**Software-Engineering I**

**Programmentwurf  
TINF19B1  
3.+4. Semester (2020/2021)**

***Thema:   
Eventplaner***

DHBW Karlsruhe  
Studiengang Angewandte Informatik

Dr.-Ing. R. Lutz

Institut für Automation und angewandte Informatik (IAI)

Karlsruher Institut für Technologie (KIT)

Bearbeitende:

<Name1> und <Name2>

Inhalt

[1. Einleitung 3](#_Toc68107624)

[2. Lastenheft 3](#_Toc68107625)

[2.1. Zielsetzung 3](#_Toc68107626)

[2.2. Anwendungsbereiche 3](#_Toc68107627)

[2.3. Zielgruppen, Benutzerrollen und Verantwortlichkeiten 3](#_Toc68107628)

[2.4. Zusammenspiel mit anderen Systemen 3](#_Toc68107629)

[2.5. Produktfunktionen 4](#_Toc68107630)

[2.6. Produktdaten 5](#_Toc68107631)

[2.7. Produktleistungen 5](#_Toc68107632)

[2.8. Qualitätsanforderungen 5](#_Toc68107633)

[3. Aufgaben 6](#_Toc68107634)

[3.1. Analyse 6](#_Toc68107635)

[3.2. Sequenzdiagramm und Aktivitätsdiagramm 6](#_Toc68107636)

[3.3. Entwurf 6](#_Toc68107637)

[3.4. Implementierung 7](#_Toc68107638)

[4. Vereinfachungen für den Programmentwurf 8](#_Toc68107639)

[5. Besonderheiten 8](#_Toc68107640)

# Einleitung

Für unser sehr erfolgreiches Startup-Unternehmen im Bereich Event-Management benötigen wir ein neues Planungssystem, um alle Event-Daten noch besser und effizienter erfassen und verwalten zu können.

Dabei planen und veranstalten wir Events mit Teilnehmerzahlen bis ca. 1000 Personen (Kongresse, Hochzeiten, Empfänge, Vernissagen, Sportveranstaltungen, Wettbewerbe, Konzerte usw.). Die meisten Events führen wir für 50 - 100 Teilnehmer durch.

Bisher vor kurzem war es möglich, mit Hilfe von *Excel* die Planung und Verwaltung unserer Events durchzuführen, was durch die stark steigende Anzahl an Events nun nicht mehr auf Dauer realisierbar ist.

# Lastenheft

## Zielsetzung

Ziel des Entwicklungsauftrags ist eine Software für die Planung und Durchführung von Veranstaltungs-Events. Alle Daten sollen zentral gespeichert werden, da mehrere Benutzer gleichzeitig auf die Daten und Termine zugreifen werden.

Ein selektiver Import und Export von Daten über lesbare Dateien muss für Backups und zum Datenaustausch möglich sein.

Eine intuitive, leicht bedienbare Benutzeroberfläche setzen wir als selbstverständlich voraus. Es sollen keine besonderen Computerkenntnisse zur Bedienung der Software erforderlich sein.

## Anwendungsbereiche

Die Software soll ausschließlich für die Planung und Verwaltung von Events, Kunden, Ausrüstung, Locations und Angestellten und den damit direkt verbundenen Elementen verwendet werden. Sie soll im Alltag auf Laptops eingesetzt werden.

## Zielgruppen, Benutzerrollen und Verantwortlichkeiten

Es soll verschiedene Benutzerrollen geben:

* Organisatorinnen und Organisatoren pflegen die jeweiligen Event-Daten
* Beschaffungspersonal (besorgen und verwalten benötigte Utensilien, die mehrfach verwendet werden können). Es hat lesenden Zugriff auf von Organisatorinnen und Organisatoren freigegebene Teilevents. Sie können in Gruppen organisiert sein (d.h. es gibt eine/n Gruppenleiterin bzw. einen Gruppenleiter) für Aktionen, die nicht oder nur schwer allein zu realisieren sind.
* Personalmitarbeiter pflegen Mitarbeiterdaten im System
* Montageleiterinnen und Montageleiter (Leitung von Mitarbeitern für den Auf- und Abbau der benötigten Geräte, Bauten und Einrichtungen). Die Rolle soll lesenden Zugriff auf alle Daten haben, die in ihren Arbeitsbereich fallen.
* Eine hauptverantwortliche Person (Administrator) hat Vollzugriff auf sämtliche Daten, vor allem für deren Import und Export sowie deren Backup.

## Zusammenspiel mit anderen Systemen

Die Daten über die Angestellten (Gehälter bzw. Löhne, Steuern, Kranken- und Rentenversicherung usw.) werden separat durch ein vorhandenes Personalbuch­haltungs­programm verwaltet und müssen hier nicht berücksichtigt werden. Die finanztechnischen Daten werden über unser vorhandenes Finanzsystem erfasst und müssen hier ebenfalls nicht berücksichtigt werden.

