**Jira Synchronizer für das interne Leistungserfassungssystem**

**Individuelle Praktische Arbeit**

Simon andré Lötscher

Softec AG

Inhaltsverzeichnis

[Teil 1 – Umfeld und Ablauf 6](#_Toc161903122)

[1 Aufgabenstellung 6](#_Toc161903123)

[1.1 Titel der Arbeit 6](#_Toc161903124)

[1.2 Ausgangslage 6](#_Toc161903125)

[1.3 Detaillierte Aufgabenstellung 6](#_Toc161903126)

[1.3.1 Kriterien für den Import von Zeitbuchungen einer Aufgabe 6](#_Toc161903127)

[1.3.2 Felder welche beim Import von Jira Zeitbuchungen abgefüllt sein müssen 7](#_Toc161903128)

[1.3.3 Out of Scope 7](#_Toc161903129)

[1.3.4 Vorgaben 7](#_Toc161903130)

[1.4 Mittel und Methoden 7](#_Toc161903131)

[1.5 Vorkenntnisse 7](#_Toc161903132)

[1.6 Vorarbeiten 8](#_Toc161903133)

[1.7 Neue Lerninhalte 8](#_Toc161903134)

[1.8 Arbeiten in den letzten 6 Monaten 8](#_Toc161903135)

[2 Projektaufbauorganisation 8](#_Toc161903136)

[2.1 Projektumfeld 8](#_Toc161903137)

[2.1.1 Softec AG 8](#_Toc161903138)

[2.2 Projektorganisation 9](#_Toc161903139)

[2.3 Projektmethode 10](#_Toc161903140)

[2.3.1 Vergleich von Projektmethoden 10](#_Toc161903141)

[2.3.1.1 IPERKA 10](#_Toc161903142)

[2.3.1.2 Wasserfall 10](#_Toc161903143)

[2.3.1.3 Scrum 10](#_Toc161903144)

[2.3.2 Auswahl der Projektmethode 10](#_Toc161903145)

[2.3.3 Anwendung der ausgewählten Projektmethode 10](#_Toc161903146)

[2.4 Verwendete Technologien und Tools 11](#_Toc161903147)

[2.4.1 Entwickler-Tools 11](#_Toc161903148)

[2.4.1.1 Editoren 11](#_Toc161903149)

[2.4.1.2 Source Code Verwaltung 11](#_Toc161903150)

[2.4.1.3 Datenbank Management Tool 11](#_Toc161903151)

[2.4.2 Dokumentation 12](#_Toc161903152)

[2.4.3 Organisation und Sicherung der Arbeitsergebnisse 12](#_Toc161903153)

[3 Projektplanung 13](#_Toc161903154)

[3.1 Use Case 13](#_Toc161903155)

[3.2 Anforderungsliste 13](#_Toc161903156)

[3.3 User Stories 15](#_Toc161903157)

[3.4 Zeitplan 21](#_Toc161903158)

[3.4.1 Soll-Ist-Vergleich 22](#_Toc161903159)

[4 Arbeitsjournal 23](#_Toc161903160)

[Teil 2 – Projekt 34](#_Toc161903161)

[5 Kurzfassung 34](#_Toc161903162)

[6 Ausgangslage 35](#_Toc161903163)

[6.1 Einleitung 35](#_Toc161903164)

[7 Funktionalitätsumsetzung 36](#_Toc161903165)

[7.1 Einleitung 36](#_Toc161903166)

[8 Testing 37](#_Toc161903167)

[8.1 Einleitung 37](#_Toc161903168)

[8.2 Umfeld 37](#_Toc161903169)

[8.2.1 Testanlage 37](#_Toc161903170)

[8.2.2 Mittel und Methoden 37](#_Toc161903171)

[8.3 Testdrehbuch 37](#_Toc161903172)

[8.4 Testprotokoll 37](#_Toc161903173)

[8.4.1 Unit-Tests 37](#_Toc161903174)

[8.4.2 Abnahmetests 37](#_Toc161903175)

[8.5 Testreport 37](#_Toc161903176)

[8.5.1 Resultatübersicht 37](#_Toc161903177)

[8.5.2 Auswertung 37](#_Toc161903178)

[9 Schlusswort 38](#_Toc161903179)

[9.1 Fazit 38](#_Toc161903180)

[9.1.1 Projektplanung 38](#_Toc161903181)

[9.1.1.1 Zeitplanung 38](#_Toc161903182)

[9.1.1.2 Projektmethode 38](#_Toc161903183)

[9.1.2 Implementation 38](#_Toc161903184)

[9.1.3 Dokumentation 38](#_Toc161903185)

[9.1.4 Testing 38](#_Toc161903186)

[9.2 Nächste Schritte 38](#_Toc161903187)

[10 Verzeichnisse 39](#_Toc161903188)

[10.1 Glossar 40](#_Toc161903189)

[10.2 Literatur 41](#_Toc161903190)

[10.3 Abbildungsverzeichnis 42](#_Toc161903191)

[11 Anhang 43](#_Toc161903192)

[11.1 Git-Log 43](#_Toc161903193)

Über dieses Dokument

Die Projektdokumentation der IPA ist, wie von PkOrg vorgegeben, in zwei Teile unterteilt. Einzelne Themengebiete werden zusätzlich in Kapitel aufgeteilt. In den Fussnoten werden Informationen zu den in diesem Projekt verwendeten Technologien und Tools angegeben. Die verwendeten Quellen wurden in der Literatur festgehalten.

Akronyme, Glossar, Literatur und das Abbildungsverzeichnis sind im Kapitel Verzeichnisse zu finden.

Typografische Elemente

Im folgenden Abschnitt werden die typografischen Elemente dieses Dokuments genauer gezeigt und erklärt.

**ⓘ Informationsblock**

Ein solcher Informationsblock beinhaltet Informationen über ein zu dieser Arbeit zusammenhängendes Thema. Dieser Block soll dabei helfen, den Gesamtzusammenhang dieser IPA zu verstehen.

/\*\*

\* Source code snippets are also marked in information blocks.

\* The whole source code and all comments are written in english.

