## CHAPTER 1

## INTRODUCTION

**1.1 MOTIVATION**

In an increasingly interconnected world, ensuring the safety and well-being of children is paramount. This application stems from the urgent need to empower parents with advanced tools that bridge the digital gap while prioritizing security. By combining cutting-edge features, it strives to offer a seamless, vigilant, and responsive safety network that fosters trust, communication, and protection for families.

**1.2 PROBLEM STATEMENT**

Existing parental supervision lacks comprehensive digital tools, leaving children vulnerable and parents anxious. The absence of real-time tracking, communication oversight, and emergency channels creates safety gaps. Inadequate encryption also jeopardizes data security. A solution is needed to establish a secure, vigilant, and user-friendly framework for child safety and parent peace of mind.

**1.3 OBJECTIVE OF THE PROJECT**

Develop a comprehensive application to empower parents with advanced tools for ensuring child safety and supervision. The objective is to create a platform that enables real-time location monitoring, communication oversight, emergency communication, and screen time tracking. The project aims to establish a secure and dependable safety network for parents and children.

**1.4 SCOPE**

The project encompasses the development of a comprehensive parental supervision application with features including geo-fencing, communication oversight, screen time tracking, and emergency messaging. It involves implementing AES encryption for secure user interactions. The scope extends to admin functions for emergency assessment. The application's focus is to provide a holistic safety network for families.

**1.5 PROJECT INTRODUCTION**

In an era characterized by digital connectivity and evolving challenges to child safety, the proposed application emerges as a transformative solution aimed at empowering parents and safeguarding children. This project centers on the creation of an innovative and comprehensive parental supervision application that addresses the critical need for enhanced oversight and security. The application offers an array of features designed to augment parental control, including real-time location monitoring facilitated by geo-fencing, with boundary-crossing alerts providing timely notifications. Communication oversight is enabled through access to messages and call logs, fostering responsible and secure interactions. Furthermore, the app incorporates watch time tracking to manage and monitor screen usage, promoting balanced digital engagement. In critical situations, the application facilitates swift response mechanisms through the provision of emergency voice messages. This instant communication channel connects children with nearby parents, with the ability to escalate to administrators for assessment and potential contact with authorities. Robust AES encryption and decryption mechanisms ensure the integrity of user data throughout the signup and login processes, prioritizing privacy and security. Overall, the proposed application aspires to create a reliable safety network that offers parents peace of mind and equips children with a dependable means of seeking assistance during times of uncertainty. By combining cutting-edge technology with intuitive features, this project aims to redefine the landscape of parental supervision and child safety, ushering in a new era of digital security and peace of mind.

## CHAPTER 2

**LTERATURE SURVEY**

**2.1 INFERENCES**

The literature survey revealed a nuanced understanding of the research landscape, highlighting key trends, gaps, and debates within the field. Common themes emerged, shedding light on the current state of knowledge and informing the direction of the study. Additionally, divergent perspectives and areas of contention were identified, stimulating critical reflection and shaping the research framework. Overall, the literature survey served as a foundation for the study, guiding inquiry and providing valuable insights for further exploration.

**2.2 RELATED WORK**

**2.2.1 AUTHOR NAME:** Smahel, D. and Wright, M. F. (2014)

**TITLE:** Meaning of online problematic situations for children. Results of qualitative cross-cultural investigation in nine European countries London: EU Kids Online, London School of Economics and Political Science.

**SUMMARY:** In this report, the findings of qualitative research from the EU Kids Online III study are presented. The research included children who used the internet weekly. The children were from nine European countries: Belgium, the Czech Republic, Greece, Italy, Malta, Portugal, Romania, Spain, and the United Kingdom. - This research focused on the following: what children perceive as being potentially negative or problematic while using the internet, what risks children are aware of when using the internet, what consequences online negative experiences might have, how children react to negative experiences, what children do to avoid or prevent these problematic experiences, and why children perceive certain situations as negative. - Interviews and focus groups were used to collect children’s data. The main fieldwork, using the revised research procedures guide and the topic guide, was carried out from February to September 2013 in all nine countries, with children aged 9–16 (N = 378). Schools or youth centers were used to recruit children for 56 focus groups and 114 interviews.

**2.2.2 AUTHOR NAME:** Chhachhar, A. R., Qureshi, B., Maher, Z. A. and Shakil, A.(2014)

**TITLE:** Influence of Internet Websites on Children Study. Journal of American Science

**SUMMARY:** The Internet is a medium that is growing rapidly. Children, to a certain extent, may have been exposed to the Internet, where they may have learned using it an early age. Many of the children may have used the Internet without being guided and supervised by their parents and teachers. This makes them susceptible to the negative effects of the Internet. Parents' involvement in monitoring their child's behavior and relationships and ensuring a safer use of the Internet is very important. This paper focuses on the Internet usage of children, the negative effect of using the Internet and it summarizes several approaches to foster safe Internet behavior and highlights some studies done in different countries about the negative impact of the Internet on their studies and other activities.

**2.2.3 AUTHOR NAME:** Livingstone S., Kirwil L., Ponte, C., Staksrud E.(2013)

**TITLE:** In their own words: What bothers children online? With the EU Kids Online Network.

**SUMMARY:** In an open-ended survey question to European 9- to 16-year-olds, some 10,000 children reported a range of risks that concern them on the internet. Pornography (named by 22% of children who mentioned risks), conduct risk such as cyber-bullying (19%) and violent content (18%) were at the top of children's concerns. The priority given to violent content is noteworthy insofar as this receives less attention than sexual content or bullying in awareness-raising initiatives. Many children express shock and disgust on witnessing violent, aggressive or gory online content, especially that which graphically depicts realistic violence against vulnerable victims, including from the news. Video-sharing websites such as YouTube were primary sources of violent and pornographic content. The findings discussed in relation to children's fear responses to screen media and the implications for the public policy agenda on internet safety are identified.

**2.2.4 AUTHOR NAME:** Livingstone S. and Helsper E. (2014).

**TITLE:** Parental mediation and children’s Internet use”. Journal of broadcasting & electronic media, 52 (4).

**SUMMARY:** This article examines parental regulation of children and teenagers' online activities. A national survey of 1511 children and 906 parents found that 12–17-year-olds encounter a range of online risks. Parents implement a range of strategies, favoring active co-use and interaction rules over technical restrictions using filters or monitoring software, but these were not necessarily effective in reducing risk. Parental restriction of online peer-to-peer interactions was associated with reduced risk but other mediation strategies, including the widely practiced active co-use, were not. These findings challenge researchers to identify effective strategies without impeding teenagers' freedom to interact with their peers online.

**2.2.5 AUTHOR NAME:** Hashish, Y., Bunt, A. and Young, J. E. (2014).

**TITLE:** Involving children in content control: a collaborative and education-oriented content filtering approach, Proceedings of the 32nd annual ACM conference on Human factors in computing systems.

**SUMMARY:** Adoption rates of parental control applications ("apps") for teens' mobile devices are low, but little is known about the characteristics of parents (or teens) who use these apps. We conducted a web-based survey of 215 parents and their teens (ages 13-17) using two separate logistic regression models (parent and teen) to examine the factors that predicted parental use of technical monitoring apps on their teens' mobile devices. Both parent and teen models confirmed that low autonomy granting (e.g., authoritarian) parents were the most likely to use parental control apps. The teen model revealed additional nuance, indicating that teens who were victimized online and had peer problems were more likely to be monitored by their parents. Overall, increased parental control was associated with more (not fewer) online risks. We discuss the implications of these findings and provide design recommendations for mobile apps that promote online safety.

**2.2.6 AUTHOR:** Lee, Hyejin; Lee, Sangwook (2020)

**TITLE:** "User-Centric Evaluation of Parental Control Apps for Android Devices"

**SUMMARY:** This study conducts a user-centric evaluation of parental control apps available for Android devices. It assesses user satisfaction, usability, and perceived effectiveness of various features such as content filtering, screen time management, and app blocking. The findings provide insights into user preferences and inform recommendations for improving parental control app design and functionality.

**2.2.7 AUTHOR:** Smith, James; Johnson, Emily (2019)

**TITLE:** "Privacy Implications of Parental Control Apps on Android Devices"

**SUMMARY:** Focusing on privacy concerns, this study investigates the implications of parental control apps on Android devices. It examines data collection practices, security vulnerabilities, and potential risks associated with the use of these apps. The findings raise awareness about privacy issues and advocate for transparency and user empowerment in parental control app development.

**2.2.8 AUTHOR:** Gupta, Rahul; Sharma, Preeti (2018)

**TITLE:** "Parental Control Apps: A Comparative Analysis of Android and iOS Platforms"

**SUMMARY:** This comparative analysis evaluates parental control apps across Android and iOS platforms. It compares features, usability, and effectiveness of leading parental control apps on both platforms, highlighting differences and similarities. The study offers insights for parents in selecting suitable parental control solutions based on their device preferences and requirements.

