Pflichtenheft

Projektverwaltung

Software-Engeering WiSe 22/23

Oliver Sabelfeld

Inhaltsverzeichnis

[1 Einleitung 2](#_Toc119486240)

[2 Ausgangssituation und Zielsetzung 2](#_Toc119486241)

[2.1 Stakeholder 2](#_Toc119486242)

[2.2 Ziele 2](#_Toc119486243)

[3 Dekomposition des Systems 2](#_Toc119486244)

[4 Funktionale Anforderungen 3](#_Toc119486245)

[5 Nicht-funktionale Anforderungen 5](#_Toc119486246)

[6 Sicherheitsanforderungen 5](#_Toc119486247)

[7 Lieferumfang 5](#_Toc119486248)

[8 Abnahmekriterien und Vorgehen zur Ausgangsprüfung 6](#_Toc119486249)

[9 Abkürzungsverzeichnis 6](#_Toc119486250)

[10 Literaturverzeichnis 6](#_Toc119486251)

[11 Abbildungsverzeichnis 8](#_Toc119486252)

# Einleitung

Innerhalb der Hochschule soll die Projektvergabe mithilfe einer Software vereinfacht werden. Hierbei soll es möglich sein, Projektvorschläge seitens der Studenten einzureichen. Außerdem soll der Verwalter die Anträge einfach verwalten können (akzeptieren, ablehnen, Ergänzung)

# Ausgangssituation und Zielsetzung

Die bisherige Projektverwaltung erfolgte hauptsächlich mündlich und schriftlich. Dieser Prozess des Themenvorschlags, Fordern von Ergänzungen und Annehmen des Vorschlags erweist sich als unverhältnismäßig aufwändig für die heutige Zeit.  
Die zu entwerfende Software soll die bisherige Vergabe der Projekte effizienter gestalten, um auch den Zeitaufwand der Dozenten zu verringern und somit auch die Kosten zu reduzieren. Außerdem soll es Studenten zu einem angenehmeren Studium verhelfen.

## Stakeholder

Die Anwendung umfasst zwei Akteure. Dies umfasst Student und Verwalter.

## Ziele

Pflichtziel:  
Die Pflichtziele bestehen aus:

* Das Softwareprodukt
  + unterstützt bei Prozess der Projekteinreichung
  + wird von zwei Arten von Benutzern verwendet, Student und Verwalter
  + ermöglicht den Studenten Anträge zu stellen zu Ergänzungen
  + ermöglicht den Studenten eine Terminansicht
  + Soll dem Verwalter ermöglichen Termine verwalten zu können
  + Soll dem Verwalter ermöglichen Projekte zu verwalten

## Wunschziele

Es gibt keine genannten Wunschziele.

## Abgrenzung

Die Kommunikation soll nicht per Mail erfolgen.

# Dekomposition des Systems

Die Software wird aufgeteilt in Frontend (Anwendung selbst) und Backend, was für die Datenpersistierung verantwortlich ist. Hierbei wird als Fremdsoftware Firebird verwendet.

Realisierung der Anwendung auf

* H01 Desktop-PC mit installiertem Windows.
* H02 Internetverbindung notwendig

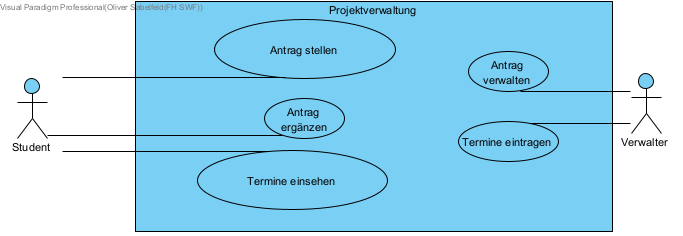


Abbildung Kontextdiagramm

# Funktionale Anforderungen

## Software

Verwalter

|  |  |
| --- | --- |
| V01 | Es soll eine Softwareapplikation entwickelt werden, die diese Situation deutlich vereinfacht. |
| V02 | Der Verwalter soll zustande sein, Projekte zu verwalten. Das umfasst: Annehmen, Ablehnen und Fordern von Ergänzungen der Projekte |
| V03 | Organisieren von Terminen zu akzeptierten Projekten zwecks Vorträge. Diese sollen dem Studenten bzw. der ganzen Gruppe angezeigt werden, falls das Projekt akzeptiert wurde |

