

Software Engineering Labor-Übung, LVNr: 050015/7

Übungsleiter: Brezany

Dokument: Anforderungsanalyse und Use Case Modell I v.1.0

Projekttitel: „WhatShallIDo“ – a project for the hesitating

Gruppenmitglieder:

<b>MatNr:</b>	<b>Nachname:</b>	<b>Vorname:</b>	<b>e-mail:</b>
0603069	TSIRIGOTI	Anastasia	tsirigoti_anastasia@yahoo.gr
0427435	LADENGRUBER	Roland	rolandl@gmx.at
0525983	TSCHETSCHONIG	Karl	karl@i-dreams.net

Datum: Samstag, 11. November 2006

# 1. Requirements Analysis

## 1.1 Name of the Project

We decided to name our project “WhatShallIDo”, because this name is easy to remember and indicates the purpose of the project.

## 1.2 Short Description

The purpose of this project is to help hesitating people deciding on a various range of topics. A client application will allow the user to select various so-called “modules”, which include suggestions and possibilities on a specific topic.

First of all a module has to be created from a user (every user has the right to create, maximum 10 per user). The new module is assigned to the creator, what means, that this user can administer this module. Other users subscribe to this module and can see the top list of suggestions for the subscribed modules.

For instance: A user adds a module called “What shall I eat”, the program will return a top list of food choices. These choices are determined by other users’ preferences (choices) and maintained by the server application. It follows that the application will ask the user to make daily choices for the subscribed modules.

## 1.3 Functional Requirements

### 1.3.1 Description of Functions

#### Client Application Requirements

The following list represents the major functional requirements:

- subscribe to a module
- show the top list of a subscribed module
- create / edit own module (maximum of 10 modules)
- show top lists of week / month
- multiple user management on one client application
- ability to change user data (username, password, settings,...)
- interact with other users through items
- open links to website from items
- user preferences include: nick name, password, location, street, country, date of birth)

#### Additional requirements for Power Users

- create / edit own module (maximum of 10 modules)
- show personalized lists for subscribed modules
- instant messenger like user list to interact immediately with other users