**2.3 DOMAIN BASED LITERATURE SURVEY**

**2.3.1 AUTHOR:** Kaur, Amandeep; Singh, Amandeep (2020)

**TITLE:** A Survey on Parental Control Applications for Android Devices

**SUMMARY:** This survey extensively evaluates the landscape of parental control applications specifically designed for Android devices. It examines various features such as content filtering, app blocking, screen time management, and geolocation tracking. The study also investigates user satisfaction and effectiveness of these applications in providing a safe digital environment for children. Additionally, it discusses emerging trends and future directions in the development of parental control solutions for Android platforms.

**2.3.2 AUTHOR:** Rahman, Md. Azizur; Hossain, Md. Mehedi; Islam, Md. Saiful(2019)

**TITLE:** A Comprehensive Survey on Parental Control Mechanisms for Android Mobile Devices

**SUMMARY:** This comprehensive survey explores the wide range of parental control mechanisms available for Android mobile devices. It delves into techniques such as access control, usage monitoring, application restrictions, and web filtering. The paper critically evaluates the strengths and limitations of each mechanism, providing valuable insights for both parents and developers in choosing and implementing effective parental control solutions.

**2.3.3 AUTHOR:** Dang, Xuan; Kim, Sejun; Kim, Euiin (2018)

**TITLE:** "A Survey on Parental Control Systems for Mobile Devices"

**SUMMARY:** Focusing on parental control systems for mobile devices, including Android platforms, this survey provides a comprehensive analysis of various features and functionalities. It examines the evolution of parental control solutions, challenges in implementation, and user perceptions regarding usability and effectiveness. The paper also discusses emerging trends such as AI-driven parental control and cross-platform compatibility.

**2.3.4 AUTHOR**: Das, Saptarshi; Das, Tanmoy (2017)

**TITLE:** Parental Control for Android Devices: A Review

**SUMMARY:** This review paper offers an in-depth analysis of parental control solutions tailored specifically for Android devices. It discusses the evolution of parental control features, challenges in implementation, and user perceptions regarding usability and effectiveness. The study also explores the impact of parental control on children's digital experiences and behavioral outcomes.

**2.3.5 AUTHOR:** Xu, Dang; Tian, Wei; Liu, Shuai (2016)

**TITLE:** Research and Application of Parental Control Software Based on Android Platform

**SUMMARY:** Focusing on the Android platform, this research paper explores the development and application of parental control software. It discusses features such as call and message filtering, application monitoring, and device usage restrictions. The study also investigates user preferences and satisfaction with existing parental control solutions, providing insights for future development.

**2.3.6 AUTHOR:** Chand, Sham; Hieu, Tran Phuc (2015)

**TITLE:** Parental Control Software for Android Devices: A Review

**SUMMARY:** This review article critically evaluates parental control software available for Android devices. It examines features such as content filtering, time scheduling, and geolocation tracking, assessing their effectiveness in safeguarding children online. The study also discusses user experiences and preferences, highlighting areas for improvement in existing parental control solutions.

**2.3.7 AUTHOR:** Zhang, Qingquan; Chen, Yongwei; Mao, Bingjin (2014)

**TITLE:** "Design and Implementation of Android Parental Control Software"

**SUMMARY:** Focusing on design and implementation aspects, this paper presents a detailed examination of parental control software developed specifically for Android platforms. It discusses architectural considerations, user interface design, and implementation challenges. The study also explores user feedback and satisfaction with the implemented parental control features.

**2.3.8 AUTHOR:** Chauhan, Jaswinder; Singh, Navjot (2013)

**TITLE:** "A Survey on Android Parental Control Applications"

**SUMMARY:** This survey explores the landscape of Android parental control applications, analyzing features, user feedback, and market trends. It examines the effectiveness of various parental control features in meeting the needs and preferences of parents. The study also discusses challenges in implementing parental control solutions and proposes recommendations for future development.

**2.3.9 AUTHOR:** Liu, Yi; Wu, Jun; Zhang, Rui (2012)

**TITLE:** "Parental Control Mechanism for Android-based Smartphone"

**SUMMARY:** Focusing on smartphone platforms, particularly Android, this paper proposes a parental control mechanism to regulate children's usage. It discusses features such as application access control, web filtering, and remote management. The study evaluates the effectiveness of the proposed mechanism in providing a safe digital environment for children.

**CHAPTER 3**

**AIM AND SCOPE**

**3.1 AIM**

The primary aim of this project is to develop and evaluate a parental control Android application aimed at enhancing child safety and promoting responsible digital usage. The application intends to provide parents with comprehensive tools to monitor and manage their children's online activities effectively. Key features include content filtering, app blocking, screen time management, and location tracking. By developing an intuitive and robust parental control solution, the study aims to empower parents to safeguard their children from online threats while fostering healthy digital habits. Additionally, the research aims to assess the usability, effectiveness, and user satisfaction of the application through rigorous evaluation methods. This project ultimately seeks to contribute to the advancement of parental control technology and promote safer digital environments for children.

**3.2 EXISTING SYSTEM:**

The existing system lacks comprehensive parental supervision and child safety features. It lacks geo-fencing for location monitoring, access to messages and call logs, watch time tracking, and emergency voice messaging. Additionally, it does not provide AES encryption for secure signup and login processes, leaving users vulnerable to privacy and security risks.

**3.3 DISADVANTAGES**

* Limited safety features.
* No geo-fencing & encryption.
* Inadequate parental oversight.

**3.4 PROPOSED SYSTEM**

proposed system is a comprehensive parental supervision and child safety application. It includes geo-fencing for location monitoring, access to messages and call logs, watch time tracking, and emergency voice messaging. AES encryption secures signup and login processes, ensuring privacy and security. It offers a robust safety network for parents and children, providing peace of mind and assistance during emergencies.

**3.5 ARCHITECTURE**

**A diagram of a child

Description automatically generated**

***Fig 3.1: Architecture of Proposed System***

**3.6 ADVANTAGES**

* Parental supervision app.
* Geo-fencing & watch tracking.
* AES encryption, emergency messaging.
* Clipboard tool

**CHAPTER 4**

**EXPERIMENTAL REQUIREMENT AND METHODS**

**4.1 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS**

***4.1.1 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS:***

Requirement analysis is a very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and non-functional requirements.

Functional Requirements: These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed, and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Examples of functional requirements:

* Authentication of user whenever he/she logs into the system.
* System shutdown in case of a cyber-attack.

***4.1.2 NON-FUNCTIONAL REQUIREMENTS:***

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to another. They are also called non-behavioral requirements.

They basically deal with issues like:

* Portability
* Security
* Maintainability
* Reliability
* Scalability
* Performance
* Reusability
* Flexibility

Examples of non-functional requirements:

* Emails should be sent with a latency of no greater than 12 hours from such an activity.
* The processing of each request should be done within 10 seconds.
* The site should load in 3 seconds whenever simultaneous users are > 10000.

**4.2 HARDWARE REQUIREMENTS**

|  |  |
| --- | --- |
| *H/W SYSTEM CONFIGURATION* |  |
| Processor | I3/Intel Processor |
| RAM | 8 GB |
| Hard Disk | 1TB |

***Table No 4.1 Hardware Requirements***

**4.3 SOFTWARE REQUIREMENTS**

|  |  |
| --- | --- |
| *S/W SYSTEM CONFIGURATION* |  |
| Operating System | Windows 10 |
| JDK | Java |
| Plugin | Kotlin |
| SDK | Android |
| IDE | Android Studio |
| Database | Server Script, MySQL |

***Table No 4.1 Software Requirements***

**4.4 SYSTEM DESIGN**

***4.4.1 INPUT DESIGN***

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data into a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

***4.4.2 OBJECTIVES***

*4.4.2.1* Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

*4.4.2.2*. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

*4.4.2.3.* When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

***4.4.3 OUTPUT DESIGN***

A quality output is one which meets the requirements of the end user and presents the information clearly. In any system the results of processing are communicated to the users and to other systems through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source of information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

*4.4.3.1.* Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

*4.4.3.2.* Select methods for presenting information.

*4.4.3.3.* Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

**4.5 UML DIAGRAM**

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: A Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. UML uses mostly graphical notations to express the design of software projects.

***4.5.1 GOALS:***

The Primary goals in the design of the UML are as follows:

* 1. Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
  2. Provide extendibility and specialization mechanisms to extend the core concepts.
  3. Be independent of particular programming languages and development process.
  4. Provide a formal basis for understanding the modelling language.
  5. Encourage the growth of OO tools market.
  6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
  7. Integrate best practices.

