

Rocket Alarm

Machbarkeitsstudie

	Name	Datum	Unterschrift
Erstellt	Christoph Kern	23.02.2017	<i>Kern Christoph</i>
Erstellt	Salih Alesevic	23.02.2017	<i>Salih Alesevic</i>
Erstellt	Vincent Schwarz	23.02.2017	<i>Vincent Schwartz</i>
Geprüft			
abgenommen			

Version	Autor	QS	Datum	Status	Kommentar
1.0	Christoph Kern, Salih Alesevic, Vincent Schwarz		20.02.2017	Endversion	Finaler Entwurf

Inhaltsverzeichnis

Einführung	3
1 Projektdaten	3
1.1 Projektbeschreibungen	3
2 Voruntersuchung des Produkts	3
2.1 IstZustand.....	3
3 Produktauswahl.....	3
3.1 Trendanalyse	3
3.2 Marktanalyse.....	3
4 Soll-Zustand.....	4
4.1 Muss-Ziele	4
4.2 Kann-Ziele	4
4.3 Nicht-Ziele	5
5 Produktfunktionen.....	5
5.1 Benutzerverwaltung	5
5.2 Freundesliste verwalten.....	7
5.3 Abstimmung verwalten	12
5.4 Statistik verwalten.....	15
5.5 Einstellungen ändern.....	15-19
5.6 Aktivitätsdiagramm.....	19-23
6 Technische Machbarkeit	23
6.1 Technologien.....	23
6.2 Umsetzung	23
7 Wirtschaftliche Machbarkeit.....	24
7.1 Personalaufwand	24
7.2 Investitionsaufwand	24
7.3 Nutzen	21
7.4 Risikoanalyse	24
8 Persönliche Machbarkeit.....	25
8.1 Nutzwertanalyse.....	25
9 Projektorganisation	25
10 Projektplanung	26
11 Management Summary:	26

Einführung

Es soll ein Spiel entwickelt werden, in dem man ein Flugobjekt navigiert. Ziel des Spiels ist es auf einen zukommende Raketen auszuweichen. Es gibt unterschiedliche Schwierigkeitsstufen, die man aber nicht selbst auswählen kann. Die Schwierigkeit ändert sich von alleine mit der Zeit, desto länger man am Leben ist, desto schwieriger wird es den Raketen auszuweichen. Es gibt kein vordefiniertes Ende, das Spiel ist zu Ende sobald man 0% Lebenspunkte hat.

1. Projektdaten

1.1 Projektbeschreibung

Im Rahmen des Projekts Rocket Alarm soll ein Spiel entwickelt werden, in dem man mit einem Helikopter auf einen zukommende Raketen ausweichen kann. Wenn man von einer Rakete getroffen wird verliert man 25% seiner Lebenspunkte und wenn man 4 mal getroffen wird ist das Spiel vorbei. Man kann einen Highscore erreichen der sich nach beenden eines Spieles speichert. Der Highscore ergibt sich durch die Zeit die man Überlebt hat. Je länger man spielt, desto schwieriger wird das Spiel und desto mehr Punkte bekommt man.

2. Voruntersuchung des Produkts

2.1 Ist-Zustand

Momentan gibt es kein Spiel das es ermöglicht als Helikopter von der Seite kommenden Raketen auszuweichen und somit seine Reaktionszeit zu steigern.

3. Produktauswahl

3.1 Trendanalyse

In der Zeit in der wir gerade Leben sind Computer- und Handyspiele einfach überall aufzufinden. Ob am Handy in der U-Bahn oder zu Hause am Computer Spiele werden überall benutzt.

3.2 Marktanalyse

Es gibt bereits sehr viele Spiele auf dem Markt mit mehreren Inhalten. Doch das Spiel Rocket Alarm ist einzigartig und so auf dem Markt noch nicht vorhanden. Wir sind die einzigen die ein Spiel anbieten in dem man einen Helikopter durch Raketen steuern kann um zu überleben.

4. Soll-Zustand

4.1 Muss-Ziele

1. Spielespaß

Durch eine schöne 2D Oberfläche mit einem Selbst programmierten Algorithmus wird es möglich auf spielerische Weise Raketen von der Seite aus zu weichen und da es Highscores gibt auch ein gewisser Drang das Spiel zu spielen und immer besser zu werden.

2. Funktionsfähigkeit des Spiels

Der Hubschrauber soll fähig sein durch ein 2D Feld zu fliegen und von der Seite kommenden Raketen aus zu weichen. Weiterst sollen die Raketen je nach Spieldauer immer schneller auf den Helikopter zu fliegen und dem Benutzer somit das Spiel erschweren.

4.2 Kann-Ziele

1. Multi-Player

Neben dem Single-Player Spielmodus kann auch noch ein Multi-Player Spielemodus eingebaut werden indem man gegen einen anderen Spieler spielen kann und der mit dem Höherem Highscore gewinnt oder, dass einer der Spieler die Raketen rauf und runter steuern kann.

1. Competitive gameplay

Neben den Muss-Zielen kann man eine Rangliste der Spieler erstellen in der es Ränge von Bronze-Diamant bekommt und je nach Highscore-Punktezahl kommt man in eine bestimmte Rangliste. Wenn man in Diamant ist gibt ist der Spieler mit den meisten Punkten der beste Spieler.

