

Motivation

(When learning a new programming language, quick feedback is the key.) Student should be able to get feedback in relatively short time, where it should be easy to see mistakes that were made and get relevant feedback from the teacher. A student should be able to submit code in an easy fashion and somehow run and debug the code. The results should be shown in a simple way, where the teacher can add comments and view the code that the students submit. Autograder fills many of the needs that teachers have when grading code. The system uses a webpage where all aspects of the course are managed. This includes grouping of students and grading, as well as reviewing the code of the student and giving feedback. The current frontend is created with a web based system and a server side that handles the logic. It completes many of the goals of the original author, however, there are some problems that needs to be addressed. One of the goals of the original author is to simplify the process of grading the assignments. The number of students in a course can range from a small class of 10 students to a big class of 30-40 and even 100 students. The system as of now works great with smaller classes, however, when that number increases, problems emerge. We will therefore revisit these parts of the system and improve on them. We will also try to foresee how the use of the system, and the userbase may change, so that the frontend can still work independently of how the backed is structured and its solutions. The system should work for both small classes and bigger classes. The frontend should not be the limiting factor when choosing a programming language. In addition, it should also be a fun and productive experience both for students and teachers, rather than limiting. Furthermore, there have been some thoughts about mobile solutions and other features of the webpage. Now that the system is maturing it will be a natural progression to think about mobile solutions. In the ever growing mobile-market, an increasing number of hand-held devices have been introduced to the market, and users will expect this to be integrated in the Autograder system. We will therefore have this in mind when creating the new Autograder frontend.

The advanced programming courses usually consists of multiple assignments with short deadlines. The quality of the feedback can vary depending on a number of factors. To help teachers and students, Autograder should have features for commenting and giving

feedback to the student. The original Autograder has been used on a number of courses at bachelor and master level. This enables us to talk to users about the current system, and ways we can improve it. There has been some personal experience with the system also, and thoughts have been made on how the new system can be improved. We propose a new Autograder frontend system as well as some backend solutions, to not only streamline the work flow for student and teachers, but improve responsiveness of the page.