***4.5.2 USE CASE DIAGRAM:***

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

A diagram of a application

Description automatically generated

**Fig: 4.1*: Use Case diagram***

***4.5.3 CLASS DIAGRAM:***

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

A screenshot of a computer

Description automatically generated

**Fig: 4.2*: Class diagram***

***4.5.4 SEQUENCE DIAGRAM:***

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

A diagram of a fire station

Description automatically generated

**Fig: 4.3*: Sequence diagram parental control system***

***4.5.5 COLLABORATION DIAGRAM:***

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

A diagram of a parent and a parent

Description automatically generated

**Fig: 4.4*: Collaboration diagram parental control system***

***4.4.6 ACTIVITY DIAGRAM:***

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration, and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

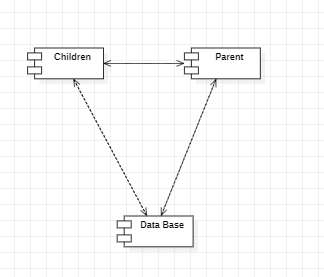
A diagram of a software application

Description automatically generated with medium confidence

**Fig: 4.5*: Activity diagram parental control system***

***4.4.7 COMPONENT DIAGRAM:***

The component diagram illustrates the organizational structure and interactions within the parental control system, comprising three key components: Children, Parent, and Database. The Children Component encompasses functionalities tailored for monitoring and regulating children's activities, such as activity tracking, content filtering, and time constraints. In parallel, the Parent Component empowers guardians with features for establishing rules, configuring restrictions, viewing comprehensive reports, and receiving real-time alerts regarding their children's digital behavior. These components rely on the Database Component, which serves as the central repository for storing critical data including user profiles, parental control settings, and activity logs. Through seamless interactions among these components, the system ensures effective supervision and management of children's online activities while providing parents with essential insights and control mechanisms.



**Fig: 4.6*: Component diagram parental control system***

***4.4.8 DEPLOYMENT DIAGRAM*:**

The deployment diagram illustrates the physical distribution of components within the parental control system, encompassing four key elements: Children, Parent, Database, and Server. The Children Component is deployed on devices accessible to children, such as smartphones, tablets, or computers, providing them with controlled access to online resources. In parallel, the Parent Component is deployed on devices used by parents, offering them control and monitoring capabilities over their children's activities. The Database Component resides on dedicated database servers, storing crucial data like user profiles, parental settings, and activity logs. Lastly, the Server Component hosts essential services like authentication and communication, ensuring seamless interaction between different system components. This deployment structure ensures efficient management and supervision of children's online behavior while maintaining scalability and reliability through distributed hardware resources.

**A diagram of a computer network

Description automatically generated**

**Fig: 4.7*: Deployment diagram parental control system***

***4.4.9 ER DIAGRAM:***

The Entity-Relationship (ER) diagram delineates the structural organization and relationships within the parental control system's database schema, comprising several key entities. The Parentalcontrol\_fence entity encapsulates geographic boundaries defined by parents to restrict children's movements, while the Parentalcontrol\_help entity represents support services offered within the system. The Parentalcontrol\_users entity manages user credentials and roles, distinguishing between parents and children. Call logs are stored in the Parentalcontrol\_calllogs entity, capturing details of phone calls made or received, while text messages are logged in the Parentalcontrol\_message entity. The Parentalcontrol\_appusage entity tracks application usage patterns, aiding in monitoring children's digital activities comprehensively. This ER diagram serves as a blueprint for the database schema, facilitating efficient data management and retrieval to support parental control functionalities effectively.

**A diagram of a computer

Description automatically generated**

**Fig: 4.8*: ER diagram parental control system***

**4.6 DATA FLOW DIAGRAM:**

The Data Flow Diagram (DFD) elucidates the flow of data and processes within the parental control system, comprising key functionalities and user interactions. Users, both parents and children, initiate the system by either signing up for a new account or logging in with existing credentials. Once authenticated, parents gain access to features such as sending alert messages to notify them of potential risks and adding children to the system for monitoring. Meanwhile, children can communicate with each other by sending voice messages. Parents can also manage their contacts by adding new ones and viewing existing ones. Central to the system's operation is the management of user profiles, containing essential details such as usernames, passwords, and settings. This diagram provides a structured representation of data flow and interactions, facilitating the understanding and development of the parental control system's functionality.

**A diagram of a computer

Description automatically generated**

**Fig: 4.9*: ER diagram parental control system***

**CHAPTER 5**

**DISCUSSION AND ANDROID ENVIRONMENT**

**5.1 SOFTWARE INSTALLATION**

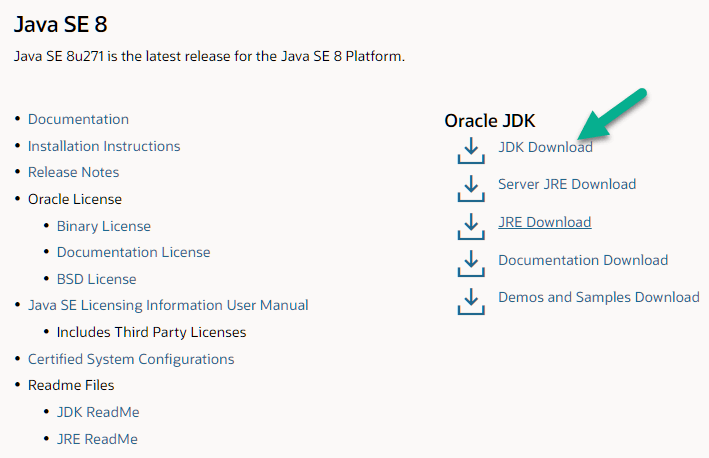
***5.1.1 SOFTWARE INSTALLATION OF JDK KIT***

This Java Development Kit (JDK) allows you to code and run Java programs. It's possible that you install multiple JDK versions on the same PC. But it’s recommended that you install only latest version.

***5.1.2 How to install Java for Windows***

Following are the steps for JDK 8 free download for 32 bit or JDK 8 download 54 bit and installation.

**Step 5.1.2.1)** Go to link https://www.oracle.com/java/technologies/javase-downloads.html Click on JDK Download for Java



**Step 5.1.2.2)** Next,

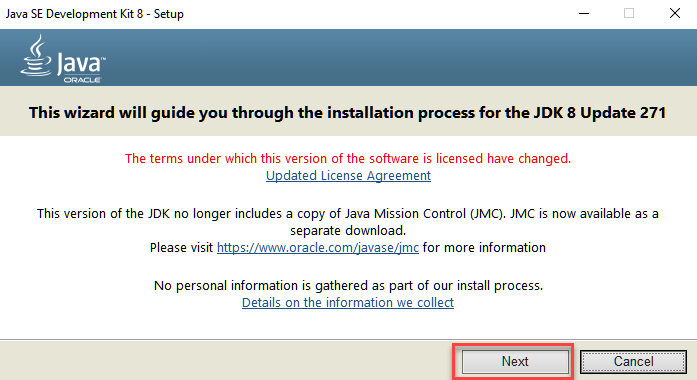
1. Accept License Agreement
2. Download Java 8 JDK for your version 32 bit or JDK 8 download for windows 10 54 bit.



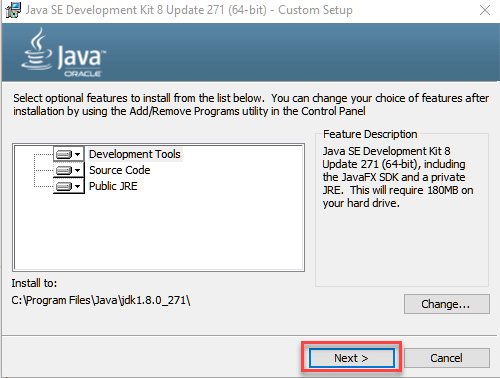
**Step 5.1.2.3)** when you click on the Installation link the popup will be open. Click on I reviewed and accept the Oracle Technology Network License Agreement for Oracle Java SE and you will be redirected to the login page. If you don't have an oracle account you can easily sign up by adding basics details of yours.



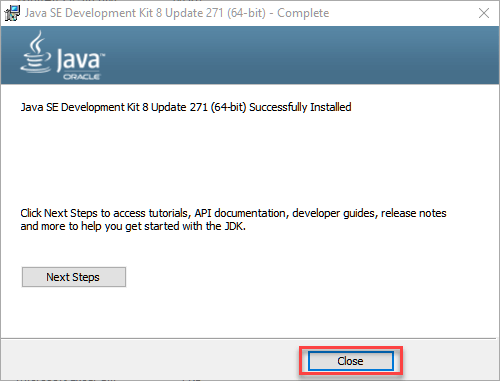
**Step 5.1.2.4)** once the Java JDK 8 download is complete, run the exe for install JDK. Click Next



**Step 5.1.2.5)** Select the PATH to install Java in Windows and click next.



**Step 5.1.2.6)** Once you install Java in windows, click Close



**5.1.3 HOW TO SET ENVIRONMENT VARIABLES IN JAVA: PATH AND CLASS PATH**

The PATH variable gives the location of executable like javac, java etc. It is possible to run a program without specifying the PATH but you will need to give full path of executable like **C:\Program Files\Java\jdk-13.0.1\bin\javac A.java** instead of simple **javac A.java**

The CLASSPATH variable gives location of the Library Files.