\*/

console.log('Hello World!');

Dokumentenhistorie

|  |  |
| --- | --- |
| Datum | Beschreibung |
| TT.MM.JJJJ | - Dokument erstellt anhand Aufgabenstellung und Kriterienkatalog  - Arbeitsjournal Tag 1 |
| TT.MM.JJJJ | -  - Arbeitsjournal Tag 2 |
| TT.MM.JJJJ | -  - Arbeitsjournal Tag 3 |
| TT.MM.JJJJ | -  - Arbeitsjournal Tag 4 |
| TT.MM.JJJJ | -  - Arbeitsjournal Tag 5 |
| TT.MM.JJJJ | -  - Arbeitsjournal Tag 6 |
| TT.MM.JJJJ | -  - Arbeitsjournal Tag 7 |
| TT.MM.JJJJ | -  - Arbeitsjournal Tag 8 |
| TT.MM.JJJJ | -  - Arbeitsjournal Tag 9 |
| TT.MM.JJJJ | -  - Arbeitsjournal Tag 10  - Fertigstellung / Überarbeitung |

Teil 1 – Umfeld und Ablauf

1. Aufgabenstellung
   1. Titel der Arbeit

Jira Synchronizer für das interne Leistungserfassungssystem

* 1. Ausgangslage

An der Softec werden Zeitbuchungen in einem internen Leistungserfassungssystem (LEIS) vorgenommen. Ein Grossteil der Mitarbeiter arbeitet zusätzlich mit Jira und erfässt dort dieselben Zeitbuchungen auf den bearbeiteten Tickets. Dies führt zu einem Mehraufwand, welcher sich je nach Mitarbeiter auf wöchentlich mehrere Stunden belaufen kann. Derzeit wird das aktuelle LEIS durch eine Neuentwicklung abgelöst. Im Rahmen dieser Neuentwicklung ist der Wunsch nach einer Synchronisierung der in Jira erfassten Zeiten in das neue LEIS aufgekommen, damit die Zeiten nicht in zwei Systemen manuell eingeben werden müssen.

* 1. Detaillierte Aufgabenstellung

Der Jira Synchronizer soll als Kommandozeilen Programm mit der Programmiersprache C# implementiert werden. Die Parameter für die Ausführung des Kommandozeilen Tools sollen Flags für einen Development Modus (-dev / -d) umfassen. Für die Synchronisierung soll das Kommandozeilen Programm die REST API von Jira verwenden und anhand einer Projekt Whitelist die Aufgaben in Jira ermitteln, welche Zeitbuchungen haben und diese dann für den Import vorsehen. Dabei müssen entsprechende Importregeln beachtet werden. Der Import muss grundsätzlich beliebig oft am Tag gestartet werden können, allerdings darf immer nur eine Instanz des Imports laufen. Der Import und die Businesslogik sollen mittels Unit Tests getestet werden.

* + 1. Kriterien für den Import von Zeitbuchungen einer Aufgabe
* Projekt der Jira Aufgabe ist in der LEIS Datenbank in der Tabelle «T\_WHITELIST» erfasst.
* Die Jira Aufgabe muss über Zeitbuchungen verfügen.
* Das Feld "LEIS Konto" der Jira Aufgabe ist mit einem passenden LEIS Konto abgefüllt.
* Sollte das Feld leer oder mit einem ungültigen LEIS Konto abgefüllt sein, dann muss dies in ein Error Log geschrieben werden.
* Der Mitarbeiter muss auf das LEIS Konto buchen dürfen, sprich er muss berechtigt sein. Ist dies nicht der Fall, so darf die Zeitbuchung nicht in der LEIS Datenbank erfasst werden und es muss ein Eintrag im Error Log erfasst werden.
* Die LEIS Buchung muss mit der Jira ID der Jira Zeitbuchung verknüpft werden.
* Es werden die Jira Zeitbuchungen der letzten 7 Kalendertage importiert, sofern diese nicht aus einem vorherigen Durchlauf importiert wurden.
* Bei aktivierten Development Modus sollen alle Zeitbuchungen auf einem bestimmten Mitarbeiter verbucht werden. Dies dient ausschliesslich der Kontrolle während der Entwicklungszeit. Dies ist nötig, weil die Development (und Test-) Datenbank anonymisiert ist. Der Mitarbeiter für diese Ausführung soll in den AppSettings definiert sein.
  + 1. Felder welche beim Import von Jira Zeitbuchungen abgefüllt sein müssen
* Projekt
* Leistungsart (hier wird immer «Realisierung» als Wert verwendet)
* Mitarbeiter
* Datum (von/bis)
* Geleistete Zeit
* Beschreibung (hier wird jeweils die Jira Ticket Nummer verwendet)
* Interner Kommentar (hier wird der Kommentar der Zeitbuchung in Jira verwendet)
* Verrechenbar (wird von der Projekteinstellung in der LEIS Datenbank abgeleitet)
* Zeitklasse (hier wird immer «Normalzeit» als Wert verwendet)
  + 1. Out of Scope
* Die in der Jira Aufgabe eingetragene geleistete Zeit wird eins zu eins übernommen, auch wenn die Zeitbuchung grösser als 24 Stunden ist.
  + Wenn die Geleistete Zeit länger als 24h ist, soll eine Warnung in das Error Log geschrieben werden.
* Zeitklasse "Überzeit" wird nicht automatisch gesetzt.
* Jira Zeitbuchungen, welche schon importiert wurden werden nicht erneut auf Änderungen geprüft und werden auch nicht automatisch aktualisiert.
* Rundung von Zeitbuchungen auf 15 Minuten werden nicht automatisch vorgenommen (bei den meisten Projekten müssen Mitarbeiter auf 15 Minuten Intervalle runden, jedoch nicht bei allen).
* Projekte werden in LEIS nicht automatisch angelegt.
* Die Whitelist muss manuell in der LEIS Datenbank abgefüllt werden.
* Zeitbuchungen werden immer für die vergangenen sieben Tage importiert.  
  Es wird dabei nicht beachtet, ob bei einem Monatswechsel der Vormonat für die Zeitbuchungen durch Mitarbeiter bereits gesperrt wurde.
  + 1. Vorgaben

Der Jira Synchronizer wird als Kommandozeilen Programm erstellt, welches in einem Scheduler zeitgesteuert und automatisch ausgeführt werden können soll.

