Map-Diary

Fachbereich: Applikationsentwicklung / App-Entwicklung

Projektleitung: Andy Zhang

Co-Leitung: Noel Knubel

Co-Leitung: Kajitha Kenkatharan

Betreuer: Yves Peissard  
Datum: 17.06.2015

Firma: ICT Berufsbildungscenter AG

# Abstract (Kurzbeschreibung)

Unser Projekt Map Diary (Zusammensetzung von der Nutzung von Google-Maps und Funktionen von einem Tagebuch) ermöglicht die Dokumentation der Wege, die man gegangen ist. Der Benutzer kann dafür ein Passwort festlegen und kann dann mit einer Google-Maps Karte Tracks hinzufügen. Ein Track sind mehre Way-Points, die einen ganzen Weg markieren. Wenn der Benutzer einen Track eingeben möchte, muss er mit einem Start-Button die Aufnahme beginnen. Diese startet dann an dem aktuellen Standpunkt des Benutzers. Das Programm nimmt dann solange auf, bis der Benutzer beim erneuten Öffnen Stop drückt. Hier kann er einen Namen eingeben und der Track wird gespeichert. Mit einem Optionsmenü (in der Karten\_Activity oder Option\_Activity) kann der Benutzer alle Tracks auflisten. Wenn er nun den gewünschten Track auswählt, springt er automatisch zu der Position auf der Karte.

Inhaltsverzeichnis

[1 Abstract (Kurzbeschreibung) 1](#_Toc422228502)

[2 Rahmenbedingungen (Aufgabenstellung) 5](#_Toc422228503)

[2.1 Einführung 5](#_Toc422228504)

[2.2 Funktionale Anforderungen 5](#_Toc422228505)

[2.2.1 Umfang 5](#_Toc422228506)

[2.2.2 GUI 5](#_Toc422228507)

[2.2.3 LifeCycle 5](#_Toc422228508)

[2.3 Nichtfunktionale Anforderungen 5](#_Toc422228509)

[2.3.1 Technik 5](#_Toc422228510)

[2.3.2 Fremder Code 5](#_Toc422228511)

[2.3.3 Wartbarkeit 5](#_Toc422228512)

[2.4 Lieferumfang 6](#_Toc422228513)

[2.4.1 APP 6](#_Toc422228514)

[2.4.2 Dokumentation 6](#_Toc422228515)

[2.4.3 Präsentation 6](#_Toc422228516)

[3 Beurteilungskriterien (gemäss Excel „Mobile APP Bewertung“) 6](#_Toc422228517)

[4 Projektorganisation 7](#_Toc422228518)

[4.1 Organigramm 7](#_Toc422228519)

[5 Pflichtenheft 8](#_Toc422228520)

[5.1 Konkurrenzanalyse 8](#_Toc422228521)

[5.2 Anforderungsanalyse 8](#_Toc422228522)

[5.2.1 Ziele 8](#_Toc422228523)

[5.2.2 Funktionale Anforderungen 9](#_Toc422228524)

[5.2.3 Nichtfunktionale Anforderungen 9](#_Toc422228525)

[6 Planung 11](#_Toc422228526)

[6.1 Arbeitspaket A – Titel/Bezeichnung z.B. Projektstart 12](#_Toc422228527)

[6.2 Arbeitspaket B 12](#_Toc422228528)

[6.3 Arbeitspaket C 12](#_Toc422228529)

[6.4 Zeitplanung 13](#_Toc422228530)

[6.4.1 GANTT 13](#_Toc422228531)

[7 Technische Dokumentation (Produktdokumentation) 14](#_Toc422228532)

[7.1 Tests 14](#_Toc422228533)

[7.2 Testauswertung 15](#_Toc422228534)

[8 Benutzerhandbuch 15](#_Toc422228535)

[8.1 Anmelden, Login, Logout oder z.B. Inbetriebnahme bei einem Gerät 15](#_Toc422228536)

[8.2 Funktionen 15](#_Toc422228537)

[9 Weiterentwicklungsmöglichkeiten 16](#_Toc422228538)

[10 Fazit 16](#_Toc422228539)

[11 Abkürzungen / Begriffserklärung 16](#_Toc422228540)

[12 Quellen 16](#_Toc422228541)

[A Anhang 17](#_Toc422228542)

[A.1 Schema 17](#_Toc422228543)

[A.2 Fertigungsdokumente 17](#_Toc422228544)

[A.3 Beschreibung der Ordnerstruktur 17](#_Toc422228545)

[A.4 Testprotokolle 17](#_Toc422228546)

[A.5 JavaDoc 17](#_Toc422228547)

[A.6 Mindmaps 17](#_Toc422228548)

[A.7 … 17](#_Toc422228549)

[A.8 Ressourcen 18](#_Toc422228550)

[A.8.1 Bilder 18](#_Toc422228551)

[A.9 IPERKA 19](#_Toc422228552)

[A.10 Usw. 20](#_Toc422228553)

Abbildungsverzeichnis

[Abbildung 1: Organigramm der Projektgruppe 6](#_Toc368916081)

[Abbildung 2: Bewertung der Risiken 10](#_Toc368916082)

[Abbildung 3: IPERKA 23](#_Toc368916083)

