

Projektplanung „Gamer’s Memory“

von Alena Schönberg
erstellt am 07.09.2020

Inhaltsverzeichnis

1. Vorgaben.....	1
2. Ausgangsidee.....	1
3. Allgemeine Beschreibung des Projektziels.....	2
4. Ist-Analyse.....	2
5. Soll-Analyse.....	2
6. Aufgabenaufteilung – Milestones.....	3

1. Vorgaben

Folgende Punkte müssen erfüllt werden:

- Singleton verwenden
- GUI-Applikation
- Einhaltung der Codevorgabe (Java/Objektorientiert)
- Verwendung Modelklassen
- Projektplanung (Ist/Soll, Milestones)
- Persistente Datenspeicherung mit MariaDb
- Persistente Datenspeicherung in einer Datei
- Stimmiger Gesamteindruck und Usability
- Eigene Arbeit, kein Kopieren (einzige Ausnahme sind Libraries und DB-Vorlagen)
- Zwei Wochen Zeit

2. Ausgangsidee

„Chris“ ist oft bei Gesellschaftsspiele-Veranstaltungen und -Messen. Dort lernt sie in recht kurzer Zeit viele neue Spiele kennen. Manche spielte sie nur an und andere mehrfach, von manchen hört sie etwas und andere sieht sie nur.

Inzwischen ist sie sich, wenn sie nach einem Spiel gefragt wird oder es sieht, nicht mehr sicher ob sie es gespielt hat und wie ihre Einschätzung dazu war. Sie wünscht sich ein digitales „Spieler-Gedächtnis“.

3. Allgemeine Beschreibung des Projektziels

Eine PC-Anwendung soll zum Verwalten von Gesellschaftsspielen, den dazugehörigen allgemeinen Daten und einer individuellen Bewertung des Spiels bereitgestellt werden. Eine einfache grafische Benutzeroberfläche soll das Erfassen, Ändern, Löschen und Anzeigen von Spielen ermöglichen.

Die Daten sollen in einer lokalen Datenbank gespeichert werden. Zusätzlich soll ein Im- und Export als CSV-File angeboten werden, um die Daten auch unabhängig von der PC-Anwendung nutzen zu können.

Die Anwendung wird für „Spiele-Kenner“ entwickelt, nicht für Gelegenheitsspieler.

4. Ist-Analyse

Die Kundin „Chris“ versucht sich die Spiele zu merken, die sie während Veranstaltungen spielt und wie sie diese bewertet. Durch die große Menge an Spielen ist ihr das nicht mehr möglich.

Sie ärgert sich oft, dass sie die Namen von Spielen vergisst, die sie nur einmal gespielt hat. Auch fehlt ihr eine Übersicht, über die Spiele in ihrem Besitz. Sie kann nicht schnell ermitteln, mit wie vielen Personen und mit welchem Zeitaufwand diese spielbar sind. So muss sie immer wieder von neuem einzelne Spiele kurz an lesen, wenn ein Spieleabend geplant wird.

Sie wünscht sich eine Anwendung, die ihr Gedächtnis entlastet und die Entscheidungsfindung (Neukauf und Spieleabend-Planung) unterstützt.

5. Soll-Analyse

Um die Kundin in ihrem Hobby zu unterstützen soll eine JavaFX-Anwendung implementiert werden. Diese soll mittels einer MariaDB die benötigten Informationen zu den gespielten Spielen verwalten können. Die erste Version der Anwendung soll so einfach wie möglich gehalten werden und vor allem eine gute Übersicht über die bereits gespielten Spiele bieten.

Die in der Ist-Analyse beschriebenen Probleme sollen behoben werden. Dazu wird der Kundin ermöglicht, die Spiele (mit allen zur Spieleabend-Planung und der Entscheidung für einen Neukauf benötigten Informationen) zu erfassen, zu speichern, einzusehen und individuell zu bewerten.

6. Aufgabenaufteilung – Milestones

1. Woche

1.1. Tag:

- a) Projektplanung
- b) Vorbereitungen: Allgemeine Struktur und erste Kommentare

1.2. Tag:

- a) Abstract Modell („ABaseModel“)
- b) Modell („Game“)
- c) Testklasse („Testdata“), Modell testen

1.3. Tag:

- a) Korrekturlesen der Projektplanung
- b) **Meilenstein:** Modellklasse + Projektplanung formuliert/korrigiert
- c) Anzeige der Spiele-Liste per GUI (GUI als Singleton)
- d) GUI-Textfelder/Buttons (beginnen)

1.4. Tag:

- a) GUI-Elemente vervollständigen
- b) Validierung der User-Eingaben
- c) GUI Testen - Zur Laufzeit: Anlegen, Editieren, Löschen

1.5. Tag (halber Tag):

- a) Prüfung auf Codevorgaben/Abschlussarbeit und ggf ändern
- b) Bugfixing-Puffer
- c) **Meilenstein:** GUI mit Anlegen, Ändern, Löschen von Spielen zur Laufzeit

2. Woche:

2.1. Tag:

- a) CSV-Handling („Ressource“)
- b) CSV Up-/Download aus GUI
- c) CSV-Handling Testen
 - Im- und Export zur Laufzeit
 - mit Daten-Änderungen aus GUI und in CSV
- d) **Meilenstein:** CSV-Im- und -Export

2.2. Tag:

- a) DB-Tabells
- b) DB-Anbindung (+ Adao/AsqlKeyWords „Kopieren erlaubt“)
- c) DB-CRUD

2.3. Tag:

- a) DB-CRUD vervollständigen
- b) DB-Testen
- c) **Meilenstein:** DB-Anbindung

2.4. Tag:

- a) Abschließende Tests und Prüfung auf Codevorgaben/Abschlussarbeit und ggf ändern
- b) Bugfixing-Puffer

2.5. Tag:

- a) Bugfixing-Puffer
- b) Finale
 - DB als SQL-Script in „Ressource“, Projekt zip (File → ExportToZipFile), Projektplanung als PDF
- c) **Meilenstein:** Abgabe (18.09.2020, spätestens 12:15 Uhr)