IT2901 - Informatics Project II

IDI Open
Programming Contest System

Haakon Konrad William Aasebø
Håkon Gimnes Kaurel
Tino Hakim Lazreg
Filip Fjuk Egge
Anders Sildnes
Eirik Fosse

May 2014

Norwegian University of Science and Technology

Supervisor: Hong Guo

Foreword

Originally inspired by the Nordic Collegiate Programming Contest (NCPC), it has been held at NTNU every spring since 2007. The format is a five-hour contest with competing teams consisting of one, two or three contestants. A team of volunteer judges write the problems and answer clarification requests during the contest, while another team hands out balloons for each solved problem. Usually a rather hectic affair, it is extremely important that everything is well prepared. The number of teams is often more than 100, with the record being 162 teams in 2011

The contest system that verifies solutions is at the heart of the contest when it is in progress, and needs to be working perfectly at all times. The system must handle several submissions per second, while verifying that each one is correct and runs within the set resource limits. Submissions must show up on the high score list, and when problems are solved the team handing out balloons must be notified. In addition to this there were a lot of other functional requirements having to do with the bureaucracy of organizing the contest

A requirement was that new features could be easily added in the future, and the code was written with this in mind. The project will now become open source, and all programming contest enthusiasts will soon be able to request and implement their desired features

All aspects of this project have been pleasing and delightful for us. The team has exceeded all our expectations and their system will be used for years to come.

Preface

Before there were computers, there were algorithms. But now that there are computers, there are even more algorithms, and algorithms lie at the heart of computing. Designing a system for eager students to hone their skill in the heart of computing has been a true joy

Our group never wanted to settle for adequacy and mere requisiteness. For the past few months, weve taught ourselves a new programming language and framework and used advanced development frameworks - while tackling many social and technical conflicts.

We have ve proven how Ambition is a dream with a V8 engine, as Elvis Presley once said.

The group would like to thank our eager customers, Finn Inderhaug Holme, Christian Chavez and Christian Neverdal Jonassen for their time to meet us and provide constructive feedback. We also owe a big thanks to our supervisor, Hong Guo, for constructive criticism and reflections; without which, we would not ascertain the peak of our own potential

Contents

Chapter 1

UI Design

This chapter contains the choices made regarding the process of designing the front-end of the application, for a more technical approach see *System Architecture chapter 6*.

1.1 Design process

The user interface provided by the previous IDI Open system consisted of a simple web interface for reading news items, registering teams for contests, and delivering submissions. GentleIDI is intended to provide more functionality through its web interface, including but not limited to judge supervision(requirement FJ-11) and user management (requirements FC-01, FC-03 and FC-04). As a consequence we had two options available: reusing and extending the existing interface design, or creating our own design from scratch.

We chose to create our own design from scratch, while still trying to keep a similar placement of elements from the previous design. The customer expressed concern regarding how contestants would react to the transition from the old interface to the new one. With this in mind we started to create mockups modelling core elements of the website. Our initial drafts consisted of simple rearrangements of elements found in the old web interface.

Beyond our three initial mockups we tried a couple of "out of the box" approaches to our designs, but none of them met our standard and was rejected for either being too time-consuming to implement or too far from what our customer wanted. We had a meeting with our customer, where we showed our mockups, and what our thoughts on design had been so far. We wanted to make sure that the customer was on the same page as us, and that we were not moving beyond the scope of the project. Our customer was not very focused on the design aspect, but one demand they had was that they wanted the new site to have the same structure as the old one. One example of what this means is that the customer wanted us to keep the menu on the left side as you can see that the old system has in Fig 1.1. We agreed, because getting used to a new website can take time, so keeping the structure similar would ease the transition for our users. With this in mind we decided to go for one of our initial mockups, the rightmost one in Fig 1.2, because it had the same structure as the old page, and we personally favoured that design. As a result, most of the elements found in the old interface can be found in the new one, and the transition between using the two is reduced to a minimum.

