**Basic Availibility User Stories**

1. The System should be eventually consistent. I should configure the system to have a 75% success rate. I should send three database updates into the system. When I call the web service that returns availability statistics, I should see that only one of the three updates failed and had to be rettied. The system should operate in the following way

1. I should set the system to become consistent every 100 seconds using the changeValue service. This should be passed in as an integer along with the name of the parameter in question.

2. I should use the changeValue service to set the availability percentage to the amount specified above. This should be passed in as an integer along eitht he nme of the parameter in question.

3. I should then send three updates to the system using the eventual consistency service using the updateData service that I developed as part of the eventual consistency service.

4.After 100 seconds, I should use the stats service to check the availability statistics. This service should return a Json result containing the percentage of requests that were successful, the percentage that failed, and the total number of requests received. The percentages should match those listed above

5. all fo the updates should have been persisted in the database, which I should be able to manually see if I inspect the database .

1. The System should be eventually consistent. I should configure the system to have a 50% success rate. I should send four updates into the system When I call the web service that returns availability statistics, I should see that only two of the four updates failed and had to be retried.

1. I should set the system to become consistent every 100 seconds using the changeValue service. This should be passed in as an integer along eitht he nme of the parameter in question.

2. I should use the changeValue service to se the availability percentage to the amount specified above. This should be passed in as an integer along eitht he nme of the parameter in question.

3. I should then send three updates to the system using the eventual consistency service using the updateData service that I developed as part of the eventual consistency service.

4.After 100 seconds, I should use the stats service to check the availability statistics. This service should return a Json result containing the percentage of requests that were successful, the percentage that failed, and the total number of requests received. The percentages should match those listed above

5. all fo the updates should have been persisted in the database, which I should be able to manually see if I inspect the database .

1. The System should be eventually consistent. I should configure the system to have a 25% success rate. I should send four updates into the system When I call the web service that returns availability statistics, I should see that three of the four updates failed and had to be retried.

1. I should set the system to become consistent every 100 seconds using the changeValue service. This should be passed in as an integer along eitht he nme of the parameter in question.

2. I should use the changeValue service to se the availability percentage to the amount specified above. This should be passed in as an integer along eitht he nme of the parameter in question.

3. I should then send three updates to the system using the eventual consistency service using the updateData service that I developed as part of the eventual consistency service.

4.After 100 seconds, I should use the stats service to check the availability statistics. This service should return a Json result containing the percentage of requests that were successful, the percentage that failed, and the total number of requests received. The percentages should match those listed above

5. all fo the updates should have been persisted in the database, which I should be able to manually see if I inspect the database .

1. The System should be eventually consistent. I should configure the system to have a 100% success rate. I should send four updates into the system When I call the web service that returns availability statistics, I should see none of the updates failed and had to be retried.

1. I should set the system to become consistent every 100 seconds using the changeValue service. This should be passed in as an integer along eitht he nme of the parameter in question.

2. I should use the changeValue service to se the availability percentage to the amount specified above. This should be passed in as an integer along eitht he nme of the parameter in question.

3. I should then send three updates to the system using the eventual consistency service using the updateData service that I developed as part of the eventual consistency service.

4.After 100 seconds, I should use the stats service to check the availability statistics. This service should return a Json result containing the percentage of requests that were successful, the percentage that failed, and the total number of requests received. The percentages should match those listed above

5. all fo the updates should have been persisted in the database, which I should be able to manually see if I inspect the database .