Homework 10

Ohad Ragolsky 7381351, Gruppe 4

## Exercise 1: User Storys

**User Story #1**

**Title:**

Registration option as a student, lecturer or administrator

---

**As a** lecturer, student or administrator **I want to** be able to register the *EGD* System.
**So that** I can get access to the System.

---

**Acceptance Criteria:**

1. Registration as a student
2. Registration as a lecturer
3. Registration as an administrator
4. The User need to register with his Uni-mail Account
5. Data of the Students (e.g. Matrikelnummer, Studiengang, Faculty ) need to be integrated from Klips2.0
6. Data of the lecturer (e.g. Matrikelnummer, Faculty ) need to be integrated from Klips2.0
7. A visual confirmation appears when the registration is successfully made.

---

**Priority:** Must have
**Effort Estimation:** 8h
**Additional Notes:**

**User Story #2**

**Title:**

Lecturers can create and manage exercise groups for their courses

---

**As a** lecturer **I want to** be able to create exercise groups for the courses I teach.
**So that** I can later manage them efficiently.

---

**Acceptance Criteria:**

Define the conditions that must be met for the story to be considered complete.

1. The user can create a new course group.
2. Lecturers shall be able to set group details like session times and the number of students each group can handle.
3. The user can deactivate.
4. The user can delete course group.
5. Before any changes are done a confirmation is needed.
6. A visual confirmation appears when changes are successfully made.

---

**Priority:** Must have
**Effort Estimation:** 12h
**Additional Notes:** dependency User Story #1

**User Story #3**
**Title:**
Lecturers can set group details

---

**As a** Lecture **I want to** be able to set group details like session times and the number of students each group can handle.
**So that** I can set the course deteils and boundaries of each course

---

**Acceptance Criteria:**

1. Calculate the capacity of the course.
2. If course capacity is full: no distribution for this course possible.
3. If course capacity is not full: recognize the available time slots of the student
   And Calculate schedule conflicts.
4. If time slot is valid and no schedule conflicts: distribute the student to the course.
5. Else: manual administration processes needed. Notify the student

---

**Priority:** Must have
**Effort Estimation:** 10h
**Additional Notes:** Dependency: User Story #2

**User Story #4**
**Title:**
Students view of available sessions

---

**As a** student **I want to** view the available sessions.
**So that** I can later register for all courses I want to attend exercises for in one semester .

---

**Acceptance Criteria:**
Define the conditions that must be met for the story to be considered complete.

1. The user need at first to log in into the system to view the available sessions.
2. The system will view of available sessions.
3.

---

**Priority:** Must have
**Effort Estimation:** 4h
**Additional Notes:** dependency: User Story #1 - user need to have an account:

**User Story #5**

**Title:**

Students can register all courses they want to attend exercises for in one semester

---

**As a** student **I want to** register for all the courses I want to participate in in one semester.
**So that** I can get my preferred group that fit my schedules

---

**Acceptance Criteria:**

1. The user need at first to log in into the system to view the available sessions.
2. The system will view of available sessions
3. The user can register for all courses they want to attend exercises for in one semester.
4. The user shell be able to mark times where they are NOT available due to other circumstances.
5. The system will then distribute students across groups in a fair manner, considering their available time slots and any schedule conflicts they may have with other courses.
6. The system will try to minimize the number of students that could not be assigned a group.
7. The system will view the selected courses and them times for a confirmation.
8. The system will send Notifications that will let students know which groups they've been assigned to.
9. If a student could not be assigned, manual administration will be necessary.

---

**Priority:** Must have
**Effort Estimation:** 14h
**Additional Notes:** dependency: User Story #4 - view of available sessions need to be available for users.

**User Story #6**

**Title:**

The system sends students Notifications of the groups they've been assigned to

---

**As a** Lecture **I want to** the system to send students Notifications of the groups they've been assigned.
**So that** the students will know he has been assigned to the course.

