Studie

|  |  |
| --- | --- |
| Status | In Prüfung |
| Projektname | Gradus |
| Projektleiter | Mirio Eggmann |
| Auftraggeber | Beat Walter |
| Autoren | Mirio Eggmann, Dario Menzel, Manuel Bieri, Nicolas Brechbühler |
| Verteiler | Entwicklerteam   * Mirio Eggmann * Dario Menzel * Manuel Bieri * Nicolas Brechbühler   Beat Walter |

**Änderungskontrolle, Prüfung, Genehmigung**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Datum | Beschreibung, Bemerkung | Name oder Rolle |
| 0.1 | 30.08.2016 | Kapitel Lösungsvarianten begonnen | Mirio Eggmann,  Dario Menzel,  Manuel Bieri,  Nicolas Brechbühler |
| 0.2 | 05.06.2016 | Variantenwahl fertiggestellt | Mirio Eggmann |
| 0.3 | 06.09.2016 | Projektplanungsdokument ausgefüllt | Mirio Eggmann,  Dario Menzel |
| 0.6 | 06.09.2016 | Planning Poker – Anforderungen mit Story Points ergänzt. | Mirio Eggmann,  Dario Menzel,  Manuel Bieri,  Nicolas Brechbühler |
| 0.8 | 12.09.2016 | Situationsanalyse, Ziele, Liste der Stakeholder und Lösungsbeschreibung ergänzt. | Mirio Eggmann |
| 0.9 | 13.09.2016 | Abkürzungen ergänzt. | Mirio Eggmann |
| 1.0 | 13.09.2016 | Prüfung des Dokumentes | Nicolas Brechbühler |
| 1.1 | 15.09.2016 | Letzte Prüfung | Mirio Eggmann |

**Definitionen und Abkürzungen**

|  |  |
| --- | --- |
| Begriff / Abkürzung | Bedeutung |
| Play Store | Google Marktplatz um Applikationen herunterzuladen. |
| SQL | «Structured Query Language»  Datenbanksprache |
| CSRF | «Cross-Site-Request-Forgery»  Webseitenübergreifende Anfragefälschung |
| XSS | «Cross-Site-Scripting»  Ausnutzen einer Computersicherheitslücke |
| PDF | «Portable Document Format»  Plattformunabhängiges Datei Format |
| REST | «Representational State Transfer»  Schnittstelle zwischen zwei Systemen |
| JPA | «Java Persistence API»  Datenbankzugriffe und objektrelationales Mapping |
| Single-page | Eine moderne Webanwendung die dynamisch Inhalt lädt. |
| Multi-page | Eine Webanwendung die bei jeder Anfrage den gesamten Inhalt neu lädt. |
| GIBB | Gewerblich-Industrielle Berufsschule Bern |
| App | Abkürzung für Applikation / Anwendung. |

**Referenzen**

|  |  |
| --- | --- |
| Referenz | Titel, Quelle |
| [1] | Spring Initializr, <https://start.spring.io> |
| [2] | Spring Boot, <http://projects.spring.io/spring-boot/> |
| [3] | Angular, <https://angular.io/> |
| [4] | Materialize, <http://materializecss.com/> |
| [5] | Hibernate, <http://hibernate.org/> |
| [6] | MySQL, <https://www.mysql.de/> |

**Inhaltsverzeichnis**

[1 Situationsanalyse 4](#_Toc461742899)

[1.1 Ausgangslage 4](#_Toc461742900)

[1.2 Stärken 4](#_Toc461742901)

[1.3 Schwächen 5](#_Toc461742902)

[2 Ziele 6](#_Toc461742903)

[2.1 Rahmenbedingungen 7](#_Toc461742904)

[2.2 Abgrenzung 7](#_Toc461742905)

[3 Liste der Stakeholder 7](#_Toc461742906)

[4 Anforderungen (Initial Product Backlog) 8](#_Toc461742907)

[5 Lösungsvarianten 8](#_Toc461742908)

[5.1 Variantenübersicht 8](#_Toc461742909)

[5.2 Beschreibung der Varianten 9](#_Toc461742910)

[6 Bewertung der Varianten (Tabelle) 10](#_Toc461742911)

[7 Lösungsbeschreibung 12](#_Toc461742912)

[8 Projektplanung 13](#_Toc461742913)

**Abbildungsverzeichnis**

[Figure 1 System-Skizze 12](#_Toc461742898)

