# MUSTE server protocol

Herbert Lange

Document version 0.0.1

## 1 Design principles

- stateful - authentication happens once

- unidirectional - client sends request, server answers

## 2 Message format

- Client messages (CM) are messages from the client sent to the server

| Message Name | Valid Responses | Description |
|---|---|---|
| CMLoginRequest | SMLoginSuccess SMLoginFail | Send login request |
| CMMOTDRequest | SMMOTDResponse SMSessionInvalid | Request a Message-of-the-day, e.g. the user survey from the server |
| CMDataResponse | SMDataReceived SMDataInvalid SMSessionInvalid | Send result of the survey |
| CMLessonsRequest | SMLessonsList SMSessionInvalid | Request available lessons |
| CMLessonInit | SMMenuList SMLessonInvalid SMSessionInvalid | Start a new lesson |
| CMMenuRequest | SMMenuList SMLessonInvalid SMSessionInvalid | Send request for menus |
| CMLogoutRequest | SMLogoutResponse | Ends a session |

- Server messages (SM) are messages from the servert sent to a client

| Message Name | Description |
|---|---|
| SMLoginSuccess | Login successful |
| SMLoginFail | Login failed |
| SMMOTDResponse | A potential html-fragment for a message of the day |
| SMSessionInvalid | Invalid Session |
| SMDataReceived | Data received |
| SMDataInvalid | Invalid data |
| SMLessonsList | Lesson listing |
| SMLessonInvalid | Invalid lesson |
| SMMenuList | List of possible menus in a lesson |

# 3 Message Datatypes

General format:

```
{"message":string, "parameters":object}
```

With `message` field containing the message name and `parameters` containing (optional) message parameters.

| Message | Parameters | |
|---------|-----------|---|
| CMLoginRequest | `{"username":string,"password":string}` | |
| CMMOTDRequest | `{"token":string}` | |
| CMDataResponse | `{"token":string,"context":string,`<br>`  "data":["field":string,"value":string]}` | **1** |
| CMLessonsRequest | `{"token":string}` | |
| CMLessonInit | `{"token":string,"lesson":string}` | |
| CMMenuRequest | `{"token":string,"score":number,`<br>`  "a":{"tree":string,"grammar":string},`<br>`  "b":{"tree":string,"grammar":string}`<br>`}` | |
| CMLogoutRequest | `{"token":string}` | |
| SMLoginSuccessful | `{"token":string}` | **2** |
| SMLoginFail | `null` | |
| SMMOTDResponse | `{"filename":string}` | **3** |
| SMSessionInvalid | `{"error":string}` | **4** |
| SMLessonsList | `{"lessons":[{"name":string,"passed":boolean,`<br>`              "exercisecount":number}]}` | **5** |
| SMMenuList | `{"passed":bool,"score":number,`<br>`  "a":{"lesson":string,"tree":string,`<br>`       "lin":[{"path":[number],"lin":string}],`<br>`       "menu":{:[[{"score":number,`<br>`                   "lin":[[[number],string]]}]]}},`<br>`  "b":{"lesson":string,"tree":string,`<br>`       "lin":[{"path":[number],"lin":string}],`<br>`       "menu":{:[[{"score":number,`<br>`                   "lin":[[[number],string]]}]]}}}:` | **5** |
| SMLessonInvalid | `null` | |
| SMDataReceived | `null` | |
| SMDataInvalid | `{"error":string}` | **6** |
| SMLogoutResponse | `null` | |

**1** `token` is an identifier assigned to the client session by the server. `context` defines the semantics of `data`.

- For *startQuestionaire* and *finalQuestionaire*: `field` can be one of *Field1* to *Field20* and `value` can either be a number between 1 and 5 for fields with Likert scale and a string for the freeform fields

