	Vereinsverwalter
Vorbedingung	Vereinsverwalter muss im System angemeldet sein
Auslösendes Ereignis	-
Ablauf	1. Akteur klickt auf Button Usergruppe anlegen 2. Neue Maske erscheint in der Akteur Daten und Rechte der Gruppe eingibt 3. Akteur speichert seine Eingaben
Alternativen	Akteur verwirft seine Eingabe
Nachbedingung Erfolg	1. Bestätigung erscheint 2. Akteur wird auf seine Benutzeroberfläche zurück verlinkt
Nachbedingung Fehlschlag	1. Bestätigung das nicht gespeichert wurde erscheint 2. Akteur wird auf seine Benutzeroberfläche zurück verlinkt
Stakeholder	Benutzer, Verein

UC-01-10 Plätze anzeigen	
Ziele	Vorhandene Plätze des Vereins anzeigen
Akteure	Vereinsverwalter
Vorbedingung	Vereinsverwalter muss angemeldet sein
Auslösendes Ereignis	-
Ablauf	1. Übersicht über alle Plätze des Vereins wird angezeigt
Alternativen	-
Nachbedingung Erfolg	-
Nachbedingung Fehlschlag	-
Stakeholder	Verein

UC-01-11 Zeiteinheiten festlegen	
Ziele	Zeiteinheiten an denen Mitglieder den Platz benutzen dürfen festlegen zu können
Akteure	Vereinsverwalter
Vorbedingung	1. Vereinsverwalter muss angemeldet sein 2. Vereinsverwalter muss sich alle Plätze seines Vereins anzeigen lassen
Auslösendes Ereignis	-
Ablauf	1. Platz auswählen 2. Auf Zeiteinheiten erstellen klicken 3. Neue Zeiteinheit für Mitglieder eintragen 4. Auf speichern klicken
Alternativen	Auf Abbrechen klicken
Nachbedingung Erfolg	Innerhalb der Zeiteinheiten können Mitglieder nun Buchungen vornehmen
Nachbedingung Fehlschlag	1. Auf schon vergebene Zeiteinheit geklickt 2. Fehlermeldung erscheint
Stakeholder	Mitglieder, Verein

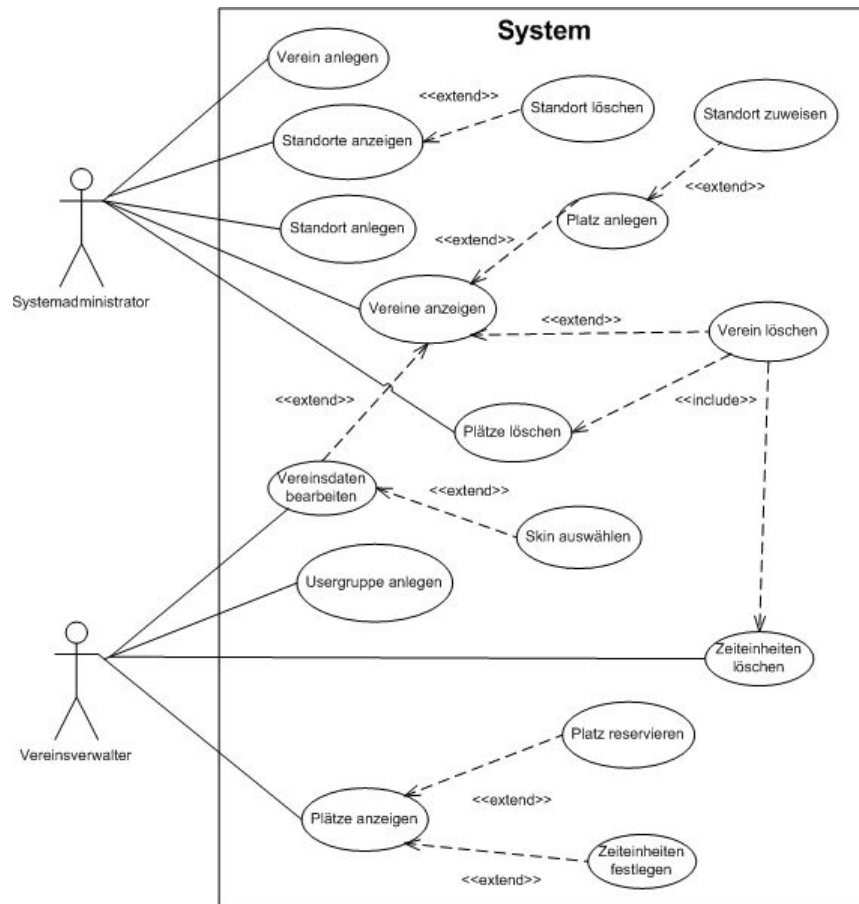
UC-01-12 Platz reservieren	
Ziele	Platz in der Stundenliste reservieren
Akteure	Vereinsverwalter
Vorbedingung	1. Vereinsverwalter muss angemeldet sein 2. Vereinsverwalter muss sich alle Plätze seines Vereins anzeigen lassen
Auslösendes Ereignis	-
Ablauf	1. Platz auswählen 2. Freien Eintrag in der nun angezeigten Stundenliste auswählen 4. Meldung erscheint ob gebucht werden soll 5. Meldung bestätigen 6. Verlinkung auf anfängliche Benutzeroberfläche
Alternativen	1. Meldung nicht bestätigen 2. Verlinkung auf anfängliche Benutzeroberfläche
Nachbedingung Erfolg	Reservierung erscheint in der Stundenliste
Nachbedingung Fehlschlag	Reservierung erscheint nicht in der Stundenliste
Stakeholder	Verein, Mitglieder

UC-01-13 Standorte anzeigen	
Ziele	Standort anzeigen lassen
Akteure	Admin
Vorbedingung	1. Admin muss angemeldet sein
Auslösendes Ereignis	-
Ablauf	1. Standort auswählen
Alternativen	-
Nachbedingung Erfolg	Standorte werden angezeigt
Nachbedingung Fehlschlag	Standorte werden nicht angezeigt
Stakeholder	Verein, Admin

UC-01-14 Standort löschen	
Ziele	Standort löschen
Akteure	Admin
Vorbedingung	1. Admin muss angemeldet sein 2. Admin muss sich den Standort anzeigen lassen
Auslösendes Ereignis	-
Ablauf	1. Standort auswählen 2. Standort löschen 3. bestätigen
Alternativen	-
Nachbedingung Erfolg	Standort ist gelöscht
Nachbedingung Fehlschlag	Standort ist nicht gelöscht
Stakeholder	Verein, Mitglieder

