Jack Davey

**Project Plan for Testing And Tailoring Cloud Storages**

**Milestones**

There are three major milestones that I hope to achieve in the duration of my MSC Project,

**Milestone 1**

This milestone concerns the implementation of eventual consistency over a standard ACID Database.

**Milestones 2**

Milestone 2 will be devoted to adding the second BASE property of eventual consistency to the system.

**Milestones 3**

Milestone 3 will be devoted to running the sample programs that I have developed and running them on a real cloud system.

**Iteration 1**

**Start Date: 8th June 2015**

**End Date: 14th June 2015**

**User Story 1**

**Allow the user to perform basic creation and deletion of data in the database.**

**User Story 2**

**Allow the user to insert, update and delete data from existing tables**

This should work by the end user being able to pass in an SQL query as JSON and have this executed by the database. The results should then be sent back as another piece of JSON. This feature will not implement eventual consistency at this point, and is partly here to allow me to ensure that I am comfortable with the tools that I am using to complete the project. Later iterations will involve the parsing of user queries so that they can be persisted in the database for certain periods of time. I will also develop a small HTML page that allows the user to type in SQL queries into a text box and then have them executed by the backend Database to ensure that the system is working as planned.

**Iteration 2**

**Start Date: 15th June 2015**

**End Date: 21st June 2015**

**User Story 1**

**Allow the user to insert, update and delete data from existing tables. Eventual consistency must be active across all of these operations.**

This story will involve the user passing in an SQL query as JSON before. I will most likely modify the format of the JSON to make it easier to parse. Rather than being sent straight to the database as before, the query will be stored in the application for a period of time. Once this time has expired, the application will then store the query in the database, following the procedure for achieving eventual consistency as outlined in my design.

**User Story 2**

**Repeat the process of adding eventual consistency for inserting and deleting data**

This story will enable insertions and deletions on the data to be eventually consistent. This is second on the list for this iteration because for deletions I will need to think very carefully about how to apply the eventual consistency algorithm with regards to stale data floating around the system. An example of this is that once a record is deleted, I will need to think about how to erase this from all the other parts of the system so it doesn’t end up getting added back in.

**User Story 3**

|  |
| --- |
| **Allow the user to synchronise the system (make everything fully consistent)** |

This should be fairly straightforward to implement, as all it involves is providing the user with a mechanism by which to make the system fully consistent without waiting for the system to do that by itself.

**Iteration 3**

**Start Date: 22nd June 2015**

**End Date: 28th June 2015**

**User Story One**

**Write a web service that allows the user to change the amount of time system waits before making everything consistent**

This web service should take a single Integer parameter. It should then change the parameter as described above. The next scheduled event should still proceed as planned however. Events should only be scheduled one at a time.

**User Story Two**

**Catch up and ensure everything is ready for the early deliverable.**

I am allowing significant time this week to ensure that everything is ready for the early deliverable which is due straight after this iteration ends.

**Iteration 4**

**Start Date: 29th June 2015**

**End Date : 5th July 2015**

**User Story 1**

**Add a web service that shows the current consistency status of the system and also add customization options**

This user story will involve writing a piece of functionality that allows the user to look at the pieces of data that are not currently within the system. For each piece of data, the web page should show what the change is, (such as updates, deletes, inserts) etc., as well as the time at which it is due to become consistent with the rest of the data.

**User Story 2**

**Perform an evaluation of the prototype implementation of eventual consistency and improve upon it**

At the beginning of the week, I will have submitted my early deliverable for the project. I would therefore like to use a significant portion of this week to evaluate the prototype for this week and improve upon it. Part of this evaluation will be done using the sample programs I am developing this week.

**User story 3**

**Research common algorithms for basic availability**

I will also spend some time this week going through all the different approaches for achieving basic Availability that I have identified in my research and choosing the model base that I would like to implement. I would aim to modify my design document showing how I would do this.

**User Story 3**

**Write some sample programs that show that the system is working**

This story will involve writing some small test applications to ensure the application is working as it should be. I plan to use code from some of the other functionalities to achieve this goal.

**Iteration 5**

**Start Date: 6th July 2015**

**End Data: 19th July 2015**

**This iteation is longer than the others, due to the fact that I will be on holiday from the 7th to the 14th July.**

**User Story One**

**Implement a mechanism inside a component of the system so it can be shutdown gracefully.**

In this iteration, I will develop a piece of code inside each system component so that it can shut itself down gracefully and any data it is holding can be transferred to another part of the system .

**User Story Two**

**Based on the Basic availability algorithms researched during the last iteration build in a complex strategy so that the whole system adapt to failures.**

This story is mainly about taking the algorithm implemented in the last iteration and implementing it in code form. It needs to cater for a wide range of situations and challenges

**User Story 3**

**Modify the sample applications produced for eventual consistency so that they run on a cloud infrastructure rather than on my system**

A large part of this story will involve working out what needs to be changed with my samples so that they can run on a cloud service. Since this might take some time, as a good bulk of the work will involve learning how to use the APIs provided by the cloud service effectively.

**Iteration 6**

**Start Date: 20th July 2015**

**End Date: 26th July 2015**

**User Story One**

**Develop a mechanism for displaying Basic Availability to end users.**

This story is about finding a way to show how available the system is to end users. It could involve providing a detailed service to report on failures, or maybe some statistics that show how much of the time the service has been down over a certain period. It is difficult to say at this point in time without knowing how exactly I will implement basic availability.

**User Story Two**

**Compare the sample applications to my system and try to gauge how effective the system is at simulating eventual consistency.**

I would do this by running the two prgrams and comparing how long they take to reach the same results. Another possibility is for me to look at the frequency by which the tow systems return inconsistent results.

**User Story Three**

**Allow the user to customize the amount of basic availability provided to the end user.**

This could be through changing the amount of data a server could have at any one time, or changing the frequency at which errors occour. .

**Iteration 7**

**Start Date: 27st july 2015**

**End Date: 2nd August 2015**

**User Story One**

**Produce some sample applications to show the system working.**

As I haven’t decided how I’m going to implement basic availability yet, I cannot say for certain what I am gong to do here. It will have similar aims to the code implemented for eventual consistency however.

**User Story Two**

**Do some research on cloud systems to see if I can figure out if I can run my sample applications on them to get the same effects**

Because of the very nature of cloud systems, I want to do some research on cloud systems to see if I can run the examples I got on them for comparison like I did with eventual consistency. Due to the nature of basic availability however, this might not be possible.

**Iteration 8**

**Start Date : 3rd August 2015**

**End Date : 9th August 2015**

**User Story 1**

**If possible modify my programs to demonstrate basic availability to run on a real cloud system.**

This may not be possible but I I can I would like to do this to help evaluate the implementation of Basic Availibility I have.

**User Story 2**

**See if I can not refine the work that I have done to see if I could match real cloud systems more accurately.**

Although this piece of work is only scheduled as one user story, It could take much longer depending on how much time I have when I get to this point in the project. The aim a this point would be to refine the results that I have got thus far to make sure it is fit for purpose with regards to conductiong research experiments or teaching.

**Any remaining time left will be used to catch up with anything I haven’t done and will be used to make sure my project is polished as can be.**