**Project Report**

**Brief Description of Project:**

The E-Guard Application’s goal is to monitor the child (set by the account manager/parent) from unhealthy items, and in a case where there are unhealthy items a screenshot is taken, and Email is sent to the parent (account manager). In this report you will find how the application works from the Design perspective using UML

**Functional Requirements:**

**-**Parent must be the only one with access to configure application

-Parent must be allowed to configure settings, and is the only one configuring settings

-Parent needs to be able to check keystroke, web activity, chat, email, and launched applications

-There should be a local Database that saves all the unhealthy activity, and synchronizes daily with remote server

-Find unhealthy activity, snapshot, get the time stamp, and send to parent

- Must block unhealthy items from popping up on the search engine

**Non-functional Requirements:**

**-**The application must be always monitoring (besides when the parent turns off monitoring)

- Application connects to the server via the IP address

- The application suspicious activity software must have quick confirmation abilities, less than 5 seconds

-Updates to the server must unnoticed

**UML Cases & their Descriptions**

1. **UML case of everything in the application working together**

Diagram

Description automatically generated

**Use Case Scenario:**

This use case focuses on the behaviors of the whole application. Taking into consideration the actors and their effects on the process of using the E-Guard application. In this scenario we can see that the parent is mostly the one interacting with the E-Guard system, the parent can login/create account, setup the app manager, and receive the emails from the E-Guard. Whereas the child doesn’t interact with the system but the E-Guard application monitor’s them in the Background. Finally, the Database is where everything is stored in terms of the unhealthy items.

**2. UML Case of the Web activity**Diagram

Description automatically generated

**Use Case Scenario:**

In this Use Case scenario, the Diagram focuses on how the E-Guard application will monitor the child’s web activity, and how the parent interacts with the Application in this scenario. We see that the parent has the ability to check the monitoring status, turn on/off the E-Guard Monitoring, and change the time’s the application can monitor the child. Again, we see that the child only interacts with the application when they are being monitored.

**3. UML Case of E-Guard Local Database**

Diagram

Description automatically generated

**Use Case Scenario:**

This Use Case focuses on the behavior of the parent and creating a local Database. The Diagram shows us that the parent accesses the application and goes to the monitoring system to create a local Data base (a locally kept file) to store all the unhealthy items that confirmed by the application and are triggered by the child. We see that the local Database is also an extension of the online Database that is kept on separate servers.

**4. UML Case Block unhealthy websites:**

Diagram

Description automatically generated

**Use Case scenario:**

In this diagram we plan out the behaviors of the E-guard application and the actors, the child, and local Database, with respect to the Website monitoring capabilities of the E-Guard. We can see that the first thing that occurs is that the monitoring system must check if the website is unhealthy, and then will decide what to do after that decision. In a case where it deems the application unhealthy then it blocks the site and adds it to the local Database. If the site is not healthy then it does nothing and continues to monitor.

**5. UML Case Remove Unhealthy words from Search Engine:**

Diagram

Description automatically generated

**Use case scenario:**

In this use case diagram, we focus on designing the behaviors of the E-guard application and all the actors (the child & local Database) in response to filtering the search engine. We see that the application is in the process of monitoring and in a case when the search engine is used, it configures the search settings to remove the unhealthy words that located in the Database. If it is unhealthy it removes it otherwise it continues to monitor.

**6. UML Case Configure Settings:**

Diagram

Description automatically generated

**Use Case Scenario:**

This Diagram focuses on the parent’s interaction with the E-Guard’s application settings. We can see that the parent has to confirm their account when trying to configure the settings, enter a passcode and then the settings are available for them to change. Otherwise the application allows them to re-enter the password, and if too many tries have been used it will lock the application.

**7. UML Case Configure timestamp**

Diagram

Description automatically generated

**UML Case Scenario:**

The UML diagram focuses on the parent’s interaction with setting up the Time zone for when the time stamp will be sent to them. The parent must choose a time zone, if they are requiring a timestamp, and they must choose the font for the time zone

**8. UML Case Send Email of unhealthy item**

Diagram

Description automatically generated

**UML Case Scenario:**

This UML Case focuses on the behavior of when an unhealthy item is found and the interactions between the child and the parent. We can see that the monitoring system confirms it’s an unhealthy item get’s the timestamp and sends the Email to the parent, and then proceeds to monitor the child.

**9. UML Case Create Keystroke Data base**

Diagram

Description automatically generated

**UML Case scenario:**

This Diagram focuses on the creation of a local database to store a combination of keystrokes that the parent has defined as unhealthy. If the child were to type something that was stored on that file, the E-Guard application will trigger it as an unhealthy item.

**10. UML case Creating a manager/Parent**

Diagram

Description automatically generated

**Use Case Scenario:**

This use case focuses on the user/parent first creation of an account. The user must meet the requirements of being an account manager for example, they must meet the age requirement of over 18, they must set restrictions, and they must add a pin.

**11. UML case Unhealthy Activity spotted**

Diagram

Description automatically generated

**Use Case scenario:**

The diagram shows the behaviors of the E-Guard application and the actors when suspicious activity is spotted. We first see the child triggering the unhealthy activity. Then the E-Guard confirms this by check both the online and the local database, and if true updates the local database

**12. UML case scenario Add unhealthy item to Database:**

Diagram

Description automatically generated

**Use Case Scenario:**

The diagram highlights the behaviors of the E-Guard when the parent or secondary account manager want to add an unhealthy item to the database. Going through the process we see that the parent chooses to display the local Database then has to add the item. The E-Guard then connects with the server Database and the local database updating both, and finally saving these changes.

**13. UML Case Monitoring system:**

Diagram

Description automatically generated

**Use Case scenario:**

The diagram focuses on the behaviors of the monitoring system and its actors, the parent and the child. The application first gets access to monitor in the background, with parents’ permission. The application then monitors in the background, it monitors keystrokes, emails, applications and texts.