UC-01-15 Standort anlegen	
Ziele	Einen Standort anlegen
Akteure	Admin
Vorbedingung	1. Admin muss angemeldet sein
Auslösendes Ereignis	-
Ablauf	1. Standortdaten in die Maske eingeben 2. bestätigen
Alternativen	-
Nachbedingung Erfolg	Standort wird angelegt
Nachbedingung Fehlschlag	Standort wird nicht angelegt
Stakeholder	Verein, Mitglieder

UC-01-16 Standort zuweisen	
Ziele	Verein einen Standort zuweisen
Akteure	Admin
Vorbedingung	1. Admin muss angemeldet sein 2. Admin muss sich Vereine anzeigen lassen 3. Admin muss einen Platz angelegt haben
Auslösendes Ereignis	-
Ablauf	1. Verein auswählen 2. Platz auswählen 3. Standort auswählen 5. bestätigen
Alternativen	-
Nachbedingung Erfolg	Standort wird zugewiesen
Nachbedingung Fehlschlag	Standort wird nicht zugewiesen
Stakeholder	Verein, Mitglieder



7.3 Kundenverwaltung

UC-02-01 Kunden registrieren	
Ziel	Einen neuen User mit seinen Daten registrieren
Akteure	Admin, Vereinsverwalter
Vorbedingung	Admin muss in dem System angemeldet sein
Auslösendes Ereignis	-
Ablauf	<ol style="list-style-type: none"> 1. Admin klickt auf "Kunden registrieren" 2. Admin gibt die Daten des neuen Users ein. 3. Admin bestätigt die Registrierung. 4. Ein automatisch generiertes Passwort wird angezeigt 5. Maske schließt sich automatisch, Admin befindet sich wieder in seiner anfänglichen Benutzeroberfläche
Alternativen	<ol style="list-style-type: none"> 1. Admin bestätigt die Registrierung nicht 2. Admin wird zu seiner Benutzeroberfläche zurück verlinkt
Nachbedingung Erfolg	Der User ist erfolgreich registriert und in der Datenbank abgebildet
Nachbedingung Fehlschlag 2. Kunde wird nicht in Datenbank abgespeichert	1. Fehlermeldung bei Eingabe falscher Daten.
Stakeholder	Kunde, Admin, Vereinsverwalter

UC-02-02 Kunden anzeigen	
Ziel	Die Daten eines registrierten Kunden anzeigen
Akteure	Admin, Vereinsverwalter
Vorbedingung	Der Admin muss angemeldet sein
Auslösendes Ereignis	-
Ablauf	<ol style="list-style-type: none"> 1. Admin klickt auf "Kunden anzeigen" 2. Admin sucht den Kunden mit der Suchfunktion oder wählt ihn aus einer Liste aus.
Alternativen	-
Nachbedingung Erfolg	Kundendaten werden gezeigt
Nachbedingung Fehlschlag	Gesuchter Kunde existiert nicht, daher wird Suchanfrage ergab keine Treffer angezeigt
Stakeholder	Admin, Vereinsverwalter

UC-02-03 Gruppe zuordnen	
Ziel	Kunde einer Gruppe zuordnen
Akteure	Admin, Vereinsverwalter
Vorbedingung	1. Der Akteur muss angemeldet sein 1. Der Akteur muss sich Kunden anzeigen lassen
Auslösendes Ereignis	-
Ablauf	1. Akteur klickt auf "Gruppe zuordnen" beim jeweiligen Kunden 2. Akteur wählt entsprechend Gruppe aus einer Liste aus 3. Akteur bestätigt
Alternativen	Admin kann Usern nur Gruppe: Vereinsverwalter zuordnen
Nachbedingung Erfolg	User ist nun einer Gruppe zugeordnet
Nachbedingung Fehlschlag	-
Stakeholder	Admin, Vereinsverwalter

UC-02-04 Kunden löschen	
Ziel	Einen registrierten User aus dem System löschen
Akteure	Admin, Vereinsverwalter
Vorbedingung	Admin muss eingeloggt sein Admin muss sich Kunden anzeigen lassen
Auslösendes Ereignis	-
Ablauf	1. Akteur lässt sich den User anzeigen 2. Akteur klickt auf User löschen und 3. bestätigt
Alternativen	Akteur bestätigt das Löschen nicht
Nachbedingung Erfolg	Der User wird aus dem System entfernt
Nachbedingung Fehlschlag	1. Akteur wurde schon gelöscht (z.B. Admin vor 10 Sekunden) 2. Eine entsprechende Meldung erscheint
Stakeholder	Kunde, Admin, Vereinsverwalter

UC-02-05 Guthaben verändern	
Ziel	Das Guthaben eines Kunden verändern
Akteure	Vereinsverwalter, Systemadministrator
Vorbedingung	Akteur muss eingeloggt sein Akteur muss Kundendaten anzeigen auswählen
Auslösendes Ereignis	-
Ablauf	1. Akteur lässt sich den User anzeigen 2. Akteur klickt auf den Menüpunkt: Guthaben aufladen 3. Akteur ändert den Betrag und 4. bestätigt.
Alternativen	Akteur bestätigt nicht
Nachbedingung Erfolg	Kunde hat ein anderes Guthaben
Nachbedingung Fehlschlag	Fehlermeldung bei falschen Betrag (minus, nicht numerische Ziffern usw.) + Kunde hat das selbe Guthaben
Stakeholder	Kunde, Vereinsverwalter

UC-02-06 Kundendaten exportieren	
Ziel	Kundendaten als CVS exportieren
Akteure	Admin, Vereinsverwalter
Vorbedingung	Admin muss eingeloggt sein
Auslösendes Ereignis	-
Ablauf	<ol style="list-style-type: none"> 1. Admin klickt auf den Menüpunkt "Kundendaten exportieren" 2. Admin wählt entweder alle Kundendaten aus oder nach Verein 3. Admin wählt lokalen Speicherort 4. Download erfolgt
Alternativen	-
Nachbedingung Erfolg	Kundendaten sind als CVS beim Admin lokal gespeichert
Nachbedingung Fehlschlag	Fehlermeldung + nichts wurde exportiert
Stakeholder	Kunde, Vereinsverwalter, Admin

