**During the Holidays**

During the holidays no meetings were held (as our group was separated), instead we communicated mainly via emails. These points summarise our decisions made and discussions we had via these emails:

* Made the decision to use email as our group’s main communication method. This was chosen as our group was widely separated and we had no chance of face to face contact
* **Brainstorming:**
  + We thought of general ideas of where we wanted to go with the system
  + Main role is to be a ‘News Feed Aggregator’ (as per the specification), as such it must be able to crawl, parse, store and display content from RSS feeds as its main functionality
  + Decided we would be making the product as a service rather than a system (so we could focus mainly on user functionality)
  + System could be broken down into 2 main parts, front-end for displaying the content to the user and backend for performing all the processing
  + Discussed the technologies we should use to construct the system:
    - PHP web server, accessible to the user via web pages
    - C++ for the backend
    - MySQL database for storing the data (MySQL was chosen as it was a cheap and simple option)
* Discussed the architecture to use for the project. Repository structure, front-end and backend talking to database which stores feed and user information
* Discussed possible users:
  + Standard users (business people, students, general public)
  + Administrators (for accessing the system)
* Discussed the content sources we could possibly crawl:
  + RSS (most importantly)
  + Emails
  + Forums
  + Standard websites (HTML)
  + Social Media (Facebook, Twitter)
  + Weather feeds
  + Decided our main task in the holidays would be to work on prototypes for each module of the project
    - This is to both test our ideas and learn how to correctly construct the module
    - The prototype for each section is a ‘throw-away prototype’, we will be building the final system when the session starts (and the prototype is done)
    - Assigned some general roles for which section we would be working on:
      * Ian – Front-end web development
      * Alex – Backend parsing of RSS and other content and crawling of feeds
      * Aron – Backend threading
      * Michael – Algorithms for queuing and documentation
    - Registered Domain Name (newsfeeder.co) in preparation
    - Created SVN to store the work and simulations done in holidays

**27/2/12**

**Progress Reports:**

* **All**
  + Fleshed out requirements and refined ideas on architecture/technologies for their section/module
  + **Ian**
    - Developed the initial front-end website (working login, display of dynamic content, etc)
    - Tested PHP PDF generation (is working, not the most efficient method though)
    - Tested Bing API for searching the internet for suitable feeds to add, works well (slow due to memory cache)
  + **Alex**
    - Written up RSS parser
    - Currently using the library libXML (has found issues with memory leaks)
    - Written up basic crawler, tested and runs quickly
  + **Aron**
    - Ran simulations of the priority queue. Is working on integrating them successfully with threading (and testing that)
  + **Michael**
    - Thought of algorithm for priority queuing
    - Tested algorithm with the help of Aron’s simulations

**Agenda:**

* Completed set tasks for first week (get group, project and supervisor), all done beforehand
* Confirmed times for meetings:
  + 4:30 Monday, after the 321 lecture
  + 3:30 Wednesday with Luke
  + Clarify any problems/questions we had with progress during break or initial brainstorming
    - Marked off any implemented functionalities (e.g. login and registering)
    - Added ‘Abuse Reporting Blacklist’ to requirements, this is the ability for web masters to report if the system is spamming/abusing their site
  + Created Initial Use Cases
  + Answered any questions on each section which had been worked on over the break:
    - Michael answered issue with how the priority algorithm handles initial start-up of the queue
    - Alex noted concerns with libXML and about whether to use Boost or P threads. Selected Boost
  + Decided upon documentation to start:
    - Formalised requirements
    - Use cases
  + Specified Ian as team leader, as he has the easiest means of communication with our supervisor Luke.
  + Specified Michael as scribe
  + Clarified questions to ask Luke at first meeting:
    - Can we have copies of the project specification?
    - Are we selling as a service or as a system?
    - What do you think of development types (prototyping, agile, RUP)?
    - Can we use VISIO?
    - Can he purchase soft-drinks (priority Vanilla Coke)?
    - Clarify Expectation of Time-Line.
    - Clarify Roles.
    - Clarify Scope.
    - Clarify Requirements.

**Work for next week:**

* **Ian**
  + Continue testing and implementing high-level features to the front-end prototype
* **Alex**
  + Work on issue with RSS memory
  + Continue work on HTML parser
* **Aron**
  + Implement ‘start-up’ priorities on the queuing and test
* **Michael**
  + Work on creating the requirements documentation
  + Documenting discussed use cases

**29/2/12**

**Agenda**

* Went through previous questions for Luke:
  + **Can we have copies of the project specification?**Copies Given
  + **Are we selling as a service or as a system?**Either, we selected to sell as a service with possible smaller applications as systems/products
  + **What do you think of development types (prototyping, agile, RUP)?**Whatever, went with prototyping/agile method
  + **Can we use VISIO?**Yes, we will be using it for major documentation
  + **Can he purchase soft-drinks (priority Vanilla Coke)?**Possible ‘donations’
  + **Clarify Expectation of Time-Line.**Generally follow subject outline, already ahead in this regard
  + **Clarify Roles.**No specific roles, chose to specify areas of project to people rather than standard roles
  + **Clarify Scope.**Done
  + **Clarify Requirements.**Added some more requirements, confirmed current ones are suitable.
* Luke brought up the following points:
  + Force a print view of a feed item’s page to retrieve the additional content (so as to remove non-necessary features like advertising)
  + Look into stress testing methods
  + For adult content filters, look at hooking into existing browser filters
  + Look into creating payment scheme for project:
    - Pay to remove advertisements from page?
    - Additional features?
  + Add localised content filtering (for example in Germany Swastikas should be censored/filtered)
  + Add ability for users to customise sheets themselves (drag and drop interface)
  + Formatting sheets is at much higher priority, as is directly related to the user’s experience
  + Restrict sheet sizes (such as only 10 items per sheet)
  + Look into restricting user privileges (limiting sheets, feeds, etc, so as to not abuse)
  + SSL? Encrypting communications between system servers and the client

**5/3/12**

**Agenda**

* **Progress Reports**
  + **Ian**:
    - SSL Implemented
    - Problem with deletion, need to add dynamic SQL generation (databaseconnection.php)
  + **Alex**:
    - Socket designed, in the progress of implanting
    - Refactoring the backend
  + **Aron**:
    - Designed customisable UI layouts
  + **Michael**:
    - Designed customisable UI layouts
* **Agenda**
  + Aron and Michael discussed the ‘customisable UI layouts’ proposed by Luke
    - Essentially the user can drag and drop feed types into a defined layout
    - The layout itself is defined by dragging and resizing areas on a sheet to be filled with content
    - To use a standard column/row type format (as is similar to standard sheets)
  + Added additional requirements, formally added  the ‘customisable UI layouts’  and what it entailed to the requirements
  + SSL
    - Group decided it would probably be by user preference and it would default to ON
    - Ask Luke for his suggestions on the security and how to implement it
* **Actions for next meeting:**
  + **Ian:**
    - Correct any  SSL issues and get prepare questions to ask Luke about the issues
  + **Alex:**
    - Continue working on Parsers (especially HTML parsing)
  + **Aron:**
    - Test threading in regards to the crawler more thoroughly (compare more threads vs fewer threads)
  + **Michael:**
    - Continue writing use cases / requirements documentation

**7/3/12**

**Agenda**

* Asked Luke about Security and SSL
  + Authentication for services (such as API and downloader application)
    - Given to registered users
  + Authentication only if we want to restrict access to registered users
  + SSL on important data (login, registering, etc)
  + No SSL on non-important data (its processing costs are too high to use for the whole site)
    - May put it on data about user preferences
    - Luke suggested defaulting to off instead of on
* Asked about initial documentation:
  + Nothing complex for the documentation, keep simple especially for user documentation
  + Format the requirements in table form
  + Start a WordPress for keeping diaries
  + Document main competitors
    - Google Reader
    - Mashable

