NodeJS: Authentication

Ashok Dudhat
Germany Startup Jobs
www.germanystartupjobs.com

Agenda

- Protected Routes
- Authentication & Authorization
- HTTP Authentication
- HTTP Authorization

Protected Routes

- Routes in our Express-App look like this:

  /

  /api

  /customers

  ...

Protected Routes

- Routes in our Express-App look like this:

  /

  /api

  /customers

  ...

- What if we want to **protect** them from certain users? I.e. users, that are unknown to us.


ROAD
CLOSED
ROAD
CLOSED
55

Protected Routes

- The guy asks for username and password!

Protected Routes

- The guy asks for username and password!
- You say your username is “hallo“ and your password is “world“.

Protected Routes

- The guy asks for username and password!
- You say your username is “hallo“ and your password is “world“.
- The guy looks up “hallo“ and “world“ in his database and finds you.

Protected Routes

- The guy asks for username and password!
- You say your username is “hallo“ and your password is “world“.
- The guy looks up “hallo“ and “world“ in his database and finds you.
- You are now **authenticated** as known user and you get your **ticket** which **authorizes** you to drive along the road.


ROUTE
66

Authentication & Authorization

Authentication: is the process of verifying that the user is **somebody** the system knows.

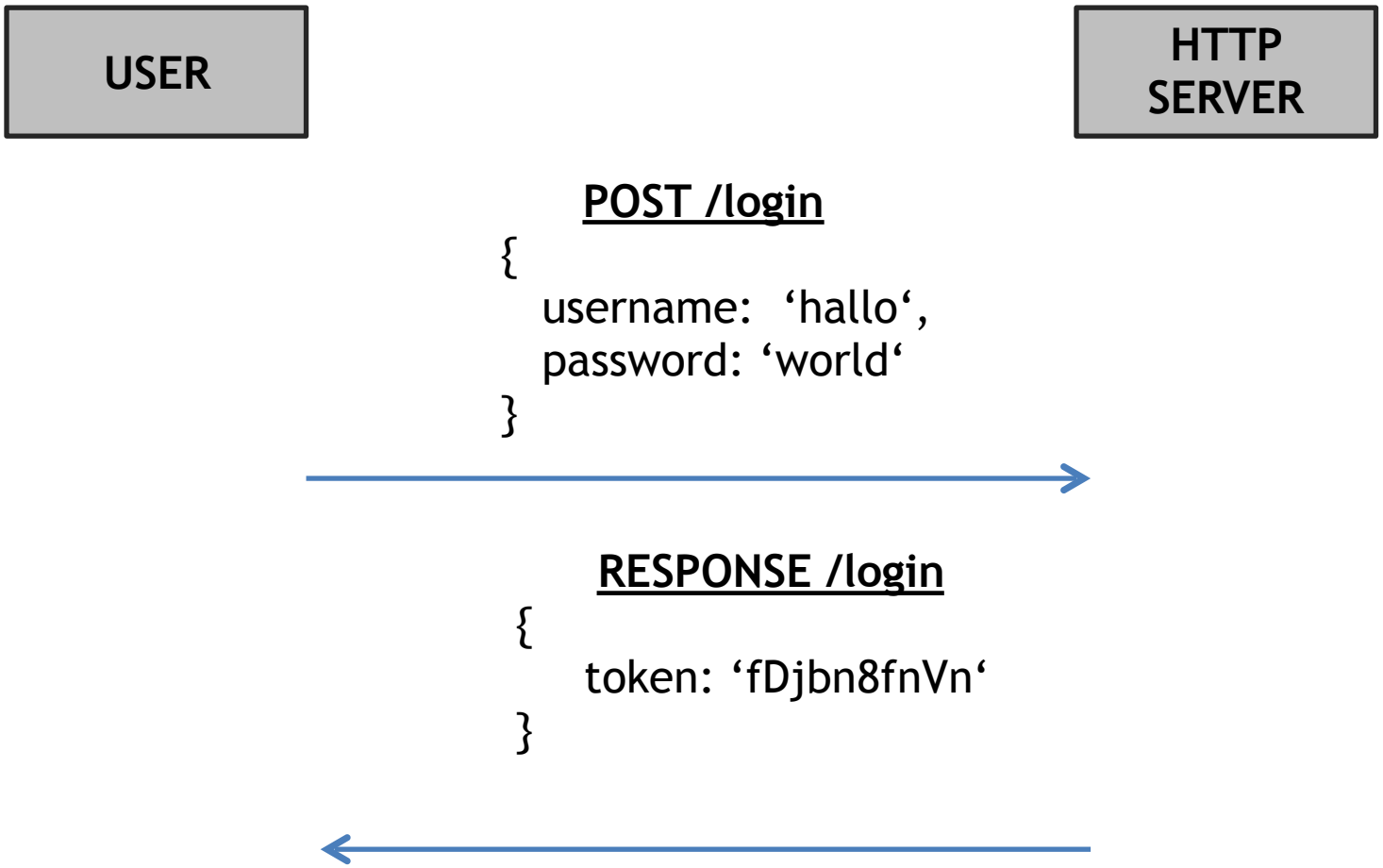
Authorization: is the process of verifying that the user has access to **something** the system owns.

Ticket: A proof that the users is authorized. Mostly it is a token.