Eine Web-Seite über unser Unternehmen existiert bereits, Anfragen für die Durchführung von Events werden per E-Mail gestellt und ist von der neuen Software unabhängig.

In einer zweiten Ausbaustufe soll es möglich sein, dass das Beschaffungspersonal und die Montageleiter über das Internet (Handy, Tablet) erledigte Aufgaben „abhaken“ können. Diese Funktionalität wird jedoch in der ersten Ausbaustufe noch nicht benötigt, die Erledigung einzelner Teilaufgaben wird dann noch per Mail oder Telefonanruf mit den Organisatorinnen und Organisatoren erledigt.

Allerdings benötigen wir ein klares Konzept, wie diese Erweiterung realisiert werden soll.

Möglichst alle Daten sollen vom alten in das neue System übertragen werden.

## Produktfunktionen

|  |  |
| --- | --- |
| /LF10/ | Der jeweilige Benutzer muss die Möglichkeit haben, über eine grafische Benutzeroberfläche alle für ihn relevanten Daten einfach und übersichtlich zu verwalten. |
| /LF20/ | Verwaltet werden sollen Events, die geplant und durchgeführt werden. Sie bestehen aus einzelnen Teilschritten (Teilevents), die parallel oder nacheinander ausgeführt werden können.  Jedes (Teil-)Event hat einen Start- und einen Ende-Termin, eine Bezeichnung (Name), Kontaktdaten (für Ansprechpersonen, z.B. Verwalter einer Location), eine Liste von benötigten Hilfsmitteln, eine Beschreibung, einen Status (erstellt, geplant, in Arbeit, fertig, usw.), die Möglichkeit, Kosten abzubilden sowie weitere Attribute. |
| /LF30/ | Hilfsmittel sind Tische, Stühle, Deko-Elemente (viele Varianten!), Gastronomie-Grills, u.v.m. Bei der Eintragung der Hilfsmittel soll jeweils die benötigte Anzahl angegeben werden können. Allen Hilfsmitteln müssen mehrere Termine zugeordnet werden können. Terminüberschneidungen müssen vermieden werden, um die Verfügbarkeit sicherzustellen. |
| /LF40/ | So genannte Event-Elemente sind:   * Catering * Musik (Bands, Musikerinnen und Musiker, DJs usw.) * Multimedia (Beschallung, Anlagen (Verstärker, Boxen, Mikrofone, ...)) * Personen (z.B. Entertainer, Pastor für Trauungen, Redner, Clowns, Comedians, Musiker (Bands) etc.) * Location (Veranstaltungsort, d.h. Lage, Adresse, Größe, u.v.m.)   Alle Event-Elemente können wiederum aufgeteilt werden (Teil-Event oder Teil-Element mit Start- und Ende-Termin sowie einen möglichen Verweis auf eine Firma incl. Ansprechpartner und evtl. Angebot, Vertragsdetails usw.) |
| /LF50/ | Für die Organisatorinnen und Organisatoren soll es möglich sein, Mails an Personen und Mitarbeiter mit Informationen über den Teilevent aus dem System heraus zu versenden. Dazu sollen die Kontaktdaten verwendet werden, die bei den Events eingetragen sind. |
| /LF60/ | Alle Angestellten müssen verwaltet werden. Jedem Teil-Event können die gewünschten Angestellten zugeordnet werden |
| /LF70/ | Zur einfacheren Eingabe der Daten soll es Auswahllisten für deren Eigenschaften geben, wo immer es möglich ist. Die Auswahllisten sollen auf einfache Weise erweiterbar sein. |
| /LF80/ | Sämtlichen Elementen sollen mehrere Bilder mit Titel zugeordnet werden können, die zentral auf einem Verzeichnis liegen sollen |
| /LF90/ | Bei der Zuordnung von eingetragenen Hilfsmitteln zu Teil-Events muss darauf geachtet werden, ob die Hilfsmittel im gewünschten Zeitraum verfügbar sind. |
| /LF100/ | Es muss möglich sein, alle Teil-Events für ausgewählte Angestellte in einer Auflistung anzeigen zu lassen. |

## Produktdaten

|  |  |
| --- | --- |
| /LD10/ | Die Daten sollen in einer zentralen Datenbasis (lesbare Dateien) abgespeichert werden. |

## Produktleistungen

|  |  |
| --- | --- |
| /LL10/ | Die Anzahl der zu verwaltenden Elemente wird auf ca. 50.000 geschätzt. |
| /LL20/ | Um bei HW- und SW-Anschaffungen und -neuerungen flexibel zu bleiben, ist auf Plattformunabhängigkeit besonders zu achten. |