- For *finishedSession* and *canceledSession*: `field` is *PlayTime* and `value` is the time used for completing the session or before canceling and going back
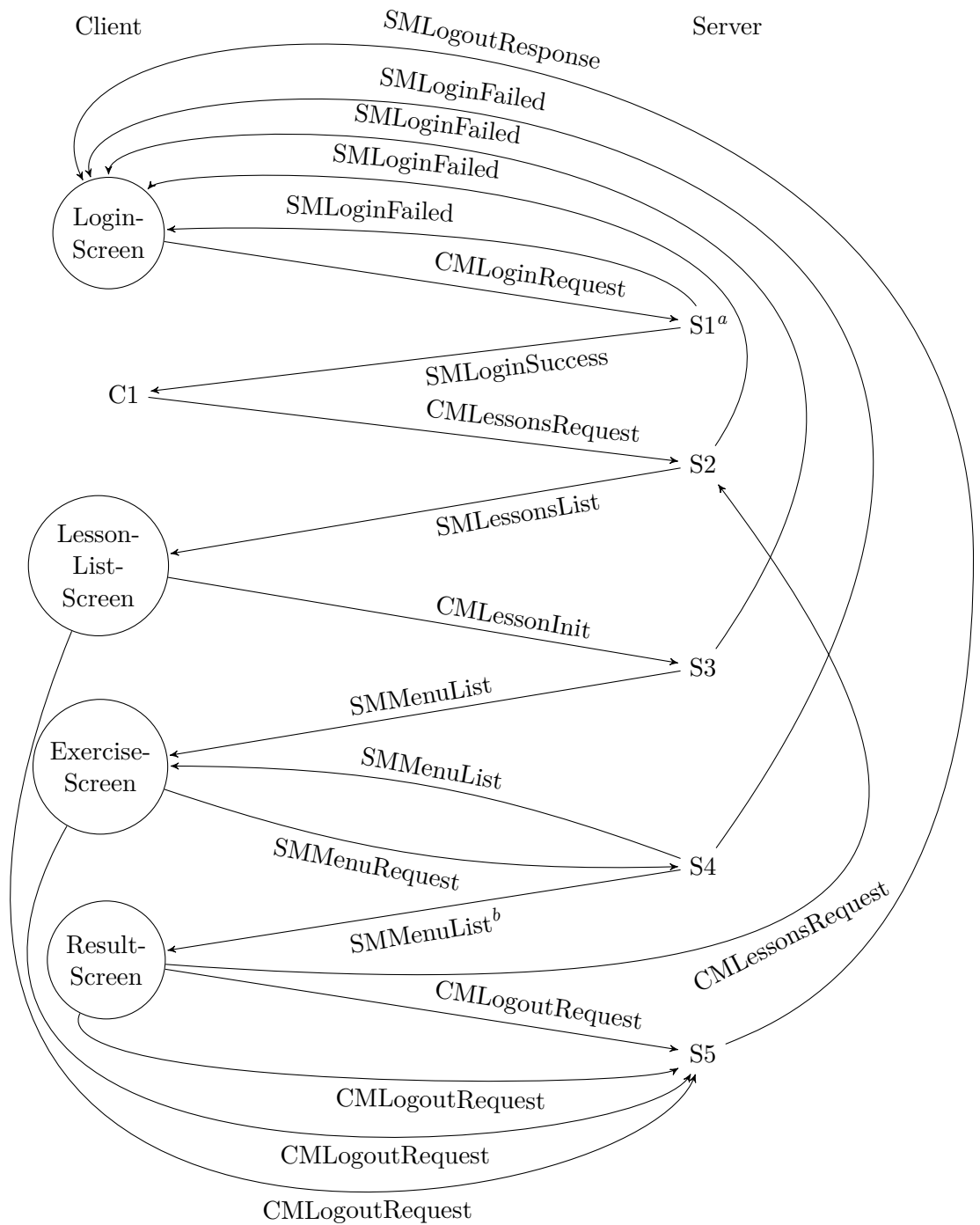
**2** The token to be used by all following client requests

**3** A file name to be displayed as a message of the day

**4** Reason like timeout or not authenticated

**5** lessen *name* and *lesson* are the same as the name of the PGF used for the lesson

**6** Potential error message

# 4 Sequences

Client      *SMLogoutResponse*    Server

*SMLoginFailed*

*SMLoginFailed*

*SMLoginFailed*

*SMLoginFailed*

Login-Screen

*CMLoginRequest*

S1[a]

*SMLoginSuccess*

C1

*CMLessonsRequest*

S2

Lesson-List-Screen

*SMLessonsList*

*CMLessonInit*

S3

*SMMenuList*

Exercise-Screen

*SMMenuList*

*SMMenuRequest*

S4

Result-Screen

*SMMenuList*[b]

