**School of Computing, University of Dundee**

08

**Fall**

Course Work Cover Sheet - The School of Computing

**Author’s Name:** Kevin Shewan (110014571)

Scott Reid (110014567)

Heather Ellis (100009618)

Euan Luke (080007823)

Patrick Cowan (100011576)

**Module Code:** AC31007

**Assignment Title :**  Assignment 2

**Date Submitted:** 14.11.12

**Date Due to be submitted:** 14.11.12

|  |  |
| --- | --- |
| Your course work may be used by The School of Computing for demonstration purposes with other classes and/or members of the public, unless you indicate otherwise. Tick the box at the right if you do *not* wish your work to be used in this way. |  |

I certify that this assignment represents all my own work and that no parts of it have been copied and that no collusion has taken place with any other person. I have also read and understood The School of Computing’s regulations as regards course work and I certify that this course work was performed according to those regulations.

I certify that any disks submitted with this assignment have been virus checked and have no viruses on them.

Signed: ………….…..Kevin ……………. ………….…..Scott…………….

………….…..Euan……………. ………….…..Patrick …………….

………….…..Heather …………….

Table of Contents

Important Information 3

Project Planning 3

Conversations with the client 3

Date: 12.9.12 3

Initial meetings with the client (19.9.12 & 21.9.12) 5

Task Breakdown 8

Completed Tasks Requirements 10

Sprint Burndown Chart 13

Velocity 15

Sprint Backlog 15

[Insert Sprint Backlog] 15

Design Mockups 15

Evidence of conversations with client and the Scrum Master resolutions 16

Quality Assurance 19

User stories and Acceptance Tests 19

Quality Testing 20

Unit Testing 20

Black Box Testing 20

JUnit Testing 22

[Complete JUnit Testing] 22

Refactoring 22

[Complete Refactoring] 22

Metrics 22

[Complete Metrics] 22

Team Quality Assurance Standards 22

Paired Programming 23

[Complete PP] 23

Usability Score and Survey 23

[Insert usability score] 23

Acceptance Tests Approved in Sprint Review 23

[Insert Aproved Accepatnce Tests] 23

Source Control 23

[Insert failed commit image] 24

Lessons learnt from last sprint 24

Unbalanced workload share 24

[Complete worshare] 24

Team attendance could be improved 24

[Complete attendace] 24

# Important Information

Our user stories were stored on my.myagilityboard.com with the user details:

**Username:** unhandledexceptions

**Password:** group5

You can see how we used the business value and effort to give us an actual value for the user story, as agreed with the client.

We held scrums every day and each day there was a different Scrum Master which meant that the rest of us could focus on other tasks and didn’t have to worry about the issues for the day. The team tried hard to perform paired programming as much as possible which presented a learning curve to all members as it took getting used to. Furthermore, the team often worked together at a group of tables to improve on communication and team productivity while complying to the Agile practice. As much as possible a seven hour day was used but other modules and time constraints prevented it from being a perfect working day hour set. It was understood by the team that in the industry this would not be the case and that the team members would be able to better follow the Agile approach.

# Project Planning

## Conversations with the client

## Date: 12.9.12

**Discussed**:

* Had initial meeting with client and asked her questions to give more details on her initial requests. See below.
* The order of priority of each request was selected by the client and the user stories are now ranked in order of priority.

**Date:** 19.9.12

**Discussed:**

* The user stories were created and approved. Some changes needed to be made and were during the meeting.
* The wording of two user stories needed to be improved and one user story was an acceptance test for another. This was added as an acceptance test for the user story numbered 5.

**Date:** 21.9.12

**Discussed**:

* The design mockups were agreed with the client for the screens. These can be seen below. She seemed pleased and liked the designs, no further changes were required.
* The business value and effort was attached to the user stories. The team has now added the effort expected and worked out the actual value of the user story.

**Date:** 24.9.12

**Discussed:**

* Shares in alphabetical order
* Total shares’ values moved up higher
* Rocket or plummet in colour and with arrows
* Plummets and rockets layered down screen.
* Best: how much it has made relative to it’s original price  
  --performance as percentage of price  
  Worst: Ranking
* ‘Goldy’ Yellow arrow to side for run

**Date:** 3.10.12

**Discussed:**

* The acceptance tests were written up and approved by the client in a final check before the sprint began.
* The team proposed two user stories and one as a potential with the client approval and understanding that the third one was only a small maybe.
* The client understood that this selection was final for this sprint and that no more items could be added until the next sprint.