**12/3/12**

**Progress Reports**

* **Ian**
  + Implemented changes suggested by Luke about SSL
  + Found issue with PHP database connections class, currently trying to fix
* **Alex**
  + Basic HTML parser finished
  + Begun wrapping classes
  + Wrote up backend class diagram for discussion in this meeting
* **Aron**
  + Tested threads, more threads superior (provided we do not pool too many), performance hit is due to queue sorting
  + Started technical documentation
* **Michael**
  + Finished initial high-level use cases
  + Still writing up requirements documentation (working on priorities currently)

**Agenda**

* Analysed the current queuing architecture:
  + Is Crawler Controller getting the work from the MySQL updater or is it from the Crawler Thread? From the Crawler Thread
  + Polling (Crawler Controller polls from database every n seconds) OR When it initialises, have it add a PLACEHOLDER item onto the queue? Decided upon Ian’s suggestion, using the placeholder
  + Essentially when a new feed is added to the database there will be a message sent to the ‘crawler controller’ which will create a placeholder item in the queue for that new feed. This feed item will be filled in when it is first reached in the queue and crawled.
* Aron and Michael explained queuing architecture to group:
  + Continuous queue
  + Passes the top X items to X waiting threads, then resorts the remaining queue after incrementing the priority
  + Passed items are crawled in these threads and added back onto the end of the queue
* Analysed Backend design (and class diagram):
  + Refactored current design to fit more with the Model View Controller pattern
  + Using thread pools for all major services (crawler, SQL, etc)
  + General design involves running individual threads for each major service
* New executable to perform the Bing API search in C++ as opposed to PHP (very low priority), currently the search is being performed very slowly (10 seconds to bring up the first result) and mem-cache is still leaking.
* Discussed Iterations: initial throwaway prototype where each subsystem and its features are implemented (to test if they are feasible), followed by a standard iterative development cycle where the product is fully designed.
* Iterations are:
  + 1 (Prototype):
    - Get off the ground
    - RSS Parsing
    - Site Login
    - Sheet Management
    - Database connections (backend)
    - Database connections (frontend)
* 2 (Prototype):
  + Threading
  + Queuing Simulation
  + Communication between frontend and backend (was not prototyped)
  + Interfacing with front and back ends (Marks completion of protyping)
  + Finalise Requirements (right at end)
* 3
  + Finalised designs
  + Begin development of final products
    - BASE REQUIREMENTS
* 4
  + More functionality
  + High level documentation
* 5
  + TESTING
  + Bug squashing

**Actions for next meeting:**

* **Ian**
  + Keep working on fixing that database controller
* **Alex**
  + Refactor backend to fit with new design
* **Aron**
  + Document the proposed backend redesign
* **Michael**
  + Continue working on requirements
  + Look into database controller with Ian

**14/3/12**

**Agenda**

* Discussed backend redesign with Luke, explained queuing method. No problems found
* Luke suggested we summarise our possible applications and addons:
  + Mobile Application – specific interface to browse feeds
    - Mobile support better option?
  + Java Download Application
    - Browser Addin or Desktop Application? Desktop Application
    - Main requirement: lets you view sheets offline
    - PDF or Offline Sheets
    - End of session
  + Sheet Designer Application
  + Storing Images
  + Add Image Searching
  + Social Network Integration
  + Public API Web Service
  + Pulling down email content (additional content type)
  + Email Notifications
  + Emailing sheets

**19/3/12**

**Progress Reports**

* **Ian**
  + Database controller still broken, currently working on it
* **Alex**
  + Backend refactoring going well
  + RSS parser and RapidXML implemented (RapidXML not leaking so far)
* **Aron**
  + Architecture documentation begun
* **Michael**
  + Initial requirements done, looking for approval

**Agenda**

* Discussed the possible extensions suggested last week, and whether we would do them:
  + YES OR NO (or wait till later to decide)
    - Mobile Application -**very very low for individual application**
      * Should we just have mobile support instead? **YES we will provide mobile style support**
      * Java Downloaded Application **Yes, will confirm the functionality at a later date**
        + PDFs or Offine? **Not yet decided**
        + Sheet Designer Application **Definitely Not (already done in UI)**

Is currently being done in UI should we add specific application?

* + - * + Image Storage  **Could be done, have to do tests on storage**
        + Image Searching **Definitely Not**
        + Social Network Integration **Yes**

What does this entail?

Authenticating using apps and logons **Yes**

Adding social network content **Yes**

* + - * + Public API Web Service **Yes if we can get the Java App done, since it will use the service**

Why and for what cases?

REST API

Already being done for Java Downloaded app

* + - * + Email as Content Type   **VERY LOW**
        + Email Sheets and Notifications   **YES**
        + Notifications **YES**, Articles sent as PDFs? **YES**

User confirmed: Do you want to send notifications? If yes, Do you want to send a PDF copy of the article also sent?

Batch emails (Digest)

* Looked into Front End Design:
  + Major suggestion was the change to Smarty templating, to replace the GUI maker class currently being used, Ian is looking into this
  + Site Structure – fine, similar to other services
  + Classes – basic name changes, removal of init.php, change to Smarty
  + UI – not discussed
  + Sheets – not discussed
  + Custom Sheet Design – not to be discussed until better defined

**Actions for next week:**

* **Ian:**
  + Keep working on fix to database connections
  + Look into Smarty templating
* **Alex:**
  + Keep refactoring backend
  + Test RapidXML further
* **Aron:**
  + Look for Smarty guidelines to use for project
* **Michael:**
  + Add new applications into the requirements

**23/3/12**

**Agenda**

* Ian has fixed issue with database controller, was an issue with scoping of for loops and some SQL queries
* Short meeting with Luke, on Friday instead of Wednesday
* Alex brought up the issue of logging:
  + Only errors currently being written to log file
  + Should implement verbose logging
  + Luke suggested we provide auditing
* Showed Luke our requirements document:
  + Suggested rearrange requirements based on relation to other requirements rather than their priority
  + Added some additional requirements:
    - Logging and Auditing
    - Learning Algorithm (very low)
    - Printing support
    - Payment system
    - Feed categories
    - …

**26/3/12**

**Progress Reports**

* **Ian**
  + Begun conversion to Smarty (needs Aron’s help to choose a templating style)
  + Found new design template to use for the front-end look
* **Alex**
  + Finished refactoring backend to fit with design documents
  + Added more functionality to RSS parser (can now handle more tags to retrieve the correct information)
* **Aron**
  + Found his old Smarty Templating guidelines, had fixed some issues with them to fit with the new project
* **Michael**
  + Added additional requirements suggested by Luke to documents

**Agenda**

* Chosen Smarty Templating Style
* Backend Requirements Finalizing
* Michael explained Priority Algorithm to Alex, to be implemented into the crawler
* Made decision to use Ian’s new design template
  + More professional
  + Provides additional JavaScript functionality the old template did not have
* Implemented smarty naming convention
* UI design style
  + Boxes to hold articles
  + Similar to the customised sheet design style (placeholders using Smarty to fill)
  + Discussed priorities for requirements, moved some applications to low, and functionality to medium

**Actions for next meeting**

* **Ian**
  + Convert frontend to use the new smarty template
* **Alex**
  + Implement the crawler queue with priority into the backend crawler
* **Aron**
  + Look into REST API’s for the Downloader Application
* **Michael**
  + Work on next iteration of priority algorithm

