

CyberSuraksha

Final Year Project Report

Submitted by

**Shrey Karani (A024),
Faiz Khan (A025),
Tanay Kikla (A028)**

Under The Guidance Of

Prof. Pintu Shah

In fulfillment for the award of the degree of

B.TECH.

INFORMATION TECHNOLOGY

At



**MUKESH PATEL SCHOOL OF
TECHNOLOGY MANAGEMENT
& ENGINEERING**

Department of Information Technology
Mukesh Patel School of Technology Management & Engineering
NMIMS (Deemed –to-be University)
JVPD Scheme Bhaktivedanta Swami Marg, Ville Parle (W), Mumbai-400 056.

March, 2020

CERTIFICATE

This is to certify that the project entitled “CyberSuraksha“ is the bonafide work carried out by Shrey Karani(A024), Faiz Khan(A025) and Tanay Kikla(A028) of B.Tech (IT), MPSTME, Mumbai, during the VIII Semester of the academic year 2019-2020, in partial fulfillment of the requirements for the award of the degree of Bachelors of Technology as per norms prescribed by NMIMS. The project work has been assessed and found to be satisfactory.

(Signature of Internal Mentor 1)

Name:

Designation:

(Signature of External Examiner)

Name:

Designation:

HOD (IT)
(Prof. Pintu Shah)

I/C Dean
(Dr. Vaishali Kulkarni)

DECLARATION

We, Shrey Karani, Faiz Khan, Tanay Kikla, roll numbers: A024, A025, A028 respectively, understand that plagiarism is defined as any one or the combination of the following:

1. Uncredited verbatim copying of individual sentences, paragraphs or illustrations (such as graphs, diagrams, etc.) from any source, published or unpublished, including the Internet.
2. Uncredited improper paraphrasing of pages or paragraphs (changing a few words or phrases, or rearranging the original sentence order)
3. Credited verbatim copying of a major portion of a paper (or thesis chapter) without clear delineation of who did or wrote what. (Source: IEEE, The Institute, Dec. 2004)

We have made sure that all the ideas, expressions, graphs, diagrams, etc., that are not a result of our work, are properly credited. Long phrases or sentences that had to be used verbatim from published literature have been clearly identified using quotation marks.

We affirm that no portion of our work can be considered as plagiarism and we take full responsibility if such a complaint occurs. We understand fully well that the guide of the seminar/project report may not be in a position to check for the possibility of such incidences of plagiarism in this body of work.

Signature:

Name:

Roll No.:

Date:

Signature:

Name:

Roll No.:

Date:

Signature:

Name:

Roll No.:

Date:

ACKNOWLEDGEMENTS

Our deepest appreciation goes to Prof. Pintu Shah whose comments and suggestions were innumerable valuable throughout the course of our project.

This project would not have been successful without the suggestions and support of our professors Prof. Ashwini Rao and Prof. Saurav Verma.

This project couldn't be completed without the effort and cooperation of each group member.

Last but not the least, we would also like to express our gratitude to our family for their moral support and warm encouragements.

Table of Contents

List of Figures	i
List of Tables	ii
Abstract	1
Chapter 1: Overview	3
1.1 Project Specification.....	3
1.2 Literature Survey.....	5
Chapter 2: Analysis & Design.....	9
Introduction.....	9
2.1 Requirement Analysis.....	9
2.1.1 User Interfaces.....	9
2.1.2 Hardware Interfaces.....	9
2.1.3 Communication Interfaces	9
2.2 Feasibility Study.....	10
2.2.1 Financial Feasibility.....	10
2.2.2 Technical Feasibility.....	10
2.2.3 Resource Feasibility.....	11
2.2.4 Social/Legal Feasibility	11
2.3 Design Development	12
2.3.1 Product Breakdown Structure:	14
2.3.2 Database Entity- Relationship Diagram:	16
2.3.3 Gameplay Walkthrough Diagram:.....	18
2.4 Technology and Software Details	21
2.5 Project Planning	23
2.5.1 Chronological Action Plan.....	23
2.5.2 Gantt chart.....	24
2.5.3 Activity Network Diagram.....	25
Chapter 3: Project Description.....	26
3.1 User Sign Up.....	26
3.2 Homepage	26
3.3 Gameplay	26
3.3.1 Network Spoofing.....	26
3.3.2 Smishing.....	26
3.3.3 Malicious Application.....	27
3.3.4 Phishing.....	27
3.3.5 Malvertising	27

3.3.6 Lost Phone.....	27
3.4 Leader Board.....	27
Chapter 4: Project Implementation	28
4.1 Database.....	28
4.2 Managing User Data.....	30
4.3 Data Manager.....	31
4.7 Scenarios.....	33
4.7.1 Menu.....	33
4.7.2 Network Spoofing	34
4.7.3 Smishing	35
4.7.4 Malicious Application	36
4.7.5 Phishing	37
4.7.6 Malvertising	39
4.7.7 Lost Phone.....	40
4.7.8 Final Result	42
Chapter 5: Project Screenshots	44
Chapter 6: Testing Report	51
6.1 Unit Testing.....	51
6.2 System Testing.....	55
Chapter 7: Results and Discussions	56
7.1 The Highest and Lowest Number of Choices	56
7.2 Lost Phone Statistics.....	58
7.3 User Age v Score.....	59
7.4 Education v Score.....	60
7.5 Assumptions.....	61
7.6 Future Work	61
Chapter 8: Conclusions	62
8.1 Conclusion	62
8.2 Future Avenues	62
REFERENCES	63

List of Figures

Figure 1: Product Breakdown Structure.....	15
Figure 2 Database ER Diagram	17
Figure 3: Game Play Walkthrough	20
Figure 4: Gantt Chart	24
Figure 5: Activity Network Diagram	25
Figure 6: Database Setup 1	28
Figure 7: Database Setup 2.....	29
Figure 8: Managing User Data	30
Figure 9: Data Manger 1	31
Figure 10: Data Manager 2.....	32
Figure 11: Menu Backend	33
Figure 12: Network Spoofing Backend.....	34
Figure 13: Smishing Backend.....	35
Figure 14: Malicious Application Backend.....	36
Figure 15: Phishing Backend 1	37
Figure 16: Phishing Backend 2.....	38
Figure 17: Malvertising Backend.....	39
Figure 18: Lost Phone Backend 1	40
Figure 19: Lost Phone Backend 2.....	41
Figure 20: Final Result Backend 1.....	42
Figure 21: Final Result Backend 2.....	43
Figure 22: Login/Signup	44
Figure 23: Sign up.....	44
Figure 25: Main Menu.....	44
Figure 24: Sign in.....	44
Figure 27: Introduction Scene 2.....	45
Figure 26: Introduction Scene 1.....	45
Figure 28: Network Spoofing Screen 1	45
Figure 29: Network Spoofing Screen 2.....	45
Figure 31: Network Spoofing Score Screen.....	46
Figure 30: Network Spoofing Screen 3.....	46