4.3 Nicht-Ziele

1. Schießen des Helikopter

Der Fliegende Helikopter soll nicht fähig sein die Raketen durch Schüsse zu zerstören oder durch das Abschießen von sonstigen Objekte einen höheren Highscore zu bekommen.

2. Ende des Spiels

Da der Highscore durch die Dauer des Spieles entsteht, soll es kein Ende des Spiels geben.

Das Spiel endet nur durch das Versagen von dem Spieler indem er auf 0 Leben kommt.

5 Produktfunktionen

5.1 Benutzerverwaltung

5.1.1 Benutzer registrieren (/LF0010/)

Jeder Benutzer kann sich selbständig in dem Spiel registrieren. Das Programm benötigt dafür die Angaben eines Benutzernamens und seines Passwortes. Danach hat er sofort Zugriff auf alle Funktionen des Spiels.

Funktion		Nutzen	Aufwand	Must Have Should Have Nice to Have
UseCase		hoch	mittel	MH
Name	Benutzer registrieren (/LF0010/)			
Art	Anwendungsfall			
Kurzbeschreibung	Jeder Benutzer kann sich selbständig am System registrieren. Das Programm benötigt dafür die Angaben eines Benutzernamens und seines Passwortes. Danach hat er sofort Zugriff auf alle Funktionen des Systems.			
Auslöser	Benutzer möchte sich bei Software registrieren			
Ergebnis	Benutzer hat ein eigenes Konto			
Akteure	neuer Benutzer			
Eingehende Informationen	Username, Passwort			
Vorbedingungen	Benutzer darf noch nicht am System registriert sein			
Nachbedingung	Benutzer kann sich bei der Software anmelden			

5.1.2 Benutzer anmelden (/LF0020/)

Bevor sich ein Benutzer anmelden kann, muss er sich am System registriert haben. Bei der Anmeldung verlangt das Programm die Angabe des Benutzernamens und des dazugehörigen Passworts.

Funktion		Nutzen	Aufwand	Must Have Should Have Nice to Have
UseCase <div> <div>Benutzer anmelden (/LF0020/)</div> <div>Name</div> </div> <div> <div>Anwendungsfall</div> <div>Art</div> </div> <div> <div>Jeder registrierte Benutzer kann sich jederzeit am System anmelden.</div> <div>Kurzbeschreibung</div> </div> <div> <div>Benutzer möchte auf Funktionen des Programms zugreifen.</div> <div>Auslöser</div> </div> <div> <div>Benutzer kann auf Funktionen des Systems zugreifen.</div> <div>Ergebnis</div> </div> <div> <div>Benutzer</div> <div>Akteure</div> </div>		hoch	gering	MH

<div><div><div>Username, Passwort</div><div>Eingehende Informationen</div></div><div>Vorbedingungen</div></div>				
<div><div><div>Nachbedingung</div><div>Benutzer kann die Funktionen des Systems nutzen</div></div></div>				

5.1.3 Benutzer abmelden (/LF0030/)

Das manuelle Abmelden vom System ist jederzeit möglich.

Funktion	Nutzen	Aufwand	Must Have Should Have Nice Have to
<p>UseCase</p> <p>Benutzer abmelden (/LF0030/)</p> <p>Name</p> <p>Anwendungsfall</p> <p>Art</p> <p>Kurzbeschreibung Jeder Benutzer kann sich jederzeit vom System abmelden.</p> <p>Auslöser Benutzer braucht die Funktionen</p> <p>Ergebnis Benutzer kann nicht mehr auf die Funktionen des Systems zugreifen.</p> <p>Akteure Benutzer</p>	hoch	gering	MH

Eingehende Informationen	keine			
Vorbedingungen	Benutzer muss am System angemeldet sein.			
Nachbedingung	Benutzer kann sich jederzeit wieder am System anmelden.			

5.2 Freundesliste verwalten

5.2.1 Freund hinzufügen (/LF0110/)

Der angemeldete Benutzer kann Freunde in seiner Freundesliste hinzufügen. Dafür muss der Benutzername des Freundes eingegeben werden.

Funktion		Nutzen	Aufwand	Must Have Should Have Nice to Have
UseCase <div>Freund hinzufügen (/LF0110/)</div> Name <div>Anwendungsfall</div> Art <div> Kurzbeschreibung Es kann jederzeit ein Freund hinzugefügt werden. </div> <div> Auslöser Benutzer möchte jemandem eine Abstimmung schicken. </div> <div> Ergebnis Freund befindet sich in der Freundesliste des Benutzers. </div> <div> Akteure Benutzer, Freund </div>		hoch	mittel	SH

Username des Freundes		Vord						
Eingehende Informationen								
Nachbedingung		Freund kann eine Abstimmung gesendet werden						

5.2.2 Freund entfernen (/LF0120/)

Der angemeldete Benutzer kann Freunde aus der Freundesliste entfernen.

