**SOFT8020 – Application Development Frameworks**

## Assignment 1 – 30%

## Due: 29th October 2017 10:00pm (end of week 7)

## **Overview**

This assignment assesses the following learning outcomes:

|  |  |
| --- | --- |
| LO1 | Configure a software project to use framework libraries. |
| LO2 | Design scalable enterprise applications with reusable components. |
| LO3 | Implement advanced data access technologies for a given requirement. |

Therefore, it will examine your ability to configure a scalable application, design it following good design practice to ensure reusability of components, and it will access a database – in this assignment, it will be a relational database.

## **The “Problem”**

Microtransactions are a popular way for game developers to increase revenue from games and to spread revenue over a longer period of time than just the sales peak around initial release. You are to develop a microtransaction system for a text-based adventure game called “Kylar’s Vengeance”.

The game’s currency is called the Kubit – players always get some free Kubits to begin with, can earn Kubits through gameplay, or purchase Kubits using real money (e.g. by credit card or Paypal). Each player begins with 1,000 Kubits and 3 pieces of equipment – a sword, a shield and basic armour, all initially at level 1 in terms of upgrades.

The player has 4 equipment slots: an up-close weapon, a distance weapon, a shield and armour.

The player can upgrade 1 piece of equipment per round of play by purchasing an upgrade. The player can also buy new equipment and sell old equipment, but only once per round of play (1 purchase and 1 sale).

Each piece of equipment has the same set of properties, e.g. damage inflicted, protection provided, and anything else you might think of. For example, at upgrade level 4, a sword could have damage of 40 and protection of 16 (e.g. you can parry with a sword). Perhaps a sword has a maximum damage of 50, while a crossbow might have maximum damage of 65.

The focus of this assignment is **not to create a game**, but to create the infrastructure for the purchase, sale and upgrading of equipment. You must decide what database tables you need and put in place a text-based menu system.

The initial menu could look something like this:

--- KYLAR’S VENGEANCE ---

Player: Jedi Jones  
Balance: 824 Kubits  
Equipment: Mace (level 3), Crossbow (level 1), Armour (level 5), Shield (level 4)

Main Menu

1. Finish Round
2. Buy new equipment
3. Sell equipment
4. Upgrade equipment

When upgrading a piece of equipment, you upgrade 1 level at a time, e.g. begin at level 1, then level 2, then level 3, etc. Finishing a round resets the ability to upgrade a piece of equipment, purchase one new piece of equipment or sell one piece of equipment in the next round.

## **Technical Notes**

This is to be a Spring Java project. You can elect to create either a Spring Legacy project (i.e. using primarily XML for configuration) or a Spring Boot project (using annotations). You must use either H2 or HSQL and load SQL schema and data scripts into an embedded database. You must use JdbcTemplate within DAO/Repository components / Spring beans.

## **Marking**

Consult the rubric in Blackboard for a breakdown of the marks and guidance on what is expected to score well under each category.

## **Extra marks**

Gain up to 5% extra by adding what you consider an extra feature. Highlight the feature in your submission and explain why you think it is worth an extra 5%.

## **Submission**

Submit via the e-assignment in Blackboard. You will need the following in your zip archive:

* A brief document outlining the high-level design of your system (1 or 2 A4 pages) + explanation of your extra feature
* Your Spring project, including SQL scripts for initial data in src/main/resources folder

## **Penalties**

Penalties for late submission are applied as per section 4.4.2 of the Exam Regulations (see <http://www.mycit.ie/examregulations>). Do not plagiarize – don’t copy code, don’t provide code, don’t work too closely with someone else on the project. Trying to be clever by taking someone else’s code and renaming classes and variables is not a good strategy and will likely result in a zero grade or worse.