

## EFFORT ESTIMATE SCHALE

|        |                                     |
|--------|-------------------------------------|
| Simple | 1-15 Working hours, incl. research  |
| Medium | 15-25 Working hours, incl. research |
| Hard   | 25+ Working hours, incl. Research   |

### Status (brief comments)

|          |  |
|----------|--|
| Done     | Validated and tested with test users and "mock school"       |
| modified |  |
| rejected | Mainly for simplicity's sake and to focus on main components |

| Topic      | User Story  | Priority |
|------------|---|----------|
| Login      | <p>RGB: As a user I can type in credentials and get directed to the role specific home screen -&gt; <b>validated</b></p> <p>Lim:</p> <p>DoD:</p> <ul style="list-style-type: none"> <li>With valid credentials, the user is logged into the account. -&gt; <b>validated</b></li> <li>The user is informed by a pop-up message if the credentials were incorrect and the login failed -&gt; <b>modified: for simplicity's sake, react components have been broadly ignored for prototype, this applies to the subsequent elements too.</b></li> </ul> <p>ET:</p> <ul style="list-style-type: none"> <li>Create a Login template -&gt; <b>validated</b></li> <li>Set up database, where corresponding user details are saved -&gt; <b>validated, "mock" school created and tested for various users</b></li> </ul> <p>Effort estimate: medium</p> | 1        |
| Logout     | <p>RGB: As a user I can click on a specific button to sign out from account. -&gt; <b>validated, database correctly responds to "logged in / logged out" status</b></p> <p>Limitations:</p> <p>DoD: a message confirms log out to user. -&gt; <b>modified: for simplicity's sake, user is redirected to login page</b></p> <p>ET:</p> <ul style="list-style-type: none"> <li>Create a log out button in nav bar.</li> <li>Create a pop-up message that appears once the user is logged out.</li> </ul> <p>Effort estimate: simple</p>   | (1)      |
| Navigation | <p>RGB: As a user I can navigate across different sections of the web application via a nav bar in order to reach desired location.</p> <p>Limitations:</p> <p>DoD:</p> <ul style="list-style-type: none"> <li>a navbar with clickable items allows moving between different parts. -&gt; <b>validated: two nav bars one general navigation, one for teachers: navigation between "teacher relevant" courses</b></li> </ul> <p>ET:</p>  | 1        |

|               |  |   |
|---------------|--|---|
|               | <ul style="list-style-type: none"> <li>Create a nav-bar that includes items, that lead to another view of the application or opens a specific menu</li> </ul> <p>Effort estimate: simple</p>   |   |
| Edit Password | <p>RGB: As a user I can click on a specific menu and see a window that allows changing the password</p> <p>Limitations: -</p> <p>DoD:</p> <ul style="list-style-type: none"> <li>With a click on the respective menu item, a password editor appears.</li> <li>After entering the new credentials, the password is updated in DB</li> <li>A feedback message is displayed to user</li> </ul> <p>-&gt; all validated: use profile button in navbar and role-specific setting options become visible.</p> <p>ET:</p> <ul style="list-style-type: none"> <li>Create a pop-up message asking for User_ID and password</li> <li>Release DB update</li> <li>Direct user to feedback back</li> </ul> <p>Effort estimate: simple</p>   | 3 |
| Workload view | <p>RGB: As a user I can click on a link and see a specific view which displays the weekly workload of a class. Clicking on scroll tabs, I can change to next and previous week.</p> <p>Limitations:</p> <p>DoD:</p> <ul style="list-style-type: none"> <li>Users see the workload in form of a colored diagram -&gt; validated</li> <li>Users can change from one week to the next -&gt; validated</li> <li>Weekdays are changing correctly -&gt; validated</li> </ul> <p>ET:</p> <ul style="list-style-type: none"> <li>Define parameters, that count the workload according to information retrieved from DB</li> <li>Create a stress bar made of green, yellow and red</li> <li>Make workload a dynamic by defining a method, that defines how much of the stress bar is revealed to the user</li> <li>Define a method that allows spreading workload proportionally over a given period of time.</li> <li>Define calendar function in day-buttons -&gt; validated and extended: holiday dates semester start and end are blocked to avoid faulty entries</li> </ul> <p>Effort estimate: Hard</p> | 2 |
| Semester view | <p>RGB: As a user, I can click on a link and see a specific view which displays all exams of the current semester in form of a table.</p> <p>Limitations:</p> <p>DoD:</p> <ul style="list-style-type: none"> <li>Semester view screen opens upon click.</li> <li>Table consisting of relevant parameters (see mockup) appears</li> <li>Table displays all the exams of a class that are currently entered in DB.</li> </ul> <p>-&gt; all validated</p> <p>ET:</p>  | 2 |

