Konzeptbericht (Applikationsentwicklung)

| Auftraggeber |  |
| --- | --- |
| Projektleiter |  |
| Autor |  |
| Klassifizierung | Nicht klassifiziert, Intern, Vertraulich, GEHEIM |
| Status | In Arbeit, Genehmigt |
|  |  |

Änderungsverzeichnis

| Datum | Version | Änderung | Autor |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Inhaltsverzeichnis

1 Zusammenfassung 3

2 Systemanforderungen 3

2.1 Übersicht Anwendungsfälle 3

2.2 Anwendungsfall XYZ 4

3 Benutzerschnittstelle 5

4 Systemarchitektur 5

4.1 Gliederung der Lösung 5

4.2 *Beschreibung der Elemente (Beispiel 1)* 7

4.2.1 *AnmeldeForm* 7

4.2.2 *Klasse KontoControl* 7

4.2.3 *Klasse Sitzung* 7

4.3 *Beschreibung der Elemente (Beispiel 2)* 8

4.3.1 *AnmeldeForm* 8

4.3.2 *Klasse KontoControl* 8

4.3.3 *Klasse Sitzung* 8

4.4 Schnittstellen 8

5 Datenmodell (grob) 9

6 Qualitätssicherung 9

7 Zusammenfassung der Projektplanung 9

7.1 *Abgleich von Planung und tatsächlichem Verlauf der Phase* 9

7.2 *Risikosituation* 9

7.3 *Planung* *der nächsten Phase* 10

# Zusammenfassung

Dieses Dokument beschreibt die technischen Anforderungen für den Rahmen des Schulprojekts zu realisierenden Keylogger.

# Systemanforderungen

Dieses Programm wird für ein Schulprojekt realisiert und wird in der Zukunft nicht mehr verwendet.

In der Initialisierung haben Sie den Problembereich analysiert und die Hauptaufgaben des Systems bestimmt. Diese sollen nun detaillierter beschrieben werden. Das Mittel hierzu sind Anwendungsfälle (englisch: use cases).

## Übersicht Anwendungsfälle

Zur Use Case Beschreibung gehört ein Use Case Diagramm nach UML. Dieses zeigt alle Akteure und das System selber. Akteure können Benutzer sein. Akteure können aber auch externe Systeme sein, zu welchen Ihr System eine Schnittstelle hat. Aber Achtung: Der Webserver, auf welchem Ihre Web-Site läuft ist kein externes System!

Beispiele:





Faustregel: Jede Hauptaufgabe aus der Voranalyse ergibt einen Use Case. Dabei sollte ein Use Case aber möglichst abgeschlossen sein. Sie müssen dazu die Hauptaufgaben evtl. etwas umformulieren und zusammenfassen/auftrennen.

Schlechte Beispiele:

* *Alter angeben* 🡺 Zu fein. Ist eher ein Schritt innerhalb des Use Cases *Konto eröffnen*
* *Budgetplaner verwenden* 🡺 Unklar. Was ist damit gemeint? Was ist das Ziel dieses Use Cases? Neues Budget erstellen, Ausgabe erfassen, …?

Gute Beispiele: Siehe obige Use Case Diagramme.

Normalerweise wird man für ein Kleinprojekt zwischen drei und ca. acht Use Cases haben. Diese fliessen direkt in die Planung ein: Jede Iteration wird einen oder zwei Use Cases zum Ziel haben.

Use Cases dienen auch als Vorlage für die Testfälle. Diese werden Sie dann im Realisierungsbericht beschreiben.