## Qualitätsanforderungen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Produktqualität | sehr gut | gut | normal | nicht relevant |
| Funktionalität | X |  |  |  |
| Zuverlässigkeit |  | X |  |  |
| Effizienz |  | X |  |  |
| Benutzbarkeit | X |  |  |  |
| Wartbarkeit |  |  | X |  |
| Übertragbarkeit |  |  | X |  |

# Aufgaben

Einzelne Lastenheftpunkte sind bewusst offengehalten. Denken Sie darüber nach, welche Informationen zusätzlich sinnvoll oder auch notwendig sind. Recherchieren Sie evtl. nach einzelnen Zusammenhängen im Internet.

## Analyse

Für die Analyse sind zu erstellen:

* Analyse des Lastenhefts (Fragen und Antworten).
* Ein Use-Case-Diagramm der gesamten Anwendung incl. Beschreibung.
* Eine Verfeinerung des Use-Case-Diagramms incl. Beschreibung. (nach Absprache)
* Ein Analyse-Klassendiagramm incl. Beschreibung (Untersuchen Sie dabei den Einsatz geeigneter Analysemuster)
* Einfache GUI-Skizzen (Mockups) von mindestens zwei wesentlichen GUI-Komponenten (Hauptseite, Tabs, etc.). Die Skizzen können mit einem einfachen Grafikprogramm erstellt werden. Auch sorgfältige Handzeichnungen sind erlaubt. Keine Login-GUI skizzieren!

## Sequenzdiagramm und Aktivitätsdiagramm

Erstellen Sie ein Sequenzdiagramm und ein Aktivitätsdiagramm (incl. Beschreibung) für folgende Szenarios (ein AD für das eine Szenario, ein SD für das andere Szenario):

* Die Aktion „Event anlegen“ durchführen. Ausgehend von einem neuen Event und leerer Datenbasis werden dessen gesamte Daten erfasst und in das System eingetragen. (dies wird als Gebrauchsanweisung für die Evaluation Ihrer Implementierung dienen)
* Die Aktionen „Event durchführen“ anhand eines praktischen Beispiels (Kongress, Abschlussfeier, Konzert o.ä.)

Die Bewertung Ihrer Diagramme erfolgt auf der Basis der Nutzung der UML-Elemente, auf Ihrer Kreativität sowie dem Detaillierungsgrad des jeweiligen Diagramms.

Fassen Sie bei beiden Diagrammen die Eingabe aller primitiven Attribute eines Elements (Float, String, Integer, …) in einer einzigen Aktion zusammen (z.B. „Attribute eintragen“).

Für das Sequenzdiagramm ist das gewählte Szenario ausführlich zu entwickeln (idealer­weise mit Pseudocode). Es sind sämtliche referenzierten Elemente zu berücksichtigen, die zugeordnet werden können.

In allen Fällen wird eine (noch) leere Datenbasis angenommen. Denken Sie an geeignete Diagrammverfeinerungen.

## Entwurf

Abzuliefern sind hier (alle Diagramme und GUIs jeweils mit Beschreibung):

* Entwurfsklassendiagramm (Untersuchen Sie dabei den Einsatz geeigneter Entwurfsmuster)
* GUI-Modellierung:  
  Es ist das Kommunikationsschema eines Teils der während der Analyse skizzierten GUI mit **UML** zu modellieren. Die Anwendung selbst soll dabei nach dem einfachen Model-View-Control-Muster aufgebaut sein. Dazu sind mindestens ein Controller, die erforder­lichen Modellklassen sowie eine unabhängige GUI (View) erforderlich.

Die meisten GUI-Elemente werden über eine einfache kleine Java-Bibliothek zur Verfügung gestellt (*swe-utils*.*jar*), deren GUI-Komponenten in das Klassendia­gramm zu integrieren sind, wenn sie verwendet werden.

Die GUI-Modellierung kann in einem separaten Diagramm mit den relevanten (gewählten bzw. benötigten) Modellklassen erfolgen, falls das Entwurfsklassen­dia­gramm sonst zu komplex werden würde.

## Implementierung

Es ist eine einfache Java-Applikation zu implementieren, die es ermöglicht, Museumsdaten anzulegen, zu ändern und zu löschen.

Zur Realisierung wird die oben bei der Entwurfsaufgabe erwähnte Java-Bibliothek zur Verfügung gestellt (*swe-utils*.*jar*), die neben mehreren GUI-Komponenten einen *CSVReader,* einen *CSVWriter* sowie mehrere Interfaces bereitstellt (in den Packages *event* und *model*).

