Project Progress Report

Hobby Horse – A Social Data Mining Project

Submitted By:

Radhika Wadegaonkar (007491666)

&

Sulagna Bal (007510100)

Approved By:

(Advisor) Prof. Magdalini Eirinaki

Date: 02/09/2012

Project Title: Hobby Horse – A Social Data Mining Project

Team members: Radhika Wadegaonkar (007491666), Sulagna Bal(007510100)

Advisor: Prof. Eirinaki Magdalini

A. Project Meeting Minutes

Meeting 1

- Topics discussed: Researching over chosen technologies

- Action/Work items:
    - Discussed & researched about the technologies that we had to work with.
    - Developed a high level work plan for researching and assigned deadlines, so that we would not waste more than required time on just researching rather than implementing.
    - We also divided the researching technologies based on the modules we are going to implement in our project.
    - In the process of research, we found a few more technologies like Solr that could be used to make the process more efficient. Solr would be basically used to index the data and efficiently implement Full-text Searching.
    - We are still struggling between choosing ROR and PHP as the client side scripting language.
    - For the backend, we finalized to use JAVA and Cassandra as mentioned in the reports.
    - We also discussed and researched about the technologies: Maven and Spring, with respect to using them in our project as a Build tool and injection tool respectively.

- We also discussed about the various frameworks and plugins we could use to implement RESTful Web-Services in Java, our backend. We thought about using javax.rs, wink and so on.

Date/Time: 02/17/2012, 07:00pm

Mode: Skype

Location: N/A

Attendees: Radhika Wadegaonkar & Sulagna Bal

A. Project Meeting Minutes

Meeting 2

- Topics discussed: Finalizing technologies and start working on building a small prototype of the backend (RESTful Web-Services)

- Action/Work items:
    - After the discussion and research of the technologies to be used and implemented, we had tried to develop a small prototype of the backend flow of our project.
    - So, to start with we installed and configured our systems entirely with what was needed to get a small implementation of the RESTful Web-Services
    - We finalized the following technologies:
        - RESTful Webservices with javax.rs
        - Java
        - Maven
        - Spring
        - PHP (Client side scripting), that will consume the backend web-services
        - Tomcat Server
        - Cassandra
    - We finally were able to install and configure everything properly. We also created the database in Cassandra.
    - We then successfully implemented fetching and inserting data into the database (Cassandra) from Java.

- This small setup also consisted of the maven and Spring integration.
- We planned to further add Spring injections and thus modularize the whole project in terms of using the DAO classes and Manager classes instead of accessing the Database directly.

Date/Time: 02/23/2012, 07:00pm

Mode: Skype

Location: N/A

Attendees: Prof. Magdalini Eirinaki, Radhika Wadegaonkar & Sulagna Bal

A. Project Meeting Minutes

Meeting 3

- Topics discussed: Giving a brief description of the work completed by us to the professor. Also, discussing about further steps to implement and work flow approach.

- Action/Work items:
    - We almost had a small prototype of the flow of the backend that we would be using in our project.
    - We discussed with the professor, our technologies chosen and how we researched around it.
    - We were only missing the deployment of the project as a web-app correctly as it had some configuration issues.
    - However, the prototype now had Spring, Maven integration in it. To discuss this with our professor, we had developed this over just one table of our database - the User table.
    - Professor, then asked us to meet again when we were completely ready with the same and also if we were ready to discuss the logical implementations of the Recommendations algorithm that we would be using with our User Comments, etc.
    - We also discussed about meeting after 3weeks or earlier to discuss the above mentioned issues and topics.

Date: 03/23/2012

Project Title: Hobby Horse – A Social Data Mining Project

Team members: Radhika Wadegaonkar (007491666), Sulagna Bal(007510100)

Advisor: Prof. Eirinaki Magdalini

A. Project Meeting Minutes

Meeting 4

- Date/Time: 03/05/2012
- Mode: Skype
- Location: N/A
- Attendees: Radhika Wadegaonkar & Sulagna Bal
- Topics discussed: Working prototype of the application, Structure of code.
- Action/Work items:
    - We successfully built a small working prototype of our application that had one webservice running in it related to the users of the application.
    - Here, we had to make some changes in the technologies and little bit structural change. We made use of the Spring MVC framework and architecture.
    - Our code now encapsulates the MVC model at the server side.
    - We also discussed about the various use cases to decide on the web-services.
    - We also discussed about the client side implementation and how the requests and responses would be consumed on each side.
    - We planned to have a complete Object Oriented Structure with only objects being passed to and from the server and client sides.
    - However, we still had some refactoring to do with respect to the structure of the code and architecture.