UC-02-07 Newsletter verschicken	
Ziel	Vereinsverwalter kann Newsletter an registrierte User schicken
Akteure	Vereinsverwalter, Admin
Vorbedingung	Vereinsverwalter muss eingeloggt sein
Auslösendes Ereignis	-
Ablauf	<ol style="list-style-type: none"> 1. Vereinsverwalter klickt auf den Menüpunkt "Newsletter verschicken" 3. Wählt in der Maske die Empfänger aus 4. Er schreibt den Newsletter. 5. Vereinsverwalter klickt auf senden
Alternativen	Vereinsverwalter klickt auf Abbrechen
Nachbedingung Erfolg	Die ausgewählten Mitglieder bekommen Newsletter
Nachbedingung Fehlschlag	Fehlermeldung + Newsletter wird nicht verschickt
Stakeholder	Kunde, Vereinsverwalter

UC-02-08 registrieren	
Ziel	Kunde kann sich bei dem System registrieren
Akteure	Kunde
Vorbedingung	-
Auslösendes Ereignis	-
Ablauf	<ol style="list-style-type: none"> 1. Kunde klickt auf den Menüpunkt "registrieren" 2. Kunde gibt seine Daten in die Maske ein 3. Kunde bestätigt das captcha 4. Kunde schickt seine Registrierung ab
Alternativen	Admin registriert Kunden
Nachbedingung Erfolg	Kunde wird für das System freigeschalten
Nachbedingung Fehlschlag	<ol style="list-style-type: none"> 1. Fehlermeldung bei falscher Captcha Eingabe 2. Kund wird nicht registriert
Stakeholder	Admin, Kunde, Vereinsverwalter

UC-02-09 anmelden	
Ziel	Kunde meldet sich an und wird zu seinem Menü weitergeleitet
Akteure	Kunde
Vorbedingung	Kunde muss registriert sein
Auslösendes Ereignis	-
Ablauf	1. Kunde gibt seinen Namen und Passwort ein 2. Kunde bestätigt
Alternativen	-
Nachbedingung Erfolg	Kunde wird zu seinem Menü weitergeleitet
Nachbedingung Fehlschlag	Fehlermeldung aufgrund falschem Username/Passwort+ Kunde bleibt ausgeloggt
Stakeholder	Kunde

UC-02-10 Guthaben abfragen	
Ziel	Aktuelles Guthaben des Kunden anzeigen
Akteure	Kunde
Vorbedingung	Kunde muss angemeldet sein
Auslösendes Ereignis	-
Ablauf	1. Kunde klickt auf den Menüpunkt "Guthaben abfragen"
Alternativen	-
Nachbedingung Erfolg	Aktuelles Guthaben wird angezeigt
Nachbedingung Fehlschlag	Fehlermeldung + Guthaben wird nicht angezeigt
Stakeholder	Kunde

UC-02-11 Buchungshistory anzeigen	
Ziel	Buchungshistory des Kunden anzeigen
Akteure	Kunde
Vorbedingung	1. Kunde muss angemeldet sein 2. Kunde muss in dem Menüpunkt Guthaben abfragen sein
Auslösendes Ereignis	-
Ablauf	Buchungshistory wird automatisch zur Guthabenabfrage dazu angezeigt
Alternativen	-
Nachbedingung Erfolg	Die Buchungshistory des Kunden wird angezeigt
Nachbedingung Fehlschlag	Fehlermeldung + Buchungshistory wird nicht angezeigt
Stakeholder	Kunde

UC-02-12 Platz reservieren	
Ziel	Kunde reserviert einen Platz
Akteure	Kunde
Vorbedingung	Kunde muss angemeldet sein
Auslösendes Ereignis	-
Ablauf	<ol style="list-style-type: none"> 1. Kunde klickt auf "Platz reservieren" 2. Kunde wählt einen Platz aus 3. Kunde markiert in der Stundenliste einen oder mehrere freie Slots 4. Kunde bestätigt 5. Guthaben des Kunden wird um Preis für die Buchung vermindert
Alternativen	Kunde bestätigt nicht
Nachbedingung Erfolg	Platz ist reserviert
Nachbedingung Fehlschlag	Fehlermeldung + Platz ist nicht reserviert
Stakeholder	Kunde

UC-02-13 löschen	
Ziel	Kunde ist aus dem System gelöscht
Akteure	Kunde
Vorbedingung	Kunde muss angemeldet sein
Auslösendes Ereignis	-
Ablauf	<ol style="list-style-type: none"> 1. Kunde klickt auf "Löschen" 2. Kunde bestätigt
Alternativen	Admin löscht Kunden
Nachbedingung Erfolg	Kunde ist aus dem System gelöscht
Nachbedingung Fehlschlag	Fehlermeldung + Kunde wird nicht gelöscht
Stakeholder	Kunde

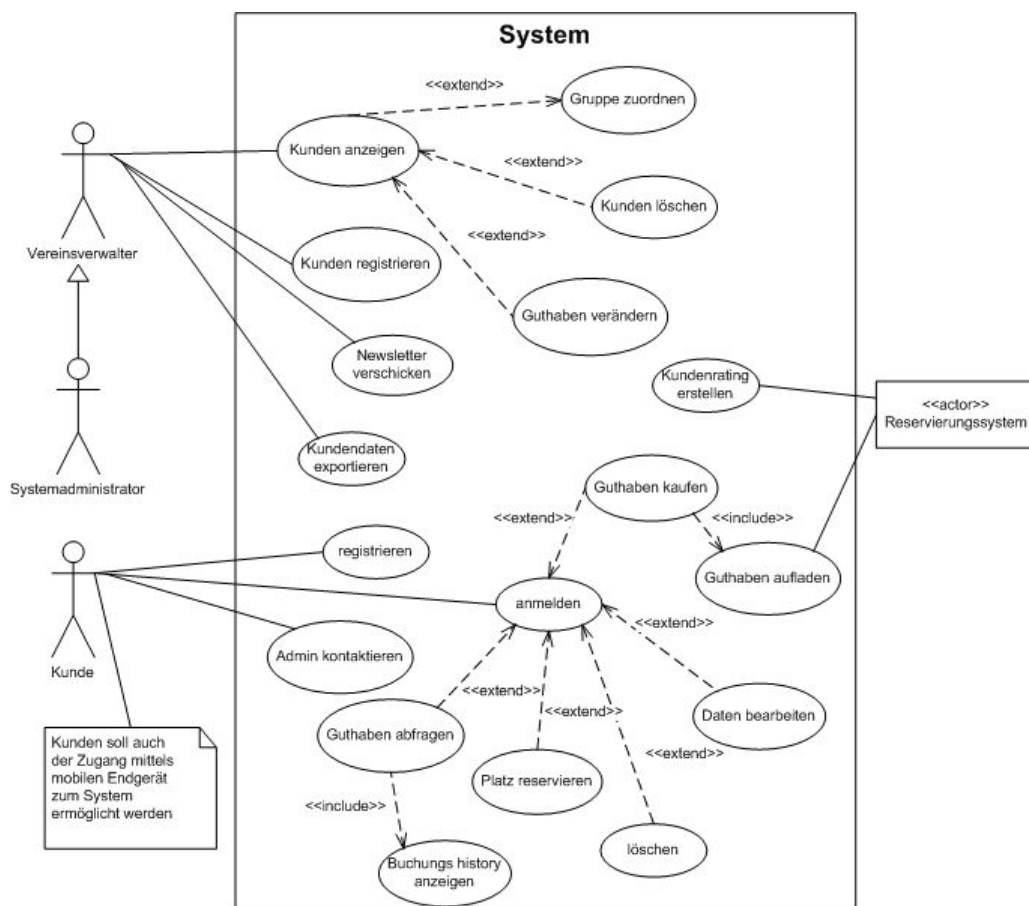
UC-02-14 Daten bearbeiten	
Ziel	Kunde bearbeitet seine Daten
Akteure	Kunde
Vorbedingung	Kunde muss angemeldet sein
Auslösendes Ereignis	-
Ablauf	<ol style="list-style-type: none"> 1. Kunde klickt auf "Daten bearbeiten" 2. Kunde bearbeitet seine Daten und 3. er bestätigt.
Alternativen	-
Nachbedingung Erfolg	Kundendaten werden verändert
Nachbedingung Fehlschlag	Fehlermeldung + Kundendaten bleiben gleich
Stakeholder	Kunde