**Date:** 10.10.12

**Discussed**:

* The sprint review went well except that one share price was wrong. It was realised that in a last minute update there had ben a change of characters which meant that it was reading in the American prices rather than the British prices. This was fixed and signed off on.
* All acceptance tests were passed and signed off by the client who seemed pleased that all her requirements were met.
* The new user stories were then selected based on the team’s velocity and the acceptance tests have been approved by the client however a couple of changes are to be made and sent back for a final review.

**Date:** 6.11.12

**Discussed**:

* Did a final review of the acceptance tests and the design mockups before the commencement of the sprint. Client seemed quite pleased with the work of the team.
* There was a couple of minor improvements needed to be completed with regard to the design mockups.

**Date:** 12.11.12

**Discussed**:

* The design mockups were edited and submitted to the client for approval. These were approved with some minor changes made.
* The client is now pleased and knows what to expect in the sprint review. They have understood that they cannot make any more changes now.

**Date:** 14.11.12

**Discussed**:

* The sprint review went

## Initial meetings with the client (19.9.12 & 21.9.12)

1. **(5)An estimate of how much her whole portfolio is worth**
   1. How often does she want it displayed?
      1. *Only when she goes to that section of app*
   2. How does she want it to be displayed?
      1. *Just a single value and name*
2. **(4)An estimate of how much each set of shares is worth**
   1. How often does she want an update?
      1. *When she opens up the app*
   2. What currency does she want the app to have?
      1. *Local (GB£) based on what it is currently valued at*
   3. How is an estimate made?
      1. *Through working out what the current value of the shares are worth at that moment*
3. **(7)To know how much money she has lost or gained in the week**
   1. How does she want this displayed?
      1. *Just a single statement*
   2. How often does she want this to be updated?
      1. *Every time she opens the app*
4. **(2)An alert to be raised if any of the share prices plummet or rocket**
   1. What is a rocket?
      1. *10% increase*
   2. What is a plummet?
      1. *20% decrease*
   3. What market are we using?
      1. *London Stock Exchange*
   4. What type of alert does she want?
      1. *Badge icon and a noise when she opens the app*
   5. What details does she want shown? Price opening, Change in value? Money made/lost?
      1. *How much it has changed by only*
5. **(8)An alert to be raised if there is a run on any of her shares**
   1. What is a ‘run’?
      1. *When more than 10% of the shares are sold*
   2. What types of alerts does she want?
      1. *A notoficiation at the top of the screen*
6. **(3)A summary of the best and worst performing shares of the week**
   1. Overall or hers personally?
      1. *Just hers – no login required as personal app*
   2. What type of display does she want?
      1. *Just a single value for the best and a single value for the worst*
   3. Does she want just a snapshot of a particular time period, ie the end of the week?
      1. *No – see above*
7. **(6)Information about the weekly high and low for each share**
   1. What is a ‘week’?
      1. *Monday 0800 – Friday 1630*
8. **(1)To see the rise and fall in prices of the shares each day**
   1. How does she want the shares displayed? Graphs? Sections?
      1. *A grid like system*
   2. How often do you want to refresh the app?
      1. *When I open the app*
   3. What share information do you want shown?
      1. *The change in share price, no decimals*

**GUI/Other Questions**

1. **What colour schemes do you want? (Need for colour blind users)**
   1. *Plain and simple, based on share apps already existing*
2. **What type of menu do you want? Logo based, text based or swipe based?**
3. *Liked the scrolling menu*
4. **Do you want a ‘ticker’ on screen to scroll through the updates?**
   1. No, a simple box at the top will be okay
5. **Will you be the only user or will it be a shared app?** 
   1. *Only user*