**28/3/12**

**Agenda**

* Discussed possible properties for the sheet and sheet view:
  + Font size (for readability purposes)
  + Font colour
  + Background colour
  + Header colour
  + Generic sheet icon (or header) VS a custom sheet icon
  + Could possibly restrict the colour options of non-subscription users (for example they might only be able to use black text)
* Made the decision not to use tokens, as it is a subscription service not one where users pay for pieces of extra functionality
  + Also going implement a payment gate for users to buy their subscription
  + Luke explained tokens were only suggested if we didn’t want to implement an actual payment method. He said decision was fine, though a low priority
* Luke added the requirement of administrators being able to manage these subscriptions
* We explained to Luke the notifications system and how it would work on the sheets.
  + Luke said was good functionality, especially the ability to email notifications
  + Told us to change the actual sheet notifications from a medium to low priority
* Went through requirements document with Luke, he made the following suggestions:
  + Add a category explanation in the glossary
  + Add a glossary diagram to show how the items, feeds and sheets interact with each other (1 sheet may contain many feeds, etc)
  + Told us to split the non-functional requirement ‘Sustainability’ into 3:
    - Sustainability – if the system can maintain itself (clearing logs, etc)
    - Maintainability – how easy it is for someone to come and fix part of the system
    - Good design – coding style, structure, etc

**2/4/12**

**Progress Reports:**

* **Ian**:
  + UI with new template refactored
  + Separation between profile settings and account settings
* **Alex**:
  + Log manager
* **Aron**:
  + Requirements
  + Investigating web service for Java
* **Michael**:
  + Requirements
  + Meeting minutes

**Agenda:**

* Had a number of discussions concerning:
* Separation of the modules in the backend
  + Current modules in the backend are for statistics, crawling, admin (and the master for starting all other modules and threads)
* Messaging Interface
  + New table in schema ‘Notification’
  + SQL generation on item update
  + Front end generates document
* Gave the API higher importance (as suggested by Shaun)
  + Instead of having individual database connectors for each subsystem of the project, thinking of integrating them all into a shared library
    - Will require a cross-language library
    - Looking into SWIG as a possible option for wrapping the shared library
    - We will only proceed with this if it is possible to run the shared library on all subsystems (C++, Java and PHP)
* Looked over the requirements to be submitted this week. Made no major changes to the document other than adding the ‘Notifications’ requirement to the front-end

**Actions for next week’s meeting**:

* **Aron**
  + Look into adding generalised Database Shared Library
* **Ian**
  + Notifications, add feed widget
* **Alex**
  + Fully implement master thread
  + Start administrator service and integration with the front-end
* **Michael**
  + Finalise requirements document

**5/4/12**

**Agenda**

* A short meeting today due to the business of Easter, only Michael and Aron could attend.
* The main aim of this meeting was to look through the requirements document with Luke and check everything was fine. Luke had the following suggestions:
  + Remember to add the glossary diagram, for explain the relationships between sheets, feeds and items
  + Change the ID’s to be more relevant to the requirements document, since at this stage they aren’t being used in any other document. Changed from ID’s like ‘FUSR1’ to ‘F2.1’
  + Add a requirements summary table to explain the priorities without the assessors needing to read through the full descriptions
  + Remove the time constraint as it is not well defined (only wrote the requirement for the project as a whole, not the individual parts)
* Explained to Luke our idea for using a shared library for database access, and our use of SWIG. Luke agreed with this decision.
* Luke explained he will not be able to meet with us for a while, due to the Easter break and him travelling to Singapore

**12/4/12**

* **Progress Reports**
  + **Ian –**Currently converting the abuse service to use the new UI theme
  + **Alex –**Implemented master thread and started on the implementation of the admin service
    - **Problems Found for Admin Service:**
      * Authentication – Using basic authentication for prototype. Symmetrical key encryption for final product
      * How do we notify when a module is stopped or started? – Poll from front end and keep querying until the system registers it as having started/stopped
  + **Aron –**Investigated shared library and SWIG, so far it is looking possible but difficult
  + **Michael  –**Added all meeting minutes to the blog site and started on new documentation (vision for technical documentation, etc)
* **Agenda**
  + Meeting in Easter holidays, hence the later data, no meeting with Luke this week
  + Had discussion on the shared library
    - Aron explained the general idea of the library
      * Act as a database connector to all the parts of the system (front end, backend and downloader app)
      * Main function is to be used as a query builder where each part sends its parameters and receives the response
  + Discussed iterations
    - Main decision was to finish our initial throw-away prototype, so we could start on the actual product.
    - Already have both the frontend and backend working together and performing the base (and most high) requirements, in a previous SVN build. So will be using this as a definition of the prototype.
    - The reason for this decision was so we can have a major milestone for the project, meaning we will know how far we have progressed in actual development.
  + Ian purchased ‘News Feeder’ shirt, showed it to the rest of the team. Everyone agreed to purchase a shirt as well, so our group will look professional for trade show.
  + Discussed technical documentation
    - Mainly talked about how we need to start this as it is the next major deliverable
    - Asked Michael to find out the sections required for this document so it can be discussed next week
  + Discussed abuse reporting
    - Authentication – we will be using basic authentication, where initially the web master will login using the data from the crawler when crawling their site. They will then login by providing a username and password that they set after the initial login.
    - …
* **Actions for next meeting**
  + **Ian**
    - Implement abuse system authentication (and generation of meta data for initial login)
  + **Alex**
    - Rollback to working crawler module and integrate with current solution for prototype
    - Continue work on the admin service
  + **Aron**
    - Continue work with SWIG
    - Add queue threading to prototype
  + **Michael**
    - Add queue priority algorithm to prototype
    - Get list of technical documentation requirements

**16/4/12**

**Progress Report:**

* **Alex**
  + Added crawler into prototype, ready to have priority queue integrated
  + Still implementing admin service
* **Aron**
  + Still looking into the shared library and SWIG
  + Made progress with SWIG, is looking more practical at the moment
* **Ian**
  + Finished the abuse reporting service
* **Michael**
  + Looked into technical documentation
  + Designing layout creation interface

**Agenda:**

* Short meeting and no extra meeting this week due to Luke’s availability
* Alex brought up issue with parser, it currently does not correctly register redirections (it attempts to crawl the page that is redirected to)
  + Ian explained how to fix this using the Curl library, an extra option can be set to not follow redirects
* Discussed our next project deliverables, namely:
  + Technical Documentation
  + User Manual
* Discussed the possible sections and who could do them for the technical documentation:
  + Vision – Michael
  + Requirements – Done
  + Constraints and Operational Requirements – Michael
  + Software/Technologies – Aron
  + Hardware Specifications – Ian
  + Host Purchasing and Domain Registration – Ian and Alex
  + Development tools – Michael
  + Architecture document – Everyone
    - Repository Structure
    - Subsystems/Modules
    - System Interaction
* UI Design – Michael and Ian
* Design Document (sections still to be defined) – Everyone
  + UI
  + Classes
  + Data Flow
    - Classes
    - Data Dictionary
    - DFD’s
    - Entity Relationship Diagrams
* Communications
  + Meetings
  + Diary
  + Blog
  + Emails

**Actions for the next meeting:**

* **Ian**
  + Write up dummy PHP protocol for the admin service so the front end interface can be developed
* **Alex**
  + Continue working on the admin service
* **Aron**
  + Complete SWIG investigation and confirm if SWIG and the Shared Library are valid choices
  + Implement crawler threading
* **Michael**
  + Implement crawler queuing
  + Investigate layouts further

**23/4/12**

**Progress Reports**

* Ian
  + Completed testing of server socket on client
  + Implemented SSL for socket
* Alex
  + 50% of admin service implemented
* Aron
  + Began implementing crawler, still discussing the possible ways to increase performance of the current model
* Michael
  + Finalised priority algorithm and began implementing crawler queue items