UC-02-15 Guthaben kaufen	
Ziel	Kunde hat mehr Guthaben
Akteure	Kunde
Vorbedingung	Kunde muss angemeldet sein
Auslösendes Ereignis	-
Ablauf	1. Kunde klickt auf "Guthaben kaufen" 2. Kunde gibt den Betrag an 3. Kunde bestätigt.
Alternativen	-
Nachbedingung Erfolg	Kunde bekommt Guthaben
Nachbedingung Fehlschlag	Fehlermeldung + Kunde bekommt kein Guthaben
Stakeholder	Kunde, Verein

UC-02-16 Guthaben aufladen	
Ziel	Kunde bekommt sein Guthaben aufgeladen
Akteure	Kunde, Reservierungssystem
Vorbedingung	Kunde muss Guthaben kaufen
Auslösendes Ereignis	Guthaben kaufen
Ablauf	1. Das System ladet dem gewählten Kunden Guthaben auf
Alternativen	-
Nachbedingung Erfolg	Das Guthaben wurde aufgeladen
Nachbedingung Fehlschlag	Fehlermeldung + Das Guthaben wurde nicht aufgeladen
Stakeholder	Kunde

UC-02-17 Admin kontaktieren	
Ziel	Der Kunde kontaktiert online den Admin
Akteure	Kunde
Vorbedingung	-
Auslösendes Ereignis	-
Ablauf	1. Kunde klickt auf "Admin kontaktieren" 2. Kunde füllt die Maske aus 3. Kunde klickt auf senden
Alternativen	-
Nachbedingung Erfolg	Admin bekommt eine Nachricht
Nachbedingung Fehlschlag	Nachricht wird nicht übermittelt
Stakeholder	Kunde, Admin

UC-02-18 Kundenrating erstellen	
Ziel	Kundenrating mit bestimmten Daten wird erstellt
Akteure	Reservierungssystem
Vorbedingung	-
Auslösendes Ereignis	-
Ablauf	1. Das Reservierungssystem nimmt vorher festgelegte Daten 2. und alle Kunden 3. und erstellt automatisch ein Rating.
Alternativen	-
Nachbedingung Erfolg	Kundenrating wird erstellt
Nachbedingung Fehlschlag	Fehlermeldung + Das Kundenrating wird nicht erstellt
Stakeholder	Admin



7.4 Reportverwaltung

UC-03-01 Auswertung wählen	
Ziel	Bestimmte Auswertung wird ausgewählt und angezeigt
Akteure	Admin
Vorbedingung	Admin muss eingeloggt sein
Auslösendes Ereignis	-
Ablauf	1. Admin klickt auf "Auswertung wählen" 2. Admin wählt einen Report aus
Alternativen	-
Nachbedingung Erfolg	Report wird angezeigt
Nachbedingung Fehlschlag	Fehlermeldung + Report wird nicht angezeigt

UC-03-02 Reports on demand generieren	
Ziel	Bestimmte Reports on demand werden generiert
Akteure	Admin
Vorbedingung	Admin muss eingeloggt sein
Auslösendes Ereignis	-
Ablauf	1. Admin klickt auf "Reports on demand generieren" 2. Admin wählt den gewünschten Report aus.
Alternativen	-
Nachbedingung Erfolg	Report wird generiert
Nachbedingung Fehlschlag	Fehlermeldung + Report wird nicht generiert

UC-03-03 Top-Tageszeiten	
Ziel	Die Top-Tageszeiten werden angezeigt
Akteure	Admin
Vorbedingung	Admin muss eingeloggt sein Reports on demand generieren
Auslösendes Ereignis	Reports on demand generieren
Ablauf	1. Admin wählt den Report "Top-Tageszeiten" aus
Alternativen	-
Nachbedingung Erfolg	Der Report "Top-Tageszeiten" wird angezeigt
Nachbedingung Fehlschlag	Fehlermeldung + kein Report wird angezeigt

UC-03-04 Auswärtige Kunden vs. Mitglieder	
Ziel	Der Report "Auswärtige Kunden vs. Mitglieder" wird angezeigt
Akteure	Admin
Vorbedingung	Admin muss eingeloggt sein Reports on demand generieren
Auslösendes Ereignis	Reports on demand generieren
Ablauf	1. Admin wählt den Report aus
Alternativen	-
Nachbedingung Erfolg	Der Report wird angezeigt
Nachbedingung Fehlschlag	Fehlermeldung + kein Report wird angezeigt

UC-03-05 Buchungsbelegung	
Ziel	Der Report "Buchungsbelegung" wird angezeigt
Akteure	Admin
Vorbedingung	Admin muss eingeloggt sein Reports on demand generieren
Auslösendes Ereignis	Reports on demand generieren
Ablauf	1. Admin wählt den Report aus
Alternativen	-
Nachbedingung Erfolg	Der Report wird angezeigt
Nachbedingung Fehlschlag	Fehlermeldung + kein Report wird angezeigt