The task had to be completed in time for milestone M-03, so our main concern was designing

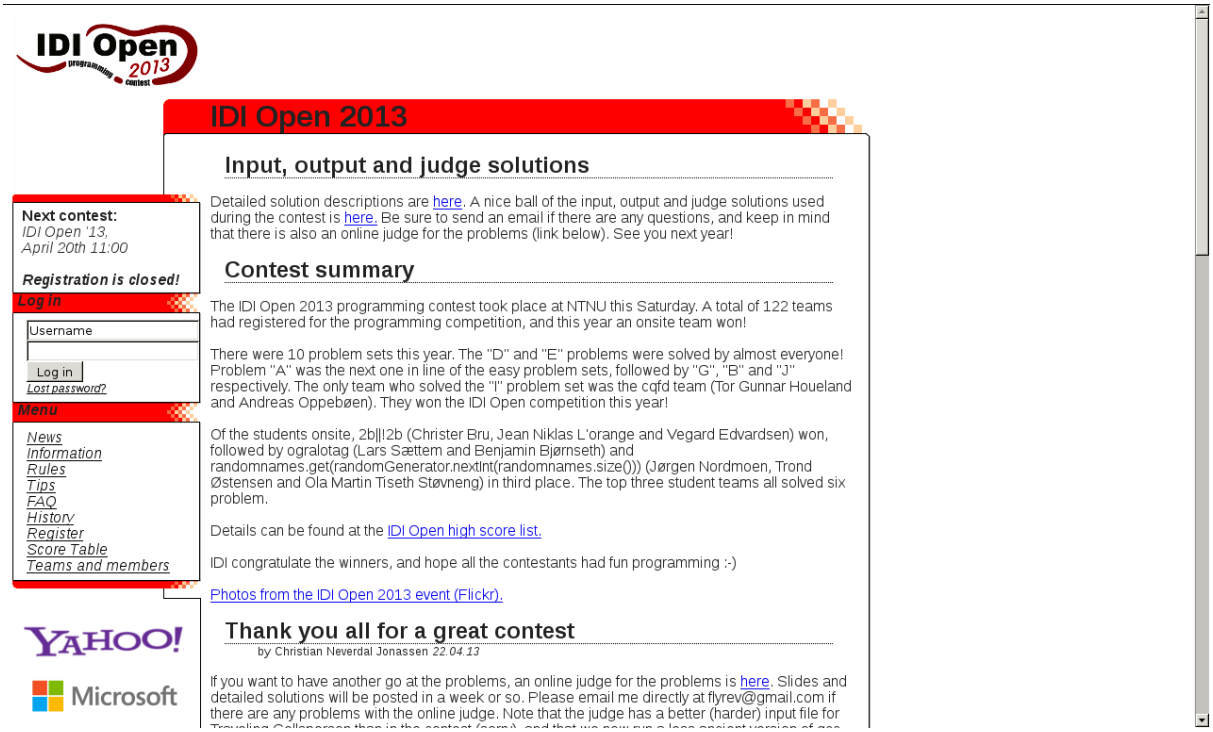

Figure 1.1: User Interface of the old system

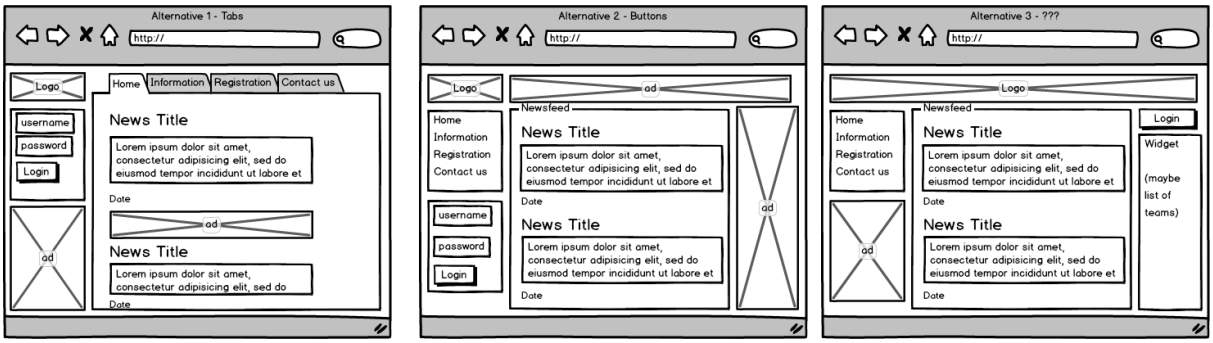

Figure 1.2: Initial mockups

for the functionality needed for that particular milestone. However, we also had mockups for functionality outside of this milestone. After milestone M-03 was met, we introduced new designs for new functionality through continuous work on top of a template.