Figure 32: Smishing Screen 1	46
Figure 33: Smishing Screen 2	46
Figure 35: Malicious App Screen 1	47
Figure 34: Smishing Score Screen	47
Figure 37: Malicious App Screen 3	47
Figure 36: Malicious App Screen 2	47
Figure 39: Phishing Screen 1	48
Figure 38: Malicious App Score Screen	48
Figure 41: Phishing Screen 3	48
Figure 40: Phishing Screen 2	48
Figure 43: Malvertising Screen 1	49
Figure 42: Phishing Score Screen	49
Figure 45: Malvertising Score Screen	49
Figure 44: Malvertising Screen 2	49
Figure 47: Phone Lost Screen 2	50
Figure 46: Phone Lost Screen 1	50
Figure 49: Leaderboard Screen	50
Figure 48: Phone Lost Score Screen	50
Figure 50: Types of Software Testing	51
Figure 51: Choices Statistics	56
Figure 52: Lost Phone Statistics	58
Figure 53: User Age v Score	59
Figure 54: Education v Score	60

List of Tables

Table I: Literature Survey	5
Table II: Chronological Action Plan	23
Table III: Unit Testing User Profile/Login	52
Table IV: Unit Testing Scenarios	53

Abstract

Over the past few years India has seen a sharp rise in the number of internet enabled mobile devices and in parallel the number of cybercrimes relating to mobile devices. The country needs a major awareness program that reaches all sections of society. Attempts to tackle this problem have not been very successful because in the past the style of awareness training has not been wide-ranging, or the style of education has been monotonous and thus overlooked by the general populace.

To begin with, we started looking at games and research papers in similar domains. We found cybersecurity awareness games which were directed towards certain groups of people or certain organizations (the military, for instance) and this was the case for most of the games. We tried to find games which could help a regular person with limited experience with technology find the difference between different security threats but to no avail. Keeping this in mind, we wanted to develop an awareness game for the common people, who need it the most as they are the most vulnerable to these attacks because of being unaware. The second thing we tried to think upon was why a game is used as a means to create awareness when people could find out about it very easily just by browsing the internet. Our research into the domain indicated that users respond to and retain information better when it is presented in an interactive way.

Our attempt here is to create awareness in individuals of all sections of the society through a game. We decided to choose a game as according to our research, interaction and information retained have a direct relation. The more interactive the communication between the UI and the user, the more information is retained by the user. The best way to do this was to make a scenario-based trivia game. In the game, the user is presented with various scenarios that he/she can relate to and may encounter in the future. Each level in the game deals with a different threat and the user must make choices when placed in that scenario.

In the game, the user is presented with various scenarios that he/she can relate to and may encounter in the future. Each level in the game deals with a different threat and the user must make choices when placed in that scenario.

After the game has been released, the data generated by the user will be analyzed to check the usability and effectiveness of the game. The data will also indicate which threats the bulk of the population is susceptible to.

Chapter 1: Overview

Introduction

After going through many research papers and online games about cybersecurity, we found that these games are not interactive enough for the users and after a point, become dull and tedious due which users lose interest in the games and at the same time, they are not able to grasp and retain the information that these games are trying to showcase. Secondly, the target audience of these games are certain special people from particular organizations like the military or office employees. There wasn't any game for the common people with limited knowledge about cybersecurity threats. Seeing all this, we wanted to develop a game which could target such people and not a specific set of audience, like other games currently available.[11][16][14]

The objective of the project is to test and analyse cyber security awareness through a cyber-security awareness game and test effectiveness of such application. The application will contain scenarios which will test and educate the users about the need and importance of cyber security. In the latter half of the project we will analyse the data generated by the users after using the game. The final deliverables of this project will be CyberSuraksha, a cyber security game with an analysis report of user behavior.

1.1 Project Specification

Cyber security is the protection of internet-connected systems, including hardware, software and data, from cyber-attacks. In this project, we plan to develop a mobile game to test, enhance and analyze cyber security awareness among the layman.

Cyber security and cyber security awareness are critical to survival in an online world dominated by growing virtual crime. Most people know about costly identity theft and reputation-destroying network hacks. Organizations implement firewalls, comprehensive cyber security defense systems, and sophisticated IT protocols to keep themselves safe from online threats. The problem is, without an embedded culture of cyber security awareness and enforcement, all the measures taken for online protection go to waste.

According to the IC3 annual report for 2018, Internet-enabled theft, fraud, and exploitation were responsible for \$2.7 billion in financial losses in 2018. Compared to \$1.4 billion for the previous year, the rise is staggering. An in-depth review of the report shows a growing trend of financial losses over the years. Another worrying indicator in the report is the continuously rising number of cybercrime complaints, with 2018 showing an average of 900 complaints per day.[1][2]

Focusing on the national level, according to a report by Symantec Corp in 2016, India now ranks 3rd, next to US and China, as a source of malicious activity. Cybercrimes reported in India rose 19 times between the period of 2005 to 2014 and according to the National Crime Records Bureau, Maharashtra reported the highest number of cybercrimes in the state level with Uttar Pradesh, Karnataka, Telangana and Rajasthan next in line. A major factor contributing to the rapid rise of cybercrimes is the availability of cheaper internet and internet enables devices, especially smartphones. The number of smartphone users in India is expected to cross 750 million by 2022 and with this increase comes more frequent cybercrimes. India already ranks second for the number of mobile malware attacks and in 2018, 3000 android malwares were reported each day. Accordingly, research has found that up to 94% of cybersecurity breaches were caused due to human error. Hence, a large number of cybercrimes that occur yearly could be reduced only with awareness knowledge.[2][3][4][5]

1.2 Literature Survey

There are various ways of spreading awareness and providing training regarding the basics of cyber-attacks and defense. Lectures, videos, computer based training and interactive games are some modes of information delivery. Computer based training and video lectures are often used within enterprises for employee education. These methods are often accused of becoming monotonous and thus resulting in lower retention of details. In the research we conducted, we found that although each mode of delivery has its benefits, interactive games promote greater enthusiasm and are likely to finish the course, especially younger users.

