

Dokumentart	Dokumentation
Titel	IPA Bericht
Produkt	Inserator
Projekt	Probe IPA

Erstellt	Alain Keller	MO CDC VT	10.11.2017
		Org.-Einheit	Datum

IPA Bericht	Alain Keller	Version 1.0	1 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

Inhaltsverzeichnis

TEIL 1: UMFELD & ABLAUF	4
1 Aufgabenstellung	4
1.1 Funktionen nicht angemeldeter Nutzer	4
1.2 Funktionen angemeldeter Nutzer	4
1.3 Funktionen Admin	4
1.4 Technologien	4
1.5 Wahlkriterien	5
2 Projektorganisation	5
2.1 Beteiligte Personen	5
3 Ausgangslage	5
3.1 Vorkenntnisse	5
4 Mittel & Methoden	5
4.1 Software	5
4.2 Style Guide and Coding Conventions	5
5 Firmenstandards	6
5.1 Dokumentationsvorlage	6
6 Zeitplan	7
6.1 Meilensteine	8
7 Arbeitsprotokoll	9
7.1 Mittwoch, 01.11.2017	9
7.2 Donnerstag, 02.11.2017	10
7.3 Freitag, 03.11.2017	12
7.4 Mittwoch, 08.11.2017	13
7.5 Freitag, 10.11.2017	15
8 Kurzzusammenfassung	16
8.1 Ausgangssituation	16
8.2 Projektbeschrieb	16
8.3 Umsetzung	16
8.4 Resultat	16
TEIL 2: PROJEKT	17
9 Projektmanagement	17
9.1 Versionierung	17
9.2 Backupsystem	17
9.3 Entwicklungsumgebungen	17
10 Informieren	17
10.1 Ist-Analyse	17
10.2 Soll-Analyse	17
10.3 Systemgrenzen	18
11 Planen	19
11.1 Funktionale Anforderungen	19
11.2 Priorisieren	20
11.3 Architektur	20
11.4 Testkonzept	25

IPA Bericht	Alain Keller	Version 1.0	2 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

11.5	Testfälle	26
12	Entscheiden.....	28
12.1	Benutzeridentifizierung	28
12.2	Datenbankanbindung	28
13	Realisieren.....	29
13.1	Verzeichnisstruktur	29
13.2	Klassen.....	30
13.3	Gui Screenshots	31
14	Kontrollieren.....	34
14.1	Testübersicht.....	34
14.2	Testfazit	34
14.3	Testprotokoll.....	35
14.4	Soll – Ist Vergleich.....	36
15	Auswerten.....	37
15.1	Reflexion	37
15.2	Schlusswort	37
ANHANG		38
16	Abbildungsverzeichnis	38
17	Tabellenverzeichnis.....	39
18	Literaturverzeichnis.....	40
19	Quellenverzeichnis	40
20	Glossar.....	40
21	Source-Code.....	41

IPA Bericht	Alain Keller	Version 1.0	3 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

Teil 1: Umfeld & Ablauf

1 Aufgabenstellung

Es soll eine Webseite erstellt werden, über welche man Inserate erstellen kann. Auf der Webseite können Benutzer Inserate für Produkte und Dienstleistungen erstellen und ansehen. Die Webseite richtet sich an Desktop und Laptop-User. Ein Responsive-Design ist nicht Pflicht.

Die Benutzer der Webseite können sich über ein Login mit Benutzernamen und Passwort anmelden. Hat ein User noch kein Login, kann er sich registrieren. Es werden folgende Daten für die Registration benötigt: Name, Vorname, Benutzername, E-Mail, Telefonnummer und das Passwort. Die Registration muss nicht über E-Mail bestätigt werden. Nur angemeldete Benutzer können Inserate erstellen. Es können auch die eigenen Inserate angesehen werden, und bei Bedarf deaktiviert werden, falls diese nicht mehr aktuell sind. Deaktivierte Inserate werden für die anderen Benutzer nicht mehr angezeigt. Das Login wird nicht gespeichert und der Benutzer muss sich bei Neustart des Browsers neu anmelden. Der Benutzer kann sich auch während dem Benutzen der Website abmelden. Zusätzlich gibt es ein Admin Account. Dieser kann alle Inserate ansehen und deaktivieren. Jedoch darf er keine Inserate erfassen.

Auf der Mainpage sieht der Benutzer die Inserate der anderen Personen. Es werden pro Seite 10 Inserate angezeigt und über eine Seitennavigation am unteren Ende der Seite kann man zwischen den Seiten navigieren. Die Inserate werden nach dem Erstelldatum Sortiert. Weitere Filter, und Sortiermöglichkeiten sind nicht vorhanden. Öffnet man ein Inserat wird man auf eine detailliertere Seite weitergeleitet, welche weitere Informationen enthält. Ein Inserat enthält folgende Informationen: Titel, Beschreibung, Vorname Ersteller, E-Mail-Adresse Ersteller, Telefonnummer Ersteller, Ort und Art des Angebotes. Auf der Hauptübersicht sieht man den Titel, die Art und der Ort.

1.1 Funktionen nicht angemeldeter Nutzer

Nicht angemeldete Nutzer können die Inserate der anderen Nutzer ansehen. Sie dürfen keine Inserate erfassen oder Deaktivieren.

1.2 Funktionen angemeldeter Nutzer

Angemeldete Nutzer dürfen Inserate erfassen, ihre eigenen deaktivieren und die Inserate anderer Nutzer einsehen. Zusätzlich hat der angemeldete Nutzer die gleichen Funktionen wie der nicht angemeldete Nutzer.

1.3 Funktionen Admin

Der Admin darf Inserate von anderen Nutzern deaktivieren und somit aus der öffentlichen Anzeige entfernen. Er darf keine Inserate erstellen.

1.4 Technologien

Zur Entwicklung der Seite wird HTML, CSS, Javascript, JQuery & PHP verwendet. Der Server wird in PHP 7 objektorientiert implementiert. Für das Frontend wird HTML und CSS verwendet. Zur Steigerung der Benutzerfreundlichkeit werden zusätzliche Funktionen mit Javascript und JQuery verwendet. Die Datenbank wird mit MySQL entwickelt. Da die Webseite von mehreren Nutzern gleichzeitig benutzt werden kann, muss die Transaktionssicherheit gewährleistet sein.

IPA Bericht	Alain Keller	Version 1.0	4 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

1.5 Wahlkriterien

225 – Versionsverwaltung
 166 – Codingstyle, Lesbarer Code
 167 – Coding Dokumentiert
 123 – Kommentare im Quellcode
 164 – Codierung: Fehlerbehandlung
 125 – Gliederung des Programms
 130 – Vollständiges ERM

2 Projektorganisation

2.1 Beteiligte Personen

Name	Funktion	
Alain Keller	IPA-Kandidat	
Remo Steinmann	Experte	

Tabelle 1: Beteiligte Personen

3 Ausgangslage

3.1 Vorkenntnisse

- HTML
- CSS
- Javascript
- PHP 7
- C#

4 Mittel & Methoden

4.1 Software

- Windows 10
- Word 2016
- Excel 2016
- PHP Storm 2017.1.3
- Google Chrome V 61.0.3163.100
- Mozilla Firefox

4.2 Style Guide and Coding Conventions

IPA Bericht	Alain Keller	Version 1.0	5 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

5 Firmenstandards

5.1 Dokumentationsvorlage

Die verwendete Dokumentationsvorlage entspricht dem Siemens Firmenstandard und wurde mithilfe des Tools «Docugate» generiert.

IPA Bericht	Alain Keller	Version 1.0	6 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

6 Zeitplan

IPA Tag		1			2			3			4			5		
Wochentag		Mittwoch			Donnerstag			Freitag			Mittwoch			Freitag		
Datum		01.11.2017			02.11.2017			03.11.2017			08.11.2017			10.11.2017		
Block (in 2h)		1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1	Informieren															
1.1	Aufgabenstellung	Soll														
		Ist														
1.2	Ist-Analyse	Soll														
		Ist														
1.3	Soll-Analyse	Soll														
		Ist														
2	Planen															
2.1	Architektur erarbeiten	Soll														
		Ist														
2.2	Anforderungen beschreiben	Soll														
		Ist														
2.3	Prioritäten setzen	Soll														
		Ist														
2.4	Testkonzept erstellen	Soll														
		Ist														
3	Entscheiden															
3.1	Userauthentifikationskonzept	Soll														
		Ist														
3.2	Technologieentscheidungen fällen	Soll														
		Ist														
4	Realisieren															
4.1	HTML Startseite erstellen	Soll														
		Ist														
4.2	CSS Design machen	Soll														
		Ist														
4.3	Registrierung & Login	Soll														
		Ist														
4.4	Inserat erstellen & Löschen	Soll														
		Ist														
4.5	Inserate anzeigen	Soll														
		Ist														
4.6	Errorhandling	Soll														
		Ist														
4.7	Cleancode & Kommentare	Soll														
		Ist														
4.8	Klassendokumentation	Soll														
		Ist														
5	Kontrollieren															
5.1	Tests durchführen & Dokumentieren	Soll														
		Ist														
6	Auswerten															
6.1	Soll - Ist vergleich	Soll														
		Ist														
6.2	Fazit erstellen	Soll														
		Ist														
6.3	Dokumente für die Abgabe vorbereiten	Soll														
		Ist														
6.4	Abgabe	Soll														
		Ist														
7	Sonstiges															
7.1	Arbeitsjournal	Soll														
		Ist														
7.2	Expertentermin	Soll														
		Ist														

IPA Bericht	Alain Keller	Version 1.0	7 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

6.1 Meilensteine

Meilenstein	Beschreibung	Datum
Meilenstein 1: Start	Start des Projektes	01.11.2017
Meilenstein 2: Planung	Planung vollständig beendet, Start der Realisierung	03.11.2017
Meilenstein 3: Realisierung	Realisierung vollständig beendet, Start der Tests	08.11.2017
Meilenstein 4: Abgabe	Abgabe der IPA	10.11.2017

Tabelle 2: Projektmeilensteine

IPA Bericht	Alain Keller	Version 1.0	8 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

7 Arbeitsprotokoll

7.1 Mittwoch, 01.11.2017

Tätigkeitsliste	
Nr./Thema	Beschreibung ausgeführte Tätigkeit
Teil 1 IPA Bericht	Ich habe den ersten Teil des IPA-Berichters erstellt. Dazu gehörte der Projektauftrag, die Projektorganisation und meine Vorkenntnisse
Teil 2 VCS	Damit ich meine Dateien von Beginn an Versionisiert habe, erstellte ich heute das GIT-Repository. Dieses ist in GutHub unter folgendem Link erreichbar: https://github.com/LordMcFungus/Probe-IPA .
Teil 2: Informieren	Ich habe heute den Informieren-Teil von IPERKA abgeschlossen. Zum Informieren gehörte die Analyse der Ausgangslage und die Soll-Situation.
Teil 2: Use Cases	Anhand der Soll-Analyse erstellte ich die UseCase-diagramme für das Projekt. Diese erstellte ich mit Hilfe von PlantUml, einem Tool, um Textbasierte UMLS zu erstellen. Ein Vorteil dieses Tools ist es, dass man sich nicht mit dem Design auseinandersetzen muss. Dafür sollte man den Syntax kennen um Diagramme zu schreiben
Teil 2: Komponenten	Ich startete mit der Architektur. Dafür machte ich mir Gedanken welche Komponenten im Programm existieren werden, und was für Verantwortlichkeiten diese haben sollen. Ich habe schon damit begonnen
Journal	
Erreichte & unerreichte Ziele	<ul style="list-style-type: none"> • Informieren abgeschlossen ✓ • UseCase abgeschlossen ✓ • Architektur angefangen ✓
Aufgetretene Probleme & erarbeitete Lösungen	Mir ist noch nicht ganz klar wie das Komponentendiagramm aussehen soll. Im Internet werden verschiedene Meinungen vertreten
Reflexion	Der Heutige Tag verlief recht gut. Ich konnte mich gut an den Zeitplan halten und konnte ohne grössere Probleme das Projekt starten. Es ist teils schwer in einem Raum mit 6 anderen Lernenden zu arbeite, ohne dass man teils abgelenkt ist. Jedoch war es oft still und ich konnte mich gut konzentrieren.