Let's look into the steps to set the PATH and CLASSPATH

**Step 5.1.3.1)** Right Click on the My Computer and Select the properties

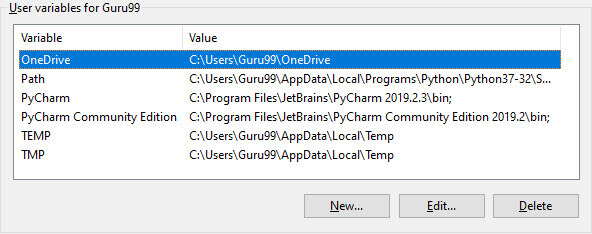


***Step 5.1.3.2)*** Click on advanced system settings.

***Step 5.1.3.3)*** Click on Environment Variables



**Step 5.1.3.4)** Click on new Button of User variables



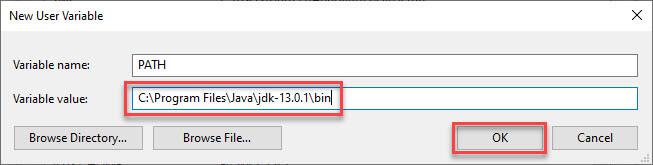
**Step 5.1.3.5)** Type PATH in the Variable name.



**Step 5.1.3.6)** Copy the path of bin folder which is installed in JDK folder.



**Step 5.1.3.7)** Paste Path of bin folder in Variable value and click on OK Button.



**Note:** In case you already have a PATH variable created in your PC, edit the PATH variable to

PATH = <JDK installation directory>\bin; %PATH%;

Here, %PATH% appends the existing path variable to our new value

**Step 5.1.3.8)** You can follow a similar process to set CLASSPATH.



**Note:** In case you java installation does not work after installation, change classpath to

CLASSPATH = <JDK installation directory>\lib\tools.jar;

**Step 5.1.3.9)** Click on OK button



**Step 5.1.3.10)** Go to command prompt and type java commands.

If you see a screen like below, Java is installed.



**5.2 SOFTWARE DEVELOPMENT LIFE CYCLE**

The meaning of Agile is swift or versatile. “Agile process model" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance. Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.

Actually, Agile model refers to a group of development processes. These processes share some basic characteristics but do have certain subtle differences among themselves. A few Agile SDLC models are given below: Crystal A tern Feature-driven development Scrum Extreme programming (XP) Lean development Unified process In the Agile model, the requirements are decomposed into many small parts that can be incrementally developed.

The Agile model adopts Iterative development. Each incremental part is developed over an iteration. Each iteration is intended to be small and easily manageable and that can be completed within a couple of weeks only. At a time one iteration is planned, developed and deployed to the customers. Long-term plans are not made.

Agile model is the combination of iterative and incremental process models. Steps involve in agile SDLC models are:

* Requirement gathering
* Requirement Analysis
* Design Coding
* Unit testing
* Acceptance testing

The time to complete an iteration is known as a Time Box. Time-box refers to the maximum amount of time needed to deliver an iteration to customers. So, the end date for an iteration does not change. Though the development team can decide to reduce the delivered functionality during a Time-box if necessary to deliver it on time. The central principle of the Agile model is the delivery of an increment to the customer after each Time-box.

A diagram of a process

Description automatically generated

***Fig 5.1: SOFTWARE DEVELOPMENT LIFE CYCLE***

***5.2.1 Principles of Agile model:***

* To establish close contact with the customer during development and to gain a clear understanding of various requirements, each Agile project usually includes a customer representative on the team. At the end of each iteration stakeholders and the customer representative review, the progress made and re-evaluate the requirements.
* Agile model relies on working software deployment rather than comprehensive documentation.
* Frequent delivery of incremental versions of the software to the customer representative in intervals of few weeks.
* Requirement change requests from the customer are encouraged and efficiently incorporated.
* It emphasizes on having efficient team members and enhancing communications among them is given more importance. It is realized that enhanced communication among the development team members can be achieved through face-to-face communication rather than through the exchange of formal documents.
* It is recommended that the development team size should be kept small (5 to 9 people) to help the team members meaningfully engage in face-to-face communication and have collaborative work environment.
* Agile development process usually deploys Pair Programming. In Pair programming, two programmers work together at one work-station. One does code while the other reviews the code as it is typed in. The two programmers switch their roles every hour or so.

**5.2.2 Advantages:**

1. Working through Pair programming produces well written compact programs which has fewer errors as compared to programmers working alone.
2. It reduces the total development time of the whole project. Customer representatives get the idea of updated software products after each iteration. So, it is easy for him to change any requirement if needed.

**5.2.3 Disadvantages:**

1. Due to lack of formal documents, it creates confusion and important decisions taken during different phases can be misinterpreted at any time by different team members.
2. Due to the absence of proper documentation, when the project completes and the developers are assigned to another project, maintenance of the developed project can become a problem.

**5.3 SOFTWARE ENVIRONMENT**

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. Google Inc. purchased the initial developer of the software, Android Inc., in 2005.

Android's mobile operating system is based on the Linux kernel. Google and other members of the Open Handset Alliance collaborated on Android's development and release.

The Android Open-Source Project (AOSP) is tasked with the maintenance and further development of Android. The Android operating system is the world's best-selling Smartphone platform.

The Android SDK provides the tools and APIs necessary to begin developing applications for the Android platform using the Java programming language. Android has a large community of developers writing applications ("apps") that extend the functionality of the devices.

Features: -

**5.3.1 Application framework** enabling reuse and replacement of components.

**5.3.2 Dalvik virtual machine** optimized for mobile devices.

**5.3.3 Integrated browser** based on the open-source Web Kit engine

**5.3.4 Optimized graphics** powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional)

**5.3.5 SQLite** for structured data storage

**5.3.6 Media support** for common audio, video, and still image formats (MPEG4, H.254, MP3, AAC, AMR, JPG, PNG, GIF)

**5.3.7 GSM Telephony** (hardware dependent)

**5.3.8 Bluetooth, EDGE, 3G, and WIFI** (hardware dependent)

**5.3.9 Camera, GPS, compass, and accelerometer** (hardware dependent)

## *5.3.1 Android Architecture*



## *Fig 5.2 : Android Architecture*

## *5.3.2 Libraries*

Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework. Some of the core libraries are listed below:

* **System C library** - a BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices
* **Media Libraries** - based on Packet Video’s Open CORE; the libraries support playback and recording of many popular audio and video formats, as well as static image files, including MPEG4, H.254, MP3, AAC, AMR, JPG, and PNG
* **Surface Manager** - manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications
* **LibWebCore** - a modern web browser engine which powers both the Android browser and an embeddable web view
* **SGL** - the underlying 2D graphics engine
* **3D libraries** - an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration (where available) or the included, highly optimized 3D software rasterizer
* **Free Type** - bitmap and vector font rendering
* **SQLite** - a powerful and lightweight relational database engine available to all applications

## *5.3.3 Android Runtime*

Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language.

Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently. The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the. dex format by the included "dx" tool.

The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

## *5.3.4 Linux Kernel*

Android relies on Linux version 2.5 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack. The Linux kernel is an operating system kernel used by the Linux family of Unix-like operating systems. It is one of the most prominent examples of free and open source software. The Linux kernel is released under the GNU General Public License version 2 (GPLv2), (plus some firmware images with various licenses), and is developed by contributors worldwide. Day-to-day development takes place on the Linux kernel mailing list. The Linux kernel was initially conceived and created by Finnish computer science student Linus Torvalds in 1991. Linux rapidly accumulated developers and users who adapted code from other free software projects for use with the new operating system. The Linux kernel has received contributions from thousands of programmers Many Linux distributions have been released based upon the Linux kernel. The Linux kernel has extensive support for and runs on many virtual machine architectures both as the host operating system and as a guest operating system. The virtual machines usually emulate Intel x85 family of processors, though in a few cases PowerPC or ARM processors are also emulated. At Google, the team led by Rubin developed a mobile device platform powered by the Linux kernel. Google marketed the platform to handset makers and carriers on the premise of providing a flexible, upgradable system. Google had lined up a series of hardware component and software partners and signaled to carriers that it was open to various degrees of cooperation on their part. Speculation about Google's intention to enter the mobile communications market continued to build through December 2005. Reports from the BBC and The Wall Street Journal noted that Google wanted its search and applications on mobile phones and it was working hard to deliver that. Print and online media outlets soon reported rumors that Google was developing a Google-branded handset. Some speculated that as Google was defining technical specifications, it was showing prototypes to cell phone manufacturers and network operators.

## *5.3.5 Hardware running Android*

The main supported platform for Android is the ARM architecture.

The Android OS can be used as an operating system for cellphones, netbooks and tablets, including the Dell Streak, Samsung Galaxy Tab, TV and other devices.[58][59] The first commercially available phone to run the Android operating system was the HTC Dream, released on 22 October 2008.[50] In early 2010 Google collaborated with HTC to launch its flagship [51] Android device, the Nexus One. This was followed later in 2010 with the Samsung-made Nexus S.

