University of Aberdeen Codemarker

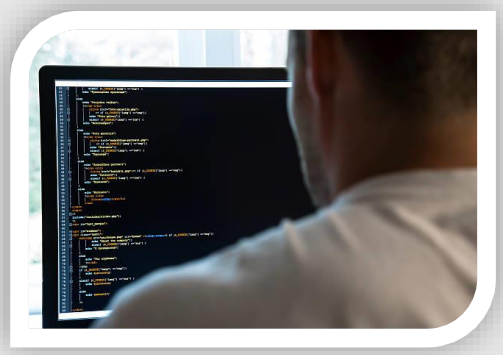
White Paper

~$ ./CODEMARKER.sh

TEAM ALPHA:

Konrad Dryja (Team Leader)
Stefan Rudvin (Deputy Team Leader)
Tautvydas Cerniauskas
Elzbieta Futkowska
Adrien Mainka

C:\> TABLE_OF_CONTENTS:



C:\> PREFACE:

Programming is a cornerstone of the computing science material supplied by Universities across the globe. The task of assessing code produces in these courses is often done by hand, which is counter-productive and inefficient.

We live in times of omnipresent automation and seeking release from manual labour such as copying and pasting code. Our team decided to develop a solution which automates professor's work of assessing code, and simply returns ready data to analyse or monitor students' performance. From now on this solution will be called Codemarker.

It is worth noting that there was an existing project in place, which is lacklustre in features and uses deprecated technologies. Our task is to improve the existing solution by using modern technologies which ensure full security and adaptability.

C:\> VISION:

Codemarker was developed to help Computing Science teachers with marking programming exercises. Since the inception phase we've focused on maximising accessibility and usability, making sure that nobody will have any problems with using the software.

Given an ideal scenario, a coordinator should be able to set up new assessments for his students, and check up on the results a week later to acquire their scores.

Major features that are core to the system:

- For students:
  - Submit code in multiple programming languages.
  - Access submissions history.
  - See obtained marks.
- For lecturers and coordinators:
  - Upload new assessments.
  - Extract trends based on students' scores.
  - Find out which students may be struggling with the course material.

C:\> COMPETITION:

When it comes to the competition of Codemarker, there is one site that seems to be its biggest opponent: Codio. It is a grading IDE connected to a cloud server with an integrated debugger of 5 programming languages. The IDE is a Virtual Desktop application that allows a teacher to communicate with students working on their own devices. However, Codio is a very expensive program that requires a fee for all students using the system. Also, a major drawback of this system is that the students are not graded immediately, while in our project results are shown to students instantly after code is submitted. The reason Codemarker is fast is because uses automated test cases to assess submissions.

On the other hand, in Codio's default way of marking assignments, the teacher still has to go through the code manually, which completely misses the point of an automated grading system. Another difference between Codio and Codemarker is that our system runs on University's server, which gives the institution more control over the system, and ensures the system's availability for all users. It also allows for direct transfer of grades to the university's database.

Apart from Codio, there does not seem to be any major competition for Codemarker except the existing system at the University of Aberdeen. Syntax checkers exist along with style and validation services, but they are not comparable to Codemarker as they do not check the outputs of submitted programs. This niche market allows Codemarker to be the first platform of its kind, and be attractive to many educational institutions due to its innovative approach to optimal grading.

C:\> BUSINESS_CONTEXT_AND_IMPACT:

A program like Codemarker will be appealing to many computer science teachers, as marking assignments and finding similarities between student's submissions tend to be extremely time consuming.

Codemarker could greatly improve the current system of marking programming assignments by saving the reviewers time, as well as giving a report to students immediately after they have submitted their work.

The web application can be implemented in the institution's own network, while offering a direct connection with university's database, and an innovative approach to teaching computer science. The client will be able to access all the aforementioned features and give system access to relevant students, as well as course and assignment coordinators.