Tabellenverzeichnis

[Tabelle 1: Test Case 17](#_Toc368916084)

[Tabelle 2: Testresultate 18](#_Toc368916085)

# Rahmenbedingungen (Aufgabenstellung)

## Einführung

Für das Modul Mobile APP Entwicklung ist eine native Android **APP mit mindestens 3 Activities** zu erstellen. Während der gesamten Dauer des Moduls wird im Rahmen des Projektmanagements diese Dokumentation vervollständigt. Die **Bewertungskriterien können im Beurteilungs-Excel nachgelesen werden**. Wichtig ist, dass am Ende des Moduls eine voll funktionsfähige APP präsentiert werden kann, welche die folgenden Anforderungen erfüllt:

## Funktionale Anforderungen

### Umfang

Die APP muss aus **mindestens 3 Activities** bestehen und über **eine komplexere Komponente** wie z.B. Sensoren, DB oder MAPs verfügen.

### GUI

Mindestens **eine Activity** der APP muss **in der üblichen XML-Syntax** geschrieben werden. Das ist vor allem wichtig, wenn ein Game erstellt wird, in dem andere GUI-Technologien verwendet werden.

### LifeCycle

Android verfügt über einen LifeCycle der zu gewissen Lebensphasen der APP, respektive der Activity, verschiedene Funktionen ansteuert. Während der Entwicklung muss daher darauf geachtet werden, dass der **LifeCycle korrekt verwendet** wird.

### User Stories

1. Der Benutzer verlangt, dass er den PIN-Code eingeben kann, damit er sein Tagebuch öffnen kann.
2. Der Benutzer möchte seine laufende Route aufnehmen, wenn er auf Aufnahme starten klickt und will seine Route stoppen und speichern, wenn er auf stopp klickt.
3. Der Benutzer will eine Übersicht haben von seine gespeicherte Route haben und wenn er eine seiner Route klickt, will er auf der Karte die Route sehen.

## Nichtfunktionale Anforderungen

### Technik

Das GUI ist nach den Best Practice und gängigen ergonomischen Standards zu gestalten. Dabei muss ein klarer Ablauf den Benutzer durch die verschiedenen Funktionen der APP führen. Als wichtiges Kriterium muss die APP so gestaltet werden, dass die APP **auf allen Zielgeräten richtig dargestellt wird**. Im Minimum muss die APP auf Mobiltelefonen gut dargestellt werden.

### Fremder Code

Im Internet gibt es relativ viele Beispiele für APPs. Es ist nicht verboten sich an diesen zu orientieren. Der eigene Code darf aber **keine Teile von fremden APPs enthalten** (Thematik Urheberrecht). Die einzigen Ausnahmen sind Methodenaufrufe zu APIs oder Frameworks, die zum Erstellen der APP verwendet werden.

### Wartbarkeit

Damit die fertige APP auch später weiterentwickelt werden kann, sollte der Code möglichst **Wartungsfreundlich** gestaltet werden. Dazu muss der **Code mit Kommentar beschrieben** werden. Folgende Code-Elemente sind zu beschreiben:

GUI: Falls die Verständlichkeit nicht von selbst gegeben ist, sind bei den entsprechenden Code-Zeilen (auch XML) Kommentare zu setzten.

Klassen: Die Funktion jeder Klasse muss mit 1-2 Sätzen beschrieben werden.

Methoden: Der Methodenname muss selbsterklärend gewählt werden. Fall innerhalb der Methode die Funktion der Zeilen nicht selbsterklärend ist, muss entweder die Methode entsprechend mit Kommentaren versehen werden, oder falls es nur wenige Zeilen betrifft, müssen diese beschreiben werden.

## Lieferumfang

### APP

Es muss eine voll funktionsfähige APP abgegeben werden, die aus mindestens 3 Activities besteht. Alle Formulare müssen funktionieren und die Inhalte (Texte, Bilder, usw.) müssen korrekt dargestellt werden.

### Dokumentation

Während der Umsetzung muss eine Dokumentation erstellt werden, in der folgendes ersichtlich ist:

* Ausgangslage
* Ziele
* Testverfahren
* Technische Dokumentation
* Ausblick

### Präsentation

Abschluss des Projekts bildet eine **Verkaufspräsentation von max. 10 Minuten**. Diese beinhaltet die Vorstellung der APP. Nebst dem Verwendungszweck ist auch die Art der Präsentation wichtig. Diese muss **gut vorbereitet** werden, damit potentielle Käufer die APP auch haben möchten.

# Beurteilungskriterien (gemäss Excel „Mobile APP Bewertung“)

Die Beurteilungskriterien sind im Excel „Mobile APP Bewertung“ im Detail ersichtlich. Folgende Teile werden beurteilt:

20% Mock UPs

30% Fachgespräch über eine umfangreichere Komponente

10% Verkaufspräsentation

40% Produktbewertung (Code und Dokumentation)

# Projektorganisation

In diesem Kapitel werden die Zuständigkeiten im Projekt definiert.

Die Aufteilung bezieht sich auf die Verantwortlichkeiten und nicht auf die Arbeiten selber! Jeder muss in allen Teilbereichen (Schema, Software, Gehäuse usw.) mitarbeiten.

