DC1FTS: Final week evaluation exercise

Purpose

The purpose of this document is to specify an online project that the students on the BSc Digital & Technology Solutions will complete as part of their initial training with Aston University.

1 Introduction

1.1 Guidelines

Please read this document in full before you start any work and make sure you understand what is required.

The following general instructions apply:

Time:	You have FOUR days (Monday to Thursday) to work on the given task. Try to complete as much of the project as possible in the time available rather than aiming to finish everything.
References:	You can consult your notes, seek online help, copy and edit previous code as needed. However, we expect the work you submit to be your own.
Questions:	You can ask the module staff for clarification of any issues and for specific technical help.
Backup:	Make a copy of your work each night to a USB drive or similar!

1.2 Teams

Individuals will be required to work in teams for the purposes of this exercise. The composition of the teams has been given to you.

1.3 Schedule

The work schedule for the activities described in this document is as follows.

Day 1 (Scoping and Organising)

- The module facilitators will deliver a briefing about the exercise.
- Teams will be given time to prepare for a 'client meeting'.
- The 'client meeting' should contain a material to explain how your team plans on delivering the project and organising the work.
- Teams will have an opportunity to ask the client for clarification regarding aspects of the requirements.
- **Milestone**: You team should complete and demonstrate the first **M** requirement in a short presentation at the end of the day.

Day 2-3 (Development and Testing)

- Teams will hold a Scrum meeting (or similar; no more than 15 minutes) twice a day, during which team members will report on previous progress and will be given new tasks.
- Teams will also have access to the client for clarifications.
- Teams will be expected to demonstrate their progress to the client.

- **Milestone for day 2**: Your team should complete and demonstrate the next three **M** requirements in a short presentation at the end of the day.
- **Milestone for day 3**: Your team should complete and demonstrate the final **M** requirements and first two **S** requirements in a short presentation at the end of the day.

Day 4 (Final Development, Testing, and Demonstration Preparation)

- Twice-daily Scrum meetings and access to client will continue.
- Teams are expected to balance work between completion of tasks, checking of deliverables (code, release notes, etc.) and preparation for the final presentation.
- **Milestone**: Your team should complete and demonstrate as many of the remaining MoSCoW requirements as you manage to complete in a short presentation at the end of the day.

Day 5 (Demonstration and Retrospective)

- One hour will be available for finishing touches and final preparation for the demonstration.
- Solutions shall be handed in by 10:00am on Friday.
- A complete submission must contain all of the below:
  - A zip archive with the project source code in full (all Java and SQL source files along with any other code necessary to run the application).
  - The presentation slides. If your class diagram, ERD, or component diagram are not included in the slides then they must be included separately in your submission.
  - A short user guide explaining how to install/set up and use the application. Any known bugs should also be documented and explained/justified here.
  - A reflection document including answers to the questions suggested in section 3.3 below, as well as any other considerations regarding your teamwork experience.
- **No late submissions are allowed**. If any of the items described above is missing, or submitted late, it will not be marked. If any source files are missing, this preventing the assessors from running the project and replicating the claimed results, the project will not be marked.
- In the final presentation and demonstration, each team will be allocated time to present, followed by time for Q&A. See section 3.4 for more details.

2 Project Scope and Requirements

The project is to create a new web-based outlet for the RentATree company. RentATree is an environmentally friendly enterprise that buys various types of potted Christmas trees from wholesale nurseries and rents them out to customers, for given amounts of time, over the winter holidays. Recyclable artificial trees (unlike retail ones, which cannot be responsibly disposed of) are available as well. When customers return their trees, they become available to be rented out again. When natural trees grow too big for their pots, the company plants them outdoors, as part of a reforestation programme. When necessary, artificial trees are recycled responsibly. Although RentATree caters to individual households also, their customer base is mainly formed of businesses, such as department stores, banks, city councils and restaurant chains, likely to rent several trees, of different types and heights, to display in offices, branches and public squares.

2.1 Outline

RentATree want a website that offers information regarding their potted trees and allows people to place orders all year round. Trees may be rented for a specified amount of time (no less than three days), between the 1st of December and the 14th of January of each year.

The interface should ensure that visitors (not necessarily logged in) of the site can easily find the following information for any kind of potted tree available:

- The type of tree (fir, pine, spruce, cedar, etc.);
- The material of the tree (natural, PVC[1], PE[2], etc.);
- The height (approximate, in centimetres);
- A description (containing details about tree appearance, scent and other features);
- The name of the supplier;
- The daily price for renting the type of tree, given the material of which it is made;
- The stock level for each day (1st of December to 14th of January) - updated live.