UC-03-06 Meistgebuchte Wochen	
Ziel	Der Report "Meistgebuchte Wochen" wird angezeigt
Akteure	Admin
Vorbedingung	Admin muss eingeloggt sein Reports on demand generieren
Auslösendes Ereignis	Reports on demand generieren
Ablauf	1. Admin wählt den Report aus
Alternativen	-
Nachbedingung Erfolg	Der Report wird angezeigt
Nachbedingung Fehlschlag	Fehlermeldung + kein Report wird angezeigt

UC-03-07 Kundenliste	
Ziel	Der Report "Kundenliste" wird angezeigt
Akteure	Admin
Vorbedingung	Admin muss eingeloggt sein Reports on demand generieren
Auslösendes Ereignis	Reports on demand generieren
Ablauf	1. Admin wählt den Report aus
Alternativen	-
Nachbedingung Erfolg	Der Report wird angezeigt
Nachbedingung Fehlschlag	Fehlermeldung + kein Report wird angezeigt

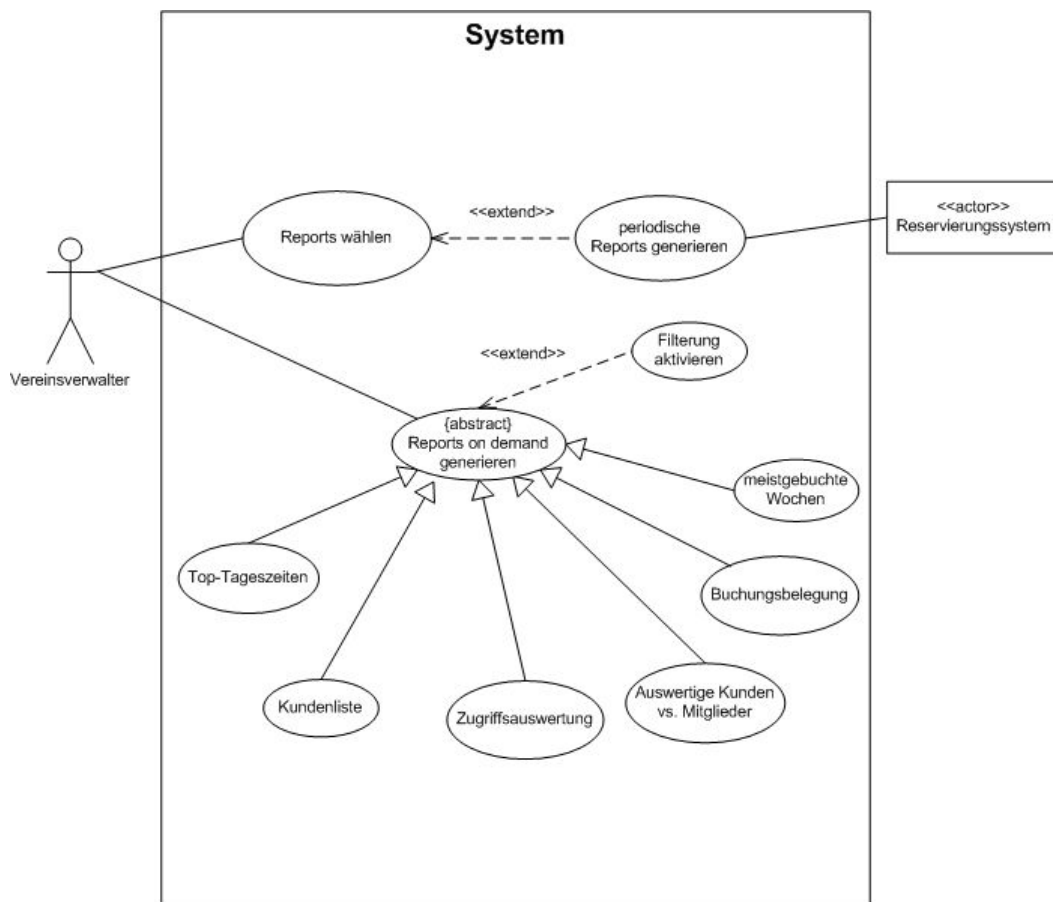
UC-03-08 Zugriffsauswertung	
Ziel	Der Report SZugriffsauswertung" wird angezeigt
Akteure	Admin
Vorbedingung	Admin muss eingeloggt sein Reports on demand generieren
Auslösendes Ereignis	Reports on demand generieren
Ablauf	1. Admin wählt den Report aus
Alternativen	-
Nachbedingung Erfolg	Der Report wird angezeigt
Nachbedingung Fehlschlag	Fehlermeldung + kein Report wird angezeigt

UC-03-09 Filterung aktivieren

Ziel	Filter für Reports sind aktiviert
Akteure	Admin
Vorbedingung	Admin muss eingeloggt sein Reports on demand generieren
Auslösendes Ereignis	Reports on demand generieren
Ablauf	1. Admin wählt den Filter für die Reports aus
Alternativen	-
Nachbedingung Erfolg	Der Filter filtert die Reports
Nachbedingung Fehlschlag	Fehlermeldung + Reports werden nicht gefiltert

UC-03-10 periodische Reports generieren

Ziel	periodische Reports werden von dem System generiert
Akteure	Reservierungssystem
Vorbedingung	-
Auslösendes Ereignis	-
Ablauf	1.Reservierungssystem generiert nach einer bestimmten Zeit automatisch Reports
Alternativen	-
Nachbedingung Erfolg	Reports werden generiert
Nachbedingung Fehlschlag	Fehlermeldung + keine Reports wurden generiert



