This report should include:

1. Cover page (as described previously)
2. SAS
3. Breakdown of individual contributions
4. Key Personnel information

Also, this report should be posted to the team website.

This assignment is a portion (question 3) of the final report (Software Design Document, available in the blank forms page).

1. Proposed software architecture
   1. Overview
      * Same as last two reports
      * Plus Nine basic components (see below)
   2. Subsystem decomposition – Identify the subsystems and the responsibilities of each. You should use component diagrams.
      * Alexa server
      * Alexa VUI
      * APH O&M Alexa skill
      * Lamdba Service
      * JS Controller
      * Amazon DynamoDB
   3. Hardware/software mapping – How will subsystems be assigned to hardware? You should use deployment diagrams.
      * Alexa server – Amazon servers
      * APH O&M Alexa skill– Amazon servers
      * Lamdba Service– Amazon servers
      * JS Controller– Amazon servers
      * Amazon DynamoDB– Amazon servers
      * Alexa VUI - Alexa terminals
   4. Persistent data management – Identify the data which will be persistent. Describe the file system or database to be used, including a complete database design.
      * DynamoDB
   5. Access control and security – For each different actor (user, system administrator, etc.) describe the operations they will be enabled to use. Describe authentication and security provisions.
   6. Global software control – Describe the control flow (e.g. procedural, event-driven, threaded). Procedural control flows should be described using activity diagrams. Event-driven flows are best described using sequence and state diagrams (use UML diagram standards).
   7. Boundary conditions – describe how the system will be started up, initialized and shut down. How will it respond to errors and exceptions? Any daily/weekly/monthly/yearly efforts necessary? All organizations buy new computers every 3-4 years, how to migrate to new server? Be able to bulk dump all data to file and bulk load all data from same file.

Regardless of the techniques being used, we can say that any system can be said to be

composed of nine basic component types:

* Use Cases
* Functions
* Triggers
* Data Stores
* Data Flows
* Data Elements
* Processors
* Data Storage
* Data Connections
* Actors/External Entities

Use Cases are an ordered set of processes, initiated by a specific trigger (e.g., transaction,

end of day), which accomplish a meaningful unit of work from the perspective of the

user.

Functions are context independent processes that transform data and/or determine the

state of entities.

Triggers are the events that intiate Use Cases. There are three types of triggers: time

triggers, state triggers and transaction triggers.

Data stores are data at rest. Data flows are data in movement between two processes, a

process and a data store, etc.

Data elements are the atomic units within data flows and data stores.

Processors are the components which execute the processes and events (i.e., computers

and people).

Data storage is the repository in which the data stores reside (e.g., disks, tapes, filing

cabinets).

Data connections are the pipelines through which the data flows flow (e.g.,

communications network, the mail).

Actors/External entities are people or systems outside the scope of the system under

investigation but with which it must interface.

Each of these components has many properties or attributes which are needed to fully

describe them. For example, in describing a process we can state its algorithm, who or

what executes it, where it takes place, when it takes place, how much information it must

process, etc. Figure 1 lists the properties which can be described for the various

component types.

In a given project and for a given component, the properties which must be

gathered/defined may vary. The SDLC must allow for this flexibility versus an all-ornothing

approach.