C:\> COMMERCIALISATION_AND_MARKETING:

Codemarker is a program that can be implemented as a part of the university's network, which allows the institution to modify the system UI to integrate it with their own branding. A system administrator is able to grant access to course coordinators, who can add assignment coordinators who can create new assignments and add students to them. On first login, a guide will explain the use of the platform by submitting a sample program, creating a sample assignment and showing some sample results.

After initial testing, the platform can be implemented at the University of Aberdeen. As this will be Codemarker's first client, the team will obtain feedback from users for improvements that can be implemented during the testing phase. After receiving relevant commentary and adding appropriate updates to the system, an advertising campaign can be started. The heads of Computing Science departments in various universities will be approached by the Codemarker team who will present the finished system.

The educational institutes will then be able to access the system for free for a trial period of 1 year. After that, they will be presented with the possibility of renewing the licence for an annual fee.

As Codemarker is a start-up company, the development costs are low to allow a competitive price in comparison to exiting systems. The cost of hiring the team of developers (4 members) for a year is 87 360 pounds. That calculated at 14 pounds per hour (rate comparable to a university demonstrator) and 30 hours a week.

However, in order make the project possible the Codemarker team needs funds to cover initial expenses, such as the first year of development and the advertising campaign. Those costs will be covered either by a loan, crowd-funding, or by third party investors who will become shareholders of the system. All additional start-up expenses will be claimed against future revenues generated by the company. Nonetheless, the program will be in the development stage throughout the fund seeking process.

The annual fee of Codemarker system will be 4000 pounds. However, because of the free trial, the project will only be able to gain its initial revenue after the first year of having customers.

C:\> TECHNICAL_DETAILS:

Base Project and Reflection:

The solution currently in place was developed by 2nd year undergraduate students from the Aberdeen University Computing Science Department using Ruby as the base. Our team debated the possibility of expanding and upgrading the existing system, but after lengthy discussions we decided to abandon the idea mainly due to the following issues:

- Deprecated technologies – some frameworks and gems utilised became abandoned with no support what-so-ever leaving us stranded in case of any problems.
- Inexperienced developers – most of us are not familiar with Rails and its functionalities. Learning the new stack would consume more time and resources than the potential gain obtained by following the existing project.

Nevertheless, it served as a great base for our plans and architecture design.

Technologies overview:

Considering that University of Aberdeen promotes advanced Python programming teaching methods and making sure that students are most comfortable with writing code in it, we decided that Python based technologies were going to be our best choices.

Finally, we settled to use Django as our backend framework, as it is the most mature Python framework with a large community that ensures long term support for our application and the possibility of future expansion.

Django is also advertised as open-source with no fees, allowing our project to avoid additional costs in its development phase.