Website visitors (not necessarily logged in) should also be able to:

- View the entire RentATree stock, with the information mentioned above displayed for each tree item;
- Search for specific tree items (based on type, material, height, price, supplier or any combination of these features);
- Select the required number of trees of the desired type (e.g., 2 PVC cedar trees), within the limits of the available stock, and add the items to their basket;
- Specify the lease start date and end date for each item added to the basket;
- Select a transport option for collecting and returning each item in the basket: directly from/to the warehouse or via the company's delivery/pick-up service;

[1] Polyvinyl chloride, a type of plastic
[2] Polyethylene, a type of plastic

- Choose a transport slot (am/pm) for collecting and returning each item in the basket: the whole day incurs no extra cost, whereas a preference for either am or pm is charged with £3.99;
- Remove items from basket;
- View the basket.

Only logged in users may finalise an order by paying for the contents of the basket. Visitors (without an account) lose the content of their basket when leaving the website. Logged in users have access to their basket even after leaving and returning to the website. When finalising an order, users pay a deposit for each individual tree in the basket (the amount depends on the type and material of the tree), which will be reimbursed when the tree is returned. If a customer has successfully collected and returned at least three trees and has never missed a collection/return appointment, further orders will no longer require a deposit. If a customer has failed to collect or return one tree, the deposit will be reinstated at the next order, regardless of the number of successful collections/returns. If a customer has failed to collect/return two trees, a warning email will be sent and the deposit will be doubled. If a customer has failed to collect/return three trees, that user's account will be terminated and no other orders from that customer will be allowed.

Since RentATree is just starting, a "rent one, get one half-price" sale is on. For instance, if the basket contains one natural fir tree and 2 PVC fir trees, the customer will pay full rent for the natural tree, full rent for one PVC fir tree and half-rent for the other PVC tree (as plastic trees are cheaper to rent than natural trees).

At the beginning of the lease, customers may collect the trees from the RentATree warehouse or have them delivered to their home address. At the end of the lease, customers may return the trees to the warehouse or have them picked up by the company's transport service. Combinations are allowed, e.g, one item in the basket may be home delivered and returned at the warehouse by the customer, while another item in the basket may be picked up from the warehouse by the customer and returned by way of the company's transport service.

The system should keep track of the customers' purchase history, in order to make relevant suggestions - e.g., newly arrived trees supplied by a nursery a customer has bought from before will be recommended to that customer.

Administrator accounts must be made available for RentATree staff. Once logged in, administrators manually decrease stock when natural trees outgrow their pots and are planted or when artificial trees wear out and are recycled. Administrators manually increase the relevant stock when trees are returned by customers or new batches are delivered by suppliers. Administrators also create new records for new types of trees and delete the records of trees no longer supplied. Note that the system should decrease stock automatically when trees are ordered, without administrator intervention. For each customer, administrators update the number of "hits" (trees successfully collected **and** returned) and "misses" (trees unsuccessfully collected **or** returned).

RentATree is undertaking a parallel re-branding exercise. This will not be completed in time for you to put in place the new design before your deadline, so you must instead ensure that the site is clean-looking and easy to use, and that it should be simple to adapt it to the company's eventually-chosen style.

It is important that the new application is available for user acceptance testing by the deadline (Friday).

You do not have to organise web hosting in order to complete this task: a website running on localhost is acceptable. Use the Friday afternoon web development tutorials to guide you through this aspect, and through the technicalities of GUI development.

Alongside the website functionality, RentATree wish to build a strong case in favour of their business as opposed to traditional alternatives, involving cut trees disposed of at kerbside. To this end, they have also commissioned from you a study regarding the benefits of renting live trees over buying felled ones. They are also interested in the disposal of natural trees as opposed to throwing away artificial ones and the impact that discarding trees has on the environment. They are interested in your advice, based upon your analysis of available data, regarding ways in which they can use the new website to popularise the issues above. RentATree has suggested the following sources, but you are free to find and use others if you are aware of something more appropriate or useful:

- `http://www.telegraph.co.uk/gardening/gardeningadvice/11259120/Christmas-trees-are-they-really-sustainable.html`
  and
  `http://www.bbc.co.uk/news/uk-england-38129835`
  On the sustainability of natural Christmas trees as opposed to artificial ones.
- `https://www.envirowaste.co.uk/blog/articles/christmas-waste-statistics-making-christmas-green/`
  The carbon footprint of Christmas products across the UK.
- `https://www.theguardian.com/environment/2014/jan/03/christmas-tree-recycling-london`
  The cost of Christmas tree recycling in London.