Tabelle 3: Arbeitsprotokoll 01.11.2017

IPA Bericht	Alain Keller	Version 1.0	9 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

7.2 Donnerstag, 02.11.2017

Tätigkeitsliste	
Nr./Thema	Beschreibung ausgeführte Tätigkeit
Teil 2: Komponenten	Heute habe ich das Komponentendiagramm fertiggestellt. Ich habe nun einen groben Plan, wie das Frontend mit dem Backend kommunizieren wird und wie ich die Daten von der Datenbank bekomme. Zusätzlich habe ich eine kurze Beschreibung erstellt, die das Komponentendiagramm näher erläutert
Teil 2: Anforderungen	Bis jetzt hatte ich die Anforderungen noch nicht wirklich schön gegliedert aufgeschrieben. Dies habe ich heute gemacht und kann nun meine Informationen schneller holen und habe auch allenfalls auftretende Problemquellen gefunden.
Teil 2: ERM	Als nächsten Task habe ich das ERM erstellt. Damit ich dieses Fachgerecht erstellen kann, und das ERD direkt in Code exportieren kann, installierte ich mir noch die MySql Workbench. Die Grundstruktur des ERDs hatte ich relativ schnell, doch es gab immer wieder noch kleine Verfeinerungen. Anschliessend habe ich die Beziehungen und Tabellen in der Dokumentation näher beschrieben.
Teil 2: Testcases	Anschliessend an das ERM erstellte ich die Testcases. Als Grundlage dafür nahm ich die Use-Cases, welche ich gestern erstellt haben. Aufgrund der in denen beschrieben Funktionalitäten erstellte ich Testgruppen in denen ich jeweils mehrere Positiv-, sowie auch Negativtestfälle beschreibe.
Expertenbesuch	Heute fand der erste Expertenbesuch statt. Ich sollte meine Reflexion ausführlicher machen. Ansonsten ist es gut.
Teil 2: Priorisieren	Gegen Ende des Tages begann ich die Tasks für die Realisierung zu priorisieren. Damit soll ich sicherstellen, dass ich nicht bei einem unwichtigen Task zu lange hängenbleibe und somit die Hauptfunktionalitäten nicht implementieren kann.
Journal	
Erreichte und Unerreichte Ziele	<ul style="list-style-type: none"> • Architekturteil der Planung abschliessen ✓ • Anforderungen detailliert und verständlich dokumentieren ✓ • ERM erstellen und abschliessen ✓ • Testcases Erstellen und abschliessen ✗ • Priorisierung abschliessen ✓
Aufgetretene Probleme & erarbeitete Lösungen	Die meisten Probleme sind heute beim Erstellen des ERDs und der Aufsetzung der Datenbank aufgetreten. Zuerst konnte ich die Workbench nicht installieren, weil ich angeblich den Visual C++ Treiber von 2015 nicht installiert habe, ich hatte denjenigen von 2017. Den erforderlichen Treiber konnte ich dann auch nicht installieren, weil ich schon eine neuere Version haben. Nachdem ich meinen aktuellen Treiber deinstallierte, konnte ich den alten installieren und die Workbench erfolgreich aufsetzen. Weiter Probleme sind bei den CREATE-Scripts von MySql aufgetaucht. Immer wieder hatte ich Fehler in der generierten Abfrage. Die meisten konnte ich dann aber damit beheben, dass ich die bestehende Datenbank löschte und neu aufsetzte, anstatt die bestehende zu überschreiben.
Reflexion	Heute war ein strenger Tag. Vor allem die Arbeit mit MySql war mühsam. Jedoch bin ich auch mehr selber schuld, da ich die nötige Software nicht im Voraus installiert habe sondern erst heute. Das nächste Mal werde ich da sicher besser aufpassen. Die Arbeit im Sitzungszimmer verlief auch mehr

IPA Bericht	Alain Keller	Version 1.0	10 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

	<p>oder weniger ok. Es gab wieder Situationen, in denen es schwer war sich zu konzentrieren, doch mehrheitlich war es gut.</p> <p>Ich bin mit meinen tageszielen nicht ganz durchgekommen. Ich hatte aber auch einige unproduktive Minuten, die ich sicherlich sinnvoller hätte einsetzen können. Ich hoffe, dass ich am Freitagmorgen schnell mit meinen Pendenzen durch bin und mit der Implementierung beginnen kann.</p>
--	--

Tabelle 4: Arbeitsprotokoll 2.11.2017

IPA Bericht	Alain Keller	Version 1.0	11 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

7.3 Freitag, 03.11.2017

Tätigkeitsliste	
Nr./Thema	Beschreibung ausgeführte Tätigkeit
Teil 2 Testcases	Da ich gestern noch nicht fertig wurde mit den Testcases, stellte ich diese heute Morgen noch fertig.
Teil 2: Entscheidungen	Heute Morgen habe ich den Entscheiden-Teil von IPERKA gemacht. Die Entscheidungen bezogen sich hauptsächlich auf lösungsvarianten von Problemen wie die Benutzerauthentifizierung.
Teil 2 HTML und CSS	Nach dem Entscheiden konnte ich endlich mit der Realisierung beginnen. Zuerst habe ich das Grundgerüst im HTML gemacht. Da ich aber die meisten Daten dynamisch lade, sieht die Seite momentan noch leer aus. Zusätzlich habe ich noch die verschiedenen Formulare zu Registrieren und Einloggen implementiert.
Teil 2 Registrierung	Nach dem Design begann ich mit der Implementation der registrierung. Für diese brauchte ich jedoch länger als geplant, da ich auf einige Probleme gestossen bin. Ich habe es dennoch fertig geschafft, wenn auch mit Verzug
Teil 2 Login	Mit dem Login konnte ich heute nur knapp anfangen. Geplant war aber, dass ich fertig werde.
Journal	
Erreichte und nicht erreichte Ziele	<ul style="list-style-type: none"> • Testcases ✓ • Entscheidungen ✓ • HTML & CSS ✓ • Registrierung ✓ • Login ✗ • Errorhandling ✗
Probleme	Die meisten Probleme hatte ich heute während der Implementation. Zuerst funktionierte Ajax nicht so wie ich wollte. Doch nach ein Bisschen Googeln und Pascal fragen hat es dann funktioniert. Ajax ist für mich eine neue Technologie. Mein fehlendes Fachwissen ist wahrscheinlich auch der Hauptgrund für meinen Verzug. Danach hatte ich noch ein paar kleine Logikprobleme bei der Implementation der Registrierung. Doch auch diese konnte ich beheben und die Registrierung funktioniert jetzt. Jedoch habe ich einen Teil noch nicht erledigt, das Errorhandling. Server intern funktioniert sie zwar, der Benutzer bekommt aber noch kein brauchbares Feedback
Reflexion	Die grösste Lektion von Heute war wohl, dass ich zu wenig Erfahrung habe in der Webentwicklung. Ich habe viel Zeit durch den Kampf mit den Technologien verloren, die ich sinnvoller hätte nutzen können. Dafür habe ich heute viel neues gelernt, was ich vorher nicht wusste. Durch die vielen Verzögerungen in der Implementation konnte ich auch meinen Zeitplan nicht ganz einhalten. Da werde ich am nächsten Mittwoch sicher noch viel zu tun haben.

Tabelle 5: Arbeitsprotokoll 23.03.2017

IPA Bericht	Alain Keller	Version 1.0	12 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

7.4 Mittwoch, 08.11.2017

Tätigkeitsliste	
Nr./Thema	Beschreibung ausgeführte Tätigkeit
Login	Heute habe ich das Login fertig erstellt. Da ich am Vortag nicht den Zeitplan einhalten konnte, musste ich dies entgegen des Zeitplanes heute erledigen,
Inputvalidierung	Nach der Implementation des Logins versuchte ich die Inputvalidierung der Registrationen des Logins zu verbessern. Ich versuchte eine Überprüfung einzubauen, damit ein Username nicht zweimal erstellt werden kann.
Inserat Erfassen	Danach habe ich die Logik für das Erfassen eines Inserates implementiert. Dabei konnte ich viel von der Registrierung übernehmen
Inserat anzeigen	Nach dem Erfassen implementierte ich die Anzeige des Inserates. Dies verlief recht gut. Einzig hatte ich Probleme mit der Typisierung von Javascript.
Errorhandling	Ich hätte noch geplant, dass ich das Errorhandling verbessere und ausbaue, jedoch fehlte mir am Schluss die Zeit und ich musste diesen Task auslassen.
Expertengespräch	Im Verlauf des Tages hatte ich das Expertengespräch mit Remo. Er hat mir noch ein paar Hinweise gegeben, wo ich meine Dokumentation noch verbessern könnte.
Journal	
Erreichte & unerreichte Ziele	<ul style="list-style-type: none"> • Login✓ • InputvalidierungX • Inserat Erfassen✓ • Inserat anzeigen✓ • ErrorhandlingX • Dokumentation Realisierung✓
Aufgetretene Probleme und Lösungen	Am meisten Probleme hatte ich heute mit PDO. Ich versuchte aus einer Abfrage die Anzahl gefundener Datensätze auszulesen, ohne dass ich durch jeden durchiterieren muss. Jedoch habe ich dies nicht wirklich geschafft. Da ich das brauchte, um zu überprüfen ob ein Username schon erfasst ist, habe ich nach einer neuen Lösung gesucht. Schlussendlich habe ich das Feld in der Datenbank auf Unique geschaltet und den Fehler über die Insert Abfrage abgefangen. Des Weiteren hatte ich Probleme mit der Typisierung von Javascript. Da man einer Variable keinen bestimmten Datentyp zuweisen kann, gab das Resultat einer einfachen 1+1 Rechnung 11 statt 2. Diesen Fehler konnte ich mit einem einfachen castInt() beheben.
Reflexion	Heute war ein strenger Tag. Ich konnte leider nicht alle Tasks erfüllen, die ich mir vorgenommen habe. Ich musste die Input-Validierungsmeldungen auf das Frontend kürzen und kann keine Usermeldungen vom Server aus schicken. Ich habe diese Anforderungen absichtlich nicht implementiert, da es mir wichtiger war die Grund Funktionalität des Programmes zu implementieren. Die Realisierung ist jedoch noch nicht abgeschlossen, da ich noch

IPA Bericht	Alain Keller	Version 1.0	13 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

	nicht dokumentiert habe. Ich werde am kommenden Freitag zusätzlich noch die Realisierungsdoku schreiben müssen.
--	---

Tabelle 6: Arbeitsprotokoll 24.03.2017

IPA Bericht	Alain Keller	Version 1.0	14 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

7.5 Freitag, 10.11.2017

Tätigkeitsliste	
Nr./Thema	Beschreibung ausgeführte Tätigkeit
Realisierung Dokumentieren	Heute Morgen habe ich die Realisierung Dokumentiert. Dabei ging ich auf die Dateistruktur und die einzelnen Klassen ein.
Testen	Nach der Realisierung habe ich gemäss meinen Testfällen die Applikation getestet. Die Implementierten Funktionen haben grössten Teils die Tests bestanden. Einzig die fehlenden haben logischerweise nicht bestanden
Kontrollieren	Zusätzlich zum Testen erstellte ich einen Soll – Ist vergleich, in welchem ich mich auf die vergangen Soll-Analyse beziehe.
Auswerten	In der Auswertung stellte ich die Reflexion und das Fazit des Projektes
Dokumentanhang erstellen	Für die Projektabgabe habe ich den Anhang mit meinem geschriebenen Code befüllt.
Journal	
Erreicht & Unerreichte Ziele	<ul style="list-style-type: none"> • Realisierung dokumentierenX • Testen✓ • Kontrollieren✓ • Auswerten✓ • Dokumentanhang erstellen✓ • Dokument nach Kriterien überprüfenX
Aufgetretene Probleme und Lösungen	Das heutige Hauptproblem war Word. Nachdem ich den ganzen Morgen an der Dokumentation gearbeitet habe, und dieses jeweils zwischenspeicherte, schloss ich mein Word. Als ich die Doku wieder öffnete war die ganze Arbeit vom Morgen verschwunden. Ich versuchte die alte Version zu wiederherstellen doch es funktionierte nicht. Ich habe danach Remo gefragt, ob ich am Samstag dem 11.11 meine finale Version abgeben kann. Ich werde nun heute eine Version abgeben und die überarbeitete Version am Samstag bis um 18:00 Uhr abgeben. Sonst hatte ich heute keine weiteren Probleme.
Reflexion	Bis am Mittag lief der heutige Tag eigentlich recht gut. Doch nachdem mir Word einen Strich durch die Rechnung machte, wurde ich unruhig. Ich konnte jedoch die verlorenen Teile nochmals schreiben, wenn auch nicht ganz so detailliert. Ich habe vor allem die Dokumentation der Realisierung nicht so genau gemacht wie am Morgen. Da ich am Nachmittag die Auswertung und die Kontrolle eingeplant hatte, legte ich vor allem auf diese beiden Teile mehr Gewicht

Tabelle 7: Arbeitsprotokoll 28.03.2017

IPA Bericht	Alain Keller	Version 1.0	15 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

8 Kurzzusammenfassung

8.1 Ausgangssituation

Für das Modul 223 wurde ein Kleinprojekt gesucht, welches Objektorientiert implementiert wird und über eine Datenbank verfügt. Das Projekt sollte vom Kandidaten innerhalb von 5 Tagen realisierbar sein, wobei zu beachten war, dass von diesen 5 Tagen mindestens 2 Tage dokumentiert wird.

8.2 Projektbeschreibung

Das Projekt ist eine Webseite, auf der man Inserate erfassen kann. Diese können von anderen Benutzern angesehen werden. Der Benutzer muss angemeldet sein um ein Inserat zu erfassen und seine eigenen Inserate zu löschen. Für das Projekt werden folgende Techniken verwendet: HTML, CSS, JavaScript, PHP und MySql. Ausserdem wird die Javascriptlibrary JQuery verwendet, damit über Ajax mit dem Server kommuniziert werden kann.

8.3 Umsetzung

Das aufgetragene Kleinprojekt wurde mit der Projektmanagementmethode IPERKA realisiert. Innerhalb von 5 Tagen wurde das Projekt abgewickelt und abgeschlossen.

8.4 Resultat

Das Projekt konnte in dieser Zeit nicht komplett fertiggestellt werden. Die aktuelle Version ist jedoch lauffähig.

IPA Bericht	Alain Keller	Version 1.0	16 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

Teil 2: Projekt

9 Projektmanagement

Als Projektmanagementmethode verwende ich in diesem Projekt IPERKA. Es ist einfach aufgebaut und eignet sich gut für Einzel Projekte über eine kurze Laufzeit

9.1 Versionierung

Als VCS verwende ich Git über die Server von Github. Ich habe mich für Git entschieden, da ich in vergangenen Schulprojekten schon Git über GitHub verwendet habe. Pro Tag werde ich mindestens einen Commit ausführen. Zusätzlich werde ich mindestens nach jedem Implementierten Feature, oder bevor ich meinen Laptop für mehr als eine viertel Stunde verlasse einen Commit durchführen.

9.2 Backupsystem

Neben dem regelmässigen speichern auf GitHub wird am Ende eines jeden Tages eine Kopie auf Dropbox gespeichert. Dies stellt die Verfügbarkeit der Arbeit sicher, falls der Laptop und GitHub gleichzeitig nicht erreicht werden können.