Funktion	Nutzen	Aufwand	Must Have Should Have Nice Have to

UseCase	hoch	gering	SH
<div><div>Name</div><div>Freund entfernen (/LF0120/)</div></div>			
<div><div>Art</div><div>Anwendungsfall</div></div>			
<div><div>Kurzbeschreibung</div><div>Es kann ein Freund, der sich in der Freundesliste befindet, entfernt werden.</div></div>			
<div><div>Auslöser</div><div>Benutzer möchte einen Freund aus seiner Freundesliste entfernen</div></div>			
<div><div>Ergebnis</div><div>Freund befindet sich nicht mehr in der Freundesliste des Benutzers</div></div>			
<div><div>Akteure</div><div>Benutzer, Freund</div></div>			

Eingehende Informationen	Username des Freundes
Vorbedingungen	Freund befindet sich in der Freundesliste des Benutzers
Nachbedingung	Keine

5.5 Einstellungen ändern

5.5.1 Hilfe anzeigen (/LF0410/)

Der angemeldete Benutzer kann ein Hilfefenster öffnen, in dem die wichtigsten Funktionen kurz beschrieben werden. Weiteres sind auch die Richtlinien des Umgangs mit den Abstimmungen einzulesen.

Funktion	Nutzen	Aufwand	Must Have Should Have Nice to Have

UseCase		hoch	mittel	MH
Name Art Kurzbeschreibung Auslöser Ergebnis Akteure	Hilfe anzeigen (/LF0410/)			
	Anwendungsfall			
	Die Hilfe (wie die Software bedient wird) und die Richtlinien (an die sich die Benutzer halten müssen) der Software werden aufgerufen.			
	Der Benutzer verwendet die Software das erste Mal oder möchte die Richtlinien überprüfen.			
	Ein Hilfefenster befindet sich im Vordergrund der Applikation.			
	Benutzer			

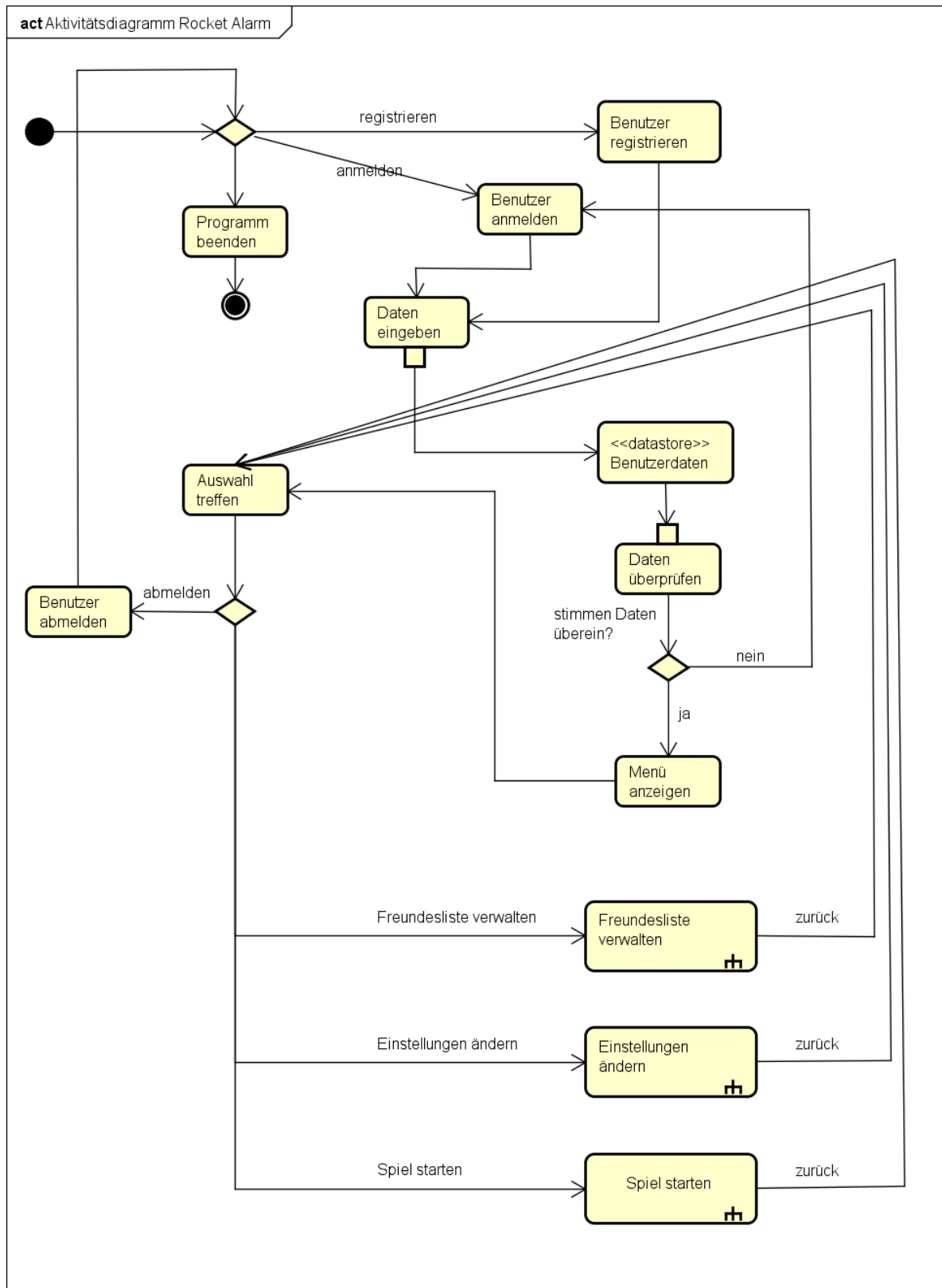
5.5.3 Passwort ändern (/LF0430/)

Der angemeldete Benutzer hat die Möglichkeit, sein Passwort unter Angabe seines alten Passwortes, seines neuen Passwortes und einer Bestätigung seines neuen Passwortes (erneute Eingabe) zu ändern.