## Organigramm



Abbildung 1: Organigramm der Projektgruppe

Organigramm anpassen (z.B. für 1-3 Teilprojektleiter). Bei Einzelprojekten gibt es 1 Person die Projekt- und Teilprojektleiter ist.

# Pflichtenheft

## Informationsbeschaffung/Konkurrenzanalyse

Wir haben 3 Apps nach deren Funktionen und Vor-Nachteile geprüft. Folgende Dinge haben sich ergeben:

### Konkurrenz-App: GPS Navigation Sygic

Beschreibung:

Eine Offline Navi mit Tom-Tom Maps. Kostenlose GPS-Navigation, Routenplanung und kostenlose Maps-Updates

Pegi: 3

Downloads: 10 Millionen

Bewertung: 4.3

Kategorie: Reisen & Lokales

Stärken:

-Design ist nicht schlecht, angenehm zu bedienen. Besitzt ein Willkommensfenster

-Registrierung ist freiwillig und kann übersprungen werden.

Schwächen:

-App braucht sehr viel Speicher, da jede Landkarte manuell noch mal runtergeladen werden muss.

-App braucht sehr lange fürs Herunterladen.

-App erhängt sich nach dem Herunterladen und funktioniert nicht mehr.

-Laut Kommentare von anderen Anwendern sehr viele Fehlermeldung nach dem manuellen Herunterladen einer Landkarte.

### Konkurrenz-App: HERE, Nokia Apps LLC

Beschreibung:

Ein Map-Tool von Nokia. Kostenlose Karten, Registrierung, Routenplanung, offline Karten, Updates.

Pegi: 3

Downloads: 1 Million

Bewertung: 4.4

Kategorie: Reisen & Lokales

Stärken:

-Ansprechendes, schmales, eckiges Design.

-Offline-Modus ermöglicht Verwendung ohne Internet.

-Registrierung wurde/wird unaufdringlich aufgefordert.

-Offene Hantierung mit Nutzer- und Datenschutzbedingungen.

-Möglichkeit Feedback zu geben.

Schwächen:

-Für viele Funktionen wie Karten Offline nutzen setzen und teilen von Way-Points braucht man ein Account.

-Das Karten-Material ruckelt bei schnelleren Bewegungen teils stark.

-Teils hoher Daten- und Arbeitsspeicherverbrauch im Hintergrund (nach kurzem Testen 50mb Datenverbrauch, 150mb RAM).

Der App Meine Tracks hat ähnliche Funktionen wie unser App.   
Die Grösse ist je nach Gerät Variiert. Angeboten von Google Inc. . Installationen 10.000.000 bis 50.000.000. Bewertung Fünf Sterne 109.315 und ein Stern 10.783. Aktualisiert seit 5. Mai 2015.

Vorteile:  
 - 3D Aussicht.  
 - Dies zeichnet den Route des Benutzers, Geschwindigkeit, Distanz und Höhe.  
 - Die Daten werden live gezeigt.  
 - Sprachmeldungen werden über Benutzers Fortschritt anhören lassen.  
 - Die gemachte Tracks können veröffentlicht werden.

Nachteile:  
 - Der User kommt am Anfang nicht draus, was der App genau von der User verlangt.  
 - Den User wurde auch nicht erklärt, wie man eine Route zeichnen kann  
 - Beim Route zeichnen, wenn man klickt passiert nichts.  
 - Bevor man die Route zeichnen kann muss man in App selber noch einmal den GPS starten,  
 aber dies wurde am Anfang nicht erwähnt.  
 - Aber auch wenn man den GPS von App einschaltet man kommt nicht draus, weil da nichts   
 zeichnet.

### Das sollten wir besser machen:

* Auf unser App sollte der User am Anfang nicht verwirrt sein. Er sollte gute Übersicht und Einleitung bekommen. Unser App soll nicht viel Speicher brauchen und länger dauern zu Herunterladen. Das Programm soll nötiges Feedback geben, damit der User mit bekommt was im Hintergrund passiert.

## Anforderungsanalyse

Das Projekt sollte zum Schluss folgendes Beinhalten:  
-Die Dokumentation (Dazu gehören Anhänge)

### Ziele

Welche Ziele muss das Produkt am Schluss erfüllen?

Oft erhält man nicht ein ausformuliertes Lastenheft, oder es gibt Missverständnisse bei der Vorstellung, was das Projekt am Schluss beinhalten soll. Darum ist es wichtig in dieser Phase eng mit dem Auftraggber zusammen zu arbeiten. Bei der Zielformulierung gilt es folgendes zu beachten:

|  |  |
| --- | --- |
|  | * ***Erreichbarkeit*** |
|  | Die formulierten Ziele müssen erreicht werden können. Dies sollte gemeinsam mit dem Auftraggeber abgeklärt werden. |
|  | * ***Vollständigkeit*** |
|  | Alle zu erreichenden Ziele müssen niedergelegt werden. Es dürfen später im Pflichtenheft keine neuen Ziele mehr definiert werden, die nicht Teil des Auftrages sind. |
|  | * ***Verständlichkeit*** |
|  | Eine klare und eindeutige Zielformulierung ist unabdingbar für die zielgerichtete Erstellung eines Pflichtenheftes. Nur wenn die zu erreichenden Ziele verständlich sind, können alle die mitarbeiten helfen diese zu erreichen. |
|  | * ***Messbarkeit*** |
|  | Ziele dürfen nicht mit Begriffen wie z. B. "besser" oder "schneller" usw. beschrieben werden. In der Zielsetzung muss klar ausgedrückt werden, was z. B. mit "besser" gemeint ist. Die Verringerung der Fehlerquote in der Auftragserfassung um 10% ist ein klar messbares Ziel. Die Halbierung der Durchlaufzeiten im Lager kann ebenfalls gemessen werden. Erst mit der Messbarkeit eines definierten Ziels ist es möglich, den Erfüllungsgrad der im Pflichtenheft definierten Anforderungen zu überprüfen. |
|  | * ***Konsistenz*** |
|  | Die definierten Ziele müssen widerspruchsfrei (konsistent) beschrieben werden. Erst mit sich ergänzenden Zielen ist es möglich, effektiv die Erstellung eines Pflichtenheftes anzugehen. |
|  | * ***Lösungsneutralität*** |
|  | Bei den zu erreichenden Zielen muss das "Was", nicht das "Wie" beschrieben werden. |

#### Ziele

Abgabe Pflichtenheft!

An dieser Stelle muss das Pflichtenheft dem Auftraggeber zur Annahme abgegeben werden. Mach mit ihm einen Besprechungstermin ab und schicke ihm vorgängig das Dokument per Mail.

Nun wird vom Auftraggeber entschieden ob das Projekt so durchgeführt werden kann. In diesem Schritt kann es auch Anpassungen geben.

**Warnung:**

**Wer hier ohne Rücksprache mit dem Auftraggeber weiterfährt, gefährdet das Projekt.** Der Auftraggeber könnte die Annahme des Projekts am Schluss verweigern.

**In unserem Fall würde das einer Note 1 entsprechen.**

### Funktionale Anforderungen

|  |  |  |
| --- | --- | --- |
| ID | Beschreibung | Art |
| FA-01 | Der Benutzer kann sich mit einem PIN-Code anmelden. | Muss |
| FA-02 | Der Benutzer kann eine Aufnahme starten, die die aktuell gelaufene Route aufzeichnet und mit bestätigen von Stopp speichert. | Muss |
| FA-03 | Der Benutzer kann mit einem Suchfeld nach einem Track suchen | Soll |
| FA-04 | Der Benutzer kann an einem beliebigen Punkt oder beim aktuellen Standort einen Way-Point setzen können. | Kann |
| FA-05 | Der Benutzer kann Tracks editieren oder wieder löschen. | Kann |

### 

### Nichtfunktionale Anforderungen

|  |  |  |
| --- | --- | --- |
| ID | Beschreibung | Art |
| NFA-01 | Der Benutzer kann eine Auflistung in Form einer Liste öffnen, die alle Tracks zeigt. | Muss |
| NFA-02 | Bei Auswahl eines Tracks wird er zu dieser Position auf der Karte weitergeleitet. | Muss |
| NFA-03 | Wenn man auf einen Track tippt, wird der jeweilige Name oder Info angezeigt. | Soll |
| NFA-04 | Datum und Uhrzeit sollen automatisch eingefügt werden. | Kann |
| NFA-05 | Der Benutzer kann neben dem Namen auch weitere Informationen wie Beschreibung, Datum, Uhrzeit, Bilder und Ort bestimmen. | Kann |

# Planung

In diesem Schritt werden die Arbeitspakete definiert. Normalerweise wird ein Arbeitspaket nur von einer Person abgearbeitet. Jedoch macht das während der Basisausbildung nicht überall Sinn. Vor allem wenn verlangt wird, dass alle Projektarbeiter alles kennen lernen.

Folgende Kriterien muss ein Arbeitspaket erfüllen:

1. Es muss von einem (nicht mehreren) Projektmitarbeiter erledigt werden können.
2. Der Mitarbeiter kann (muss nicht) das Paket in Teilschritte unterteilen => IPERKA.  
   (siehe Kapitel 11.6 IPERKA)

Das bedeutet, der Umfang der Arbeit die das Paket hat, muss sinnvoll gewählt werden. Zum Beispiel könnte ein Paket sein, die Dokumentation erstellen. Das gibt aber viel zu tun für eine Person, darum kann das Pakete in kleinere Pakete unterteilt werden. Zum Beispiel unter anderem in Risikoanalyse und Arbeitspakete definieren.

Mögliche Arbeitspakete sind:

* Windows Server installieren, Verzeichnisdienst aufsetzen.
* HTML Grundgerüst erstellen
* Firewall konfigurieren
* Benutzerdokumentation
* Testfälle erstellen
* Testfälle durchführen
* Testfälle dokumentieren

Die Arbeitspakete werden im folgenden Kapitel zur Erstellung der Zeitplanung (GANTT) verwendet.