9.3 Entwicklungsumgebungen

Den Code entwickle ich in PHP-Storm. Es ist eine IntelliJ basierte IDE, welche ich auch schon von vergangenen Projekten kenne. Den Apache und MySQL-Server verwalte ich über Xampp, da sich dieses gut für solche Projekte eignet und gratis verfügbar ist.

10 Informieren

In diesem Teil trage ich die Informationen zusammen und beschreibe diese.

10.1 Ist-Analyse

Zurzeit besteht noch nichts von der gewünschten Applikation.

10.2 Soll-Analyse

Es soll eine WebApp erstellt werden, über welche Benutzer Inserate erstellen und ansehen können.

10.2.1 Frontend

Das Frontend wird mit Hilfe von HTML, CSS und Javascript erstellt. Der Benutzer sieht auf der Startseite die schon erstellten Inserate der anderen Benutzer. Diese werden in mehrere Seiten aufgeteilt. Je Seite werden 10 Inserate angezeigt. Auf der Inserat-Liste werden nicht alle Informationen angezeigt. Wählt der Nutzer ein Inserat aus, wird dieses Geöffnet und man sieht mehr Details. Der Benutzer kann sich auf der Webseite registrieren und Anmelden. Als angemeldeter Nutzer kann er ein eigenes Inserat erstellen, und die erstellten Inserate deaktivieren.

IPA Bericht	Alain Keller	Version 1.0	17 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

10.2.2 Backend

Das Backend ist vor allem dazu da, die Schnittstelle zwischen Frontend und Datenbank zu machen. Die Logindaten werden übers Backend validiert, die Inserate werden ausgelesen und gespeichert. Das Backend soll objektorientiert mit Hilfe von PHP 7.1 geschrieben werden.

10.2.3 Datenbank

Die Datenbank wird mit MySQL gemacht. Die Transaktionssicherheit muss jederzeit gewährleistet sein. In der Datenbank werden die Benutzerdaten und die Inserate gespeichert.

10.2.4 Design und Layout

Es stehen keine konkreten Anforderungen zum Design. Die Webseite muss jedoch nicht Responsive implementiert sein.

10.2.5 Errorhandling

Fehler im Programm sollten erkannt werden und entsprechend damit umgegangen werden.

10.2.6 Source Code

Der Source-Code soll verständlich sein und kommentiert. Es sollen die Clean-Code Guidelines eingehalten werden.

10.2.7 Tests

Die Applikation wird nach der Implementation ausführlich gegen die Anforderungen getestet. Dabei wird auf Blackboxtesting gesetzt.

10.3 Systemgrenzen

Die Applikation ist ein in sich geschlossenes System, welches in einem Browser angezeigt wird. Alles was ausserhalb dieser Applikation ist, gehört nicht zum System.

IPA Bericht	Alain Keller	Version 1.0	18 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

11 Planen

11.1 Funktionale Anforderungen

Die Soll-Analyse wird nun nochmals verfeinert und in funktionale- und nichtfunktionale Anforderungen gegliedert.

Name	Beschreibung
Registrieren	<p>Der Benutzer kann sich auf der Seite Registrieren. Für die Registration muss der Benutzer folgende Daten angeben:</p> <ul style="list-style-type: none"> • Name • Vorname • Benutzername • E-Mail • Telefonnummer • Passwort <p>Es gibt keine Bestätigungsemail bei der Registration Bei Falscheingaben bekommt der Benutzer sinnvolle Rückmeldungen.</p>
Login	<p>Ein registrierter Nutzer kann sich mit seinen Daten einloggen. Für das Login braucht er seinen Benutzernamen und das Passwort. Bei Falscheingaben bekommt der Benutzer sinnvolle Rückmeldungen.</p>
Logout	<p>Ein eingeloggtter Nutzer kann sich ausloggen. Seine aktuelle Sitzung wird somit beendet und er hat nur noch Zugriff auf die Funktionen eines nicht angemeldeten Nutzer.</p>
Inserat erstellen	<p>Ein angemeldeter Nutzer kann ein Inserat erstellen. Dazu muss in einem Inserat folgende Daten angeben:</p> <ul style="list-style-type: none"> • Titel des Inserates • Beschreibung • Vorname des Erstellers • Die E-Mail des Erstellers • Die Telefonnummer des Erstellers • Ort des Angebotes • Art des Angebotes <p>Hat ein Nutzer ein Inserat erstellt, wird dieses in der Übersicht auf der Mainpage angezeigt. Bei Falscheingaben bekommt der Benutzer sinnvolle Rückmeldungen.</p>
Inserat Übersicht	<p>Auf der Startseite werden die Inserate in einer Liste angezeigt. Die Liste enthält nicht alle Informationen eines Inserates, sondern nur den Titel, die Art des Inserates und der Ort. Nach 10 Inseraten muss man über eine Navigation auf die nächste Seite gehen, um die folgenden Inserate zu sehen. Die Inserate werden nach der Erfassungszeit absteigend sortiert; die neusten sind zuoberst.</p>
Inserat öffnen	<p>Auf der Übersichtsseite kann man die Einzelnen Inserat öffnen um weitere Details zu erfahren. Die angezeigten Details entsprechen den selben Daten wie beim Inserat erstellen.</p>
Inserat Löschen	<p>Ein angemeldeter Nutzer kann seine Inserate Löschen, falls diese nicht mehr aktuell sind. Hat ein Nutzer ein Inserat gelöscht, kann es von anderen Nutzern nicht mehr gesehen werden.</p>
Adminbereich	<p>Ein als Admin gekennzeichnete User kann sich anmelden und hat die Möglichkeit alle Inserate zu löschen, auch wenn sie nicht von ihm stammen</p>

Tabelle 8 Funktionale Anforderung

IPA Bericht	Alain Keller	Version 1.0	19 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

11.1.1 Berechtigungen

Funktion	Nicht angemeldeter Nutzer	Angemeldeter Nutzer	Admin
Inserate Ansehen	✓	✓	✓
Inserate Öffnen	✓	✓	✓
Inserat erstellen	✗	✓	✗
Eigene Inserate Löschen	✗	✓	✗
Beliebiges Inserat Löschen	✗	✗	✓

Tabelle 9 Berechtigungsmatrix

11.2 Priorisieren

Alle Funktionalen Anforderungen haben höchste Priorität. Die Reihenfolge der Implementierung erfolgt gleich wie die Auflistung der funktionalen Anforderungen.

11.3 Architektur

11.3.1 Use Cases

Use Case Nr.	1
Use Case Bezeichnung	User Registrierung
Akteur	Benutzer
Vorbedingung	Benutzer hat die Seite geöffnet und ist nicht eingeloggt
Use Case Beschreibung	Nutzer öffnet Registrationsseite Nutzer gibt valide Daten ein Nutzer wird Registriert
Resultat	Nutzer ist Eingeloggt auf der Startseite

Tabelle 10: Use Case #1

Use Case Nr.	2
Use Case Bezeichnung	User Login
Akteur	Benutzer
Vorbedingung	Benutzer hat die Seite geöffnet und hat schon einen Account
Use Case Beschreibung	Nutzer öffnet das Loginfenster Nutzer gibt seine Daten ein Nutzer wird Eingeloggt
Resultat	Nutzer ist Eingeloggt auf der Startseite

IPA Bericht	Alain Keller	Version 1.0	20 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

Tabelle 11: Use Case #2

Use Case Nr.	3
Use Case Bezeichnung	Logout
Akteur	Benutzer
Vorbedingung	Benutzer hat sich eingeloggt
Use Case Beschreibung	Nutzer loggt sich aus
Resultat	Der Nutzer ist abgemeldet. Er kann nicht mehr auf seine Daten zugreifen.

Tabelle 12: Use Case #3

Use Case Nr.	4
Use Case Bezeichnung	Inserat ansehen
Akteur	Benutzer
Vorbedingung	Benutzer befindet sich auf der Startseite
Use Case Beschreibung	Nutzer öffnet ein Inserat
Resultat	Es wird die Detailseite des Inserates angezeigt.

Tabelle 13: Use Case #4

Use Case Nr.	5
Use Case Bezeichnung	Inserat erstellen
Akteur	Benutzer
Vorbedingung	Benutzer hat sich eingeloggt
Use Case Beschreibung	Nutzer öffnet Seite zum neues Inserat erstellen Nutzer gibt valide Daten ein Inserat wird gespeichert
Resultat	Inserat ist auf der Datenbank gespeichert und kann von anderen Nutzern auf der Startseite angesehen werden.

Tabelle 14: Use Case #5

Use Case Nr.	6
Use Case Bezeichnung	Inserat Löschen
Akteur	Benutzer
Vorbedingung	Benutzer hat sich eingeloggt, Nutzer hat schon ein Inserat erstellt
Use Case Beschreibung	Nutzer öffnet Seite seiner Inserate Nutzer löscht ein Inserat
Resultat	Das Inserat wird nicht mehr angezeigt.

Tabelle 15: Use Case #6

Use Case Nr.	7
--------------	---

IPA Bericht	Alain Keller	Version 1.0	21 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

Use Case Bezeichnung	Inserat erstellen
Akteur	Administrator
Vorbedingung	Administrator hat sich eingeloggt, Es wurden schon Inserate erstellt
Use Case Beschreibung	Administrator öffnet Inserateseite Administrator Löscht beliebiges Inserat
Resultat	Das Inserat wird nicht mehr angezeigt.

Tabelle 16: Use Case #7

11.3.2 Use Case Diagramm

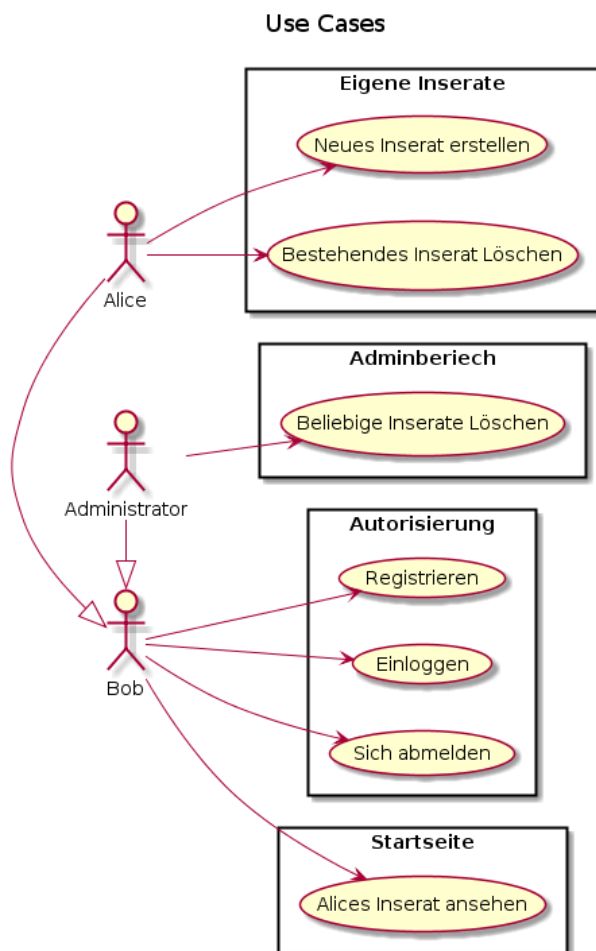


Abbildung 1 Usecase Diagramm

IPA Bericht	Alain Keller	Version 1.0	22 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

11.3.3 Komponenten

Das Programm ist in drei Systemen aufgeteilt, welche miteinander kommunizieren müssen. Diese drei Systeme umfassen das Frontend, das Backend und die Datenbank. Um den Datenaustausch zwischen den einzelnen Dokumenten genauer zu erläutern habe ich folgendes Komponentendiagramm erstellt.