* 1. Mittel und Methoden
* Arbeitslaptop (durch Softec zur Verfügung gestellt)
* Visual Studio Professional 2022 und Visual Studio Code
* Microsoft SQL Server Management Studio 19
* Microsoft SQL Datenbankserver 2016
* Als Programmiersprachen werden C# und T-SQL verwendet.
  1. Vorkenntnisse

Als Lehrlingsprojekt durfte ich mit der Neuentwicklung des neuen LEIS beginnen. Voraussetzung für das neue LEIS ist es, auf die alte LEIS Datenbank aufzusetzen. Als solches ist auch die Kenntnis des alten Datenbankschemas und seiner Eigenheiten aufzuführen. Da das LEIS und seine Datenbank beide mittlerweile weit über zehn Jahre alt sind, verfügt die Datenbank über ein paar Eigenheiten. Als solches arbeitet sie exklusiv mit Triggern und verfügt über keine Foreign Keys.

Ich arbeite täglich mit der Programmiersprache C# und mit der Sprache T-SQL bin ich durch meine Arbeit mit der LEIS Datenbank sowie einigen anderen Projekten der Softec vertraut. Ich verwende Visual Studio Professional 2022 und Visual Studio Code praktisch täglich, das Microsoft SQL Server Management Studio 19 habe ich auch schon oft verwendet, und mit dem Microsoft SQL Datenbankserver 2019 bin ich vertraut.

* 1. Vorarbeiten

Zu den Vorarbeiten gehört eine Machbarkeitsstudie, in welcher ich schaute ob ich die Jira API wie erwartet ansteuern und Daten davon importieren kann. Das aus der Studie hervorgegangene Werkstück wurde wieder restlos gelöscht. Backups gibt es ebenfalls keine.

Weiterhin ist hier die begonnene Neuentwicklung vom LEIS sowie das Studium des Datenbankschemas aufzuführen. Für die IPA wird eine anonymisierte Version der LEIS Datenbank verwendet.

* 1. Neue Lerninhalte

Trotz der durchgeführten Machbarkeitsstudie ist die Jira API als neuer Lerninhalt aufzuführen, da die Machbarkeitsstudie lediglich grob testete, ob die Jira API wie erwartet ansteuerbar ist, nicht wie sie detailliert funktioniert, wie man sich darauf zu orientieren hat, oder was potentielle Schwierigkeiten sind.

* 1. Arbeiten in den letzten 6 Monaten
* Neuentwicklung vom LEIS
* Reports auf Basis von DevExtreme Reports erstellen
* Erweiterungen an bestehenden ASP.net Core Web APIs für die Reports und für die Angular Webapplikationen
* QR Code Rechnungsgenerator als Kommandozeilen Programm

Es waren alle aufgeführten Aufgaben von grösserem Umfang. Dabei kamen die Programmiersprachen C# und T-SQL, sowie das Framework Angular zum Einsatz. Diese Arbeiten wurden in Vollzeit ausgeführt.

1. Projektaufbauorganisation
   1. Projektumfeld

[Umwelt des Projektes, Gruppen des Projektes identifiziert, Abgrenzungen des Projektes darstellen]

* + 1. Softec AG
  1. Projektorganisation

**Dani Bomatter Hauptexperte**

E-Mail dani.bomatter@sisag.ch

Telefon 041 875 07 33

**Stefan Spirgi Nebenexperte**

E-Mail stefan.spirgi@vontobel.com

Telefon 079 217 74 18

**Jens Eger Verantwortliche Fachkraft**

E-Mail jens.eger@softec.ch

Telefon 041 747 07 23

**Michael Näf Unterstützende Fachkraft**

E-Mail michael.naef@softec.ch

Telefon 041 747 07 39

**Christian Schambron Leu Berufsbildner**

E-Mail christian.schambron-leu@bildxzug.ch

Telefon 041 544 77 04

**Simon André Lötscher Lernender / Entwickler**

E-Mail simon.loetscher@softec.ch

Telefon 041 755 34 06

* 1. Projektmethode
     1. Vergleich von Projektmethoden
        1. IPERKA
        2. Wasserfall
        3. Scrum
     2. Auswahl der Projektmethode
     3. Anwendung der ausgewählten Projektmethode
  2. Verwendete Technologien und Tools
     1. Entwickler-Tools
        1. Editoren
        2. Source Code Verwaltung
        3. Datenbank Management Tool
     2. Dokumentation
     3. Organisation und Sicherung der Arbeitsergebnisse

1. Projektplanung
   1. Use Case

Das Use Case Diagramm stellt die Benutzer- sowie Entwickleranforderungen grafisch dar.

Der Benutzer ist bei uns der Mitarbeiter welcher auf Jira seine Leistungen in den jeweiligen Projekten festhält.

Entwickler ist die Person, welche die Applikation entwickelt und in der Zukunft verwaltet.

Ein Bild, das Text, Entwurf, Diagramm, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 3.1 Use Case Diagramm

* 1. Anforderungsliste

Die Anforderungsliste wird der detaillierten Aufgabenstellung sowie den individuellen Bewertungskriterien entnommen.

**Nummer Beschreibung Priorität**

A1 **Testing**

A1.1 Die Applikation wird durch Unit Tests getestet. Hoch

A1.2 Die Unit Tests decken den Grossteil der Applikation ab. Hoch

A1.3 Manuelle Tests der Applikation sind erstellt. Mittel

A1.4 Die Durchführung aller Tests inklusive deren Resultate ist dokumentiert. Hoch

A1.5 Die Solution verfügt über ein separates Unit Test Project. Mittel

A2 **Logging**

A2.1 Während der Laufzeit der Applikation wird ein Log geführt. Hoch

A2.2 Unerwartete Fehler in der Applikation werden im Log als Fehlermeldung Hoch

hinterlegt.

A2.3 Fehlt das Feld «Leis Konto» im Jira Projekt, muss dies im Log als Fehler Hoch

hinterlegt werden.

A2.4 Hat der Mitarbeiter im Ticket keine Berechtigung auf das Projekt wird Hoch

die Leistung nicht abgefüllt und es wird ein Fehler im Log hinterlegt.