The majority of the front end is stylized using bootstrap[Link til kilde] as a framework, enabling

us to create a site which is both highly maintainable and aesthetically pleasing at the same time. The admin interface was created using django-admin-interface. Grappelli was used as a skin to give it a modern look. The look of the final page can be viewed in Fig 1.3.

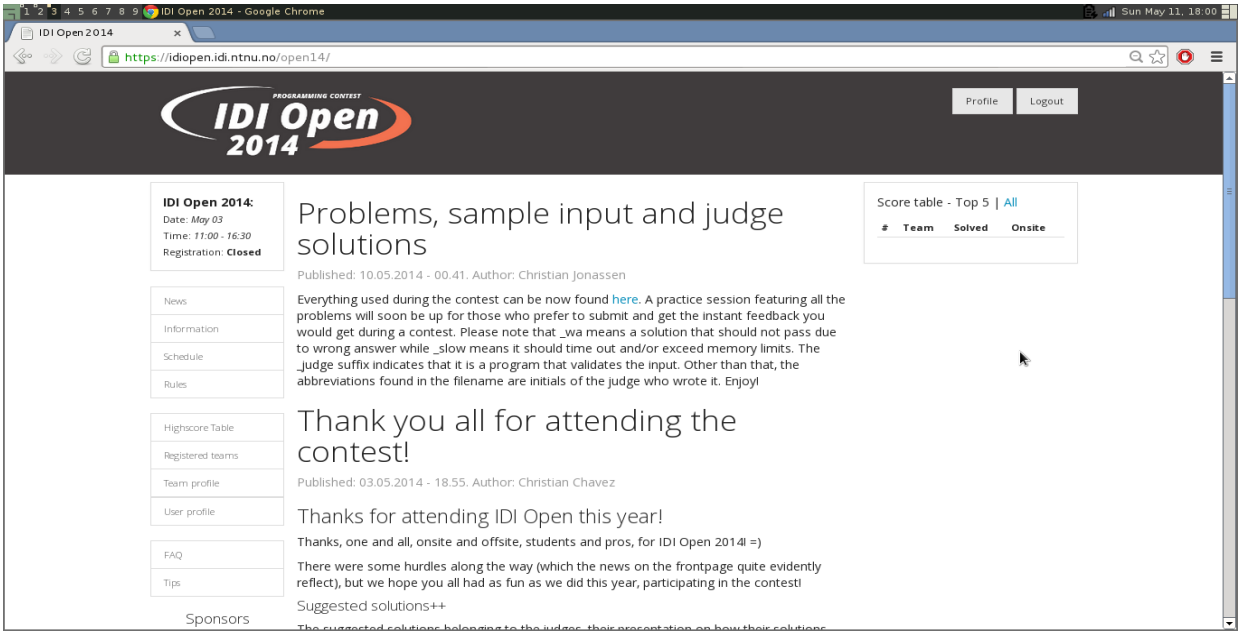

Figure 1.3: Final page

The grey header was in our initial design coloured blue, but was changed one week before M-07. This illustrates the strongest functionality of the design, namely customization. It is possible, by only uploading a new CSS file, to change the whole feel of the website and give every contest its own theme. The change from blue to grey was made as a consequence of IDI Open changing to a new logo. By comparing Fig 1.1 and Fig 1.3, you can see that we kept the same structure, but still made some significant changes to the design.

1.2 User interface

The user interface is designed by using a base template. The template is the same for every part of the webpage, and contains a content block that changes while you navigate through the different parts. This makes it easier to add new content to the user interface, because you already have the base, and don't need to worry about the header, footer, or the menu. We wanted to make it easy for future developers to take over GentleIDI after us, and therefore we focused on a versatile user interface, in case they want to add new functionality.

The menu is placed to the left, coping with the western norm stating that eye placement is natural to the left[1]. We designed the menu to be versatile, this was highly prioritized by our

[1] http://research.microsoft.com/en-us/um/people/cutrell/chi09-buschercutrellmorris-

Figure 1.4: Various buttons used on our website. From left to right: the button to go to the contest page, the button to see a user profile, the button to leave a team

customers. Admins can choose what they want to show in the menu, except for *Register user* and *Register team* that are "hardcoded" on request from the customer. As mentioned in Design process 1.1, we designed the user interface after a principle of versatility. Admins can also change the logo, the sponsor images and the contact information in the footer.

