

Online Game Store

Second Year Project (Computing - Software Development)



Final Project Report

Author: Ryan Deering

Ryan Deering – X00144631

Daria Denisiuk – X00145914

Chellie Delaney – X00144579

Supervisor: Eileen Costelloe

<https://github.com/ritadaria1/2ndYearProject/>

Table of Contents:

Class Diagrams	3
Chapter 1: An Introduction	5
Chapter 2: Iteration 1 Report – Chellie Delaney	5
Chapter 3: Iteration 2 Report – Ryan Deering	7
Chapter 4: Iteration 3 Report – Ryan Deering	8
Chapter 5: Conclusion	10
Acknowledgements:	10

Identification of Project Contributions:

Ryan Deering – Iteration 2 & 3 reports, intro and conclusion of Final Project Report, login system + BCrypt implementation, CRUD Admin + Customer, product image upload functionality, Discount system, Review system, Statistics class, logfile, testing, Change Password + Address + Profile Picture functionality in profile page, content filter implementation, Eircode regex method implementation.

Daria Denisiuk – Basket functionality, CRUD Product, stock management, Stripe implementation, image upload functionality, Email + PDF receipt, testing, all web design, product search functionality.

Chellie Delaney – Iteration 1 report.

Login Details + Discount Codes:

For an account of the Admin class:

Email: admin@cdrqgames.com

Password: *password*

For an account of the Customer class:

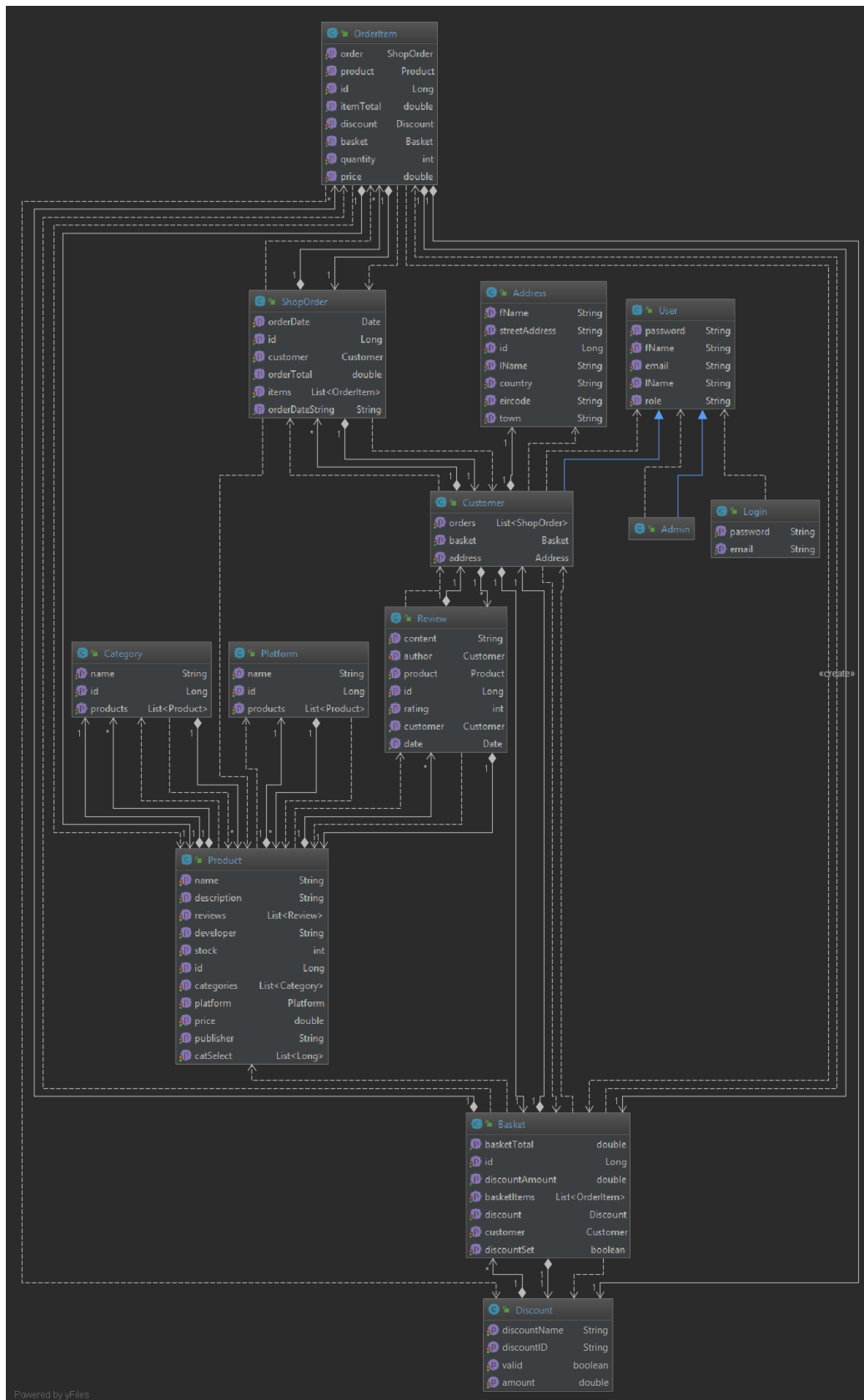
Email: customer@cdrqgames.com

Password: *password*

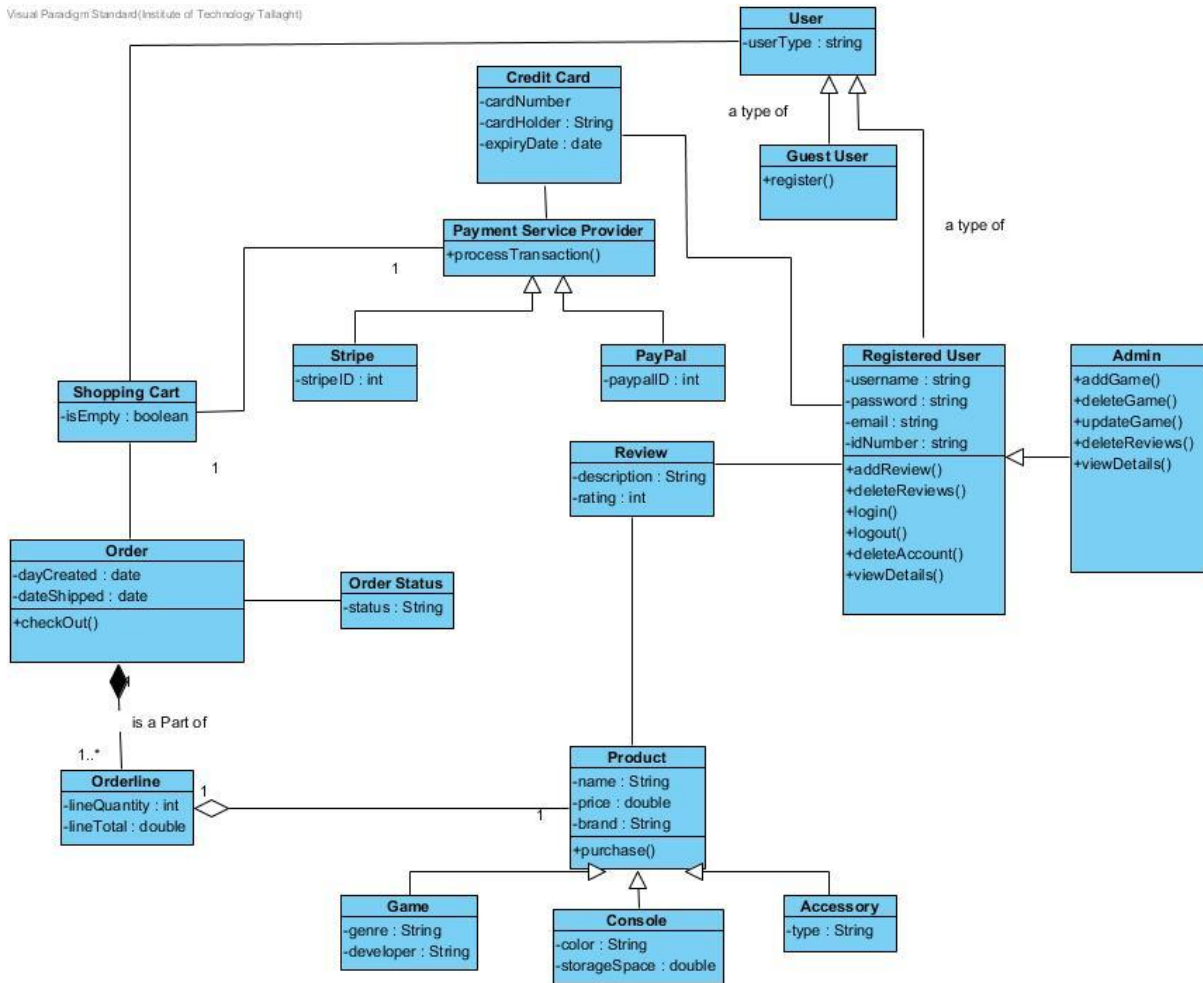
Discount Code: BIGSAVINGS Amount: 15% off Validity: true