A2.5 Fehlen andere Angaben müssen diese als Warnung im Log erwähnt werden. Mittel

A3 **Schichtentrennung**

A3.1 Die Applikation ist in eine Persistenz-, Service-, und eine Applikationsschicht Hoch

getrennt (Core, Infrastructure, Application).

A3.2 Das Projekt ist gemäss anderen ähnlichen Firmenprojekten aufgebaut . Mittel

A4 **Software-Architektur**

A4.1 Es wurde eine passende Klassenstruktur gewählt. Hoch

A4.2 Die Klassen haben wohldefinierte, dokumentierte Schnittstellen. Mittel

A4.3 Die Klassen sind in passenden Modulen untergebracht. Hoch

A5 **Code**

A5.1 Mögliche Fehler werden erkannt und behandelt. Hoch

A5.2 Der Code ist gemäss Softec Vorgaben aufgebaut (Namensgebung, Lesbarkeit). Hoch

A6 **Robustheit**

A6.1 Abfüllen von fehlenden Daten wird verhindert. Hoch

A6.2 Gültigkeit angegebener Daten wird geprüft. Mittel

A6.3 Gefundene Unstimmigkeiten werden im Log erfasst. Mittel

A7 **Git-Repository**

A7.1 Der Code ist auf einem Github Repository vorhanden. Hoch

A7.2 Die Dokumentation ist auf dem selben Repository vorhanden. Hoch

A7.3 Notizen, Entwürfe, Bilder sind im Repository vorhanden. Tief

A8 **Applikation**

A8.1 Die Applikation kann über einen Scheduler aufgerufen werden. Hoch

A8.2 Der Applikation kann ein Development Flag ab Aufruf übergeben werden. Hoch

A8.3 Die fertige Applikation ist ein Kommandozeilenprogramm. Hoch

* 1. User Stories

Die User Stories werden benötigt um den Zeitplan sowie das im Verlauf des Projekts verwendete Kanban Board zu erstellen. Sie werden für eine möglichst genaue Planung des Projektes benötigt.

Projektplanung – S1

Anforderungen

A4.1, A4.2

Beschreibung

Als Kandidat ist es mir wichtig, mein Projekt möglichst gut zu planen damit ich im vorgegebenen Zeitraum alle Anforderungen realisieren und das Projekt umsetzen kann.

Tasks

**Nummer Beschreibung Zeit Ablauf**

T1.1 Use Cases definieren, graphisch darstellen. 1h 1

T1.2 Anforderungen definieren. 2h 2

T1.3 User Stories erarbeiten. 3h 3

T1.4 Zeitplan erstellen. 1h 4

T1.5 Vervollständigung Aufgabenstellung. 2h 5

T1.6 Vervollständigung Projektplanung. 2h 6

T1.7 Projektauftrag dokumentieren. 5h 7

T1.8 Ausgangslage dokumentieren. 1h 8

T1.9 Architektur dokumentieren. 1h 9

T1.10 Klassenstruktur mit Diagramm erstellen. 2h 10

**Aufwand Total** 20h

Akzeptanzkriterien

* Use Cases wurden definiert sowie per Grafik der Dokumentation hinzugefügt.
* Anforderungen wurden auf der detaillierten Aufgabenstellung aufbauend festgelegt.
* User Stories wurden erarbeitet und mit realistischer Aufwandsschätzung ausgestattet.
* Zeitplan wurde erstellt und auf PkOrg abgegeben.
* Die Aufgabenstellung wurde dokumentiert.
* Die Projektplanung wurde dokumentiert.
* Der Projektauftrag wurde dokumentiert.
* Die Ausgangslage wurde auf der detaillierten Aufgabenstellung aufbauend dokumentiert.
* Der Architekturentscheid wurde dokumentiert.
* Eine Klassenstruktur mit allen voraussichtlich benötigten Klassen wurde erstellt. Deren Beziehungen sind klar definiert und gut ersichtlich.

Projektinitialisierung – S2

Anforderungen

A1.5, A3, A4.3, A7, A8.1, A8.3

Beschreibung

Als Kandidat möchte ich meine Solution gemäss meiner gewählten Architektur erstellen und soweit vorbereiten dass ich damit in den nächsten Schritten ohne Unterbrechungen weiter arbeiten kann.

Tasks

**Nummer Beschreibung Zeit Ablauf**

T2.1 Github Repository aufsetzen und mit lokalem Repository verbinden. 1h 1

T2.2 Solution mit allen Projekten (Core, Infrastructure, Application, 1h 2

Unit Tests) erstellen.

T2.3 Datenbank um «T\_WHITELIST» Tabelle erweitern. 1h 3

T2.4 «T\_LEISTUNGSERFASSUNG» Tabelle in der Datenbank wird um 1h 4

Spalte «LEI\_JIRA\_ID» erweitert.

T2.5 Alle in der Klassenstruktur enthaltenen Klassen werden als 2h 5

Entitäten im Core Projekt und als ViewModel im Application Projekt

Erstellt.

**Aufwand Total** 6h

Akzeptanzkriterien

* Initial Commit auf Github Repository wurde erstellt.
* Solution ist wie finale Applikation strukturiert (Application Project soll Kommandozeilen Project sein)
* Datenbank ist auf die weitere Entwicklung des Projektes vorbereitet.
* Alle in der Klassenstruktur enthaltenen Klassen wurden in die Solution eingebaut. Vorhanden sind die Entitäten im Core Projekt welche mit der Datenbank kommunizieren und die ViewModels im Application Projekt welche von der Applikation verwendet werden.

Abfragen Jira Rest API – S3

Anforderungen

A5.1, A6.1, A6.2

Beschreibung

Als Kandidat möchte ich erste Abfragen an die Jira API senden und Antwort erhalten. Hierbei möchte ich nur Anfragen an Projekte senden, welche in der «T\_WHITELIST» Tabelle in der Datenbank enthalten sind.