**Agenda**

* Ian introduced a new feature: Starred Items
  + Click on star for a feed or item
  + These items would be ‘favourited’ and visible for a longer time on the user’s sheets
  + Possible to introduce a sheet made up of only a user’s starred items/feeds (like the starred section of an email inbox)
* Discussed how we would be storing the favourite icons from the crawled feeds database
  + Database table with:
    - Domain
    - Fav Icon
  + Any articles from a particular domain will display that fav icon
  + Problems with this solution:
    - Some domains may consist of multiple fav icons (such as a domain where there are multiple blogs)
    - This solution assumes the fav icon will be readily available as a fav.ico file (which is not always the case) – RESOLVED, we will check for this .ico or otherwise by the linking done in the domain pages
  + System will store the icons based on whether they are already stored in the database or not (to prevent duplicate images being stored wasting space)
* Discussed the security, mainly SSL
  + Ian has purchased a SSL certificate and has implemented it on the frontend
  + Looking into how simple it is to implement SSL for the backend as well
* Further discussed the ‘patch and match’ system
  + Will be a rule based system
  + Mostly likely method will be going to the container(s) of the body of a HTML page where text similar to the summary is held, then get all content within them
  + 3 categories of tags being parsed, each will be handled differently:
    - Keep tags and content
    - Remove tags and keep content
    - Remove tags and content
  + All styling will be removed from tags (we will substitute with our own)
  + Images will be stored

**Actions for the Next Meeting**

* Ian
  + Test socket integration with Alex
  + Implement SSL
* Alex
  + Finish implementing the admin service
  + Test sockets with Ian (without using SSL)
* Aron
  + Continue implementing the crawler queue
  + Decided is SWIG is a viable option for the shared library
* Michael
  + Continue implementing the crawler queue
  + Start on vision document

**30/4/12**

**Progress Reports**

* Ian
  + Finished admin service frontend
* Alex
  + Finished admin service backend
* Aron
  + SWIG working for:
    - Java
    - C++
    - Not PHP yet (Aron asked Ian to help him integrate due to issues with namespaces)
* Michael
  + Began vision document and gathered list of documents to include in technical documentation

**Agenda**

* Discussed SWIG, Aron explained it was going well. We decided to definitely use SWIG and the shared library in our solution
* Item Retention
  + Based off algorithm an algorithm. This will take into account the following:
    - Starred items retained forever
    - How many on sheets
    - Date retention (unless starred feeds will always be dumped after 12 months)
    - Based on feed updates (feeds updated more often will be held for shorter, since more content will be passing through)
    - Limited number of items (e.g. only holding 10 items for all feeds)
* Documentation breakdown:
  + Subsystems:
    - Downloader App  – Aron
    - Front end (PDF generator included) – Ian
    - Back end – Alex
    - Shared Library – Michael
    - Web Service – Aron
    - Database – ???
  + Vision    – Michael
  + Requirements – Done
  + Use Cases (diagrams, descriptions) – ALL
  + Architecture – Aron, ALL
  + Class Diagrams – ALL
  + UI Design – Ian
  + DFD – Talk about with Luke
  + State Machine Diagrams – Talk about with Luke
  + Interaction Diagrams – started
  + USER MANAUL – IAN and Aron

**Actions for Next Week**

* **Ian**
  + Start writing the user manual
* **Alex**
  + Start formal class diagrams for backend sub-system
* **Aron**
  + Finish implementing SWIG with PHP
  + Start on designing the Downloader Application (structure, etc)
* **Michael**
  + Continue working on the vision document

**2/5/12**

**Agenda**

* First meeting with Luke in 3 weeks, so focused mainly on explaining our progress and decisions we had made over that time, including:
  + The use of a Shared Database Library (SWIG), Luke approved of this design
  + Use of SSL, explained the front-end had been tested using SSL
  + Demonstrated the Admin Service to Luke
  + Storage of images (particularly favourite icons) in the database. We explained we would be storing all images in the database, in 100×100 size thumbnails with links to the full image (to save space)
  + Methods of item retention (explained our algorithm ideas)
  + …
* Discussed with Luke our technical documentation, in terms of what was expected:
  + Mainly diagrams and architecture documents to explain how we intended to build the system
    - Sub-systems explanation (why we broke them down into the sub-systems, can they be run independently, etc)
    - Class diagrams and explanations
    - Use cases
    - User manual (can be with or without screenshots for the preliminary document, we chose to include them in this version)
  + DFD’s, State diagrams and other lower level diagrams not as important
* Ian suggested to Luke the idea of adding a ‘Bookmarklet’ which the user can use to automatically add a feed to News Feeder
  + Good idea
  + Added as a low level requirement

**7/5/12**

**Progress Reports**

* **Ian**
  + Has been fixing issues with the CSS (such as it cutting off heading text), had a few issues which needed a rollback
  + Started the user manual, added registration, login, account management, etc. Hasn’t been able to get too far into because of the front-end issue
* **Alex**
  + Drew up his class diagrams and explanations, needs to confirm how we want to structure the class diagrams before he finishes them
* **Aron**
  + Got SWIG working with PHP, confirmed that it will work as expected
* **Michael**
  + Finished vision document

**Agenda**

* Discussed how we wanted to present the technical diagrams
  + VISIO will be used for all diagrams (since we all have access to it)
  + Everyone responsible for their own sub-system
  + Class diagrams – name and relationships only, with descriptions and summaries of each class written up in another section (no methods and variables yet)
  + User manual – not too in depth, pictures not used for reference only to illustrate the steps. Don’t have to include screenshots for non-finished sections
  + UI design – screenshots of current design as well as mockups. No specific method for mockups, can use whatever program we see fit
  + Discussed Aron’s success with SWIG
    - Explained the problem with PHP (namespace issue)
    - Discussed how the system will be structured:
      * Model View Controller (without the views…) using factory patterns to a degree
* Passed this sub-system onto Michael for development and implementation
* Confirmed everyone knew which sections of the documentation they were doing (see last week’s documentation)

**Actions for Next Week**

* **Ian**
  + Will finish the user manual and attempt to complete the use cases
* **Alex**
  + Will finish the backend class diagrams and start on the architecture document for the backend
* **Aron**
  + Will start writing design documents for the Java Downloader Application and Web Service
* **Michael**
  + Will start design documentation for shared database library

**9/5/12**

**Agenda**

* Alex showed us his documentation progress (class diagrams and use cases)
* Luke explained how our project was evaluated in the meeting
  + Relatively few comments (good thing, means there were no problems)
  + “Potential” High Distinction
* Ian discussed our process of adding feeds to a sheet:
  + Add feeds you are interested in (added via search and pre-defined) on one screen
  + Then on another screen set the properties of each feed (e.g. number of items to show, show images, etc)
  + This was done as currently feeds are added via an input box (which does not provide the room necessary for setting the properties for all functions, unless is done one at a time)
* Add multi-language support  as a requirement
  + Low level requirement as there is a-lot of work
  + Will be based off the browser accept-language
  + Store language in database (retrieved from RSS or HTTP headers)
* Parsing HTML
  + Can’t use RapidXML as HTML is not valid XML
  + Possible solutions:
    - Use Tidy to parse HTML into XML then parse using RapidXML
    - Write our own HTML parser
  + Main problem with using Tidy is that it is an executable, which we will be calling via system calls in the document
  + Essentially all we need is a ‘DOM constructor’, we could then run our own parser to extract content
  + Ian and Alex said they would look into finding another solution, this time from a C++ source to adapt from, and failing that would begin work on a custom DOM constructor
* Explained patch and match methods to Luke
  + By default:
    - Using the RSS summary look for the body content
    - Grab the element this content is in
    - Extract all content in this element
  + Not robust enough, it will be fixed by using ‘rules’

**14/5/12**

**Progress Reports**

* **Ian**
  + Company Website complete overhaul, now complete
  + Tested new HTML parser (HTMLCXX)
  + Working on User Manual
* **Alex**
  + Finished use Cases for backend
  + Finished class diagrams for backend
* **Aron**
  + Working on class diagrams for Downloader Applciation
* **Michael**
  + Working on class diagrams for Shared Database Library