---

**Acceptance Criteria:**

1. If the student to the course distributed. Notify student
2. Else: manual administration processes needed. Notify the student

---

**Priority:** Must have
**Urgency:**
**Effort Estimation:** 10h
**Additional Notes:** Two-Way Dependency: User Story #5 - Students can register all courses they want to attend exercises for in one semester and User Story

## Exercise 2. Retrospective

a) A quick look at the burndown chart reveals that the team was working about 4 issues and faced challenges resolving several issues, which caused delays in the schedule during days 0–2, 3–4, and 5–8, because there the Tasks weren't updated.
By the end of the deadline, not all issues were resolved.
When looking on the issue board we can estimate the issues that they had difficulties with.
By examining the issue board, we can identify the difficulties the team encountered:
**Days 0–3:** It looks like the team appears to have worked on creating an architectural overview of the system. But in days 0-2 the Team was behind schedule.
after it the team was back on Track and this task was completed, bringing them back on schedule (issue #8).
**Days 3–5:** The next task was probably to create an overview of all available groups (issue #6). This took them 5 hours spread across 2 sprint days. Despite the effort, they were still slightly behind schedule (1 hour) because of a slowdown in days 3-4 .
**Days 5–9:** Assuming the team worked on creating the database (issue #7), but they likely faced significant challenges (days 5-8). This delayed them further, and the task was only completed by day 9, but let them still be behind the schedule .
**Day 9 onward:** it looks like they started to Work on issue #1. They started late on day 9, leaving insufficient time to complete it before the sprint ended.

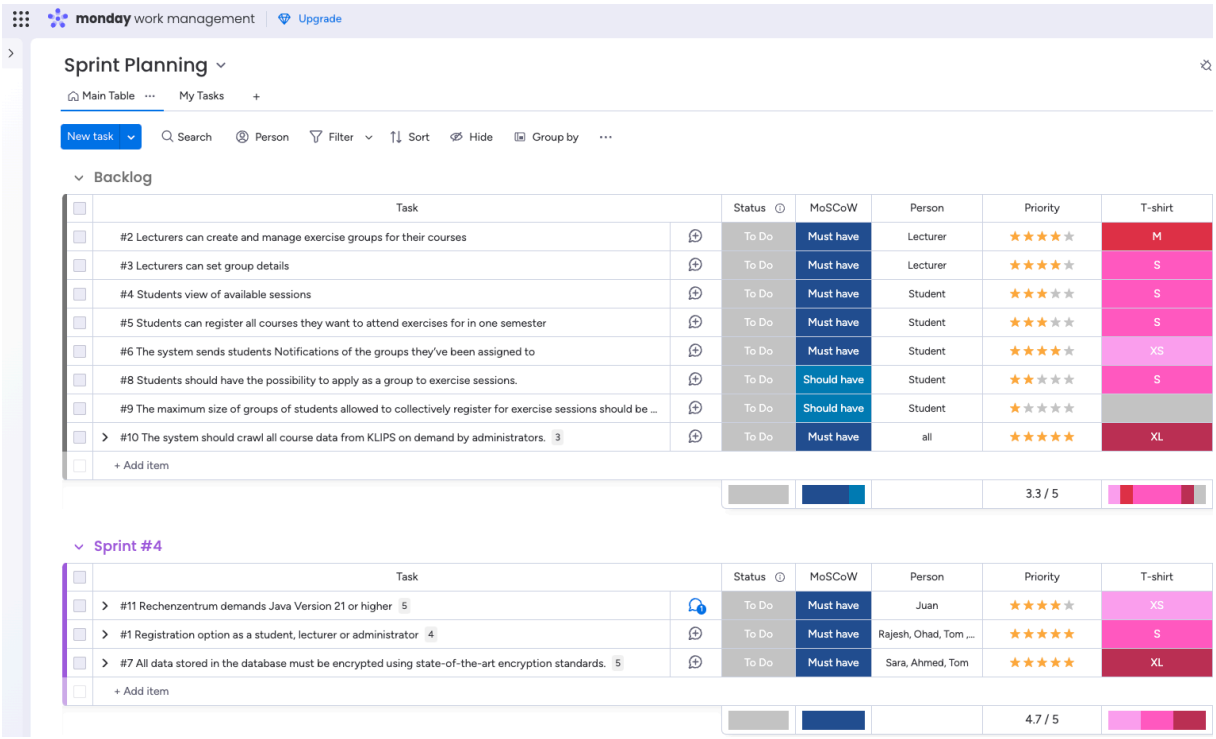
b) **Additional factors contributing to delays:**
- **(days 3-4)** The new update of Spring deprecated some API endpoints which required some minor refactoring.
- **(from Day 9):** The rework of the group distribution algorithm took longer than expected and was not as straight forward as initially planned
- **(Days 5–9):** Timothy "left" the team, impacting productivity.
- **(Days 0–2 but could be also 5–9):** No one was allowed to come to office for 2 days, so everything had to be coordinated online. .