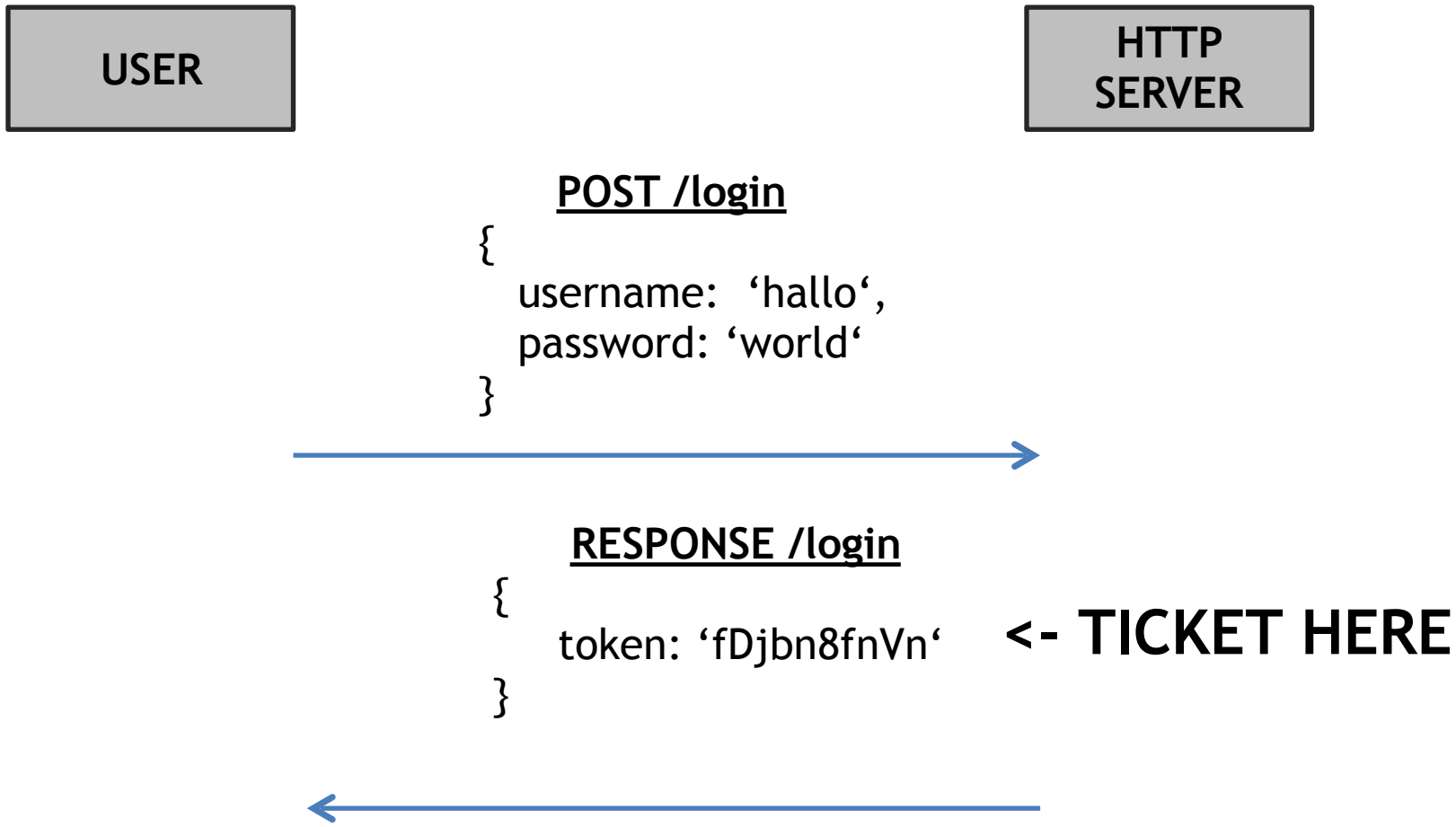
HTTP Authentication

HTTP Authentication

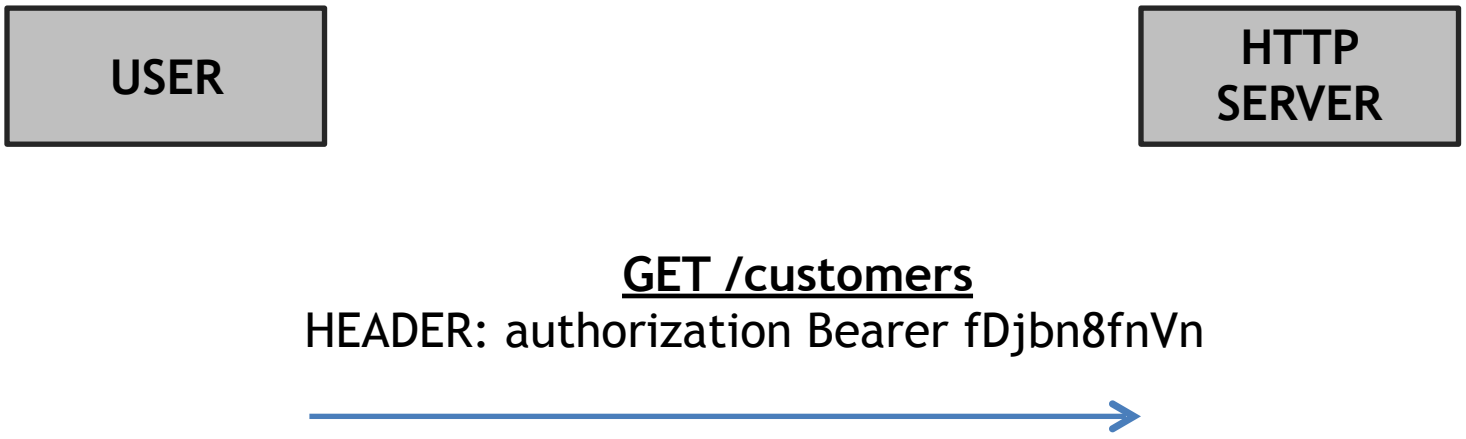
HTTP Authentication Flow



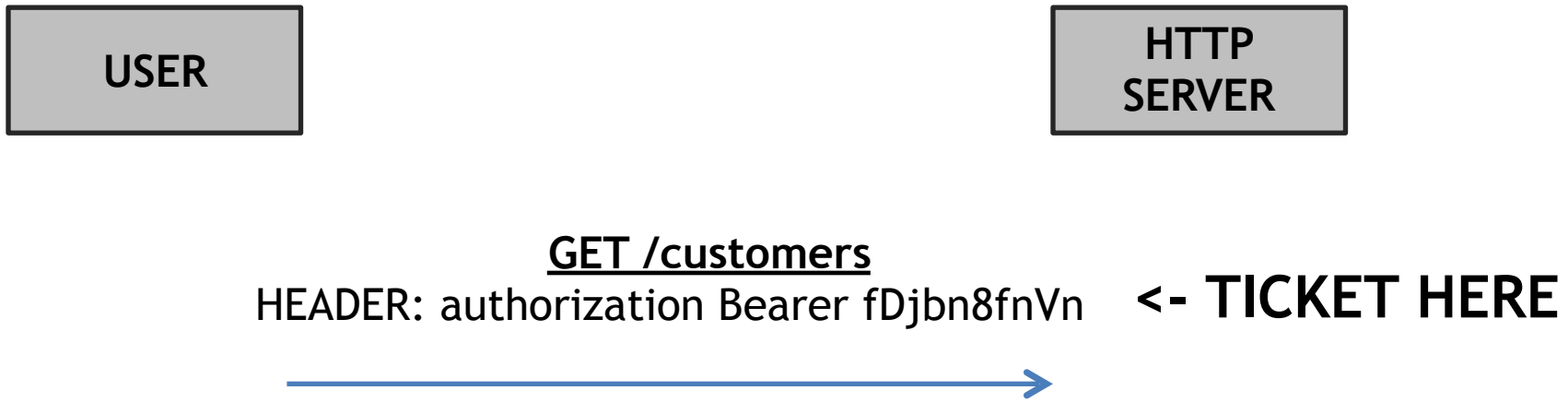
HTTP Authentication



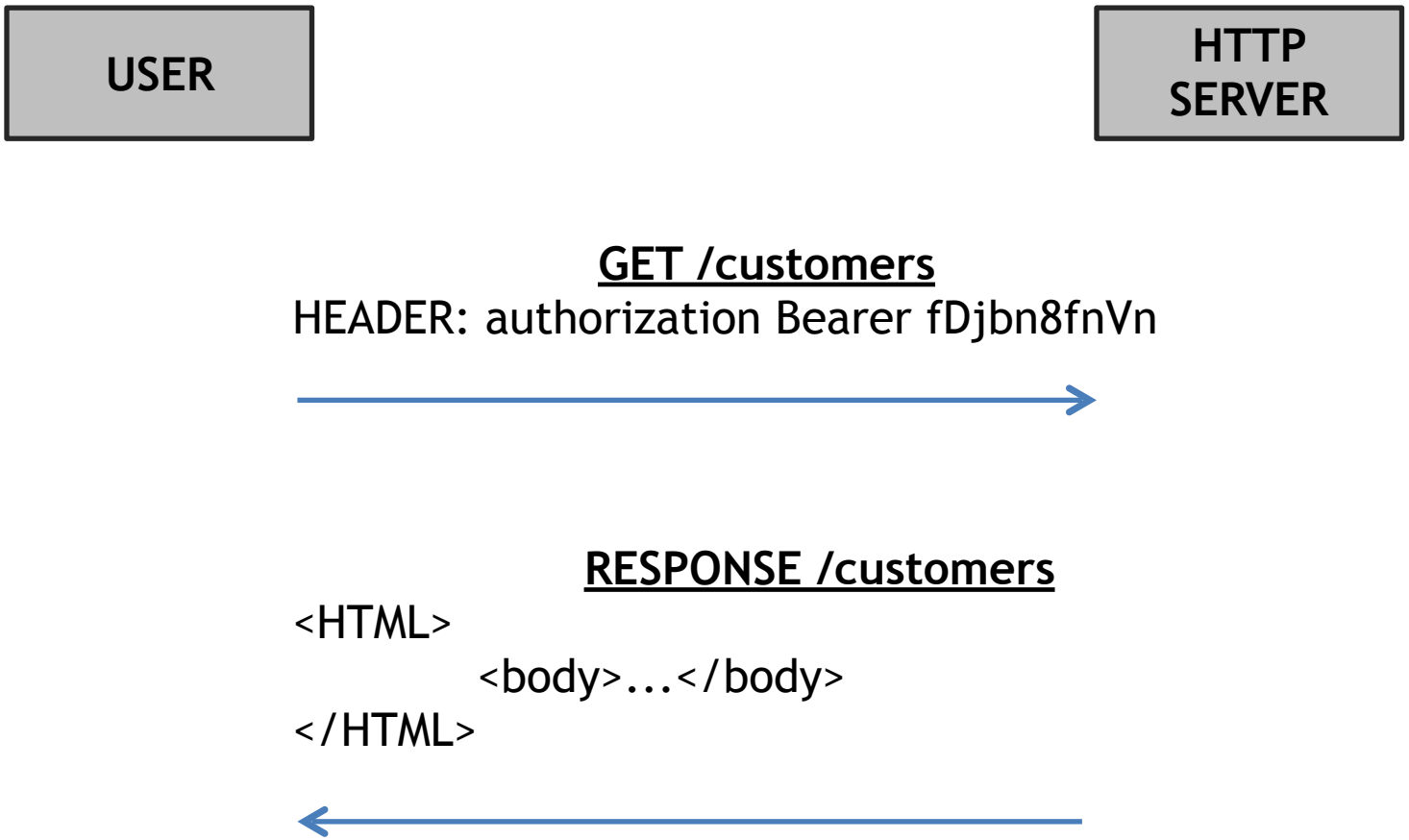
HTTP Authorization



HTTP Authorization



HTTP Authorization



Authentication Types

There are two types of Authentication you can use in any web application development.

1. **Session-based Authentication**
2. **Token-based Authentication**

NodeJS Sessions & Cookies

1. Stateless VS Stateful
2. Cookies & Sessions
3. SessionAuthentication
4. Task

1. Introduction

- Stateless

1. Introduction

- Stateless

