Konzeptbericht (Applikationsentwicklung)

| Auftraggeber | Beat Walter |
| --- | --- |
| Projektleiter | Agash Thamotharampillai |
| Autor | Agash Thamotharampillai, Marc Trittibach, Jonathan Camenzind |
| Klassifizierung | Intern |
| Status | In Arbeit |
|  |  |

Änderungsverzeichnis

| Datum | Version | Änderung | Autor |
| --- | --- | --- | --- |
| 04.03.2014 | 0.1 | Dokument wurde erstellt. | Jonathan Camenzind |
| 11.03.2014 | 0.2 | Use-Case Diagramm erstellt, Klassendiagramme und Systemdiagramme erstellt. | Agash Thamo. |
|  |  |  |  |

Inhaltsverzeichnis

1 Zusammenfassung 3

2 Systemanforderungen 4

2.1 Übersicht Anwendungsfälle 4

2.2 Anwendungsfall Client::Tasteneingabe 5

2.3 Anwendungsfall Server:: Tastatureingaben Einsehen 5

3 Benutzerschnittstelle 6

3.1 Client 6

3.2 Server 6

4 Systemarchitektur 6

4.1 Subsysteme 6

4.2 Gliederung der Lösung 6

4.2.1 Klassendiagramme 6

4.3 *Beschreibung der Elemente (Beispiel 1)* 8

4.3.1 *AnmeldeForm* 8

4.3.2 *Klasse KontoControl* 8

4.3.3 *Klasse Sitzung* 8

4.4 *Beschreibung der Elemente (Beispiel 2)* 9

4.4.1 *AnmeldeForm* 9

4.4.2 *Klasse KontoControl* 9

4.4.3 *Klasse Sitzung* 9

4.5 Schnittstellen 9

5 Datenmodell (grob) 10

6 Qualitätssicherung 10

7 Zusammenfassung der Projektplanung 10

7.1 *Abgleich von Planung und tatsächlichem Verlauf der Phase* 10

7.2 *Risikosituation* 10

7.3 *Planung* *der nächsten Phase* 11

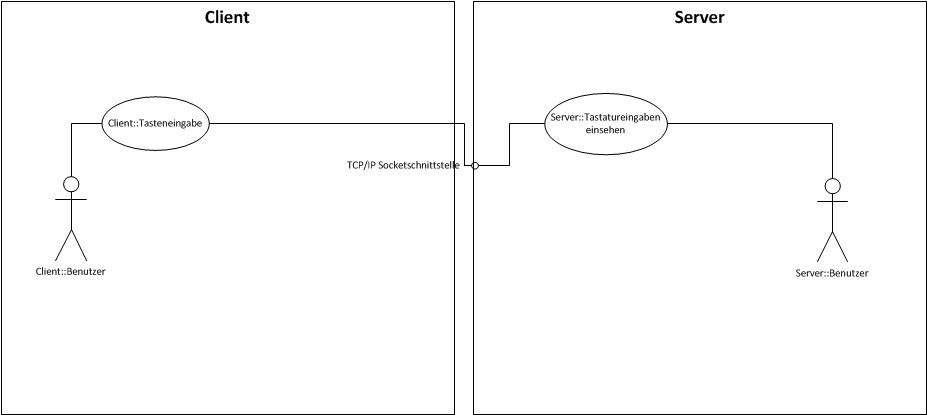
# Zusammenfassung

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

# Systemanforderungen

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



## Anwendungsfall Client::Tasteneingabe

|  |  |
| --- | --- |
| **Anwendungsfall Client::Tasteneingabe** | |
| Kurzbeschreibung | Der Benutzer auf dem Clientcomputer macht eine Eingabe auf der Tastatur. |
| Akteure | Der Benutzer auf dem Computer wo der Keylogger-Client ausgeführt wird. |
| Vorbedingungen | Damit dieser Anwendungsfall geschehen kann, muss die der Keylogger-Client auf dem Computer ausgeführt werden. |
| Ablauf | 1. Der Benutzer befindet sich in einer Applikation auf dem Computer (Windows selbst oder irgendwas sonst) 2. Der Benutzer tätigt in dieser Applikation eine Tasteneingabe |
| Resultat | Die eingegebene Taste wird temporär Zwischengespeichert und bei Gelegenheit an den Server gesendet |
| Ausnahmen | Solange der Keylogger auf dem Client ausgeführt wird sollten keine Ausnahmen möglich sein. Falls keine Verbindung zum Server existiert werden die Daten temporär abgespeichert und bei Verbindung mit dem Server an den Server gesendet. |

## Anwendungsfall Server:: Tastatureingaben Einsehen

|  |  |
| --- | --- |
| **Anwendungsfall Server:: Tastatureingaben Einsehen** | |
| Kurzbeschreibung | Der Benutzer auf dem Servercomputer liest die an den Server gesendeten und ausgewerteten Daten aus. |
| Akteure | Der Benutzer auf dem Computer wo der Keylogger-Server ausgeführt wird. |
| Vorbedingungen | Der Server hat bereits Daten erhalten und ausgewertet. |
| Ablauf | 1. Der Benutzer navigiert zum Dateipfad, wo die Auswertungsdatei existiert. 2. Der Benutzer öffnet die Datei und sieht die Daten an. |
| Resultat | Der Benutzer kennt die Tasteneingaben (Welche Taste, welches Programm, welche Zeit, welcher Client) |
| Ausnahmen | Es sind noch keine Daten auf dem Server vorhanden. Ergo: Die Datei ist leer. |

# Benutzerschnittstelle

## Client

Der Client besitzt an sich keine Schnittstelle, über welcher der Benutzer eventuelle Bedienelemente steuern könnte. Einzig die Konfiguration des Clients kann man mittels XML-Files verändern.

## Server

Der Server bietet, wie auch der Client, keine „richtige“ Schnittstelle. Die Informationen werden in Files gespeichert, diese sind die Schnittstelle zwischen dem Benutzer und dem Server. Ebenfalls, lässt sich der Server über XML-Files konfigurieren.

# Systemarchitektur

## Subsysteme



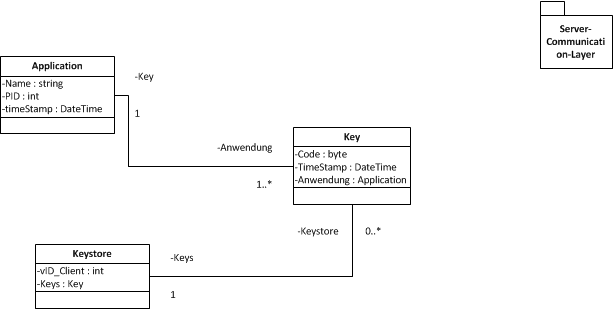
In der Initialisierung haben Sie Lösungsvarianten ausgearbeitet und einen Variantenentscheid getroffen. Die Architektur der gewählten Lösungsvarianten haben Sie in einem Blockdiagramm grob beschrieben. Hier geht es nun darum die Architektur zu verfeinern.

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

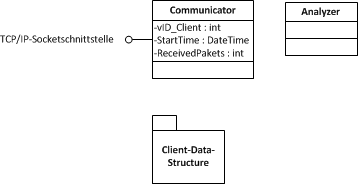
## Gliederung der Lösung

### Klassendiagramme

#### Client



#### Server



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 |  |  |