Projekt: U-Car

Projektplan

# Dokumenteninformation

## Änderungsgeschichte

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Änderungsgrund | Autorin |
| 24.03.23 | 1.0 | Erstellung | Wenzl |
| 29.03.23 | 1.1 | Aktualisierung Arbeitspakete, Meilensteine, Arbeitsergebnisse | Wenzl |
| 11.04.23 | 1.2 | Ergänzung Unterstützende Prozesse | Wenzl |
| 12.04.23 | 1.3 | Hinzufügen von Verantwortlichkeiten | Wenzl |
| 12.04.23 | 1.4 | Aktualisierung Prototypfunktionen | Kronberg |
| 03.05.23 | 1.5 | Änderung Fehlerbehandlung | Wenzl |
| 20.05.23 | 1.6 | Aktualisierung Arbeitspakete | Wenzl |
| 21.05.23 | 1.7 | Dateipfade eingefügt | Wenzl |
| 25.05.23 | 1.8 | Überarbeitung Funktionsumfang | Wenzl |
| 16.06.23 | 1.9 | Aktualisierung Zeitschätzung | Wenzl |
| 05.07.23 | 1.10 | Ergänzung Pfade zu Projektergebnissen | Wenzl |

## Inhalt

[1 Dokumenteninformation 2](#_Toc135935518)

[1.1 Änderungsgeschichte 2](#_Toc135935519)

[1.2 Inhalt 3](#_Toc135935520)

[2 Einführung 5](#_Toc135935521)

[2.1 Definitionen und Abkürzungen 5](#_Toc135935522)

[2.2 Referenzen 5](#_Toc135935523)

[3 Projektübersicht 5](#_Toc135935524)

[3.1 Zweck und Ziel 5](#_Toc135935525)

[3.2 Annahmen und Einschränkungen 5](#_Toc135935526)

[3.3 Arbeitsergebnisse 5](#_Toc135935527)

[4 Projektorganisation 6](#_Toc135935528)

[4.1 Organisationsstruktur 6](#_Toc135935529)

[4.2 Externe Schnittstellen 6](#_Toc135935530)

[5 Management-Abläufe 6](#_Toc135935531)

[5.1 Ressourceneinschätzungen 6](#_Toc135935532)

[5.2 Projektplan 6](#_Toc135935533)

[5.2.1 Zeitplan (Interne Meilensteine) 6](#_Toc135935534)

[5.2.2 Besprechungen 7](#_Toc135935535)

[5.2.3 Abgaben 7](#_Toc135935536)

[6 Risiko Management 7](#_Toc135935537)

[7 Arbeitspakete 7](#_Toc135935538)

[7.1 Prototyp 9](#_Toc135935539)

[7.2 Abzüge im Funktionsumfang 9](#_Toc135935540)

[8 Infrastruktur 9](#_Toc135935541)

[9 Unterstützende Prozesse 10](#_Toc135935542)

[9.1 Konfigurationsmanagement 10](#_Toc135935543)

[9.2 Fehlermanagement 10](#_Toc135935544)

[9.3 Qualitätsmanagement 10](#_Toc135935545)

[9.4 Zeiterfassung 10](#_Toc135935546)

# Einführung

## Definitionen und Abkürzungen

GUI: Graphic User Interface

## Referenzen

# Projektübersicht

Im Rahmen dieses Projekts soll eine Anwendung für Car-Sharing namens „U-Car“ entwickelt werden. Fahrzeugeigentümer können ihre Fahrzeuge registrieren und für bestimmte Zeiträume zur Buchung freigeben. Andere Nutzer können eine Reservierung oder Buchung anfragen, die durch den Eigentümer bestätigt werden muss. Dadurch können sich die Nutzer für ihren gewünschten Zeitraum und gegen eine Gebühr ein Auto ausleihen, während die Fahrzeugeigentümer am Monatsende eine Gutschrift je nach Entleihdauer erhalten.

## Zweck und Ziel

Die Anwendung „U-Car“ soll es Privatpersonen ermöglichen, ihre wenig genutzten Autos zu vermieten oder ein Auto zu buchen, falls sie selbst keines besitzen.

## Annahmen und Einschränkungen

Die Speicherung der Nutzer- und Fahrzeugdaten erfordert eine Schnittstelle zu einer SQLite-Datenbank, ohne die unsere Anwendung nicht funktionsfähig ist.

## Arbeitsergebnisse

|  |  |
| --- | --- |
| Arbeitsergebnis | Beschreibung |
| U-Car | fertige, funktionsfähige Anwendung |
| Quellcode | Dokumentation des gesamten Quellcodes der Anwendung |
| Dokumentation | Projektplan, Anforderungsspezifikationen, Domänenmodell, Softwareentwurf mit Klassendiagrammen, Testplan und -dokumentation, Abschlusspräsentation |
| Website | Informationen zum Projekt |
| Zeiterfassung | Dokumentation der benötigten Arbeitszeit |

Die Tabelle wird im weiteren Projektverlauf aktualisiert.

