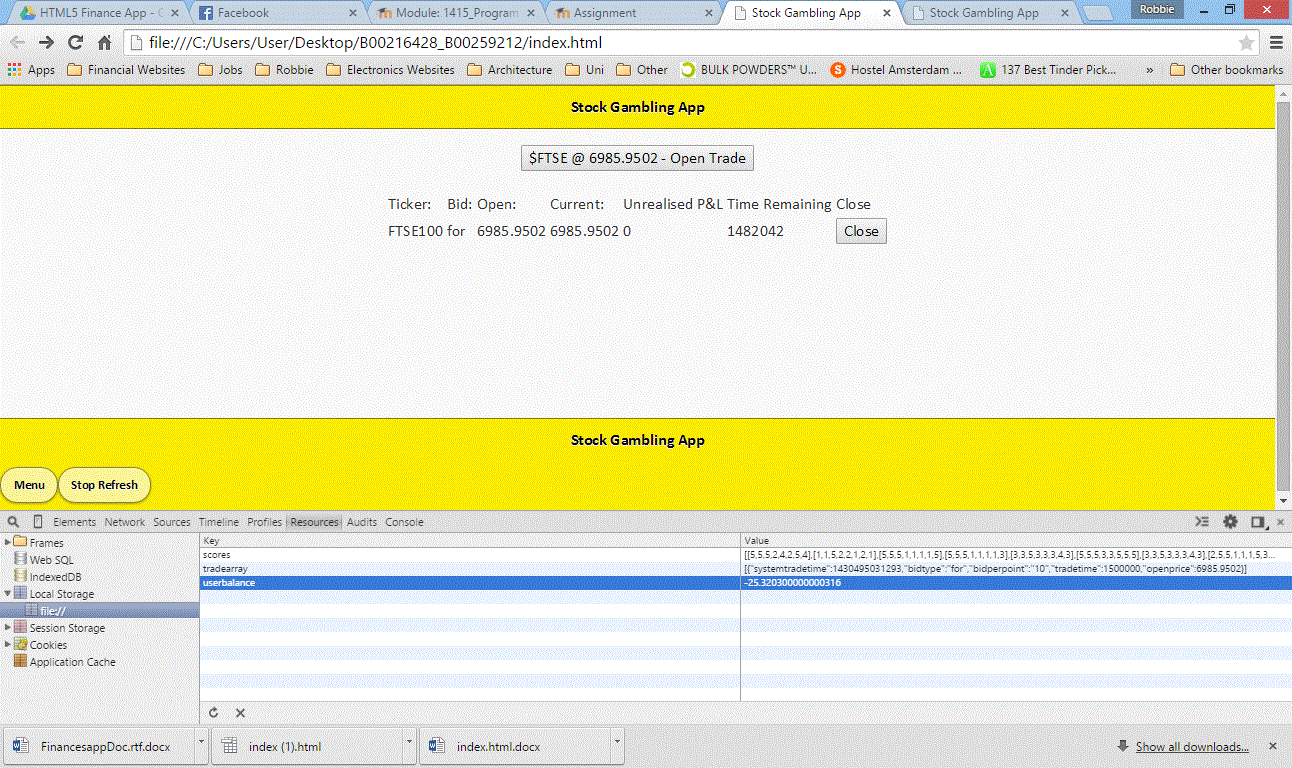
# Programming for Mobile Devices : Project Specification



## **1. Project Team**

The members of this project team, and intended responsibilities are:

Robert Miller - B00216428: App design, UI design & coding

Michael Farren - B00259212: App design, documentation & testing.