## Task Breakdown

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |
| **Sprint implementation days** | 6 |  |  | **Effort** | **Remaining on implementation day…** | | | | | |
| **Trend calculated based on last** | 6 | **Days** | **Totals** | **109** | **99** | **92** | **76** | **41** | **24** |  |
| **Task name** | **Story ID** | **Responsible** | **Status** | **Est.** | **1** | **2** | **3** | **4** | **5** | **6** |
| Update task breakdowns | 3 | Euan | Done | 1 | 2 | 0 | 0 | 0 | 0 | 0 |
| Research how to implement a burndown chart | 3 | Heather | Done | 2 | 4 | 3 | 0 | 0 | 0 | 0 |
| Create the Sprint burndown chart | 3 | Heather | Done | 4 | 1 | 1 | 2 | 0 | 0 | 0 |
| Write up what the team classes as 'done' | 3 | Scott | Done | 3 | 4 | 0 | 0 | 0 | 0 | 0 |
| Design Mockup for the user interface element required to show client profit or loss. | 3 | Scott | Done | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| Confirm the designs with the client | 3 | Kevin | Done | 1 | 1 | 2 | 0 | 0 | 0 | 0 |
| Research options for UI elements and types of implementation | 3 | Patrick | Done | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| Implement code for the user interface element of the story | 3 | Kevin | Done | 8 | 8 | 10 | 10 | 7 | 5 | 0 |
| Write JUnit Tests | 3 | Euan | Done | 5 | 6 | 3 | 2 | 0 | 0 | 0 |
| Research options for calculation of profit or loss | 3 | Kevin | Done | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| Implement code for the backend calculation of profit or loss | 3 | Heather | Done | 8 | 8 | 6 | 2 | 0 | 0 | 0 |
| Perform black box testing | 3 | Patrick | Done | 5 | 5 | 6 | 3 | 0 | 0 | 0 |
| Review results of black box testing | 3 | Euan | Done | 3 | 3 | 3 | 3 | 0 | 0 | 0 |
| Implement any changes from the black box testing | 3 | Scott | Done | 4 | 4 | 4 | 4 | 4 | 2 | 0 |
| Update the report about testing | 3 | Heather | Done | 2 | 2 | 2 | 2 | 2 | 2 | 0 |
| Refactor Code | 3 | Kevin | Done | 4 | 4 | 4 | 4 | 4 | 4 | 0 |
| One page document on any public methods that the classes created may have and also the external classes that it uses | 3 | Euan | Done | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |  |  |  |  |  |  |  |  |  |  |
| Research options for calculation of the share price. | 4 | Kevin | Done | 2 | 2 | 0 | 0 | 0 | 0 | 0 |
| Design Mockup for the user interface element required to show client breakdown of share set. | 4 | Patrick | Done | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| Confirm the designs with the client | 4 | Heather | Done | 1 | 1 | 1 | 2 | 0 | 0 | 0 |
| Implement code for a calculation method for the share price. | 4 | Scott | Done | 5 | 5 | 8 | 5 | 4 | 1 | 0 |
| Write Junit Tests | 4 | Euan | Done | 5 | 6 | 6 | 3 | 0 | 0 | 0 |
| Assess the accessor method to get the data. | 4 | Patrick | Done | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| Store the data received in the database to be accessed later. | 4 | Kevin | Done | 5 | 5 | 5 | 5 | 0 | 0 | 0 |
| Research options for UI elements and types of implementation. | 4 | Heather | Done | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| Implement code for the user interface element of the story. | 4 | Patrick | Done | 5 | 5 | 5 | 7 | 3 | 0 | 0 |
| Implement black box testing | 4 | Euan | Done | 5 | 3 | 3 | 2 | 0 | 0 | 0 |
| Review results of black box testing | 4 | Scott | Done | 3 | 3 | 3 | 3 | 1 | 0 | 0 |
| Implement any changes from the black box testing | 4 | Kevin | Done | 4 | 4 | 4 | 4 | 4 | 2 | 0 |
| Update the report about testing | 4 | Heather | Done | 2 | 1 | 1 | 1 | 1 | 0 | 0 |
| One page document on any public methods that the classes created may have and also the external classes that it uses | 4 | Euan | Done | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| Refactor code | 4 | Kevin | Done | 5 | 5 | 5 | 5 | 5 | 3 | 0 |
| Write up final report | 4 | Heather | Done | 4 | 4 | 4 | 4 | 4 | 4 | 0 |