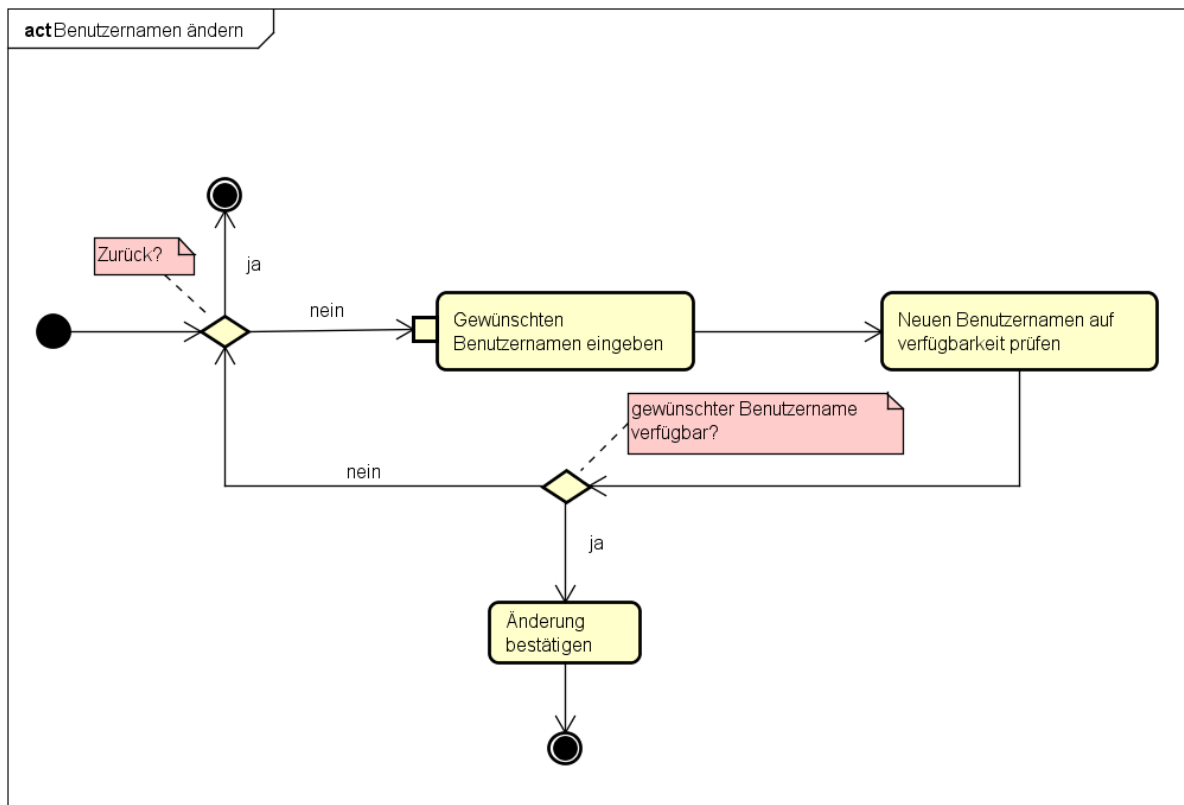
Funktion	Nutzen	Aufwand	Must Have Should Have Nice Have to
<p>UseCase</p> <p>Passwort ändern (/LF0430/)</p> <p>Name</p> <p>Anwendungsfall</p> <p>Art</p> <p>Kurzbeschreibung Das Passwort des Benutzers wird durch ein neues ersetzt.</p> <p>Auslöser Das Passwort des Benutzers soll durch ein neues ersetzt werden.</p>	hoch	hoch	MH

<div><div>Ergebnis</div><div>Der Benutzer kann sich mit einem neuen Passwort einloggen.</div></div>	<div>Akteure</div>			
<div><div>Eingehende Informationen</div><div>altes Passwort, neues Passwort, Bestätigung des neuen Passwortes</div></div>				
<div><div>Vorbedingungen</div><div>Benutzer hat sich angemeldet</div></div>				
<div><div>Nachbedingung</div><div>Benutzer hat sich mit neuem Passwort eingeloggt.</div></div>				