There are various games, mostly developed by organizations for internal use that are used for cyber security training. These games are focused on enterprise level issues of cyber security. Other games that are meant for broader audiences focus on only a few specific issues and not a larger pool of issues. Many of the games focus on computer cyber-attacks and not on the more common mobile devices. Hence, we chose to develop a cybersecurity game for a broad audience and chose to deal with cybersecurity issues they are most likely to face.[6]

Table I: Literature Survey

Ref.	Game/Paper	Game Type	Intended Learning	Target Audience	Pros	Cons
[7]	Keep Tradition Secure	1. Online 2. Interactive location	1. Cyber-security 2. Online security	Students and faculty of Texas A&M University	1. Interactive and interesting interface 2. Multiple Incentives to complete the game	1. Target audience Limited 2. Limited Question bank.
[8]	Cyber Ciego	3D virtual world Scenario Simulation	Information Security for enterprise	Science curriculum students	1. Simulations make a more	1. Focused only on Network Security.

					interesting game. 2. In-depth coverage of a particular topic. 3. Scenario development	2. Not open to the public.
[9]	Game of Threats	Card based	Cyber breach (companies being hacked into and losing data)	Board and employees of an organization	Players get to play both as the attackers and the hacker. Impact of the decisions can be observed at real time.	It is a paid application Focused on threats on a organization
[10]	Cyber Bowl	1. Online 2. Multiple choice Q/A	Common cyber-security and best practices	Students and faculty of Florida State University	1. Easy to use. 2. Incentives to complete the game. 3. Covers multiple topics.	1. Not in depth (Basic) 2. Does not instruct on the ways to implement the practices.
[11]	Cybersecurity awareness game utilizing social media framework	Online 2. (Social Networking site)	Common Cybersecurity attacks	Social network users	1. Comprehensive database of questions	1. Not interactive

					2. Leader board and badges	
[12]	The Weakest Link	1. Online 2. Multiple choice Q/A	Common cyber-security and best practices	Employees of an organization	1. Simple to play 2. Provides additional information	1. Target audience is limited.
[13]	Cyber Security Lab	1. Online 2. Different coding, password cracking etc. challenges	1. Social Engineering 2. Cyber-security defences	Students	1. Easy to play 2. Covers multiple topics	1. Monotonous gameplay
[14]	Cyber Awareness Challenge	1. Training Style 2. Problem solving	1. Sensitive Information 2. Malicious Code	Civil and military employees	1. Interactive 2. Detailed	1. Limited audience
[15]	Targeted Attack: The Game	1. Online 2. Video integrated questionnaire	1. Sensitive Company Information	Employees of an organization	1. Detailed 2. Easy to understand	1. Limited audience
[16]	1. Password protector 2. Malware guardian	1. Mobile memory test game 2. Mobile 2D graphic game	1. Setting proper passwords 2. Malware defence	School Students (Age 12-18)	1. Visually appealing 2. Easy to understand and play	1. Limited issues handled 2. Monotonous gameplay 3. Limited audience
[17]	CyberAware	Mobile problem-solving mini games	1. Firewall technologies 2. Antivirus software	Primary school students	1. Engaging gameplay	1. Limited audience 2. Limited issues handled

			3. Email spam		2. ARCS motivational model	
[18]	Enhancing Cybersecurity Learning through an Augmented Reality-based Serious Game	AR Multiplayer game	1. Malware defence 2. Identity Theft 3. Over sharing	High school Students	1. Unique gameplay 2. Simple concepts	1. Limited audience 2. Limited issues handled 3. Difficult implementation

As it is evident from Table I above, in most or all of these games, the main disadvantages identified are either them having a limited target audience or these games being monotonous for the users. For example, the first game analysed, Keep Tradition Secure which is an online game, is targeted towards the faculty and students of Texas A&M University. Another example, Password Protector, has both a limited audience as it is aimed towards school students of ages 12-18, and a monotonous gameplay.[7][16]

On the other hand, the goal of this project was to develop a game which can be played by any individual, of any age, any education level and any occupation. To meet this goal the most common cybersecurity threats that these individuals may face were chosen. Making this game a scenario-based trivia game was an attempt to make it more interactive for the users so that they don't find it monotonous like the other games identified. Doing this will also lead to retention of more information by the users.[20]

Chapter 2: Analysis & Design

Introduction

To develop an effective game, it was essential to understand the key areas to consider when designing digital learning games. Through research into this field three key areas of game design were identified, namely; Immediate actionable feedback, Narrative context, i.e., prioritizing learning content and crafting other aspects such as story and rewards around the learning objectives and lastly, Agency, i.e., to inspire users through the messages provided in the game to enact better security standards. The game was designed keeping these three points in mind and the best example for this comes from the narration. The narration provides immediate feedback with information that is actionable and fits the context of the game. Along with information, the narration along with the points system is the primary source of agency in the game, it provides positive and negative feedback based on player choices. The game was developed by following an iterative development methodology, developing a scenario first, playtesting and evaluating it before developing the next scenario.[19]

2.1 Requirement Analysis

2.1.1 User Interfaces

The user interface is an application which can be accessed through any Android smartphone. The UI is very intuitive for the user, it is clear and works the way a user will expect it to. On starting the application, the user will see a login screen and after logging in, they will go through a few cybersecurity related scenarios where a question will be posited to them in each scenario and they will have to answer those using buttons. The UI and gameplay are very straight forward as the user will also get a narrative of what is going on in each scene.

2.1.2 Hardware Interfaces

- An Android smartphone

2.1.3 Communication Interfaces

- Wi-Fi router or Mobile Data for an internet connection

2.2 Feasibility Study

2.2.1 Financial Feasibility

The project does not have any expenses. Although the system consists of a login system so the operation requires a bandwidth. Users would not be incurring any costs. The target audience are the people who are very prone to cybersecurity attacks because of their incautious nature. Additionally, no specific hardware or software was required for development of the game or the database. The tools used are all open source and all the assets required were developed in house. The proposal is financially feasible financially.

