12 MATSE-SE Projekt: Kanban-Tafel

|  |  |
| --- | --- |
|  | **Ziel**: Sie entwickeln in Projektgruppen eine digitale Variante einer Kanban-Tafel (Instrument des Projektmanagements). |

**„In einem erfolgreichen Projekt müssen nicht nur Alle am gleichen Strang ziehen, sondern auch in die gleiche Richtung.“**

# Zielstellung des Projekts

In diesem Projekt werden Sie über vier Wochen, in den Lernfeldern:

* 11 „Parallele Prozesse gestalten und in Netzwerken programmieren“ (frm)
* 12 „Vorgehensmodelle des Software-Engineering auswählen und projektorientiert anwenden“ (brü)
* 13 „Komplexe Softwaresysteme im Projekt konzipieren und realisieren“ (brü),

eine netzwerkbasierte digitale Mehrbenutzer-[Kanban-Tafel](https://de.wikipedia.org/wiki/Kanban-Tafel)) entwerfen, implementieren, dokumentieren und präsentieren. Neben ihren programmiertechnischen Fähigkeiten, werden Sie ihre Kenntnisse im Bereich des Projektmanagements vertiefen.

# Projektauftrag

Entwickeln Sie ein Softwaresystem in Java, welches eine netzwerkbasierte digitale Mehrbenutzer-Kanban-Tafel (auch Kanban-Board genannt) realisiert.

* Dokumentieren Sie die Arbeitsrollen in ihrer Gruppe.
* Erstellen Sie einen realistischen Zeitplan für das Projekt. Setzen Sie sich ein realistisches Gesamtziel und definieren Sie Milestones
* Erstellen Sie UML Anwendungsfalldiagramme, um wesentliche Funktionalitäten abzubilden.
* Erstellen Sie UML Klassendiagramme, welche die notwendigen Klassen und deren Beziehungen zueinander beschreiben.
* Erstellen Sie UML Sequenzdiagramme, welche die Prozesskommunikationen abbilden.
* Implementieren Sie eine netzwerkbasierte digitale Mehrbenutzer-Kanban-Tafel, nach dem [Client-Server-Prinzip](http://userpage.fu-berlin.de/~schmiete/vorlesung/ws2008/cs_part1.pdf)
* Nutzen Sie zur Übersicht der Projektschritte/ Ziele ebenfalls ein (analoges) Kanban-Board. Die Kategorien/Spaltenbezeichnungen legen sie selbstständig fest.

# Vorbereitende Aufgaben

Bearbeiten Sie die Dokumentation (Projektziele, Projektrollen, Milestones) **nur einmal je Projektgruppe**!

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Projekt-Vorbereitungen** | **Erledigt** |
| 1 | Erhalt der Projektunterlagen |  |
| 2 | Verschaffen Sie sich einen Überflick über die Ausbaustufen mit den jeweiligen Anforderungen. |  |
| 3 | Lesen Sie die Beschreibungen der einzelnen Ausbaustufen. Setzen Sie sich gemeinsam ein vorläufiges Ziel, welches für Sie realistisch erscheint. |  |
| 4 | Teilen Sie die unterschiedlichen [Arbeitsrollen ein (siehe Abschnitt Projektrollen)](#_Projektrollen_2). |  |
| 5 | Füllen Sie für eine Zeitplanung das Raster „[Milestones](#_Milestones_1)“ aus. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Technische Vorbereitungen** | **Erledigt** | **Geprüft** |
| 1 | Wählen Sie **eine** Entwicklungsumgebung (IDE) z.B. Eclipse, Net Beans, IntelliJ. Achten Sie darauf, dass die IDE **Java FX** unterstützt!  **Tipp**: Diese Wahl sollte gemeinschaftlich in der Gruppe getroffen werden. |  |  |
| 2 | Richten Sie sich einen Git-Hub ein und verwenden Sie diesen zum [Austausch](https://rogerdudler.github.io/git-guide/index.de.html) des Quellcodes. |  |  |
| 3 | Einigen Sie sich auf ein Werkzeug zur Erstellung von UML Klassen-, Anwendungsfall- und Sequenzdiagrammen. |  |  |

Bearbeiten Sie die folgenden Aufgaben, dokumentieren Sie ihre Entscheidungen im Kapitel [Zieldefinition](#_Zieldefinition).