**Tabellenverzeichnis**

[Tabelle 1 Stärken der bestehenden Lösung 4](#_Toc461742887)

[Tabelle 2 Schwächen der bestehenden Lösung 5](#_Toc461742888)

[Tabelle 3 Ziele 6](#_Toc461742889)

[Tabelle 4 Stakeholder 7](#_Toc461742890)

[Tabelle 5 Initial Product Backlog 8](#_Toc461742891)

[Tabelle 6 Variantenübersicht 8](#_Toc461742892)

[Tabelle 7 Variante a 10](#_Toc461742893)

[Tabelle 8 Variante b 10](#_Toc461742894)

[Tabelle 9 Variante c 11](#_Toc461742895)

[Tabelle 10 Variante d 11](#_Toc461742896)

[Tabelle 11 Abgedeckte Anforderungen mit der Lösungsbeschreibung 13](#_Toc461742897)

# Situationsanalyse

## Ausgangslage

In der App geht es darum, dass der Benutzer seine Schulnoten verwalten kann. Noten können eingetragen, bearbeitet und gelöscht werden. Darüber hinaus hat der Benutzer all seine Noten immer komfortabel im Überblick. Heute gibt es bereits Lösungen, die eine Verwaltung von Noten ermöglichen. Keine bestehende Lösung ist wirklich zufriedenstellend. Einige haben gute Benutzeroberflächen, sind aber von den Funktionen her nicht brauchbar. Andere haben viele Möglichkeiten, jedoch sind diese nicht benutzerfreundlich. Bei der Post CH AG gibt es ebenfalls eine bestehende Lösung, die aber Verbesserungspotential bietet. Die Problemstellung bei der neuen Lösung ist es, die Applikation den Wünschen und Bedürfnissen der Benutzerpalette so anzupassen, dass der Grossteil der User zufrieden ist. Sei es das User Interface oder der Ablauf innerhalb des Programms. Die Applikation soll stets verfügbar sein, auch unterwegs auf dem Smartphone. Das Programm soll alle nötigen Funktionen beinhalten, dabei aber die Benutzerfreundlichkeit beibehalten.

## Stärken

Tabelle 1 Stärken der bestehenden Lösung

| Nr. | Beschreibung |
| --- | --- |
| 01 | Die Schulnoten und Fächer sind in Semester unterteilt. |
| 02 | Noten von einem Fach können mit einer Gewichtung, Datum, Begründung und einem Typ gespeichert werden. |
| 03 | Die Noten können in einem Dokument exportiert werden. |
| 04 | Die Noten können nach Datum, Typ und Fach sortiert werden. |

## Schwächen

Tabelle 2 Schwächen der bestehenden Lösung

| Nr. | Beschreibung | Beurteilung | Ursache |
| --- | --- | --- | --- |
| S-01 | Unübersichtliche Darstellung der Noten bei vielen Einträgen | Optimierungspotential durch Unterteilung aller Noten in die verschiedenen Fächer | Fehlende Unterteilung in einzelne Fächer in der Anwendung |
| S-02 | Es können keine eigenen Fächer hinzugefügt werden | Optimierungspotential, wenn ein Fach nicht besteht und sonst jedes Mal ein Antrag an den Systemadministrator gemacht werden muss | Fehlende Funktionalität ein Fach hinzuzufügen in der Anwendung |
| S-03 | Die Anwendung kann nur im Intranet der Post CH AG über einen Desktop PC oder Laptop erreicht werden | Hohes Optimierungspotential, wenn die Möglichkeit besteht die Applikation extern und über das Handy zu erreichen. | Fehlende Möglichkeit die Applikation ausserhalb des Post Netzes zu verwenden |
| S-04 | Es ist nicht möglich das Prüfungsdokument hochzuladen | Es könnte durchaus nützlich sein, wenn man ein Bild oder eine PDF Datei des Prüfungsdokumentes bei der jeweiligen Note anhängen könnte. | Fehlende Möglichkeit eine Datei einer Note anzuhängen |
| S-05 | Das Design der Applikation ist veraltet und somit macht es nicht Lust die Applikation zu nutzen. | Für den Benutzer wäre es sicherlich angenehm eine Applikation mit ansprechendem und modernen Design zu nutzen. | Die Applikation ist nicht mehr die neuste und somit wäre eine Revision angebracht. |