Student

|  |  |
| --- | --- |
| S01 | Es soll eine Möglichkeit zur Termineinsicht geben |
| S02 | Der Student soll für seine Gruppe (2-3 Studenten) Projektvorschläge vorlegen können. |
| S03 | Der Vorschlag soll enthalten: Titel, kurze und lange (1-2 Seiten) Beschreibung, Skizze (3-4 Sätze), Ansprechpartner |
| S04 | Der Student kann an mehreren Projekten teilnehmen |
| S05 | Der Student kann sich einer Gruppe zuordnen |

## UseCases

Verwalter

|  |  |
| --- | --- |
| UV01 | Verwalten der Anträge |
| UV02 | Fordern von Ergänzungen |
| UV03 | Annehmen des Antrags |
| UV04 | Ablehnen des Antrags |
| UV05 | Organisieren von Terminen zwecks Vortrags |

Student

|  |  |
| --- | --- |
| US01 | Stellen des Antrags |
| US02 | Bearbeiten des noch nicht bewertenden Antrags |
| US03 | Ergänzung des Antrags bei entsprechender Forderung |
| US04 | Einsehen der Termine |

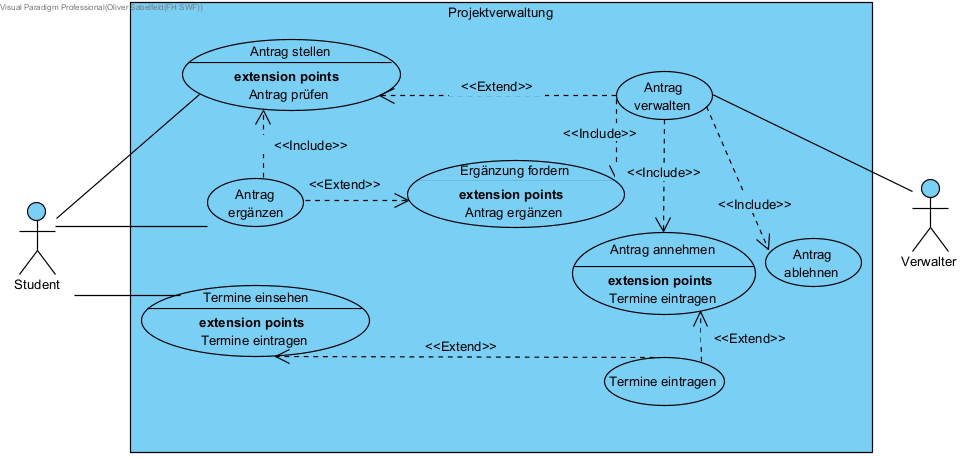


Abbildung Präzisierung des Kontextdiagramms

## Ein Bild, das Text, Himmel, Screenshot enthält. Automatisch generierte Beschreibung

Abbildung Business Process Diagramm

## komplexe Anwendungsfälle einzeln in Tabellenform

|  |  |
| --- | --- |
| Nr.: 1 | Name: Antrag einstellen (vom  Studenten) |
| Kontext | Projektverwaltung |
| Auslöser | Student reicht Antrag ein |
| Umfang (Scope) | Eingaben aller erforderlichen Textfelder, setzen des Status Antrags auf submitted |
| Ebene (Level) | 1 |
| Hauptakteur (Primary Actor) | Anwender: Studentengruppe/Student |
| Beteiligte und deren Ziele/Interessen | |  |  | | --- | --- | | Beteiligter | Ziel | | Student/  Gruppe | Der Student kann für seine Gruppe die für der Fach notwendigen Antrag für ein Projekt ansehen. | | Verwalter | Der Verwalter muss die Anträge nicht mehr aufwändig organisieren. | |
| Vorbedingungen | Es muss ein Erstantrag sein |
| Mindestgarantie | - |
| Erfolgsgarantie | Antrag wird erfolgreich im System eingetragen. |
| Erfolgsszenario | |  |  | | --- | --- | | Schritt | Aktion | |  | Eintragen aller erforderlichen Daten | |  | Absenden des Formulars | |
| Misserfolgsszenario | |  |  | | --- | --- | | Schritt | Aktion | |  | Es nicht alle Daten vorhanden | |  | Das Formular kann ohne alle Pflichtfelder nicht abgesendet werden | |
| Erweiterungen, Alternativen | |  |  | | --- | --- | | Schritt | Aktion | |  | Lückenhaften Antrag einreichen | |  |  | |

|  |  |
| --- | --- |
| Nr.: 2 | Name: Antrag kommentieren (vom Dozenten) |
| Kontext | Projektverwaltung |
| Auslöser | Antrag wurde von der Gruppe eingereicht, und der Dozent wählt es aus und klickt auf „Ergänzung fordern“ |