# Ausbaustufen

## Funktionale Anforderungen

**Die folgende Beschreibung der Anforderungen ist nicht in allen Punkten präzise und umfasst nicht alle Details, dies ist gewollt! Treffen Sie auf Basis ihres Wissens die richtigen Entscheidungen.**

### Ausbaustufe: Basisanforderungen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Anforderung** | **Bereich[[1]](#footnote-1)** | **Erledigt** | **Geprüft** |
|  | Zählen Sie Funktionen auf, die eine digitale Kanban-Tafel erfüllen muss. Nutzen Sie dazu die Methode des Mindmappings. | 13 |  |  |
|  | Überführen Sie die Funktionalitäten aus Schritt 1 in ein dafür geeignetes UML-Diagramm. | 13 |  |  |
|  | Konzipieren Sie die grafische Oberfläche einer Kanban-Tafel, mit Hilfe einer Skizze und/oder einem Zeichenprogramm. **(Client)** | 12 |  |  |
|  | Erstellen Sie eine grafische Nutzeroberfläche für ein Kanban-Board mit Hilfe von JavaFX (Scenebuilder). **(Client)** | 11 |  |  |
|  | Die Speicherung der Kategorien und Post-ist erfolgt in einzelnen XML-Dokumenten. Legen Sie ein entsprechendes Schema für diese Dokumente fest. **(Server)** | 12 |  |  |

**Hinweis**: Alle Schritte dieser Ausbaustufe sind in den Tutorials zu [JavaFX 8](http://code.makery.ch/library/javafx-8-tutorial/) und [Java/XML](http://openbook.rheinwerk-verlag.de/javainsel/javainsel_16_001.html) verfügbar!

### Ausbaustufe: Funktionalitäten erweitern

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Anforderung** | **Bereich** | **Erledigt** | **Geprüft** |
|  | Ermöglichen Sie es Post-its auf dem Kanban-Board hinzuzufügen. | 11 |  |  |
|  | Mehrere Nutzer sollen auf dem Kanban-Board Post-its erstellen können und innerhalb der Spalten verschieben | 11 |  |  |
|  | Die Post-its sollen automatisch mit einem Namen versehen werden, wenn ein Nutzer diesen in eine neue Spalte überführt. | 11 |  |  |

### Ausbaustufe: Anbindung an LDAP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Anforderung** | **Bereich** | **Erledigt** | **Geprüft** |
|  | Anbindung an den LDAP Server der Schule() . Erkennung von Lehrer und Schülerprofilen anhand der „uid“, Zuweisung der Klassen (im LDAP Profil gespeichert) | 11 |  |  |
|  | Login über HEMS-LDAP Server | 11 |  |  |

Daten HEMS LDAP:

* ldap://130.83.152.28:389
* uid={0},ou=accounts,dc=linuxmuster-net,dc=lokal

### Ausbaustufe: Flexibler Aufbau des Kanban-Board

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Anforderung** | **Bereich** | **Erledigt** | **Geprüft** |
|  | Erstellung weiterer Kategorien (Spalten) für das Kanban-Board. | 11 |  |  |
|  | Ermöglichen Sie es Spaltenbezeichnungen umzubenennen. | 11 |  |  |

### Ausbaustufe: Eigene Funktionen

Definieren Sie weitere sinnvolle Funktionalitäten, dokumentieren Sie diese und implementieren sie diese.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Anforderung** | **Bereich** | **Erledigt** | **Geprüft** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Nicht funktionale Anforderungen

### Ausbaustufe Dokumentation (Basisanforderungen)

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Anforderung** | **Erledigt** | **Geprüft** |
|  | Verwenden Sie zur Koordinierung der Aufgaben innerhalb ihrer Gruppe ein (analoges) Kanban-Board. Dokumentieren Sie (per Foto) den Fortschritt/Veränderungen auf dem Board |  |  |
|  | Schriftliche Dokumentation der **Arbeitsrollen**. |  |  |
|  | Dokumentation der **Milestones.** |  |  |
|  | Dokumentation der **Änderungen** von **Milestones** (bei Bedarf). |  |  |
|  | Einigen Sie sich darauf welche Programme Sie zur **Programmierung** und zur **Dokumentation** verwenden wollen (astah community, yEd, Visio). Alle Gruppenmitglieder sollten das Programm auf dem PC haben! |  |  |
|  | Darstellung von mindestens vier Klassen der Model-Ebene als UML Klassendiagramm. |  |  |
|  | Darstellung der Struktur der XML-.Dokumente. |  |  |
|  | Dokumentation der grafischen Oberfläche (Entwurf/Realisierung) |  |  |
|  | Nennung von Problemen im Projekt und wie diese gelöst wurden. |  |  |

### Anforderungsstufe – Dokumentation (Vertiefung)

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Anforderung** | **Erledigt** | **Geprüft** |
|  | Stellen Sie die Model Klassen als UML-**Klassendiagramm** dar. |  |  |
|  | Stellen Sie den Ablauf des Login, sowie zwei weiterer elementarer Funktionalitäten/Interaktionen als UML-**Sequenzdiagramm** dar. |  |  |