## **2. Detailed Project Spec**

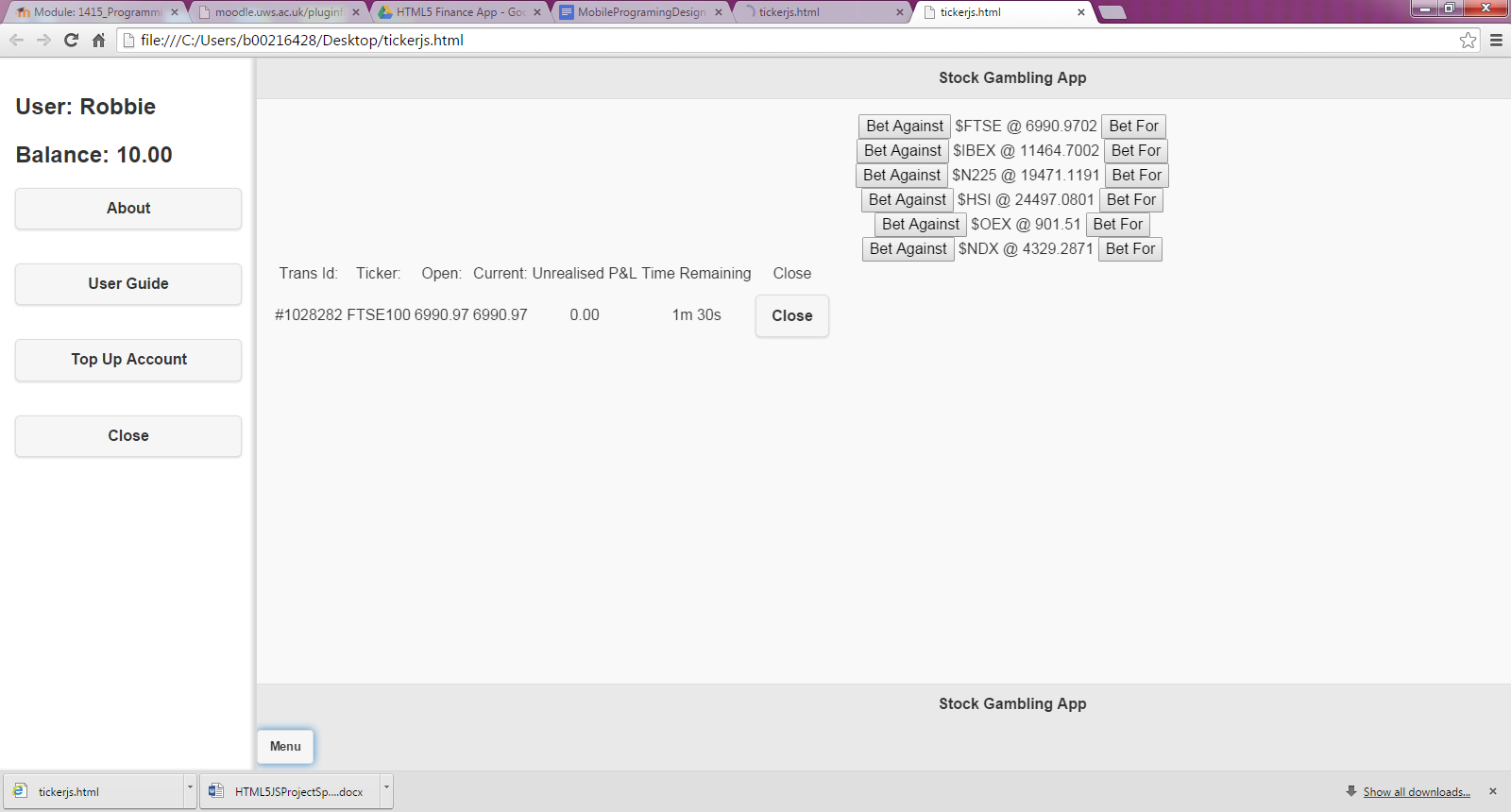
The project we submit will be a cross-platform Mobile application running through jQuery Mobile and javascript, (jQuery), which fetches stock market data in json format and allows the user to place fantasy trades on the stock market by clicking buttons:

1. The program will allow a user to add their name.
2. The program will allow a user to click buttons to big for and against the market and trigger “x” minuet trades on the stock market based on real live data.
3. The program will display the details of a trade and confirm before execution.
4. The program will display historical data of trades.
5. The program will display the details of all open trades.
6. The application should provide the following information:
   1. How to play the game

## **3. User Interface Prototype**

The image bellow shows an early mock-up / prototype of the user interface. The included HTML file (index.html) shows the coding for this.

The trade window will be the default display with options to execute a trade for and against a market. The trade window will also display open trades the user has executed. The side panel will display user name and account balance. The side panel will also have links to the “about page”, the “user guide” and the “top up account” page.



## **4. Design Documentation**

There is one application specific type required for this project:

Quote Type - This constructor returns the financial data from the external javascript feed.