### Completed Tasks Requirements

|  |  |
| --- | --- |
| **Task Description** | **What encompasses done** |
| Update task breakdowns | When the team decides that the task breakdown is complete |
| Research how to implement a burndown chart | When the team can make changes to their tasks and the burndown chart created updates automatically |
| Create the Sprint burndown chart | When the team can make changes to their tasks and the burndown chart created updates automatically |
| Write up what the team classes as 'done' | When the teams agrees on the ‘passing crtieria’ for each task |
| Design Mockup for the user interface element required to show client profit or loss. | When the team agrees that the mockup complies to the user story as agreed with the client |
| Confirm the designs with the client | When the client agrees that the mockup complies to the user story as already agreed |
| Research options for UI elements and types of implementation | When the team has agreed that the UI decided on complies with the user story as already agreed with the client |
| Implement code for the user interface element of the story | When the product ‘passes’ the acceptance tests |
| Write JUnit Tests | When the team has agreed that the main areas of testing have been checked |
| Research options for calculation of profit or loss | When the team has agreed that the calculation decided on complies with the user story as already agreed with the client |
| Implement code for the backend calculation of profit or loss | When the product ‘passes’ the acceptance tests |
| Perform black box testing | When the team is satisified that they have covered all the external factors |
| Review results of black box testing | When the results have been discussed with the team |
| Implement any changes from the black box testing | When the team has decided that the tests have now been completed |
| Update the report about testing | When the sections are completed to the team’s satisfaction |
| Refactor Code | When the team is satisfied that they cannot improve it anymore |
| One page document on any public methods that the classes created may have and also the external classes that it uses | When the team has decided that it is completed |
| \*\*\*\*\*\*\*\*\*\*\*\*\* | \*\*\*\*\*\*\*\*\*\*\*\*\*\* |
| Research options for calculation of the share price. | When the team has agreed that the calculation decided on complies with the user story as already agreed with the client |
| Design Mockup for the user interface element required to show client breakdown of share set. | When the team agrees that the mockup complies to the user story as agreed with the client |
| Confirm the designs with the client | When the client agrees that the mockup complies to the user story as already agreed |
| Implement code for a calculation method for the share price. | When the product ‘passes’ the acceptance tests |
| Write Junit Tests | When the team has agreed that the main areas of testing have been checked |
| Assess the accessor method to get the data. | When the data is accessible |
| Store the data received in the database to be accessed later. | When the data is able to be read from the database |
| Research options for UI elements and types of implementation. | When the team has agreed that the UI decided on complies with the user story as already agreed with the client |
| Implement code for the user interface element of the story. | When the product ‘passes’ the acceptance tests |
| Implement black box testing | When the team is satisified that they have covered all the external factors |
| Review results of black box testing | When the results have been discussed with the team |
| Implement any changes from the black box testing | When the team has decided that the tests have now been completed |
| Update the report about testing | When the sections are completed to the team’s satisfaction |
| One page document on any public methods that the classes created may have and also the external classes that it uses | When the team has decided that it is completed |
| Refactor code | When the team is satisfied that they cannot improve it anymore |
| Write up final report | When all the sections are completed to the team’s satisfaction and that it ‘answers’ the brief |

## Sprint Burndown Chart

The team used a spread sheet which used a template of a burndown chart from :

*http://epf.eclipse.org/wikis/scrum/Scrum/guidances/templates/burndown\_chart\_D182CF23.html.*

This enabled the team to see the changes in the burndown chart as the sprint progressed daily.

Having this real time update of the progress kept the team motivated and ensured that the work times were consistent or as near to possible. It also provided a clear warning sign when the team was overworking themselves on a particular day, this improved the team’s consistently with work loads and effort. Thus allowing the team to be more Agile and follow the principles closer.