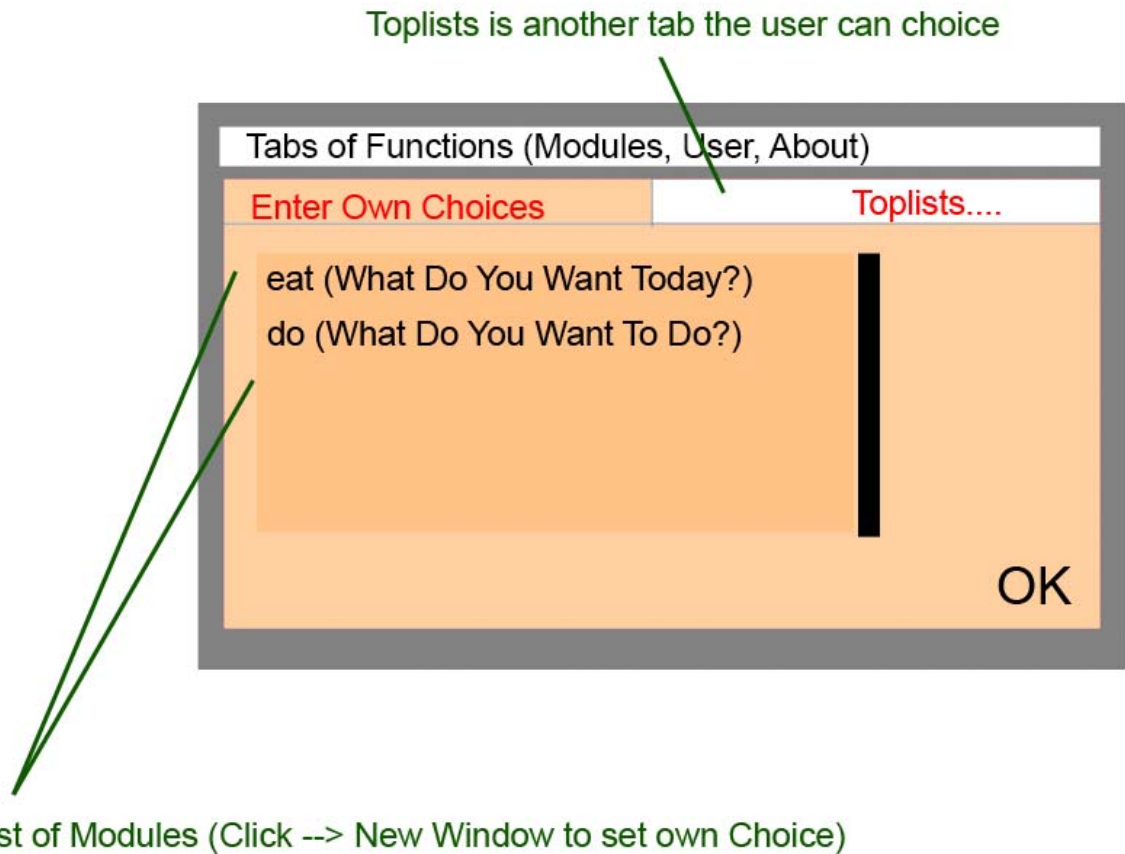
#### Server Application Requirements

The following list represents the major functional requirements:

- filter check for spam words
- filter check for duplicates in modules and choices
- if there are more than 500 entries for a module, the system automatically moves entries older than a day to the statistics tables
- if there are less than 500 entries for a module, move all entries that are older than the latest one of this 500 entries to the statistics tables
- delete old statistics data on a regular basis

### 1.3.2 User Interface

The following graphic will outline the user interface:



## 1.4 Non Functional Requirements

### 1.4.1 Quality Requirements

#### **Usability**

The main quality factor for this program is usability. The procedure starting from the installation of the program and ending in uninstallation must be as simple as it can be realized, because usability is the main reason besides the idea that will keep users wanting to use this application.

#### **Reliability**

Another main characteristic that the program must have is reliability, otherwise users won't choose the program again and may find a competitor program.

#### **Safety and Security**

To avoid any misuse of the system and user data, the data and the transfer must be encrypted.

### 1.4.2 Technical Requirements

- Java Version 1.5 on both Client and Server
- Webserver
- Database
- Client Hardware: at least 1GHZ
- Server Hardware: at least 3GHZ
- Internet Connection

### 1.4.3 Realization Requirements

#### **Installation**

We plan to make the installation for the user as easy as possible. Therefore the setup application will automatically install all files, set registry entries and install the Java JVM if necessary.

#### **Starting the Program**

The program can be started via the start menu or automatically through on startup of the computer.