2.2.2 Technical Feasibility

The project is an Android application. The tools and technologies used for the implementation of this project are:

- For game development: Unity3D Engine
Unity is a cross-platform game engine which can be used to create three-dimensional, two-dimensional, virtual reality and augmented reality games and other simulations and experiences. Unity is the best tool for us as it has user friendly features, allowing us to evaluate changes instantly in play mode. The scripting language used (C#) is also easy.
- For assets: Magica Voxel
Magica Voxel is a free lightweight 8-bit voxel art editor and GPU based interactive path tracing renderer. It enabled us to create our own assets compatible with Unity instead of buying them online. This helped us give the game a unique and personal identity. Magica Voxel runs on most systems and has minimal hardware requirements.
- For login system: Firebase
Firebase is a mobile and web application development platform which provides services such as analytics, authentication, real-time database etc. Firebase was the best choice as Firebase allows sign-in option using email & password, Google, Facebook etc. It also allows users to store data in real-time.

Each of the technologies are freely available and the technical skills required are manageable.

2.2.3 Resource Feasibility

Resources that are required for the project includes,

- Programming Device (Laptop)
OS: Windows 7 SP1+, 8, 10, 64-bit versions only; macOS 10.12
- Programming Tools
Visual Studio
- Programming Individuals
- Certification Courses

2.2.4 Social/Legal Feasibility

This project is made with the help of freely available development tools and provide the system as an open source system. There are no social/legal issues associated with it.

2.3 Design Development

The first aspect to be decided for any game is the type and its corresponding gameplay. Through the literature survey, various types of games designs that could be used for this project were identified. Some of the leading candidates were role-playing simulation games, card-based games, questionnaires and puzzle-based games. To appeal to the broadest audience possible while also ensuring they complete the game, the game mechanics had to be simple and not divert the players attention from the purpose of the game, hence card-based and puzzle-based gameplay were removed from consideration. The game also had to be interesting and needed to keep the player invested, this was not possible with questionnaires as players often feel like they are taking a test when playing such games. Thus, the optimal choice was a (RPG) role-playing game. RPGs keep the player engaged while also not diverting them with unnecessary learning curves. An RPG would essentially just put the player in the shoes of a character. This would enable the development of various scenarios that the character goes through.

During development, the scenarios were designed to mimic the experiences of the common man. This was decided to show the user that they too could and might already have faced the threats represented in the game. The scenarios were chosen by analyzing the relevance of various threats and the scenario they may occur in, to the common population. Hence, it was decided that the game would be a RPG that depicts the journey of a common person from their home to work and back home again, with each location along the way representing a new level and new threat.

Following the theme of RPGs, an important element of these games is the idea of text and/or audio narration. It is a tried and tested way of conveying information to the player, from older RPG games such as Diablo and Fallout to more recent indie titles such as Stardew Valley, narration has proved to be a simple and effective way of sharing information with the player especially in lower budget game productions. Hence, this element was included in the game to as simple way of sharing information while also following in the footsteps of classic RPGs. The narration of the game was decided while keeping in mind the principle of positive and

negative feedback, i.e., rewarding players with positive feedback when they make a right decision and providing a strong warning when they make a wrong choice. [21][22][23]

The gameplay was decided to be a simple one touch and best answer selection type game to keep it simple enough for any kind of user but still short and engaging enough to keep users invested. The assets and environment for the game were chosen as a result of their times, art style of blocks pixels in fashion especially in mobile games.

2.3.1 Product Breakdown Structure:

The product breakdown structure catalogues the desired products needed to achieve the project. It provides an exhaustive structure of the deliverables that make up the project. The project is divided into five broad product categories namely, Models, Sound, Animations, Database and Logic.

The models include the characters and their personal effects along with all the other objects, environmental settings and structures. Also included are the screens that the player interacts with and the rewards that they receive.

The sound production includes the music as well as narration. These sounds and music elements have been collected and edited to suit the game.

The animations in the game include the character animations, the animation of moving objects and the transitions of the screens and environment.

The database is hosted on google firebase and hosts user details and the answers they select. This information is used for login authentication, points calculation and data analysis.

The game logic is an essential part of the product as it includes the functions required to calculate the points for each user, the narration to be played based upon the choice a user makes, the login/signup function and the transfer to and from the database.

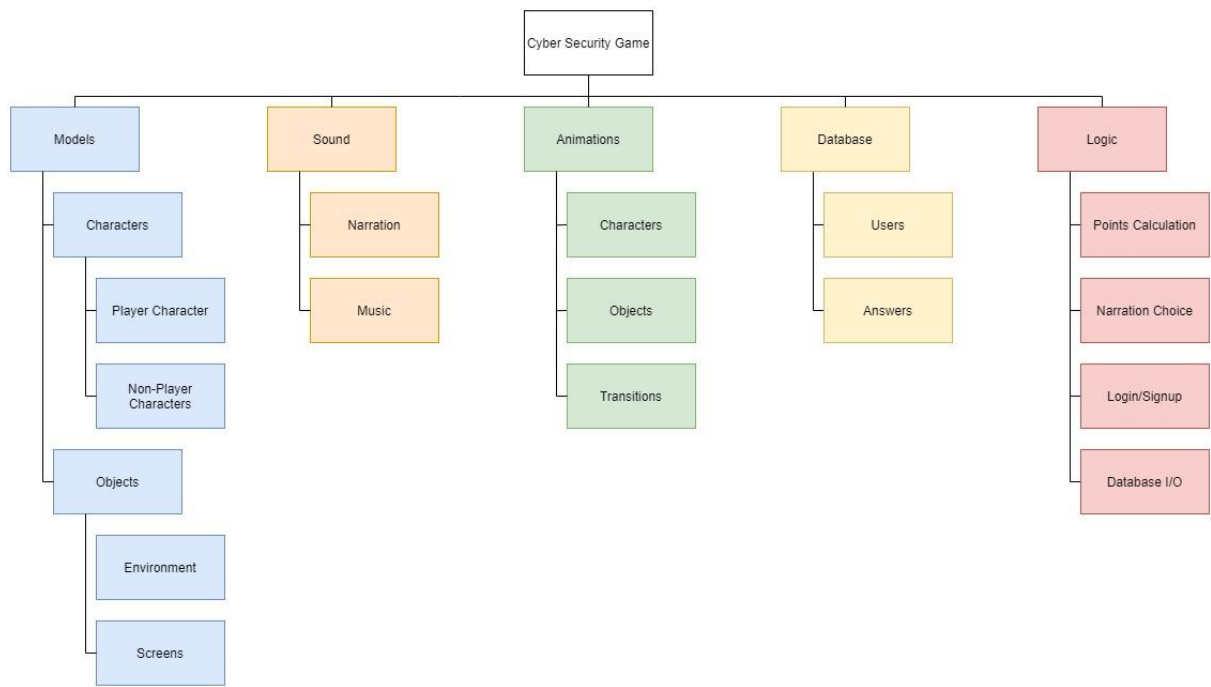


Figure 1: Product Breakdown Structure

2.3.2 Database Entity- Relationship Diagram:

We have used the Firebase Realtime Database for data storage and analytics.