### Präsentation der Projektergebnisse

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Anforderung** | **Erledigt** | **Geprüft** |
|  | Es wird deutlich dargelegt welche **Ausbaustufe** und welche einzelnen Features realisiert wurden. |  |  |
|  | **Probleme** in der Entwicklung (fachliche als auch gruppeninterne) werden genannt und die angewendete Lösungsstrategie kurz erläutert. |  |  |
|  | **Aufbau** und **Strukturierung** (Sachliche Gliederung, Inhalte werden logisch und nachvollziehbar dargelegt, zielorientierte Darlegung der Ergebnisse) |  |  |
|  | Sprachliche Gestaltung (Ausdrucksweise, Satzbau, Stil) |  |  |
|  | Zielgruppengerechte Darstellung (Medieneinsatz, Visualisierung, Körpersprache. |  |  |

Die Anforderungen an die Projektpräsentation basiert auf den Kriterien zur [Bewertung von Abschlussprojekten im IT-Bereich der IHK Darmstadt](https://www.darmstadt.ihk.de/produktmarken/aus_und_weiterbildung_channel/pruefungen/downloads/Muther/IT_Hinweise_Projektarbeit/2551050).

# Tipps zur Vorgehensweise in Projekten

1. Verschaffen Sie sich einen Überblick über die **funktionalen** und **nicht** **funktionalen** Anforderungsbereiche.
2. Achten Sie auf eine sinnvolle **Modellierung** der Klassen sowie der verwendeten Datenstrukturen.
3. Achten Sie darauf die **Dokumentation** kontinuierlich zu pflegen, nicht erst am Schluss!
4. Gehen Sie **arbeitsteilig** vor. Das Projekt beinhaltet viele Teilaufgaben, die sinnvollerweise unabhängig voneinander gelöst werden. Es ist nicht notwendig, dass alle Gruppenmitglieder an den gleichen Teilaufgaben arbeiten!
5. Setzen Sie sich **realistische Ziele** (z.B. Realisierung der Ausbaustufe III), hierbei kann die [SMART-Formel](https://de.wikipedia.org/wiki/SMART_(Projektmanagement)) helfen.
6. Unterteilen Sie das Ziel in realistische **Milestones** (Teilziele).
7. Haben Sie keine Angst das **Gesamtziel** oder **Milestones** zu **ändern**!
8. Fokussieren Sie sich zunächst auf eine **funktionsfähige Basisversion**. Die weiteren Ausbaustufen sind teilweise erst dann (sinnvoll) umsetzbar, wenn das Gesamtprogramm zumindest rudimentär funktioniert.
9. Nutzen Sie den umfangreichen **Zeitrahmen** bereits **am Anfang** aus!
10. Nutzen Sie **Hilfsangebote**! Fragen sie die Lehrkraft oder tauschen sie sich mit anderen Projektgruppen aus! Die Projektgruppen stehen nicht im Wettbewerb zueinander.

# Stammblatt-Projektarbeit

## Gruppenmitglieder

Tragen Sie die Gruppenmitglieder ein (maximal 4 Personen pro Gruppe!). Ergänzen Sie später die Zuweisung der Projektrollen, siehe folgenden [Abschnitt Projektrolle](#_Projektrollen_3)n.

|  |  |  |
| --- | --- | --- |
|  | Gruppenmitglied | Projektrolle(n) |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |

## Projektrollen

In einem Projekt fallen viele unterschiedliche Aufgaben an. Für bestimmte Aufgaben ist es sinnvoll eine Person zu definieren, die immer als Ansprechpartner gelten kann! Grundsätzlich ist wechselseitiges informieren über den aktuellen Projektstand sinnvoll. Dabei sollten **positive Entwicklungen** und **mögliche Konflikte und Risiken** mitgeteilt werden.

Informieren Sie sich über gängige Rollen im Projektmanagement/Softwareentwicklung. Wählen Sie Rollen aus, die für ihr Projekt sinnvoll/notwendig erscheinen.