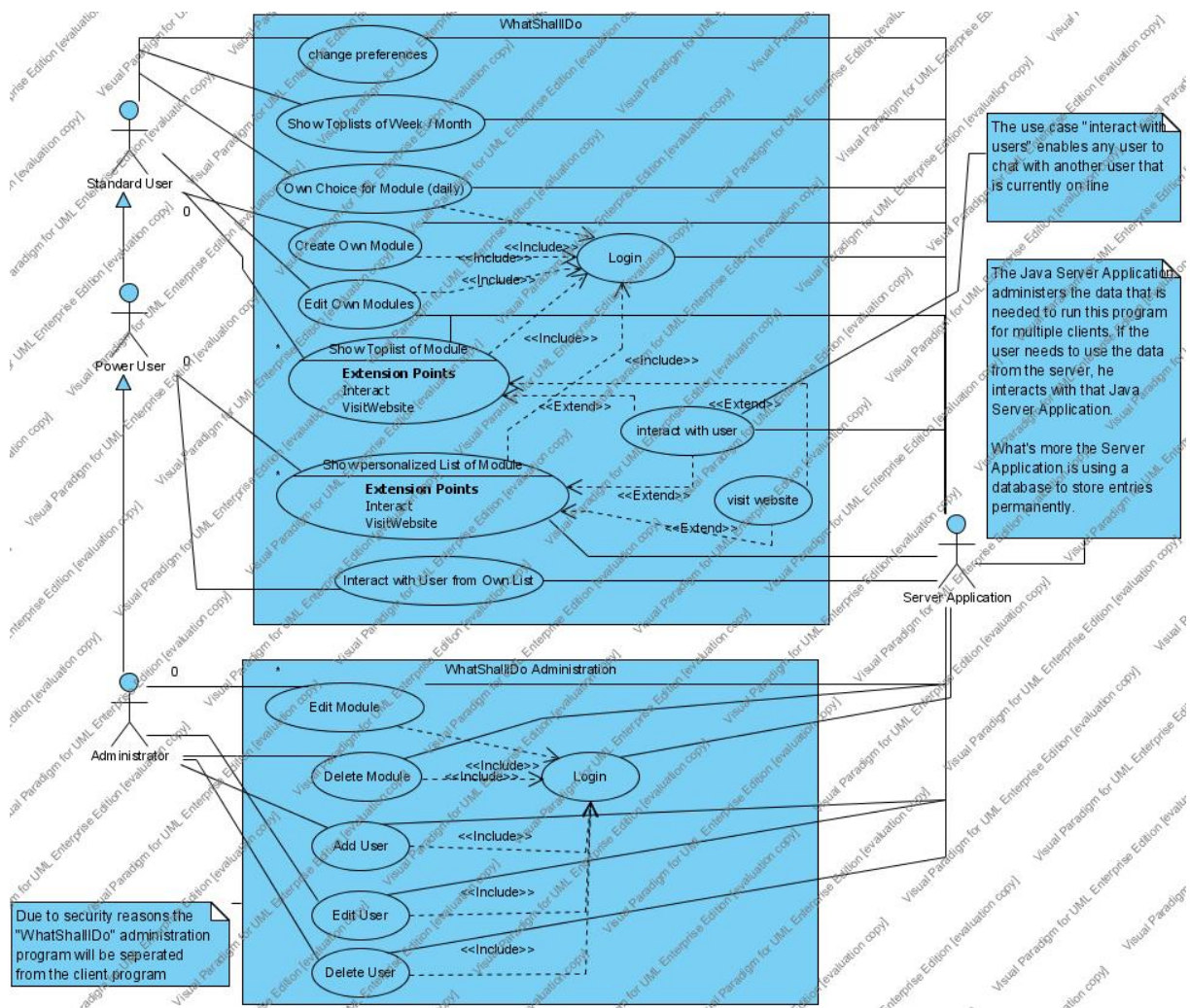
#### **Uninstall**

The uninstall procedure shall make sure, that the whole content of the program are removed completely from the client machine.

### 1.4.4 Miscellaneous

There are currently no additional requirements.

## 2. Use Cases



## 2.1 Use Case “Show Toplist of Modules”

**Target:** List the top choices for a specific module

**Category:** Primary

**Precondition:** module has already been created, Logged on

**Postcondition after success:** actual top list visible

**Postcondition after failure:** Error message displayed

**Actors:** Standard User, Java Server

**Triggering Event:** User Request

**Description of basic process:**

1. [Client] Request from User to receive the top list of a specific module
2. [Client] Checking connection
3. [Server] Receiving Request
4. [Server] Authorizing User
5. [Server] Preparing top lists for return
6. [Server] Returning top list
7. [Client] Receiving Answer
8. [Client] Formatting Data and Displaying

**Extension Interact (RMI Callback):**

9. [Client] Clicking on Item
10. [Server] Receiving Request
11. [Server] Authorizing User
12. [Server] Selecting online user by preference of location\*
13. [Server] Sending best matches to requesting user
14. [Client] Selecting a user for chat request
15. [Server] Receiving Request
16. [Server] Connecting to remote
17. [RemoteClient] Accepting Chat Request
18. [Server] Establishing Connection between Client and RemoteClient
19. [Client+RemoteClient] Receiving Chat Messages, Formatting Output

**Extension VisitWebsite:**

9. [Client] Clicking on Link nearby item
10. [Client] Opening Webbrowser with link