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in real time to every connected client. When programs need to communicate with each other over networks, they have to agree on a format for any data that they exchange with each other.

The Database contains a root node -> done-ae5d8

There are 2 child nodes to the root node -> users and usersAnswers

Each user has a unique UserID which acts as a primary key and connects the two children node.

The users Node contains the following fields: -

1. UserID: varchar
2. Name: char
3. Education: char
4. Occupation: char
5. HasAntiVirus: boolean
6. Age: int

The usersAnswers Node contains the following fields: -

1. UserID: varchar
2. Name: char
3. Network Spoofing: char
4. Smishing: char
5. MaliciousApplication: char
6. Malvertising: char
7. LostPhone: char
8. Score: int

The Database stores the following information and links each user to his/her selected answers.

The following data is further used to perform analytics based on different criteria

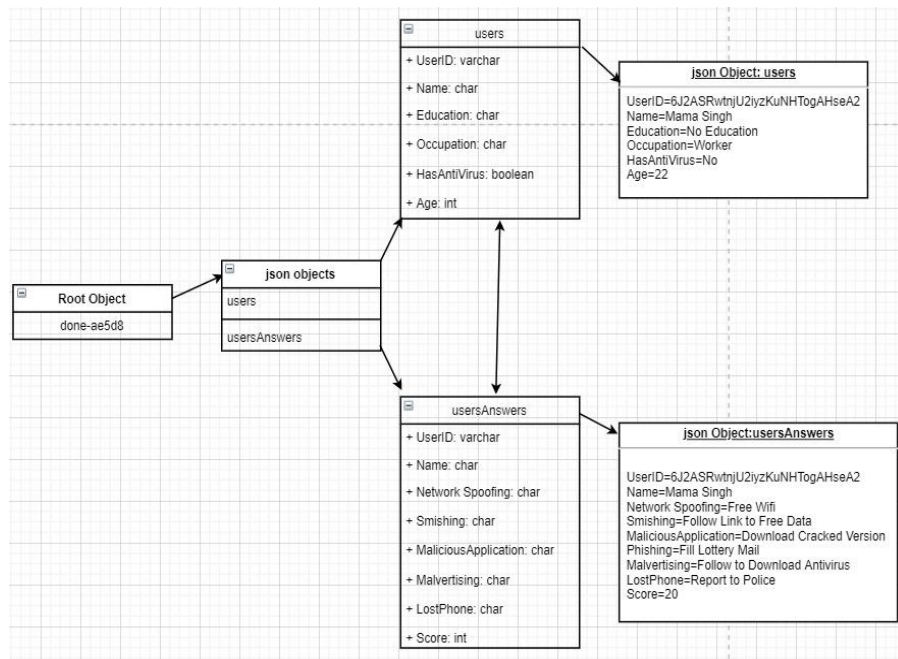


Figure 2 Database ER Diagram

2.3.3 Gameplay Walkthrough Diagram:

Throughout the game the player will be asked to make choices on behalf of the central character Ahaan. Based on the choices the player makes, he or she will receive points to be displayed on the leaderboard.

After signing up or signing in the actual game begins when the character leaves his house for a nearby café. At the café the character takes a seat and pulls out his phone to read the news. Here the character has three options, either to use the café's free Wi-Fi service, a paid password protected Wi-Fi or his own mobile data.

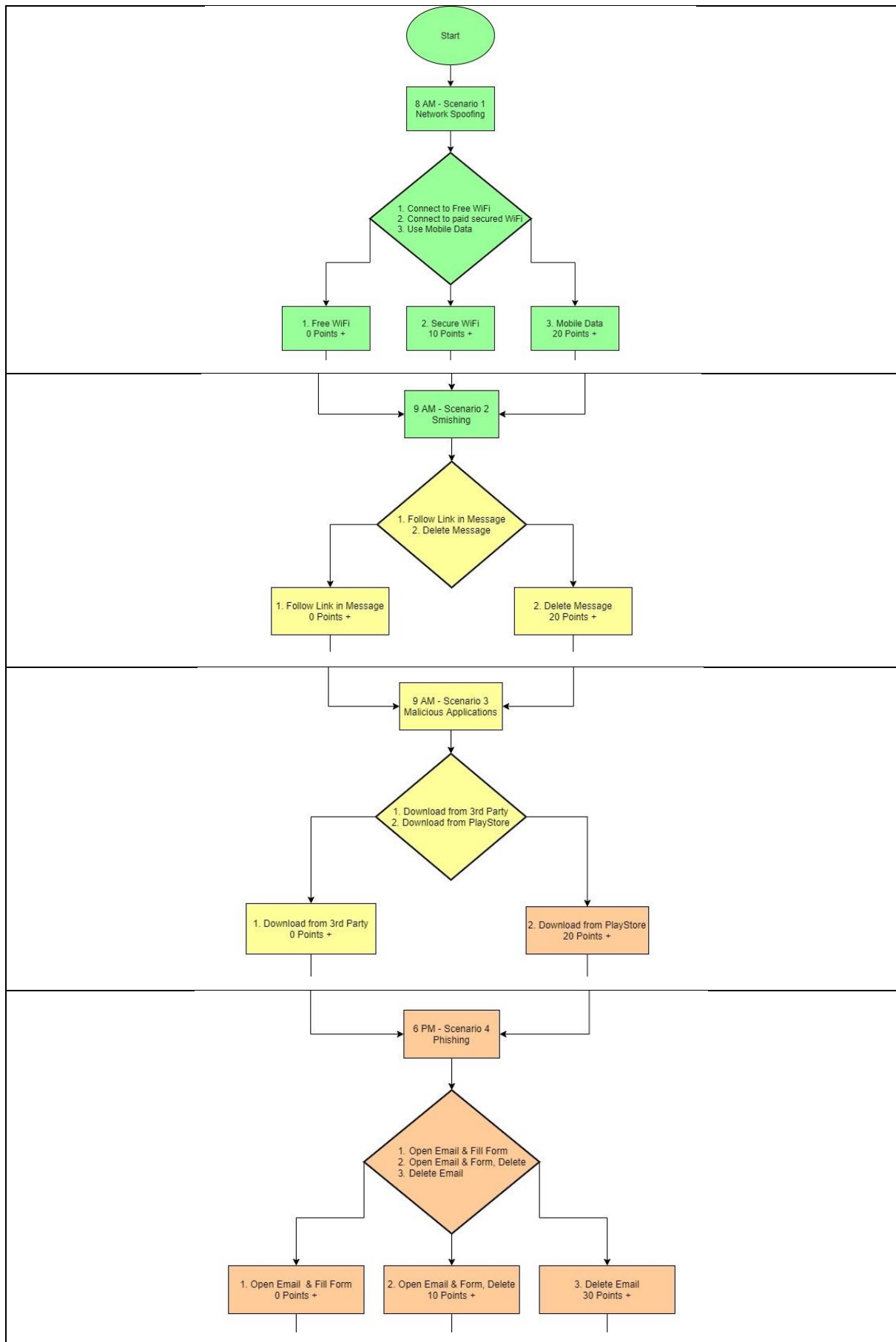
The character leaves the café to catch his bus to work. He stops at the bus station when he receives a message indicating he has received additional 10GB of mobile data free of cost, all he has to do is follow the link in the message. The player may follow or delete this message.