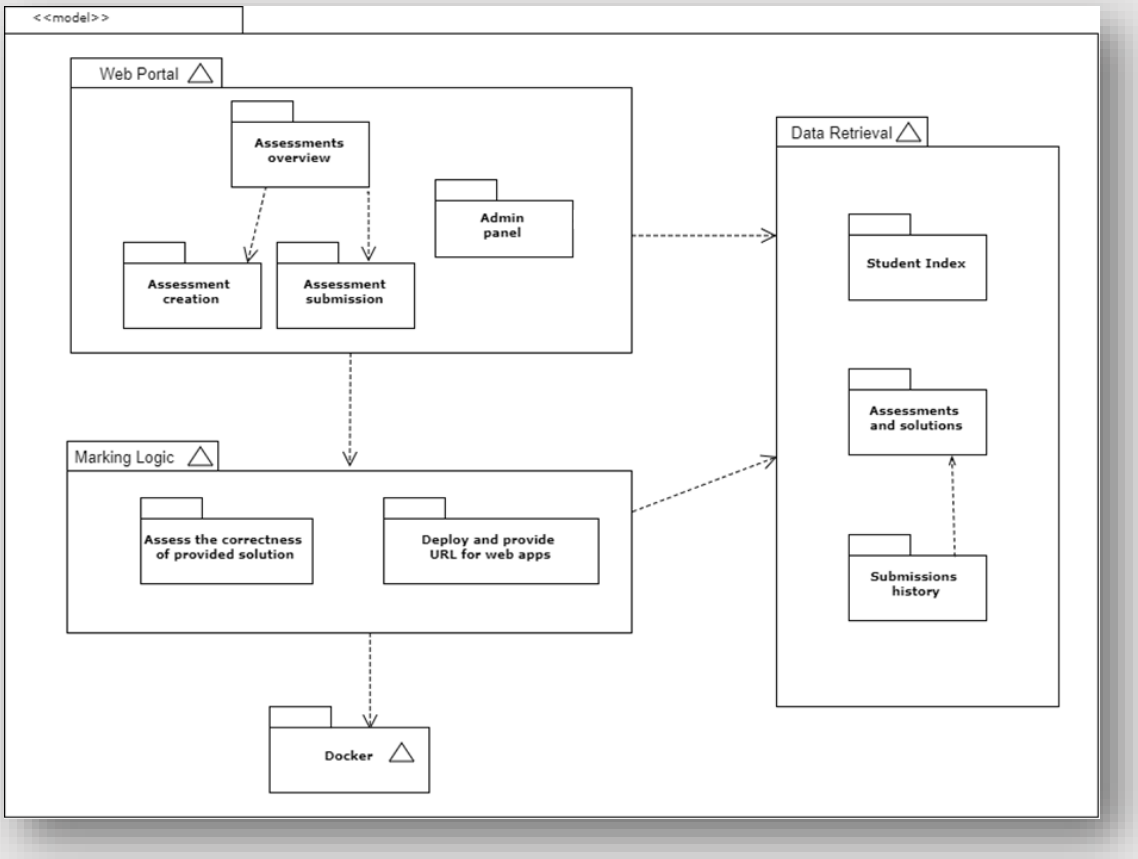
Although Django has a capability of serving as a full-stack framework we have decided not to use its presentation components in Codemarker. This decision was mostly dictated by the fact that it doesn't offer as much modularity and isn't as feature-packed as stand-alone front-end frameworks. We decided to use ReactJS for our front-end needs, since it is one of the most popular JavaScript frameworks with solid documentation and support.

Finally, having both front- and back-ends covered, due to a nature of our application we also had to think about isolation of the marking component from the rest of the application logic. It is exceptionally crucial to maintain full security

as the final goal is deployment in teaching environments and marking students pursuing their degrees. Running the code on the same machine and setup was out of the question as any malicious code could be run remotely. That's when Docker shines, and provides us with containers (later referred as docks) with specific software pre-installed on them ready to run and assess the code. The docks run in their own complete virtual system, preventing attackers from accessing host systems. It enables us to create docks for explicit tasks – for example deploying Rails applications or even projects developed by other CS3028 students – without having to manually prepare the environment. Docker also provides a handful of APIs which will be consumed by Codemarker to create and deploy assessments.

C:\> SYSTEM_ARCHITECTURE:

Our system has been designed with a three-tier architecture in mind, separating the presentation, business-logic and data layers to maximise security and usability:



As shown on the diagram above, Docker is a separate entity that does not depend on other components. It creates potential for modularity, since future expansion we may consider extending it to additional software capable of handling different types of assessment and structures. All-in-all the system has been designed with maintainability in mind.

C:\> REQUIREMENTS:

After discussing the needs with our client, the following requirements have been identified – expressed as user stories, which provide a starting point to our project:

Must have

1. As a student, I want to be able to submit my code in either Java, Ruby or Python so that I can obtain marks for my in-course work.
2. As a coordinator, I want to make sure that students fulfilled my assessment task, so that I can mark them accordingly.

Should have

3. As a coordinator, I want to create new assessments on-the-go so that I can monitor my student's activity.
4. As a coordinator, I want to know which students do not submit their work so that I can provide them with either help or C6s.

Could have

5. As a coordinator, I want to see which test fails the most so that I can tweak in the future and discuss specific topics during my lectures.
6. As a coordinator for AWAD, I want to be able to upload students' RoR apps so that they become instantly available for me to mark.