**Agenda**

* Ian got our opinions on the new company website
  + Positive reception, only minor changes such as separating out some tabs and adding documents
  + Discussed what settings are available for a content placeholder
    - What format/properties do you wish to display
      * Feed
        + Feed
        + Title (ALWAYS DISPLAYED)
        + Images
        + Body
        + Summary
        + Comments
        + GPS
        + Author
        + Posted By
        + Date
        + Link to original item
      * Weather
        + Temperature metrics (C vs F)
        + Days in forecast
        + Location
* Censoring
  + Filtering and Censoring
* Number of columns
* Content Placeholder Title
* What feeds are inside it
* Added tables to database schema:
  + Layout
  + GroupPlaceholder
  + ContentPlaceholder
  + (all different types of content placeholders)

**Actions for Next Week**

* **ALL**
  + Continue working on documentation for week 13
* **Ian**
  + Finish the user manuals
* **Alex**
  + Write architecture for backend
* **Aron**
  + Finish class diagrams for downloader application
* **Michael**
  + Finish class diagrams for shared database library and write architecture for library

**16/5/12**

**Agenda**

* Short meeting with Luke this week
* Focused on explaining how our custom layout system was going to work
  + First admins/subscribed users can create layouts, essentially tables with set of rows and columns
  + Then can add group placeholders to these layouts, these will be the areas content is placed in
  + Then ALL users can add content placeholders into these group placeholders, these content placeholders will be the actual location content is displayed
    - Content placeholders can come in a number of different types (news feed, weather, forum, etc)
    - Each type has its own specific settings based on what is being displayed
    - As a user creates a content placeholder they can set a number of options, such as how content is displayed (e.g. display images), the number of columns in that content placeholder, etc
* Ian raised some issues with the idea of the group placeholders
  + Argued they were just an additional step without any real benefit (as settings were now in the content placeholders instead), and was adding additional complexity for the user
  + Aron and Michael explained why it had been broken up into content AND group placeholders
    - Done to separate admin vs standard user actions, as standard users will never see the layout or group placeholder steps. When modifying, a standard user will only see a sheet with predefined areas they can drag content placeholders into
  + Ian agreed this was an acceptable solution
* Talked to Luke about some Technical Documentation choices
  + Using DFDs – Only if deemed beneficial to the development
  + Using State Diagrams – Again only if deemed beneficial

**21/5/12**

**Progress Reports**

* **Ian**
  + Finished User Manual
* **Aron**
  + Finished class diagrams for downloader application
* **Michael**
  + Finished vision and database library class diagram
* **Alex**
  + Finished Technical Documentation for backend
  + Made one of the project lab PC’s externally available

**Agenda**

* Went through what documents we needed to finish one last time:
  + Vision – Michael **DONE**
  + System Overview
  + Subsystems:
    - Downloader App  – Aron
    - Front end (PDF generator included) – Ian **DONE**
    - Back end – Alex **DONE**
    - Shared Library – Michael
    - Web Service – Aron
  + Requirements – **DONE**
  + Use Cases (diagrams, descriptions) – ALL **DONE**
  + Architecture – ALL
  + Class Diagrams – ALL
  + UI Design – Ian **DONE**
  + DFD – Talk about with Luke **Decided against, no benefit at the moment**
  + State Machine Diagrams – Talk about with Luke **Decided against, no benefit at the moment**
  + Interaction Diagrams – **DONE**
  + USER MANAUL – IAN and Aron **Front-end manuals done**

**Actions for Next Meeting**

* **ALL –**Complete assigned sections of the documentation due in week 13
* No others, everyone should focus on documentation as defined above

**23/5/12**

**Agenda**

* Short meeting with Luke
* Mainly based around the technical documentation due next week and showing Luke our current versions of the Technical Documentation and User Manuals:
  + Asked if the document was reviewed by our supervisor of the committee: **Supervisor**
  + Asked if test cases/methods were to be included: **Luke said only if it is appropriate for our project, in our case it is not really as there are no ‘defined’ test cases so we will only describe the types of tests being used**
  + Luke suggested:
    - We remove the screenshots from the administrators manual (since they will know how to use the system), possibly add a short reference sheet for the user manual since it is too long
    - Add links/references to the technologies being used
    - Expand the technologies and developer tools sections
    - Add the algorithms currently left out, namely the priority and search algorithms
    - Minor changes:
      * “Administrator Manual” name change
      * Resolution information in manuals
  + Overall said our documentation was looking good, as soon as our placeholders were filled out (vision, diagrams, etc)

**28/5/12**

**Progress Reports**

* **Ian**
  + Updated user manuals as per Luke’s suggestions
  + Updated technologies section
* **Alex**
  + Proof-read technical documentation for back-end
  + Expanded third-party libraries section
* **Aron**
  + Finished class diagrams for downloader application and added them to the document
* **Michael**
  + Finished proofreading vision and added to technical documentation
  + Added shared database library class diagrams and descriptions
  + Added description of priority algorithm

**Agenda**

* Everyone had done good work on the documentation
* IMPORTANT point: Ian had found out the Bing API (used for our search) was being moved to the Azure Marketplace. The actual change had no major implications (simply changing the location the API calls were pointing at). However, a major change was that the system is now PAID. Currently our system can use the free 5000 queries a month, but if we needed to expand we would have to pay for use.
* Again with the deadline creeping up focus was on technical documentation. We had the following changes to be made:
* 1.       Remove duplicated sections and place them in the right part  
  2.       Add overall architecture section (explaining repository pattern and why it’s being used)3.       Add descriptions of architectures/patterns used for each sub-section  
  4.       Update the diagram of all sub-systems (interaction diagram)5.       Add new tables to data-dictionary  
  6.       Update database schema (ERD)

7.       Add protocol section to technical document  
8.       Add iteration plans

9.       Add testing plans  
10.   Use case diagrams for Front End

11.   Use case diagrams for Downloader Application  
12.   Descriptions and architecture of Downloader Application

13.   Proof-read

**Actions for Next Week (or more correctly the end of this week)**

* **Ian** – 5, 6, 10
* **Alex** – 7, 9
* **Aron** – 11, 12
* **Michael** – 1, 2, 4, 8
* **ALL** – 3, 13

**30/5/12**

**Agenda**

* This meeting again focused on our technical documentation and user manuals.
* The previous weekend we had given Luke our documentation, he gave us the following feedback:
  + Add News Feeder logo to documents
  + Fix the page formatting
  + Indexes and tables of contents are currently broken
  + User Manual
    - Fix table of contents
    - Variable image quality, due to issues with Paint
    - Add Description of HTTPS (possibly add security section)\
  + Admin user manual
    - Fix table of contents
    - Add Note to make admins read ‘Standard User Manual’
    - Role of administrator (in the introduction)
    - Add trouble shooting guide. What could go wrong? (e.g. if the crawler fails, why would you want to stop the crawler)
  + Technical documents
    - ERD Top Row Table move left
    - Change user description to say “there are 5 types of users” (currently it says there are only 5 users…)
    - Pages 41 and 42 images (mock-ups from layouts), low quality due to Paint
* As we are not meeting for the next few weeks, we felt it was necessary to assign our tasks (both to complete the technical documents and continue on)
  + **Alex**
    - HTMLCXX research
  + **Ian**
    - Modify document in regards to Luke’s feedback
    - Add web master use cases
  + **Aron**
    - Add Use cases for downloader application
    - Add Short description of interaction between web service and downloader application
    - Add Class diagram for downloader application
    - Add descriptions of the downloader application
    - Add the architecture of downloader application section
  + **Michael**
    - Modify document in regards to Luke’s feedback
    - Add interaction section for database library

**20/6/12**

**Progress Reports**

* No real progress due to exams being on

**Agenda**

* A quick meeting with Luke, mainly to discuss what parts we should be working on over the holidays, as well as to decide if we should be holding meetings during the holidays.
* Ian explained an issue with the current site theme, the CSS is still being buggy and not displaying some sections (such as titles) as wanted. We will be working on this over the holidays.
* Discussed the work for each team member over the break (see Actions for next meeting section below).
* Discussed meetings over the break, decided we did not require any until session starts back up as there would not be much progress to comment on.
* Created a new email thread for communication.