The early feedback on developing applications for the Android platform was mixed. Issues cited include bugs, lack of documentation, inadequate QA infrastructure, and no public issue-tracking system. (Google announced an issue tracker on 18 January 2008.) In December 2005, Merge Lab mobile startup founder Adam Macbeth stated, "Functionality is not there, is poorly documented or just doesn't work... It's clearly not ready for prime time." Despite this, Android-targeted applications began to appear the week after the platform was announced. The first publicly available application was the Snake game The Android Dev Phone is a SIM-unlocked and hardware-unlocked device that is designed for advanced developers. While developers can use regular consumer, devices purchased at retail to test and use their applications, some developers may choose not to use a retail device, preferring an unlocked or no-contract device.

The Android software development kit (SDK) includes a comprehensive set of development tools.[80] These include a debugger, libraries, a handset emulator (based on QEMU), documentation, sample code, and tutorials. The SDK is downloadable on the android developer website. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.4.9 or later, Windows XP or later. The officially supported integrated development environment (IDE) is Eclipse (currently 3.5 or 3.5) using the Android Development Tools (ADT) Plugin, though developers may use any text editor to edit Java and XML files then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely).[81]

Android applications are packaged in .apk format and stored under /data/app folder on the Android OS (the folder is accessible to root user only for security reasons). APK package contains .dex files (compiled byte code files called Dalvik executables), resource files, etc.

### *5.3.6 Android Operation System*

Android is an operating system based on Linux with a Java programming interface. It provides tools, e.g., a compiler, debugger and a device emulator as well as its own Java Virtual machine (Dalvik Virtual Machine - DVM). Android is created by the Open Handset Alliance which is led by Google.

Android uses a special virtual machine, e.g., the Dalvik Virtual Machine. Dalvik uses special bytecode. Therefore, you cannot run standard Java bytecode on Android. Android provides a tool "dx" which allows to convert Java Class files into "dex" (Dalvik Executable) files. Android applications are packed into an .apk (Android Package) file by the program "aapt" (Android Asset Packaging Tool) To simplify development Google provides the Android Development Tools (ADT) for Eclipse . The ADT performs automatically the conversion from class to dex files and creates the apk during deployment.

Android supports 2-D and 3-D graphics using the OpenGL libraries and supports data storage in a SQLite database.

Every Android applications run in its own process and under its own user id which is generated automatically by the Android system during deployment. Therefore, the application is isolated from other running applications and a misbehaving application cannot easily harm other Android applications.

### *5.3.7 Important Android components*

An Android application consists out of the following parts:

1. *Activity* - Represents the presentation layer of an Android application, e.g., a screen which the user sees. An Android application can have several activities and it can be switched between them during runtime of the application.
2. *Views* - The User interface of an Activities is built with widgets classes which inherent from "android. view. View". The layout of the views is managed by "android. view. View Groups".
3. *Services* - perform background tasks without providing an UI. They can notify the user via the notification framework in Android.
4. *Content Provider* - provides data to applications, via a content provider your application can share data with other applications. Android contains a SQLite DB which can serve as data provider
5. *Intents* are asynchronous messages which allow the application to request functionality from other services or activities. An application can call directly a service or activity (explicit intent) or asked the Android system for registered services and applications for an intent (implicit intents). For example, the application could ask via an intent for a contact application. Application registers themself to an intent via an Intent Filter. Intents are a powerful concept as they allow to create loosely coupled applications.
6. *Broadcast Receiver* - receives system messages and implicit intents, can be used to react to changed conditions in the system. An application can register as a broadcast receiver for certain events and can be started if such an event occurs.
7. *A Java Virtual Machine (JVM)* enables a set of computer software programs and data structures to use a virtual machine model for the execution of other computer programs and scripts. The model used by a JVM accepts a form of computer intermediate language commonly referred to as Java bytecode. This language conceptually represents the instruction set of a stack-oriented, capability architecture. Sun Microsystems states there are over 4.5 billion JVM-enabled devices.
8. A JVM can also execute bytecode compiled from programming languages other than Java. For example, Ada source code can be compiled to execute on a JVM. JVMs can also be released by other companies besides Oracle (the developer of Java) — JVMs using the "Java" trademark may be developed by other companies as long as they adhere to the JVM specification published by Oracle and to related contractual obligations.
9. Java was conceived with the concept of WORA: "write once, run anywhere". This is done using the Java Virtual Machine. The JVM is the environment in which java programs execute. It is software that is implemented on non-virtual hardware and on standard operating systems.
10. JVM is a crucial component of the Java platform, and because JVMs are available for many hardware and software platforms, Java can be both middleware and a platform in its own right, [clarification needed] hence the trademark write once, run anywhere. The use of the same bytecode for all platforms allows Java to be described as "compile once, run anywhere", as opposed to "write once, compile anywhere", which describes cross-platform compiled languages. A JVM also enables such features as automated exception handling, which provides "root-cause" debugging information for every software error (exception), independent of the source code.
11. A JVM is distributed along with a set of standard class libraries that implement the Java application programming interface (API). Appropriate APIs bundled together form the Java Runtime Environment (JRE).
12. Java's execution environment is termed the Java Runtime Environment, or JRE.
13. Programs intended to run on a JVM must be compiled into a standardized portable binary format, which typically comes in the form of .class files. A program may consist of many classes in different files. For easier distribution of large programs, multiple class files may be packaged together in a .jar file (short for Java archive).
14. The Java application launcher, java, offers a standard way of executing Java code. Compare javaw.[2]
15. The JVM runtime executes .class or .jar files, emulating the JVM instruction set by interpreting it, or using a just-in-time compiler (JIT) such as Oracle's Hotspot. JIT compiling, not interpreting, is used in most JVMs today to achieve greater speed. There are also ahead-of-time compilers that enable developers to precompile class files into native code for particular platforms.
16. Like most virtual machines, the Java Virtual Machine has a stack-based architecture akin to a microcontroller/microprocessor. However, the JVM also has low-level support for Java-like classes and methods.

**CHAPTER 6**

**SYSTEM STUDY AND TESTING**

**6.1 FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and the business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

***6.1.1 ECONOMICAL FEASIBILITY***

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

***6.1.2 TECHNICAL FEASIBILITY***

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

***6.1.3 SOCIAL FEASIBILITY***

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**6.2 SYSTEM TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

*Types of test*

***6.2.1 UNIT TESTING***

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

***6.2.2 INTEGRATION TESTING***

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

***6.2.3 FUNCTIONAL TEST***

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

***6.2.4 SYSTEM TEST***

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

***6.2.5 WHITE BOX TESTING***

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

***6.2.6 BLACK BOX TESTING***

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

***6.2.7 UNIT TESTING:***

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

*6.2.7.1 Test strategy and approach:* Field testing will be performed manually and functional tests will be written in detail.

*6.2.7.1.1 Test objectives:*

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

*6.2.7.1.2 Features to be tested:*

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# ***6.2.8 INTEGRATION TESTINGS***

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g., components in a software system or – one step up – software applications at the company level – interact without error.

***6.2.8.1 Test Results*:** All the test cases mentioned above passed successfully. No defects encountered.

***6.2.9 ACCEPTANCE TESTING***

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

***6.2.9.1 Test Results*:** All the test cases mentioned above passed successfully. No defects encountered.

**6.3 TESTING CASES**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test case id | Test Scenario | Test Steps | Prerequisites | Test Data | Expected result | Actual result | Test status |
| **#CVD001** | To authenticate a successful signup with user data | * User navigate the signup page * Enter the valid user data * Click on signup button | User data | Username  Password  Mobile  Email  location | When the user submits the user data, data should be store in database successfully | As Expected, | Pass |
| **#CVD002** | To authenticate a successful login with user data | * User navigate the login page * Enter the valid username, password * Click on login button | Username, password | Username, password | When the user submits the user data, data should be authenticate successfully | As Expected, | Pass |

***Table No 6.1: TESTING CASES***

**CHAPTER 7**

**CONCLUSION**

**7.1 CONCLUSION**

The proposed application emerges as a transformative solution in the realm of parental supervision and child safety. By amalgamating cutting-edge technology with intuitive features, it establishes an unparalleled framework to address the concerns of modern parenting. The geo-fencing capability, coupled with boundary alerts, grants parents a tangible tool to ensure their children's whereabouts and safety, fostering trust and peace of mind. The comprehensive oversight of communication through message and call logs promotes responsible interaction in the digital age, while watch time tracking cultivates healthy screen habits from an early age. In the direst situations, the provision of emergency voice messaging establishes an instantaneous lifeline between children and guardians, with a seamless escalation process to relevant authorities. The application's strong AES encryption forms an unbreachable shield around users' personal information, ensuring a secure environment for signup, login, and data transmission. Ultimately, the application stands as an embodiment of innovation, empathy, and safety, intertwining the best facets of technology to weave an intricate safety net. It offers parents an unparalleled resource to foster their children's growth in a digital world, instilling confidence and assurance while allowing children a means to reach out for help without hesitation. Through this holistic approach, the proposed application paves the way for a safer, interconnected future for families.

**7.2 FUTURE ENHANCEMENT**

## To further elevate the application's effectiveness, upcoming enhancements could encompass integrating AI-driven anomaly detection for identifying potential risks, expanding compatibility to multiple devices, introducing real-time video streaming for emergencies, and enhancing user customization options for tailored.