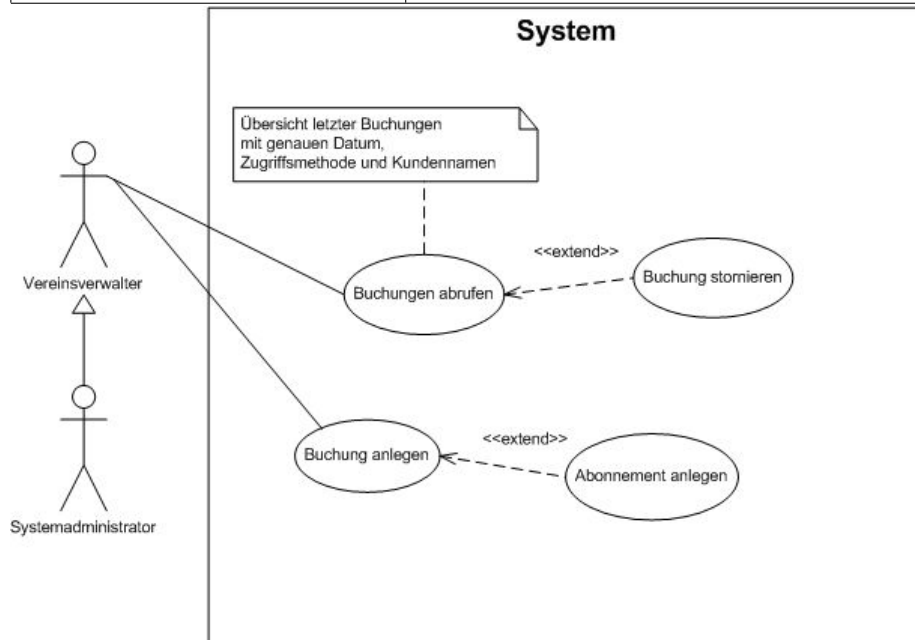
7.5 Buchungsverwaltung

UC-04-01 Buchungen abrufen	
Ziel	Übersicht aller Buchungen
Akteure	Admin, Vereinsverwalter
Vorbedingung	Admin muss angemeldet sein
Auslösendes Ereignis	-
Ablauf	1. Admin klickt auf den Button "Buchungsübersicht"
Alternativen	-
Nachbedingung Erfolg	Alle Buchungen mit genauen Datum und Kundennamen werden dargestellt
Nachbedingung Fehlschlag	-
Stakeholder	Admin

UC-04-02 Buchungen stornieren	
Ziel	Buchung aus der Datenbank entfernen
Akteure	Vereinsverwalter, Admin
Vorbedingung	1. Akteur muss angemeldet sein 2. Akteur muss die Buchungsübersicht aufgerufen haben
Auslösendes Ereignis	-
Ablauf	1. Akteur klickt auf den Button "Buchung löschen" neben der jeweiligen Buchung 2. Akteur bestätigt die Löschung
Alternativen	Akteur bestätigt nicht
Nachbedingung Erfolg	Buchung wird aus dem System entfernt
Nachbedingung Fehlschlag	1. Buchung wurde schon gelöscht 2. Entsprechende Meldung erscheint
Stakeholder	Admin, Vereinsverwalter

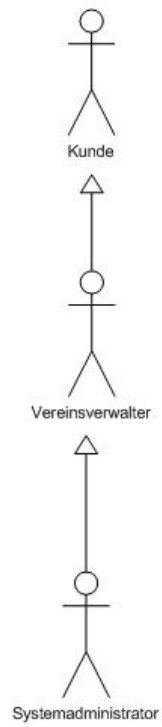
UC-04-03 Buchung anlegen	
Ziel	Buchung in Datenbank abspeichern
Akteure	Admin, Vereinsverwalter
Vorbedingung	1. Akteur muss angemeldet sein
Auslösendes Ereignis	-
Ablauf	1. Akteur klickt auf den Button "Buchung anlegen" 2. Akteur trägt in Buchungsmaske alle relevanten Daten der Buchung ein 3. Akteur klickt auf speichern
Alternativen	Akteur klickt auf Abbrechen
Nachbedingung Erfolg	Buchung wird in die Datenbank für den jeweiligen Platz gespeichert
Nachbedingung Fehlschlag	Buchung wird nicht gespeichert
Stakeholder	Admin

UC-04-04 Abonnement anlegen	
Ziel	Abonnement anlegen
Akteure	Vereinsverwalter, Admin
Vorbedingung	1. Akteur muss angemeldet sein 2. Akteur muss eine Buchung anlegen
Auslösendes Ereignis	-
Ablauf	1. Akteur klickt auf den Button "Buchung anlegen" 2. Akteur kann nun ein Abonnement anlegen. 3. Akteur füllt die Maske aus. 4. Akteur bestätigt
Alternativen	-
Nachbedingung Erfolg	Alle Buchungen seines Vereins mit genauen Datum und Kundennamen werden dargestellt
Nachbedingung Fehlschlag	-
Stakeholder	Admin



7.6 Actor hierarchy

Actor hierarchy



Kapitel 8

Nichtfunktionale Anforderungen

- **Usability**

Die Applikation soll so einfach und intuitiv wie möglich gestaltet werden, um auch unerfahrenen Benutzern eine uneingeschränkte Verwendung der Software zu ermöglichen. Hinweis- und Fehlermeldungen müssen aussagekräftig sein, so dass die Benutzer entweder selbst das Problem lösen oder aber konkrete Anfragen an den Systemadministrator stellen können. Die Usability des Systems wird mittels eines einwöchigen Testbetriebes sichergestellt. Benutzer von Tenez werden passiv beobachtet.

- **Korrektheit**

In den mathematischen Berechnungen muss sichergestellt werden, dass diese 100% fehlerfrei sind, da diese die Grundlagen für das Berichtswesen sind. Berechnungen werden auf die zweite Kommastelle genau berechnet. Um diese Anforderung zu garantieren werden spezielle Unit Tests dafür gemacht.

- **Laufzeitumgebung**

Die Applikation soll möglichst behutsam mit Speicher- und CPU-Auslastung umgehen, da eine Installation auf einem V-Server möglich sein muss. In der finalen Konfiguration sollte das System im Betrieb nie mehr als 96 Megabyte Speicher pro Vereinsinstanz belegen, ideal wäre ein Speicherverbrauch unter diesem Wert.

Kapitel 9

Lieferumfang und Abnahme

9.1 Lieferumfang

Dem Kunden wird folgendes übermittelt:

- Die Software wird in zwei Releases ausgeliefert. Die erste Version (0.1) umfasst das Layout, Benutzer Registrierung mit Login, Vereine anlegen/editieren und das Ansehen der Buchungsübersicht. Am Ende des Projektes soll dem Kunden eine vollfunktionsfähige Software der Version 1.0 übermittelt werden.
- Benutzerhandbuch
- Anforderungsdokumente

Weitere Komponenten, die nicht im Lieferumfang sind:

- Dokumente, welche zur Erstellung der Software benötigt wurden (UML-Diagramme etc.)
- Tools, die zur Erstellung der Software notwendig waren (Eclipse etc.)
- Teaminterne Dokumente, wie Arbeitslisten und PM Pläne

9.2 Abnahme

Der Abnahmetest wird in zwei Stufen durchgeführt. Da der Kunde zu Beginn möglicherweise noch nicht in der Lage ist das System auf auftretende Fehler zu überprüfen, wird als erster Schritt eine gemeinsame Prüfung von Kunde und Projektteam herangezogen. Sollte diese erfolgreich sein, beginnt der Kunde mit der eigenen Prüfung im Rahmen eines Probetriebs. Getestet werden alle funktionalen Anforderungen, wobei der Abnahmetest nicht länger als 3 Stunden dauern darf. Der Abnahmeort wird vom Kunden bestimmt.