You do not have to produce a separate report on this material. Your final presentation should contain concrete suggestions regarding the best way to incorporate the results of your data analysis on the RentATree website. **Make sure to appropriately acknowledge all data sources used in your report**.

2.2 High level tasks

The site will let users browse tree categories in stock, search for specific tree attributes, select a number of trees to rent over specific time intervals and finalise their order by paying for the contents of their basket. The back-end model should support all of these actions, along with any others arising from them, or from the requirements.

You will need to:

- Design and implement the underlying database:
  - Determine the entities and attributes within the requirements, along with any other data necessary, and identifying that data which needs to be stored within the database.
  - Design and implement the appropriate, normalized, table structures to manage that data.
  - Design sample data to allow the structure and application to be tested and demonstrated.

- Design and implement the back-end model to support operations unsuited for SQL (e.g., calculating the total cost of the basket):
  - Discuss the design of the class diagram in your groups based on the project specification (noun-verb analysis).
  - Discuss the relationships between the classes in your diagram (aggregation vs composition).
  - Analyse the way the back-end model will connect to the front end interface (how to integrate the JDBC API in your implementation).
  - Determine how to connect this middleware to the front-end interface.
  - Implement all classes within your model (start with class fields and then decide on class methods).
- Implement the front-end interface:
  - Users should be able to browse trees.
  - Users should be able to search for a particular tree, based on a combination of attributes, and have access to information about available stock.
  - Users should be able to select trees to rent out. Users may rent out multiple trees, of the same or distinct categories, each with different pick-up and return dates, in one transaction.
  - A login screen is required: users must be logged in to complete an order.
  - Screen(s) must exist to allow the user to see which trees they have selected and to allow the user to enter the details necessary to complete a purchase.
- Conduct your data analysis regarding the benefits of renting natural and ecological artificial trees, rather than buying felled ones and disposing of those irresponsibly.
  - Analyse several sources of reliable and relevant data (use those suggested by RentATree and also find others).
  - Look for relevant trends or correlations with the data and formulate a hypothesis based on that.
  - Test your hypothesis: consider further secondary sources, or interviewing other apprentices - consider the ethics and purpose of your questioning. Collect their feedback and see if your theory is confirmed or proven wrong.
  - Use your conclusions to suggest a list of concrete modifications in the algorithms, data presentation etc. implemented in the new site with the end goal of improving the site. Ensure each suggestion is backed up by the data.

2.3 Product features

You are expected to complete all the ‘Musts’ in the backlog. Note that a good quality back-end model is crucial for the efficient implementation of product features. If time permits, the solution should be extended to maintain the other features within the system. These features are detailed in the backlog:

What is MoSCoW? `http://en.wikipedia.org/wiki/MoSCoW_Method`

Feature/Story	MoSCoW
Login/logout screen or facility.	M
Browse trees in stock and search by a single (or a combination of) feature(s). Display search results.	M
Select trees, along with the rental duration, adding them to a 'basket'.	M
Place order screen showing order total and allowing users to place an order. Purchase screen requesting credit card details and allowing users to make the payment.	M
Produce a report with suggestions based on third-party data and statistics.	M
When an order is placed, the promotional deductions must be reflected in the total cost of the basket.	S
System administrators are able to update the stock and the customers' "hit"/"miss" data.	S
Users are able to perform a new single(multi)-feature search from the screen displaying the result of a previous search.	S
Users may select flexible pick-up and return options. The resulting fees as well as the correct deposit, given the "hit"/"miss" data, are reflected in the total cost.	C
When an order is placed, a file containing purchase details is available for printing.	C
Users receive confirmation emails after making purchases. Warning emails are sent out after two "misses".	C
A user registration screen captures new user information e.g. name, address, date of birth, credit card details and email address.	C
Registered users are given recommendations when they log in, based upon the types/suppliers of trees they've previously rented.	W

3 Guidelines and Hints

3.1 General Instructions

All the code should adhere to the guidelines for good practice provided. For example:

- Use software patterns and remember the core object oriented concepts you have learnt.
- Follow the standard naming conventions.
- Use clear and well-structured code.
- Use appropriate comments.
- Give meaningful names to variables.
- Have unit tests.