**Alternative processes:**

- 2a. [Client] Client can not connect
- 3a. [Client] Displaying Error Message
  
- 4b. [Server] Unauthorized User
- 5b. [Server] Returning Error Message (access denied)
- 6b. [Client] Receiving, Formatting, Displaying Error
  
- 11c. [Server] Unauthorized User

- 12c. [Server] Returning Error Message (access denied)
- 13c. [Client] Receiving, Formatting, Displaying Error
  
- 13d. [Server] No online users found
- 14d. [Server] Returning Error Message (no users found)
- 15d. [Client] Receiving, Formatting, Displaying Error
  
- 17e. [RemoteClients] Denying chat requests or not reacting
- 18e. [Server] Returning Error Message (no users available)
- 19e. [Client] Receiving, Formatting, Displaying Error

\* Other online users, who chose this item as their favourite, are selected by their specified locations (searching for the nearest location to the main user), to make it possible for people living in the same city f.e. to go out together

## 2.2 Use Case “Own Choice for Modules”

**Target:** Add new Module to own list

**Category:** Primary

**Precondition:** Modules available, Logged on

**Postcondition after success:** Module added to own list

**Postcondition after failure:** Error message displayed

**Actors:** Standard User, Java Server

**Triggering Event:** User Request

**Description of basic process:**

1. [Client] Request from User to add a new module
2. [Client] Checking connection
3. [Server] Receiving Request
4. [Server] Authorizing User
5. [Server] Checking module status
6. [Server] Subscribing user to module
7. [Server] Returning true
8. [Client] Receiving Answer
9. [Client] Displaying the new module

**Alternative processes:**

- 2a. [Client] Client can not connect
- 3a. [Client] Displaying Error Message
  
- 4b. [Server] Unauthorized User
- 5b. [Server] Returning Error Message (access denied)
- 6b. [Client] Receiving, Formatting, Displaying Error
  
- 5c. [Server] module can not be added to users list
- 6c. [Server] Returning Error Message (access denied)
- 7c. [Client] Receiving, Formatting, Displaying Error

## 2.3 Use Case “Create Own Module”

**Target:** Create a new module on the Server

**Category:** Primary

**Precondition:** Logged on

**Postcondition after success:** Module created

**Postcondition after failure:** Error message displayed

**Actors:** Standard User, Java Server

**Triggering Event:** User Request

**Description of basic process:**

1. [Client] Request from User to create a new module
2. [Client] Checking connection
3. [Server] Receiving Request
4. [Server] Authorizing User
5. [Server] Trying to create new module
6. [Server] Returning true
7. [Client] Receiving Answer
8. [Client] Displaying the new module

**Alternative processes:**

- 2a. [Client] Client can not connect
- 3a. [Client] Displaying Error Message
  
- 4b. [Server] Unauthorized User
- 5b. [Server] Returning Error Message (access denied)
- 6b. [Client] Receiving, Formatting, Displaying Error
  
- 5c. [Server] module can not be created
- 6c. [Server] Returning Error Message (access denied)
- 7c. [Client] Receiving, Formatting, Displaying Error

## 2.4 Use Case “Edit Own Modules”

**Target:** change own module preferences

**Category:** Primary

**Precondition:** Module is created and is owned by user, Logged on

**Postcondition after success:** Module preferences changed

**Postcondition after failure:** Error message displayed

**Actors:** Standard User, Java Server

**Triggering Event:** User Request

**Description of basic process:**

1. [Client] Request from User to change the preferences of a module
2. [Client] Checking connection
3. [Server] Receiving Request
4. [Server] Authorizing User
5. [Server] Trying to change module preferences
6. [Server] Returning true
7. [Client] Receiving Answer
8. [Client] Displaying the new module

**Alternative processes:**

- 2a. [Client] Client can not connect
- 3a. [Client] Displaying Error Message
  
- 4b. [Server] Unauthorized User
- 5b. [Server] Returning Error Message (access denied)
- 6b. [Client] Receiving, Formatting, Displaying Error
  
- 5c. [Server] module can not be changed (not enough permissions, not added)
- 6c. [Server] Returning Error Message (failure in changing module)
- 7c. [Client] Receiving, Formatting, Displaying Error