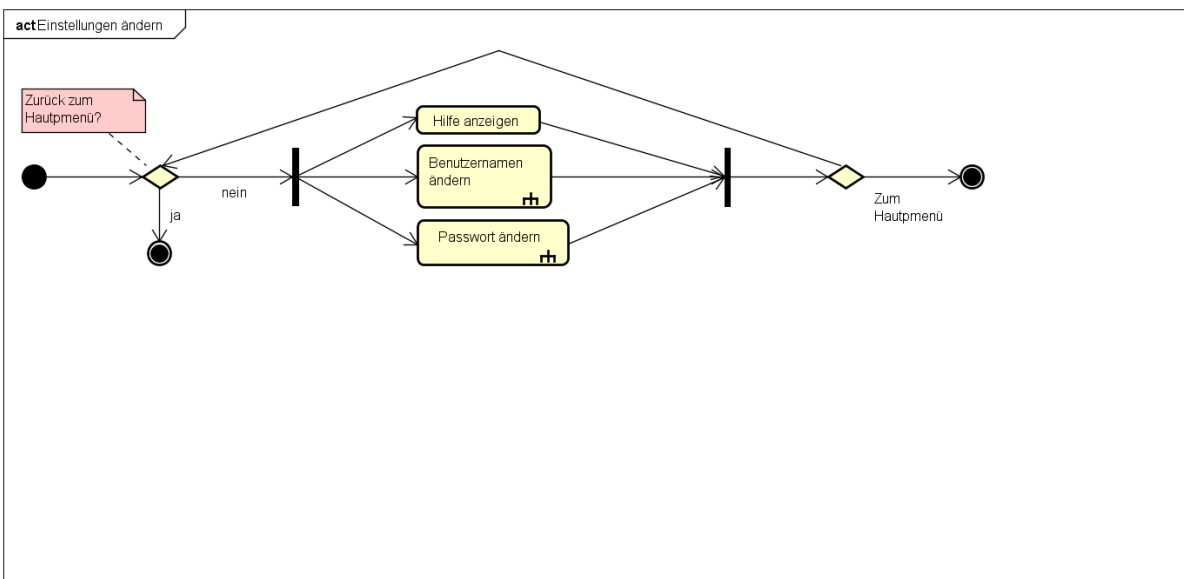
5.6.1 Aktivitätsdiagramm



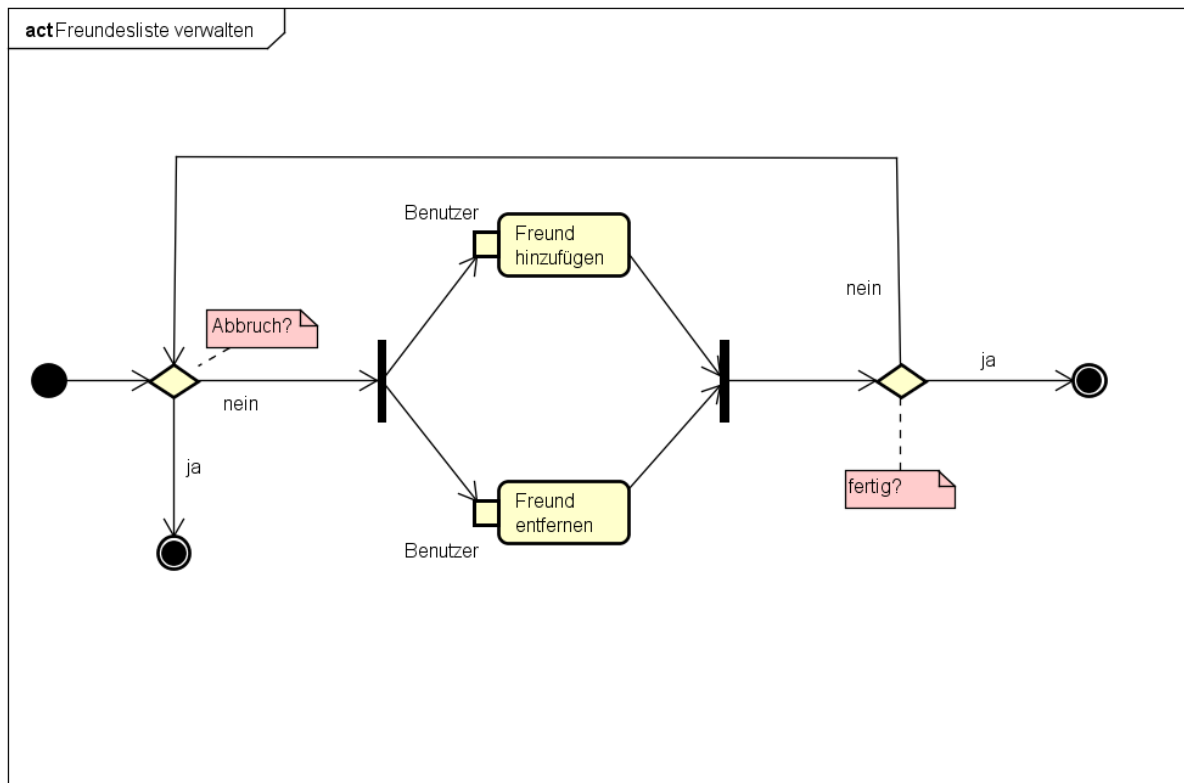
5.6.2 Benutzernamen ändern



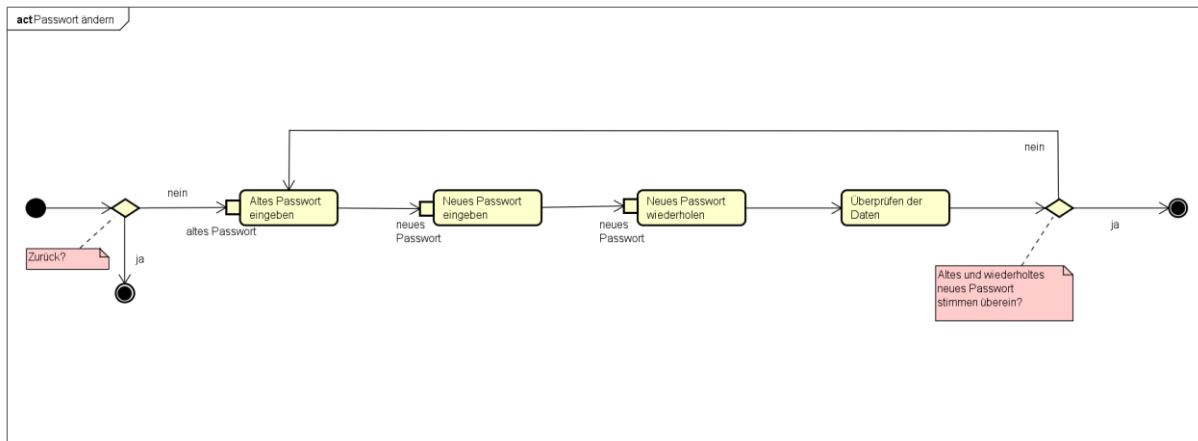
5.6.3 Einstellungen ändern



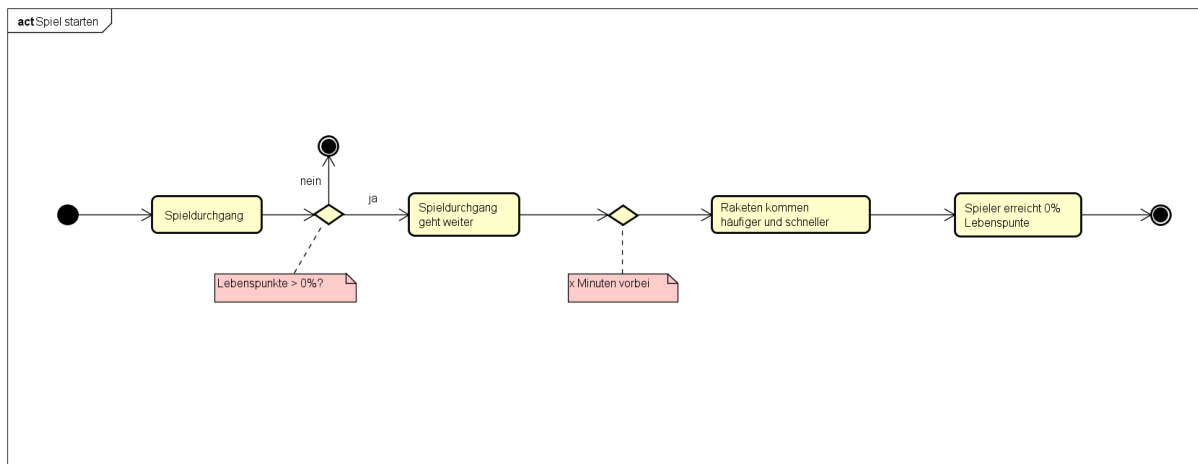
5.6.4 Freundesliste Verwalten



5.6.5 Passwort ändern



5.6.6 Spiel starten



6 Technische Machbarkeit

6.1 Technologien

Es gibt manche Risiken, die folgendermaßen gelöst werden können. Die Benutzerdaten werden an einem Server gespeichert und direkt an die betroffenen Endgeräte (Handy, Tablet etc.) gesendet, sobald sich diese mit dem Server verbinden. Eine Verbindung wird mittels asymmetrischer Verschlüsselungsmethoden hergestellt, um eine etwas sichere Sicherheit zu haben/bekommen. Wenn keine Verbindung vom Endgerät zum Server vorhanden ist (Endgerät oder Server offline) werden die Daten automatisch am Endgerät gespeichert.

6.2 Umsetzung

Weil es in der Marktwirtschaft nur ein paar entsprechende Lösungsarten gibt, ist dieses Produkt mit jeden seiner Funktionen fast einzigartig.

Das Ganze wird in der Programmiersprache Java entwickelt werden. Der eine Vorteil liegt da, dass das Team die meiste Kenntnisse in dieser Sprache erkennen lässt und somit wenig Arbeitszeit benötigt wird. Der eventuell zweite Vorteil ist die Möglichkeit auf Inhalte andersartige Dateien zugreifen zu können. Möglich ist auch, dass bei dem Ergebnis unübersichtliche Vorgänge eingesetzt werden müssen, was eine Abschwächung vom Ganzen bewirken würde. Das große Bedürfnis wird sein, dass die Leistung des Smartphones ausreichen wird.

7 Wirtschaftliche Machbarkeit

7.1 Personalaufwand

Der Umfang der Software ist nicht sehr groß, trotzdem versuchen alle Projektteammitglieder an diesem Werk mitzuarbeiten, um eine effiziente Entwicklung zu ermöglichen. Die Gesamtstundenzahl, die die Entwicklung des Systems fordern werden ist uns derzeit noch nicht bekannt. In einer ExcelDatei wird unsere Stundenanzahl notiert, damit wir sehen, wer wieviel an was gearbeitet hat.

