Agenda Week 8: 12/11/19 - 19/11/19

Attendance: All

[Meeting Minutes](https://docs.google.com/document/d/1vivLzkubWN0pprNNSw90gJGs243B-TSGLo0b2VyRsiw/edit)

What to discuss

* What we did last week
* Architecture Diagram
* What we want to achieve for next week

What you did last week

* Carried out the necessary research into our pet projects.
* Continued work on pet projects:
  + Dexter
    - Updated Architecture diagram
      * <https://drive.google.com/file/d/1xI_9VtBbh_TMzYkzqtkDKiGEtLGVSLHF/view?usp=sharing>
    - Added logging functionality to API
    - Updated Sign up/Login functions
    - Assisted Mehmet with understanding the API
    - Began researching ASP.Net Core Identity
    - Added unit test project to the API solution
      * Uncovered difficulties with testing the API Controllers so I am currently looking into a solution
  + Dan
* Pet project 1 - Social Media Integration
  + Carried out research into twitter API and the ‘tweepy’ Python framework
  + Created basic python app using the tweepy framework
  + Created UI request for social media posts and sent to Niro
  + Created repositories for reddit and twitter apps on devops
  + Niro
    - Task 11 - Navigation System
* Map out User interface - use HCI knowledge to create interface
* Research into libraries
* Get familiar with Azure
* UI navigation via through buttons/arrows
  + - Task 5 - User account screen
      * Update API to retrieve account details, so that the user can view and edit
      * AI update - new page for account details
      * Get familiar with Azure
      * Account screen with various needed options (delete account)
      * Database with stored details.
    - Task 6 - User Interface Design
      * Gather information from other peers in the group for their input in the interface (Each pet project may need additional sections within the interface so need to discuss)
      * Use HCI principles from existing knowledge
      * Create a concept for design
      * Finalise design for user interface
* Connor
  + Task 6 - Graph Visualisation
* Completed a basic graph (with dummy data)
* Started trying to implement actual bitcoin data into graph
* Started document detailing the different types of trading and how it will impact development
  + Task 7 - Live and Historical data acquisition
* Discovered and started implementing functionality for getting live data and historical data from Cryptocompare
* Started implementing functionality with graph implementation
* Discovered and started implementing functionality for getting live data and historical data from Alphavantage
  + Mehmet
    - Met up with dexter
    - Created a controller for positions
    - Hooked up the API to the ‘Positions’ storage table
    - Started with post end-point to send data to storage table
* Researching on how to implement end-point
  + - Created a position class
    - Created a position entity class

What to do for next week:

* Dexter
  + Finish researching ASP.Net Core Identity
    - If it is suitable for our project, then I also need to implement this
  + Find a solution to the Unit test project issue
  + Send a design request to Niro for the administration website
  + Finish creating user stories/use cases/sequence diagrams for administrators
* Connor
  + Implement minor external API data (historical data) into basic graph
  + Start creating further functionality for different data requests for the graphs
  + Create/finalise conceptual design for line graph
  + Further research into graph design and characteristics
* Dan
  + Carry out research into getting historical data from Reddit and Twitter APIs
  + Create separate Tables in table storage for Twitter and Reddit
  + Implement storage of posts from twitter in table storage
  + Implement storage of posts from reddit in table storage
* Mehmet
  + Finish post end-point to send data to the storage table
  + Research into the functionality of get end-point to retrieve data from storage table
  + Research and understand fiddler to test end-points
* Niro
* Implement UI requests from other members into conceptual UI Design
* Ensure experience for the user knows exactly what to do efficiently and effectively
* Match user interface with user stories
* Get UI request for graph

Individual review of progress:

* Dexter
  + Add unit tests for API work
    - Encountered a problem with unit testing the controllers, currently looking into a solution - May use mocking?
  + Deploy a second Angular web app for Administrators
    - Did not quite finish the designs in time
  + Did not research react as we are sticking with Angular
* Connor
  + Successfully created line graph with dummy data. Dummy data used as at the time a sure way of getting data from external API’s wasn't known.
  + Methods for getting external API data discovered (http get for Cryptocompare) (Json request for Alphavantage).
  + Implementing external API’s functionality has caused issues for the line graph that now does not function. Ideally the basic chart functionality with some basic external API data will be working for next week's meeting.
* Dan
  + Implement storage of obtained data in table storage - completed
  + Decide on and implement a method for choosing which posts to store - Started but not yet completed. I first wanted to find out which information is provided by Twitter and Reddit.
  + Send UI requirements to Niro - completed
  + Begin research and implementation for twitter bot - completed
* Mehmet
  + Still haven’t started coding the functionality of end-points and asked dexter for some help
* Niro
  + Implement UI requests from other members into conceptual UI Design
  + Ensure experience for the user knows exactly what to do efficiently and effectively
  + Match user interface with user stories
  + Get UI request for graph

Notes:

* Think about additional contributions
* More than the sum of the components or something like that he said.
* Discuss contributions each week - Review what we said we were going to compared to what we actually did.
* Update diagram:
  + Move crypto compare to left hand side
  + Add other crypto api
  + System components
  + Add an index?