Ryan O’Connor

Computer Science Project

Dr. Cynthia Howard

February 12, 2015

Work Breakdown

**Tasks/Thoughts**

1. Design application
   1. Class Diagram
      1. Where/how does client data come in?
      2. How are scheduled web crawls triggered?
      3. How will it communicate with the database?
      4. Where is data packaged in JSON?
   2. Data flow
   3. Written in C# .NET
      1. Determine language interoperability with crawler (Python)
2. Design web crawler
   1. Class Diagram
      1. How will concurrency be handled?
      2. How will it handle errors?
   2. Data flow
   3. Written in Python
      1. Determine language interoperability with application (C#)
3. Design database
   1. Data document structure (MongoDB)
      1. How will each BSON/JSON document be structured?
         1. Users
         2. Crawl Data
         3. Scheduling
         4. Etc.
      2. How to breakdown collections
   2. Security measures
      1. User password encryption
      2. Access control setup
4. Design mobile application
   1. User interface
      1. Design mockups
      2. State machine type diagram
   2. Class diagram
      1. In-app browser
         1. Highlight selected HTML
         2. Display overlay controls
      2. Link feed
      3. Left menu – Feed folders
      4. Modify feeds
      5. Settings/Control panel
      6. Accept incoming data
      7. Send/Package outgoing JSON data
      8. Storing/Accessing local data
5. Develop web application
   1. Bottom-up approach
      1. Develop important functions/methods first to determine difficulties or non-obvious requirements
         1. Query database
            1. Write
            2. Read
            3. Modify
         2. Communicating with client
         3. Data security – encrypting user passwords
   2. Break down points of fault into small and testable functions/methods
   3. Develop by function
      1. Create user
      2. Modify user
      3. Manual request by user
      4. Handle crawls
      5. Handle errors
      6. Encode/Decode data in JSON
      7. Send data to user
      8. Authenticate user
6. Develop web crawler
   1. Bottom-up approach
      1. Develop important functions/methods first to determine difficulties or non-obvious requirements
         1. Link extraction by tag
         2. Searching text for keywords
   2. Break down points of fault into small and testable functions/methods
   3. Develop by function
      1. Make proper web request
      2. Get links by tag
      3. Check for updates
      4. Handle errors
      5. Fix url’s
      6. Search for keywords
      7. Grab images
      8. Check site for crawl policies
7. Develop/setup database
   1. Install database and database tools
      1. MongoDB
   2. Set up access control
   3. Initialize with test users and test crawl data
8. Develop mobile application
   1. Bottom-up approach
      1. Develop by function
         1. In-app browser
            1. On-top of site overlay controls
            2. Extract tag for area touched by user
         2. Handle incoming data
         3. Send data
         4. Encode/decode data in JSON
         5. Handle errors
         6. Populate Feeds

**Milestones**

Feb 22nd – Design Finished

* Web crawler, database, Web app, and Mobile app all designed
* Web crawler development begins, database set up begins fulltime

March 15th – Web crawler developed and database setup

* Crawler functions and handles faults, database accepts data like designed
* Web application development begins fulltime

April 5th – Web application developed

* Web application handles concurrent user connections and multiple crawlers
* Mobile application development begins fulltime

April 26th – Begin testing

May 3rd – Mobile application developed

* Mobile app is fully functional and communicates with server
* Web crawler, Web app, and Mobile app all pass testing and are functional

**Tasks – Overall Process**

|  |  |  |
| --- | --- | --- |
| **Tasks** | **Time to Complete** | **Dependencies** |
| 1. Design Web Crawler | 4 hours |  |
| 1. Design Database | 1 day |  |
| 1. Design Web Application | 1 day | B, A |
| 1. Design Mobile app | 4 days | C |
| 1. Develop Web Crawler | 2 weeks | A |
| 1. Develop Database | 1 week | B |
| 1. Develop Web Application | 3 weeks | C, E |
| 1. Develop Mobile app | 4 weeks | D |
| 1. Test Crawler | 4 days | E |
| 1. Test Web Application | 4 days | G |
| 1. Test Mobile Application | 4 days | H |



**Tasks - Design Process**

|  |  |  |
| --- | --- | --- |
| **Tasks** | **Time to Complete** | **Dependencies** |
| 1. Crawler Class Diagram | 2 hours |  |
| 1. Crawler Data Flow | 2 hours | A |
| 1. Database Document Structure | 1 day |  |
| 1. Application Class Diagram | 0.5 days | A, C |
| 1. Application Data Flow | 0.5 days | A, B, C |
| 1. Mobile App UI design | 3 days |  |
| 1. Mobile App Class Diagram | 1 day |  |
| 1. Mobile App Data flow | 2 hours | G |

**Tasks - Development Process**

Each child list item (1, 2, 3, 4, etc.) is associated with its above parent task (A, B, C, etc.). When indicating child list dependencies, only other sibling items are being referred to (or other parent tasks).

|  |  |  |
| --- | --- | --- |
| **Tasks** | **Time to Complete** | **Dependencies** |
| 1. **Develop Web Crawler** | 2 weeks |  |
| 1. Make web request |  |  |
| 1. Get links by tag |  | 1 |
| 1. Check for crawl policies |  | 1 |
| 1. Handle errors |  | 2, 3 |
| 1. Fix URL’s |  | 2 |
| 1. Search By Keywords |  | 2 |
| 1. Check for updates |  | 2, 5 |
| 1. Grab images |  | 7 |
| 1. **Develop Web App** | 3 weeks | **A, C** |
| 1. Create user |  |  |
| 1. Modify User |  | 1 |
| 1. Manual Request |  | A |
| 1. Handle Crawls |  | A |
| 1. Handle Errors |  |  |
| 1. Encode/Decode in JSON |  |  |
| 1. Send data to user |  | 1, 6 |
| 1. Authenticate user |  | 1 |
| 1. **Setup Database** | 1 week |  |
| 1. Install database and tools |  |  |
| 1. Set up access control |  | 1 |
| 1. Initialize with test data |  | 1 |
| 1. **Develop Mobile Application** | 4 weeks |  |
| 1. In-app browser interface |  |  |
| 1. Implement User Interface |  |  |
| 1. Handle incoming data |  |  |
| 1. Send data |  | 5 |
| 1. Encode/decode in JSON |  |  |
| 1. Handle Errors |  |  |
| 1. Populate Feeds |  | 2 |