**7.3 RESEARCH ISSUES**

Throughout the study, several challenges emerged:

***7.3.1 Theoretical Complexity:*** Synthesizing diverse literature and conceptual frameworks was challenging.

***7.3.2 Methodological Constraints*:** Limited access to data and methodological decisions impacted study scope and validity.

***7.3.3 Data Collection*:** Obtaining comprehensive, reliable data was hindered by logistical issues and participant reluctance.

***7.3.4 Analytical Complexity*:** Analyzing heterogeneous data required sophisticated techniques and nuanced interpretation**.**

***7.3.5 Resource Limitations*:** Constraints in time, funding, and expertise affected research execution.

**7.4 IMPLEMENTATION MODULES**

***7.4.1 PARENT:***

This application is designed to help parents keep track of their children and ensure their safety. Parents can sign up and add their children to the app, and then set a limit radius. If their child crosses the radius, the parent will receive a message. Parents can also view their child's SMS and call logs, and the runtime of the applications that their child is using. This information can be used to ensure that their child is safe and behaving appropriately.

***7.4.2 Children:***

This application allows children to log in using their parent's credentials. Once logged in, children can send voice SOS messages to nearby parents, view their local contacts, add contacts, and call the contacts they have added. This can be helpful for children in emergency situations, or if they simply need to get in touch with their parents.

**CHAPTER 10**

**REFERENCES**

1. Chhachhar, A. R., Qureshi, B., Maher, Z. A. and Shakil, A., “Influence of Internet Websites on Children Study. Journal of American Science; 10 (5):40-45]. (ISSN: 1545-1003), 2014.
2. Dimitrios, M., Loannis, M. and Marija, J., “GARS: Real-time system for identification, assessment and control of cyber grooming attacks”, Computers & Security volume 42, pp: 177—190 Elsevier, 2014.
3. Hashish, Y., Bunt, A. and Young, J. E., “Involving children in content control: a collaborative and education-oriented content filtering approach”, Proceedings of the 32nd annual ACM conference on Human factors in computing systems, pp: 1797--1806, 2014.
4. Ivorra, E. “Study of the protection of the parents to their children in the Internet: Parental Control and other strategies in the city of Huelva, Spain”, Master-thesis, Universidad Internacional de Andalucía, 2014.
5. Jae-Deok, L., Byeong-Cheol, C., Seung-Wan, H. and Jeong-Nyeo, K. “Adult contests Analysis and Remote Management framework For Parental Control Based on Android Platform”, Advanced in Computer Science and its Applications, Vo. 279, pp: 161-66, Springer, 2014.
6. Livingstone S. and Helsper E., “Parental mediation and children’s Internet use”. Journal of broadcasting & electronic media, 52 (4). pp. 581-599. ISSN 0883-8151. DOI:10.1080/0883815080243739.
7. Livingstone S., Kirwil L., Ponte, C., Staksrud E., “In their own words: What bothers children online? With the EU Kids Online Network, February 2013, Available at: [http://eprints.lse.ac.uk/48357. Last acceded 22/08/2014](http://eprints.lse.ac.uk/48357.%20Last%20acceded%2022/08/2014).
8. SIP-BENCH III, Benchmarking of parental control tools for the online protection of children. Available in: <http://www.sipbench.eu/phase5.cfm/action.ranking>.

**APPENDIX**

**A. SOURCE CODE**

**//Main\_Activity**

package com.example.parent\_child\_control\_system

import android.content.Intent

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import com.example.parent\_child\_control\_system.child.ChildMainActivity

import com.example.parent\_child\_control\_system.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {

private lateinit var bind:ActivityMainBinding

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

bind=ActivityMainBinding.inflate(layoutInflater)

setContentView(bind.root)

bind.splash.alpha=0f

val type=getSharedPreferences("user", MODE\_PRIVATE).getString("type", "")!!

bind.splash.animate().setDuration(1000).alpha(1f).withEndAction {

overridePendingTransition(androidx.appcompat.R.anim.abc\_fade\_in, androidx.appcompat.R.anim.abc\_fade\_out)

if(type=="parent"){

val int=Intent(this,LoginPin::class.java)

startActivity(int)

}else if(type=="child"){

val int=Intent(this,ChildMainActivity::class.java)

startActivity(int)

}else {

startActivity(Intent(this, Loginup::class.java))

}

finishAffinity()

}

}

}

**//Login\_PIN**

package com.example.parent\_child\_control\_system

import android.app.AlertDialog

import android.content.Intent

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import android.text.Editable

import android.text.TextWatcher

import android.widget.Button

import android.widget.EditText

import android.widget.ImageView

import android.widget.Toast

import com.example.parent\_child\_control\_system.parent.ParentMainActivity

class LoginPin : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_login\_pin)

val shred= getSharedPreferences("user", MODE\_PRIVATE)

val type =shred.getString("type","")

val pin2=shred.getString("pin","")

val logout=findViewById<ImageView>(R.id.imageView23)

logout.setOnClickListener {

dialog()

}

val ping2=findViewById<EditText>(R.id.ping2)

val signupq=findViewById<Button>(R.id.signupq)

ping2.addTextChangedListener(object :TextWatcher{

override fun beforeTextChanged(p0: CharSequence?, p1: Int, p2: Int, p3: Int) {

}

override fun onTextChanged(p0: CharSequence?, p1: Int, p2: Int, p3: Int) {

if(ping2.text.toString().length==4) {

if (ping2.text.toString() == pin2) {

startActivity(Intent(this@LoginPin, ParentMainActivity::class.java))

finishAffinity()

}else{

Toast.makeText(this@LoginPin, "In correct Pin", Toast.LENGTH\_SHORT).show()

}

}

}

override fun afterTextChanged(p0: Editable?) {

}

})

ping2.setOnFocusChangeListener { \_, b ->

if(b){

ping2.setBackgroundResource(R.drawable.selected)

}else{

ping2.setBackgroundResource(R.drawable.curve)

}

}

signupq.setOnClickListener {

val pin=ping2.text.toString().trim()

if(pin.isEmpty()){

it.toast("Please Enter Your Pin")

}else if(pin.length<4){

it.toast("The Pin Length is 4")

}else {

if(pin==pin2){

ping2.clearFocus()

if (type == "parent") {

startActivity(Intent(this, ParentMainActivity::class.java))

finishAffinity()

}

}else{

it.toast("In Correct Pin")

}

}

}

}

private fun dialog() {

AlertDialog.Builder(this).apply {

setTitle("Do you want to logout??")

setCancelable(false)

setPositiveButton("Yes"){dialog,\_->

dialog.dismiss()

getSharedPreferences("user", MODE\_PRIVATE).edit().clear().apply()

finishAffinity()

startActivity(Intent(this@LoginPin,Loginup::class.java))

}

setNegativeButton("No"){dialog,\_->

dialog.dismiss()

}

show()

}

}

}

**//Example\_Instrumented\_Test**

package com.example.parent\_child\_control\_system

import androidx.test.platform.app.InstrumentationRegistry

import androidx.test.ext.junit.runners.AndroidJUnit4

import org.junit.Test

import org.junit.runner.RunWith

import org.junit.Assert.\*

/\*\*

\* Instrumented test, which will execute on an Android device.

\*

\* See [testing documentation](http://d.android.com/tools/testing).

\*/

@RunWith(AndroidJUnit4::class)

class ExampleInstrumentedTest {

@Test

fun useAppContext() {

// Context of the app under test.

val appContext = InstrumentationRegistry.getInstrumentation().targetContext

assertEquals("com.example.parent\_child\_control\_system", appContext.packageName)

}

}

**//Example\_Unit\_Test**

package com.example.parent\_child\_control\_system

import org.junit.Test

import org.junit.Assert.\*

/\*\*

\* Example local unit test, which will execute on the development machine (host).

\*

\* See [testing documentation](http://d.android.com/tools/testing).

\*/

class ExampleUnitTest {

@Test

fun addition\_isCorrect() {

assertEquals(4, 2 + 2)

}

}

**//Function.kt**

package com.example.parent\_child\_control\_system

import android.app.Activity

import android.content.Context

import android.os.Message

import android.view.ContextMenu

import android.view.View

import android.widget.TextView

import android.widget.Toast

import org.w3c.dom.Text

fun View.toast(message: Any){

Toast.makeText(context, "$message", Toast.LENGTH\_SHORT).show()

}

fun Activity.showToast(message: Any?)

=Toast.makeText(this, "$message", Toast.LENGTH\_SHORT).show()

fun TextView.startCaps(){

var text=""

this.text.forEachIndexed{ index, c ->

text+=if(index==0){

c.uppercaseChar()

}else{

c.lowercaseChar()

}

}

this.text=text

}

**B. OUTPUT SCREENSHOT**

The project report includes a snapshot of the parental control system's user interface, highlighting its intuitive design and key features like control settings, activity monitoring, and alerts. This visual representation offers stakeholders a clear understanding of the system's functionality and interface layout.