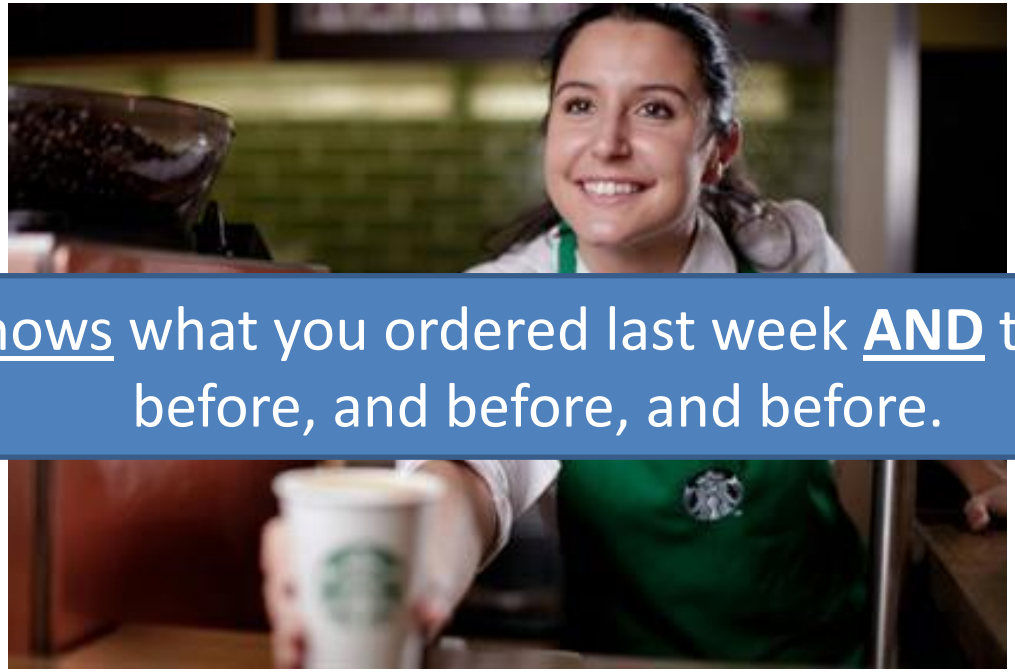
She does not know what you ordered last week.

1. Introduction

- Stateful

1. Introduction

- Stateful



She knows what you ordered last week **AND** the week before, and before, and before.

1. Introduction

- **Request:** HTTP packet sent from client to server („One coffee please“)

1. Introduction

- **Request:** HTTP packet sent from client to server („One coffee please“)
- **Response:** HTTP packet sent back from server to client after it received a Request („There you go – here it is“)

1. Introduction

- **Request:** HTTP packet sent from client to server („One coffee please“)
- **Response:** HTTP packet sent back from server to client after it received a Request („There you go – here it is“)
- **Transaction:** One pair of Request and Response („One coffee please“ – „There you go – here it is“)

1. Introduction

- **Request:** HTTP packet sent from client to server („One coffee please“)
- **Response:** HTTP packet sent back from server to client after it received a Request („There you go – here it is“)
- **Transaction:** One pair of Requests and Responses („One coffee please“ – „There you go – here it is“)
- **Session:** A set of transactions
  - „One coffee please“ – „There you go – here it is“)
  - „One latte please“ – „Okay – here it is“)
  - „One espresso please“ – „Okay.“)

