Intro: Hello everyone my name’s Alzraji Abdulla-Ali

This is my UML project.

**Slide 2:**

In the first slide we have the Class Diagram. The idea of the class diagram is to logically represent the system. It does this by listing all the classes that are going to be used and their relationships with each other. The class Diagram is the result of identifying classes from the flow of events.

I was able to break down the project into 6 main classes, User, Parent, Child, Online Database, and Database or the local database.

Some of the relationships the classes shared were composition, as we can see with the Parent and child class. The child class can not exist without the parent class.

We also see generalization or inheritance, if we look at the online Database the arrow represents inheritance. Inheritance is represented if there a class receives some of the qualities of another class.

**Slide 4: State Diagrams:**

A State Diagram shows the states/transitions a class goes through for a specific action.

In my first state diagram goes over the monitoring action. The first state, we must show the monitor so to do that the monitorOn attribute must change to on, and we must use a specific method to so. Then To check the monitor status you must have the monitor on, so that attribute must change to on. To check timestamp you must get the timestamp attribute so the state effects the timestamp attribute. To turn the system off or back on, you must change the monitorOn attribute. And finally to send to parents you must use the three attributes

**Sequence Diagrams:**

**Slide 7:**

In the first sequence diagram we have a representation of what the process of the E-Guard application will look like. First the parent must login or create an account, then turn on the monitoring system for the application. While the monitoring system is on the child browses the web until something triggers the monitoring system. The app then confirms if the item is an unhealthy item by checking with the Database. After it has been confirmed the app then notifies the parent.

At the top of the diagram, we have our 2 actors the child and the parent. Then our control, boundary, and entity objects. The boundary object is something the user interacts with directly. In this case it’s the interface. The entity is the data or the items stored in a file. Then the control object act to link between the boundary objects and entity objects.

**Slide 8:**

This Sequence diagram focuses on the order of the actions of the monitoring system. It must be turned on first, then it starts to monitor. In a case an unhealthy item is found it confirms its unhealthy with the database then notifies the parent. We also see that the monitoring system allows the parent to add and remove items from the Database. First takes the item to add from parent, then adds/removes it from the Database

**Slide 9:**

This sequence diagram focuses on order of events for confirming suspicious activity. In this sequence the child first triggers an unhealthy item, it then confirms with the data base, if it is not an unhealthy item it does nothing. Otherwise, it notifies parent