Kapitel 10

Arbeitsstruktur & Rollenverteilung

Rolle	Name	Matr.Nr.	KZ	E-Mail
TK	Dominik Gruber	0926288	937	dominik.gruber@tenez.at
TA	Philipp Naderer	0625238	937	philipp.naderer@tenez.at
TA Stv	Manuel Mayrhofer	0625013	937	manuel.mayrhofer@tenez.at
Doku	Manuel Djalili	0525999	926	manuel.djalili@gmx.at
Test	Alexander Bohn	0505808	926	jagang@gmx.at

Kapitel 11

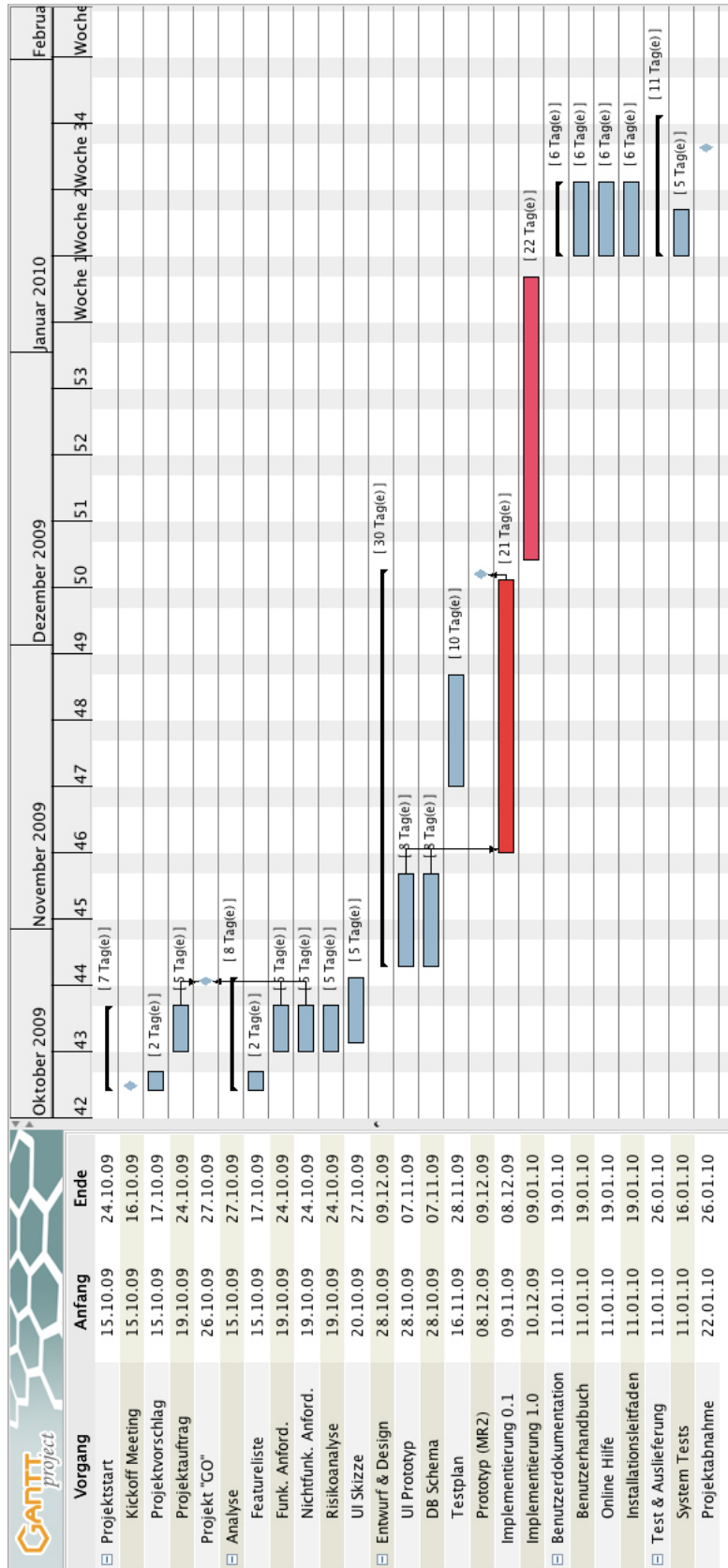
Projektplan

Phase / Aufgabe	Termin	Aufwand
Projektstart	Bis 15.10.2009	10h
Entwurf/Anforderungen	Bis 25.10.2009	50h
Entwurf/Analyse und Design	Bis 08.11.2009	100h
Implementation/Alphaversion herstellen	Bis 03.12.2009	200h
Version 0.1 abgeschlossen	Bis 08.12.2009	-h
Implementation/Erstellen der Applikation und Durchführung der Modultests	Bis 10.01.2010	250h
Implementation/Durchführen der Systemtests	Bis 17.01.2010	60h
Auslieferung/Installationshandbuch zusammenstellen, Feinschliff der Applikation und Beheben von Fehler	Bis 19.01.2010	60h
Projektabschluss	24.01.2010	20h
Gesamtaufwand Version 1.0		750h

11.1 Iceberg List

Modul	Feature Name	Prio.	Zeit	Release
1.1	HTML, CSS, Client-JS	H	45h	0.1
1.2	Macros & Filter	H	15h	0.1
1.3	Browser Testing & Browser Bug Fixing	H	10h	0.1
2.1	Vereine anlegen/editieren	H	20h	0.1
3.1	Buchungsübersicht ansehen	H	15h	0.1
5.1	User Registrierung	H	25h	0.1
5.2	User Login	H	15h	0.1
11.1	Modul Tests	H	-h	0.1
11.2	Unit Tests	H	-h	0.1
1.2	Mobile Version	M	35h	1.0
2.2	Für Vereine Plätze mit Timeslots anlegen/editieren	H	45h	1.0
2.3	Skins für Vereine	M	15h	1.0
4.1	Automatische Auswertungen	H	95h	1.0
5.3	Jetonstand abfragen	H	5h	1.0
5.4	User Stammdaten verwalten	H	15h	1.0
5.5	User Stammdaten exportieren	L	10h	1.0
5.6	Newsletter an User	M	15h	1.0
6.1	Reservierung vornehmen	H	20h	1.0
6.2	Reservierung stornieren	H	5h	1.0
8.1	Kontakt Administrator	M	5h	1.0
9.1	Logging (Zugriffe Vereinsadministratoren etc.)	L	20h	1.0
10.1	Import von bestehenden Daten	M	20h	1.0
7.1	Online-Überweisung	N	150h	2.0