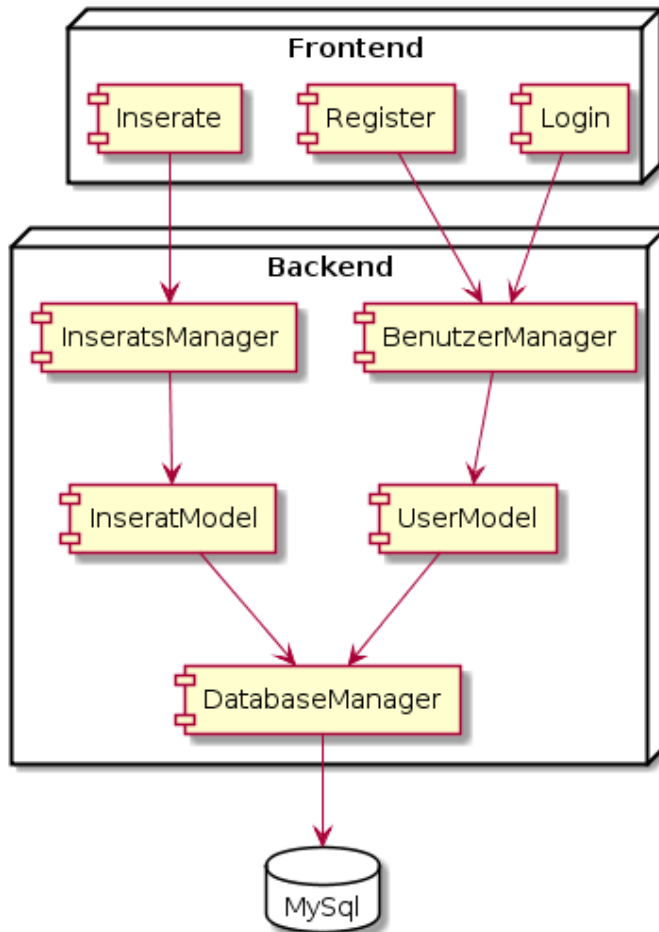


Abbildung 2 Komponentendiagramm

11.3.4 Frontend

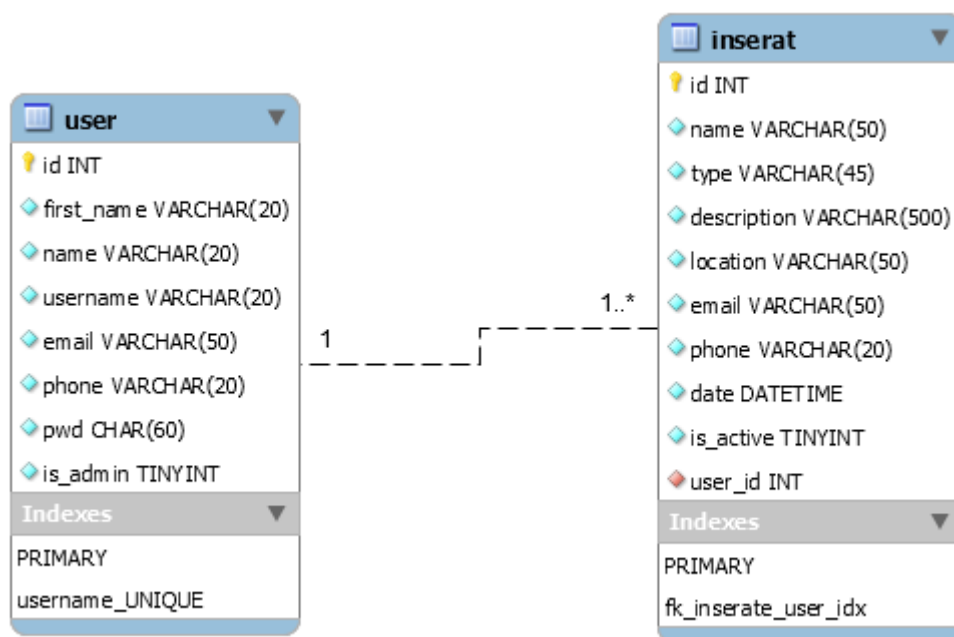
Das Frontend besitzt über mehrere Javascriptkomponenten, welche die Daten an den Server weiterleiten. Die Kommunikation zum Server erfolgt über das http und wird mit Ajax implementiert. Dies ermöglicht es, dass auf einer Seite verschiedene Controller auf dem Server unabhängig voneinander aufgerufen werden können. Zusätzlich kann das Frontend Serveranfragen machen, ohne dass die Seite danach neu geladen werden muss.

11.3.5 Backend

Im Backend werden die Daten aufbereitet um sie entweder in der Datenbank zu speichern, oder um die Daten von der Datenbank auszulesen, um diese dann auf der Webseite anzeigen zu können. Das Backend wird in Controller und Models aufgeteilt, damit eine klare Trennung der Funktionen vorhanden ist. Die Datenbank wird über einen Datenbank Manager angesprochen. Dieser enthält die Instanz, um die Datenbank ansprechen zu können. Somit soll sichergestellt sein, dass die Datenbankbindung nur dann steht, wenn sie auch wirklich gebraucht wird.

IPA Bericht	Alain Keller	Version 1.0	23 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

11.3.6 Datenbank



User				
Beschreibung	Diese Tabelle speichert die Benutzerdaten ab. Diese entsprechen denjenigen, welche in den Anforderungen gewünscht sind.			
Relationen	Ein Benutzer kann keine oder mehrere Inserate haben			
Felder	Feldname	Datentyp	Beschreibung	
	Id	INT	Dieses Feld ist der Primärschlüssel der Tabelle. Es ist ein Integer, welcher automatisch Inkrementiert.	Not Null Auto Increment PK
	first_name	Varchar(20)	Der Vorname des Benutzers	Not Null
	name	Varchar(20)	Der Nachname des Benutzers	Not Null
	username	Varchar(20)	Der Benutzername des Benutzers	Not Null
	email	Varchar(50)	Die E-Mail des Benutzers	Not Null
	phone	Varchar(20)	Die Telefonnummer des Benutzers	Not Null
	pwd	Char(60)	Das Passwort des Benutzers. Dieses wird von Bcrypt gehasht und abgespeichert. Daher wird auch kein zusätzliches Feld für einen Salt benötigt.	Not Null
	Is_admin	TINYINT	Enthält den Wert, ob ein User ein Admin ist oder ein normaler User.	Not Null

Tabelle 17 User Datenbank

IPA Bericht	Alain Keller	Version 1.0	24 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

Inserat				
Beschreibung				
Relationen				
Felder	Feldname	Datentyp	Beschreibung	
	Id	INT	Dieses Feld ist der Primärschlüssel der Tabelle. Es ist ein Integer, welcher automatisch Inkrementiert.	Not Null Auto Increment PK
	name	Varchar(50)	Dieses Feld enthält die Überschrift des Inserates	Not Null
	type	Varchar(45)	Diese Feild enthält den Inserattyp	Not Null
	description	Varchar(500)	Diese Feld enthält die Inseratbeschreibung	Not Null
	location	Varchar(50)	Dieses Feld enthält den Ort, wo das im Inserat beworbene Produkt verfügbar ist	Not Null
	email	Varchar(50)	Dieses Fled enthält die Kontaktemail für das Inserat	Not Null
	phone	Varchar(20)	Dieses Feld enthält die Kontakttelefonnummer des Inserates	Not Null
	date	Datetime	Diese Feld enthält das Datum an dem das Inserat erstellt wurde	Not Null
	is_active	Tinyint	Dieses Feld enthält den Wert, ob das Inserat noch verfügbar ist oder nicht.	Not Null
	user_id	Int	Fremdschlüssel für den User, der dieses Inserat erstellt hat.	Not Null

Tabelle 18 Inserat Datenbank

11.4 Testkonzept

11.4.1 Testmethoden/Testmittel

Das Programm wird mit Blackboxtesting getestet.

11.4.2 Testumgebung

Die Testumgebung setzt sich folgendermassen zusammen:

- Software:
 - Windows 10 Professional
 - XAMPP

IPA Bericht	Alain Keller	Version 1.0	25 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

- Hardware:
 - Dell XPS 15

11.5 Testfälle

11.5.1 Registrierung

Nr.	Vorbedingungen	Testschritte	Erwartetes Ergebnis
1.1	Startseite ist Geöffnet und der Benutzer ist noch nicht eingeloggt	1. Registrierungsknopf drücken	Oben rechts erscheint das Registrierungsformular. Der Rest der Seite bleibt gleich
1.2	Registrierungsformular ist geöffnet, User ist noch nicht Angemeldet oder Registriert.	1. Folgende Daten eingeben: Name: Hans Vornamen: Zimmer Benutzernamen: Hansli E-Mail: haz@hans.ch Telefonnummer: 076 666 66 66 Passwort: Schlachtlauch Passwort Schlachtsellerie Wiederholen: 2. Registrieren drücken	Registrierung Fehlgeschlagen. Passwörter werden gelöscht und rot umrahmt
1.3	Registrierungsformular ist geöffnet, User ist noch nicht Angemeldet oder Registriert.	1. Folgende Daten eingeben: Name: Hans Vornamen: Zimmer Benutzernamen: Hansli E-Mail: haz@hans.ch Telefonnummer: 076 666 66 66 Passwort: Schlachtlauch Passwort Schlachtlauch Wiederholen: 2. Registrieren drücken	Registrierung ist erfolgreich Hans ist eingeloggt
1.4	Registrierungsformular ist geöffnet, User ist noch nicht Angemeldet oder Registriert. Hansli hat sich schon registriert	1. Folgende Daten eingeben: Name: Hans Vornamen: Zimmer Benutzernamen: Hansli E-Mail: haz@hans.ch Telefonnummer: 076 666 66 66 Passwort: Schlachtzwiebel Passwort Schlachtzwiebel Wiederholen: 2. Registrieren drücken	Registrierung ist nicht erfolgreich. Es besteht schon ein User mit demselben Namen. Das Feld Username wird rot umrahmt

Tabelle 19 Registrierung Testfall

IPA Bericht	Alain Keller	Version 1.0	26 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

11.5.2 Login

Nr.	Vorbedingungen	Testschritte	Erwartetes Ergebnis
2.1	Startseite ist Geöffnet	1. Loginknopf drücken	Oben rechts erscheint das Loginsformular. Der Rest der Seite beliebt gleich
2.2	Registrierungsformular ist geöffnet, Hansli ist registriert, Hansli ist noch nicht angemeldet	1. Folgende Daten eingeben: Benutzername: Hansli Passwort: Schlachtlauch 2. Login drücken	Login erfolgreich, Hansli ist eingeloggt
2.3	Registrierungsformular ist geöffnet, Hansli ist registriert, Hansli ist noch nicht angemeldet	1. Folgende Daten eingeben: Benutzername: Peterli Passwort: Schlachtlauch 2. Login drücken	Login nicht erfolgreich, Meldung wird Angezeigt

Tabelle 20: Login Testfall

11.5.3 Logout

Nr.	Vorbedingungen	Testschritte	Erwartetes Ergebnis
3.1	Startseite ist Geöffnet	1. Loginknopf drücken	Oben rechts erscheint das Loginsformular. Der Rest der Seite beliebt gleich

Tabelle 21: Logout Testfall

11.5.4 Inserate erstellen

Nr.	Vorbedingungen	Testschritte	Erwartetes Ergebnis
4.1	Startseite ist Geöffnet, Hansli ist angemeldet	1. «Inserat erstellen» Knopf drücken	Das Inserat Erstellen Formular wird geöffnet
4.2	Registrierungsformular ist geöffnet, Hansli ist registriert, Hansli ist noch nicht angemeldet	1. Folgende Daten eingeben: Titel: Mostmaschine Art: Verkaufen Beschreibung Eine Alte Moschtmachine in gutem Zustand steht zum Verkauf. Preis ab 500 CHF. Email most@hansli.ch Telefon 076 666 66 66 Ort Schönholzerswilen 2. Inserat speichern drücken	Inserat wurde erfolgreich erstellt. Hansli wird zurück zur Hauptseite geleitet und das Inserat sollte bei den neusten angezeigt werden.
4.3	Registrierungsformular ist geöffnet, Hansli ist registriert, Hansli ist noch nicht angemeldet	1. Folgende Daten eingeben: Titel: Mostmaschine Art: Verkaufen Beschreibung Eine Alte Moschtmachine in gutem Zustand steht zum Verkauf. Preis ab 500 CHF.	Inserat wurde nicht abgespeichert. E-Mail Feld wird Rot umrahmt.

IPA Bericht	Alain Keller	Version 1.0	27 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

		Email @. Telefon 076 666 66 66 Ort Schönholzerswilen 2. Inserat speichern drücken	
--	--	---	--

Tabelle 22 Inserate erstellen Testfall

11.5.5 Inserate ansehen

Nr.	Vorbedingungen	Testschritte	Erwartetes Ergebnis
5.1	Startseite ist Geöffnet, Hansli ist nicht angemeldet	1. Inserat mit dem Titel Mostmaschine öffnen	Die Detailseite des Inserates wird geöffnet

Tabelle 23 Inserate ansehen Testfall

11.5.6 Inserat Löschen

Nr.	Vorbedingungen	Testschritte	Erwartetes Ergebnis
6.1	Startseite ist Geöffnet, Hansli ist angemeldet	1. «Meine Inserate» Knopf drücken	Die Meine Inserate Seite wird angezeigt.
5.2	Meine Inserate Seite ist Geöffnet, Hansli ist angemeldet	1. «Löschen» Knopf neben dem Inserat Mostmaschine drücken	Inserat Mostmaschine wird gelöscht

Tabelle 24 Inserat Löschen Testfall

12 Entscheiden

12.1 Benutzeridentifizierung

Für die Benutzerautorisierung bestehen zwei Möglichkeiten. Die eine erfolgt über Cookies und die andere über die PHP-Session. Ich implementiere die Authentifizierung über die Session, da ich dies schon gemacht habe und weniger aufwändig ist zu implementieren.

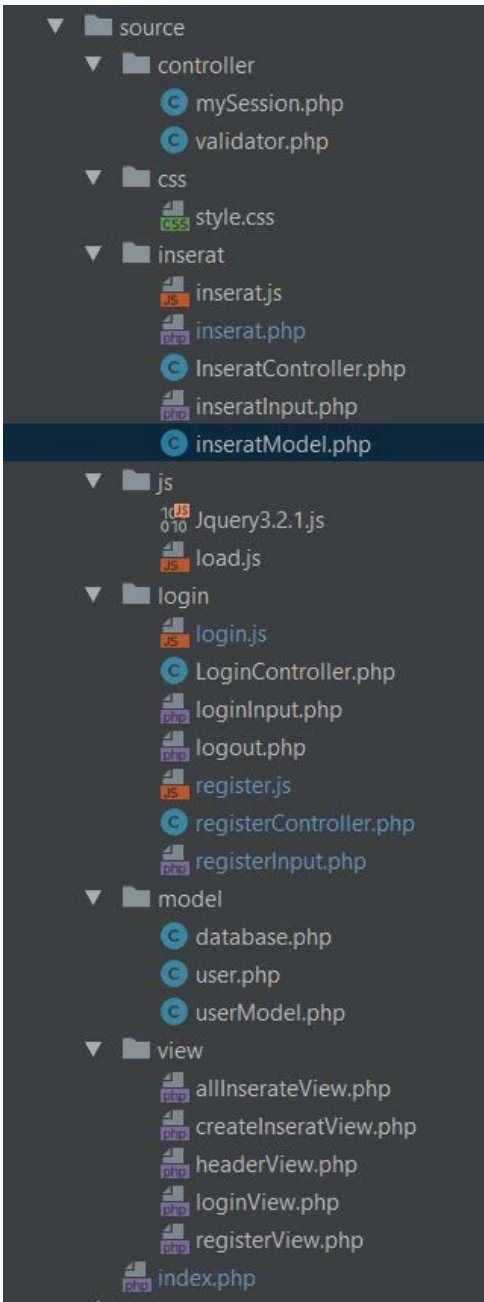
12.2 Datenbankankbindung

Die Datenbankankbindung erfolgt über PDO. Ich entschied mich für PDO weil es unabhängig vom benutzten DBMS benutzt werden kann

IPA Bericht	Alain Keller	Version 1.0	28 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

13 Realisieren

13.1 Verzeichnisstruktur



Der Source-code wird in verschiedene Ordner unterteilt. Diese Ordner richten sich jeweils nach den Funktionalitäten der darin enthaltenen Klassen.