Discount Code: CHEAPSKATE Amount: 10% off Validity: false

Class Diagrams:



*Class diagram of the model classes generated by IntelliJ IDEA Ultimate 2018.3.
Methods withheld for readability purposes.*



*Purposed class diagram of the project before development began.
Made using Visual Paradigm.*

Chapter 1 – An Introduction:

We began by brainstorming ideas and came up with the idea of an eCommerce video game shop, as a common interest of all project members was video games. We were taught Play Framework 2.6 using an eCommerce template, so we felt this would be the easiest route for development.

One of the goals of this project was to explore different technologies using Java and the Play Framework. We feel that this was achieved, as we used various libraries during development. The web design is eCommerce-driven, displaying a variety of different products to entice the customer into buying one, as well as advertising for the discount system. A lot of our design and programming matches our Project Plan & Requirements, with adjustments due to our use of the Agile Programming methodology.

Another goal we had in mind during design of the project was making the website look aesthetically pleasing and making it stand out from other projects. All the while keeping a functional, easy to use design. We feel we accomplished this during development.

Chapter 2 – Iteration 1 Report:

The features we implemented are:

- User and Admin registration
- Login and Logout functionality
- Product Database Functionality, E.G. adding, updating and removing entries from the table
- Basic Front-End Functionality & Design
- Search Products by Name
- Shopping Cart and Purchasing

We as a group first had to discuss the various features that we viewed as essential for the backbone of our project. The functions we decided on were basic user integration, the product database and the ability to access it, front end implementation, a search feature and finally the ability to purchase products. These were each functions that we decided as a group needed to be implemented first before any more specialised options were added.

The features each had to be implemented in a specific order to ensure they all worked well together and worked properly. User features were all added first. This was to ensure we could properly test the various additional features that had to be implemented in the later feature. To test this feature we created a simple front-end system that allowed users to login and showed a simple appearance change if the user was an admin.

The next feature added was the product database. This had to be added second as Admin had to be allowed to make changes to the various tables. The product database contains two tables, the games table and the accessories table. Users had to be able to view these tables and admin had to be able to alter the tables.

These features then allowed us to implement both the shopping cart and the search feature. The search feature has only had the most basic implementation, allowing users to search the product database by product name. While the shopping basket allows users to select a product and add it to their cart. They can then increase or decrease the quantity of the product they are purchasing. Users can add many different products to their basket and keep the basket saved between sessions. We also have the initial stages for purchasing set up, however it is not actually linked to any purchase system yet, this is for Iteration 2 with Stripe and PayPal.

We did find ourselves having issues with the Azure Dev Ops Repository It would only allow Ryan to commit changes to the repository, so we have had to move our repository to GitHub. We also found ourselves facing issues with setting up the VM at home. However, we have progressed quite well on our project and are confident in our ability to complete the project and add the various additional features we have planned.

When testing the various features we had implemented in our project, we took a more hands on approach rather than using automated systems. Due to the simplicity of the testing in the various systems we decided that it would be a more efficient use of our time to manually test each of the features ourselves. We found that most of the features were simple to implement successfully, however we did have the occasional issue with the addition of images to our products. This feature needed to be reimplemented several times due to issues in storing the images we wanted to add. We have overcome those issues and will keep an eye out on this feature in the future to insure no additional features cause any conflicts with the imaging system.

We hope to expand the Search feature, the Shopping Cart and begin work on the Comments system over the next week. While we know we will not be able to completely implement the Reviews system due to the lack of individual pages present for each product, but we feel even a simple page that allows us to add Reviews to show we have the system in place will allow us to more easily implement changes and create a small Reviews section in the future.

Chapter 3 – Iteration 2 Report:

To preface, Iteration 2 was not as chaotic as Iteration 1. The advancement of CAs and assignments and our necessity to complete them meant a slower pace of work compared to the blazing fast Iteration 1, in which we had implemented our basket, a basic search, the majority of our CRUD (including Registration) functions as well as a password hashing authentication system using the Java library BCrypt. However, this does not infer that Iteration 2 was uneventful.