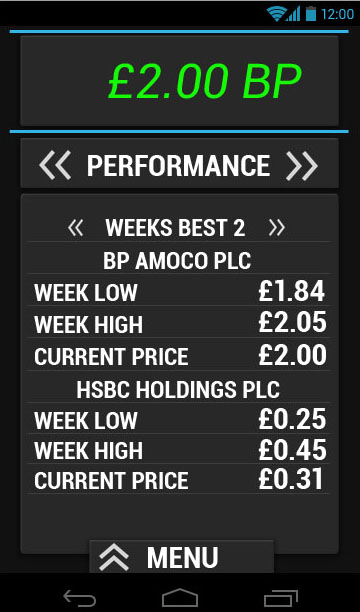
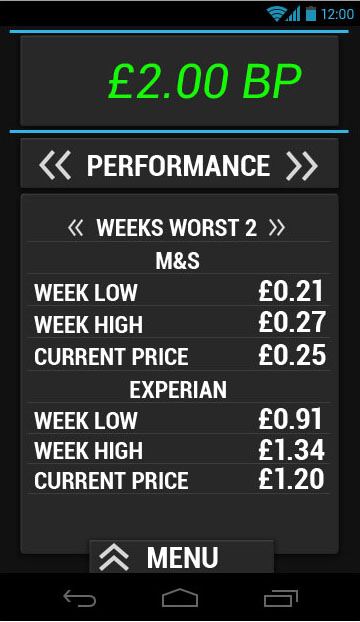
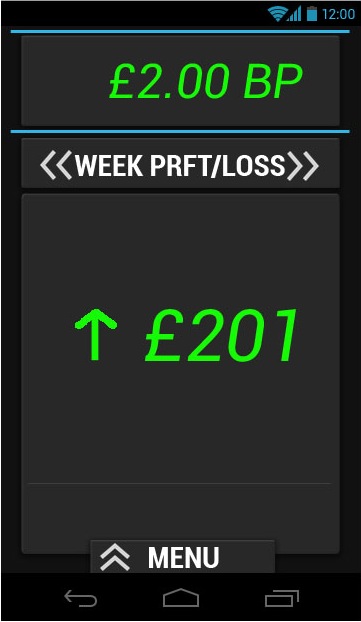
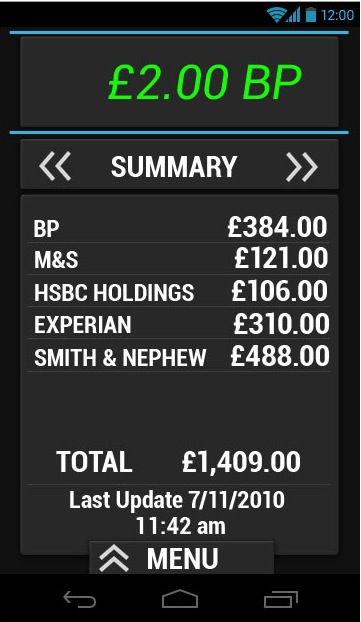
### Velocity

The team’s velocity was calculated to be *eight user story points or ‘giraffes’*. This was calculated on the basis of the sprint burndown chart showed that the team’s estimations were accurate as all tasks were completed on time. The original estimation was based on the first sprint with a bit extra extended to the expected load of this sprint as more testing was involved.

## Sprint Backlog

## [Insert Sprint Backlog]

## Design Mockups

## Evidence of conversations with client and the Scrum Master resolutions

Hi Janet, 11 November 2012 22.36

 Sure, we will make those changes.

Many thanks.

Heather

Sent from a mobile device so please excuse any grammar errors.

On 11 Nov 2012, at 20:58, Janet Hughes <[jhughes@computing.dundee.ac.uk](mailto:jhughes@computing.dundee.ac.uk)> wrote:

Thanks.  For the first screen, could you change “PRFT” to be “PROFIT”?

For the second screen, could you remove the .00 for the total so that it simply rounds up or down to integer pounds and could you add a little more space between that line and the “last update” line?

I like the font and the colours!

Coffee Janet

**From:** Heather Ellis [<mailto:heather@heatherellis.net>]  **Sent:** 08 November 2012 21:02 **To:** Janet Hughes **Subject:** Final confirmation of mockups

Hi Janet.

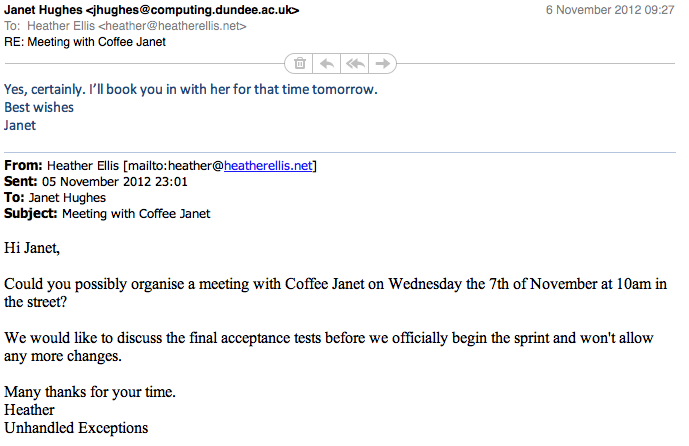
As promised, here are the mockups with your changes. I hope that these are acceptable, if not, please let us know.