Tasks

**Nummer Beschreibung Zeit Ablauf**

T3.1 «T\_WHITELIST» Tabelle in der Datenbank befüllen. 1h 1

T3.2 Atlassian API-Token für Jira erstellen. 1h 2

T3.3 Alle Zeitbuchungen welche noch nicht in der Datenbank 2h 3

vorhanden sind, werden für die letzten 7 Tage importiert.

T3.4 Testen ob alle nötigen Angaben mit plausiblen Werten gefüllt sind. 1h 4

T3.5 Testen ob hinterlegter Mitarbeiter innerhalb einer Buchung die 1h 5

benötigten Rechte auf das hinterlegte Leis Konto hat.

**Aufwand Total** 6h

Akzeptanzkriterien

* «T\_WHITELIST» Tabelle enthält einige Jira Projekte sowie deren entsprechende Leis Konten.
* Gültiger Atlassian API-Token wurde für die IPA erstellt.
* Zeitbuchungen werden, sofern sie in der Datenbank nicht schon enthalten sind, von der Jira REST API importiert.
* Es wird überprüft ob alle nötigen Angaben in den importierten Zeitbuchungen plausibel vorhanden sind.
* Es wird überprüft ob der in der Zeitbuchung hinterlegten Mitarbeiter Rechte auf das hinterlegte Leis Konto hat.

Einfügen von Daten in Datenbank – S4

Anforderungen

-

Beschreibung

Als Benutzer möchte ich, dass, wenn ich meine Daten korrekt in Jira eingegeben habe, diese ohne dass ich weiter etwas machen muss, für mich im Leis verbucht werden.

Tasks

**Nummer Beschreibung Zeit Ablauf**

T4.1 Nach erfolgreichen Plausibilitätstests und potenziellem 3h 1

Error-Handling werden Leistungen in «T\_LEISTUNGSERFASSUNG»

mit dem passenden User sowie der Id des Jira Projektes verbucht.

**Aufwand Total** 3h

Akzeptanzkriterien

* Sofern Plausibilitäts- und Berechtigungstests erfolgreich abgeschlossen wurden, werden Daten in die Tabelle «T\_LEISTUNGSERFASSUNG» geschrieben.
* Die Id des Jira Projekts wird für künftige Datenerfassungen hinterlegt.
* Zeiten werden nicht gerundet oder angepasst.

Logging – S5

Anforderungen

A2, A6.1

Beschreibung

Als Entwickler möchte ich verhindern, dass ungültige Daten in die Datenbank geschrieben werden oder zum Absturz der Applikation führen, Fehler sollen geloggt werden und nicht plausible Daten sollen ebenfalls als Warnungen geloggt werden.

Tasks

**Nummer Beschreibung Zeit Ablauf**

T5.1 Zum Start der Applikation soll das bestehende Logfile geöffnet 1h 1

werden, oder falls keines vorhanden ist, ein neues erstellt werden.

T5.2 Werden unvollständig erfasste Zeiterfassungen von der Jira REST 1h 2

API importiert, soll ein Fehler im Log erfasst werden.

T5.3 Werden Daten, welche nicht plausibel sind von der Jira REST API 2h 3

importiert, soll eine Warnung im Log erfasst werden.

T5.4 Sollten sonstige Fehler innerhalb der Applikation vorkommen, 3h 4

sollen diese ebenfalls geloggt werden.

T5.5 Wird die Applikation beendet, soll dies im Log vermerkt sein. 1h 5

**Aufwand Total** 8h

Akzeptanzkriterien

* Ein Logfile soll vorhanden sein, welches mit Zeitstempeln den Start sowie das Ende eines Applikationsdurchlaufes beschreiben.
* Im Logfile werden Fehlermeldungen mit passenden Fehlercodes ebenfalls mit Zeitstempel erfasst.
* Im Logfile werden Warnungen mit passenden Codes ebenfalls mit Zeitstempel erfasst.

Development Modus – S6

Anforderungen

A8.2

Beschreibung

Als Entwickler möchte ich einen Development Modus, in welchem alle importierten Daten auf einen Development User gespeichert werden. Nötig ist dieser Development Modus da die Datenbank auf der Testumgebung anonymisiert ist, und somit Daten keinem spezifischen User zugewiesen werden können.

Tasks

**Nummer Beschreibung Zeit Ablauf**

T6.1 Wird die Applikation ab Start mit dem Flag «-d» oder «-dev» 1h 1

aufgerufen, wird sie im Development Modus gestartet.

T6.2 Im Development Modus werden alle Daten mit dem Development 1h 2

User gespeichert.

**Aufwand Total** 2h

Akzeptanzkriterien

* Applikation kann in einem Development Modus gestartet werden.
* Läuft die Applikation im Development Modus, werden alle Zeiterfassungen mit dem Development User gespeichert.

Testing – S7

Anforderungen

A1

Beschreibung

Als Entwickler ist es mir wichtig, dass meine Applikation fehlerfrei läuft. Ich möchte, dass ich wenn ich die Applikation weiter entwickle oder verändere mögliche Fehler sofort auffallen. Mein Ziel ist es nach Test Driven Development vorzugehen.

Tasks

**Nummer Beschreibung Zeit Ablauf**

T7.1 Mit Unit Tests soll eine möglichst hohe Code Coverage erreicht 8h 1

werden, wobei nur sinnvolle Methoden mit Unit Tests getestet

werden sollen.

T7.2 Wo Unit Tests nicht möglich oder sinnlos sind, sollen manuelle 4h 2

Tests durchgeführt werden. Für diese wird ein Testskript

geschrieben, sodass sie immer gleich durchgeführt werden

können.

T7.3 Am Ende der Entwicklungszeit sollen End-To-End Tests 2h 3

durchgeführt werden, womit die Funktionsweise der Applikation

als Ganzes getestet werden.

**Aufwand Total** 14h 4

Akzeptanzkriterien

* Methoden werden wo sinnvoll getestet, private Methoden werden nicht getestet oder gemockt. Die Datenbank wird ebenfalls gemockt.
* Ein Testskript soll klar beschreiben wie ein Test durchzuführen ist und soll nur Teile der Applikation testen welche durch die Unit Tests nicht schon abgedeckt sind.
* End-To-End Tests sind Teil der manuellen Tests und verfügen ebenfalls über ein Testskript.
* Alle Tests werden erfolgreich abgeschlossen.