We started with fixing our image upload. Daria had implemented this feature while I was working on the CRUD product functions. However, the ImageScalr based function was not cooperating with the rest of our code, after hours of debugging. Instead, we opted for an ImageMagick based function that worked flawlessly with the rest of our codebase.

After fixing image upload, we focused on implementing new functions for Iteration 2. I focused on stock management, while Daria focused on implementing PayPal & Stripe into our website as a way to purchase a product. On the stock management side of things, it was successfully implemented with customers being able to decrement and increment the number of items in their basket. I made a function that obfuscates the exact stock number from the Customer's view of the frontend, replacing the 'amount of stock' integer with a function that returns 'Out of stock,' 'limited in stock,' 'in stock!' as well as taking away the ability to purchase an item if it's out of stock. It should also be mentioned that Customers can also cancel their orders within the timeframe of an hour & view all their previous orders. PayPal and Stripe functionality was added, with Stripe working perfectly. However, Daria had trouble finding 'dummy' details to test PayPal so that feature is on ice while we find a way to test it. It will be ready in Iteration 3.

After the payment methods were complete, I went to create a page for every product. When you go into the frontend, there is a hypertext link on every product's name displayed, when you click on it you can go into it and it'll dynamically call elements in order to show you more details about it, such as who's the developer of the game, the genre, etc. This is not quite the case now, as the Product class needs to be extended with more values, for a full database for testing, but this will be properly implemented in the start of Iteration 3 as we gear towards polish, design and further testing. This uses some fonts from Google Fonts that we hope to use as we finalise design towards the end of Iteration 3, as they're aesthetically pleasing and will help the project stand out.

Daria implemented a receipt system, using JavaScript with a library called jsPDF, which allows for client-side generation of receipts via email. The customer's email is used, and the email is sent to them containing the contents of their order. The email contains a PDF that the customer can open.

Lastly, I have implemented a discount system. This uses a discount code that will be advertised on the website. There is a form on the basket page to pass the value of the discount code, as well as a Boolean check for validity as discount codes expire. If it is valid, it passed through and adjustments are made to the order depending on how much they save with the discount code. This is passed to their final order, where they are charged less depending on their coupon.

From a project management point of view, we've had a few problems with work distribution. We have only two members active in the repo, however there is three in the group. We are hoping to rectify this soon however for now, there are features specified in our design document namely reviews and an advanced search, that haven't been implemented as a result. Despite that, the project is progressing smoothly and well on track for Iteration 3.

Chapter 4 – Iteration 3 Report:

Iteration 3 wrapped up with a variety of new features and polish. I implemented the following: the discount system in full functionality, a review system for the product page, functionality for the profile page, a logfile in the root of the project folder and a Statistics page. Daria implemented printing a PDF and emailing it to the user's email and the entirety of the project's web design.

Continuing from the Iteration 2 report, I had to fix a few problems with the discount system. But it works perfect now, with discounts being applied correctly providing you enter the discount code at the basket page. However, if a discount is not valid, it will not work and will kick you back to the basket page. The review system is most likely the biggest addition of Iteration 3 however. Customers can now add their own reviews to a product regardless of whether they have bought the product or not (this is very similar to a lot of ecommerce websites behavior, as the customer could've bought the product elsewhere.) They can rate the product 1-5 stars, taken in via a numerical value at the review page for the product as well as a description limited by 512 characters. Upon submission of the review, an average rating is calculated and displayed on the product page. A customer can only a review product once, and cannot edit or delete their review. However, an admin can delete their review for them. Upon submission of the review, there is a check for derogatory language. This is done via a publically available CSV of swears in various languages. However, we have made sure to add a few non-derogatory words in order to show the effectiveness of this function in a professional situation, i.e. the final project presentation. If any derogatory language is found, the customer booted back to the review submission page and asked to alter their language.

Daria came up with a concept for a profile page, although initially scrapped early in development, we decided to implement it as it was proven to be feasible without eating too much time or effort into other parts of the project. She came up with the design, while I provided functionality. This is in the form of the following: a profile picture upload, the ability to change your address (this is crucial for the purchase process.) and the ability to change your password. There was some difficulties with the image upload due to misunderstandings of how Play Framework 2.6 handles forms on a webpage. However, this was quickly rectified.