Many thanks,

Heather

Unhandled Exceptions

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

****

**From:** [jhughes@computing.dundee.ac.uk](mailto:jhughes@computing.dundee.ac.uk) [[jhughes@computing.dundee.ac.uk](mailto:jhughes@computing.dundee.ac.uk)]

**Sent:** 04 October 2012 13:26

**To:** Euan Luke

**Subject:** Re: [spam ?] Re:

Fine, will do.

Janet

Sent using BlackBerry® from Orange

**From:** Euan Luke <[E.Luke@dundee.ac.uk](mailto:E.Luke@dundee.ac.uk)>

**Date:** Thu, 4 Oct 2012 12:06:48 +0000

**To:** [jhughes@computing.dundee.ac.uk](mailto:jhughes@computing.dundee.ac.uk)<[jhughes@computing.dundee.ac.uk](mailto:jhughes@computing.dundee.ac.uk)>

**Subject:** RE: [spam ?] Re:

Janet,

We have booked room 1.06 and were hoping to have the presentation there if that is ok?

Euan

**From:** [jhughes@computing.dundee.ac.uk](mailto:jhughes@computing.dundee.ac.uk) [[jhughes@computing.dundee.ac.uk](mailto:jhughes@computing.dundee.ac.uk)]

**Sent:** 04 October 2012 12:59

**To:** Euan Luke

**Subject:** [spam ?] Re:

Certainly, Euan. Consider it booked. Where shall I meet you?

Janet

Sent using BlackBerry® from Orange

**From:** Euan Luke <[E.Luke@dundee.ac.uk](mailto:E.Luke@dundee.ac.uk)>

**Date:** Thu, 4 Oct 2012 11:37:14 +0000

**To:** Janet Hughes<[j.z.hughes@dundee.ac.uk](mailto:j.z.hughes@dundee.ac.uk)>

**Subject:**

Dear Janet,

As Scrum leader for the group Unhandled Exceptions I am writing to ask if we could have the ten o'clock slot with you next Wednesday ( 10th of October) for our presentation of our working prototype from the first sprint.

Regards,

Euan Luke

**From:** Patrick Cowan

**Sent:** Thursday, October 11, 2012 6:50 PM

**To:** Janet Hughes

**Subject:** Submission

Hi Janet,

    I was scrum leader today and our group had one problem, none of us actually knew how to submit the first assignment tomorrow. Do we have to submit a hard copy to you or do we submit it via blackboard?

Thanks for the help,

    Patrick Cowan

**From:** Janet Hughes [[jhughes@computing.dundee.ac.uk](mailto:jhughes@computing.dundee.ac.uk)]

**Sent:** Tuesday, October 02, 2012 1:29 PM

**To:** Patrick Cowan

**Subject:** RE: Scrum

Hi, team.

1000 meeting is fine tomorrow, yes.

Re the user stories, I have some questions:-

a.       are they presented here in the order you recommend for development?  If so, which one(s) will be done in this first sprint thingy?

b.      What will **I** get from the first one listed?

c.       What’s all that API/accessor/mutator stuff?

d.      The things I most wanted of all (total value of portfolio) isn’t shown – does that mean it can’t be done?

Thanks

Janet

**From:** Patrick Cowan [mailto:p.cowan@[dundee.ac.uk](http://dundee.ac.uk/)]  **Sent:** 01 October 2012 20:47 **To:** Janet Hughes **Subject:** Scrum

Dear Coffee Janet,

As Scrum leader for the group Unhandled Exceptions I am writing to ask if you could find a moment to look over our user stories and acceptance tests and make sure everything we have written is how you want it to be. I would also like to ask if we could have the ten o'clock slot with you on Wednesday during the morning lab to discuss, go over and sign off the acceptance tests which we have produced. I have attached the user stories and acceptance tests with this e-mail.