After this the bus arrives and the character enters and takes a seat in the bus. While seated the character gets a glance of the game the passenger seated beside him is playing. The character has nothing to do and is intrigued by the game, so he searches for it on the internet. He finds out that the game is a paid application on the official store but also finds a cracked version of the game for free on a third-party store. The player must decide the store the character downloads the game from.

Once at the office, the character starts his work. At some point in time he receives an email proclaiming that he is the winner of a lucky jackpot and must follow the link in the email to a website to fill in his details for the transfer of funds. The player must decide what action the character takes.

After work is over, the character takes the bus back home. On the bus, while streaming a TV series on his phone a popup appears on the browser warning him about a virus that has infected his device and that he must install the appropriate anti-virus to get rid of the virus from his phone. The player must decide if the threat is genuine or benign and decide whether to install the antivirus or ignore the message.

The character gets off the bus when he reaches his destination. At the door of his home he realizes he has misplaced his phone. The player must decide what must be the first thing the character must do in this situation. This is the last scenario of the game after which the player is presented with the leaderboard and the trophy he or she has earned.



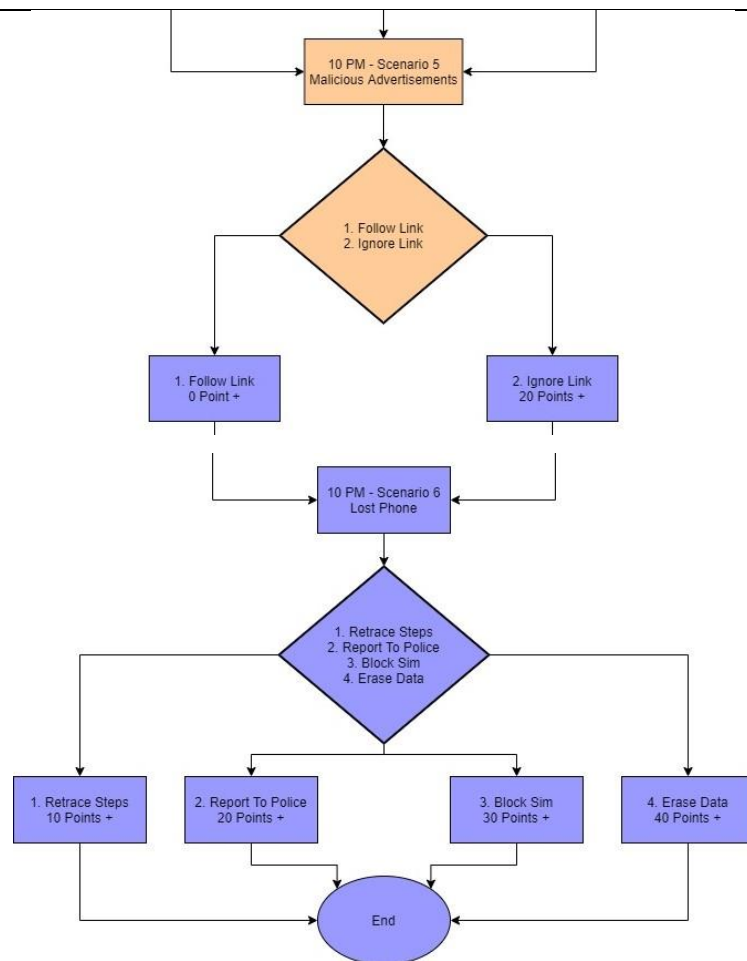


Figure 3: Game Play Walkthrough

2.4 Technology and Software Details

To reach the widest possible audience we decided to implement the game in Android as most of the population in India is using Android enabled devices. To develop the game, we use Unity as the game engine and Magica Voxel to make the individual assets. The characters and environments were first modeled and prototyped in Magica Voxel and later imported to Unity to include animations, scene changes and integration with android. For the gameplay we decided to go with a minimalistic interface approach with easy to understand interactions with a focus on replicating the situations any person may come across in their daily routine. The simple one touch controls ensure that game is easy to play while the environments and perks are included to keep the user engaged and the game does not feel like a chore.

Unity is a cross-platform game engine developed by Unity Technologies. Unity gives developers the ability to create games in both 2D and 3D, and the engine offers a primary scripting API in C#, for both the Unity editor in the form of plugins, and games themselves. We chose to use Unity because of its ease of use and community support for any technical issues.

For development on Unity the installation requirements are:

- OS: Windows 7 SP1+, 8, 10, 64-bit versions only; macOS 10.12+
- CPU: SSE2 instruction set support.
- GPU: Graphics card with DX10 (shader model 4.0) capabilities.

For running Unity games, the requirements are:

Desktop:

- OS: Windows 7 SP1+, macOS 10.12+, Ubuntu 16.04+
- Graphics card with DX10 (shader model 4.0) capabilities.
- CPU: SSE2 instruction set support.

iOS:

- iOS 9.0 or higher.