| Umfang (Scope) | Auswahl von Checkboxen, Eingabe der Nachricht, setzen des Status |
| Ebene (Level) | 1 |
| Hauptakteur (Primary Actor) | Anwender: Verwalter |
| Beteiligte und deren Ziele/Interessen | |  |  | | --- | --- | | Beteiligter | Ziel | | Student/  Gruppe | Der Student/Gruppe kann erfahren, was in dem Antrag fehlt und was entsprechend nachgereicht werden soll. | | Verwalter | Der Verwalter muss z.B. keine Mails bezüglich nachfragen beantworten. | |
| Vorbedingungen | Der Antrag existiert, ist nicht zugelassen worden |
| Mindestgarantie | - |
| Erfolgsgarantie | Erfolgreich kommentiert und Status als editRequested gesetzt |
| Erfolgsszenario | |  |  | | --- | --- | | Schritt | Aktion | |  | Prüfung, ob alles der Antrag den Wünschen entspricht | |  | Verfassen des Kommentars oder Checkboxen ausgewählt | |  | Absenden | |
| Misserfolgsszenario | |  |  | | --- | --- | | Schritt | Aktion | |  | Prüfung, ob alles der Antrag den Wünschen entspricht | |  | Das Kommentarfeld ist leer und die Checkboxen | |  | Der Kommentar kann nicht abgesendet werden | |
| Erweiterungen, Alternativen | |  |  | | --- | --- | | Schritt | Aktion | |  | Ergänzung ohne Kommentar fordern | |  |  | |

|  |  |
| --- | --- |
| Nr.: 3 | Name: Eintragen von Terminen (vom Dozenten) |
| Kontext | Projektverwaltung |
| Auslöser | Antrag wurde von der Gruppe eingereicht, und der Dozent hat es angenommen und klickt auf Termin eintragen |
| Umfang (Scope) | Eintragen von zwei Termin und klicken auf Speichern |
| Ebene (Level) | 1 |
| Hauptakteur (Primary Actor) | Anwender: Verwalter |
| Beteiligte und deren Ziele/Interessen | |  |  | | --- | --- | | Beteiligter | Ziel | | Student/  Gruppe | Der Student/Gruppe sieht, wann er seine Vorträge halten | | Verwalter | Der Verwalter muss die Termine nicht selbst aktiv mitteilen | |
| Vorbedingungen | Der Antrag existiert, ist bereits zugelassen worden |
| Mindestgarantie | - |
| Erfolgsgarantie | Erfolgreich gesetzte Termine |
| Erfolgsszenario | |  |  | | --- | --- | | Schritt | Aktion | |  | Einreichen des Antrags vom Studenten | |  | Annehmen des Projekts vom Verwalter | |  | Eintragen der Termine | |  | Clicken auf Speichern | |
| Misserfolgsszenario | |  |  | | --- | --- | | Schritt | Aktion | |  | Einreichen des Antrags vom Studenten | |  | Annehmen des Projekts vom Verwalter | |  | Kein Eintragen der Termine in Textfelder | |  | Kann nicht gespeichert werden, da Terminfelder leer | |
| Erweiterungen, Alternativen | |  |  | | --- | --- | | Schritt | Aktion | |  | - | |  |  | |