## Anwendungsfall XYZ

Zu jedem Use Case folgt nun eine Beschreibung des Ablaufs. Im Zentrum stehen die Aktionen der beteiligten Akteure und die Reaktion des Systems auf diese Aktionen. Halten Sie sich für die Beschreibung der Use Cases an die nachfolgende Tabellenvorlage. Nehmen Sie bei der Beschreibung der Anwendungsfälle Bezug auf das Use Case Diagramm und auf die Entitätstypen aus der Voranalyse, respektive aus Abschnitt 4.1.

|  |  |
| --- | --- |
| **Anwendungsfall XYZ** | |
| Kurzbeschreibung | Zweck des Anwendungsfalls in ein, zwei Sätzen |
| Akteure | Welche Akteure und externen Systeme sind beteiligt? |
| Vorbedingungen | Welche Voraussetzungen müssen erfüllt sein, damit der Anwendungsfall überhaupt ausgeführt werden kann? |
| Ablauf | Einzelne Schritte des Anwendungsfalls |
| Resultat | Welche Resultate ergeben sich nach Ablauf des Anwendungsfalls? |
| Ausnahmen | Welche Ausnahmen (z.B. Fehlersituationen) können im Ablauf des Anwendungsfalles auftreten? |

Beispiel:

|  |  |
| --- | --- |
| ***Anwendungsfall „Neues Benutzerkonto erstellen“*** | |
| *Kurzbeschreibung* | *Ein Benutzer registriert sich bei der Budgetplaner-Anwendung und erstellt dabei ein neues Benutzerkonto.* |
| *Akteure* | *Benutzer* |
| *Vorbedingungen* | *Der Benutzer hat noch kein Benutzerkonto.* |
| *Ablauf* | 1. *Der Benutzer öffnet im Browser den Startlink der Budgetplaner-Anwendung.* 2. *Das System stellt die Startseite dar.* 3. *Der Benutzer wählt auf der Startseite den Link „Registrieren“* 4. *Das System präsentiert die Registrierungsseite.* 5. *Der Benutzer gibt seinen Namen, seine E-Mail-Adresse, seinen Benutzernamen und zweimal das gleiche Passwort ein.* 6. *Das System validiert die Angaben des Benutzers und erstellt ein neues Benutzerkonto. Anschliessend präsentiert das System die Bestätigungsmeldung auf der Registrierungsseite.* 7. *Der Benutzer kann jetzt zur Startseite zurückkehren oder sich direkt anmelden.* |
| *Resultat* | *Ein neues Benutzerkonto-Objekt mit den Angaben des Benutzers ist erstellt und in der Datenbank gespeichert worden.* |
| *Ausnahmen* | * *Die Validierung der Benutzerangaben in Schritt 6 schlägt fehl.* * *Es besteht bereits ein Benutzer mit diesem Benutzernamen (Schritt 6).* |

# Benutzerschnittstelle

Beschreiben Sie hier die Benutzerschnittstelle Ihres Systems anhand von Skizzen. Erstellen Sie bei Desktop-Applikationen für jedes Fenster und für jeden Dialog oder, im Falle von Web-Applikationen, für jede Seite eine Skizze. Auf den Skizzen müssen alle wichtigen Elemente der Benutzeroberfläche eingezeichnet sein und ihre Funktion muss im Text kurz beschrieben sein. Sehr wichtig dabei ist, dass die Navigationspfade klar ersichtlich sind. Das grafische Design ist in den Skizzen zweitrangig. Sie können allenfalls das grafische Design anhand eines Beispiels verdeutlichen.

Für die Erstellung der Skizzen können Sie auf zwei Arten vorgehen:

1. Von Hand erstellte Skizzen (gescannt, oder mit Zeichnungstool).
2. Erstellen eines Prototyps.  
   Implementieren Sie in einer ersten Iteration die Fenster und Dialoge Ihrer Anwendung, wobei die Elemente noch keine oder nur rudimentäre Funktion haben. Im Falle einer Web-Applikation erstellen Sie die Seiten in HTML. Achtung: Das Design ist auch hier zweitrangig. Sie können allenfalls ein CSS als Design-Vorschlag erstellen. Im Bericht fügen Sie sodann Screenshots des Prototypen ein.

# Systemarchitektur

In der Initialisierung haben Sie Lösungsvarianten ausgearbeitet und einen Variantenentscheid getroffen. Die Architektur der gewählten Lösungsvarianten haben Sie in einem Blockdiagramm grob beschrieben. Hier geht es nun darum die Architektur zu verfeinern.

Zum Beispiel: Wenn Sie eine Web-Applikation als Block beschrieben haben, so geht es jetzt darum die innere Struktur dieser Web-Applikation festzulegen.