Android:

- OS 4.1 or later; ARMv7 CPU with NEON support or Atom CPU; OpenGL ES 2.0 or later.

WebGL:

- Any recent desktop version of Firefox, Chrome, Edge or Safari.

For development on Magica Voxel the minimum installation requirements are:

- OS: Windows 7 SP1+, 8, 10; macOS 10.12+
- CPU: Intel Core i3-9100
- GPU: AMD Radeon RX 560X
- RAM: 8 GB

2.5 Project Planning

2.5.1 Chronological Action Plan

Table II: Chronological Action Plan

SN	Activity	Duration in days	Start Date	End Date	Predecessors
1	Domain Analysis	8	08-07-19	16-07-19	-
2	Literature Review	9	17-07-19	25-07-19	Domain Analysis
3	Structural and Gameplay Design	31	01-08-19	31-08-19	Literature Review
4	UI Design	20	01-09-19	20-09-19	Structural and Gameplay Design
5	Game Development	31	21-09-19	25-12-19	UI Design
6	Application Development	39	25-12-19	25-01-20	UI Design
7	Usability Testing and Improvements	15	01-01-20	10-02-20	Game Development, Application Development
8	Data Collection and Data Analysis	35	10-02-20	20-02-20	Usability Testing and Improvements
9	Black Book and Report Generation	16	21-02-20	15-03-20	Data Collection and Data Analysis

2.5.2 Gantt chart

The timeline of the project is illustrated through the Gantt chart. It presents the schedules, duration and overlapping of each of the activities.

The project was initiated by conducting a domain analysis to understand the field of games as a medium of awareness and knowledge. Next, we began the analysis of previous games and research work dealing with cybersecurity awareness. During this period, we went through nearly twelve games and research papers. After the review we began designing the structural and gameplay elements of the game. Through the research we had finalized on six threats to focus on. Now we were evaluating the best ways to present those threats in a simple manner. After this we began prototyping the UI design, and possible asset designs for the game. Game development began with the design of the assets and was followed by the development of the database and android application. In the final stages of the project the game was tested for usability to make slight improvements to the game, data was collected from users for analysis and finally the report and documentation was produced.

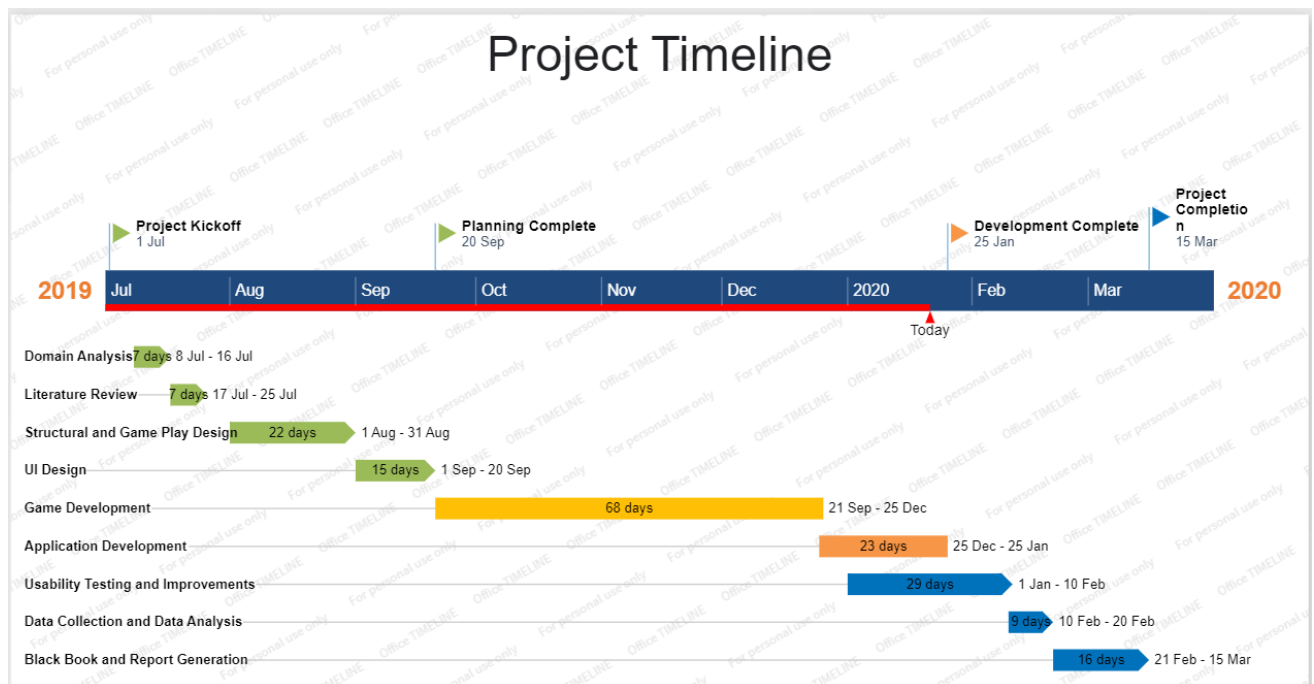


Figure 4: Gantt Chart

2.5.3 Activity Network Diagram

The activity network diagram presents the activities involved in the project as well as their sequencing. It presents the tasks involved in a simple manner for easier referencing.