Date/Time: 03/15/2012

Mode: Skype

Location: N/A

Attendees: Radhika Wadegaonkar & Sulagna Bal

A. Project Meeting Minutes

Meeting 5

- Topics discussed: Discussed about the various webservices, testing frameworks, Logical issues with implementation of the algorithms, comparison of spring v/s wink, comparison of thrift v/s hector, client side structure.
- Action/Work items:
  - We compared our technologies like the spring MVC framework with the Apache Wink framework to deploy our webservices with.
  - We finally chose Spring MVC as our final framework and started building the webservices.
  - We also had a comparison between the APIs Thrift and Hector for Cassandra to work with Java. We realized that thrift is a lot verbose as compared to Hector. After considering the fact that it is widely used and has a lot of active support community, we stuck to Thrift.
  - However, we are still considering and analyzing Hector and even other such APIs.
  - Listed all the web services that will be required as a starting point to get the application started.
  - We divided the web services amongst ourselves and are currently working on them.
  - We also developed a prototype of the client side structure using PHP and the frameworks Zend and CodeIgniter.
  - We decided to follow a DTO (Data Transfer Approach) approach throughout our project.

Date/Time: 03/27/2012

Mode: Skype

Location: N/A

Attendees: Radhika Wadegaonkar & Sulagna Bal

A. Project Meeting Minutes

Meeting 6

- Topics discussed: Choosing MySQL over Cassandra, Design and architecture of the client side code, Basic flow of our first cut, PHP Frameworks to be used.
- Action/Work items:
  - We decided to use MySQL replacing Cassandra for the prototype purposes of our system. We have proposed building our application using Cassandra as a future scope.
  - We discussed about the architecture of our client tier implementation and code. We decided to build our own platform libraries to make it simpler for the DTO.
  - We discussed and implemented a prototype of the client tier structure and designed it in a way so that we had a similar POJO kind of class on our PHP as well as JAVA Side.
  - We decided on a generic class and function that would identify the type of the incoming object for GET requests and type of outgoing object (POST requests) and then wrap it in a JSON object or as desired based on different parameters. We decided to write wrapper classes for this.
  - We also discussed about the folder structure and the frameworks we would be using for the same. We started to research with cakePHP, Zend, symphony, Joomla, codeIgnitor and so on.
  - However, since we had to develop a middle-tier and consume the web services and also request data via web services, we chose Zend for this purpose.
  - Also, since we wanted to maintain an MVC structure all throughout our system, we chose CodeIgnitor as our primary framework to layout our models controllers and views.

Date/Time: 04/06/2012

Mode: Skype

Location: N/A

Attendees: Radhika Wadegaonkar & Sulagna Bal

A. Project Meeting Minutes

Meeting 7

- Topics discussed: Libraries to be used for multiple video calling, CSS template discussion, Finishing up remaining web-services and Test Suites, Logging in the site with Facebook and using the data for temporary data mining.
- Action/Work items:
  - We discussed about the libraries that are available in PHP for implementing the multiple video-calling concept. We researched the google hangout API. However found it to be a little complex based on our needs.
  - We then found an open source library called opentok who had the multiple video calling implementations not only in PHP, but also in Java, Ruby and JavaScript.
  - To start with, their website had an example JavaScript that worked like a charm for us, and then we decided to integrate the open source library into our PHP code.
  - We also researched a lot on the free CSS templates that are available. We finally deicded on a clean 2-column based CSS template that looked perfect for our site.
  - We intergrated all these modules into our system and our basic structure was then in place. We had an entire flow, where the user could login and data was checked against the MySQL database as well as error was sent in case of incorrect login.
  - We finished up all the remaining web-services such as those related to comments, lessons and badges. With this our backend was pretty much in place and we could fetch and retrieve data from client tier through the middle tier using the Zend and CodeIgnitor frameworks.
  - We then researched about the Facebook API and how we could let our users login using their Facebook accounts. After some research, we managed to get the login

working. However, we learnt that facebook only needed a domain name and not anything like “localhost”. So in the process we learnt how to create a virtual domain for our local systems.