# Ziele

Tabelle 3 Ziele

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nr. | Kategorie | Beschreibung | Messgrösse | Schwachpunkte | Priorität |
| Z-01 | Grundfunktion | Der Benutzer kann eigene Fächer erstellen. | Ein Formular zur Erstellung eines Faches steht zur Verfügung. | S-02 | Muss |
| Z-02 | Grundfunktion | Das Layout ist ansprechend und funktional zu gebrauchen. | Es wird ein Material Design verwendet und der Aufbau wird so einfach wie möglich gehalten. | S-05 | Muss |
| Z-03 | Grundfunktion | Der Benutzer kann in wenigen Klicks eine neue Note hinzufügen. | Eine Note kann mit max. 10 Klicks erstellt werden. |  | Muss |
| Z-04 | Grundfunktion | Der Benutzer hat die Möglichkeit nur die Noten eines bestimmten Faches anzeigen zu lassen. | Es gibt eine Ansicht, welche nur die Note des jeweiligen Faches anzeigt. | S-01 | Muss |
| Z-05 | Sicherheit | Es werden erforderliche Massnahmen getroffen um die Seite vor diversen Bedrohungen zu schützen. | Die Webapplikation wird vor SQL Injection, XSS und CSRF abgesichert. |  | Muss |
| Z-06 | Qualität | Es wird mit einer Testinstallation auf einem Webserver bewiesen das die Applikation funktioniert. | Die Applikation kann fehlerfrei installiert werden und anschliessend die Grundfunktionen durchgetestet werden. |  | Muss |
| Z-07 | Termin | Die Realisierung wird im zeitlichen Rahmen abgeschlossen. | Die Applikation ist nach 7 Wochen fertiggestellt. |  | Muss |

## Rahmenbedingungen

* Jeden Dienstagmorgen, bevor mit den Teilaufträgen am Projekt weitergefahren wird, wird in fünf bis zehn Minuten der erzielte Fortschritt und wie es weitergeht für die nächsten Tage besprochen. Ebenfalls wird auch angeschaut, ob jeder noch genug zu tun hat.
* Jeden Dienstag von 08:00 bis 11:30 Uhr arbeiten alle Teammitglieder gemeinsam in den von der GIBB zur Verfügung gestellten Räumlichkeiten am Projekt weiter.
* Am Schluss der drei Arbeitsstunden, welche jeden Dienstagmorgen zur Verfügung stehen, wird besprochen, welche Hausaufgaben jedes Teammitglied erledigt, wenn es welche fertigzustellen gibt. Auf diese Weise wird sich das Team in der Woche darauf nicht mehr mit alten, langwierigen Dingen beschäftigen, sondern könnte gleich mit den nächsten Aufgaben weiterfahren.
* Als Arbeitsmaschinen können die privaten Laptops oder die von der GIBB zur Verfügung gestellten Desktop Computer in Verwendung mit der Gibbix benutzt werden. Die Tools zur Programmierung dürfen selbst gewählt werden.

## Abgrenzung

Das Projekt wird zwar eine potentielle Ersatzlösung für das bestehende Post-Notenerfassungstool, jedoch werden wir das Produkt nicht direkt integrieren. Wir realisieren ein Programm, welches anschliessend von der zuständigen Abteilung untersucht werden kann und bei Zufriedenheit eingebaut wird. Dies ist aber nicht mehr in unserer Verantwortung. Wir werden das Projekt ebenfalls nicht Post-exklusiv machen und daher noch einen eigenen Webserver aufsetzen, der eine Instanz des Produktes zur Verfügung stellen wird.

# Liste der Stakeholder

Tabelle 4 Stakeholder

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Vorname, Name | Rolle / Funktion | Kontakt |
| STK-01 | Mirio Eggmann | Entwickler, Teilprojektleiter | mirio.eggmann@post.ch |
| STK-02 | Dario Menzel | Entwickler, Projektleiter | dario.menzel@post.ch |
| STK-03 | Manuel Bieri | Entwickler | manuel.bieri@post.ch |
| STK-04 | Nicolas Brechbühler | Entwickler | nicolas.brechbuehler@post.ch |
| STK-05 | Thomas Käser | Ansprechperson PostFinance | thomas.kaeser.1@postfinance.ch |
| STK-06 | Beat Walter | Auftraggeber | beat.walter@gibb.ch |