## Aussagen zur GUI

Die Anwendung soll über eine Grafische Nutzeroberfläche bedienbar sein. Hierbei habe die beiden Profile verschiedene Ansichten.  
Es soll seitens der Studenten folgende Fenster geben:

1. ~~Login~~
2. Startseite mit allen eingeschrieben Kursen
3. Anzeige von Kursen, denen man beitreten kann
4. Beantragen / Bearbeiten von Anträgen
5. Ansicht von Terminen

Seitens des Verwalters gibt es folgende Fenster:

1. ~~Login~~
2. Startseite mit allen eingeschrieben Kursen
3. Detailseite eines Kurses: Anzeige von allen Anträgen einer Lehrveranstaltungen
4. Detailseite eines Eintrags mit Verwaltungsoptionen (Annehmen/Ablehnen, ~~Termine~~)

# Nicht-funktionale Anforderungen

|  |  |
| --- | --- |
| T01 | Die Problemstellung soll in einer Form einer Softwareapplikation gelöst werden |
| T02 | Es soll keine Kommunikation per Mail erfolgen |
| T03 | Die Daten sollen entsprechend in einer Datenbank gespeichert werden |
| T04 | Die Differenzierung der Benutzergruppen erfolgt über ein Rollensystem bzw. verschiedene Profile |

# Sicherheitsanforderungen

Um unbefugten Zugriff abzuwehren, wird die Anwendung Login haben. Jeder Student hat lediglich Zugriff seine eigenen Daten bzw. die der Gruppe. Der Verwalter hat Zugriff auf seine eigenen Lehrveranstaltungen und kann sie entsprechend verwalten.  
Die Daten selbst werden im Klartext in einer Datenbank gespeichert. Der Zugriff darauf ist entsprechend geschützt. Das Passwort wird selbstverständlich mit SHA-256 gehasht gespeichert. Die Datenbank selbst wird von einem Administrator verwaltet.

# Lieferumfang

Das Produkt wird in einer ausführbaren .exe Datei ausgeliefert. Es ist keine Installation notwendig.

# Abnahmekriterien und Vorgehen zur Ausgangsprüfung

## Testfälle

Testfall 1: Erfolgsfall

* Student reicht Eintrag ein
* Antrag erscheint beim Verwalter
* Verwalter akzeptiert und trägt Termine ein
* Student kann Termine einsehen

Testfall 2: Antrag stellen

* Student startet Eintrag
* Student füllt nicht alle Pflichtfelder aus
* Student drückt auf „Absenden“
* Es erscheint eine Fehlermeldung

Testfall 3: Kommentieren von Anträgen ohne Kommentar

* Student reicht Eintrag ein
* Verwalter kommentiert, setzt allerdings keinen Kommentar
* Fehlermeldung erscheint

Testfall 4: Absenden von unvollständigen Anträgen

* Student füllt nicht alle Pflichtfelder aus
* Student klickt auf Absenden
* Fehlermeldung erscheint

# Abkürzungsverzeichnis

Projektbeschreibung: Enthält Titel, kurze und lange (1-2 Seiten) Beschreibung, Skizze (3-4 Sätze), Ansprechpartner

Ansprechpartner: kann bei der Hochschule oder einem Unternehmen arbeiten

# Literaturverzeichnis

# Abbildungsverzeichnis

[Abbildung 1 Kontextdiagramm 3](#_Toc119579558)

[Abbildung 2Präzisierung des Kontextdiagramms 4](#_Toc119579559)

[Abbildung 3 Business Process Diagramm 4](#_Toc119579560)