Daneben ist eine Mini-Test-Applikation gegeben, die die Funktionsfähigkeit der GUI-Komponenten demonstriert (Start mit *java -jar swe-utils.jar*). Details sind der Java-Dokumentation der Bibliothek zu entnehmen.

Zur leichteren und zukunftssicheren Evaluation Ihres Programmentwurfs soll die Java-Applikation als eine Desktop-Applikation mit CSV-Dateien (alternativ XML oder JSON) als zentrale Datenbasis realisiert werden, die von beliebigen Rechnern aus gestartet wird. Dabei sind mehrere Dateien analog zu Datenbanktabellen zu erzeugen.

Einzelne Aufgaben

* Hauptaufgabe ist die Realisierung einer MVC-Applikation mithilfe des Observer-Patterns entsprechend des vorgegebenen GUI-Entwurfs und der gegebenen Java-Bibliothek.
* Die Erzeugung der Instanzen soll in einer Entity-Factory erfolgen und zur Verwaltung der Instanzen ist ein Entity-Manager zu realisieren (beides siehe Vorlesung).
* Beim Anlegen eines Events muss für die Zuordnung von Hilfsmitteln sichergestellt sein, dass es keine zeitlichen Überschneidungen gibt (LF90).
* Es muss eine ausführbare JAR-Datei abgegeben werden, die mit

„java -jar SWE-PE-2021\_Eventplaner\_<name1>\_<name2>.jar OPTIONEN“

gestartet werden kann. Hierfür ist ein BASH-Skript namens *startApp* zu erstellen.

* Geprüft wird das Anlegen eines Events mit der Zuordnung aller zugehörigen Elemen­te. Nach dem Anlegen wird die Applikation erneut gestartet und geprüft, ob alle Daten korrekt abgespeichert und beim Laden wieder zugeordnet werden.

Verwendung von CSV-Dateien:

* Die Daten sollen in CSV-Dateien vorliegen und können mittels den gegebenen Bibliotheksklassen *CSVReader* und *CSVWriter* gelesen bzw. beschrieben werden. Zur Vereinfachung können die Daten jeweils komplett geschrieben werden.
* Abgegeben werden soll ein ZIP-File (oder TAR-File) mit allen Java- und CSV-Dateien (letztere gesammelt in einem eigenen Verzeichnis):

„SWE-PE-2021\_Eventplaner\_<n1>\_<n2>.zip (tar oder tar.z)

* Als OPTIONEN in der Startanweisung soll der Pfad zu den CSV-Dateien sowie zu einer Properties-Datei angegeben werden können:

„java -jar SWE-PE-2021\_Eventplaner\_<n1>\_<n2>.jar **–d <csvpath> –p <propfile>**“

# Vereinfachungen für den Programmentwurf

1. Es muss nicht dafür gesorgt werden, dass auf dieselben Daten bzw. CSV-Dateien nicht gleichzeitig zugegriffen werden kann, d.h. es ist kein *Locking*-Mechanismus erforderlich.
2. Eine Protokollierfunktion und ein Login-Vorgang sind für die Anwendung nicht erforderlich (in der Realität natürlich schon!).
3. Zeitliche Überschneidungen sind natürlich bei allen Elementen mit mehreren Terminangaben möglich und müssten sowohl beim Anlegen als auch bei Änderungen von Terminen berücksichtigt werden. Im Programmentwurf sollte dies in der Modellierung berücksichtigt werden, bei der Implementierung ist jedoch nur eine Überprüfung beim Zuordnen eines Hilfsmittels zu einem Event erforderlich.
4. In einem vollständigen Modell finden wir natürlich bei fast allen Elementen, die einem Event zugeordnet werden, eine N:M-Beziehung vor. Bei der Implementierung müssen jedoch nur die N:M-Beziehungen zwischen Event und zugeordneten Mitarbeitern (s. Benutzerrollen) realisiert werden, alle anderen Elemente dürfen mit einer einseitigen Zuordnung implementiert werden (s. LF100)

Alle hier aufgeführten, farbig hinterlegten Zeilen bitte nach Erledigung der Aufgaben löschen!

Nun die komplette Aufgabenstellung kopieren und hier einfügen (🡺 als originale Aufgabenstellung und als Template für die Fragen und Antworten 🡺 Überschrift „Analyse“).

Hier kommen nun neben „Analyse“ Ihre eigenen Abschnitte hin, **letzter** Abschnitt ist (mit erhöhter Nummer):

# Besonderheiten

In diesem Abschnitt nennen Sie alle Besonderheiten, die ich als Bewertender zusätzlich beurteilen soll, damit ich Ihre besonderen Ideen und Realisierungen beim Korrigieren nicht übersehe.