Finalisierung Applikation – S8

Anforderungen

A5.1, A5.2

Beschreibung

Als Entwickler möchte ich mit einer sauber geschriebenen Applikation abschliessen. Dafür möchte ich sowohl während als auch am Ende der Entwicklung des Projektes Code Guidelines einhalten. Am Ende der Entwicklungszeit möchte ich das Projekt nochmals überarbeiten, um zu gewährleisten, dass diese Guidelines eingehalten werden.

Tasks

**Nummer Beschreibung Zeit Ablauf**

T8.1 Klassen-, Methoden-, und Variablennamen sollen sinnvoll und 1h 1

aussagekräftig sein.

T8.2 Wo nötig Code kommentieren (soll grösstenteils self-documenting 1h 2

sein, das heisst Methoden und Variablennamen sollen beschreiben

was sie machen und Kommentare sollten nicht nötig sein).

T8.3 Code aufräumen, unschönen Code refactorn. 2h

**Aufwand Total** 4h

Akzeptanzkriterien

* Klassen-, Methoden-, und Variablennamen sind selbstbeschreibend.
* Kommentare sind nur an wenigen Orten notwendig und verwendet.
* Code ist sauber und gut lesbar.

Finalisierung Dokumentation – S9

Anforderungen

-

Beschreibung

Als Kandidat will ich eine detaillierte und inhaltreiche Dokumentation erstellen. Wichtig ist mir hierbei dass ich so viele Bewertungskriterien wie möglich abdecken kann und parallel zur Applikationsentwicklung dokumentieren kann.

Tasks

**Nummer Beschreibung Zeit Ablauf**

T9.1 Arbeitsjournal führen. 10h 1

T9.2 Dokumentation vervollständigen. 7h 2

**Aufwand Total** 17h

Akzeptanzkriterien

* Das Arbeitsjournal wurde täglich geführt, überarbeitet und für jeden Tag final abgeschlossen.
* Die Dokumentation ist vollständig, ohne Lücken und wurde gegen Ende nochmals überarbeitet.
* Alle Bewertungskriterien konnten erfolgreich abgearbeitet werden.
  1. Risikoanalyse
     1. Motivationsverlust – R1

Beschreibung

Der Kandidat verliert die Motivation seriös an der Arbeit weiterzuarbeiten oder erleidet ein Burnout.

Massnahmen

Arbeiten, für welche sich der Kandidat nur schwer motivieren kann, werden in den ersten paar Tagen erledigt. Während der ersten paar Tage der Arbeit wird ausserdem länger gearbeitet, sodass Zeitdruck vermindert wird und der Kandidat sich auf entspanntere weitere Arbeitstage freuen kann. Weiterhin wurde der Zeitrahmen der IPA so gewählt, dass die zehn Arbeitstage in drei Wochen aufgeteilt sind, und zwischen den Arbeitsblöcken grössere Pausen vorkommen (Schule, Ostern). Im Zeitplan wurde ausserdem darauf geachtet, dass grosse Arbeitsblöcke (wie den Projektauftrag zu dokumentieren) auf zwei Tage aufgeteilt wurden, statt ein Tag für ein Thema aufzuwenden.

Risikohöhe

Eintrittswahrscheinlichkeit: Tief

Auswirkungen: Hoch

* + 1. Unfall – R2

Beschreibung

Der Kandidat erleidet eine Verletzung und kann aufgrund dieser für einen längeren Zeitraum nicht weiterarbeiten.

Massnahmen

Das Expertenteam und die Fachkräfte werden über den Unfall sofort aufgeklärt. Weitere Schritte werden abgeklärt.

Risikohöhe

Eintrittswahrscheinlichkeit: Tief

Auswirkungen: Hoch

* + 1. Krankheit – R3

Beschreibung

Der Kandidat kann aufgrund einer Krankheit für eine Zeit nicht am Projekt weiterarbeiten.

Massnahmen

Das Expertenteam und die Fachkräfte werden über den Unfall sofort aufgeklärt. Weitere Schritte werden abgeklärt.

Risikohöhe

Eintrittswahrscheinlichkeit: Tief

Auswirkungen: Mittel

* + 1. Technische Störung (Verlust Laptop, Datenverlust) – R4

Beschreibung

Der Laptop des Kandidaten geht kaputt oder die Daten darauf gehen verloren.

Massnahmen

Daten werden auf dem persönlichen Verzeichnis des Kandidaten auf dem Softec Server gespeichert und es werden alle Daten zusätzlich täglich auf einem Github Repository gespeichert. Bei Geräteverlust muss ein Ersatzgerät eingesetzt werden.

Risikohöhe

Eintrittswahrscheinlichkeit: Mittel

Auswirkungen: Tief

* + 1. Zeitplan verschätzt – R5

Beschreibung

Der Kandidat verschätzt sich im Zeitplan und ist im Verzug.

Massnahmen

Der Zeitplan ist relativ flexibel gestaltet und da die Mindestgrösse eines Eintrages eine Stunde beträgt wurden viele Schritte sehr grosszügig geschätzt.