## Arbeitspaket A – Informationsbeschaffung

Beschreibt alle Arbeitspakete nach diesem Muster. Kopiert die Tabelle für jedes Arbeitspaket und macht euch Gedanken was in diesem Arbeitsschritt passieren sollt. Das Arbeitspakete wird danach von jemandem aus eurer Gruppe nach der Arbeitsmethode IPERKA bearbeitet (siehe Anhang).

|  |  |
| --- | --- |
|  | Beschreibung |
| Anfangsvoraussetzungen | Bevor wir mit diesem Paket beginnen können, müssen wir uns auf ein genaues Projekt einigen. In unserem Fall wäre das die App Map-Diary. |
| Ziele: (Ergebnisbeschreibung) | -Wir wissen mehr über unsere Konkurrenz und können so gewisse Standards festlegen, die unsere App besser machen kann. -Wir wissen mehr über unsere benötigten Ressourcen (Wissen wie Bücher, nötige API’s usw.) und können in Zukunft einfach darauf zugreifen. |
| Abhängigkeiten: | Dies ist das erste Paket. Daher gibt es noch keine Abhängigkeiten. |
| Aufwand in 2h Schritten: | Ca. 2-4 Stunden |
| Benötigte Mittel: | Es sind keine speziellen Hilfsmittel nötig. |
| Meilenstein [JA/NEIN] | Dieser Schritt ist kein Meilenstein, da nur Vorbereitungen getroffen wurden. Falls es ein Meilenstein ist, muss er auch im GANTT Diagramm eingetragen werden. Siehe Zeile Meilensteine. |

## Arbeitspaket B – App-Struktur und MockUps erstellen

|  |  |
| --- | --- |
|  | Beschreibung |
| Anfangsvoraussetzungen | Alle nötigen Informationen sollten gesammelt worden sein. Ausserdem sollten alle wichtigen Funktionen bekannt sein. |
| Ziele: (Ergebnisbeschreibung) | -Wir erstellen mehrere App-Strukturen und MockUps, das unseren zukünftigen Aufbau und Design unserer App zeigt. -Wir einigen uns auf ein MockUp, damit wir eine gute Vorlage für unsere App besitzen. |
| Abhängigkeiten: | Abhängig von Paket A. |
| Aufwand in 2h Schritten: | Ca. 4-6 Stunden |
| Benötigte Mittel: | Eventuell für MockUps und App-Struktur Papier und Stift. |
| Meilenstein [JA/NEIN] | Dieser Schritt ist ein Meilenstein. Wir haben nun ein detailliertes MockUp, dass unser zukünftiges Design zeigt und eine App-Struktur, die den genauen Zusammenhang zwischen den Activitys darstellt.  Falls es ein Meilenstein ist, muss er auch im GANTT Diagramm eingetragen werden. Siehe Zeile Meilensteine. |

## Arbeitspaket C – Nachführung der Doku

|  |  |
| --- | --- |
|  | Beschreibung |
| Anfangsvoraussetzungen | Alle Vorbereitungen wurden getroffen und alle offenen Fragen beantwortet. |
| Ziele: (Ergebnisbeschreibung) | -Wir führen die Dokumentation nach. Damit sind wichtige Schritte wie definieren von Arbeitspakete, bearbeiten des Pflichtenheftes etc. gemeint. |
| Abhängigkeiten: | Arbeitspaket A und B sollten abgeschlossen sein. |
| Aufwand in 2h Schritten: | Ca. 2-4 Stunden |
| Benötigte Mittel: | Es sind keine speziellen Hilfsmittel nötig. |
| Meilenstein [JA/NEIN] | Dieser Schritt ist kein Meilenstein, da die Doku noch nicht abgeschlossen ist. Falls es ein Meilenstein ist, muss er auch im GANTT Diagramm eingetragen werden. Siehe Zeile Meilensteine. |

## Arbeitspaket D – Aufbau des GUIs

|  |  |
| --- | --- |
|  | Beschreibung |
| Anfangsvoraussetzungen | Alle Vorbereitungen wurden getroffen und alle offenen Fragen beantwortet. |
| Ziele: (Ergebnisbeschreibung) | -Wir bauen das GUI in Android Studio nach. Dafür verwenden wir hauptsächlich XML Files und den Values Ordner. |
| Abhängigkeiten: | Arbeitspaket A und B sollten abgeschlossen sein. |
| Aufwand in 2h Schritten: | Ca. 4-8 Stunden |
| Benötigte Mittel: | Hier wird zum ersten Mal Android Studio eingesetzt. |
| Meilenstein [JA/NEIN] | Dieser Schritt ist ein Meilenstein, da wir nun endlich ein sichtbares Resultat haben und auf dem Emulator bzw. Handy ausführen können. Falls es ein Meilenstein ist, muss er auch im GANTT Diagramm eingetragen werden. Siehe Zeile Meilensteine. |

## Arbeitspaket E – Einbauen der Logik

|  |  |
| --- | --- |
|  | Beschreibung |
| Anfangsvoraussetzungen | Das GUI ist abgeschlossen und es können jetzt die Methoden eingebaut werden. |
| Ziele: (Ergebnisbeschreibung) | -Wir bauen die Logik unseres Programms auf. Wir sorgen dafür, dass die Google-Maps API funktioniert. Ausserdem sollte die Navigation eingabaut werden, damit das Programm bedienbar ist. -Animationen wie die Side-Bar sollen funktionieren und die Parameter sollen so vorbereitet sein, dass die Track aufzeichnen Funktionen einfach eingebaut werden kann. |
| Abhängigkeiten: | Arbeitspaket A, B und C sollten abgeschlossen sein. |
| Aufwand in 2h Schritten: | Ca. 24 Stunden |
| Benötigte Mittel: | Es sind keine speziellen Hilfsmittel nötig. |
| Meilenstein [JA/NEIN] | Dieser Schritt ist kein Meilenstein, da dies nur eine Massenabgrenzung ist, damit wir nicht zu viel in einem Arbeitspaket haben.  Falls es ein Meilenstein ist, muss er auch im GANTT Diagramm eingetragen werden. Siehe Zeile Meilensteine. |