**Actions for Next Meeting**

* **Ian**
  + Help Alex with patch and match
  + Fix CSS for front-end
  + Feature implementation (any major features we had forgotten0
    - Layouts
    - Printable view
* **Alex**
  + Investigate HTMLcxx further
  + Patch and match algorithm and implementation
  + Integrate queuing
* **Aron**
  + Develop basic Java application
    - Get info from web service
    - Build web service
  + Implement queuing for backend
* **Michael**
  + Implement algorithm for backend queue priority
  + Write database library

**26/7/12**

**Progress Reports**

* **Alex**
  + Fully researched HTMLcxx
  + Finished patch and match algorithm, currently in the process of testing
* **Ian**
  + Working on the template engine
* **Michael**
  + Working on shared database library
* **Aron**
  + Working on downloader application

**Agenda**

* Aron not present at the meeting due to him being in Sydney
* Discussed and resolved some issues with the databases
  + Some names not plurals
  + Using text for avatar images instead of blobs
* Ian explained the database would need to be migrated due to his VPN providers complaining about excessive storage usage. Decided we would move it to Alex’s server.
* Alex explained how his improved patch and match algorithm was working
  + Checks if the content is provided in ‘content-embedded’ element, if it is do not attempt the patch and match
* Explained the constant value added that is used to calculate if the current element contains the main body of the article
* Discussed how many levels to search up the tree for relevant images
  + Depends on content and will be tested, will only retrieve at most 3 images
* Decided on new time for group meeting

**Actions for Next Meeting**

* **Alex**
  + Test patch and match algorithm thoroughly
  + Prepare server for database migration
* **Ian**
  + Update the template engine to handle more than 2 columns and give it the ability to switch between different column layouts
* **Michael**
  + Continue working on the shared database library
* **Aron**
  + Continue working on the downloader application

**27/7/12**

**Agenda**

* First meeting with Luke since break
* Luke gave our marks for the preliminary documentation and diaries
* Luke told us he would bring the comments he had regarding the documentation in next week.
* Aron showed his progress on the Java downloader application
  + Login screen and sheets look the same as the online version (using the dark theme)
  + His system currently more advanced than the online version, supporting layouts with both rows and columns, instead of just columns like the online version
* Alex and Ian brought up a possible issue with the patch and match algorithm. Currently was searching based off of the amount of content (characters) in a section to determine the body.
  + Ian suggested this could be an issue on pages with large ads, so instead it should search for more direct matches with the content.
    - These matches would be in the center of the RSS, as the start and end can contain pointless text (such as extra tags)
  + Alex coded up the original idea and tested it. The original algorithm worked for his examples, but he is in the process of testing more thoroughly to ensure it works as intended.
* Discussed the week 6 prototype with Luke
  + Doesn’t have to display full functionality only really the high level requirements
  + Only has to work for a fixed amount of content (e.g. we could get it working for only a few feeds, doesn’t have to be completely successful in all cases)
  + Luke explained he will be away in week 6, so we will be presenting the prototype probably in week 7.

**2/8/12**

**Progress Reports**

* **Ian**
  + Continuing work on layouts
  + Finished drag and drop interface, now a matter of storing the layout and tracking the elements
* **Alex**
  + Improved RSS parser completed (RSSXParser)
  + Investigating ATOM feeds
* **Aron**
  + Continuing work on the downloader application
  + Finished the general application interface
* **Michael**
  + Continuing work on database library
  + Finished the first model version of the library, now attempting to SWIGify it

**Agenda**

* Aron demoed the downloader application to us
  + Showed the ability to use content placeholders
  + Showed the ability to login to the newsfeeder service
  + Showed the SQL-lite database being used to store the offline information
* Ian displayed the drag and drop interface
  + Allows the adding of feeds
  + Allows the adding of content placeholders
* Alex brought up a major issue with the old patch and match algorithm (and hence why work was started on improving the old one)
  + It has problems regarding the size modifier being used to find the correct container element
  + At the moment it works for a lot of the articles, but when the description text is for an element not in the main section, it either:
    - Pulls out the content from the wrong section (such as a description section on the side)
    - Finds the largest section and pulls out the incorrect content (such as including the ads)
* Ian suggested a change to this, being instead of using a size modifier, it uses purely a matching system, and it ranks based on how well certain sections match.
* It was decided both schemes would be compared by Alex writing up his and Ian writing up his, then comparing the success rate on bad sites.
* Both are to be tested next week

**Work For Next Meeting**

* **Ian**
  + Investigate his new method for patch and match
  + Continue working on the edit layouts section
* **Alex**
  + Investigate and add functionality to parse ATOM feeds
* **Aron**
  + Continue work on the downloader application
* **Michael**
  + Continue work on the database library

**3/8/12**

**Agenda**

* Meeting with Luke
* Demonstrated the downloader application
  + Luke was impressed with how similar it looked to the site
  + Asked about the ability to generate PDFs
    - We explained this is actually done by the front-end using a request to the PHP PDF generator.
* Demonstrated the edit mode
  + Luke was again impressed
  + He suggested that the descriptions of the different types of content (i.e. feeds and weather) be clearer
  + Asked where the settings were
    - Explained they weren’t implemented yet
    - Also, that many of the settings (such as Adult filter) would be profile settings rather than content placeholder settings
* Asked Luke for his opinion of the new patch and match method
  + He said there was no real reason why it couldn’t work
  + Confirmed the best way to tell which was best would be to indeed test it
  + Suggested that we could perhaps combine the algorithms to use one then the other (if the first wasn’t successful)

**9/8/12**

**Progress Reports**

* **Ian**
  + Worked on layouts
  + Finished the ability for the front-end to display a sheet’s content in a defined layout view
  + Started work on a weather widgit
* **Alex**
  + Implemented the ATOM parser
* **Aron**
  + Continuing work on the downloader application
  + Starting to transfer queuing simulation to real backend
* **Michael**
  + Continuing work on database library
  + Having trouble with SWIG

**Agenda**

* Ian demonstrated the content display in a layout
  + Everyone agreed it looks good
  + Only change was allowing the user to change the content placeholder names (to reflect what they want to show there)
* Alex brought up a major issue with the database library
  + In its current state it can’t handle NULLABLE variables
  + This is because it doesn’t allow for nulls to be passed into the functions (as instead it is using native types, which are not nullable)
  + Michael suggested he changes it to use pointers for all variables (thus nulls can be passed in when an attribute is to be set to null)
    - The main problem with this is that all the pointer memory must be managed
    - Also, SWIG may have an issue with passing pointers
* Michael agreed that he would update it for next week
* In regards of the Patch and Match algorithm
  + Ian hadn’t had time to work on the new algorithm, so he suggested we use Alex’s for now
  + Unfortunately, this does not solve the current issues
  + Ian suggested that for now we focus on just retrieving some content (whether its valid or not)
    - This means that, if the Patch and Match fails, instead of returning nothing (as Alex is currently doing) we return the original description text
    - This is what is being done currently if the description text is a complete match for the whole article (done by looking at the RSS tags)
    - Alex agreed this was suitable and was going to update the system to do this