|              |  |   |
|--------------|--|---|
|              | <ul style="list-style-type: none"> <li>Set up a table (double -array) and create a method, so that each box of the table is a date and the first column includes the number of the week</li> <li>Write a method, that maps the DB information about exams into the correct box</li> </ul> <p>Effort estimate: modified: Hard</p>   |   |
| Admin view   | <p>RGB: As an admin I have access to all the student and teacher functions and can set relevant parameters that will apply to all the users in the system.</p> <p>Limitations:</p> <p>DoD: Upon successful login, admins access their role specific home screens and are able to successfully manage the available options</p> <p>ET:</p> <ul style="list-style-type: none"> <li>Create fields that allow editing exam specific information -&gt; <b>validated</b></li> <li>Update database with the provided information -&gt; <b>validated</b></li> <li>(processing information into other channels is implicit)</li> <li>Create fields that allow registering new users and classes -&gt; <b>validated</b></li> <li>Create button that safes classes, exams etc. as static. Once the button is activated, a pop-up message appears to tell the admin that the information ist locked -&gt; <b>partly incorporated</b></li> <li>Create a button to unlock the safed information. Once the button is activated, a pop-up message appeard to force the admin to assure the unlocking of the information -&gt; <b>rejected</b></li> <li>Make sure, the information added and edited by the admin is shared in a database -&gt; <b>validated</b></li> </ul> <p>Effort estimate: Medium</p> | 1 |
| Create exams | <p>RGB: As a teacher, I can add exams in order to reserve a date and make the information available to other users.</p> <p>Limitations: -</p> <p>DoD:</p> <ul style="list-style-type: none"> <li>With a valid entry, a new exam is added to the exam calendar of a class so that the information is available to the users</li> <li>The chosen date is then blocked for new exam entries.</li> <li>The information is displayed on both semester view and the exam bar of the relevant class</li> </ul> <p>ET:</p> <ul style="list-style-type: none"> <li>Create a GUI with a field where information in form of a text can be entered and 6 fields with dropdown menues from whom a teacher can select a parameter</li> <li>Make sure the information typed into the GUI are stored and are comunicated to the semester view.</li> <li>Create a pop up message that appears, if a conflict between an already stored exam and a newly created exam occurs. Make sure the newly created exam is not stored as long as the conflict exists. -&gt; <b>React components rejected, functions implemented and validated</b></li> <li></li> </ul> <p>Effort estimate: Medium</p>   | 2 |