Our supervisor, Eileen Costelloe, came up with two suggestions for the development of the project, as we were far into development at the start of Iteration 3. She asked us to implement a log file & a Statistics page for the admin to gain insights into how the website is performing economically. I proceeded to implement these, starting with the log file. To do this, I created a static method in one of our Controller classes, that takes in a string of what a function is doing. For example, in the registerUser method, there is one of these. I have put them in major methods that alter the database, as a logfile is usually used for security purposes. There is an example of how the log method works on the next page.

For statistics, I used an open source library called *JFreeChart* to graph up a pie chart based on what video game platform is selling the most games. Video games are often on different proprietary platforms different from each other and made by different companies, so therefore it is important to distinguish this from a statistics point of view. Demonstrated also on the statistics page, are methods to show the most popular game bought regardless of platform, the number of Customers in the database, the most popular discount used in the database.


```
HomeController.log("added a product called" + newProduct.getName());
```

```
1. public static void log(String activity) {  
2.     User u = User.getLoggedIn(session().get("email")); // gets User object  
3.  
4.     try {  
5.         DateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");  
6.         Date date = new Date();  
7.         String content = (dateFormat.format(date)) + " | " + u.getfName() + " "  
8.         + u.getlName() + " " + activity + " under role " + u.getRole() + "\n";  
9.         Files.write(Paths.get("logfile.txt"), content.getBytes(), StandardOpenO  
10.         ption.APPEND); //writes to txt file in root folder  
11.     } catch (IOException e) {  
12.         e.printStackTrace();  
13.     }  
14. }
```

Output of logfile:

```
16/04/2019 19:41:52 | Ryan Deering added a product called New Game under role customer
```

Daria began to implement the web design, in order that the frontend looks a lot more aesthetically pleasing so it is suitable for presentation, not debugging. She used other websites that sell games as a template for her design, combining it with intuitive, friendly design so it's accessible to the customer. She implemented a Contact Us page, that send messages from customers to an email that also handles receipts being sent.

She also worked on receipts during this iteration, using a free library called jsPDF which gets the information of the order in JavaScript and returns how much is spent, what it's spent on, as well if there is a discount used. As well downloading a PDF, a feature of Play Framework called Play Mailer is used to send emails through Google's SMTP server, to send a receipt containing the customer's address, how much they spent on their order, what games are in their order and if a discount code is used.

We made use of JUnit 5 as well as the Selenium IDE for our testing of the web application. Both were intuitive to work with and incredibly useful once our final web design was completed, so Selenium would not confuse certain elements of a web page with others.

And that concludes Iteration 3. I put the project on a Raspberry Pi running DietPi and hosted a web server for friends and family to attempt to break the project, which resulted in extra exceptions being put in place for when user input is being asked. It was satisfying to complete a project of such scale, and to exercise creativity in programming.

Chapter 5 – Conclusion:

This project was designed with a three-person team in mind, but during development this was narrowed down to two owing to unforeseen circumstances. As a result, development was more difficult than anticipated, due to a greater workload on two participants. Despite this impediment, we are confident that our project meets the high-quality standards not just set by ourselves, but by our lecturers and supervisors as well. SCRUM meetings were arranged every few days to keep track of our progress and where we were on the project. However, sometimes it was difficult to arrange these in addition to our usual time schedule and modules. Despite this, I feel we communicated well during the course of development. We worked together to make sure our code did not collide or conflict with each other's, for smoother development. We initially had some trouble with Azure DevOps and opted to use GitHub instead for actual version control. We still used Azure DevOps for project management, however. We feel that we used every ounce of what we've learned in the course thus far to complete this project and it was greatly satisfying to put everything we've learned into practice.

All the core functionality we drafted up in the *Project Plan & Requirements* was achieved. Unique features we set aside to differentiate the project from others, were a discount code system as well as a review system. These were both implemented in the project. There were some difficulties with the discount system owing to design troubles, but this did not last long and were fixed relatively quickly.

If we had more time, I would've liked to implement managing orders from the admin panel, so that the admin can cancel a customer's order or modify it as well change the shipping status of the order. I would've liked to get PayPal working as a payment method along with Stripe, however the API was too complicated for a reasonable implementation in the short timeframe we had for the project. Additional polish on the frontend would be a bonus too. However, these are the only things I would change. The project that was developed conforms mostly to the spec we designed ourselves, with the limitations we were put under and we are proud as of a result.

Acknowledgements:

I would like to offer my special thanks to my girlfriend Caitlin, for helping test the project, as well as listening to me vent my frustrations when things went wrong, as well as provide feedback on the project. I would like to acknowledge the support provided by my family during development of the project. – Ryan Deering