**Work For Next Meeting**

* **Ian**
  + Work on the edit layout section so it can save layout information back to the database
* **Alex**
  + Update the patch and match to return description content on failure
  + Fix some remaining issues with the ATOM parser
* **Aron**
  + Continue work on the downloader application
* **Michael**
  + Update database library to use the pointers

**10/8/12**

**Agenda**

* Short meeting with Luke
* Demonstrated the edit mode again
  + Luke liked the
  + He also suggested we add the ability to change the content placeholder titles
  + He liked the addition of the weather widget, but suggested we add the ability to add multiple widgets to display the temperature in different locations (not just the local one)
* Asked Luke for his opinion on our compromise regarding patch and match
  + He said as long as it works in the majority of cases it will be fine
  + Confirmed that the returning the description worst-case is suitable (rather than no content)

**16/8/12**

**Progress Reports:**

* Ian
  + PDF Generation working (aside from some layout issues)
  + CPH news content integrated
  + Weather widgit working
  + Edit is propagating search field in the weather widgit
* Alex
  + Reviewed queuing model
* Aron
  + Queuing model
  + Split up the code in backend
* Michael
  + Database library

**Agenda**

* Michael discussed the fact that diaries were coming up so they needed to be finished
* Discussed the Java Downloader Application
  + Showed us how it was handling the layout centric items
    - Content Placeholders, rows, columns, are supported
  + Discussed what the default theme should be for the downloader application (light or dark)
    - Currently using dark, decided to change to light.
* Discussed the progress on the queuing model since last week
  + Threads now correctly pooling as had been done in the simulation
  + Aron displayed another simulation of the queuing
    - Handled 500,000 items with no problems (maximum limit on our project for the moment)
    - Majority of the downtime is going to be due to the parsing for content and the actual network communication (latency, etc)

**Work for next week**

* Ian
  + Finish edit section
  + Bing search
* Alex
  + Finish ATOM parser
  + Fetching images
  + Improve stripping
* Aron
  + Finish queuing
  + Changing to white background
* Michael
  + Finish DB Library

**17/8/12**

**Agenda**

* The main issue discussed with Luke was ideas about the trade-show as a form needs to be filled out by week 7 (and Luke is away in week 6)
  + Discussed the items needed for the booth
    - Tables and boards for displaying posters provided
    - Added the requirement of 2 computers (the ones we were using in the lab)
      * One for backend to run on
      * One for the people to use at trade-show
      * (Logic behind this is no-one should be able to touch and potentially break the backend)
* Brought up the poster, Luke said it should be an attempt to market our solution (not an explanation of what we have done)
  + So bright and colourful
  + Not much text (to ensure we don’t overwhelm the user)
* Ian and Michael brought up the idea of a video presentation (to be used in the FAQ section of the website and at the trade-show)
  + Essentially is to act as a demonstration video of how to use newsfeeder (create sheets, add feeds, etc)
  + Also a good idea since Luke’s feedback suggested we could work on the user manual content.

**24/8/12**

**Progress Reports**

* **Ian**
  + Edit section has been finished, sheets can now be edited and saved to the database.
* **Alex**
  + ATOM parser has been finished
  + Image thumbnails being generated after page crawl
* **Aron**
  + Queuing model is complete and integrated with the back-end
* **Michael**
  + Database library in its C++ form is complete

**Agenda**

* Quick meeting today since Michael couldn’t stay till the end.
* Alex demonstrated the back-end’s ability to generate images
  + Showed an example of a parsed image (complete with letterboxing)
* Alex brought up the issue of image extensions, we are supporting:
  + PNG
  + GIF (if it’s an animation we take the first frame)
  + JPEG
  + These should be fine as they are the most common formats, anything else will be thrown away during parsing
* Discussed the reasons why the SVN was rolled back:
  + Alex was having issues because Aron had separated all classes into their own files (currently some files and headers contain more than one class)
  + Alex was having issues with the namespaces
  + Change was from 158 -> 155
  + Only real addition lost was the database library which was re-added later that week.
* Michael discussed his progress with the database library
  + Complete in C++ form
  + Backend is to be hooked up to it as soon as possible
  + Other sections to be hooked up once it was SWIGified

**Work For Next Meeting**

* **Ian**
  + Fix PDF generation
  + Fix searching to use the new API (Azure now)
* **Alex**
  + Implement the database library
* **Aron**
  + Work on the downloader application
* **Michael**
  + Implement the database library with the backend
  + SWIGify the database library

**25/8/12**

**Agenda**

* A short meeting with Luke this week
* Discussed the progress since last week
  + Image generation (Luke confirmed it was alright if we assume these are the standard image types)
  + Database library progress
  + Showed off the fully functioning edit mode
* Continued the discussion from last week about the trade-show materials
  + Luke confirmed we would be able to request a router (or hub) to provide a networked connection between the 2 machines (he thought it had been done before)
  + Suggested we confirm this with Saff before we rely on it (Ian sent off an email)
  + Alex offered himself up to do any graphics related work regarding the poster (since he has the most experience regarding this)
* Ian and Michael asked Luke if it would be possible for him to narrate the user-guides discussed last week, Luke said it wouldn’t be possible as it would be a conflict of interest.
* Asked Luke when he would have a chance to look at our diaries, he said probably week 7

**30/8/12**

**Progress Reports**

* Ian
  + Has no internet currently, so is unable to do much work on the project
  + Continuing development of layouts
* Alex
  + Integrated database library with backend
* Aron
  + Continued development of web-service
* Michael
  + Integrated database with backend

**Agenda**

* Discussed the changes made to the backend to implement the database library
  + Was relatively easy to implement, not many changes had to be made (due to only having a few database calls, and also through the use of stubs when designing the crawler)
  + Only major changes were:
    - Added an extra class to represent a queue item (essentially a feed but with less data to reduce the load on the queue)
    - Updating the database library to use the configuration file used by the backend.
  + Michael brought up diaries. Due this week, mostly done but we need to publish a few which are only drafts at the moment.
* Discussed prototype coming up in week 9
  + Decided we would mainly focus on getting the base and high level requirements done (any medium levels are an added bonus)
  + Base Level
    - Front-end Database Interaction – done
    - Login – needs validation
    - Manage Users – done
    - Add Feeds – done
    - Backend Database Interaction – done
  + High Level
    - Account Registration – needs validation and some more work
    - Search for feeds – working, needs to be added back into the project
    - View sheets – done
    - Additional Media Images – done, needs to be tested (backend crawling for images)
    - Manage Sheets – done
    - Manage Feeds – done
    - Backend Monitoring – needs to be tested again
    - Crawl RSS Feeds – done
    - Crawler Priority – done
    - Parse RSS Feeds – done
    - Advanced Parsing Techniques (patch and match) – done, needs testing
    - Logging – NOT done

**Work For Next Week**

* Ian
  + Continue working on the layouts
  + Add validation to the front-end site (especially for login)
* Alex
  + Ensure the high level requirements/features of the backend are running as expected (testing and performance)
* Aron
  + Continue working on the downloader application
* Michael
  + Help Ian with validation checks on front-end
  + SWIGify Database Library

**6/9/12**

**Progress Reports**

* Ian
  + Finished fixing up layout issues
* Alex
  + Testing performance of the parser
  + Found good example feeds for testing
* Aron
  + Continuing work on downloader application and web service
  + Continuing work on priority queue
* Michael
  + Attempting to SWIGify database library

