**Project Objectives**

**First Milestone– Eventual Consistency**

In this first milestone, I will look at implementing eventual consistency over an acid database. The back-end of my system will be accessed through a web service. I will also build a front-end that shows what pieces of data are currently inconsistent with the master copy of the database. A user will be able to configure the amount of consistency that the application uses. At one end of the scale, requests are just passed straight through to the database, at the other end of the scale, requests are stored in the JVM(Java Virtual Machine) for a period of time before being processed.

Implemented features will include the ability to add data to the database, modify data already in the database and be able to retrieve data. Implementation will also include a web service as part of the front-end that shows the data that is currently inconsistent with the master, and also shows how much time the user has left for a refresh.

In terms of the features expected, development of the system will enable the user to be able to perform all standard database operations. It is expected that eventual consistency is to be simulated, as described above, and furthermore to be simulated whenever one of these operations is performed. The application will also have functionality that shows the eventual consistency operations currently in the system, such as through the web service to show consistency. Parameters will be added to control the amount of consistency used in the application.

**Second Milestone – Basic Availability**

The second stage of the project would be to develop the features for emulating basic availability within the application. This would allow development to build on top of implemented features for the eventual consistency requirement, making development more straightforward. The main requirement of the feature would be that it should be possible to simulate certain parts of the system failing and have operational stability.

In order to accomplish this, the system has an robust method of recovering lost data in the event of node failure. As a result of this, the first part of this milestone will be based around developing policies to facilitate this transfer a way will be developed to visually represent component failure, possibly achieved by modifying information about the actors of the system, showing the time since an actor was last disabled.

Differing levels of system availability will be accessible to the users, allowing the user to send individual components of the system a shutdown message, as well as autonomous component shutdown.

**Third Milestone – Evaluation**

Implementation of milestones one and two, with use of developed test programs would give the possibility within the scope of this project to compare real world cloud databases with the developed system.

A major part of the iteration would be adapting the sample programs in order to allow them to run on cloud platforms such as Amazon. Test scenarios would be developed to ensure that it effective emulation of BASE properties, thus allowing for effective evaluation against live systems.