ImplementationPatterns.pdf contains a list of good coding practices and is available on Blackboard. This should be referenced and followed accordingly.

The finished screens should have a professional “look and feel”. It is your job to make sure that the screens present all the required data in an understandable and professional layout with appropriate help and messages as needed.

It is your responsibility to decide, within your teams, on the appropriate information to use for the database tables.

3.2 Development advice

Can you write a one-page document describing the scope of what you are trying to achieve? What are you doing (and equally important, not doing)?

How are you going to break the work down?

What questions do you need to ask?

Who is doing what?

Do not ‘gold-plate’, invent requirements or extend the scope beyond the existing specification.

How will you know when you are done?

Rewrite the requirements as stories. Convert the one line requirements into something clear with a goal you can test.

Your own implementation will be based on your own unique approach to the problem space. Pay attention to the time constraints in which you are working.

Think how you will communicate in the team. Professional solutions for agile development - such as Trello - are preferred over paper task lists. Code versioning managers - such as svn or git - are better than sharing code over Google Drive or Dropbox. **Note**: While Trello is straightforward to use, git and svn require some specific knowledge in order to use them to their full potential.

Think how you will communicate to the client at the end. As you design your solution, think how you are going to justify those decisions in the final presentation.

3.3 Reflection and evaluation

You should complete this element as a team at the end of the 4th day, by holding a retrospective, and include the answers to the questions below in your presentation. Please also include them electronically in your submission.

- Which bits of the exercise did you find easy/hard and why?
- What went well?
- What went badly?
- What would you do differently next time, in terms of process?
- Given more time what would you do to improve your solution?

3.4 Presentation instructions

Each team will have a total of 30 minutes to present their work. There will be an additional 10 minute slot allocated to Q&A. The presentation slot will be divided between two talks:

- A **presentation aimed at the client** will be delivered over the first 15 minutes. This should focus on the benefits of your solution to the client, and will describe the given task and your solution. Explain the project goals and demonstrate/justify the implemented features. The high level artefacts (class diagram, ERD and component diagram) may be included in the client presentation only if they are crucial to the narrative (e.g., used to explain solution design and product features). In that case, they should be explained in plain language, without excessive technical detail.
- A **presentation aimed at your managers** will be delivered over the remaining 15 minutes. This should include your high-level diagrams, demonstration of good working and coding practices (e.g. TDD, SQL procedures executed via `CallableStatement` instances to protect against SQL injection, appropriate exception handling, resource management etc.) and any other technical detail that you deem necessary to demonstrate to your managers that you are professional developers/data analysts.

There will be time at the end of the two presentations for Q&A. All team members are expected to talk for equal amounts of time during the two presentations, and all may be asked questions on any aspect of the system.

3.5 Interacting with the client

The module tutors will serve as technical consultants throughout project week. A client will be brought in to answer questions about project requirements and end features (client meeting timings are to be found in the module schedule on Blackboard) as well as to view demos from all teams for each of the daily milestones.

Take a professional approach when interacting with the client. Address him/her respectfully, ask confident and informed questions only after you have carefully studied the specifications (section 2) within your team and make sure to demonstrate how client feedback is integrated in your code from one prototype to the next. Take meeting minutes and share them with the client to get his/her approval. You can refer to the minutes when unsure about the way a feature should be implemented. Also, in the final presentation, you may bring up the client approved minutes if asked to justify a design decision, explain the presence/absence of a given feature, or discuss the client's reaction to aspects of your system.

4 Marking

Task		Marks
System design		25
	Java class diagram, ERD, component diagram	
Technical execution		50
	MoSCoW requirements, cohesion and low coupling, testing and exception handling, good coding practices	
Presentation		25
	Talk clarity and conciseness, slide structure and content	
Total		100

The project work submitted by each team will be assessed according to the marking scheme above, resulting in one mark for every team. If every team member contributed equally, individual marks will be the same as the team mark. If contributions were different (according to the fourth piece of the submission pack - see section 1.3, day 5), individual marks will be scaled accordingly. Please note that the maximum individual mark is the team mark (for example, in a team of 3, those who contribute 33% or higher will get the team mark, whereas those who contribute less than 33% will have their individual marks scaled down).