Verzeichnis	Beschreibung
Controller	Enthält generelle Controllerklassen wie den mySession Singleton
Css	Enthält die Stylesheetdateien
Inserat	Enthält alle Klassen, welche für die Aufbereitung und Speicherung von Inseraten zu tun haben
Js	Enthält allgemeine Javascripts und Libraries
Login	Enthält alle Klassen, welche benötigt werden um den User zu registrieren, Anzumelden und abzumelden.
Model	Enthält die Dateien, welche direkt mit der Datenbank zu tun haben.
View	Enthält die verschiedenen View-Dateien, welche im Frontend angezeigt werden
Source	Enthält die index.php Datei, sowie die Unterordner, welche oben erwähnt wurden.

Abbildung 3 Ordnerstruktur

IPA Bericht	Alain Keller	Version 1.0	29 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

13.2 Klassen

13.2.1 mySession

MySession ist ein Singleton, welcher die Session verwaltet. Er ist dafür verantwortlich, dass die Session zentral verwaltet wird. Man kann über den Singleton die Sessionvariable 'user' setzen, diese auslesen oder die Session zerstören.

13.2.2 Database

Database ist ein Singleton, welcher die Datenbankverbindung verwaltet. Er stellt sicher, dass nur eine Datenbankverbindung steht und nicht mehrere gleichzeitig existieren. Über eine Funktion kann von aussen die Verbindung beantragt werden.

13.2.3 RegisterController

Über den Registercontroller werden die Daten des Registrierungsformular aufbereitet, damit sie in die Datenbank geschrieben werden können.

13.2.4 LoginController

Der Logincontroller enthält die Funktionen zum Einloggen und abmelden eines Nutzers.

13.2.5 Validator

Der Validator überprüft die Inputs vom Registrierungsformular und Inseratformular. Ist mindestens ein Eintrag falsch, gibt er false zurück.

13.2.6 InseratController

Der Inseratcontroller stellt die Verbindung zwischen dem Model und der View her. Der Controller verfügt über drei Methoden. Über eine werden die eingegebenen Daten vom Inseratformular für das Model aufbereitet. Über die zweite Methode werden die Seiten auf der Startseite geladen. Diese enthalten maximal 10 Previews von Inseraten. Über die dritte Funktion kann die Anzahl Seiten an 10 Inseraten abgefragt werden.

13.2.7 InseratModel

Das InseratModel liest die Inserat-Daten aus der Datenbank heraus, oder schreibt neue oder geänderte Daten in die Datenbank. Inserate können direkt im zehnerpack geladen werden. Dafür muss man angeben, von welcher Position an gesucht werden soll. Eine weitere Funktion ermöglicht es ein Inserat anhand seiner ID zu laden. Des Weiteren gibt es noch die Funktion um ein neues Inserat in die Datenbank zu speichern.

IPA Bericht	Alain Keller	Version 1.0	30 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

13.3 Gui Screenshots

13.3.1 Startseite

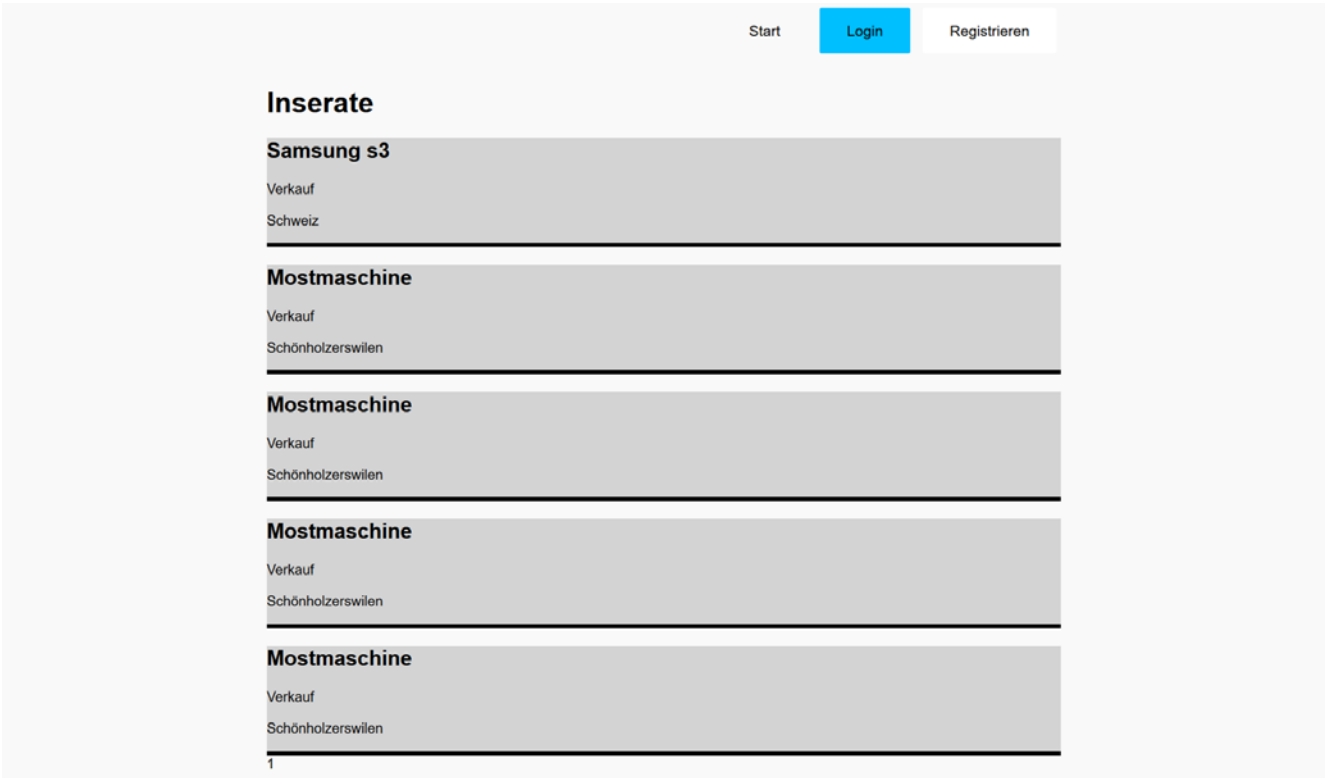


Abbildung 4 Startseite Screenshot

IPA Bericht	Alain Keller	Version 1.0	31 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

13.3.2 Registrierung

Start

Login

Registrieren

Registrieren

Benutzername

Name

Vorname

E-Mail

Telefon

Passwort

Passwort Wiederholen

Abbrechen

Registrieren

Abbildung 5 Registrierungsseite Screenshot

13.3.3 Login

Start

Login

Registrieren

Benutzername

Passwort

Abbrechen

Einloggen

Abbildung 6 Loginseite Screenshot

IPA Bericht	Alain Keller	Version 1.0	32 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

13.3.4 Inserat erstellen

Start

Meine Inserate

Inserat erfassen

Logout

Inserat Erfassen

Titel

Beschreibung

E-Mail

Telefon

Ort

Art des Angebotes

Abbrechen

Inserat erstellen

Abbildung 7 Inserat erstellen Seite

13.3.5 Inserat

Zurück

Mostmaschine

Typ

Verkauf

Beschreibung

Eine Alte Moschtmachine in gutem Zustand steht zum Verkauf. Preis ab 500 CHF.

Ort

Schönholzerswilen

Telefon

076 666 66 66

E-Mail

most@hans.ch

Abbildung 8 Inseratseite

IPA Bericht	Alain Keller	Version 1.0	33 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

14 Kontrollieren

14.1 Testübersicht

Kategorie	Ergebnis Testfälle
Erfolgreich	10
Teilweise Erfolgreich	2
Nicht erfolgreich	2

Tabelle 25: Ergebnis Testfälle

14.2 Testfazit

Das Produkt kann im aktuellen Status nicht veröffentlicht werden. Es mehr als ein Viertel der Testfälle sind nicht oder nur Teilweise bestanden. Da die nicht bestandenen Fällen zu explizit gewünschten Anforderungen gehören, gelten diese als kritisch. Zuerst sollten die fehlenden Funktionen implementiert werden, bevor die Applikation wieder getestet wird und gegebenenfalls veröffentlicht wird.

IPA Bericht	Alain Keller	Version 1.0	34 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

14.3 Testprotokoll

14.3.1 Registrierung

Test Nummer	Fazit	Abweichungen
1.1		-
1.2		Passwörter werden nicht gelöscht und nicht rot umrahmt
1.3		-
1.4		Das Feld Username wird nicht rot umrahmt

Tabelle 26 Registrierung Testprotokoll

14.3.2 Login

Test Nummer	Fazit	Abweichungen
2.1		-
2.2		-
2.3		-

Tabelle 27 Login Testprotokoll

14.3.3 Logout

Test Nummer	Fazit	Abweichungen
3.1		-

Tabelle 28 Logout Testprotokoll

14.3.4 Inserate erstellen

Test Nummer	Fazit	Abweichungen
4.1		-
4.2		-
4.3		-

Tabelle 29 Inserate erstellen Testprotokoll

14.3.5 Inserate ansehen

Test Nummer	Fazit	Abweichungen
5.1		-

Tabelle 30 Inserate ansehen Testprotokoll

14.3.6 Inserate Löschen

Test Nummer	Fazit	Abweichungen
6.1		Seite öffnet nicht
6.2		Kann nicht getestet werden, da die Seite nicht existiert.

Tabelle 31 Inserate Löschen Testprotokoll

IPA Bericht	Alain Keller	Version 1.0	35 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

14.4 Soll – Ist Vergleich

Für die Funktionalen Anforderungen siehe: 11.1

Name	Ist-Abweichung
Registrieren	Bei Falscheingaben wird keine Rückmeldungen zurückgegeben wenn diese vom Server erkannt wurden.
Login	Bei Falscheingaben wird keine Rückmeldungen zurückgegeben wenn diese vom Server erkannt wurden.
Logout	-
Inserat erstellen	Bei Falscheingaben wird keine Rückmeldungen zurückgegeben wenn diese vom Server erkannt wurden.
Inserat Übersicht	-
Inserat öffnen	-
Inserat Löschen	Die Inserat Löschen Funktion wurde wegen Zeitmängel nicht implementiert und sind daher nicht vorhanden.
Adminbereich	Der Adminbereich und seine weiteren Funktionen wurden wegen Zeitmangel nicht implementiert und sind daher nicht vorhanden.

Tabelle 32 Soll Ist vergleich

Die Applikation konnte in der gegebenen Zeit nicht vollständig implementiert werden. Wegen Zeitmängel musste ich gewisse Funktionen streichen, oder konnte sie nicht vollständig implementieren.

IPA Bericht	Alain Keller	Version 1.0	36 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

15 Auswerten

Das Projekt wird Ausgewertet und über die Arbeitsweise reflektiert.

15.1 Reflexion

Das Projekt ist nun abgeschlossen. Es war mein erstes Projekt seit längerem, welches ich alleine von Anfang bis Ende durchführte. Ich bin jedoch nur mässig zufrieden mit dem Resultat. Ich konnte meinen Auftrag nicht komplett abschliessen, habe jedoch eine lauffähige Applikation realisiert. Ich habe in diesem Projekt einige neue Aspekte der Webentwicklung kennengelernt. Ich habe zum Beispiel das erste Mal mit Ajax gearbeitet. Jedoch hat mir Ajax auch einige kleinere Probleme beschert. Mit meiner Architektur bin ich am wenigsten zufrieden. Diese hatte ich in der Planung zu wenig hinterfragt und war dann in der Realisierung nicht sicher ob dies wirklich sinnvoll ist. Dies liegt sicher auch daran, dass ich nie wirklich Produktiverfahrungen im Objektorientiertem Programmieren, sowie Server – Client Kommunikation habe.

15.2 Schlusswort

Wenn ich nachträglich auf die Aufgabenstellung zurückblicke, hätte ich schon früher merken sollen, dass ich das Projekt mit meinem vorherigen Wissen nicht hätte komplett abschliessen können. Es war jedoch eine gute Erfahrung, denn ich kann nun sicher besser einschätzen was ich innerhalb von zwei Tagen realisieren kann. Ich habe während dieser Probe-IPA einige Erfahrungen sammeln können, die mir sowohl in der richtigen IPA als auch in anderen zukünftigen Projekten nützlich sein können.

IPA Bericht	Alain Keller	Version 1.0	37 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

Anhang

16 Abbildungsverzeichnis

Jegliche Abbildungen in diesem Dokument wurden selber erstellt.