# Anforderungen (Initial Product Backlog)

Tabelle 5 Initial Product Backlog

| Initial Product Backlog | | |
| --- | --- | --- |
| **Anforderungen / Tätigkeiten** | **Betreffende Ziele** | **Aufwands-schätzung (Story Points)** |
| Ich als Benutzer kann einen neuen Account erstellen. |  | 20 |
| Ich als Benutzer kann mich mit meinem Account anmelden. |  | 5 |
| Ich als Benutzer kann Noten erfassen. | Z-03 / Z-04 | 8 |
| Ich als Benutzer kann ein Fach hinzufügen. | Z-01 | 5 |
| Ich als Benutzer kann meinem Account einen Beruf hinzufügen. |  | 13 |

# Lösungsvarianten

## Variantenübersicht

Tabelle Variantenübersicht

|  |  |
| --- | --- |
| Variante | Bezeichnung |
| Variante a | Statische multi-page Applikation |
| Variante b | Dynamische single-page Applikation |
| Variante c | Native Android / iOS mobile Applikation |
| Variante d | Native Windows Desktop Applikation |

## Beschreibung der Varianten

**Variante a**

Mit dieser Variante kann man eine Computer-freundliche Anwendung schaffen, die aber z.B. auf dem Smartphone nicht sehr praktisch ist. Ein Vorteil ist, dass das gesamte Entwicklerteam bereits Erfahrung mit solchen Applikationen hat. Die Möglichkeiten sind etwas eingeschränkt, weil es nicht sehr einfach ist von einer solchen Anwendung eine Smartphone-Applikation zu erstellen. Jedoch ist die Anwendung über das Internet erreichbar und somit grundsätzlich auf allen Geräten nutzbar.

**Variante b**

Diese Variante ermöglicht es das Programm auf Computern und Smartphones zu nutzen. Es handelt sich dabei um eine Webanwendung, welche auf modernen Technologien aufbaut. Das Design kann sehr ansprechend gestaltet werden und bei Bedarf ist es später möglich, eine Smartphone-Applikation daraus generieren zu lassen. Generell ist die Anwendung jedoch über das Internet erreichbar und somit von allen Geräten aus nutzbar.

**Variante c**

Bei dieser Variante wird eine Smartphone App entwickelt, welche man anschliessend aus dem App Store (Play Store) herunterladen könnte. Der Nachteil dieser Variante ist, dass die Anwendung anschliessend nur auf Smartphones verwendet werden kann und nicht auf dem Computer. Somit werden deutlich weniger Geräte unterstützt, als bei einer Webapplikation. Jedoch ist eine solche Applikation meistens performanter, als eine Internet-Applikation. Die Möglichkeiten sind jedoch etwas eingeschränkt, da man wie bereits erwähnt, nur eine Smartphone App erstellt.

**Variante d**

Diese Variante bietet dem Nutzer seine Noten per Computerprogramm zu bearbeiten. Der Vorteil dieser Möglichkeit ist, dass die Applikation unter Umständen viel flüssiger läuft, als eine Webanwendung. Jedoch ist hier das Problem, dass die Applikation nur auf einem Computer funktioniert und somit nicht auf dem Smartphone genutzt werden kann. Die Möglichkeiten sind aber begrenzt, da man eine Computer-App erstellt.

# Bewertung der Varianten (Tabelle)

**Variante a)**

Tabelle 7 Variante a

| Kriterium | Gewicht | Punkte | Total |  |
| --- | --- | --- | --- | --- |
| Abdeckung der Anforderungen | 5 | 8 | 40 |  |
| Realisierbarkeit, Risiken | 5 | 9 | 45 |  |
| Kosten | 3 | 9 | 27 |  |
| Sicherheit | 4 | 8 | 32 |  |
| Möglichkeiten | 4 | 7 | 28 |  |
| **Gesamtbeurteilung** |  |  | 172 |  |

Beurteilungen: 1-10, Gewichtung 1-5

**Variante b)**

Tabelle 8 Variante b

| Kriterium | Gewicht | Punkte | Total |  |
| --- | --- | --- | --- | --- |
| Abdeckung der Anforderungen | 5 | 10 | 50 |  |
| Realisierbarkeit, Risiken | 5 | 8 | 40 |  |
| Kosten | 3 | 9 | 27 |  |
| Sicherheit | 4 | 9 | 36 |  |
| Möglichkeiten | 4 | 9 | 36 | Es könnte leicht noch weiter in eine hybrid mobile Applikation umgewandelt werden. |
| **Gesamtbeurteilung** |  |  | 189 |  |