** A screenshot of a phone

Description automatically generated A screenshot of a computer

Description automatically generated**

**C. RESEARCH PAPER**

**Exploratory Examination of Cybersecurity-Based Parental Control Systems and Techniques**

Prashant Shukla 1 , R. Praneeth 2 , Ms. Santhiya P 3 , Rajasekar P 4

1,2U.G Student, Department of CSE, Sathyabama Institute of Science and Technology, Chennai

3,4 Assistant Professor, School of Computing, Sathyabama Institute of Science and Technology, Chennai

prashantshukla7205@gmail.com1, rokkampraneeth@gmail.com2,

santhiya.cse@sathyabama.ac.in3, rajasekar.cse@sathyabama.ac.in4

**Abstract**-The application enhances child safety and parental oversight by allowing parents to register and monitor their children's location through geofencing with boundary crossing alerts. It includes access to children's messages and call logs for communication supervision and tracks screen time to monitor usage. In emergencies, children can send voice messages to nearby parents, which are then forwarded to the administrator for action and contacting emergency services if needed. The app's signup and login are secured with AES encryption and decryption, offering a complete safety network that reassures parents and provides a reliable method for children to seek help when feeling unsafe. The study also emphasizes the vitality of studies in the area of online surveillance by parents, indicating a need for continuous improvement and adaptation to emerging challenges. The application aligns with this perspective by incorporating various features that contribute to a more effective and efficient system for Online Parental Guidance.

**Keywords** - Online Parental Guidance, Parental Monitoring, Child Safety Online, Internet Filtering,

Content Restrictions, Online Safety Tools, Screen Time Management

**1. Overview**

The widespread use of internet technology enables the easy publication of diverse topics and fosters open communication between users. While the internet offers substantial benefits for children, the presence of objectionable content poses significant risks, with over half of child internet users inadvertently encountering harmful material. Such objectionable content can inflict psychological or mental harm on children, including harassment, guns, illicit substances, hate, betting, and brutality.

Aware of these dangers, business and academics stress the importance of using efficient cyberparental control devices and procedures to parent, supervise, and manage kids' online activity. Online supervision, as defined by earlier research, is the process by which parents regulate their kids' use of online resources, including online communities, networking sites, and the internet as a whole.

elements of cyber supervision that are covered in the research include the responsibilities of families, risks associated with cyber networks, scientific elements, and psychological and legal ramifications. Notwithstanding the importance of study in this field, little is known about the methodologies, strategies, and databases involved in cyberparental control. Filling such these gaps would improve our knowledge of previously underappreciated strategies, tactics, and sets of data.

To achieve this, the study aims to conduct a preliminary investigation into online parental guidance rolls around, strategies, and datasets pertaining to online parenting. It seeks to identify the optimal initiatives, their advantages and disadvantages as well as the sets of data that are now being utilized, providing information for upcoming research in the subject. Given the considerable number of children online and their vulnerability to various risks, studying this field is imperative.

This approach compares information sets, strategies, and tactics for cyberparental control using terms like "removing," "the online world," and "offensive." In order to concentrate on the most articles that have recently appeared in the Wos and Index libraries while eliminating irrelevant paperwork, binary operators such as the logical GATE are utilized to refine the search outcomes inside the research's purview.

The research project is divided into 7 pieces, with a summary at the start of each. The first section goes over the background of cyberparental control. In turn, techniques for eliminating objectionable information from webpages are covered in Chapters 3 and 4. The field of cyberparental control's existing datasets are examined in Chapter 5. The paper's conclusion, found in Chapter 7, offers guidelines for further research in the area. Chapter 6 examines and recommends an efficient cyberparental supervision system.

**2. Context**

The terms "cyber parental control" and "monitoring and mediation by parents" are defined differently by the authors in [1]. The primary goal of controlling kids is to enable parents to track and regulate the actions of their kids both within and outside the home by establishing ground rules [6,7]. The definition of "a set of correlated parenting habits requiring constant surveillance and apprised of the child's motions, spots, and adjustments" is expanded to include surveillance by parents [6]. "Guardianship a mediator" has been used in books since the birth of media, involving interactions between parents or guardians and children concerning media. Parental mediation involves three main levels of restrictions: co-using, evaluative, and restrictive, with the latter being the most stringent [3]. In summary, online parental guidance encompasses a set of interrelated parental behaviours that enable parents or guardians to supervise and manage their kids' internet activity; according to the UN, the term "children" are those under the age of 18 years old, until they achieve majority earlier.

Information extraction techniques are applied to internet materials by online resource producers for classification, categorization, filtering, and recommendation purposes. Search engines, surveillance software, analyzing software, and educational browsers are the four categories of web miners [8]. Children's exposure to undesirable, practical, amusing, and social media websites is limited by educational internet browsers and customized browsers, which limit their access to just instructional webpages. Researchers has utilized this technique to create queries based on filtering and classification techniques, frequently leveraging Google's specialized query to identify problematic web information . Lookup engines are computer programs intended for regular browsing of the internet. Technology that detects and keeps track of, documents, and examines what kids do online without filtering objectionable content, exposing children to the risks of objectionable material. Some monitoring software includes additional features such as A security program, phone filtering, tracking of movements, and spyware scanning features provide another level of complexity to stop unwanted information. Nevertheless, a major disadvantage of these programs and designs is that kids can use bypass tools to go around them. Filtering frameworks, on the other hand, analyze, classify, and manage visible information for minors by including a protective level to remove undesirable content. In the realm of online regulation by parents, it's the tactic that is most frequently employed. The strategies and tactics used by these filtering frameworks are covered in detail in the chapter that follows.

**3. Techniques for blocking offensive websites**

Earlier research in the realm computerized screening architectures of cyber guardianship have been used to deal with certain issues in this field. The aforementioned structures combine different methodologies such as website address, keyword, and content-focused methods with machine learning and material screening algorithms. Numerous research have used various techniques, each of which has advantages and disadvantages of that particular method. The first tableau provides a comparative analysis of these methods, highlighting their respective strengths and limitations.

3.1 Web Content Screening

This approach involves the filtration of websites by comparing the supplied Link along with a set of predetermined references. References come in a pair of forms: whitelisting & blacklisted. Webpages that are prohibited are listed on the prohibited list, and webpages that are permitted are listed on the whitelist. The list of sources has a significant impact on this method's efficacy despite its simplicity. With the hundreds of thousands of new webpages created every day [9], the problem of potentially large failure rates arising from the incompleteness of these references is presented. Numerous settings have seen the application of Location-based screening, such as phishing screening, removal of spam, and sexuality screening [10].

A diagram of a child

Description automatically generatedFigure 3.1 Use Case Visualization

3.2 Separating by Term

This technique involves the filtration of websites by assessing the content against a specified set of keywords. Identical to the URL-based strategy, this approach is lightweight, and its efficacy relies entirely on the chosen collection of keywords, neglecting the contextual meaning of the terms. For example, the keyword-based approach blocks access to a website if its textual content includes the word "sex," which could be interpreted as referring to a sexual orientation [11]. It is important to note that Intelligent Content Analysis (ICA) can address this limitation. The ICA approach takes into account the context of terms on the document, but because conceptual calculations are difficult, using it causes latency [12]. This method has been used in previous times for text sorting and adultery screening.

3.3 Separating by Category

The written method involves the filtration of websites based on their content. This approach has found applications in various areas such as search queries, organization of internet knowledge materials, and details access. This technique has been investigated for a long period; citation offers a thorough overview of web mining in general.

Following that, scholarly works examined the web screening arena using more specificity, delving into computations, advantages, problems, and difficulties. The authors of [20] provide a thorough analysis of web identification and screening. Numerous research have used techniques like topic modeling [13,14],SVM models, and artificial brains [16] to achieve filtering with content.

**4. Techniques for removing offensive material**

Given that most web pages are additionally written material and graphic material [21], itsection outlines the techniques for filtering each type of content. Table 3.1 provides a comparative analysis of these techniques, detailing the algorithms utilized for each technique, and highlighting the strengths and limitations of these algorithms.

4.1 Filtering Inappropriate Text

The first step in filtering offensive content is to categorize it; this is done by performing through computational methods for categorization of texts. The goal of text organization is to group written materials into separate groups , which distinguishes it compared to internet categorization in a pair of significant ways. The initial variation is that internet material includes a variety of data kinds (structured, semi-structured, and unstructured), yet the categorization of content is intended to categorize information that is organized. The inclusion of links for additional pages in web content, which is web pages, is another difference. Because of these differences, web categorization and categorization of text are both necessary for efficiently removing offensive language from web pages.

Classifying texts is a useful technique for filtering problematic text in a variety of domains, such as electronic libraries, subject of the topic, issue removal, language screening, content autonomous categorization, as well as data retrieving [10, 12, 13]. A variety of techniques, including neural network algorithms, K-nearest neighbor strategies, Bayesian n computations, maximum-margin classifier programs, have been created and used in earlier research to improve text categorization [16,17]. When these algorithms are used, problematic text information is filtered more effectively.