Regards,

    Patrick Cowan

# Quality Assurance

## User stories and Acceptance Tests

**USER STORY**

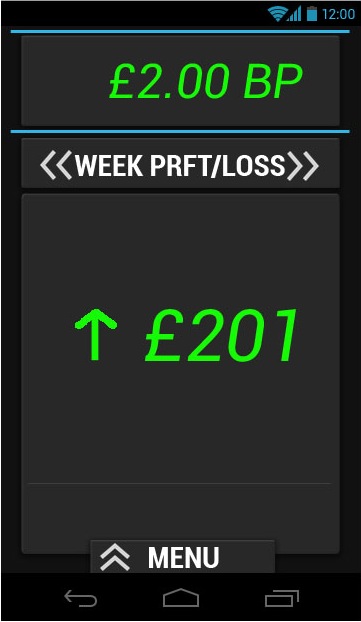
Business Value: 90

Effort: 4

Actual value: 22.5

When is it done: when the value calculated matches that as the online Yahoo Finance

**As a user, I want to view my weekly profit or loss in one line so that at a glance I can see how my shares performed this week.**

**ACCEPTANCE TESTS**

The design should match the mockup to the left.

The client should be able to calculate whether the weekly profit or loss the shares from the close of day at 1630 the Friday before. This data should be comparable to Yahoo’s finance on a laptop and the data output on the Android device.

The final figure shown should be rounded to the nearest pound with a profit showing in green and a loss showing in red.

If there is no data available at all, a message should read that there is no data available. If there is no data available for one company, a message appears on screen with the total and a message that states that a it is incomplete due to omitted data.

**USER STORY**

Business Value: 70

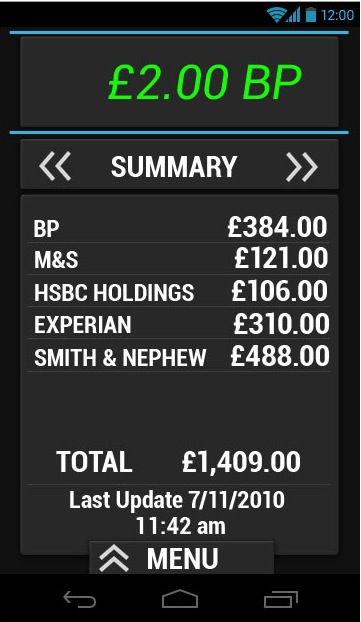
Effort: 4

Actual value: 17.5

When is it done: When the calculation matches that to the Yahoo Finance site.

**As a user I want the app to give me a current update of what each company's shares are currently worth.**

**ACCEPTANCE TESTS**

The design should match the mockup to the left.

This data should be comparable to Yahoo’s finance on a laptop and the data output on the Android device and proven to be correct.

The values shown should include pence. The shares should be in alphabetical order and a sentence should be at the bottom which states:

“Price correct of {insert date}”.

If there is no data available at all, a message should read that there is no data available. If there is no data available for one company, a message appears on screen with the total and a message that states that a it is incomplete due to omitted data.

## Quality Testing

### Unit Testing

The program was tested using various different testing methods to ensure that the application would be reliable and produce the desired results consistently. A breakdown of the testing procedures can be located below. These were useful in ensuring that the code was delivering what was expected and ensuring that the application would meet the user’s needs.

### Black Box Testing

Black box testing was used to ensure that the program functions as expected and consistently performs according to the client’s needs. This method of testing does not require direct access to the code as it is ‘hidden in the black box’. It allowed the team to focus on testing the functionality of the code and therefore making it more reliable.

#### User story 3

**As a user, I want to view my weekly profit or loss in one line so that at a glance I can see how my shares performed this week.**

*Remove redundancies and giving a reference.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **External Conditions** | **Valid Equivalence Class** | **Invalid Equivalence Class** | **Boundary** | |
| **Lower** | **Upper** |
| **Historical Data for the previous week** | 0-150,000  (A) | <0 (B)  >150,000 (D) | -1,0,1 (C) | 149,000 , (E) 150,000 , 150,001 |
| **Current Data** | 0-100,000  (F) | <0 (G)  >100,000 (I) | -1,0,1 (H) | 99,999 , (J)  100,000 , 100,001 |
| **Companies** | 5  (K) | <5 (L)  >5 (M) | 4,5,6 | 4,5,6 |