**Agenda**

* Again the focus of this meeting was on what requirements were met
* Ian showed us the working layouts
  + No longer having issues with the storing of the content placeholders
  + Fixed issue with the weather no displaying in the right hand column
  + Fixed issue storing the order of the content placeholders (was using the order in which they were added, not arranged)
* Alex showed some examples of parser issues
  + ABC
    - Causes issues because the URL for articles is not unique…
    - Can be fixed by making the unique constraint on TITLE AND URL
  + Pre tags
    - Parser is stripping white-space so these tags no longer hold the white space, meaning the text is badly arranged
    - Can fix by keeping whitespace inside these tags
  + Style attributes on strange tags gets stripped
    - Not much we can do about this, especially since styles set here may ruin our own display
  + Gizmodo feeds
    - Some items have links in the description, these links are getting parsed into the main body (thus it shows the “HOME FACEBOOK ETC” in the sheet)
    - Again, not much we can do as we have no real way of telling when this is happening
  + Alex resolved to fix what issues he could for prototype

**Actions For Next Week**

* Ian
  + Continue working on high level requirements that aren’t fully implemented
* Alex
  + Fix parser issues discussed (if possible)
* Aron
  + Finish up work on priority queue
* Michael
  + Work on SWIG

**7/9/12**

**Agenda**

* Meeting with Luke
* Main focus on asking Luke what we should be presenting for prototype
  + General demonstration of our progress so far
  + Attempt to get our Base AND High level requirements done, seems better if we have done all the ‘required’ work for the prototype
  + Focus on the requirements mainly
    - I.e. Features we have implemented
    - Patch and Match
    - Layouts
    - Logging
    - Crawling
    - Etc
  + Marketing Strategy is also important
  + Don’t have to really discuss the development unless it is relevant as to why we have done this (or we are asked about it)
  + Don’t need to prepare anything special for the demonstration (no need for slide show or anything)
* Discussed with Luke whether or not we would have the demonstration with just him or others
  + Said probably with him and Daniel
  + Offered to do together or separately
    - Decided on separate as it gives us an opportunity to test our script out on Luke

**13/9/12**

**Progress Reports**

* Ian
  + Implemented logging on the front-end
  + Re-added the admin-service back again
* Alex
  + Fixed issues brought up about the parser last week (the ones that could be fixed)
* Michael
  + Finished SWIGifying database library (not tested)

**Agenda**

* Aron wasn’t available for this meeting as he was in Queensland for work
* Quick meeting this week as Michael had to leave early
* Again the meeting focused on getting all functionality down for prototype
  + Added in logging
  + Admin service now tested
  + Parser fully tested
  + Image generation tested
  + Fixed validation issue with logon (logging on even if not a user)
* Discussed the main part still needing to be implemented for next week is the priority queue
  + Needs to be linked with the database library AND linked with the parser
* Ian told us about an issue with the search AJAX calls
  + Apparently the system is stalling until all results have been returned
  + Eliminates the whole point of the AJAX…
  + Possibly an issue with AJAX and Memcache being used together

**Actions For Next Week**

* Ian
  + Fix issue with AJAX
  + Continue working on any validation issues
* Alex
  + Integrate the crawler and parser with the priority queue
* Michael
  + Integrate the shared library with the priority queue

**14/9/12**

**Agenda**

* Meeting with Luke
* Focus of meeting on the prototype next week
* Gave Luke a run-through of all the implemented functionality
  + No real issues were found
  + A validation issue regarding adding an avatar for a user, setting as NULL instead
* Luke said the presentation was good (10 out of 10 of course)
* Discussed with Luke how we could demonstrate the crawler and the parser
  + Luke suggested we use debug statements to say what is being crawled and what is being added
  + Currently have debug statements for testing, so easy to convert
  + Also suggested clearing the database so the first crawl actually populates the sheets

**20/9/12**

**Progress Reports**

* Ian
  + Attempting to get the searches AJAX calls working correctly
  + Fixed multiple validation issues
* Alex
  + Finished fully implementing the queuing system for the back-end
* Aron
  + Working on demo of downloader application
* Michael
  + Assisting in the implementation of the queue system, and integration with the shared database library

**Agenda**

* This meeting mainly focusing on getting the prototype fully working for the product demonstration
  + Discussed what we should present for the prototype
    - FEATURES (especially ones that aren’t in other systems)
    - Priority queue
    - Patch and Match
    - Web Service
    - Marketing Strategy
* Discussed how the priority queue had been implemented and any issues that had been had
  + Compilation issue with shared library (found to be issue with ordering of the library links)
  + Issue when the number of threads is equal to (or outnumbers) the number of feeds, causes segmentation fault as thread is trying to retrieve a feed that is not there
* Added content for parsing during the demonstration
  + Simple wordpress page on the local machine
  + Truncated database to illustrate crawling
* No meeting with Luke this week, as he is watching our demonstration in that time instead.
* Also, Luke is away in all other weeks, so no meetings with him there either
* No meeting in holidays, focus on other subjects then

**Actions For Next Week**

* ALL
  + Fix any issues found in the demonstration
  + Have fun in the break! (catch up with all other work)

**4/10/12**

**Progress Reports**

* Alex
  + Finished up the statistics chron-job for the back-end
* Michael
  + Finished up changes to the database library for the chron-job to access

**Agenda**

* Small group meeting this week
  + Ian, sick and unable to come to uni
  + Aron, in Sydney for work
* Main focus of this week was the poster for trade-show
* Discussed possible ideas for the poster
  + Standard looking poster, large screenshots, flashy buzz-words, etc
  + Poster looks like a news-paper
  + Poster looking like a sheet from Newsfeeder
* Decided the poster looking like a news-paper (the one like news feeder was too difficult, also the background would have either been white (boring) or dark grey (to hard to read))
* Discussed what content to put on the poster
  + Basic introduction paragraph we had been using for other documents
  + Lists of good features
  + Compatibility (mobile and browser)
  + Screenshots of the front-end and downloader application
  + Alex suggested that we add an image similar to those in Apple advertisements, with multiple different devices all showing the application being used.

**Actions For Next Week**

* Alex
  + Continue working on downloader application and web service
* Michael
  + Implement the fixes for the shared library

**10/10/12**

**Progress Reports**

* Aron
  + Finishing up both downloader application and web service
* Michael
  + Finishing up content for tradeshow poster

**Agenda**

* Small group meeting again this week
  + Ian, away in Sydney for work
  + Alex, wasn’t available during the meeting time
* Focused on the progress of the downloader application and web service
  + Web service is REST
  + JSON objects used for procedure calls
* Discussed and fixed some issues with the SWIG shared library
  + Wasn’t correctly adding the namespace
    - Had to hard-code namespace in manually)
  + Compilation issue with the existing SWIG makefile, libraries were being imported in the wrong order
    - Similar issue had occurred with back-end compilation
    - Fixed by changing to correct order

**Actions For Next Week**

* Aron
  + Continue working on downloader application and web service
* Michael
  + Implement the fixes for the shared library

**18/10/12**

**Progress Reports**

* Ian
  + Finishing up polish on
* Alex
  + Finished off poster and printed
* Aron
  + Finishing up web service for downloader application
* Michael
  + Writing up slideshows for tradeshow and presentation
* ALL
  + Documentation finalising

**Agenda**

* Main focus of this meeting was what we had left to do for the project
  + Final polish on all sections
  + Altering class diagrams to the new system spec
  + Decided to change the way we do them
  + First a diagram of the sub-system’s classes as a whole
  + Then individual classes showing off class members
* Other main focus was the slideshows we had to prepare
  + One for tradeshow, simply 6-7 slides long
    - TITLE
    - INTRO
    - ARCHITECTURE
    - FRONT END
    - BACK END
    - DOWNLOADER APP
    - DATABASE
  + Other slides less fancy, but much more informative
  + Asked Daniel for advice on what to put in them
    - Development model
    - Road-blocks
    - Special Features
    - Technologies
  + Decided to use basic dot-point format for slides to act as guidelines for our presentation

**Actions For Next Week**

* ALL
  + Finalise any outstanding items
  + Make documentation up to date
  + Prepare slides for presentation