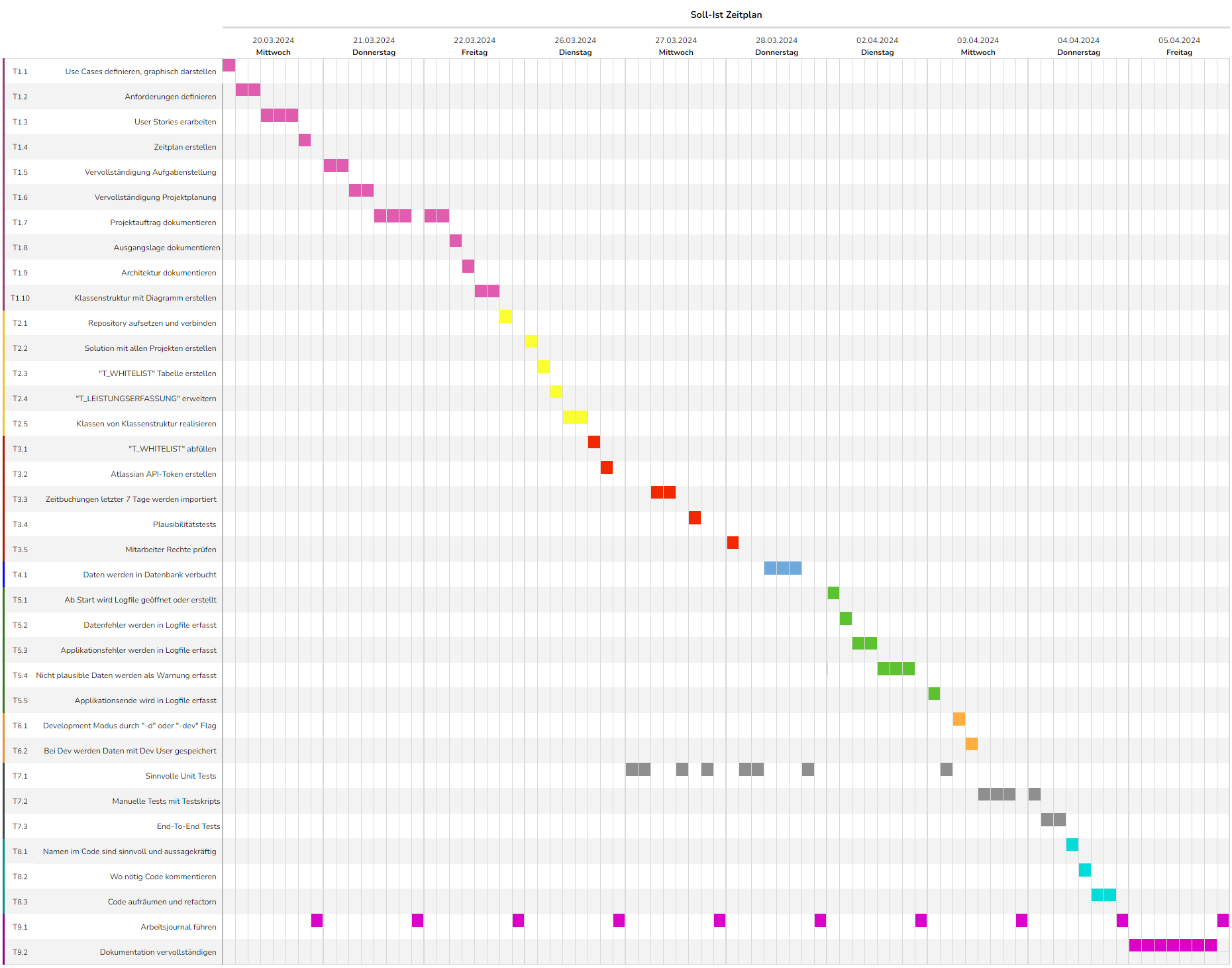
Risikohöhe

Eintrittswahrscheinlichkeit: Hoch

Auswirkungen: Mittel

* 1. Zeitplan

Der Zeitplan besteht aus den in den User Stories definierten Tasks. Die Tasks sind gemäss ihrer User Story gefärbt und werden in Reihenfolge abgearbeitet.



Ein Bild, das Text, Screenshot, Farbigkeit enthält.

Automatisch generierte Beschreibung

* + 1. Soll-Ist-Vergleich

[Insert Soll-Ist-Zeitplan]

Tag 1

[…]

Tag 2

[…]

Tag 3

[…]

Tag 4

[…]

Tag 5

[…]

Tag 6

[…]

Tag 7

[…]

Tag 8

[…]

Tag 9

[…]

Tag 10

[…]

1. Arbeitsjournal

Tag 1 – Projektplanung

|  |  |
| --- | --- |
| **Datum** | 20.03.2024 |
| **Projektstatus** | ⦁ Im Zeitplan |
| **Überzeiten** | 1.5h (geplant) |
| **Ungeplante Arbeiten** | Keine |
| **Tagesablauf** | Ich habe heute Morgen mit meiner IPA begonnen. Zunächst spielten meine Nerven etwas verrückt, doch nach einer Stunde Orientierung und Einarbeitung fand ich meinen Rhythmus ziemlich schnell und konnte innerhalb des Zeitplans weiterarbeiten.  In meiner nervöseren Anfangsphase setzte ich das Github Repository auf, wobei ich bei der Zeitplanung vergass dieses am Anfang aufzuschreiben und ich nun gemäss Zeitplan einen Task welcher erst auf Freitag geplant war heute schon erledigt habe.  Als nächstes definierte ich den Use Case meiner Applikation und setzte diesen mittels einer einfachen Grafik um. Danach definierte ich meine Anforderungen welche ich der detaillierten Aufgabenstellung sowie den individuellen Bewertungskriterien entnahm. Aus der detaillierten Aufgabenstellung konnte ich schliesslich die User Stories erarbeiten, und aus diesen wiederum meinen Zeitplan gestalten. Mit der Gestaltung des Zeitplans war ich längere Zeit unzufrieden, dies und die Gestaltung des Kanban Boards als Teil der Zeitplanung führten dazu, dass die Erstellung des Zeitplans länger dauerte als erwartet. |
| **Reflexion** | Insbesondere zu Beginn des Tages war ich sehr zerstreut und die vielen komplexen Arbeiten, die mich noch erwarteten überforderten mich etwas. Ich entschied mich die Arbeit langsam und bestimmt anzugehen, und nach etwas Einarbeitungszeit konnte ich mich entspannen und die Arbeit auch tatsächlich geniessen.  Ich habe es mir zu Ziel gesetzt zu Beginn der Arbeit etwas Überzeit zu generieren damit ich es gegen Ende der Arbeit etwas entspannter angehen kann und die trockeneren ersten Schritte (Dokumentation) schneller erledigt habe. Dieser Plan hat gut funktioniert, wobei ich mir unsicher war wie ich ihn im Zeitplan wiederspiegeln sollte, als solches sind alle Tage immer noch in acht Stunden unterteilt. |

