

Pflichtenheft

Einleitung

Jedes Semester müssen Sie Projekte verwalten, die von 2-3 Studenten bearbeitet werden sollen. Um die Themen der Projekte kümmern sich die Studenten selbst, d.h. sie schlagen Ihnen das Thema vor und Sie müssen entscheiden, ob das Thema den Ansprüchen genügt und zugelassen wird, ob noch Ergänzungen zu berücksichtigen sind oder das Thema abgelehnt werden muss.

Hierzu wird eine Software entwickelt, um Ihnen die Verwaltung der Projekte leichter zu gestalten.

Ausgangssituation

Bisher wurde diese Projektverwaltung durch Stift und Papier von den Mitarbeitern der FH selbst umgesetzt. Es gab keine zentrale Verwaltung.

Nun soll ein Softwareprodukt entwickelt werden, welches Ihnen -- den Projektverwalter -- bei der Verwaltung der studentischen Projekten hilft.

Pflichtziel

- Softwareprodukt, das
 - die Mitarbeiter der FH bei der Verwaltung von Projektanträgen innerhalb ihrer Lehrveranstaltung unterstützt
 - Studenten sollen Projektanträge digital einreichen können
 - Projektanträge müssen persistiert werden

Wunschziel

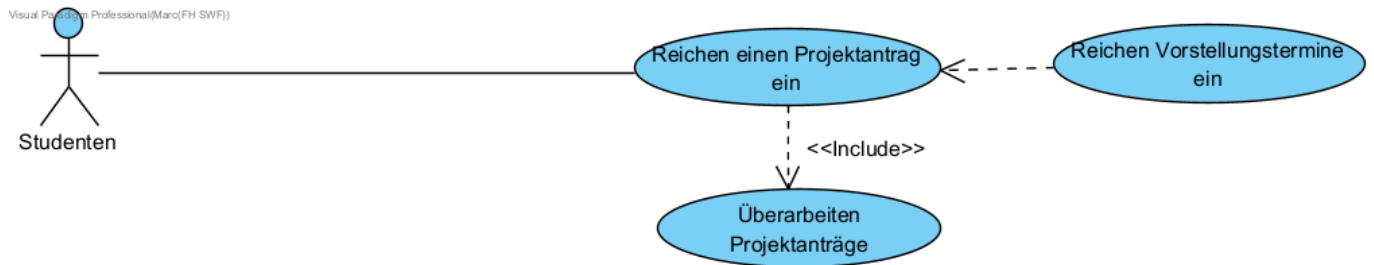
- Um keine extra Nutzerdatenbank zu warten => Login mit FH-Daten

Abgrenzung

- Noten für die fertigen Projekte und Vorstellungen sind nicht zu verwalten

Dekomposition

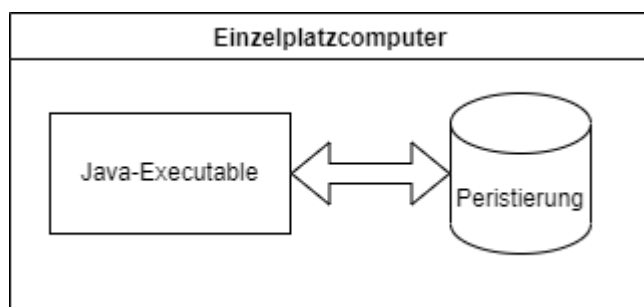
Use Case Diagramm



Kontext des Softwareproduktes

1. Die Anwendung soll für Mitarbeiter als auch Studenten leicht nutzbar sein, daher brauchen wir eine GUI
2. Die Daten müssen über längere Zeiträume gespeichert sein, daher brauchen wir eine Art der Persistierung, z.B. eine Datenbank oder Flatfile
3. Die Anwendung soll direkt auf dem Computer laufen; nicht im Webbrowser. Eine Java-Anwendung ist wünschenswert.

Architektur



Zusammenwirken mit anderen Produkten

1. Die einfache Verwaltung der Projektanträge macht es wesentlich leichter die Noten für die einzelnen Studenten am Semesterende in die Software der FH einzutragen.