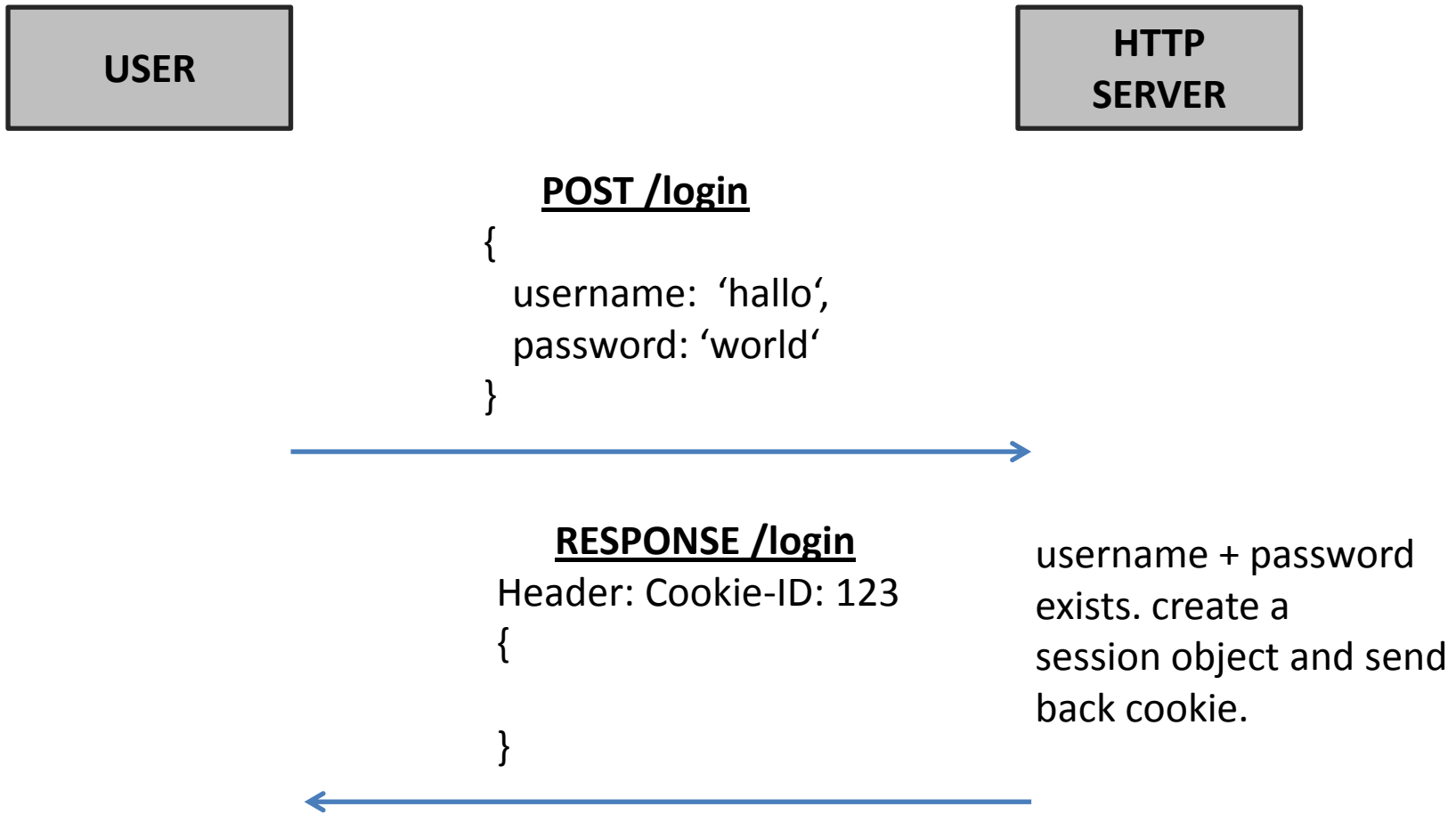
2. Cookies & Sessions

- **Cookies:** is an ID that is sent in the HTTP-Header and identifies a session (saved mostly client-side).
- **Session:** A set of transactions, saved server-side mostly in the memory, as a file or on the database.

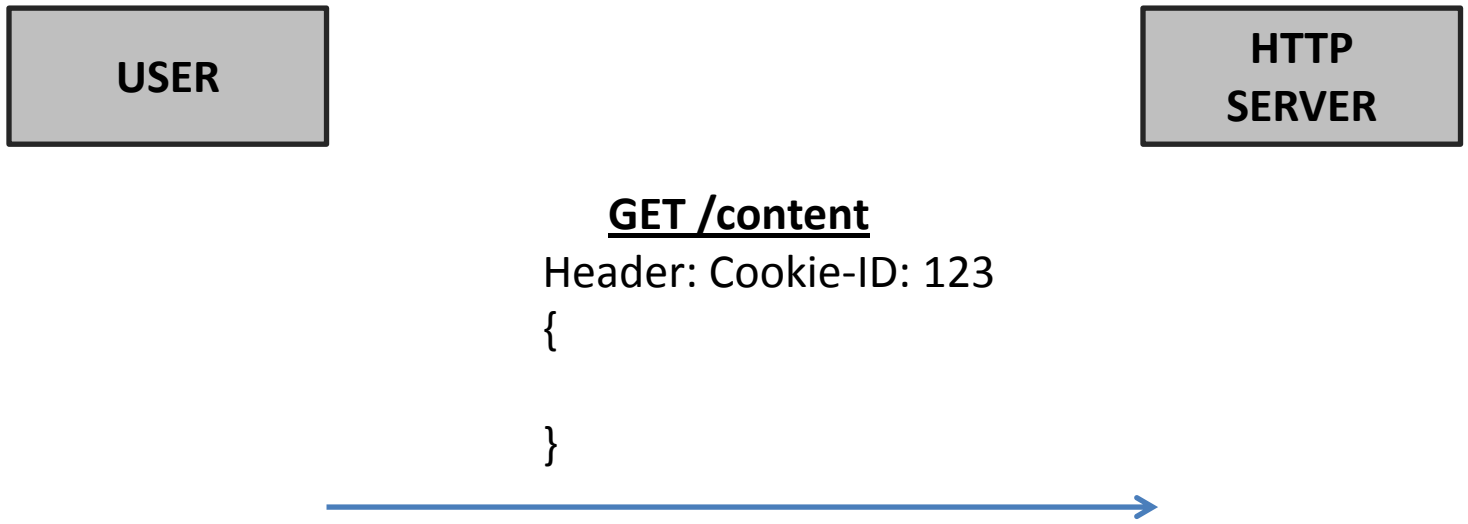
- No JavaScript necessary. When a browser receives a reponse, it automatically saves the cookie in all future requests.
- Getting rid of cookies: delete them in browser configs.