c) How would you proceed with the unfinished issue #1?
At the Sprint Review Meeting we will talk about why the issue #1 couldn't be complete.
At the Sprint Retrospective Meeting we will talk about what went well and went should be improved.
Then it will probably need to be continued and done at the next spring (Sprint 2).

## Exercise 3. Sprint Planning

### Sprint Goal: initialize the system for lecturer

**a)**

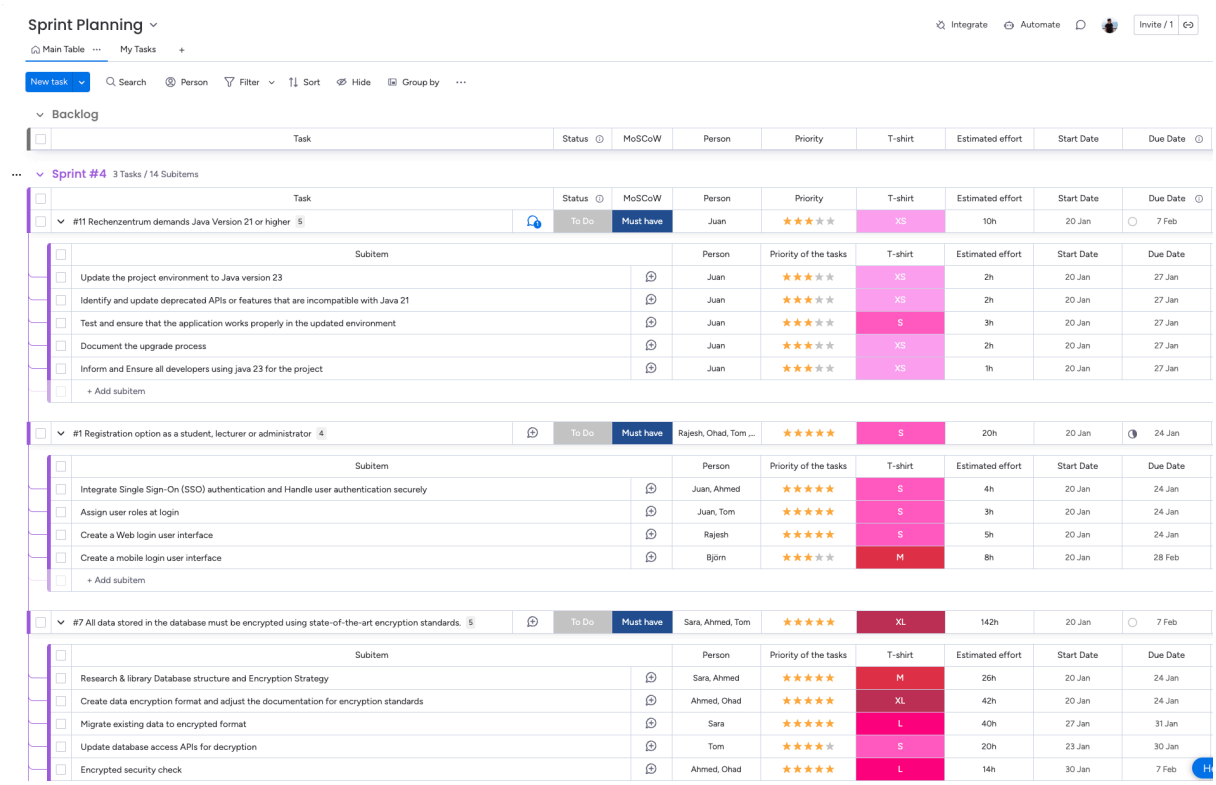



**b) pre-selection of issues that would contribute to your sprint goal:**

#7: All data stored in the database must be encrypted using state-of-the-art encryption standards.