This is a representation of a market entry and includes:  
 Ticker, Exchange Name, Symbol, Ask, Bid, High, Low, Last etc

Trade Type - This constructor holds and returns data of current and expired trades.

This holds data such as: Trans ID, Ticker, Open Time, Open Price, Current Price, End Price, Close Time.

Methods available to quote instances are:

* updateQuotes(): returns the object containing the relevant ticker data.
* reloadWindow(): reload page every x seconds.
* stopReload(): stop page reloading.
* promptOKFunc(): executes user defined trade, taken from the get slider values method. The bid value is then added to the table “results”.
* promptCancelFunc(): if the trade is cancelled null is logged in the console and the page is reloded.
* popUpBid():when .openTrade Button is clicked, this method calls, promptOkFunc(), or promptCancelFunc(), with the open bid string passed in.
* tradeEnd(): backend function to complete trade result and update userdata etc.
* closeTrade(): a user function to close a trade early.
* clockCounter(): a function that counts down till the end of a trade.

## **5. Application data and functions**

The following functions are application level functions (i.e. these belong to the application

window – essentially global variables):

* initialise(): this function begins the trading window and checks for connectivity and browser support. It limits trade execution during intermittent signal.
* Panel(): displays side panel to user.
* displayHistory(): displays users trade history.
* topUp(): a function to update the user balance
* changeName(): a function to change the users name.
* clockCounter(): a function that counts down till the end of a trade.

## **6. Project Aims**

The above design brief indicates the features that we fully expect to implement in the final

application. In addition to these features, we will attempt to include the following features:

* Trade Geolocation tracking - The geolocation of the user will be logged during each transaction to provide better insights to the app developers as to where the app is used most.
* Win Lose (P&L) Graphs - A data visualisation module will be created to graphically display how the user is doing with their trades within the app.

## **7. Testing**

## Testing was done through Google Chrome's Developer Tools, and the Apple iPhone 3Gs, Nokia Lumia and Samsung Galaxy S 3, were used for testing. This was to check the app worked with multiple operating systems and screens sizes as well as older less up to date devices. The iPhone 6 was also used to test the app's functionality on a more up to date device.

## Testing was done at various stages to make sure that all individual parts were working independent before combining and adding to the index page. This was done so that errors were easier to find and amend.

## **8. App Appraisal**

**Core Features**

The core features hoped to be completed fell slightly short of our goal of having multiple stock tickers that the user is able to bid on, we only managed to implement one single ticker due to the difficulty we had with finding a free financial data feed and after this was “found” we had a problem where the data provider has slightly stopped providing updated market data.

The code works; the problem lies with the data feed, as we aren’t getting updated data now unlike at the beginning of the project, too much testing has had the feed shut down.

All other core features only complementary to the main feature of trading on the market.

**Extra Features**

We hadn’t managed to implement any extra features however research was done into drawing with html5 & canvas for the purposes of displaying graphs of the market data.

**External Resources**

We chose a difficult road with regards to this project, firstly; it was hard to find a financial data feed provider that is free, they may exist but we found it hard and went with a financial data providers’ demo feed which we found at: <http://www.financialcontent.com/support/documentation/json_quote_api.php>

This feed has changed from the beginning of the project as it’s no longer being updated with current financial data, they may have noticed our traffic accessing their data feed and stopped it from being updated. However this is not a problem as the code has the correct logic to perform as expected. The only downside is that profit & loss can’t be made anymore in the “game”.

## **9. Team Appraisal**

Robbie took a leading role in the programming of this project as he saw career value in having worked not only with data feeds, but with financial data feeds and information, with the vision that it will show off the skills demanded by industry when upon completion of university.

Michael took a leading role in the project planning and execution assisting Robbie with the necessary to complete this project and hand in the deliverable within the deadline. He assisted with program design, user interface design, testing and documentation.

Both members shared equally in the work with Robbie taking a leading role in the programming and Michael taking lead with the testing and documentation roles.

We have enjoyed working together on this project and reaching a deliverable product.

## **10. Project Conclusion**

We fell slightly short of having a complete stock market system with multiple ticker options, we however feel the project has been a good outcome as you are able to atleast bid on the FTSE100.

We hope this project as part of our portfolio shows the scope of our ambition and how slightly below this we have came.

Above is a sceenshot of the app 15 minutes before deadline after an unsuccessful trade after the feed decided to work momentarily.