Buttons, images and icons were surrounded with boxes, to show that they are different elements. There is also one big box surrounding a group of elements, for example the sponsors. This is consistent with the gestalt law of proximity, that constitutes that humans will naturally group objects that are close to each other, and view them as distinct. This helps the user quickly understand the user interface.

"To strive for consistency" is the first of Shneiderman's eight golden rules of interface design[2], and we tried to follow this while making design decisions. As can be seen in Fig 1.4, we decided to use colours that represents the action each button is connected to. The red button marks that pressing this will have permanent consequences. We added a textbox prompt that the user has to answer after pressing a red button, that constitutes to Schneiderman's fifth and sixth rule, for easy reversal of actions and error handling. This wasn't added initially, but we noticed while testing the system that without a prompt, it could be possible to leave your team by mistake.

For the contest page, Fig 1.5, we wanted to give the contestant a good overview of all the problems, their submissions to them, feedback, if they solved the problem and the score. It is important to not bury information to deep in a website. It could be challenging to balance this while trying not to overload the page with too much information. We had this in mind when designing this page. We got valuable feedback from the customer concerning what they wanted to be present on the contest page. They wanted it to be easy for the contestants to access everything they need during the competition, through the contest page. After feedback from the customer, we added links to the clarification page and highscore table on the contest page. This lowers the short-term memory load on the contestants, which is consistent with Shneiderman's eight rule, because they will have everything accessible on the same page.

1.3 Admin interface

Django comes with an extensive admin interface, that provides functionality for adding, removing and changing parts of the system. The interface consists of everything we as developers want the admins to be able to change. We decided to use Grappelli, an app for the django admin interface that also provided us with more adequate functionality, e.g. auto-completion, rich text editors, drag'n drop and more.