Beschreibung der zu nutzenden Schnittstellen

1. Das Filesystem soll genutzt werden um Daten zu persistieren

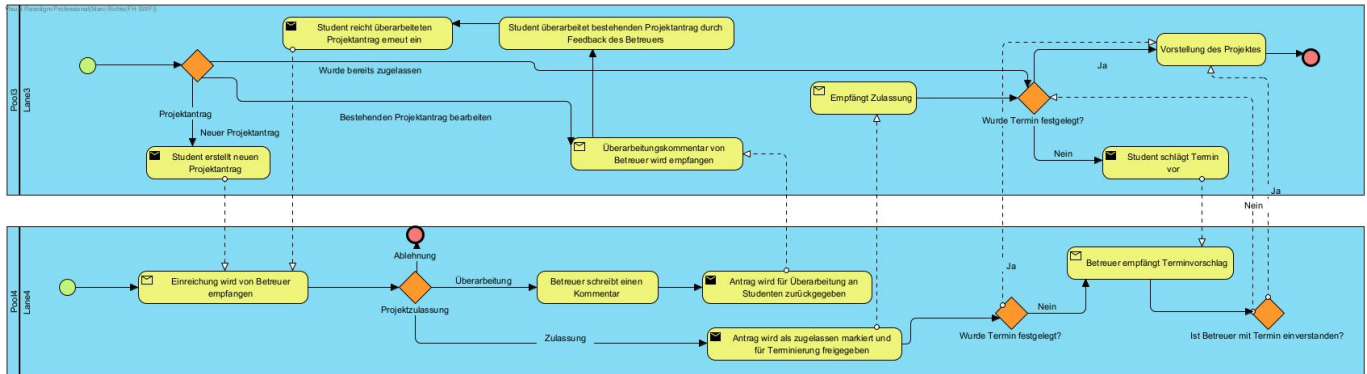
Präzisierung der Unterstützung in den Projektphasen

1. In der Projektphase der Implementation muss eine IDE verwendet werden

2. In der Produktion muss eine Java-Runtime vorhanden sein
3. In der Produktion muss eine Datenpersistierung vorhanden sein, z.B. über Flatfile, Sqlite, MySql etc.

Funktionale Anforderungen

Ablaufdiagramm



Folgende Funktionalitäten sind bereitzustellen:

Software:

1. Projektanträge von 2-3 Studenten pro Projektgruppe
2. Projektantrag enthält Kurzbeschreibung
3. Projektantrag enthält Titel
4. Projektantrag enthält Ausführliche Beschreibung (1-2 DIN A4 Seiten)
5. Projektantrag enthält Namen der Lehrveranstaltung
6. Projektantrag enthält Kontaktdaten für Ansprechpartner in Unternehmen
7. Projektantrag kann von Studenten eingereicht werden
8. Projektantrag kann von Studenten überarbeitet werden
9. Projektantrag wird für Studenten Read-Only, sobald er eingereicht wird.
10. Projektantrag kann von Mitarbeiter zugelassen werden
11. Projektantrag kann von Mitarbeiter abgelehnt werden
12. Projektantrag kann von Mitarbeiter zur Überarbeitung kommentiert und an den Studenten zurückgegeben werden (siehe Software Requirement #7). Hier wird der Projektantrag wieder als Schreibbar für den Studenten markiert.
13. Nach der Zulassung können Studenten zwei Termine zur Vorstellung vorschlagen
14. Vorschläge für Termin können von Mitarbeiter abgelehnt werden
15. Bei Ablehnung eines Termins muss der Student neue Termine vorschlagen können
16. Vorschläge für Termin können von Mitarbeiter angenommen werden
17. Der Zugang wird durch Username + Password geschützt

Hardware:

1. Individuelle PC-Nutzung notwendig
2. Realisierung auf deinem Desktop-PC
3. Zugriff aufs File-System für Persistierung

Organisatorische Anforderungen:

1. Einzelplatzanwendung
2. Programmierung als Java-Applikation
3. Ein Internetzugang ist nicht notwendig (lokale Anwendung)
4. Rechteverwaltung: Mitarbeiter können alle Anträge sehen; Studenten jedoch nur ihren eigenen Antrag
5. Studenten sind Nutzer
6. Mitarbeiter der FH sind Admins
7. Daten sollen persistiert werden
8. Eine graphische Oberfläche soll angeboten werden

Szenarien

Nr 1	Name: Studenten reichen einen Projektantrag ein
Kontext	Studenten können Projektanträge mit Kurzbeschreibung, detaillierter Beschreibung usw einreichen
Auslöser	Student möchte Projektantrag einreichen
Ebene	1
Hauptakteur	Anwender: Student (Endbenutzer)
Beteiligte und deren Ziel/Interessen	Beteiligter
Student	Möchte einen Projektantrag für das jeweilige Modul in seinem Studium einreichen
Vorbedingungen	Muss eingeschrieben sein
Mindestgarantie	Speicherung des Projektantrages in der Datenbank
Erfolgsgarantie	Projektantrag wird an den Mitarbeiter zur Bewertung geleitet
Erfolgsszenario	Schritt
1	Prüfung ob alle für den Antrag benötigte Daten eingetragen wurden
2	Daten werden gespeichert und an den Mitarbeiter zur Bewertung geleigtet
Misserfolgsszenario	Schritt
1	Projektantrag hat zusätzliche Informationen, die nicht in den vorgegebenen Feldern eingegeben werden können
Erweiterungen, Alternativen	Schritt
1a	Ein Feld für Speicherung von Anhängen

Nr 2	Name: Überarbeiten von Projektanträgen
Kontext	Studenten müssen Projektanträge bearbeiten, wenn diese von Mitarbeitern der FH bemängelt wurden
Auslöser	Mitarbeiter der Fachhochschule möchte eine Ausbesserung des Projektantrages
Ebene	2
Hauptakteur	Student
Beteiligte und deren Ziel/Interessen	Beteiligter
Student	Möchte den Projektantrag überarbeiten
Mitarbeiter	Kann die jeweiligen Projektanträge in Revision geben
Vorbedingungen	Ein Projektantrag muss bereits eingereicht worden sein, und vom Mitarbeiter als Verbesserungswürdig eingestuft worden sein.
Mindestgarantie	Speichern des aktuellen Projektantrages sowie neuer Einreichung des Antrages beim Mitarbeiter
Erfolgsgarantie	Der aktualisierte Projektantrag wird neu eingereicht
Erfolgsszenario	Schritt
1	Prüfung ob der aktualisierte Antrag korrekt eingegeben wurde
2	Aktualisierung des Antrages für den Mitarbeiter, der diese in einer Warteschlange nun erneut abarbeiten kann
Misserfolgsszenario	Schritt
1	Projektantrag hat zusätzliche Informationen, die nicht in den vorgegebenen Feldern eingegeben werden können
Erweiterungen, Alternativen	Schritt
1a	Ein Feld für Speicherung von Anhängen

Nr 3	Name: Vorstellungstermine einreichen
Kontext	Nachdem ein Projekt erfolgreich zugelassen wurde, kann ein Student Vorstellungstermine einreichen
Auslöser	Zugelassener Projektantrag
Ebene	2
Hauptakteur	Student

Nr 3	Name: Vorstellungstermine einreichen
Beteiligte und deren Ziel/Interessen	Beteiligter
Student	Möchte Vorstellungstermine einreichen
Vorbedingungen	Der Projektantrag muss vom Mitarbeiter zugelassen worden sein
Mindestgarantie	Speicherung von Vorstellungsterminen
Erfolgsgarantie	Vorstellungstermine wurden erfolgreich gespeichert und werden dem Mitarbeiter gezeigt
Erfolgsszenario	Schritt
1	Prüfung ob Daten in der Zukunft liegen
Misserfolgsszenario	Schritt
1a	Invalide Daten
Erweiterungen, Alternativen	-