Abbildung 1 Usecase Diagramm	22
Abbildung 2 Komponentendiagramm	23
Abbildung 3 Ordnerstruktur	29
Abbildung 4 Startseite Screenshot	31
Abbildung 5 Registrierungsseite Screenshot	32
Abbildung 6 Loginseite Screenshot	32
Abbildung 7 Inserat erstellen Seite	33
Abbildung 8 Inseratseite	33

IPA Bericht	Alain Keller	Version 1.0	38
Inserator	Probe IPA		von
Siemens Schweiz AG	Intern		63

17 Tabellenverzeichnis

Tabelle 1: Beteiligte Personen	5
Tabelle 2: Projektmeilensteine	8
Tabelle 3: Arbeitsprotokoll 01.11.2017	9
Tabelle 4: Arbeitsprotokoll 2.11.2017	11
Tabelle 5: Arbeitsprotokoll 23.03.2017	12
Tabelle 6: Arbeitsprotokoll 24.03.2017	14
Tabelle 7: Arbeitsprotokoll 28.03.2017	15
Tabelle 8 Funktionale Anforderung	19
Tabelle 9 Berechtigungsmatrix	20
Tabelle 10: Use Case #1	20
Tabelle 11: Use Case #2	21
Tabelle 12: Use Case #3	21
Tabelle 13: Use Case #4	21
Tabelle 14: Use Case #5	21
Tabelle 15: Use Case #6	21
Tabelle 16: Use Case #7	22
Tabelle 17 User Datenbank	24
Tabelle 18 Inserat Datenbank	25
Tabelle 19 Registrierung Testfall	26
Tabelle 20: Login Testfall	27
Tabelle 21: Logout Testfall	27
Tabelle 22 Inserate erstellen Testfall	28
Tabelle 23 Inserate ansehen Testfall	28
Tabelle 24 Inserat Löschen Testfall	28
Tabelle 25: Ergebnis Testfälle	34
Tabelle 26 Registrierung Testprotokoll	35
Tabelle 27 Login Testprotokoll	35
Tabelle 28 Logout Testprotokoll	35
Tabelle 29 Inserate erstellen Testprotokoll	35
Tabelle 30 Inserate ansehen Testprotokoll	35
Tabelle 31 Inserate Löschen Testprotokoll	35
Tabelle 32 Soll Ist vergleich	36

IPA Bericht	Alain Keller	Version 1.0	39 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

18 Literaturverzeichnis

Das Literaturverzeichnis beinhaltet alle Webseiten, welche während der Realisierung zur Hilfe verwendet wurden.

Was	Quelle
-----	--------

19 Quellenverzeichnis

Das Quellenverzeichnis beinhaltet alle Dokumente & Webseiten, von welchen Texte/Bilder/Vorlagen/Programmcode übernommen und/oder angepasst wurden.

Was	Quelle
PHP Problembehebungen	http://php.net/manual/de/

20 Glossar

Abkürzung	Definition
Ajax	Asynchrone Dateiübertragung zwischen Client und Server, welcher es ermöglicht, Daten vom Server zu laden ohne die Seite neu zu laden.
Plant UML	Textbasiertes Tool um UMLs zu erstellen

IPA Bericht	Alain Keller	Version 1.0	40 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

21 Source-Code

21.1.1 Index.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Inserator</title>
    <link rel="stylesheet" href="css/style.css">
    <script src="inserat/inserat.js"></script>
    <script src="login/login.js"></script>
    <script src="login/register.js"></script>
    <script src="js/load.js"></script>
    <script src="js/Jquery3.2.1.js"></script>
</head>
<body onload="loadContent()">
    <div id="container">
        <header>
            <div id="headerContent" >
                <div id="headerButtonContainer">
                    <a onclick="showLogin()" class="headerButton"
id="loginButton">Login</a>
                    <a onclick="showRegister()" class="headerButton" id="registerButton"
>Registrieren</a>
                </div>
                <div class="floatClear"></div>
            </div>
        </header>

        <main id="mainContent">

            </main>
        </div>

</body>
</html>
```

21.1.2 registerView.php

```
<?php
/**
 * Created by PhpStorm.
 * User: Alain
 * Date: 03.11.2017
 * Time: 09:31
 */

echo '<div id="LoginContent">
    <h1 id="RegisterTitle">Registrieren</h1>

    <form onsubmit="register(); return false;" id="registerform">
        <label for="RegisterUsername" class="RegisterLabel">Benutzername</label> <br />
        <input type="text" id="RegisterUsername" name="RegisterUsername"
class="RegisterInput" required="required"> <br />

        <label for="Name" class="RegisterLabel">Name</label> <br />
        <input type="text" id="Name" name="Name" class="RegisterInput" > <br />

        <label for="Surname" class="RegisterLabel">Vorname</label> <br />
        <input type="text" id="Surname" name="Surname" class="RegisterInput" > <br />

        <label for="Mail" class="RegisterLabel">E-Mail</label> <br />
```

IPA Bericht	Alain Keller	Version 1.0	41 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

```

<input type="email" id="Mail" name="Mail" class="RegisterInput"> <br />

<label for="Phone" class="RegisterLabel">Telefon</label> <br />
<input type="tel" id="Phone" name="Phone" class="RegisterInput"> <br />

<label for="RegisterPassword" class="RegisterLabel">Passwort</label> <br />
<input pattern="^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[$@!%*?&_-])[A-Za-z\d$@!%*?&_-]{8,}"
    title="Mindestens 8 Zeichen lang. Inklusive Sonderzeichen, Grossbuchstaben
und Zahlen." type="password"
    id="RegisterPassword" name="RegisterPassword" class="RegisterInput"> <br/>

<label for="RepPassword" class="RegisterLabel">Passwort Wiederholen</label> <br />
<input pattern="^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[$@!%*?&_-])[A-Za-z\d$@!%*?&_-]{8,}"
    title="Mindestens 8 Zeichen lang. Inklusive Sonderzeichen, Grossbuchstaben
und Zahlen." type="password"
    id="RepPassword" name="RepPassword" class="RegisterInput" > <br/>

<input type="reset" name="Abbrechen" class="ResetButton" id="Reset"
value="Abbrechen">
<input type="submit" name="Registrieren" class="SubmitButton" id="Submit"
value="Registrieren">

</form>
</div>';

```

21.1.3 loginView.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Alain
 * Date: 03.11.2017
 * Time: 10:56
 */

echo '<form id="loginform" onsubmit="login(); return false;">

    <label for="Username" class="LoginLabel">Benutzername</label> <br />
    <input type="text" id="Username" name="Username" class="LoginInput"
required="required"> <br />

    <label for="Password" class="LoginLabel">Passwort</label> <br />
    <input type="password" id="Password" name="Password" class="LoginInput"
required="required"> <br />

    <input type="reset" class="ResetButton" id="ResetLogin" value="Abbrechen">
    <input type="submit" class="SubmitButton" id="SubmitLogin"
value="Einloggen">

</form>
<div id="loginErrorContainer"></div>';

```

21.1.4 HeaderView

```

<?php
/**
 * Created by PhpStorm.
 * User: Alain
 * Date: 03.11.2017
 * Time: 09:31
 */

require_once "../model/user.php";

```

IPA Bericht	Alain Keller	Version 1.0	42
Inserator	Probe IPA		von
Siemens Schweiz AG	Intern		63

```

require_once "../controller/mySession.php";

$currentSession = mySession::getInstance();

$user = $currentSession->getCurrentUser();

if($user) {
    echo "
        <div>

        </div>
        <div id=\"headerButtonContainer\">
            <a onclick=\"loadContent()\" class=\"headerButton\"
id=\"myInserateButton\">Start</a>
            <a onclick=\"showMyInserate()\" class=\"headerButton\"
id=\"myInserateButton\">Meine Inserate</a>
            <a onclick=\"openInseratForm()\" class=\"headerButton\"
id=\"makeInseratButton\">Inserat erfassen</a>
            <a onclick=\"logout()\" class=\"headerButton\" id=\"logoutButton\">Logout</a>
        </div>
        <div class=\"floatClear\"></div>;
    } else {
        echo '<div id="headerButtonContainer">
            <a onclick="loadContent()" class="headerButton" id="myInserateButton">Start</a>
            <a onclick="showLogin()" class="headerButton" id="loginButton">Login</a>
            <a onclick="showRegister()" class="headerButton"
id="registerButton">Registrieren</a>
        </div>
        <div class="floatClear"></div>';
    }
}

```

21.1.5 createInserateView.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Alain
 * Date: 03.11.2017
 * Time: 09:31
 */

echo '<div id="LoginContent">
    <h1 id="RegisterTitle">Inserat Erfassen</h1>

    <form onsubmit="createInserat(); return false;" id="inseratForm">
        <label for="Title" class="RegisterLabel">Titel</label> <br />
        <input type="text" id="Title" name="Title" class="RegisterInput"
required="required"> <br />

        <label for="Description" class="RegisterLabel">Beschreibung</label> <br />

        <textarea rows="4" cols="50" id="Description" name="Description" maxlength="500">
        </textarea><br />

        <label for="Mail" class="RegisterLabel">E-Mail</label> <br />
        <input type="email" id="Mail" name="Mail" class="RegisterInput"> <br />

        <label for="Phone" class="RegisterLabel">Telefon</label> <br />
        <input type="tel" id="Phone" name="Phone" class="RegisterInput"> <br />

        <label for="Place" class="RegisterLabel">Ort</label> <br />
        <input type="text" id="Place" name="Place" class="RegisterInput"> <br />

        <label for="Type" class="RegisterLabel">Art des Angebotes</label> <br />
        <input type="text" id="Type" name="Type" class="RegisterInput"> <br />
    </form>
</div>'

```

IPA Bericht	Alain Keller	Version 1.0	43 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

```

        <input type="reset" name="Abbrechen" class="ResetButton" id="Reset"
value="Abbrechen">
        <input type="submit" name="Registrieren" class="SubmitButton" id="Submit"
value="Inserat erstellen">

    </form>
</div>';

```

21.1.6 allInserateView.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Alain
 * Date: 03.11.2017
 * Time: 09:33
 */
require_once '../inserat/InseratController.php';

$currentPage = (int)filter_input(INPUT_GET, 'CurrentPage', FILTER_SANITIZE_NUMBER_INT) ??
1;

$controller = new InseratController();

$maxPages = $controller->getNumberOfPages();

$insertate = $controller->showInserate($currentPage);

$buttons = "";

$buttons = $currentPage == 1 ? "" : "<div> <a class='navigationButton'
onclick='previousPage()'>Vorherige Seite</a></div>";
$buttons = $currentPage == $maxPages ? $buttons."<div> <a
class='navigationButton' onclick='nextPage()'>Nächste Seite</a></div>";

echo "    <div >
        <h1>Inserate</h1>
    </div>
    <div id=\"inserateContainer\"/>
        $insertate
    </div>
    <div> $buttons $maxPages</div>";

```

21.1.7 userModul.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Alain
 * Date: 03.11.2017
 * Time: 08:48
 */
require_once "database.php";
require_once "user.php";
require_once "../controller/mySession.php";

class userModel
{
    private $connection;
    private $mySession ;
}

```

IPA Bericht	Alain Keller	Version 1.0	44 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

```

public function __construct()
{
    $this->connection = database::getConnection();
    $this->mySession = mySession::getInstance();
}

public function registerUser(string $username, string $password, string $surname,
string $name, string $mail, string $phone) {
    $hashedpassword = password_hash($password, PASSWORD_BCRYPT);

    try
    {
        $this->connection->beginTransaction();
        $sql = "INSERT INTO `inserter`.`user` (`first_name`, `name`, `username`,
`email`, `phone`, `pwd`) VALUES ('$surname', '$name', '$username', '$mail', '$phone',
'$hashedpassword')";
        $stmt = $this->connection->prepare($sql); // Prevent MySQL injection. $stmt
means statement
        $stmt->execute();
        $this->connection->commit();

    } catch (PDOException $e) {
        $this->connection->rollBack();
        echo "ERROR";
        return false;
    }

    // $user = new user($name, $surname, $username, $mail, $phone);

    $user = $this->loadUserByUsername($username);

    $this->mySession->setCurrentUser($user);

    return true;
}

public function isUserExisting(string $username) {
    /* $sql = "SELECT id FROM user WHERE username = '$username'";
    $stmt = $this->connection->prepare($sql); // Prevent MySQL injection. $stmt means
statement
    $stmt->execute();
    */

    /*
    if ( $stmt->() >=1)
    {
        return true;
    }
    */

    return true;
}

public function loadUserByUsername($username)
{
    $sql = "SELECT * FROM user WHERE username = '$username'";
    $stmt = $this->connection->prepare($sql); // Prevent MySQL injection. $stmt means
statement
    $stmt->execute();

    if ($stmt->rowCount() <= 0) {
        echo 'Error';
        return null;
    } else {

```

IPA Bericht	Alain Keller	Version 1.0	45 von 63
Inserter	Probe IPA		
Siemens Schweiz AG	Intern		

```

        return $stmt->fetch();
    }

}
}

```

21.1.8 database.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Alain
 * Date: 03.11.2017
 * Time: 11:05
 */
class database
{
    private static $instance;
    private $conn;

    /**
     * Database constructor.
     */
    private function __construct()
    {
        try
        {
            $this->conn = new PDO('mysql:host=localhost;dbname=inserator', 'root', '');
        }
        catch (PDOException $e)
        {
            echo 'Error: ' . $e->getMessage();
            exit();
        }
    }

    public static function getConnection()
    {
        return self::getInstance()->conn;
    }

    public static function getInstance()
    {
        if (!self::$instance) { // If no instance then create one
            self::$instance = new self();
        }
        return self::$instance;
    }
}

```

21.1.9 registerInput.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Alain
 * Date: 03.11.2017
 * Time: 09:43
 */

require_once "registerController.php";

```

IPA Bericht	Alain Keller	Version 1.0	46 von 63
Inserator	Probe IPA		
Siemens Schweiz AG	Intern		

```

/**
 * Saintizes user Inputs, andgive it to the regsiterController
 */

$username = filter_input(INPUT_GET, 'Username', FILTER_SANITIZE_STRING) ?? "";
$name = filter_input(INPUT_GET, 'Name', FILTER_SANITIZE_STRING) ?? "";
$surname = filter_input(INPUT_GET, 'Surname', FILTER_SANITIZE_STRING) ?? "";
$mail = filter_input(INPUT_GET, 'Mail', FILTER_SANITIZE_EMAIL) ?? "";
$phone = filter_input(INPUT_GET, 'Phone', FILTER_SANITIZE_STRING) ?? "";
$password = filter_input(INPUT_GET, 'Password') ?? "";
$repeatPassword = filter_input(INPUT_GET, 'RepPassword') ?? "";