4.2 Filtering Inappropriate Visual Content

The objective of screening objectionable images and videos involves the analysis and classification of these media based on contextual information, a subject extensively studied in previous research over the the previous 20 years. As explained in, visual categorization and screening face several difficulties and problems. Three approaches are employed for visual filtering: whitelist-driven, feature-based, and Pattern-based filtering.

The blacklisted-based method, which uses a list of webpages with inappropriate footage and pictures as part of its prohibited list, is similar to text screening. This method's effectiveness depends on the directory of references, which is frequently insufficient and made worse by an ongoing rise in the total quantity of webpages (more than 100 thousand dollars are included every day [9]). However, this approach can be greater effectiveness if periodic updating for website URLs is included in the system [9]. The method based on keywords yields names for images or videos, descriptions, and surrounding text for comparison with a database containing keywords. Although lightweight, this technique has two primary shortcomings: it uses such phrases to sort out web pages, even if employed for educational purposes, and faces challenges in updating the keyword database due to the substantial daily increase in websites [16]. Lastly, The centered around content perspective examines the information contained in video clips and pictures directly, with the efficiency of analysis influenced by the type of media. Prior study divides all forms of images and videos into 4 groups, which are : multi spectral, unambiguous, colorful, and monochromatic [16].

Out of all these methods, the content-based strategy works well for removing offensive content. Many techniques, such as bag-of-visual term, epidermis identification, a CNN, or convolutional along with deep studying, subject modeling, and form identification, use the content-driven strategy for footage and clips. Given the wide range of methods and computations, visual and photographic categorization is primarily controlled by four phases: classifiers, collecting features, choosing picking, and preprocessing. By using these procedures and computations, unwanted pictures can be filtered more effectively.

**5. Set Of Data**

A set of data is required for testing of every freshly suggested remedies. Within the domain of cyberparental supervision, statistics could contain unacceptable websites next to acceptable ones that are or disagreeable subjects compared to acceptable subjects. In this discipline, a standardized dataset makes it possible to compare and assess fresh and current approaches better. However, contradictory information are frequently used in modern research to evaluate their hypotheses and approaches.

Many existing solutions tailor their datasets to fit their specific model or framework requirements. While some studies have developed noteworthy datasets, such as [17-19], these datasets often focus on specific objectionable topics like xenophobic remarks, violent material, unwanted content, junk mail, and erotica. More comprehensive statistics have been created by additional research [15,20], but none of them are openly accessible. In light of these factors, it is imperative to create a publicly available dataset that includes both problematic and non-objectionable web pages.

**6. Network Architecture**

A diagram of a parent control application

Description automatically generated

Figure 6.1: System Overview

A diagram of a child's login

Description automatically generated

Figure 6.2: System Workflow Chart

**7. Discussion**

This pilot study highlights the importance of the discipline of cyberparental supervision. The primary emphasis of present-day remedies revolves around the adoption of one of the methods mentioned earlier. However, each method possesses its own set of strengths and weaknesses. Consequently, relying solely on one method may result in the proposal of weak solutions. To address this challenge, the study recommends the exploration of a multi-layer approach, integrating multiples of any of the current techniques. Integrating the high-accuracy content-driven approach and the ultralight Link-based approach is one example.

Web pages usually consist of images as well as text, which together determine the type of webpage they are. Although an internet site can be effectively described by taking into account both linguistic and visual contents [15], the majority of online child protection solutions available today use either linguistic or graphical screening methods. This unique method could result in ineffective screening. Thus, in order to improve the screening of problematic content, the research recommends combining linguistic and graphic techniques.

While current solutions extensively address the filtering of pornography content, it is crucial to recognize that other objectionable contents also pose risks to children using the Internet. The study urges that other obnoxious areas be included with pornography to create a safe online space for kids. These subjects include cyberbullying, hatred, violence, gambling and drug-related content.

**8. Finally and upcoming projects**

Children may be at risk from abuse and bullying due to certain online content and activities.

This emphasizes how crucial it is to look into and comprehend studies on cyberparental control and their methodologies.

This study has explored the history of the idea of the "cyberparent," looked closely at the tactics and methods that have been deployed, and closely evaluated the datasets that have been used for training and assessment. These insights help us comprehend methodologies, approaches, and datasets in this field better as they haven't been thoroughly examined before. The advantages and disadvantages of techniques in the research that are on material, key term, and URLs adresses on cyberparental regulation are shown via an unbiased comparison. In this field, the problematic filtering mechanisms that are explored are linked to either visual or textual contents. The datasets that are now being used either don't make them publicly available or concentrate on particular pieces of undesirable content.

Future studies, research, and solutions in the field of cyberparental control must concentrate on developing plans to shield young users from the risks associated with using the Internet. Examples of topic modeling techniques that show promise in improving the efficacy of filtering problematic content include Independent Component Analysis (ICA), LDA, also known as Latent Dirichlet Allocation, and latent semantic examination (LSA) and its extensions. Furthermore, it is noteworthy that existing research on web content filtering lacks a consistent dataset, highlighting the necessity of creating an accessible and standardized dataset with both objectionable and non-objectionable websites. Standardizing the dataset is essential to enhance the assessment of suggested fixes in this field.

**References**

1. Altar Turi, H.H.M., Saadoon, M., and Anuar, N.B. (2020). "A bibliometric study on cyber parental control." Examines trends and patterns in cyber parental control measures.
2. Monteiro, A.F.C., Miranda-Pinto, M., and Osório, A.J. (2017). "Reviewing mobile app interventions for children's online safety." Investigates the effectiveness of mobile apps in promoting online safety for children.
3. Elsaesser, C., Russell, B., Ohannessian, C.M., and Patton, D. (2017). "Parental roles in preventing adolescent cyberbullying." Explores parental strategies to prevent cyberbullying among adolescents.
4. Livingstone, S., Haddon, L., Görzig, A., and Ólafsson, K. (2011). "EU Kids Online survey findings on internet risks and safety." Discusses findings from a survey on internet risks and safety among European children.
5. Valcke, M., De Wever, B., Van Keer, H., and Schellens, T. (2011). "Long-term study on safe internet use for young children." Examines the long-term effects of safe internet use among young children.
6. Dishion, T.J., and mcmahon, R.J. (1998). "Parental monitoring for preventing problem behavior." Explores the role of parental monitoring in preventing problem behavior among adolescents.
7. Kerr, M., and Stattin, H. (2000). "Parental knowledge and adolescent adjustment." Investigates the relationship between parental knowledge and adolescent adjustment.
8. Hilal, S., and Gupta, N. (2013). "Web content mining for children's mobile search." Discusses the role of web content mining in improving children's mobile search experiences.
9. Netcraft (2018). "Web server survey." Provides insights into the landscape of web servers as of 2018.
10. Thomas, K., Grier, C., Ma, J., Paxson, V., and Song, D. (2011). "Real-time URL spam filtering service." Describes the design and evaluation of a real-time URL spam filtering service.
11. Narayanan, B.K., Moses, S., and Nirmala, M. (2018). "Filtering adult content to protect minors online." Discusses strategies for filtering adult content to protect minors online.
12. Lee, P.Y., Hui, S.C., and Fong, A.C.M. (2003). "Analysis for web filtering." Presents a structural and content-based analysis for web filtering.
13. Duan, J., and Zeng, J. (2013). "Detecting objectionable web content using topic modeling." Discusses the use of topic modeling for detecting objectionable web content.
14. Liu, S., and Forss, T. (2015). "Text classification models for web content filtering." Discusses text classification models for filtering objectionable web content.
15. Rajalakshmi, R., Tiwari, H., Patel, J., Kumar, A., and Karthik, R. (2020). "Designing a kids-specific URL classifier using neural networks." Presents the design of a URL classifier tailored for children using neural networks.
16. Zaidan, A.A., Karim, H.A., Ahmad, N.N., Zaidan, B.B., and Sali, A. (2013). "AI-based anti-pornography system using skin detection." Reviews an automated anti-pornography system using artificial intelligence.
17. Ma, J., Saul, L.K., Savage, S., and Voelker, G.M. (2009). "Learning to detect malicious websites from suspicious urls." Discusses a machine learning approach for detecting malicious websites from suspicious urls.
18. Rao, R.S., Vaishnavi, T., and Pais, A.R. (2020). "Detecting phishing websites by inspecting urls." Describes a method for detecting phishing websites by inspecting urls.
19. Sahingoz , O.K., Buber, E., Demir, O., and Diri, B. (2019). "Machine learning for phishing detection from urls." Discusses the use of machine learning for detecting phishing websites from urls.
20. Zhao, C., Zhang, Y., Zang, T., Liang, Z., and Wang, Y. (2018). "Identifying objectionable domain names using passive DNS traffic." Presents a method for identifying objectionable domain names using passive DNS traffic.
21. Rohit R, Hasan Firnas I, Abishek M, Albert Mayan J, Dhamodaran S (2023),"Web Application Security Testing Framework using Flask”, International Conference on Applied Artificial Intelligence and Computing, ICAAIC 2023,pp. 1646-1652

**D. CERTIFICATE**

****