3. Session Authentication (1/4)

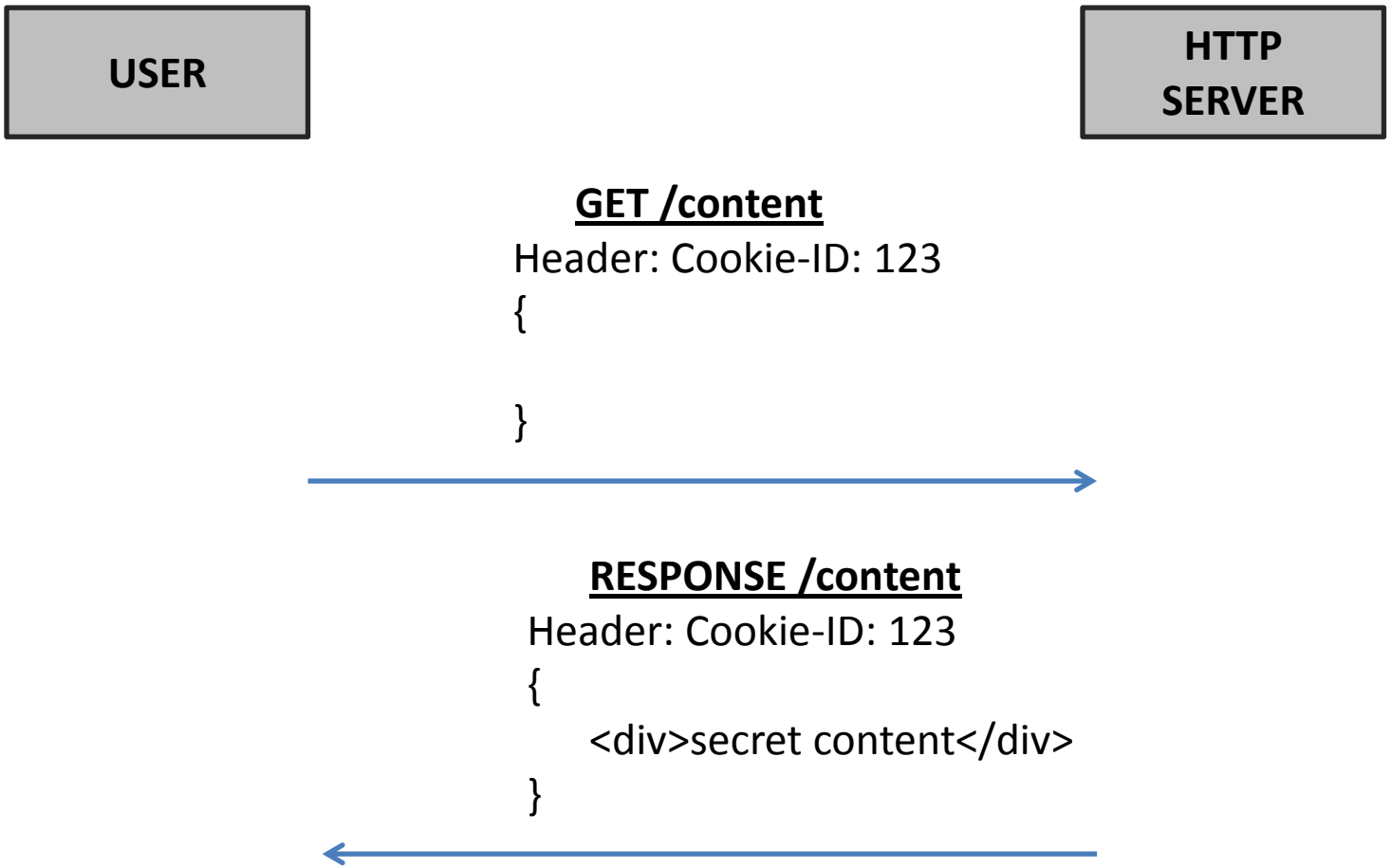
3. Session Authentication (2/4)