- Since Facebook policy states we cannot store any user information in our database, we planned to implement sessions to store the facebook data for each user when they login. So, all this data would always be on the fly, so yet usable and not stored into the database. We decided to use this facebook information about the users to fetch their likes on pages and basic interests. So, based on these we could send them suggestions of the available lesson categories and active lessons as well.
- In parallel, we discussed some of the test cases and added them for each of the corresponding classes in our maven structure. As stated earlier, since maven has n inbuilt unit testing, the tests run everytime the build is run. With some configurations, it is possible to run the build by skipping the tests, however we always ran the tests while building so we are sure that we did not change any previous data and the addition of new logic did not break the old existing logic.
- We prepared our code and cleaned it for the demo next week. We now had a small prototype where the user could login using Facebook or create new account, see available lessons, participate in lessons and involve him/herself into a multiple video calling sessions.

Date/Time: 04/12/2012

Mode: Skype

Location: N/A

Attendees: Prof. Eirinaki, Radhika Wadegaonkar & Sulagna Bal

A. Project Meeting Minutes

Meeting 8

- Topics discussed: Demo of the first-cut prototype and discussion on using the existing Java libraries to implement Data Mining
- Action/Work items:
    - We demoed our first cut prototype to Prof. Eirinaki and also discussed and explained to her our entire code structure at the data tier level, the middle tier level and the client tier.
    - Our demo consisted of the operations:
        - Login / Authentication
        - Create new Account
        - Login with Facebook
        - Browse Available Lessons sorted by pre defined categories
        - Participate in available lessons
        - Create new lesson and be an expert hobbyist!
        - Start multiple video calling session.
    - Prof. Eirinaki then discussed a lot of helpful information about the different data mining. We dicussed about the Ranking methodology that we could implement to display the average ranking of each expert hobbyist.
    - Prof. Eirinaki also informed us about the exiting data mining algorithms such as the “Association Rules” and explained in detail as to what it does and how it can be implemented from scratch or also re-use if already existing.
    - We thus decided on implementing the Apriori algorithm alongwith the Ranking methodology towards completing our final demo prototype.

- Prof. Eirinaki also mentioned that we need to have a large amount of data to implement data mining and thus need to pre-fill the database with a lot of dummy data before hand.
- Prof . Eirinaki also suggested us to implement the logic of comments, badges and feedbacks in our system as a part of our final demo. The suggestions would be based on the comments and ratings of the users on a particular lesson as well.

Date/Time: 04/23/2012

Mode: Skype

Location: N/A

Attendees: Radhika Wadegaonkar & Sulagna Bal

A. Project Meeting Minutes

Meeting 9

- Topics discussed: Implementation of comments, badges, final report submission and data mining algorithms
- Action/Work items:
  - We discussed and researched a lot about the Apriori data-mining algorithm and tried to find existing Java libraries for the same. We did find a few relevant ones.
  - We also decided on the logic that we are implementing on the badges. We decided to assign badges to the novice hobbyists based on the number of lessons he/she has attended.
  - We also discussed that we would use the simple mathematical formula of calculating an average of the total ratings received by an expert hobbyist. The user can then view the available lessons sorted by the rankings of the expert hobbyists.
  - We also discussed about the changes needed to be made to our report to work towards the final submission.
  - We then divided the remaining work with respect to implementing comments and badges logic in the data tier, middle tier as well as client tier side.

- **B. Project Deliverables**

B.1 System Design

- Completed %: 100%
- Open issues: -
- Recommended solutions: -

B.2 System Implementation

- Completed %: 70%
- Open issues: Implementing the data-mining algorithms such as Aprirori and pre-filling database with dummy data.
- Recommended solutions: Use existing Java libraries that have already implemented this.

B.3 System Testing and Experiment

- Completed %: 70%
- Open issues: Testing data-mining rules. Few validation checks on lesson creation and deletion.
- Recommended solutions: Create dummy data and write test cases for the same.

B.4. Demo

- Completed %: 50%
- Open issues: Data mining implementation on recommendation using Data mining Apriori algorithm and Association Rule.

- Recommended solutions: Understand the Apriori Algorithm and implement it in our system.

B.5. Report

- Completed % : 85%

- Open issues: Need to add more on the implementation details with the data-mining algorithms used.

- Recommended solutions: Understand the Apriori Algorithm and implement it in our system. And finally add that implementation details in the report.