**Dokumentieren Sie die vergebenen Rollen und durch welche Mitglieder ihrer Gruppe diese ausgeführt werden.**

## Zieldefinition

Notieren Sie unter „Zieldefinition 1“ ihr Gesamtziel in Forme in Form einer Ausbaustufe, die Ihnen als Gruppe als realistisch erscheint. Passen Sie im Projektverlauf ggf. das angestrebte Ziel an, notieren Sie das neue Gesamtziel unter „Zieldefinition 2“ etc.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Angestrebte Ausbaustufen | | Zielanpassung | | |
|  | **funktionale Anforderungen** | **nicht funktionale Anforderungen** | **Anpassung der Zieldefinition** | **Datum und Grund der Änderung** | **Kommentar** |
| Zieldefinition 1: |  |  |  |  |  |
| Zieldefinition 2: |  |  |  |  |  |
| Zieldefinition 3: |  |  |  |  |  |

## Milestones

Milestones (dt. Meilensteine) sind Zwischenziele, die eine Zeitangabe verfügen. Milestones unterstützen eine realistische Zeitplanung. Große Ziele wirken schnell unrealistisch und bieten keinen Anhaltspunkt an dem die Arbeit begonnen werden kann. Modularisieren (unterteilen) Sie daher das Projektziel in Zwischenschritte und überlegen Sie sich eine sinnvolle Bearbeitungsreihenfolge. Passen Sie bei Bedarf die Meilensteine (Ziel und/oder Zeitpunkt) an. So können diese z.B. früher oder später erreicht werden oder unter Umständen vollständig entfallen.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Zieldatum | Milestone | Ziel erreicht am | Verantwortliche\*r | Problem/Verzögerung, weil... | Lösungsidee |
|  | 21.8.2017 | Erhalt des Projektauftrages |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | 13.09.2017 | Finale Abgabe des Projekts |  |  |  |  |
|  | 14.09.2017 | Projektpräsentation |  | Alle |  |  |
|  | 15.09.2017 | Reflexion  Abschließende Evaluation |  | Alle |  |  |

# Bewertungshinweise

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **LF[[2]](#footnote-2)** | **Kriterium** | **Wertung**  0 bis 10 | **Kommentar** |
| **Arbeitsprozess (20%)** | | | | |
|  | alle | Hoher Arbeitseinsatz |  |  |
|  | alle | Aktive Projektarbeit, guter Lernzuwachs |  |  |
| **I Modellierung (10%)** | | | | |
|  |  | Einhaltung von Model-View-Control (MVC) |  |  |
|  |  | Sinnvolle Aufteilung der Klassen |  |  |
|  |  | Sinnvolle Wahl der Attribute, Methoden und Datenstrukturen |  |  |
| **II Programmierung (40 %)** | | | | |
|  | 11 | Lauffähigkeit des Programms |  |  |
|  | 13 | Kommentierung des Programmcodes nach JavaDoc ist vorhanden |  |  |
|  | 11 | GUI mit JavaFX   * Nutzergerechter Aufbau der GUI * Umfang der Funktionalität |  |  |
|  | alle | Umfang der Ausbaustufen/Features |  |  |
|  | 11 | Login via HEMS-LDAP |  |  |
|  | 11 | Reibungslose Kommunikation zwischen Clients und Server |  |  |
|  | 11 | Ausnahmebehandlungen bei Fehlern / Stabilität des Clients und Servers |  |  |
|  | 11 | Sinnvoller Einsatz von Threads |  |  |
|  | 11 | Mehrbenutzerfunktionalität – keine Deadlocks/ sinnvolle Sperrung von Ressourcen/kritische Abschnitte |  |  |
|  | 11 | Synchronisation zwischen Client und Server |  |  |
|  | 11 | Persistente Speicherung in XML-Dokumenten |  |  |
|  | 11 | Besondere Features |  |  |
| **IV Dokumentation (15%)** | | | | |
|  | 13 | Rollenverteilung u. Aufgabenverteilung |  |  |
|  | 13 | Milestones   * Definition der Milestones * Zieladaption mit Begründung |  |  |
|  | 13 | Vollständig erstelltes/generiertes JavaDoc |  |  |
|  | 12 | * UML-Klassendiagramme der Model-Ebene * UML-Sequenzdiagramme * UML-Anwendungsfalldiagramme |  |  |
|  | 12 | Umgang mit Problemen   * Probleme wurden antizipiert * Probleme wurden benannt * Lösungsstrategien wurden dokumentiert |  |  |
| **IV Präsentation (15%)** | | | | |
|  | 13 | Aufbau und Strukturierung |  |  |
|  | 13 | Sprachliche Gestaltung |  |  |
|  | 13 | Zielgruppengerechte Darstellung |  |  |

1. **Legende**: „Bereich“ gibt das zugehörige Lernfeld an. Die Angabe **LF** entspricht Lernfeld 11 (frm), 12(brü) **und** 13 (brü). [↑](#footnote-ref-1)
2. **Legende**: „Bereich“ gibt das zugehörige Lernfeld an. Die Angabe **LF** entspricht den Lernfeldern 11 (frm), 12 (brü) **und** 13 (brü). [↑](#footnote-ref-2)