**Aufgabenliste**

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Tag 2 – Projektplanung (Teil 2)

|  |  |  |
| --- | --- | --- |
| **Datum** | 21.03.2024 | |
| **Projektstatus** | | ⦁ Im Zeitplan |
| **Überzeiten** | | 1.5h (geplant) |
| **Ungeplante Arbeiten** | | Keine |
| **Tagesablauf** | | Auch heute bin ich etwas nervös in den Tag gestartet, jedoch um einiges entspannter als gestern. Als erstes habe ich die Aufgabenstellung dokumentiert, wobei ein Grossteil davon aus der detaillierten Aufgabenstellung von PkOrg entnommen wurde. Die Aufgabenstellung konnte schneller erledigt werden als erwartet, ich bin aktuell dem Zeitplan etwas voraus. |
| **Reflexion** | | [Kritische Würdigung Arbeit & Vorgehensweise] |

**Aufgabenliste**

Tag 3 –

|  |  |
| --- | --- |
| **Datum** | TT.MM.JJJJ |
| **Projektstatus** | ⦁ Im Zeitplan |
| **Überzeiten** | Keine |
| **Ungeplante Arbeiten** | Keine |
| **Tagesablauf** | [Erfolge / Misserfolge] |
| **Reflexion** | [Kritische Würdigung Arbeit & Vorgehensweise] |

**Aufgabenliste**

Tag 4 –

|  |  |
| --- | --- |
| **Datum** | TT.MM.JJJJ |
| **Projektstatus** | ⦁ Im Zeitplan |
| **Überzeiten** | Keine |
| **Ungeplante Arbeiten** | Keine |
| **Tagesablauf** | [Erfolge / Misserfolge] |
| **Reflexion** | [Kritische Würdigung Arbeit & Vorgehensweise] |

**Aufgabenliste**

Tag 5 –

|  |  |
| --- | --- |
| **Datum** | TT.MM.JJJJ |
| **Projektstatus** | ⦁ Im Zeitplan |
| **Überzeiten** | Keine |
| **Ungeplante Arbeiten** | Keine |
| **Tagesablauf** | [Erfolge / Misserfolge] |
| **Reflexion** | [Kritische Würdigung Arbeit & Vorgehensweise] |

**Aufgabenliste**

Tag 6 –

|  |  |
| --- | --- |
| **Datum** | TT.MM.JJJJ |
| **Projektstatus** | ⦁ Im Zeitplan |
| **Überzeiten** | Keine |
| **Ungeplante Arbeiten** | Keine |
| **Tagesablauf** | [Erfolge / Misserfolge] |
| **Reflexion** | [Kritische Würdigung Arbeit & Vorgehensweise] |

**Aufgabenliste**

Tag 7 –

|  |  |
| --- | --- |
| **Datum** | TT.MM.JJJJ |
| **Projektstatus** | ⦁ Im Zeitplan |
| **Überzeiten** | Keine |
| **Ungeplante Arbeiten** | Keine |
| **Tagesablauf** | [Erfolge / Misserfolge] |
| **Reflexion** | [Kritische Würdigung Arbeit & Vorgehensweise] |

**Aufgabenliste**

Tag 8 –

|  |  |
| --- | --- |
| **Datum** | TT.MM.JJJJ |
| **Projektstatus** | ⦁ Im Zeitplan |
| **Überzeiten** | Keine |
| **Ungeplante Arbeiten** | Keine |
| **Tagesablauf** | [Erfolge / Misserfolge] |
| **Reflexion** | [Kritische Würdigung Arbeit & Vorgehensweise] |

**Aufgabenliste**

Tag 9 –

|  |  |
| --- | --- |
| **Datum** | TT.MM.JJJJ |
| **Projektstatus** | ⦁ Im Zeitplan |
| **Überzeiten** | Keine |
| **Ungeplante Arbeiten** | Keine |
| **Tagesablauf** | [Erfolge / Misserfolge] |
| **Reflexion** | [Kritische Würdigung Arbeit & Vorgehensweise] |

**Aufgabenliste**

Tag 10 – Abgabe

|  |  |
| --- | --- |
| **Datum** | TT.MM.JJJJ |
| **Projektstatus** | ⦁ Im Zeitplan |
| **Überzeiten** | Keine |
| **Ungeplante Arbeiten** | Keine |
| **Tagesablauf** | [Erfolge / Misserfolge] |
| **Reflexion** | [Kritische Würdigung Arbeit & Vorgehensweise] |

**Aufgabenliste**

Teil 2 – Projekt

1. Kurzfassung

Ausgangslage

Umsetzung

Ergebnis

1. Ausgangslage
   1. Einleitung

[Beschreibe dein System (Schnittstellen, Datenbank, …)]

1. Funktionalitätsumsetzung
   1. Einleitung

[Dokumentiere deine Implementation + Relevante Themen / Methoden + Gedankenwege]

1. Testing
   1. Einleitung

[Beschreibe: Warum ist Testing wichtig]

* 1. Umfeld
     1. Testanlage
     2. Mittel und Methoden
  2. Testdrehbuch

[Was muss getestet werden]

* 1. Testprotokoll

[Durchführung der «Drehbuch-Tests» mit Resultat]

* + 1. Unit-Tests

[Automatisierte Tests einzelner Komponenten]

* + 1. Abnahmetests

[Manuelle Tests basierend auf Anforderungen]

* 1. Testreport
     1. Resultatübersicht

[Auflistung durchgeführter Tests mit Resultat]

* + 1. Auswertung

[Auswertung der Resultatübersicht]

1. Schlusswort
   1. Fazit

[Kritisch Würdigen, Schlüsse ziehen, Erfolge / Misserfolge, Gelerntes, Persönliche Bilanz]

* + 1. Projektplanung
       1. Zeitplanung
       2. Projektmethode
    2. Implementation
    3. Dokumentation
    4. Testing
  1. Nächste Schritte

1. Verzeichnisse

Akronyme

I

IPA *Individuelle Praktische Arbeit*

* 1. Glossar

|  |  |
| --- | --- |
| **<Wort>** | <Beschreibung> |
|  |  |

* 1. Literatur

1. <Beschreibung>
   * [<link>](http://doc.aldebaran.com/2-8/index_dev_guide.html)
   1. Abbildungsverzeichnis

3.1 Use Case Diagramm 11

1. Anhang
   1. Git-Log