3. Session Authentication (3/4)



3. Session Authentication (4/4)



4. Task

4. Task

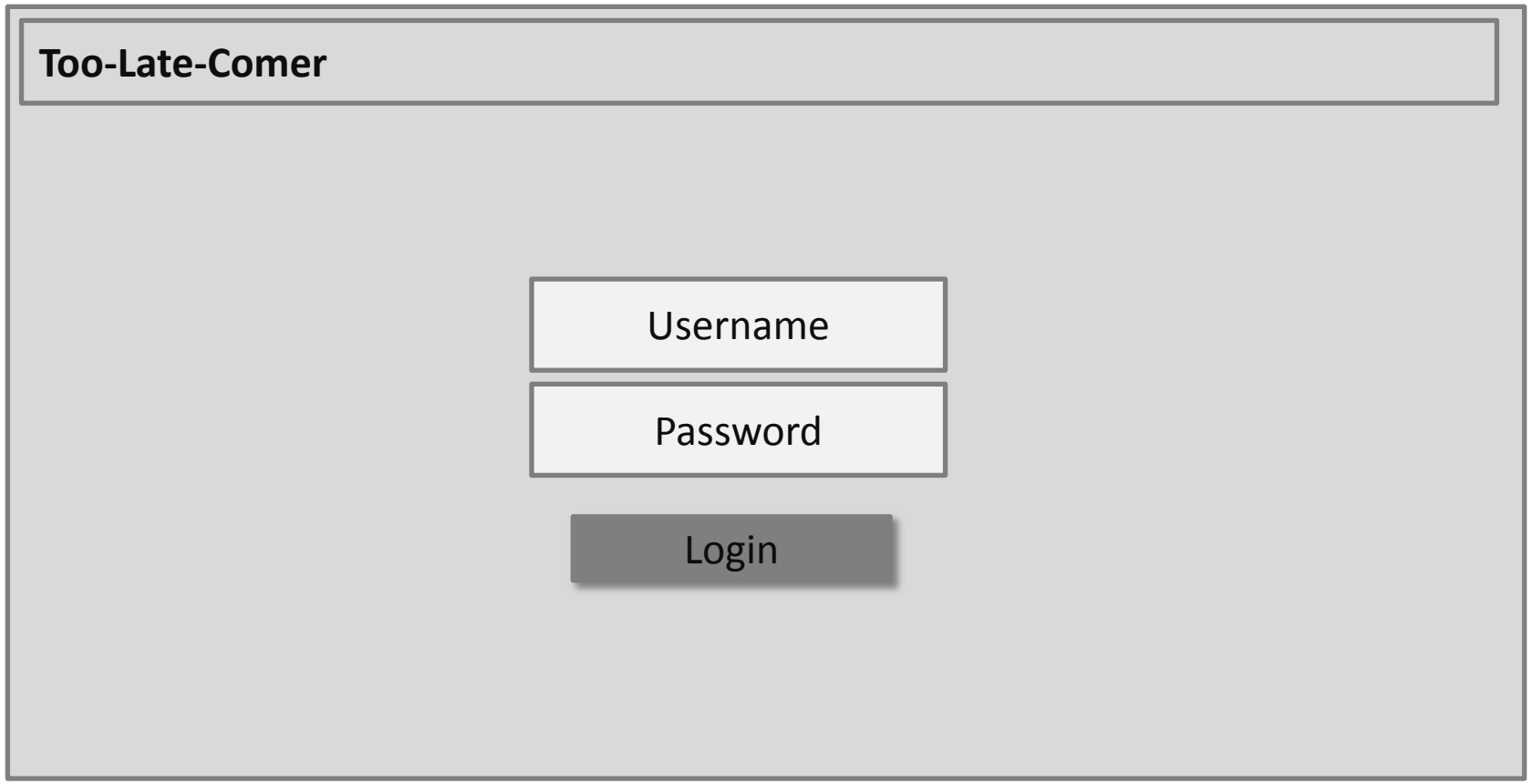
Create a small Too-late-comer website with
Webpack, Jquery, Bootstrap and MongoDB.
The user can keep track of students who come too late.

1) It has three navigation points.
    1) Logout
    2) New Later-Comer (a form where a new too-late comer can be saved)
    3) History (shows a history of people coming too late)