## 2.5 Use Case “Show Toplists of Week/Month”

**Target:** list top items of a specific week or month

**Category:** Primary

**Precondition:** Module is created and is owned by user, Logged on

**Postcondition after success:** toplist of week/month visible

**Postcondition after failure:** Error message displayed

**Actors:** Standard User, Java Server

**Triggering Event:** User Request

**Description of basic process:**

1. [Client] Request from User to display toplist of specific week/month of a module
2. [Client] Checking connection
3. [Server] Receiving Request
4. [Server] Authorizing User
5. [Server] Preparing top lists for return
6. [Server] Returning top list
7. [Client] Receiving Answer
8. [Client] Formatting Data and Displaying

**Alternative processes:**

- 2a. [Client] Client can not connect
- 3a. [Client] Displaying Error Message
  
- 4b. [Server] Unauthorized User
- 5b. [Server] Returning Error Message (access denied)
- 6b. [Client] Receiving, Formatting, Displaying Error

## 2.6 Use Case “Show personalized list of module”

This use case is very similar to “Show Toplists of Module” except that the list itself is constructed depending on contact associations as well as prior interactions with specific users.

## 2.7 Use Case “Login”

**Target:** Authorizing user

**Category:** Secondary

**Precondition:** none

**Postcondition after success:** Other use cases accesible

**Postcondition after failure:** Error message displayed

**Actors:** Standard User, Java Server

**Triggering Event:** Program starting

**Description of basic process:**

1. [Client] starting the program
2. [Client] Checking connection
3. [Server] Receiving Request
4. [Server] Authorizing User
5. [Client] Receiving Answer
6. [Client] Formatting Data and Displaying

**Alternative processes:**

- 2a. [Client] Client can not connect
- 3a. [Client] Displaying Error Message
  
- 4b. [Server] Unauthorized User
- 5b. [Server] Returning Error Message (access denied)
- 6b. [Client] Receiving, Formatting, Displaying Error

## 2.8 Use Case “Interact with user from Own List”

**Target:** directly interacting with a specific user

**Category:** Primary

**Precondition:** Logged on

**Postcondition after success:** interaction

**Postcondition after failure:** Error message displayed

**Actors:** Power User, Java Server

**Triggering Event:** clicking on user

**Description of basic process:**

1. [Client] Opening contact list
2. [Client] choosing user for interaction
3. [Client] Connect to Server
4. [Server] Authorizing User
5. [Server] Connecting to remote user
6. [RemoteClient] Accepting Chat Request
7. [Server] Establishing Connection between Client and RemoteClient
8. [Client+RemoteClient] Receiving Chat Messages, Formatting Output

**Alternative processes:**

- 3a. [Client] Client can not connect
- 4a. [Client] Displaying Error Message
  
- 4b. [Server] Unauthorized User
- 5b. [Server] Returning Error Message (access denied)
- 6b. [Client] Receiving, Formatting, Displaying Error
  
- 7e. [RemoteClients] Denying chat requests or not reacting
- 8e. [Server] Returning Error Message (no users available)
- 9e. [Client] Receiving, Formatting, Displaying Error

## 2.9 Use Case “Edit Module”

**Target:** changing the preferences of a module

**Category:** Primary

**Precondition:** none

**Postcondition after success:** module edited

**Postcondition after failure:** Error message displayed

**Actors:** Administrator, Java Server

**Triggering Event:** Program starting

**Description of basic process:**

1. [Client] select module
2. [Client] Sending Request
3. [Server] Receiving Request
4. [Server] Authorizing User
5. [Client] Receiving Answer
6. [Client] Formatting Data and Displaying
7. [Client] changing preferences
8. [Client] Sending Requests (Changes)
9. [Server] Receiving Request
10. [Server] Authorizing User
11. [Server] Saving Settings
12. [Server] Returning true

**Alternative processes:**

2a. [Client] Client can not connect

3a. [Client] Displaying Error Message

4b/10b. [Server] Unauthorized User

5b/11b. [Server] Returning Error Message (access denied)