|                              |   |   |
|------------------------------|---|---|
| Edit Exams                   | <p>RGB: As a teacher, I can edit all the exams I created, change exam parameters and information if necessary.</p> <p>Limitations: -</p> <p>DoD:</p> <ul style="list-style-type: none"> <li>Given a valid entry, exam information is edited in the relevant places (cf. Create exams)</li> </ul> <p>ET:</p> <ul style="list-style-type: none"> <li>Create editing field and task relevant options</li> <li>Create a pop up message that appears, if a conflict between a already stored exam and a newly created exam occurs.</li> <li>Make sure the newly created exam is not stored as long as the conflict exists -&gt; React components rejected, functions implemented and validated</li> </ul> <p>Effort estimate: Simple</p>   | 3 |
| Landing page (role specific) | <p>Admin/ Teacher Student (combined):</p> <p>RGB: As a student/teacher/admin I reach a role specific landing page after logging in to see my role specific functions.</p> <p>Limitations: -</p> <p>DoD:</p> <ul style="list-style-type: none"> <li>Student landing pages shows class specific exams (list and semester plan) and workload diagram.</li> <li>Teacher landing page, gives access to seeing all the classes (and their list, workload diagram, semester plan) Teachers can create and edit exams.</li> </ul> <p>-&gt; All implemented and validated</p> <ul style="list-style-type: none"> <li>Admin landing page gives access to all the classes, allows editing exam entries and allows defining exam types, etc. (see admin view). &gt; access and certain editing options have been implemented, not every single feature that could be interesting for an admin has been created (focusing effort)</li> </ul> <p>ET:</p> <ul style="list-style-type: none"> <li>Create different views for each user group</li> <li>Define, which of these views appears first to the user, once the login process was successful</li> </ul> <p>Effort estimate: Medium</p> | 1 |
| Create and edit school       | <p>RGB: As an admin I can create and edit a school (client) to make eXama accessible to users and modify it according to a school's needs.</p> <p>Limitations: -</p> <p>DoD:</p> <ul style="list-style-type: none"> <li>After creating a new school, a new school is created in DB -&gt; rejected: scope, necessity. Clients (schools) would buy some sort of a school-specific, licensed eXama release.</li> <li>When creating a new school, admin can select between different options (e.g. types of exams), create classes and register users. -&gt; validated</li> <li>Once created an admin can edit entries: <ul style="list-style-type: none"> <li>Adding new teachers, classes, exam types</li> <li>(possibly adding homework to workload)</li> </ul> </li> </ul> <p>ET:-</p> <p>Effort estimate: Medium</p>   | 3 |

|                         |  |   |
|-------------------------|--|---|
| Class select option     | <p>RGB: As an admin and teacher one can select one's own classes (teacher) or all classes (admin) to view the classes workload and exams.</p> <p>DoD:</p> <ul style="list-style-type: none"> <li>• When an admin clicks on class dropdown menu, the admin views all classes of school and can select any class.</li> <li>• When a teacher clicks on class dropdown menu, the teacher views all classes he/she is teacher of. If teacher teaches only one class, teacher cannot select between different classes, "Class" is set to this one class.</li> </ul> <p>--&gt; implemented in role-specific side navbar.</p> <p>ET:</p> <ul style="list-style-type: none"> <li>• Define a template where the admin can assign a teacher to a class and the subject of the class</li> <li>• Make sure the database stores the combination of class and its teachers combines with the subject they teach</li> <li>• Link the information stored in the database to the view where exams are defined and edited</li> <li>• Create a button in the nav-bar, where a teacher or admin can choose a class out of a dropdown menu</li> </ul> <p>Effort estimate: Simple</p> | 2 |
| Create, add, edit users | <p>As an admin, I can create, add and edit users to keep classes up-to-date.</p> <p>DoD:</p> <ul style="list-style-type: none"> <li>• The admin can create new user to database list with required attributes.</li> <li>• User info can be updated and is stored in database.</li> <li>• User can be assigned to class.</li> </ul> <p>--&gt; all implemented and validated</p> <p>ET:</p> <ul style="list-style-type: none"> <li>• Create a template, where the admin can add new users. The template must include first and last name, class, birthdate, address, profile</li> </ul> <p>Effort estimate: Simple</p>   | 2 |
| Create, add, edit class | <p>As an admin, I can create, add and edit classes to keep school up-to-date.</p> <p>DoD:</p> <ul style="list-style-type: none"> <li>• The admin can create new classes in database list with required attributes (class teacher, profiles).</li> <li>• Class info can be updated and is stored in database.</li> <li>• Class can be assigned to class teacher?</li> </ul> <p>--&gt; all implemented and validated</p> <p>ET:</p> <ul style="list-style-type: none"> <li>• Create a template, where the admin can add new classes. The template must include a dropdown menu for class-teacher and one initial dropdown menu for profile and a Button to add another profile.</li> <li>• Make sure, that an activation of the button for adding a profile adds another profile dropdown menu</li> </ul> <p>Effort estimate: Simple</p>   | 2 |

- Save the status description in a pdf and upload it into your github repo (name the file "statusOfUserStories\_PROJECTNAME.pdf").