Alle abzuliefernden Dokumente sind im Ordner main/Documentation auf dem GitLab Repository zu finden, der Quellcode für U-Car befindet sich in main/Code.

Die Website ist unter <https://ucar23.github.io/ucar/> erreichbar, dort kann ebenfalls ein Installer für die Anwendung heruntergeladen werden.

# Projektorganisation

## Organisationsstruktur

Unser Team besteht aus folgenden Mitgliedern:

|  |  |
| --- | --- |
| Mitglied | Verantwortlichkeit |
| Sandra Edigin | *Virtuelle Maschine* |
| Rosine Kenmo |  |
| Corinna Kronberg | *Datenbank* |
| Enya Wenzl | *GitLab, Projektautomatisierung* |

Die Tabelle wird im weiteren Projektverlauf aktualisiert.

## Externe Schnittstellen

* SQLite-Datenbank
* GitLab-Repository

# Management-Abläufe

## Ressourceneinschätzungen

Für die Fertigstellung des Projekts stehen 15 Wochen und 150 Arbeitsstunden pro Person zur Verfügung, was einen Gesamtaufwand von 600 Arbeitsstunden und eine wöchentliche Arbeitszeit von 10 Stunden für jeden ergibt.

## Projektplan

### Zeitplan (Interne Meilensteine)

|  |  |  |
| --- | --- | --- |
| Meilenstein | Frist | Beschreibung |
| I1 | 29.03.23 | * Entwurf von Use Cases, Domänenmodell, Projektplan |
| I2 | 12.04.23 | * Fertigstellung von I1 |
| I3 | 19.04.23 | * Entwurf Architektur, Klassendiagramme, Datenbank, Testplan |
| I4 | 03.05.23 | * Fertigstellung von I3 * Implementierung der grundlegenden Klassen und Architektur * Einbindung der Datenbank |
| I5 | 17.05.23 | * Code Review jedes Teammitglieds |
| I6 | 24.05.23 | * Fertigstellung von I5 * Fertigstellung des Prototyps |
| I7 | 14.06.23 | * Fertigstellung der Anwendung * Durchführung aller nötigen Tests |
| I8 | 23.06.23 | * Abschlusspräsentation, Website * Abgabe des Projekts |

Die Tabelle wird im weiteren Projektverlauf aktualisiert.

### Besprechungen

Jede Woche ist eine Besprechung geplant, die mittwochs um 8:15 Uhr an der OTH Regensburg stattfinden soll und an der alle Teammitglieder teilnehmen sollen. Dabei werden die Ergebnisse der letzten Woche und der Projektfortschritt besprochen sowie Aufgaben für die nächste Woche verteilt. Bei Bedarf können auch weitere Besprechungen zu anderen Zeiten oder mit einzelnen Mitgliedern abgehalten werden.

### Abgaben

|  |  |  |  |
| --- | --- | --- | --- |
| Meilenstein | Typ | Frist | Beschreibung |
| MS1 | Dokumentation | 21.04.23 | Projektplan, Anforderungen, Domänenmodell |
| MS2 | Dokumentation, Prototyp | 26.05.23 | Architektur, Entwurf, Prototyp |
| MS3 | Dokumentation, Anwendung | 09.07.23 | Schlussabgabe des gesamten Projekts |

# Risiko Management

|  |  |
| --- | --- |
| Risiko | Gegenmaßnahmen |
| Datenbankanbindung | frühzeitige Ausarbeitung der Schnittstelle |

# Arbeitspakete

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Arbeitspaket | Inhalt | Verantwortliche | Zeit (h) | Abhängigkeit |
| Meetings |  | alle | 60 |  |
| Projektplanung | Vision-Dokument, Projektplan, Zeiterfassung, Versionskontrolle | alle | 40 |  |
| Domänen-modellierung | Domänenmodell, SSD, Contracts | Edigin, Wenzl | 30 |  |
| Anforderungen | funktional und nicht-funktional, Use Cases | Kenmo, Kronberg | 20 |  |
| GUI Design | Architektur, UML | Wenzl | 15 |  |
| Domain Logic Design | Architektur, UML | Edigin, Kenmo | 30 |  |
| Datenbank Design | Architektur, UML | Kronberg | 15 |  |
| Testplanung | Testplan, -case, -suite | alle | 20 |  |
| Testdurchführung | Programmierung | alle | 20 |  |
| Testintegration | Anbindung von googletest | alle | 15 |  |
| Code Review |  | alle | 15 |  |
| Automatisierung (Projekt) |  | Wenzl | 15 |  |
| Datenbank |  | Kronberg | 30 | Design |
| Einbindung der Datenbank |  | Kronberg | 10 | Datenbank |
| GUI (Grundgerüst) | Implementierung Ansichten, Anbindung | Wenzl | 20 | Design |
| Klassenimplementierung (Grundgerüst) | Klassen, grundlegende Methoden | Edigin, Kenmo | 35 | Design |
| UC1 | Konto registrieren/anmelden | Edigin | 30 |  |
| UC2 | Fahrzeug registrieren | Kronberg | 25 |  |
| UC3 | Fahrzeug suchen | Wenzl | 15 |  |
| UC4 | Fahrzeug buchen | Wenzl | 20 |  |
| UC5 | Fahrzeug abholen/zurückbringen | Kenmo | 30 |  |
| UC6 | Buchung stornieren | Kronberg, Wenzl | 20 |  |
| Weitere Funktionen/Ansichten | Profilanzeige, Detailansichten | Edigin | 15 |  |
| Problembehebung |  | alle | 30 |  |
| Website |  | Edigin | 8 |  |
| Abschluss-dokumentation |  | alle | 8 |  |
|  |  | Gesamt: | 591 |  |
|  |  | Puffer: | 9 |  |

Die Tabelle wird im weiteren Projektverlauf aktualisiert.