6b/12b. [Client] Receiving, Formatting, Displaying Error

## 2.10 Use Case “Delete Module”

**Target:** deleting a module

**Category:** Primary

**Precondition:** none

**Postcondition after success:** module deleted

**Postcondition after failure:** Error message displayed

**Actors:** Administrator, Java Server

**Triggering Event:** Program starting

**Description of basic process:**

1. [Client] select module
2. [Client] Sending Request
3. [Server] Receiving Request
4. [Server] Authorizing User
5. [Client] Receiving Answer

**Alternative processes:**

- 2a. [Client] Client can not connect
- 3a. [Client] Displaying Error Message
  
- 4b [Server] Unauthorized User
- 5b [Server] Returning Error Message (access denied)
- 6b [Client] Receiving, Formatting, Displaying Error

## 2.11 Use Case “Add User”

**Target:** adding a user

**Category:** Primary

**Precondition:** none

**Postcondition after success:** user added

**Postcondition after failure:** Error message displayed

**Actors:** Administrator, Java Server

**Triggering Event:** Program starting

**Description of basic process:**

1. [Client] entering user data
2. [Client] Sending Request
3. [Server] Receiving Request
4. [Server] Authorizing User
5. [Server] Generating User ID
6. [Server] Verifying user data
7. [Server] Saving user
8. [Server] returning summary
9. [Client] Receiving Answer
10. [Client] Output of user data as confirmation

**Alternative processes:**

- 2a. [Client] Client can not connect
- 3a. [Client] Displaying Error Message
  
- 4b [Server] Unauthorized User
- 5b [Server] Returning Error Message (access denied)
- 6b [Client] Receiving, Formatting, Displaying Error

## 2.12 Use Case “Edit User”

**Target:** changing user

**Category:** Primary

**Precondition:** user must exist

**Postcondition after success:** user has been modified

**Postcondition after failure:** Error message displayed

**Actors:** Administrator, Java Server

**Triggering Event:** Program starting

**Description of basic process:**

1. [Client] choose user
2. [Client] Sending Request
3. [Server] Receiving Request
4. [Server] Authorizing User
5. [Server] selecting user which should be changed
6. [Server] Verifying user data
7. [Server] saving user changes to user
8. [Server] returning summary
9. [Client] Receiving Answer
10. [Client] Output of user data as confirmation

**Alternative processes:**

2a. [Client] Client can not connect

3a. [Client] Displaying Error Message

4b [Server] Unauthorized User

5b [Server] Returning Error Message (access denied)

6b [Client] Receiving, Formatting, Displaying Error

## 2.13 Use Case “Delete User”

**Target:** delete existing user

**Category:** Primary

**Precondition:** none

**Postcondition after success:** user deleted

**Postcondition after failure:** Error message displayed

**Actors:** Administrator, Java Server

**Triggering Event:** Program starting

### **Description of basic process:**

1. [Client] choose user
2. [Client] click on delete button to delete the user
3. [Client] Sending Request
4. [Server] Receiving Request
5. [Server] Authorizing User
6. [Server] Delete the user
7. [Client] Receiving answer

### **Alternative processes:**

2a. [Client] Client can not connect

3a. [Client] Displaying Error Message

4b [Server] Unauthorized User

5b [Server] Returning Error Message (access denied)

6b [Client] Receiving, Formatting, Displaying Error

## 2.14 Use Case “Change Preferences”

**Target:** changing preferences of user

**Category:** Primary

**Precondition:** user must exist

**Postcondition after success:** user has been modified

**Postcondition after failure:** Error message displayed

**Actors:** Administrator, Java Server

**Triggering Event:** Program starting

### **Description of basic process:**

1. [Client] Sending Request
2. [Server] Receiving Request
3. [Server] Authorizing User
4. [Server] Verifying user data
5. [Server] saving changed preferences to user
6. [Server] returning summary
7. [Client] Receiving Answer
8. [Client] Output of user data as confirmation

### **Alternative processes:**

- 1a. [Client] Client can not connect
- 2a. [Client] Displaying Error Message
  
- 3b [Server] Unauthorized User
- 4b [Server] Returning Error Message (access denied)
- 5b [Client] Receiving, Formatting, Displaying Error