Reliable Systems Design Review Lab Project Team Members: Reviewing Team Members:

1. Vote Casting
   1. Can the project representative clearly explain the steps to put a vote into the database?
   2. Can the project representative clearly explain how a vote is represented using keys and values?
   3. Does the project rely on any information (for example, the number of voters seen so far) from previous voters? If so, how is it stored in and retrieved from the database?
   4. How many bits on average does the project need to send to the database and/or read from the database for each vote that is cast?
   5. Did the project count the bits in both the keys and the values?
2. Inquiry
   1. Can the project representative clearly explain the steps to retrieve votes for a voter-id from the database?
   2. How many bits on average does that project need to send to the database and/or read from the database for each vote in the inquiry?
3. Tallying
   1. Can the project representative clearly explain the steps to walk through all of the offices and candidates that have votes?
   2. Can the project representative clearly explain how the number of votes for a voter is read from the database?
4. Design
   1. Is there a diagram showing the main components in the design, and how they are related in terms of reliability (series, parallel)?
   2. Is there a mechanism for verifying that information retrieved from the database is correct (e.g. checksum, voting)?
   3. Is there a mechanism for replicating data to improve reliability?
   4. Are the reliability mechanisms described in a modular way? That is, are the checksums and redundancy applied uniformly to all information stored in the data?
5. Reliability computation
   1. Did the project representative provide a list of the individual operations that could fail?
   2. Did the representative accurately estimate the number of bits sent to and received from the database for each of these operations?
   3. Did the estimations of bits include all of the bits in the keys, values, checksums, and any other overhead needed for redundancy?
   4. Did the computations convert the number of bits per operation into the failure rate for the operation by dividing by 1000?
   5. Did the computations combine the error rates for operations using proper series and parallel computations? For series, you add the error rates. For parallel, if they all have the same error rate you divide the error rate by 1+1/2+1/3+…+1/k for k components in parallel.
   6. Did the computations take the reciprocal of the error rate to get the MTTF, in votes?
6. Recommendations. Make at least 5 recommendations, which may include:
   1. Suggestions for increasing reliability by reducing the number of bits needed to store the information in the database.
   2. Suggestions for identifying when information is valid or corrupt.
   3. Suggestions for replicating information.
   4. Suggestions for calculating the reliability more accurately.
   5. Suggestions for making the design more modular.