## Gliederung der Lösung

Hier zeigen Sie aus welchen Elementen Ihre Lösung besteht. Elemente sind:

* Schichten/Layers.  
  Schichten gliedern Ihre Lösung horizontal. Eine Schicht bietet eine Schnittstelle der darüber liegenden Schicht an. Sie verwendet die Schnittstelle der darunterliegenden Schicht. Häufig verwendete Schichten sind Persistenzschicht (DB-Layer), Applikationsschicht (Business-Logic), Präsentationsschicht (GUI, UI), siehe auch Three-Tier-Architektur bei Web-Applikationen.
* Pakete.  
  Ein Paket dient zur Kapselung von Funktionalität, welche einen Zusammenhang hat. Beispiel: Benutzerverwaltung, Administration. Pakete können Klassen oder Module enthalten. Pakete können auch weitere Pakete enthalten. Wenn Sie spezielle Libraries verwenden, können Sie auch diese als Paket darstellen.
* Klassen.  
  Zu den Entitätstypen kommen jetzt weitere Klassen hinzu. Dies sind beispielsweise Kontroller-Klassen. Sie kapseln die Applikationslogik. Ausserdem kommen hier auch Klassen hinzu, welche Schnittstellen kapseln (GUI, Proxys, Stubs für Web-Services)
* Module.  
  Wenn Sie nicht objektorientiert entwickeln verwenden Sie Module. Ein Modul enthält dann Sammlungen von Prozeduren. Einzelne dieser Prozeduren bilden die Schnittstelle des Moduls. Ein Skript (z.B. PHP-Skript) kann als Modul dargestellt werden.

Wenn Sie objektorientiert vorgehen illustrieren Sie die Architektur mittels UML-Paket- und Klassendiagrammen.

Wenn Sie nicht objektorientiert vorgehen, verwenden Sie Blockdiagramme zur Illustration. Blöcke, welche ein Modul darstellen enthalten eine Liste der Prozeduren des Moduls. Zeichnen Sie zwischen den Blöcken die Abhängigkeiten ein (gestrichelter Pfeil ausgehend vom abhängigen Block).

Beschreiben Sie jede Klasse/jedes Modul kurz. Was ist der Zweck? Bei den Entitätstypen (objektorientierter Ansatz) können Sie auch auf die Voranalyse verweisen.

Beispiel 1: Dreischichtenarchitektur mit weiterer Unterteilung innerhalb der Applikationsschicht (objektorientiert).



## *Beschreibung der Elemente (Beispiel 1)*

### *AnmeldeForm*

*Das Formular zur Anmeldung des Benutzers.*

### *Klasse KontoControl*

*Enthält die Logik zur Verwaltung der Benutzerkontos. Dazu gehört das Anlegen von neuen Kontos, aber auch das eröffnen einer Sitzung zu einem bestehenden Konto.*

### *Klasse Sitzung*

*Kapselt alle Informationen zu einer Sitzung. Eine Sitzung entsteht, wenn sich der Benutzer anmeldet.*

Beispiel 2: Zweischichtenarchitektur mit weiterer Unterteilung der Applikationsschicht (prozedural/modular).



## *Beschreibung der Elemente (Beispiel 2)*

### *AnmeldeForm*

*Das Formular zur Anmeldung des Benutzers.*

### *Klasse KontoControl*

*Enthält die Logik zur Verwaltung der Benutzerkontos. Dazu gehört das Anlegen von neuen Kontos, aber auch das eröffnen einer Sitzung zu einem bestehenden Konto.*

### *Klasse Sitzung*

*Kapselt alle Informationen zu einer Sitzung. Eine Sitzung entsteht, wenn sich der Benutzer anmeldet.*

## Schnittstellen

Hier beschreiben Sie diejenigen Schnittstellen, welche auf Grund der vorangehenden Abschnitte noch nicht genügend definiert sind. Dazu gehören eine Beschreibung der einzelnen Meldungen, welche über die Schnittstelle übertragen werden (Form/Syntax und Bedeutung der Meldung), sowie einzuhaltende Meldungsabfolgen. Das ist nichts anderes, als ein Protokoll. Sie müssen dabei genügend formal sein, um die Schnittstelle auch implementieren zu können.

