Delft University of Technology

Software project
CSE2000

Requirements

Group 3C:
Luca Becheanu
Alex Ciurba
Alexandru Lungu
Melle Schoenmaker
Toma Zamfirescu

TU Coach: *Alan Hanjalic*
TA: *Alves Marinov*
Client: *Educational Innovation Projects*

April 30, 2021

Requirements

As a general note each user will have a role that gives access to that part of the application, marked as follows:

- Teaching assistant (marked TA)
- Head teaching assistant (marked HTA)
- Teacher/lecturer (marked TL)

Must haves

- Ability for a user to register attendance or notes per meeting, per group, and also on an individual level. (TA, HTA, TL)
- Ability for a user to enter interventions, such as deadline extensions or doing extra work (HTA, TL). (One or multiple follow ups, as threads with multiple parts).
- Users must be able to follow up on whether students lived up to these interventions by receiving a notification after the deadline. (HTA, TL)
- A way of notifying a user that deadlines for student-specific agreements passed. (TA, HTA, TL)
- A user must be able to see all groups, and have a list of the events within a group. (TL)
- Single Sign On (test server/NetID) authentication must be integrated. (First dummy login)
- Ability to upload reports from BuddyCheck and git analysis. (HTA, TL) (Should/Could)
- Ability to create, edit or import rubrics for a course edition, these rubrics should apply to every group taking part in the course edition. (HTA, TL)
- Ability to import a list of students that are each linked to a group (and a TA). (HTA, TL)
- The hierarchy order in the GUI must be as follows: Course, Course Edition, Groups, Weeks, Rubrics, Attendance, Notes, (could be one level higher, above weeks) Interventions (TA, HTA, TL)
  For TA: Interventions are not shown to TAs (unless in their group, if marked private). Only groups that belong to that TA are visible. For HTA and TL: Interventions are visible. All groups are visible.

Should haves

- A user should be able to log the problems of the group or an individual. (TA, HTA) (Must have)
- A user should be able to mark notes regarding personal reasons as confidential to prevent anyone unauthorized from seeing them. (HTA, TL)
- A user should be notified if a specific student stands out in BuddyCheck reviews. (TA)
- A user should be able to post deadlines for TAs during the project. (TL)
- A user should be able to see the deadlines imposed by lecturers. (TA, HTA, TL)
- A user should be able to filter within the list of groups to find a group by a specific student. (Head TA/TL could be able to filter groups by TA) (TA, HTA, TL) (Should/ Could have)

Could haves

- The grades per group/per course could be exported from the application and then imported to Brightspace/Osiris by the responsible user. (HTA, TL)
- The interface could contain an agenda which lists the deadlines from interventions/lecturers/assignments for the current day, coming days, and previous days.
- The interface could allow the user to change their color scheme by picking from a list of color schemes (Light, Dark, etc.).

- The user could see grade statistics per course edition. (TA, HTA, TL)
- The user could be able to import rubrics as CSV files instead of having to create rubrics inside the application. (HTA, TL)
- The user could be able to view the percentage of students that has received their grade (TA, HTA, TL)
  For TA: Show the percentage of students with a grade in the groups they manage.
  For HTA and TL: Show the percentage of students with a grade over all groups.
- The user could be able to view weeks before groups instead of weeks before groups, by swapping some setting. (TA/HTA/TL)

Won't haves

- The system won't have it's own authentication system, rather as listed in the must haves, the system will redirect to SSO.

Non-Functional

- The system backend shall be implemented in PHP using the Laravel framework.
- The system shall use MySQL databases for the data storage back-end.
- The system shall use Bootstrap/NodeJS as JavaScript frameworks for the front-end.
- The system shall use Gitlab for CI/CD and version management.
- The system must cope with usage of teams of (20+) TAs and lecturers that monitor courses of approx. 450-500 students.
- The system shall have at least 85% meaningful branch test coverage.
- The application must be working on all major browsers such as Google Chrome, Mozilla Firefox, Opera or Safari.

Suggested requirements

- Have intermediate grades/rubric feedback exportable so that they can be shared with the students.
- Have open fields in rubrics to allow for notes on each section. (Should/could haves)
- Have parts of the rubric automatically entered based on git analysis/BuddyCheck.
- Give an simplified preview/overview of each students BuddyCheck history to show if there has been improvement.
- Offer traffic light levels to represent how problematic students are on a weekly basis, in case of a yellow or red signal the TA can specify further why the student got this signal. Head TAs and teachers would be notified in case a group has students with red or yellow signals. (Instead of having 3 levels, increase the amount of levels to 5 or 6 to have a clearer priority for solving the problems for head TAs and teachers) (Good suggestions)
- Allow TAs to add important links/integrations into the application for each group. (Gitlab repository, Overleaf) (Good suggestions)
- Show a summary of previous results in rubrics, buddychecks, using a graph.
- Allow grading work to be split among TAs in case a TA has less time in the quarter after the project. (Happens reasonably often, option to add TAs would be nice) (Basically assign more than one TA to a group to allow the other TA to grade that group)
- Have the option to remove students from groups in case they failed the course.
- Allow head TAs and teachers to deactivate problem cases after they have been resolved. (Could)
- Make problem cases on the group level instead of the student level.

- Have a place to write notes visible to all head TAs and teachers involved with the course.
- Allow personal evaluation or self proposed grades from students. (Perhaps taking input from BuddyCheck)

Use cases

Teacher/lecturer (TL) use cases

- TL wants to log into the system with their NetID using Single Sign On.
- TL wants to register attendance or notes per meeting, per group, and also on an individual level.
- TL wants to add interventions, such as deadline extensions or doing extra work, and then follow up on whether students lived up to these interventions by receiving a notification after the deadline.
- TL wants to see all groups, and have a list of the events within a group.
- TL wants to upload reports from BuddyCheck and git analysis.
- TL wants to create, edit or import rubrics for a course edition, these rubrics should apply to every group taking part in the course edition.
- TL wants to mark notes regarding personal reasons as confidential to prevent anyone unauthorized from seeing them.
- TL wants to post deadlines for TAs during the project.
- TL wants to filter the list of groups to find a group or a specific student.
- TL wants to export grades per group/per course from the application and import them to Brightspace/Osiris.
- TL wants to see grade statistics per course edition.
- TL wants to import rubrics as CSV[1] files instead of having to create rubrics inside the application.
- TL wants to view the percentage of students that have received their grade.

Teaching assistant (TA) use cases

- TA wants to log into the system with their NetID using Single Sign On,
- TA wants to register attendance or notes per meeting, per group, and also on an individual level.
- TA wants to receive a notification that deadlines for student-specific agreements passed.
- TA wants to import a list of students that are each linked to a group.
- TA wants to log the problems of the group or an individual.
- TA wants to receive a notification if a specific student stands out in BuddyCheck reviews.
- TA wants to see the deadlines imposed by lecturers.
- TA wants to filter his list of groups to find a group or a specific student.
- TA wants to see grade statistics per course edition.
- TA wants to view the percentage of students (in the groups they manage) that has received their grade.

[1] https://en.wikipedia.org/wiki/Comma-separated_values