eyetrackingforwebsalience.pdf

[2] https://www.cs.umd.edu/users/ben/goldenrules.html

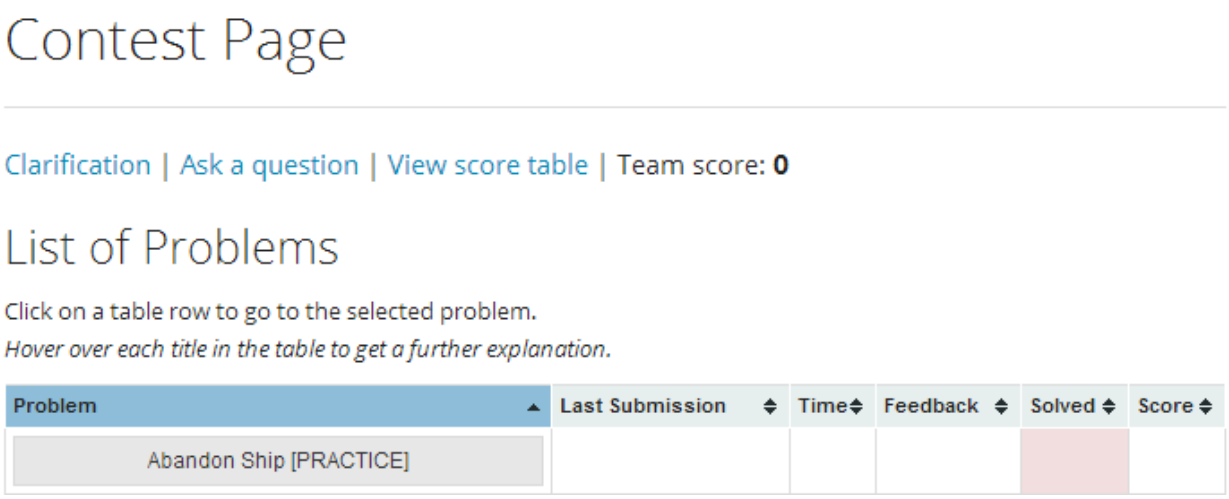

Figure 1.5: Contest page

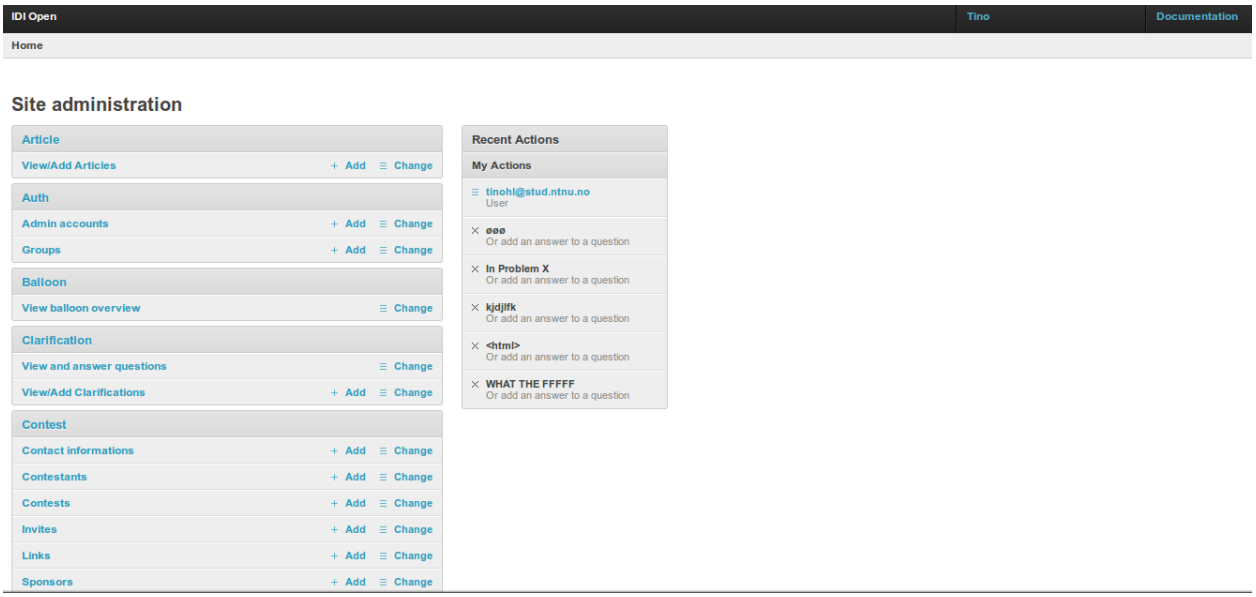

Figure 1.6: Admin Interface

The structure of the layout is simple. Each category has it's own header and everything in blue is clickable. The "Recent Actions" box is there to help admins remember what they last did, which is important to reduce the users short-term memory load, in accordance with Shneiderman's eight rule.

Originally all the names of the elements were the same as our model names. We decided to change this to more intuitively understandable expressions after a request from the customer. We extended the interface with our own custom views, "Balloon overview" and "Judge views". This allowed us to change what we wanted, while it still kept its consistency with the other parts of the admin site.