    By default, 3) is the landing page

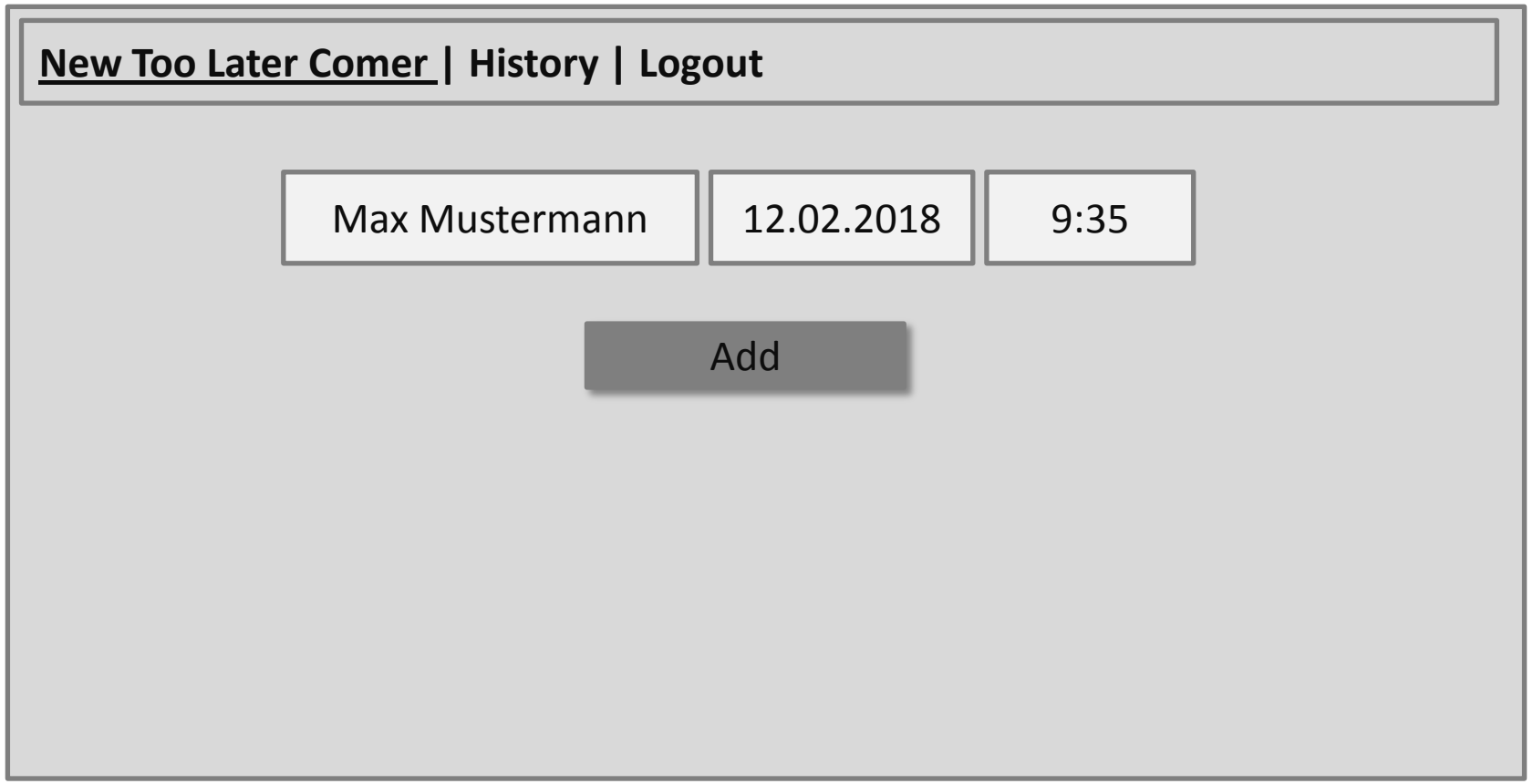
4. Task

1) New Too Later Comer GUI



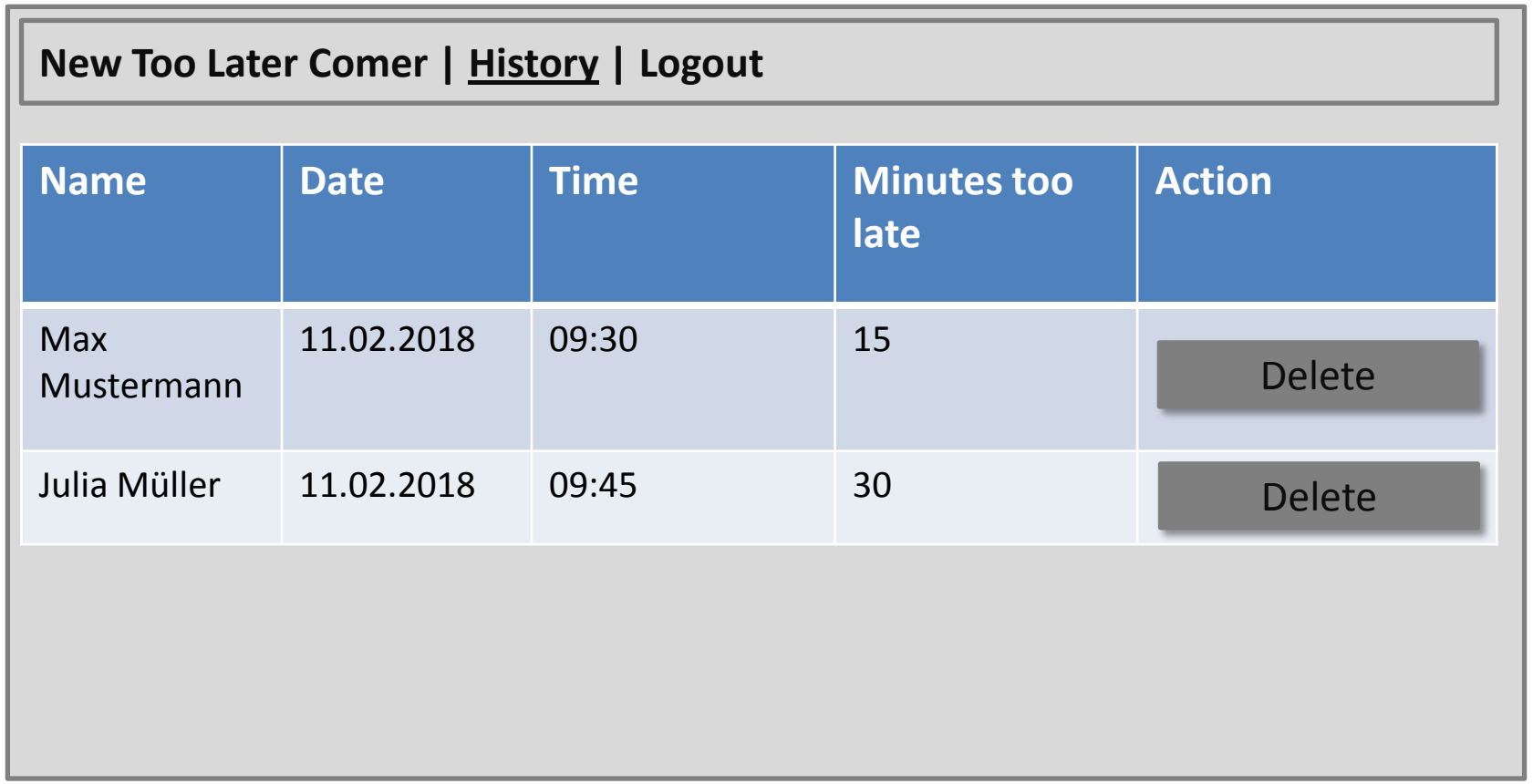
4. Task

1) New Too Later Comer GUI



4. Task

1) History



4. Task

1) Login/Logout