7.2 Investitionsaufwand

Es gibt keine Investitionsaufwände, da das Projekt im Bundesland Wien stattfindet, wo auch der Standort aller Teammitglieder ist. Die komplette Hardware ist vorhanden und gewährleistet erfolgreiches Arbeiten. Die benötigte Software ist ebenfalls vorhanden und fordert keine zusätzlichen Kosten.

7.3 Nutzen

Das Spiel ist primär für Kinder und Jugendliche gedacht. Es ist jedoch nicht ausgeschlossen, dass das Spiel weltweit verbreitet wird. Mithilfe vom Feedback wird entschieden, ob das Spiel öffentlich zugänglich sein wird.

7.4 Risikoanalyse

Bei diesem Projekt sind Risiken, wie bei jedem anderen Projekt auch, nicht auszuschließen. Einer dieser Faktoren könnte eventuell eine längere, gesundheitlich bedingte Abwesenheit sein. Je nach Zustand der erkrankten Person wird entschieden, ob die übrigen Teammitglieder den Teil übernehmen, oder ob die betroffene Person seinen Teil während oder nach dem Krankenstand erbringt. Ein kritischer Faktor bei der Umsetzung der Projektziele kann die Teamarbeit werden. Es ist wichtig, dass das Arbeitsklima unter den Mitarbeitern im Einklang ist.

Ein Problem, was jederzeit eintreten kann ist ein Virus auf den Arbeitsgeräten, Das ist leider nicht vorhersehbar und kann den Abgabetermin hinauszögern. Ein weiteres Problem, was ebenfalls jederzeit auftreten kann ist, dass ein Arbeitsgerät betriebsunfähig wird. In diesem Fall muss sich der betroffene so schnell wie möglich darum kümmern, dass er ein Ersatzgerät findet, da die Reparatur meist Wochen dauern kann und uns die Zeit fehlt den Abschluss um Wochen hinauszuzögern.

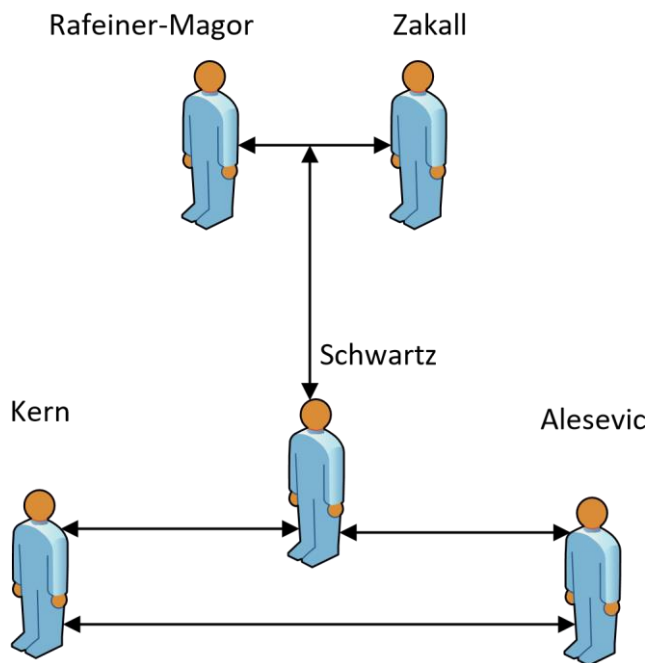
		Gewichtung in %	Linux		Windows 7/8/10		Mac OS	
			Rang	G*R	Rang	G*R	Rang	G*R
Kosten	Sprachlizenzen	20	4	60	1	50	5	30
	Entwicklungs- Umgebungslicenzen	5	2	10	1	10	3	8
	gesamt	25	6	70	2	60	8	38
Vebreitung	Sprachkenntnisse	15	3,5	52,5	2	15	1	15
	Akzeptanz	10	2,5	25	3	15	3	10
	gesamt	25	6	77,5	5	30	4	25
Objekt- orientierung	Umsetzung der OOP-Konzepte	10	1	10	1	40	2	35
	Akzeptanz	5	2	10	1,5	20	2	20
	gesamt	15	3	20	2,5	60	4	55
Einfachheit	Syntax	12	2,5	30	2	40	1	40
	Strukturierung	8	2,5	20	1	20	4	20
	gesamt	20	5	50	3	60	5	55
Vorgefertig- te API	Umfang	8	3,5	28	2	50	2	45
	Qualität	4	3	12	1,5	50	2	40
	Dokumentation	3	2,5	7,5	1	20	1	15
	gesamt	15	9	47,5	4,5	120	5	100
Gesamtbewertung			265		330		278	
Endreihung			3		5		4	