Nr 4	Name: Betreuer verwaltet Projektanträge
Kontext	Ein Betreuer verwaltet Projektanträge; dies bedeutet er kann sie ablehnen, zulassen, oder zur Überarbeitung zurückgeben.
Auslöser	Projektantrag wird von Studenten submitted
Ebene	1
Hauptakteur	Betreuer
Beteiligte und deren Ziel/Interessen	Beteiligter
Betreuer	Möchte dass das Projekt zugelassen, abgelehnt oder zur Überarbeitung zurückgegeben wird. Hier wird ein Kommentar mit den auszuführenden Ausbesserungen hinterlassen
Student	Falls das Projekt zur Überarbeitung zurückgegeben wird muss ein Student es überarbeiten
Vorbedingungen	Projektantrag muss eingegangen sein
Mindestgarantie	Projekt wird zugelassen/abgelehnt/zur Überarbeitung+Kommentar zurückgegeben
Erfolgsgarantie	Projektantrag wird bewertet

Nr 4 Name: Betreuer verwaltet Projektanträge	
Erfolgsszenario	Schritt
1	Betreuer gibt einen Kommentar mit allen auszuführenden Verbesserungen ab
2	Aktualisierung des Projektantrages
Misserfolgsszenario	Schritt
1a	Kommentar fehlt
Erweiterungen, Alternativen	Kommentar muss unbedingt hinterlassen werden
Nr 5 Name: Vorstellungsterminen zustimmen	
Kontext	Nach der Zulassung von Projekten können Studenten Vorstellungstermine terminieren. Hier kann der Betreuer Terminen jeweils zustimmen
Auslöser	Student reicht Vorstellungstermine ein
Ebene	2
Hauptakteur	Betreuer
Beteiligte und deren Ziel/Interessen	Beteiligter
Betreuer	Möchte einen Termin zur Benotung des Projektes finden
Student	Möchte das Projekt vorstellen
Vorbedingungen	Vorstellungsdaten von Studenten müssen vorliegen
Mindestgarantie	Es wird ein Termin zu/abgesagt
Erfolgsgarantie	Es wird ein Termin zu/abgesagt
Erfolgsszenario	Schritt
1	Y/N zu Terminen
Misserfolgsszenario	-
Erweiterungen, Alternativen	-

Skizze des Lebenszyklus und Gesamtarchitektur

Im Rahmen dieser Projektabgabe verzichten wir auf Prototypen und verzichten auf das Iterieren mit Nutzerfeedback.

Meilensteine

1. Grobanforderungen ☒
2. Detailstudie ☒
3. Implementierung
4. Abnahme

Sicherheitsanforderungen

Das Softwareprodukt enthält persönliche Daten und Informationen zu den Projektanträgen anderer Nutzer, daher müssen wir den Zugriff durch die Nutzerparteien (Studenten und Mitarbeiter) verschieden regulieren. Studenten dürfen nur ihren eigenen Antrag sehen. Der Zugang wird durch Nutzernamen + Passwort geschützt.

Lieferumfang

Das Softwareprodukt wird in Form einer einzelnen Executable (ausführbare Datei) ausgeliefert. Es gibt keinerlei Lizenzen; das Produkt ist auf jedem Computer frei ausführbar.

Abnahmekriterien

Folgende Tests müssen alle bestanden werden:

1. Überprüfen ob die Daten bei Programmstart geladen werden und bei Aktionen (Beantragung/Administrierung) gespeichert werden.
2. Studenten können nur auf eigene Projektanträge zugreifen
3. Studenten können Projektanträge einreichen
4. Mitarbeiter können Projektanträge ansehen und administrieren (zulassen, ablehnen, zur Überarbeitung zurückgeben)
5. Alle Aktionen müssen über eine graphische Oberfläche verfügbare sein.

Abkürzungsverzeichnis

- FH => Fachhochschule
- Persistierung => den Zustand der Daten speichern

Literaturverzeichnis

- Es wurden keine Drittparteidokumente verwendet

Abbildungsverzeichnis

[^1]: Use Case Diagram Blatt5.png - Ein Diagramm, welches alle Use-Cases der Software aufzeigt.

[^2]: Arch.png - Zeigt die Architektur unserer Software mitsamt aller Systeme und Abhängigkeiten

[^3]: BNPM Diagram.jpg - Zeigt den Ablauf eines Workflows