C:\> ACCESSIBILITY:

Our initial business plan aims deployment at various Universities across the globe. It is crucial to remember that students create a diverse group of users who need to be accommodated in case of any issues or problems. Apart from the pupils with disabilities, we also have to consider different cultures and environments – not only users from the English-speaking world.

In future revisions of our software we plan to include features such as high-contrast mode, voice-over for the most crucial components and translations into various languages. This approach would maximise user participation and overall satisfaction.

C:\> FROM_PROTOTYPE_TO_FINISHED_PRODUCT:

As Codemarker is still a prototype that is not yet ready to be officially released, there are many existing areas for further improvements in the system. The current pages and features implemented in the prototype are:

- Home page;
- Functionalities for course control:
  - Adding and deleting courses;
  - Displaying assignments per course;
  - Displaying last submissions per assignments;
- Submission page:
  - Viewing assignments (start and end date, description and additional information);
  - Uploading files for assessments;
  - Displaying report (time taken to run code, status, result and mark based on run time).

All of the above features are implemented in an intuitive and modern user interface. So far, the prototype supports only Python submissions, however, other programming languages will be available before the product launch. Besides that, the system still requires many enhancements before starting the user testing phase. The main features the Codemarker team is planning to implement in the near future are:

- Student accounts that can be assigned assignments by a course coordinator;
- Displaying the input and output for each submission in the report;
- More detailed feedback on failed submissions;
- Displaying last submissions per student in the admin page;
- Transferring assignment results to university's database;
- Sending e-mail notifications to students when they are assigned new task and when assignments have been modified;
- Support for multiple programming languages.

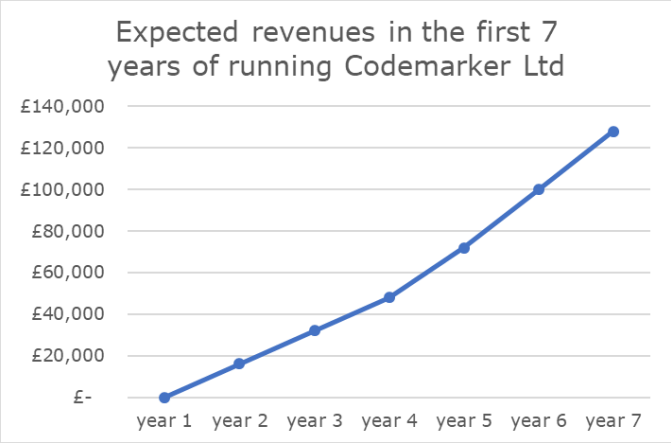
However, the Codemarker team will be keeping in touch with the computing science students at the University of Aberdeen using the current system, as well as Dr Nir Oren to identify other features that could improve the usability of the product. Furthermore, as the system grows bigger and increases its reach, it is necessary to improve its accessibility for users with various disabilities, as well as provide a number of translations, considering that the company is eventually going to target a global audience.

Nonetheless, the prototype is already a functioning system. Having such a strong foundation for further development means, the expansion of Codemarker is just a matter of time. Furthermore, this kind of basis allows the team to improve user experience without loss in the functionality of the system.

The next crucial stage of development is user testing. This evaluation of the system will take place at the University of Aberdeen, as the students already have experience working with the current software. The first stage of the process will be the comparison of learnability of the old and the new version of Codemarker. Test subjects for this step need to consist of people who have not used Codemarker before. Those will include various students, as well as computing science lecturers. In this step, testers will be presented with a task, for example "Create an assignment for a given course with given details" or "Make a submission for an assignment", and the time taken to finish the task for both systems will be compared. Afterwards, the testers will need to give feedback for both systems: Which one was easier to use and why? Which features made one systems better than the other? The second stage of will consist of testers freely using both systems. Similarly, to the first part, they will be asked to give feedback about both versions of the software: Which features of each system would you like to see being implemented in other ones? What difficulties did you encounter when using the system? What do you think could improve the usability?