## Arbeitspaket F – Einbau der Login-Funktion

|  |  |
| --- | --- |
|  | Beschreibung |
| Anfangsvoraussetzungen | Das GUI wurde grundsätzlich eingebaut. |
| Ziele: (Ergebnisbeschreibung) | -Wir bauen die Login-Funktion ein. Dafür sollte ein PIN verwendet werden, der der Benutzer beim ersten Starten bestimmen kann. Der PIN kann jederzeit geändert werden und falls es vergessen wurde mit einen speziellen Verfahren (noch nicht ausgemacht) zurückgesetzt. Wenn der Benutzer die Karten\_Activity verlässt oder die App schliesst wird er abgemeldet. |
| Abhängigkeiten: | Arbeitspaket A, B, und C sollten abgeschlossen sein. |
| Aufwand in 2h Schritten: | Ca. 16 Stunden |
| Benötigte Mittel: | Es sind keine speziellen Hilfsmittel nötig. |
| Meilenstein [JA/NEIN] | Dieser Schritt ist kein Meilenstein, da dies nur eine Massenabgrenzung ist, damit wir nicht zu viel in einem Arbeitspaket haben.  Falls es ein Meilenstein ist, muss er auch im GANTT Diagramm eingetragen werden. Siehe Zeile Meilensteine. |

## Arbeitspaket G – Einbauen der Trackaufzeichnung

|  |  |
| --- | --- |
|  | Beschreibung |
| Anfangsvoraussetzungen | Das Programm ist bis auf den Start und Stop Knopf voll funktionsfähig. |
| Ziele: (Ergebnisbeschreibung) | -Wir bauen unsere Hauptfunktion ein. Und zwar das aufzeichnen von Tracks. -Wir informieren uns genauer über die Google-API und wie wir diese Funktion einbauen können. |
| Abhängigkeiten: | Arbeitspaket A, B, C und D sollten abgeschlossen sein. |
| Aufwand in 2h Schritten: | Ca. 24 Stunden |
| Benötigte Mittel: | Es sind keine speziellen Hilfsmittel nötig. |
| Meilenstein [JA/NEIN] | Dieser Schritt ist ein Meilenstein, da wir nun alle Funktionen in unsere App eingebaut haben.  Falls es ein Meilenstein ist, muss er auch im GANTT Diagramm eingetragen werden. Siehe Zeile Meilensteine. |

## Arbeitspaket G – Das Programm testen

|  |  |
| --- | --- |
|  | Beschreibung |
| Anfangsvoraussetzungen | Alle Funktionen des Programms sind vollständig fertiggestellt (clean-Code und Kommentare ausgeschlossen). |
| Ziele: (Ergebnisbeschreibung) | -Wir testen das Programm mit verschiedenen Tools und lassen auch andere normale Benutzer die App testen. Wir beheben auftretende Bugs und sorgen dafür, dass alle Fehlerquellen abgefangen und bearbeitet werden. |
| Abhängigkeiten: | Arbeitspaket A bis F sollten abgeschlossen sein. |
| Aufwand in 2h Schritten: | Ca. 8 Stunden |
| Benötigte Mittel: | JUnit und UnitTest-Vorlagen für Benutzer. |
| Meilenstein [JA/NEIN] | Dieser Schritt ist ein Meilenstein, da unsere Programm getestet wurde und demnach stabil und zuverlässig läuft.  Falls es ein Meilenstein ist, muss er auch im GANTT Diagramm eingetragen werden. Siehe Zeile Meilensteine. |