*CMLessonsRequest*

*CMLogoutRequest*

S5

*CMLogoutRequest*

*CMLogoutRequest*

*CMLogoutRequest*

---

[a]S1,..,S5,C1 are hidden states  5

[b]If session is passed i.e. if both trees are the same

# 5 Database

## 5.1 ER-Diagram
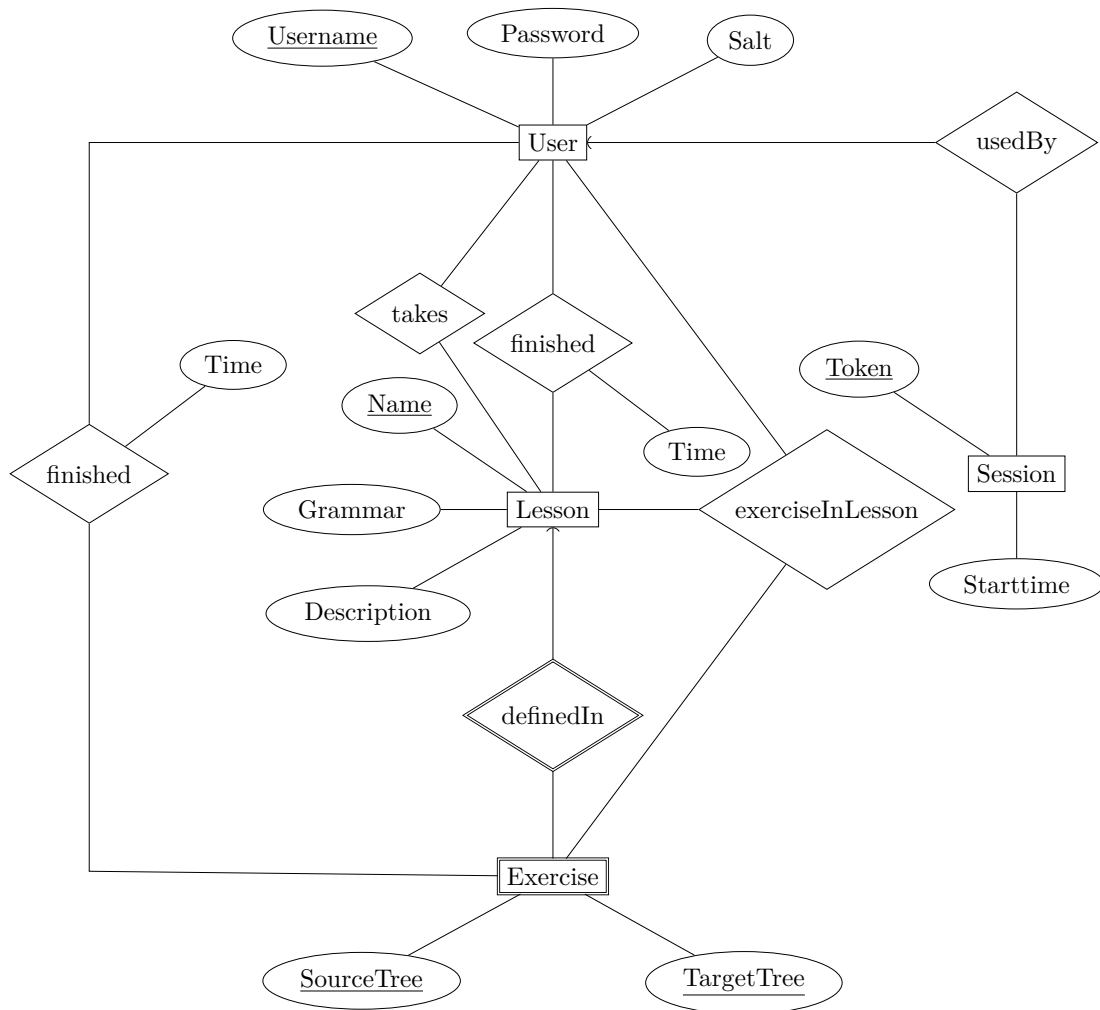


## 5.2 Schema

User(<u>Username</u>,Password,Salt)

Session(<u>Token</u>,Starttime)

Lesson(<u>Name</u>,Description,Grammar)

Exercise(<u>SourceTree</u>,<u>TargetTree</u>,<u>Lesson</u>)
  Lesson → Lesson.Name

FinishedExercise(<u>User</u>,<u>SourceTree</u>,<u>TargetTree</u>,<u>Lesson</u>,Time)

User → User.Username

(SourceTree,TargetTree,Lesson) → Exercise.(SourceTree,TargetTree,Lesson)

StartedLesson(<u>Lesson</u>,<u>User</u>)

User → User.Username

Lesson → Lesson.Name

FinishedLesson(<u>Lesson</u>,<u>User</u>,Time)

User → User.Username

Lesson → Lesson.Name

ExerciseList(<u>User</u>,<u>SourceTree</u>,<u>TargetTree</u>,<u>Lesson</u>)

User → User.Username

(SourceTree,TargetTree,Lesson) → Exercise.(SourceTree,TargetTree,Lesson)

Lesson → Lesson.Name