$controller = new registerController();

$controller->registerUser($username, $password, $repeatPassword, $surname, $name, $mail,
$phone);

```

21.1.10 registerController

<?php

```

/**
 * Created by PhpStorm.
 * User: Alain
 * Date: 03.11.2017
 * Time: 09:41
 */
require_once "../model/userModel.php";
require_once "../controller/validator.php";

/**
 * Class registerController
 * Manages methods used for register an User
 */
class registerController
{
    private $model;
    private $validator;

    /**
     * registerController constructor.
     */
    public function __construct()
    {
        $this->model = new userModel();
        $this->validator = new validator();
    }

    /**
     * Manages the Userinput to register User
     * @param $username
     * @param $password
     * @param $repeatPassword
     * @param $surname
     * @param $name
     * @param $mail
     * @param $phone
     */
    public function registerUser($username, $password, $repeatPassword, $surname, $name,
$mail, $phone)
    {
        $isInputValid = $this->validator->registerInputValid($username, $password,
$repeatPassword, $surname, $name, $mail, $phone);
    }

```

| | | | |
|--------------------|--------------|----------------|-----------------|
| IPA Bericht | Alain Keller | Version
1.0 | 47
von
63 |
| Inserator | Probe IPA | | |
| Siemens Schweiz AG | Intern | | |

```
        if(!$isInputValid){
            echo "FEHLER";
            return;
        }

        if($this->model->registerUser($username, $password, $surname, $name, $mail,
$phone)) {
            echo 'Success';
        }

    }

}
```

21.1.11 register.js

```
/**
 * Created by Alain on 03.11.2017.
 */

/**
 * Sow register form
 * Invoked by RegsiterButton in header
 */
function showRegister() {
    $.ajax({
        type: "GET",
        url: 'view/registerView.php',
        cache: false,
        success: function(data){
            $('#mainContent').empty();
            $('#mainContent').append(data);
        },
        error: function (request, status, error) {
            //Shake animation effect.
            alert(error);
        }
    });
}

/**
 * Sends inputed registerdata to server for further validation
 * invoked by registerbutton in registerform
 */
function register() {

    var username=$('#RegisterUsername').val();
    var name=$('#Name').val();
    var surname=$('#Surname').val();
    var mail=$('#Mail').val();
    var phone=$('#Phone').val();
    var password=$('#RegisterPassword').val();
    var repPassword=$('#RepPassword').val();

    var dataString =
'Username='+username+'&Name='+name+'&Surname='+surname+'&Mail='+mail+'&Phone='+phone+'&Pass
word='+password+'&RepPassword='+repPassword;

    $.ajax({
        type: "GET",
        url: "login/registerInput.php",
        data: dataString,
        cache: false,
```

| | | | |
|--------------------|--------------|----------------|-----------------|
| IPA Bericht | Alain Keller | Version
1.0 | 48
von
63 |
| Inserator | Probe IPA | | |
| Siemens Schweiz AG | Intern | | |


```

        success: function(data){

            if(data == "Success"){
                loadContent();
            }
        },
        error: function (request, status, error) {
            //Shake animation effect.
        }
    });
}

```

21.1.12 logout.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Alain
 * Date: 08.11.2017
 * Time: 08:38
 */

require_once 'LoginController.php';

$loginController = new LoginController();

$loginController->logout();

require_once '../view/allInserateView.php';

```

21.1.13 loginInput.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Alain
 * Date: 08.11.2017
 * Time: 08:49
 */

require_once "LoginController.php";

$username = filter_input(INPUT_GET, 'Username', FILTER_SANITIZE_STRING) ?? "";
$password = filter_input(INPUT_GET, 'Password') ?? "";

$controller = new LoginController();

$controller->loginPerson($username, $password);

```

21.1.14 LoginController.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Alain
 * Date: 03.11.2017
 * Time: 09:27
 */

require_once '../controller/mySession.php';
require_once '../model/userModel.php';

```

| | | | |
|---------------------------|---------------|----------------|-----------------|
| IPA Bericht | Alain Keller | Version
1.0 | 49
von
63 |
| Inserator | Probe IPA | | |
| Siemens Schweiz AG | Intern | | |

```

class LoginController
{
    private $customSession;
    private $model;

    public function loginPerson($username, $password)
    {
        $user = $this->model->loadUserByUsername($username);

        if($user) {
            $passwordCorrect = password_verify($password, $user['pwd']);

            if ($passwordCorrect) {

                $this->customSession->setCurrentUser($user);
                echo 'Success';
                return;
            }

            //Code wrong
            $this->loginError();
            return;
        }

        public function __construct()
        {
            $this->customSession = mySession::getInstance();
            $this->model = new userModel();
        }

        public function logout() {
            $this->customSession->destroySession();
        }

        private function loginError()
        {
            echo 'error';
        }
    }
}

```

21.1.15 Login.js

```

/**
 * Created by Alain on 03.11.2017.
 */

/**
 * Shows Loginformular for the User to login
 * Called when Loginbutton is clicked
 */
function showLogin() {
    $.ajax({
        type: "GET",
        url: 'view/loginView.php',
        cache: false,
        success: function(data){
            $('#mainContent').empty();
            $('#mainContent').append(data);
        },
        error: function (request, status, error) {
            //Shake animation effect.
            alert(error);
        }
    });
}

```

| | | | |
|---------------------------|---------------|----------------|-----------------|
| IPA Bericht | Alain Keller | Version
1.0 | 50
von
63 |
| Inserator | Probe IPA | | |
| Siemens Schweiz AG | Intern | | |

```
}

/**
 * sends inputed data to server for further validation
 * Invoked by Loginbutton in Loginform
 */
function login() {
    var username=$("#Username").val();
    var password=$("#Password").val();

    var dataString = 'Username='+username+ '&Password='+password;

    $.ajax({
        type: "GET",
        url: "login/loginInput.php",
        data: dataString,
        cache: false,
        success: function(data){

            if(data == "Success"){
                loadContent();
            }
            else {
                errorPrint = "<p>Login Fehlgeschlagen</p>";
                $('#loginErrorContainer').empty();
                $('#loginErrorContainer').append(errorPrint);
            }
        },
        error: function (request, status, error) {

        }
    });
}

/**
 * Sends request to Server for the User to get logged out
 */
function logout() {
    $.ajax({
        type: "GET",
        url: 'login/logout.php',
        cache: false,
        success: function(data){
            loadContent();
        },
        error: function (request, status, error) {
            //Shake animation effect.
            alert(error);
        }
    });
}
}
```

21.1.16 Load.js

```
/**
 * Created by Alain on 03.11.2017.
 */

/**
 * Loads pagecontent on first start or opening the mainpage
 */
function loadContent(){
    $.ajax({
        type: "GET",
        url: "view/headerView.php",
        cache: false,
```

| | | | |
|--------------------|--------------|----------------|-----------------|
| IPA Bericht | Alain Keller | Version
1.0 | 51
von
63 |
| Inserator | Probe IPA | | |
| Siemens Schweiz AG | Intern | | |

```

        success: function(data){
            $('#headerContent').empty();
            $('#headerContent').append(data);
        },
        error: function (request, status, error) {
            //Shake animation effect.
        }
    });

    sessionStorage.setItem("CurrentPage", 1);
    // sessionStorage.getItem("CurrentPage");
    loadInserate();
}

```

21.1.17 inseratModel.php

```

<?php

/**
 * Created by PhpStorm.
 * User: Alain
 * Date: 03.11.2017
 * Time: 08:48
 */

require_once '../model/database.php';
require_once '../controller/mySession.php';

/**
 * Class inseratModel
 * Datamodel Responsilbe for getting Inserat-datas from database and return it to
 * InseratController
 */
class inseratModel
{

    private $connection;
    private $mySession;

    /**
     * inseratModel constructor.
     */
    public function __construct()
    {
        $this->connection = database::getConnection();
    }

    /**
     * Inserts Inserat into Database
     * @param $title
     * @param $description
     * @param $mail
     * @param $phone
     * @param $place
     * @param $type
     * @return bool
     */
    public function insertInserat($title, $description, $mail, $phone, $place, $type)
    {

        $this->mySession = mySession::getInstance();

        $user = $this->mySession->getCurrentUser();
    }
}

```

| | | | |
|---------------------------|---------------|----------------|-----------------|
| IPA Bericht | Alain Keller | Version
1.0 | 52
von
63 |
| Inserator | Probe IPA | | |
| Siemens Schweiz AG | Intern | | |

```

$date = date("Y-m-d H:i:s");
$userId = $user['id'];
$isActive = true;

try
{
    $this->connection->beginTransaction();
    $sql = "INSERT INTO `inserter`.`inserter` (`name`, `type`, `description`,
`location`, `email`, `phone`, `date`, `is_active`, `user_id`) VALUES ('$title', '$type',
'$description', '$place', '$mail', '$phone', '$date', '$isActive', '$userId')";
    $stmt = $this->connection->prepare($sql); // Prevent MySQL injection. $stmt
means statement
    $stmt->execute();
    $this->connection->commit();
} catch (mysqli_sql_exception $e) {
    $this->connection->rollBack();
    echo "ERROR";
    return false;
}
echo "Success";
}

/**
 * Loads 10 Inserter from a defined Start, ordered by Date
 * @param int $start
 * @return string
 */
public function loadInserter(int $start)
{
    $sql = "SELECT inserat.name, inserat.type, inserat.location, inserat.id FROM
inserter ORDER BY inserat.date DESC LIMIT $start,10";
    $stmt = $this->connection->prepare($sql); // Prevent MySQL injection. $stmt means
statement
    $stmt->execute();

    $outputString = "";

    while ($row = $stmt->fetch())
    {
        $name = $row['name'];
        $type = $row['type'];
        $location = $row['location'];
        $id = $row['id'];

        $outputString = $outputString."<div class='inserterPreview'>
            <a href='inserter/inserter.php?id=$id'>
            <h2>$name</h2>
            <p>$type</p>
            <p>$location</p></a>
        </div>";
    }

    return $outputString;
}

/**
 * Returns a specific Inserter from the database, identified by its ID
 * @param $id
 * @return mixed
 */
public function loadInserterById($id)
{
    $sql = "SELECT inserat.name, inserat.type, inserat.location, inserat.description,
inserter.email, inserat.phone FROM inserat where inserat.id = '$id' and is_active = 1";

```

| | | | |
|---------------------------|---------------|----------------|-----------------|
| IPA Bericht | Alain Keller | Version
1.0 | 53
von
63 |
| Inserter | Probe IPA | | |
| Siemens Schweiz AG | Intern | | |

```

        $stmt = $this->connection->prepare($sql); // Prevent MySQL injection. $stmt means
statement
        $stmt->execute();

        return $stmt->fetch();
    }

    /**
     * Returns total ammount of active Inserate stored in the Database.
     * @return mixed
     */
    public function loadInserateTotal()
    {
        $sql = "SELECT count(id) as 'count' FROM inserat where is_active = 1";
        $stmt = $this->connection->prepare($sql); // Prevent MySQL injection. $stmt means
statement
        $stmt->execute();

        return $stmt->fetch();
    }
}

```

21.1.18 inseratInput.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Alain
 * Date: 08.11.2017
 * Time: 09:48
 */

require_once 'InseratController.php';

/**
 * Sanitizes input from new Inserat form and passing it forward to controller
 */

$title = filter_input(INPUT_GET, 'Title', FILTER_SANITIZE_STRING) ?? "";
$description = filter_input(INPUT_GET, 'Description', FILTER_SANITIZE_STRING) ?? "";
$mail = filter_input(INPUT_GET, 'Mail', FILTER_SANITIZE_EMAIL) ?? "";
$phone = filter_input(INPUT_GET, 'Phone', FILTER_SANITIZE_STRING) ?? "";
$place = filter_input(INPUT_GET, 'Place', FILTER_SANITIZE_STRING) ?? "";
$type = filter_input(INPUT_GET, 'Type', FILTER_SANITIZE_STRING) ?? "";

$mySession = mySession::getInstance();
if($mySession->getCurrentUser()) {
    $controller = new InseratController();

    $controller->createInserat($title, $description, $mail, $phone, $place, $type);
} else {
    echo 'not authenticated';
}

```

21.1.19 InseratController.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Alain
 * Date: 08.11.2017
 * Time: 09:48
 */

```

| | | | |
|---------------------------|---------------|----------------|-----------------|
| IPA Bericht | Alain Keller | Version
1.0 | 54
von
63 |
| Inserator | Probe IPA | | |
| Siemens Schweiz AG | Intern | | |

```

*/

require_once 'inseriratModel.php';
require_once '../controller/validator.php';

/**
 * Class InseratController
 * Controller for methods concerning Inserate
 */
class InseratController
{
    private $model;
    private $validator;

    /**
     * InseratController constructor.
     */
    public function __construct()
    {
        $this->model = new inseratModel();
    }

    /**
     * Manages the creation of a new Inserat
     * @param string $title
     * @param string $description
     * @param string $mail
     * @param string $phone
     * @param string $place
     * @param string $type
     */
    public function createInserat(string $title, string $description, string $mail, string
    $phone, string $place, string $type)
    {
        $this->validator = new validator();
        $isInputValid = $this->validator->inseratInputValid($title, $description, $mail,
    $phone, $place, $type);
        if($isInputValid)
        {
            $this->model->insertInserat($title, $description, $mail, $phone, $place,
    $type);
        }
    }

    /**
     * Loads Inserate from Model and pass it to the View
     * @param $page
     * @return string
     */
    public function showInserate($page)
    {
        $page = (int)($page * 10) - 10;
        $inserate = $this->model->loadInserate($page);
        return $inserate;
    }

    /**
     * Returns total number of pages if pagelength is equals 10
     * @return int
     */
    public function getNumberOfPages()
    {
        $result = $this->model->loadInserateTotal();

        $numberOfInserates = $result['count'];
    }
}