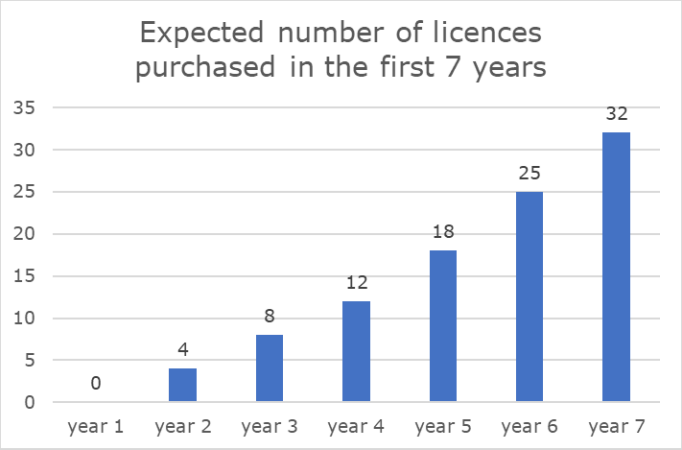
This kind of user test will result in more feedback on the current product and guide the team to further improvements expected from the audience. After adding, correcting and remodelling the system features, another user test can be conducted. This process of testing and improving will be repeated multiple times to ensure user satisfaction with the product.

Figure 1 Expected revenue based on the similar business strategies



After a series of user tests and additional improvements of the system, Codemarker will need to settle an e-commerce website on the Internet, from which the system could be purchased. Currently the website codemarker.com is not in use and is for sale from buydomains.com, however the price of the website is $4 388, which is out of the company's reach considering its low income as a start-up. A more affordable option is thecodemarker.co.uk from fasthosts.co.uk, which is currently £10.99 for 2 years. This website will provide information on the product relevant to clients and investors, contact information and an option to purchase the trial or full version of the product.

Figure 2 Expected number of users based on the market prediction



The development team is always open for feedback from users, investors and testers, which means it is always in the stage of development and will be updating along with incoming comments and criticism to constantly improve Codemarker's effectiveness, efficiency, learnability and ease of use.

C:\> BECOMING_A_COMPANY:

Before being able to generate income, Codemarker needs to be a part of a legal entity in terms of business. In this instance, the system will become a separate company limited by shares. Codemarker Ltd will need to have its own bank account, to which all the revenues earned will be paid. Through that the developers will be paid as employees and the shareholders will be issued their shares. Afterwards, the company will be set and the Memorandum of Association will be written to ensure any legal proceedings and tax disputes can be resolved legally. Lastly, when the VAT taxable turnover is more than 85 000 pounds, Codemarker Ltd will register for VAT with HM Revenue and Customs to obtain the VAT number, set the submission time for VAT return and payment and effective date of registration.

As a limited company, in order to raise capital Codemarker Ltd will sell shares to investors. The entrepreneurship is highly inspired by the current automated marking system at the University of Aberdeen and is developed with permission from Dr Nir Oren who is the current owner of the project. Considering those circumstances, The Codemarker Ltd will offer relevant royalties to its previous developers.

In the future, along with revenue growth and meeting the requirement of the 50 000-pound authorized share capital stated in Memorandum of Association, the company will enter stock market. Since the stock will be available to the general public and Codemarker's capital will be able to increase, the company will be developing on an even faster pace, as well as licence the product.

Furthermore, with more capital, the company will be able to expand its products and target other markets. It will be not only available for educational purposes, but will also create products for individual developers that would help with code optimization. Moreover, the company will be looking for possibilities to create alliances with similar types of businesses to expand its recipients and diversify its products. By growing its audience and polishing its products, Codemarker Ltd will be able increase its range and expand globally, eventually resulting in sustained growth in revenues.

~FIN

"The biggest risk is not taking any risk... In a world that changing really quickly, the only strategy that is guaranteed to fail is not taking risks." ~Mark Zuckerberg