## Arbeitspaket H – Dokumentation abschliessen

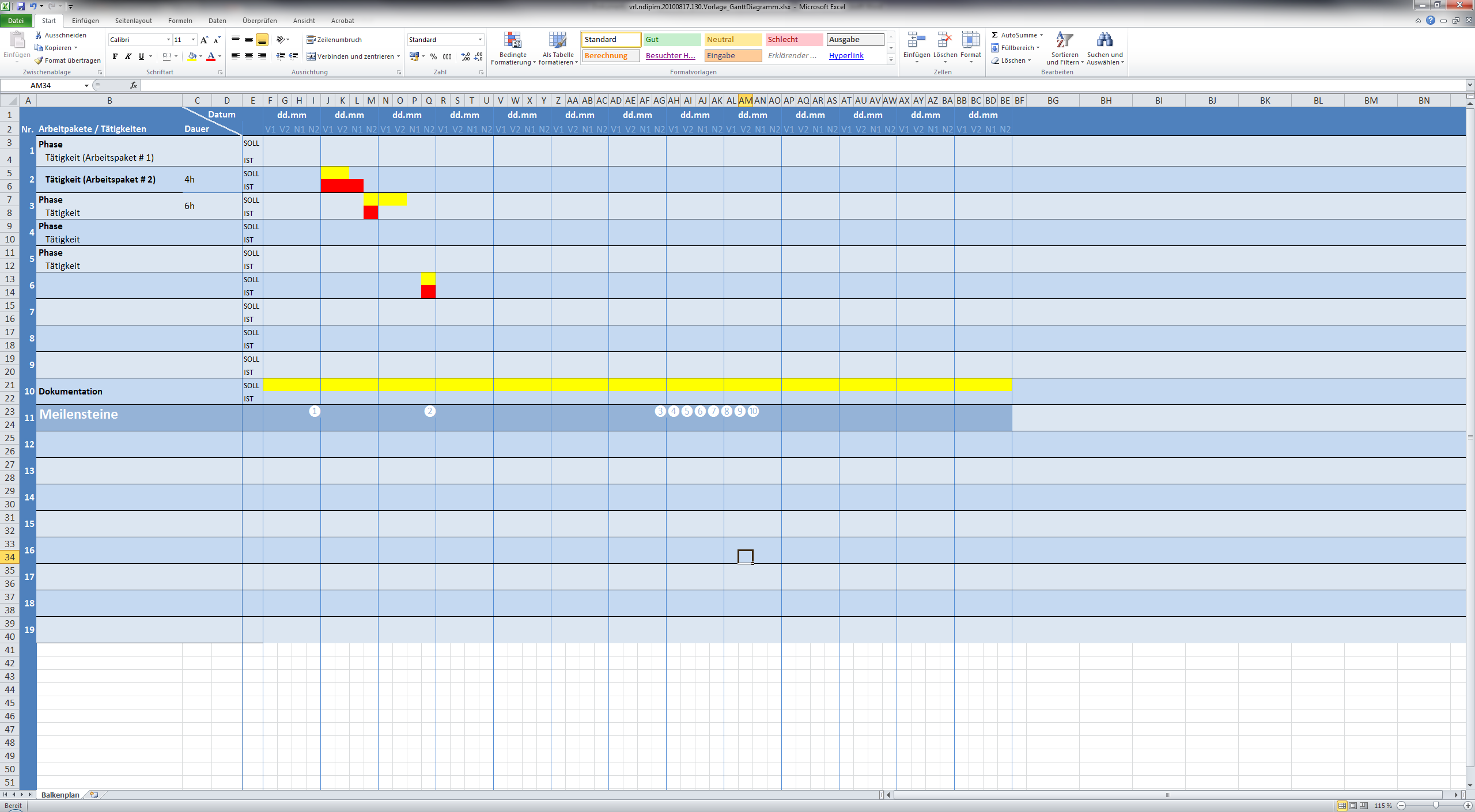
|  |  |
| --- | --- |
|  | Beschreibung |
| Anfangsvoraussetzungen | Das Programm ist vollständig abgeschlossen und getestet worden. |
| Ziele: (Ergebnisbeschreibung) | -Wir schliessen die Dokumentation ab. -Alle gelbe und violette Texte sollen nicht mehr vorhanden sein. -Benutzerhandbuch, Fazit etc. sind geschrieben. |
| Abhängigkeiten: | Arbeitspaket A bis G sollten abgeschlossen sein. |
| Aufwand in 2h Schritten: | Ca. 16 Stunden |
| Benötigte Mittel: | Es sind keine speziellen Hilfsmittel nötig. |
| Meilenstein [JA/NEIN] | Dieser Schritt ist ein Meilenstein, da unser Programm getestet wurde und demnach stabil und zuverlässig läuft.  Falls es ein Meilenstein ist, muss er auch im GANTT Diagramm eingetragen werden. Siehe Zeile Meilensteine. |

## Zeitplanung

### GANTT

Hier Excel Tabelle GANTT einfügen (auch als Bild möglich) Die Excel-Vorlage befindet sich im selben Ordner wie diese Dokumentationsvorlage.

Das Arbeitspaket „Dokumentation“ darf hier in der Planung nicht vergessen werden: Z.B. zu unterst, parallel während der ganzen Dauer einplanen. Mehr Informationen zum GANTT-Diagramm gibt es unter (Wikipedia GANTT-Diagramm).



Ein Feld entspricht 2h (wir rechnen mit einem 8h Tag)

# Technische Dokumentation (Produktdokumentation)

Hier seid ihr frei wie ihr euer Projekt beschreiben möchtet. Die folgenden Kapitel könnt ihr nach Belieben ändern, löschen oder ergänzen. Wichtig ist, dass aber alle Programmteile gut getestet werden, damit es während der Produktepräsentation nicht zu peinlichen Situationen kommt.

Mögliche Themen Elektronik: Mögliche Themen Informatik:

Hardwarebeschreibung Benutzerführung GUI  
 Elektronik Datenbank Design (ERM)  
 Mechanik Hilfsmittel (z.B. verwendetet APIs / Frameworks)

Softwarebeschreibung Verantwortlichkeiten (Tests, die ihr macht)   
 Softwaredesign Verwendete Software  
 Beschreibung der Module  
 Hilfsmittel (Compiler, IDEs usw.)