The Goal of this sprint is to ensure the database is fully encrypted using state-of-the-art encryption standards, making the system secure and ready for testing by real students.

## (c) - (f)



Sprint Planning

Main Table · My Tasks

New task · Search · Person · Filter · Sort · Hide · Group by

Backlog

| Task | Status | MoSCoW | Person | Priority | T-shirt | Estimated effort | Start Date | Due Date |
|---|---|---|---|---|---|---|---|---|

Sprint #4 3 Tasks / 14 Subitems

| Task | Status | MoSCoW | Person | Priority | T-shirt | Estimated effort | Start Date | Due Date |
|---|---|---|---|---|---|---|---|---|
| #11 Rechenzentrum demands Java Version 21 or higher 5 | To Do | Must have | Juan | ★★☆☆☆ | XS | 10h | 20 Jan | 7 Feb |

| Subitem | Person | Priority of the tasks | T-shirt | Estimated effort | Start Date | Due Date |
|---|---|---|---|---|---|---|
| Update the project environment to Java version 23 | Juan | ★★☆☆☆ | XS | 2h | 20 Jan | 27 Jan |
| Identify and update deprecated APIs or features that are incompatible with Java 21 | Juan | ★★☆☆☆ | XS | 2h | 20 Jan | 27 Jan |
| Test and ensure that the application works properly in the updated environment | Juan | ★★☆☆☆ | S | 3h | 20 Jan | 27 Jan |
| Document the upgrade process | Juan | ★★☆☆☆ | XS | 2h | 20 Jan | 27 Jan |
| Inform and Ensure all developers using java 23 for the project | Juan | ★★☆☆☆ | XS | 1h | 20 Jan | 27 Jan |
| + Add subitem | | | | | | |

| Task | Status | MoSCoW | Person | Priority | T-shirt | Estimated effort | Start Date | Due Date |
|---|---|---|---|---|---|---|---|---|
| #1 Registration option as a student, lecturer or administrator 4 | To Do | Must have | Rajesh, Ohad, Tom ... | ★★★★★ | S | 20h | 20 Jan | 24 Jan |

| Subitem | Person | Priority of the tasks | T-shirt | Estimated effort | Start Date | Due Date |
|---|---|---|---|---|---|---|
| Integrate Single Sign-On (SSO) authentication and Handle user authentication securely | Juan, Ahmed | ★★★★★ | S | 4h | 20 Jan | 24 Jan |
| Assign user roles at login | Juan, Tom | ★★★★★ | S | 3h | 20 Jan | 24 Jan |
| Create a Web login user interface | Rajesh | ★★★★★ | S | 5h | 20 Jan | 24 Jan |
| Create a mobile login user interface | Björn | ★★★☆☆ | M | 8h | 20 Jan | 28 Feb |
| + Add subitem | | | | | | |

| Task | Status | MoSCoW | Person | Priority | T-shirt | Estimated effort | Start Date | Due Date |
|---|---|---|---|---|---|---|---|---|
| #7 All data stored in the database must be encrypted using state-of-the-art encryption standards. 5 | To Do | Must have | Sara, Ahmed, Tom | ★★★★★ | XL | 142h | 20 Jan | 7 Feb |

| Subitem | Person | Priority of the tasks | T-shirt | Estimated effort | Start Date | Due Date |
|---|---|---|---|---|---|---|
| Research & library Database structure and Encryption Strategy | Sara, Ahmed | ★★★★★ | M | 26h | 20 Jan | 24 Jan |
| Create data encryption format and adjust the documentation for encryption standards | Ahmed, Ohad | ★★★★★ | XL | 42h | 20 Jan | 24 Jan |
| Migrate existing data to encrypted format | Sara | ★★★★★ | L | 40h | 27 Jan | 31 Jan |
| Update database access APIs for decryption | Tom | ★★★★☆ | S | 20h | 23 Jan | 30 Jan |
| Encrypted security check | Ahmed, Ohad | ★★★★★ | L | 14h | 30 Jan | 7 Feb |

The first two issues (#11 and # 1)are in second place. If they are not done till the end of the sprint, they should be looked at and worked on for the next sprint.
The most important issue for this spring is issue #7, we want ensure the database is fully encrypted, what making the system secure and serves as building blocks for later preparation for testing by real students and lecturers.