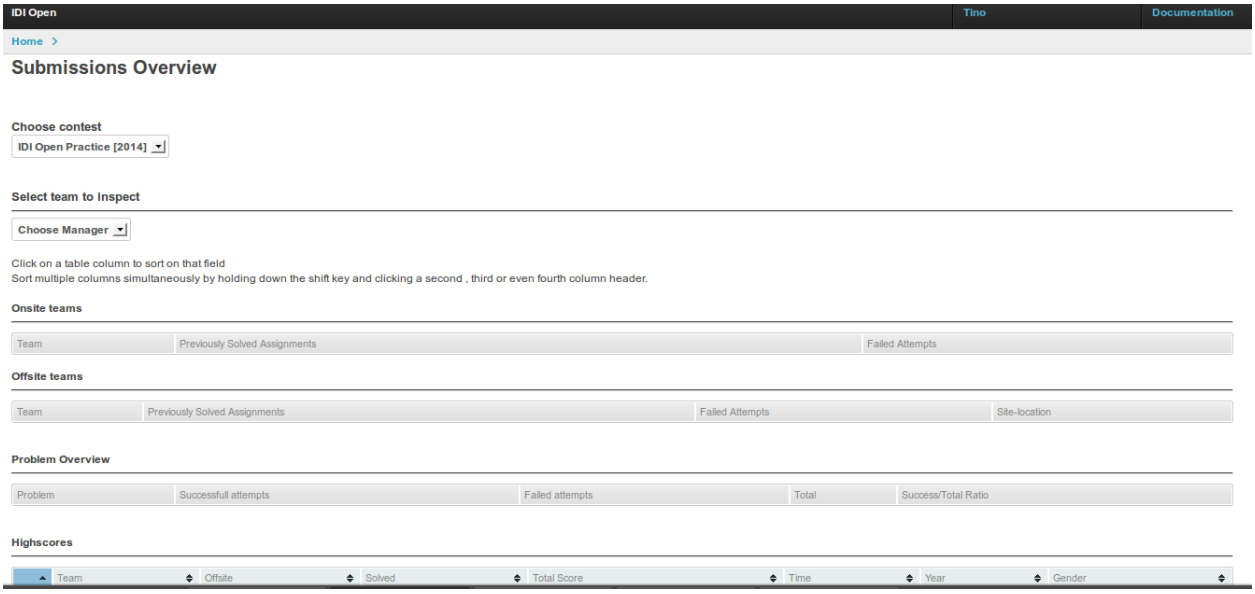

Figure 1.7: Judge views

The judge views was made primarily for judges, but could also be used by the admins. The motivation behind making this view, is that it gives the judges a better overview of the competition and how the progress is going for the different teams. We were initially told that the judges wanted a way to see if a team was struggling, so they could help that team. We wanted everything to be on one page for the judges, so they wouldn't have to constantly switch between different pages. The judge view can be seen in Fig 1.7.

Fig 1.8 shows the judge views after selecting the team "GentleCoding". It is possible to expand each submission by clicking on it. The third submission has been clicked on, so we can now choose to expand different categories. For example if a judge wants to see the source code for that submission, he/she can click on "Source code" and it will expand. Submissions that haven't been compiled are shown in red, and the other are white.

https://www.cs.umd.edu/users/ben/goldenrules.html

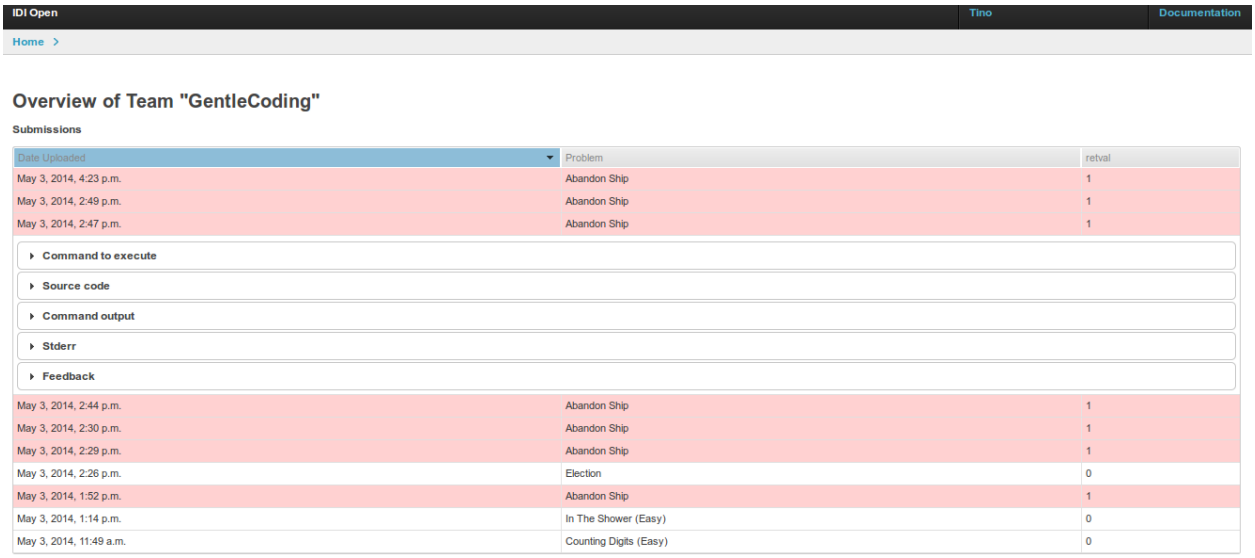

Figure 1.8: Judge views for team