Für die Wochen zwischen MS1 und MS2 (17.04. bis 21.05.) ist eine detailliertere Aufteilung als Exceltabelle im GitLab Repository bereitgestellt (main/Documentation/Arbeitspakete\_Projektstrukturplan).

## Prototyp

Für den zweiten Meilenstein ist die Abgabe eines Prototyps erforderlich, der die folgenden Funktionen umfassen soll:

* Grundlegende Implementierung und Umsetzung der Schichtenarchitektur
* Kommunikation zwischen den Schichten
* Vollständige Konfiguration der Datenbank
* GUI mit grundlegenden Funktionen für Benutzerinteraktion
* Implementierung aller im Klassendiagramm genannten Klassen
* Anmelden und registrieren der Nutzer
* Fahrzeug registrieren
* Fahrzeug suchen und buchen
* Fahrzeug abholen und zurückgeben mit Erstellung der Rechnung

Dabei haben wir uns für den Prototypen bewusst gegen diese Funktionen entschieden:

* GUI vollständig ausgebaut und nicht nur funktionale Anforderungen erfüllt, sondern auch auf nicht funktionale Anforderungen (wie Benutzerfreundlichkeit) erweitert
* Nutzer löschen
* Administrator Funktionen
* Buchung stornieren
* Fahrzeug löschen

## Abzüge im Funktionsumfang

In der Zeit zwischen MS1 und MS2 sind einige Probleme entstanden, die vor allem die Virtuelle Maschine und die Datenbank betreffen und den gesamten Entwicklungsprozess verzögern. Da der Funktionsumfang aus zeitlichen Gründen dementsprechend eingeschränkt werden muss, haben wir uns entschieden, die folgenden Funktionen vorerst zu streichen:

* Bearbeiten und Löschen von Konto und Fahrzeug durch die Nutzer
* Administrator-Funktionen

# Infrastruktur

|  |  |
| --- | --- |
| Einsatzzweck | Infrastruktur |
| *Entwicklungsumgebung* | QtCreator, Visual Studio Code, CLion |
| *Datenbank* | SQLite |
| *Versionskontrolle, Fehlerbehandlung* | GitLab |
| *Test* | googletest |
| *Projektautomatisierung* | CMake |
| *Dokumentation* | MS Office, www.diagrams.net |
| *weitere* | Virtuelle Maschine |

Die Tabelle wird im weiteren Projektverlauf aktualisiert.

# Unterstützende Prozesse

## Konfigurationsmanagement

Für das Konfigurationsmanagement verwenden wir das bereitgestellte GitLab-Repository, um Quellcode und Dokumente an einer Stelle zu sammeln und auszutauschen. Dokumente werden in den Ordner Documentation auf dem Branch main hochgeladen und die durchgeführten Änderungen in die Commit Message geschrieben. Was Teil eines anderen Dokuments ist oder später in eines eingefügt wird, wird zum entsprechenden Zeitpunkt aus dem Ordner gelöscht. Unser Quellcode befindet sich im Branch develop, in dem für jede neue Funktion einzelne feature Branches erstellt werden, in die Code in Bearbeitung hochgeladen wird. Diese werden mit develop gemerged, wenn sie funktionsfähig sind und eine Code Review stattgefunden hat.

## Fehlermanagement

Für die Fehlerbehandlung verwenden wir ebenfalls GitLab, damit gefundene Fehler direkt am geschriebenen Code dokumentiert und bewertet werden können. Der Einsatz von Issues ermöglicht es, Fehler einfach nachzuverfolgen und schnell zu beheben. Eine Issue soll dann erstellt werden, wenn eine Fehlfunktion gefunden wurde, deren Lösung nicht offensichtlich sofort behoben werden kann.

## Qualitätsmanagement

Geschriebener Code wird mindestens an den Stellen mit Kommentaren versehen, an denen seine Funktion nicht selbsterklärend ist. Durch regelmäßige Code Reviews wird die Qualität unseres Codes sichergestellt, außerdem werden die Funktionsfähigkeit der einzelnen Komponenten und ihre Kompatibilität ausreichend getestet.

## Zeiterfassung

Die Zeiterfassung erfolgt mittels einer gemeinsamen Exceltabelle, in der die Arbeitszeit jedes Teammitglieds für die einzelnen Arbeitspakete festgehalten wird.