Beispiele:

* In einem netzwerkfähigen Spiel müssen Sie die Meldungen auf der Schnittstelle zwischen den Peers bestimmen und festlegen, welche Inhalte und Auswirkungen jede Meldung hat.
* Bei einem Web-Service müssen Sie die Meldungen zwischen dem Client und dem Server definieren. Dazu verwendet man z.B. die XML-basierte Sprache WSDL. Oft kann diese Beschreibung auch aus anderen Programmiersprachen generiert werden.

# Datenmodell (grob)

Wenn Sie in Ihrem System eine Datenbank haben und den prozeduralen/modularen Weg (d.h. nicht objektorientiert) einschlagen, beschreiben Sie hier das Datenmodell mittels ERD. Leiten Sie dazu das ERD aus den Entitätstypen ab. In der Regel wird aus jedem Entitätstyp eine Tabelle. Führen Sie Primär- und Fremdschlüssel ein.

Treffen obige Bedingungen nicht auf Ihr Projekt zu, lassen Sie diesen Abschnitt weg.

# Qualitätssicherung

Dokumentation der durchgeführten QS-Massnahmen (Reviews, Tests).

**Prüfspezifikationen:**

Prüfkriterien, Checklisten oder Testfälle für die einzelnen QS-Massnahmen.

**Testfalltabellen:**

Die Testfalltabellen werden ggf. separat für einzelne oder mehrere Module erstellt.

In der Konzeptphase wird damit begonnen, Testfälle zu sammeln, soweit sie bei der Erarbeitung der Systemanforderungen und der Systemarchitektur bereits erkannt werden können.

In der Realisierungsphase werden die Testfalltabellen in den Realisierungsbericht übernommen und dort weiter detailliert.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **AFo-Nr.** | **Anwendungsfall (ggf. orientiert an Use Cases)** | **Bemerkungen** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Zusammenfassung der Projektplanung

## *Abgleich von Planung und tatsächlichem Verlauf der Phase*

Vorgegebene und erreichte Ergebnisse und Termine.

Eingetretene Risiken.

## *Risikosituation*

Haben sich an der Risikosituation Änderungen (Verschärfung/Entschärfung, neue Risiken) ergeben? Dann beschreiben Sie hier die Risikosituation neu. Führen Sie wiederum konkrete Massnahmen auf, wie Sie den Risiken in der nächsten Phase begegnen wollen.

## *Planung* *der nächsten Phase*

**Projektplanung**

* Meilensteine und wesentliche Termine
* Verweis auf Dokument Projektplan inkl. Projektrisiken

**Prüfplanung**

Fortschreibung der Übersicht über die durchzuführenden QS-Massnahmen und deren Organisation (phasenweise).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Prüfobjekt** | **Termin** | **Prüfer** | **Prüfmethode** (z.B. Review,  Black-box-Test) | **Prüfkriterien, Testfälle o.ä.** (Verweis, ggf. Link) | **Bemerkungen** (z.B.:  Prüfung erfolgreich) |
| Projektinitialisierungsauftrag |  | Lehrperson | Review |  |  |
| Studie |  | Lehrperson | Review |  |  |
| Projektauftrag |  | Lehrperson | Review |  |  |
| Dokument „Projektführung“ |  | Lehrperson | Review |  |  |
| Projektplan |  | Lehrperson | Review |  |  |
| Konzeptbericht |  | Lehrperson | Review |  |  |
| Projektergebnis: z.B.: erstellte SW, Installation, Netzwerk­konfiguration |  |  |  |  |  |
| Realisierungs­bericht |  | Lehrperson | Review,  Tests |  |  |
| Ergebnisbibliothek (Ablage, Directory o.ä.) |  | Lehrperson | Review |  |  |
| Einführungsbericht |  | Lehrperson | Review |  |  |
| Schlussbericht |  | Lehrperson | Review |  |  |