```

| | | | |
|---------------------------|---------------|----------------|-----------------|
| IPA Bericht | Alain Keller | Version
1.0 | 55
von
63 |
| Inserator | Probe IPA | | |
| Siemens Schweiz AG | Intern | | |

```

        $numberOfPages = $numberOfInserates % 10 >= 0 ? ($numberOfInserates / 10) + 1 :
$numberOfInserates / 10;

        return (int)$numberOfPages;
    }

}

```

21.1.20 Inserat.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Alain
 * Date: 08.11.2017
 * Time: 13:55
 */

require_once 'InseratController.php';

$model = new inseratModel();

$id = $_GET['id'];

$data = $model->loadInserateById($id);

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Inserator</title>
    <link rel="stylesheet" href="../../css/style.css">
    <script src="inserat.js"></script>
    <script src="../../login/login.js"></script>
    <script src="../../login/register.js"></script>
    <script src="../../js/load.js"></script>
    <script src="../../js/Jquery3.2.1.js"></script>
</head>
<body onload="loadContent()">
    <header>
        <div id="headerContent">
            <a class="headerButton" id="loginButton" href="../../index.php">Zurück</a>
            <div class="floatClear"></div>
        </div>
    </header>

    <main id="mainContent">
        <h2><?php echo $data['name']; ?></h2>
        <h2>Typ</h2>
        <p><?php echo $data['type']; ?></p>
        <h2>Beschreibung</h2>
        <p><?php echo $data['description']; ?></p>
        <h2>Ort</h2>
        <p><?php echo $data['location']; ?></p>
        <h2>Telefon</h2>
        <p><?php echo $data['phone']; ?></p>
        <h2>E-Mail</h2>
        <p><?php echo $data['email']; ?></p>

    </main>
</body>
</html>

```

| | | | |
|---------------------------|---------------|----------------|-----------------|
| IPA Bericht | Alain Keller | Version
1.0 | 56
von
63 |
| Inserator | Probe IPA | | |
| Siemens Schweiz AG | Intern | | |

21.1.21 Inserat.js

```

/**
 * Created by Alain on 03.11.2017.
 */

/**
 * Calls php method for creating new Inserat
 */
function createInserat() {
    var title=$("#Title").val();
    var description=$("#Description").val();
    var phone=$("#Phone").val();
    var mail=$("#Mail").val();
    var place=$("#Place").val();
    var type=$("#Type").val();

    var dataString =
'Title='+title+'&Description='+description+'&Mail='+mail+'&Phone='+phone+'&Place='+place+'&
Type='+type;

    $.ajax({
        type: "GET",
        url: "inserat/inseratInput.php",
        data: dataString,
        cache: false,
        success: function(data){

            if(data == "Success"){
                alert('Inserat wurde erstellt');
                loadContent();
            } else {
                alert("Fehlgeschlagen");
            }
        },
        error: function (request, status, error) {
            //Shake animation effect.
        }
    });
}

/**
 * Opens php method for showing the new Inserat Formular
 */
function openInseratForm() {
    $.ajax({
        type: "GET",
        url: 'view/createInseratView.php',
        cache: false,
        success: function(data){
            $('#mainContent').empty();
            $('#mainContent').append(data);
        },
        error: function (request, status, error) {
            //Shake animation effect.
            alert(error);
        }
    });
}

/**
 * Gos to next page in the Mainpage
 */
function nextPage() {
    var currentPage = sessionStorage.getItem("CurrentPage");

```

| | | | |
|---------------------------|---------------|----------------|-----------------|
| IPA Bericht | Alain Keller | Version
1.0 | 57
von
63 |
| Inserator | Probe IPA | | |
| Siemens Schweiz AG | Intern | | |

```
        sessionStorage.setItem("CurrentPage", parseInt(currentPage) + 1);
        loadInserate();
    }

    /**
     * Go to previous page in the Mainpage
     */
    function previousPage() {
        var currentPage = sessionStorage.getItem("CurrentPage");
        sessionStorage.setItem("CurrentPage", parseInt(currentPage) - 1);
        loadInserate();
    }

    /**
     * Loads the Inserate for one page from the Server
     */
    function loadInserate() {

        var currentPage = sessionStorage.getItem("CurrentPage");

        var dataString = 'CurrentPage='+currentPage;

        $.ajax({
            type: "GET",
            url: "view/allInserateView.php",
            data: dataString,
            cache: false,
            success: function(data){

                $('#mainContent').empty();
                $('#mainContent').append(data);
            },
            error: function (request, status, error) {
                //Shake animation effect.
            }
        });
    }
}
```

21.1.22 Style.css

```
html {
    background-color: #f9f9f9;
}
#container {
    background-color: #f9f9f9;
}

body {
    margin: 0;
    font-family: Arial, "Century Gothic";
    background-color: #f9f9f9;
}
a {
    text-decoration: none;
    color: black;
}
header {
    width: 60%;
    min-height: 50px;
    background: #f9f9f9;
    margin: 0 auto;
}

main {
    width: 60%;
}
```

| | | | |
|--------------------|--------------|----------------|-----------------|
| IPA Bericht | Alain Keller | Version
1.0 | 58
von
63 |
| Inserator | Probe IPA | | |
| Siemens Schweiz AG | Intern | | |

```

    min-height: 500px;
    background: #f9f9f9;
    margin: 0 auto;
    padding: 10px;
}

#headerButtonContainer {
    float: right;
}

.headerButton {
    -moz-border-radius: 2px;
    -webkit-border-radius: 2px;
    border-radius: 2px;
    display: inline-block;
    cursor: pointer;
    color: black;
    font-size: 17px;
    padding: 16px 31px;
    text-decoration: none;
    margin: 5px;
}

.SubmitButton {
    background-color: deepskyblue;
    border: none;
    height: 30px;
    margin-top: 20px;
}

.ResetButton {
    background-color: white;
    border: none;
    height: 30px;
    margin-top: 30px;
}

#loginButton {
    background: deepskyblue;
}

#registerButton {
    background: white;
}

.floatClear {
    clear: both;
}

.inseratPreview {
    background: lightgrey;
    border-bottom: solid black 5px;
}

```

21.1.23 Validator.php

```

<?php

/**
 * Created by PhpStorm.
 * User: Alain
 * Date: 03.11.2017
 * Time: 14:17
 */
require_once "../model/userModel.php";

```

| | | | |
|---------------------------|---------------|----------------|-----------------|
| IPA Bericht | Alain Keller | Version
1.0 | 59
von
63 |
| Inserator | Probe IPA | | |
| Siemens Schweiz AG | Intern | | |

```

/**
 * Class validator
 * Class to manage the Input Validators
 */
class validator
{
    /**
     * Validates the inputs used for register
     * @param string $username
     * @param string $password
     * @param string $repeatPassword
     * @param string $surname
     * @param string $name
     * @param string $mail
     * @param string $phone
     * @return bool
     */
    public function registerInputValid(string $username, string $password, string
    $repeatPassword, string $surname, string $name, string $mail, string $phone)
    {
        $model = new userModel();
        $passwordRegularExpression = '/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[$@!%*?&_-])[A-
Za-z\d$@!%*?&_-]{8,}/';

        $usernameValid = strlen($username) < 40 && strlen($username) > 1;

        $usernameUnique = $model->isUserExisting($username);

        $passwordValid = $password == $repeatPassword; //&&
preg_match($passwordRegularExpression, $password);

        $surnameValid = strlen($surname) < 40 && strlen($surname) > 1;

        $nameValid = strlen($name) < 40 && strlen($name) > 1;

        $mailValid = filter_var($mail, FILTER_VALIDATE_EMAIL);

        $phoneValid = strlen($phone) <= 20 && strlen($phone) >= 10;

        $validationResults = array('Username' => $usernameValid, 'Username2'
=>$usernameUnique, 'Password' => $passwordValid, 'Surname' => $surnameValid, 'Name' =>
$nameValid, 'Email' => $mailValid != "", 'Phone' => $phoneValid);

        foreach ($validationResults as $key => $value) {
            if(!$value) {
                echo $key;
                return false;
                //TODO: manage the faulty input
            }
        }
        return true;
    }
}

/**
 * Validates the Input used for creating a new Inserat
 * @param string $title
 * @param string $description
 * @param string $mail
 * @param string $phone
 * @param string $place
 * @param string $type
 * @return bool
 */
public function inseratInputValid(string $title, string $description, string $mail,
string $phone, string $place, string $type)
{
    $titleValid = strlen($title) < 50 && strlen($title) > 1;

```

| | | | |
|---------------------------|---------------|----------------|-----------------|
| IPA Bericht | Alain Keller | Version
1.0 | 60
von
63 |
| Inserator | Probe IPA | | |
| Siemens Schweiz AG | Intern | | |

```

        $descriptionValid = strlen($description) < 500 && strlen($description) > 1;
        $mailValid = filter_var($mail, FILTER_VALIDATE_EMAIL);
        $phoneValid = strlen($phone) <= 20 && strlen($phone) >= 10;
        $placeValid = strlen($place) < 50 && strlen($place) > 1;
        $typeValid = strlen($type) < 45 && strlen($type) > 1;

        $validationResults = array('Title' => $titleValid, 'Description'
=>$descriptionValid, 'mail' => $mailValid != "", 'Phone' => $phoneValid, 'Place' =>
$placeValid, 'Type' => $typeValid);

        foreach ($validationResults as $key => $value) {
            if(!$value) {
                echo $key;
                return false;
            }
        }
        return true;
    }
}

```

21.1.24 mySession.php

```

<?php

/**
 * Created by PhpStorm.
 * User: Alain
 * Date: 04.11.2017
 * Time: 18:19
 */

/**
 * Class mySession
 * Singleton to manage the Session
 */
class mySession
{
    private static $instance;

    /**
     * SessionHandler constructor.
     */
    private function __construct() {
        if (session_status() == PHP_SESSION_NONE) {
            session_start();
        }
    }

    /**
     * Returns instance of itself. If an Instance already exists return this
     * @return mySession
     */
    public static function getInstance()
    {
        if (!self::$instance) { // If no instance then create one
            self::$instance = new self();
        }
        return self::$instance;
    }

    /**
     * Set the session-value for current user
     */
}

```

| | | | |
|---------------------------|---------------|----------------|-----------------|
| IPA Bericht | Alain Keller | Version
1.0 | 61
von
63 |
| Inserator | Probe IPA | | |
| Siemens Schweiz AG | Intern | | |

```

    * @param $user User to be stored in the sessionfield 'user'
    */
    public function setCurrentUser($user)
    {
        $_SESSION['CurrentUser'] = $user;
    }

    /**
     * Returns the set user of the session. If it doesn't exists return null
     * @return $_SESSION['CurrentUser']
     */
    public function getCurrentUser()
    {
        return isset($_SESSION['CurrentUser']) ? $_SESSION['CurrentUser'] : null;
    }

    /**
     * Destroys current session
     */
    public function destroySession()
    {
        if (session_status() == PHP_SESSION_NONE) {
            session_start();
        }

        session_destroy();
    }
}

```

21.1.25 Inserter.sql

```

-- MySQL Script generated by MySQL Workbench
-- Wed Nov  8 12:21:59 2017
-- Model: New Model    Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

-- -----
-- Schema Inserter
-- -----

-- -----
-- Schema Inserter
-- -----

CREATE SCHEMA IF NOT EXISTS `Inserter` DEFAULT CHARACTER SET utf8 ;
USE `Inserter` ;

-- -----
-- Table `Inserter`.`user`
-- -----

DROP TABLE IF EXISTS `Inserter`.`user` ;

CREATE TABLE IF NOT EXISTS `Inserter`.`user` (
    `id` INT NOT NULL AUTO_INCREMENT,
    `first_name` VARCHAR(20) NOT NULL,
    `name` VARCHAR(20) NOT NULL,
    `username` VARCHAR(20) NOT NULL,
    `email` VARCHAR(50) NOT NULL,
    `phone` VARCHAR(20) NOT NULL,
    `pwd` CHAR(60) NOT NULL,
    `is_admin` TINYINT NOT NULL,
    PRIMARY KEY (`id`),
    UNIQUE INDEX `username_UNIQUE` (`username` ASC))

```

| | | | |
|--------------------|--------------|----------------|-----------------|
| IPA Bericht | Alain Keller | Version
1.0 | 62
von
63 |
| Inserter | Probe IPA | | |
| Siemens Schweiz AG | Intern | | |

```
ENGINE = InnoDB;

-- -----
-- Table `Inserator`.`inserat`
-- -----
DROP TABLE IF EXISTS `Inserator`.`inserat` ;

CREATE TABLE IF NOT EXISTS `Inserator`.`inserat` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(50) NOT NULL,
  `type` VARCHAR(45) NOT NULL,
  `description` VARCHAR(500) NOT NULL,
  `location` VARCHAR(50) NOT NULL,
  `email` VARCHAR(50) NOT NULL,
  `phone` VARCHAR(20) NOT NULL,
  `date` DATETIME NOT NULL,
  `is_active` TINYINT NOT NULL,
  `user_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_inserate_user_idx` (`user_id` ASC),
  CONSTRAINT `fk_inserate_user`
    FOREIGN KEY (`user_id`)
      REFERENCES `Inserator`.`user` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

USE `Inserator` ;

-- -----
-- Placeholder table for view `Inserator`.`view1`
-- -----
CREATE TABLE IF NOT EXISTS `Inserator`.`view1` (`id` INT);

-- -----
-- View `Inserator`.`view1`
-- -----
DROP VIEW IF EXISTS `Inserator`.`view1` ;
DROP TABLE IF EXISTS `Inserator`.`view1`;
USE `Inserator`;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

| | | | |
|--------------------|--------------|----------------|-----------------|
| IPA Bericht | Alain Keller | Version
1.0 | 63
von
63 |
| Inserator | Probe IPA | | |
| Siemens Schweiz AG | Intern | | |