*Identifying the test cases and expected output.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case** | **Input Condition Covered** | **Historical Data** | **Current Data** | **Companies** | **Expected Output** |
| 1 | B | -1 |  |  | Invalid |
| 2 | D | 150,005 |  |  | Invalid |
| 3 | G |  | -2 |  | Invalid |
| 4 | I |  | 100,003 |  | Invalid |
| 5 | L |  |  | 3 | Invalid |
| 6 | M |  |  | 7 | Invalid |
| 7 | A, F, K | 0 | 150,000 | 5 | Valid |
| 8 | C, H | 1 | 1 |  | Valid |
| 9 | E, J | 149, 000 | 99,999 |  | Valid |

#### User story 4

**As a user I want the app to give me a current update of what each company's shares are currently worth.**

*Remove redundancies and giving a reference.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Input Condition Covered** | **Current Data** | **Companies** | **Expected Output** |
| 1 | B | -3 |  | Invalid |
| 2 | D | 100,006 |  | Invalid |
| 3 | G |  | 4 | Invalid |
| 4 | H |  | 8 | Invalid |
| 5 | A, F | 0 | 5 | Valid |
| 6 | C | 1 |  | Valid |
| 7 | E | 99,999 |  | Valid |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **External Conditions** | **Valid Equivalence Class** | **Invalid Equivalence Class** | **Boundary** | |
| **Lower** | **Upper** |
| **Current Data** | 0-100,000  (A) | <0 (B)  >100,000 (D) | -1,0,1 (C) | 99,999 , (E)  100,000 , 100,001 |
| **Companies** | 5 (F) | <5 (G)  >5 (H) | 4,5,6 | 4,5,6 |

*Identifying the test cases and expected output.*

### JUnit Testing

The JUnit testing provided the team with a few challenges in that it was not strictly possible to handle the Android libraries and GitHub. However, after a bit of

## [Complete JUnit Testing]

### Refactoring

## [Complete Refactoring]

### Metrics

## [Complete Metrics]

## Team Quality Assurance Standards

1. The team will use GitHub for all the code to ensure that all team members have an up to date version of the code at all time. This also holds members responsible for sections and changes to code should anything go wrong.
2. The team will practice paired programming wherever possible to ensure that each team member has a better understanding of the code. This also increases productivity within the team.
3. The code must use testing throughout to ensure that it is a) reliable, b) maintainable and c) that it is performing as expected consistently.
4. Scrums must be held daily to ensure that the team is communicating together on a daily basis. The scrum master, who will change each day on a round-robin system due to the team’s structure, will deal with any issues that arise.
5. The code should be well tested and follow a test driven approach as much as possible whilst ensure that it still complies with the Android libraries and the version control of GitHub.
6. The team will try and follow the Agile practices as much as possible to ensure good communication and work ethics. As a result, the team will try to work in the same environment and not work independently elsewhere.

## Paired Programming

## [Complete PP]

## Usability Score and Survey

The Usability score was a 85% usability score which the team was quite pleased with and hope to improve on this in the coming sprint.

## [Insert usability score]

## Acceptance Tests Approved in Sprint Review

## [Insert Aproved Accepatnce Tests]

# Source Control

Git Hub was used through out the project to get the team familiar with source control and the benefits of being able to revert code. Although the team never had to revert to any previous changes, it was acknowledged that there were numerous benefits of using such a system.

Each team member posted their commits when their task was completed however, there was a complication in that GitHub and Eclipse were having issues communicating when code was being edited. As a result the team had to ‘push’ commits through eclipse directly. This did not seem to have an impact on accessing the up to date code though. EGit was used here to perform this task.

There was however another issue in that when using EGit it seemed that some team members then lost the ability to commit other documents and other files through GitHub directly. This was resolved by another team member committing the documents. This change only occurred after the installation of the Egit plugin but the team handled it and found a new solution. The error message that some team members began to get can be seen below.

## [Insert failed commit image]

Furthermore, there was some issues in the the differences between GitHub on a Mac and on a Windows machine but once the team noted these differences, they were able to resolve the issues that arose. These included minor misunderstandings of how the system operated and how best to make use of the commits and changes.

Overall, it was found that GitHub was useful in version control and maintaining the code for different people.

# Lessons learnt from last sprint

## Unbalanced workload share

It was brought to the team’s attention that the original work load was unbalanced in the first sprint due to misjudgements of individual tasks.

## [Complete worshare]

## Team attendance could be improved

J

## [Complete attendace]