Beurteilungen: 1-10, Gewichtung 1-5

**Variante c)**

Tabelle 9 Variante c

| Kriterium | Gewicht | Punkte | Total |  |
| --- | --- | --- | --- | --- |
| Abdeckung der Anforderungen | 5 | 6 | 30 | Keine Möglichkeit die Applikation über den Computer zu öffnen. Auch für andere Smartphone Betriebssysteme nicht nutzbar. |
| Realisierbarkeit, Risiken | 5 | 8 | 40 |  |
| Kosten | 3 | 9 | 27 |  |
| Sicherheit | 4 | 9 | 36 |  |
| Möglichkeiten | 4 | 6 | 24 |  |
| **Gesamtbeurteilung** |  |  | 157 |  |

Beurteilungen: 1-10, Gewichtung 1-5

**Variante d)**

Tabelle 10 Variante d

| Kriterium | Gewicht | Punkte | Total |  |
| --- | --- | --- | --- | --- |
| Abdeckung der Anforderungen | 5 | 6 | 30 | Nur mit Microsoft Geräten nutzbar. |
| Realisierbarkeit, Risiken | 5 | 8 | 40 |  |
| Kosten | 3 | 9 | 27 |  |
| Sicherheit | 4 | 9 | 36 |  |
| Möglichkeiten | 4 | 6 | 24 |  |
| **Gesamtbeurteilung** |  |  | 157 |  |

Beurteilungen: 1-10, Gewichtung 1-5

# Lösungsbeschreibung

Es wurde die Lösungsvariante b gewählt. Dies aus dem Grund, weil es dadurch möglich ist die Applikation auf allen nötigen Geräten zur Verfügung zu stellen. Mit der gewählten Technologie ist die Applikation webbasiert. Man nennt diese Art von Applikation auch «Single-page Application». Sie kann aber auch leicht zu einer hybriden mobilen Applikation umgewandelt werden, welche die Nutzererfahrung noch viel angenehmer macht.

**Technologie**

*Angular [3] / Materialize* *[4]*

Dies wird für die Frontend Programmierung verwendet. Angular ist ein Java Script Framework von Google, welches Clientseitig ausgeführt wird. Dazu verwenden wir das Materialize CSS Framework, welches die Google Material Design Standards zur Verfügung stellt.

*Spring Data (REST / JPA)*

Spring Data stellt dem Frontend die Daten des Backendes zur Verfügung. Dies erfolgt über JSON.

*Spring Boot [2]*

Ermöglicht einen einfachen Einstieg in die Welt von Spring. Weiter wird Spring Boot heute als Standard angeschaut und die meisten neuen Java Applikationen werden damit aufgebaut. Durch den Spring Initializr [1] ist es möglich die Spring-Grundkonfiguration mit der Auswahl der gewünschten Pakete zu generieren. Unter anderem werden auch Pakete wie Spring Security in Einsatz kommen.

*Hibernate [5]*

Ermöglicht Objektrelationales Mapping und ermöglicht somit Programmierern eine objektorientierte Sicht auf Tabellen und Beziehungen in relationalen Datenbank Management Systemen.

*MySQL [6]*

Eine viel verwendete, relationale Datenbank.

**System-Skizze**

|  |
| --- |
| Angular / Materialize |
|
| Spring Data (REST / JPA) |
|
| Spring Boot |
|
| Hibernate |
|
| MySQL |
|

Figure 1 System-Skizze

Blau: Frontend, Grün: Backend, Gelb: Persistenz

**Abgedeckte Anforderungen**

Tabelle 11 Abgedeckte Anforderungen mit der Lösungsbeschreibung

| Nr. | Anforderungen | Variante b |
| --- | --- | --- |
| 1 | Ich als Benutzer kann einen neuen Account erstellen. | Ja |
| 2 | Ich als Benutzer kann mich mit meinem Account anmelden. | Ja |
| 3 | Ich als Benutzer kann Noten erfassen. | Ja |
| 4 | Ich als Benutzer kann ein Fach hinzufügen. | Ja |
| 5 | Ich als Benutzer kann meinem Account einen Beruf hinzufügen. | Ja |

# Projektplanung

Siehe separates Dokument Projektplan.