## Tests

In diesem Kapitel definiert ihr die Tests die Ihr macht.   
Es müssen minimal 5 Tests gemacht werden.

|  |  |
| --- | --- |
| Abschnitt | Inhalt |
| ID | Testfallnummer (ST = Systemtest) |
| Anforderungen | Welche Anforderungen werden durch diesen Testfall abgedeckt. |
| Vorbedingungen | Was muss gegeben sein, damit dieser Test durchgeführt werden kann? |
| Ablauf | Welche Schritte werden bei der Durchführung des Tests durchlaufen? |
| Erwartetes Resultat | Was sollte nun passiert sein? |

|  |  |
| --- | --- |
| Abschnitt | Inhalt |
| ID | ST-01 |
| Anforderungen | FA-01; NFA-03 |
| Vorbedingungen | In der Datenbank existiert ein Benutzer, welcher gesperrt ist. |
| Ablauf | 1. Auf der Startseite wird auf den Link mit dem Label „Login“ geklickt. 2. Der korrekte Benutzername sowie das korrekte Passwort werden eingegeben. 3. Der Button mit dem Label „Login“ wird geklickt. |
| Erwartetes Resultat | Es erscheint eine Fehlermeldung, welche dem Benutzer mitteilt, dass sein Account gesperrt wurde. |

## Testauswertung

Zusammenfassung aller durchgeführten Tests. Nur fehlgeschlagene Tests und Tests mit Bemerkungen müssen in der folgenden Tabelle aufgelistet werden.

|  |  |  |
| --- | --- | --- |
| ID | Erfolgreich | Bemerkungen |
| ST-01 | Ja | Der Testfall war erfolgreich, der Testperson 1 ist jedoch aufgefallen, dass es in der angezeigten Fehlermeldung noch einen Rechtschreibfehler gibt. |
| … | … | … |
|  |  |  |
|  |  |  |

# Benutzerhandbuch

An dieser Stelle kommt ein kleines Benutzerhandbuch. Beschreibt euer Produkt so, dass es jedermann mit der Bedienungsanleitung verwenden kann.

## Anmelden, Login, Logout oder z.B. Inbetriebnahme bei einem Gerät

## Funktionen

# Weiterentwicklungsmöglichkeiten

Beschreibt hier wie es mit eurem Projekt weitergeht nachdem es fertig ist. Zum Beispiel:

* Was könnte man erweitern?
* Was könnte man optimieren?
* Gibt es optionale Arbeitspakete die das Produkt ideal ergänzen?
* Usw.

# Fazit

Hier kommt eure Reflexion zum Projekt.

* Was lief gut/schlecht?
* Wie seid ihr mit dem Endergebnis zufrieden?
* Was habt ihr gelernt?
* War alles vorhanden oder was fehlte noch?
* Usw.

# Abkürzungen / Begriffserklärung

Hier werden alphabetisch alle Fachbegriffe und Abkürzungen erklärt:

HTML Hypertext Markup Language

MySQL Rationales Datenbankverwaltungssystem

PCB Printed Circuit Board

I2C Inter-Integrated Circuit (serieller Datenbus)

# Quellen

Hier müssen sämtliche verwendete Quellen angegeben werden:

*Wikipedia GANTT-Diagramm*. (31. 10 2012). Abgerufen am 12. 11 2012 von http://de.wikipedia.org/wiki/Gantt-Diagramm

1. Anhang
   1. Schema
   2. Fertigungsdokumente
   3. Beschreibung der Ordnerstruktur
   4. Testprotokolle
   5. JavaDoc
   6. Mindmaps
   7. …
   8. Ressourcen
      1. Bilder

In folgenden Galerien sind Bilder verfügbar, die unter gewissen Bedingungen für das Projekt verwendet werden können. Die jeweiligen Nutzungsrechte können für jedes Bild verschieden sein und müssen entsprechend geprüft werden.

[search.creativecommons.org](http://search.creativecommons.org/)  
Allows you to search google and flickr for creative commons images, and several other search providers for non-image based content (try attaching "image of" or "stock photo" to your search terms.

[images.google.com](http://images.google.com/)  
Click on "advanced image search" and then select for the "Usage Rights" dropdown, ALWAYS confirm that the images you find are actually licensed as you selected.

[commons.wikimedia.org](http://commons.wikimedia.org/)  
more than 10 million categorized, photos, illustrations, footage, sound bytes etc.

[flickr.com/creativecommons](http://www.flickr.com/creativecommons/)  
More than 100 million CC licensed images from the popular photo sharing site (15 million with the most flexible "attribution only" license).

[geograph.org.uk](http://www.geograph.org.uk/)  
contributors submit images covering locations all over the uk sorted on ordiancesurvey map references.

[everystockphoto.com](http://everystockphoto.com/)  
Search a selection of free images sites, the license selector in advanced search allows you to filter by creative commons and other free license types.

[creativity103.com](http://creativity103.com/)  
unusual selection of textures, backdrops and abstract photos and illustration

[animalphotos.info](http://animalphotos.info/)  
Specialist collection of animal photos, well categorized.

[carpictures.cc](http://carpictures.cc/)  
A rev-heads wet dream, browse car photos organized by marque and model, sourced from flickr

* 1. IPERKA



Abbildung 3: IPERKA

* 1. Usw.