8 Persönliche Machbarkeit

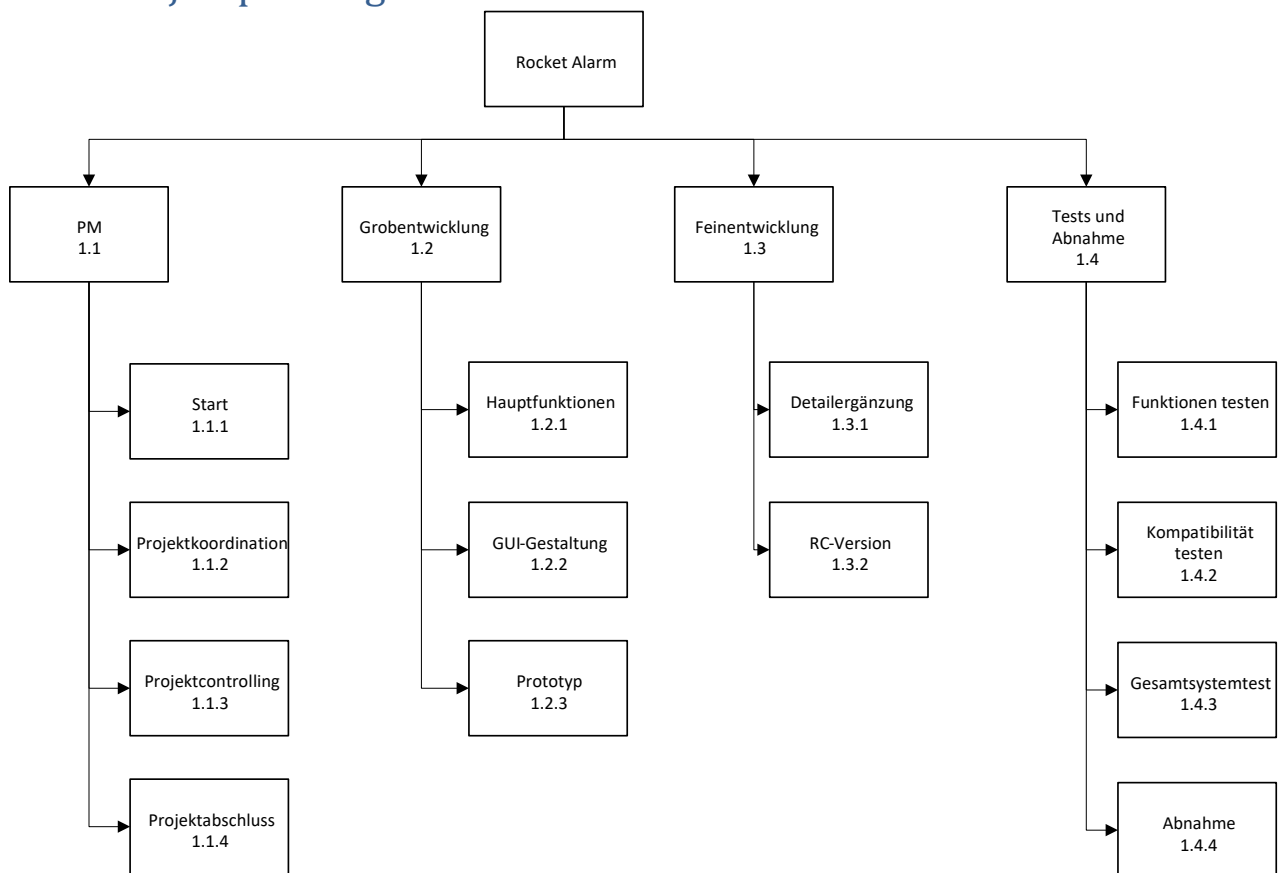
Aufgrund der Nutzwertanalyse ist ersichtlich, ob das Know-how des Teams für die bevorzugte Lösungsvariante ausreichend ist. Weiterst werden andere Möglichkeiten damit gegenübergestellt.

9 Projektorganisation

Der Grund dafür, dass dieses Projekt ins Leben gerufen wurde, ist der Projektauftraggeber Herr Prof. Rafeiner-Magor und Herr Prof. Zakall. Der Projektleiter ist Vincent Schwartz und das Team besteht noch zusätzlich aus zwei Programmierern. Die Programmierer sind Herr Kern und Herr Alesevic.



10 Projektplanung



11 Management Summary

In vielen Situationen im Leben ist einem langweilig und man weiß nicht was man machen soll. Wir haben eine Lösung gefunden um etwas gegen ihre Langweile zu tun. Wir haben ein Spiel entwickelt namens Rocket Alarm in welchem man als Helikopter von Raketen ausweichen muss und das je nach Zeit immer schwieriger wird. Dafür verwenden wir die Entwicklersprache Java. Das Spiel läuft auf Linux, Windows und OSX. Das gesamte Team ist sehr vertraut mit der Sprache Java und diese wird ausreichend sein um das Projekt durch zu führen. Weiterst wird die Entwicklung keine Finanziellen Kosten aufkommen lassen. Das komplette Projekt wird ca. bis zum Ende des Semesters benötigen. Die genauen Termine sind in der Meilensteinplanung ersichtlich.

Alles in allem kann man sagen, dass das Projekt durchführbar ist.

Meilenstein	Deliverable	Datum
Start	Alle Hefte mit den dazugehörigen Informationen	07.03.2017
Detailplanung fertig	Übersicht der Funktionen mit ihren Lösungswegen und Ersatzplänen, falls eine Funktion nicht machbar ist.	08.04.2017
Implementierung der Hauptfunktionen mit Teilen der GUI und der Serverkonfiguration	Grundcode des Systems mit den Funktionen. Außerdem wird eine GUI für den Computer vorhanden sein, die dem Freigabekandidaten noch abweichen kann.	12.05.2017
Optimierung der UseCases	Freigabekandidat des Systems und des Servers in allen Hinsichten fertig.	30.05.2017
Projektende	Fertiges Produkt auf vereinbartem Medium	01.06.2017