11.2 Projektstrukturplan



Kapitel 12

Risikoanalyse

12.1 Übersicht

ID	Datum	Bezeichnung	Typ	Ausw.	Wahrscheinl.	Schwere	Verantwortlich
1	30.10.09	Bestehende Daten	D	5	10%	0,5	DG
2	30.10.09	Usability Test	D	4	10%	0,4	BA
3	19.10.09	SVN Probleme	I	1	70%	0,7	*
4	19.10.09	Personalausfall	D	5	10%	0,5	DG
5	19.10.09	HW-Ausfall	I	2	10%	0,2	*
6	19.10.09	Benutzerschnittstelle	D	3	50%	1,5	DG
7	19.10.09	Doku-Inkonsistenzen	D	2	60%	1,2	DM
8	19.10.09	Know How im Team	D	2	90%	1,8	NP,MM
9	30.10.09	Helma Ablehnung	I	5	20%	1,0	MM
10	19.10.09	Risiko Management	D	4	70%	2,8	MM
11	19.10.09	System Architektur	D	4	30%	1,2	NP
12	19.10.09	Testfälle	D	4	70%	2,8	BA
Summe						14,6	

Zeichenerklärung: TYP (I:indirekt, D:direkt), AUSWIRKUNG (1:WENIG – 5:STARK), WAHRSCHEINLICHKEIT (0:KEINE – 100:HÖCHSTE) SCHWERE (berechnet sich aus vorangehenden Parametern, Produkt aus Auswirkung und Wahrscheinlichkeit, definiert somit Wichtigkeit für Projekt)

12.2 Beschreibung

ID	Beschreibung	Gegenstrategie
1	Best. Daten müssen zu Verfügung gestellt werden	Importfkt. wird nicht gemacht
2	Zugang zur Halle muss vorhanden sein	Usability Test wird nicht gemacht
3	Unerfahrenheit mit Google Code	zusätzl. mündl. Abgleich
4	Austritt einzelner Team Mitglieder	Wissensmngt. unter d. Mitgliedern
5	Ausfall von Workstations (Datenverlust)	regelmäßiges Comitten
6	Entw. des Sys. mit variierenden Benutzerwünschen	Kontakt mit Stakeholdern
7	Aufteilung d. Arbeitsaufg. vs Abhängigkeit d. Doku	Komm. innerhalb d. Teams
8	Mangel an fachspezifischen Wissen - Helma	Gegenseitige Hilfestellungen
9	Helma wird abgelehnt	Tutorials, Aufzeigen der Vorteile
10	Risiken werden zu spät/ nicht erkannt	Fortwährendes Risikomanagement
11	Probleme mit einzelnen SW Komponenten	Helma-Group anfragen, Meetings
12	Auswirkungen von fehlschlagenden Testfällen	Gruppeninternes Review

Kapitel 13

Informationswesen

13.1 Teaminterne Kommunikation

Alle internen Diskussionen werden über eine Mailingliste abgewickelt. Kontakt zum Kunden nehmen entweder Dominik Gruber oder Philipp Naderer auf, Ansprechpersonen für den Kunden sind ebenfalls sie.

13.2 SVN-Repository

Als Versionskontrollsystem wird SVN gewählt. Da das Projekt unter der Apache 2.0 Lizenz steht, bietet sich ein Hosting durch Google Code an. Unter der URL <http://code.google.com/p/tenez> können alle Projektdateien abgerufen werden. Es sind nur Mitglieder vom ASE-Projekt auch Mitglieder im Google Code Projekt.

13.3 Projektdokumentation

Sämtliche Dokumente werden mit LaTeX erstellt. Als Zeichensatz kommt UTF-8 zum Einsatz, um Konflikte zwischen Mac- und PC-Benutzern zu verhindern. Dokumente werden im SVN-Repository unter dem Pfad `/ase-docs` verwaltet. Zu jedem fertigen Dokument wird immer ein entsprechendes PDF erstellt. Änderungen können so in der Timeline mitverfolgt werden. Jedes Projektdokument ist in Deutsch zu halten.

13.4 Quelltexte

Alles, was sich nicht unter `/ase-docs` befindet, ist in Englisch zu verfassen. Dies umfasst alle Quelltexte und Property-Dateien. Kommentare im Quelltext sind ebenfalls Englisch! Als Zeichensatz für alle Textdateien kommt UTF-8 zum Einsatz.

Versionsprotokoll

v2.1	17.11.2009	Dominik Gruber	Iceberglist sortiert
v2.0	01.11.2009	Philipp Naderer Manuel Mayrhofer	Finalisierung für Abgabeverision
v1.6	31.10.2009	Philipp Naderer	Iceberglist, div. Korrekturen
v1.5	31.10.2009	Manuel Mayrhofer	Projektplan
v1.4	31.10.2009	Alexander Bohn Manuel Djalili	UseCase Beschreibungen & UseCase Liste
v1.4	30.10.2009	Manuel Mayrhofer	Risikoanalyse
v1.2	30.10.2009	Dominik Gruber Philipp Naderer	Endkorrektur
v1.1	26.10.2009	Alexander Bohn	Use Case Beschreibungen
v1.0	24.10.2009	Philipp Naderer	Finalisierung für Abgabe
v0.12	24.10.2009	Alexander Bohn	Use Case Diagramme
v0.11	24.10.2009	Dominik Gruber	Diverse Anpassungen
v0.10	24.10.2009	Manuel Djalili	Featureliste erweitert
v0.9	24.10.2009	Philipp Naderer	Komponentendiagramm
v0.8	24.10.2009	Manuel Mayrhofer	Gantt Diagramm
v0.7	24.10.2009	Philipp Naderer	Diverse Korrekturen,
v0.6	23.10.2009	Dominik Gruber	Projektplan, Iceberg List
v0.5	22.10.2009	Manuel Djalili	Rollenverteilung
v0.4	20.10.2009	Manuel Mayrhofer	Lieferumfang, Abnahme
v0.3	19.10.2009	Philipp Naderer	Funktionale Anforderung, Komponentendiagramm
v0.2	19.10.2009	Manuel Mayrhofer	Nichtfunktionale Anforderungen
v0.1	19.10.2009	Dominik Gruber	Erstellung (Template, Datenübernahme)