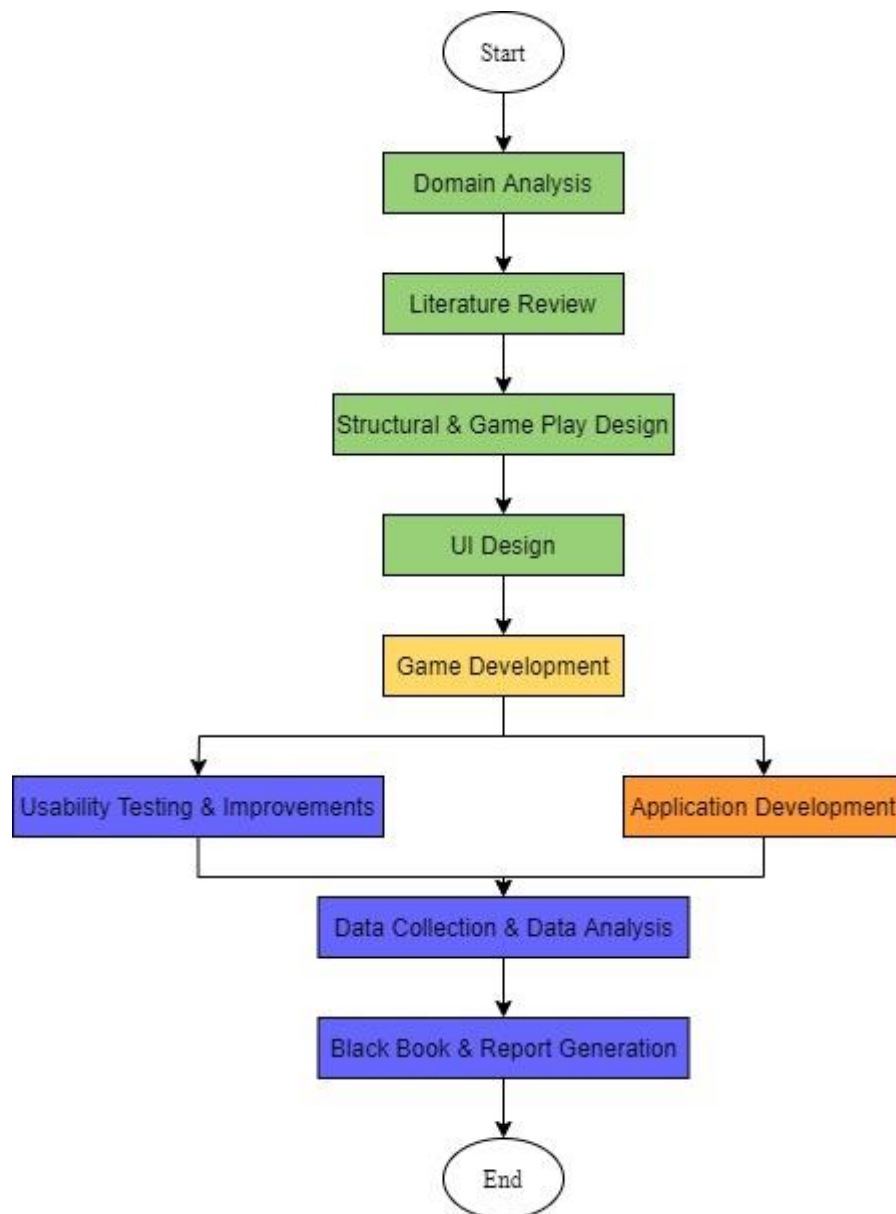


Figure 5: Activity Network Diagram

Chapter 3: Project Description

3.1 User Sign Up

When the users open the application, they will be required to sign up using an email id and a password. Information about their age, level of education, occupation, etc. will be asked and all this data will be stored in Firebase real-time database. All this data is required to make a report about the awareness of the users.

3.2 Homepage

After the user has signed in or signed up, they are taken to the homepage where the user can choose to play the game, logout or quit the game.

3.3 Gameplay

After the user has signed in/up, they will have to go through a total of six scenarios namely, 'Network Spoofing', 'Smishing', 'Malicious Application', 'Phishing', 'Malvertising' and 'Lost Phone'.

3.3.1 Network Spoofing

Network spoofing is when someone thinks an unknown network to be a known one or trusts an open network without knowing the consequences of it connecting to it.

In the first scenario, the character walks from his home to the café where he is asked to choose to connect to an open WiFi or his mobile network. The user now has to choose from the two options. Depending on the option the user chooses, they either gain points or get no points.

3.3.2 Smishing

Smishing is an attack where someone receives an SMS containing a malware. Smishing is short for 'SMS Phishing'.

In the second scenario, the character receives a message indicating to click on a link which could get him 10 GB free data. The user now has to choose whether to follow the link or delete the text.

3.3.3 Malicious Application

Malicious applications are those which contain a malware. These are usually the free applications available on third party stores.

In the third scenario, the character wants to download an application but has to choose whether to download it from a third-party store or pay for it using the official application store.

3.3.4 Phishing

Phishing is a cybercrime where someone is lured into downloading an attachment, file etc. through email to get his/her personal information including bank account details.

In the fourth scenario, the character receives an email saying that he has won a lottery of Rs. 30,000 and to claim it, he would have to fill a form which asks him about his bank account details. Now, the user has to decide whether to fill the form or delete the email.

3.3.5 Malvertising

Malvertising (Malicious Advertising) is the use of online advertising to spread malware. In the fifth scenario, the character sees a pop-up ad while browsing the internet. The ad says that his phone has been infected with a virus with a link to download an antivirus. The user now has to decide whether to follow the link or not.

3.3.6 Lost Phone

In the sixth scenario, the character has lost his phone and the user has four options to choose from. Selecting the safest option will lead to the user to gain the maximum points.

3.4 Leader Board

Once the game is complete, the user is shown the leader board for the game with the user's total points, trophy earned and top three players along with their scores. The user can go back to homepage from the leader board screen.

Chapter 4: Project Implementation

4.1 Database

Setting up Firebase and integrating it to the unity project.

FireBase is used in this project for Analytics and Authentication.

The main libraries used are: -

- Firebase
- Firebase.Analytics
- Firebase.Database
- Firebase.Unity.Editor
- System.Security.Cryptography

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

using Firebase;
using Firebase.Analytics;
using Firebase.Database;
using Firebase.Unity.Editor;

using System;
using System.IO;
using System.Text;
using System.Security.Cryptography;

//$uid == auth.uid"

/// <summary>
/// Firebase manager - 
/// </summary>
14 references
public class FirebaseManager : MonoBehaviour
{
    public static FirebaseManager instance;

    public Firebase.Auth.FirebaseUser firebaseUser;
    public DatabaseReference firebaseDatabaseReference;

    // -----
    private Firebase.Auth.FirebaseAuth auth;

    // -----
    #region SYSTEM_METHODS
    // -----
```

Figure 6: Database Setup 1

InitializeFirebase(Loader loaderRef) is used to connect the firebase engine with the unity project. It also assigns a database to the unity project which in this case is done using SetEditorDatabaseUrl("https://done-ae5d8.firebaseio.com/").

Along with this it has basic functionalities like Signup, Signin, Login and Logout.

```

public void InitializeFirebase(Loader loaderRef)
{
    FirebaseApp.CheckAndFixDependenciesAsync().ContinueWith(task =>
    {
        Debug.Log("nnn");
        var dependencyStatus = task.Result;
        if (dependencyStatus == Firebase.DependencyStatus.Available)
        {
            Debug.Log("Avai");
            // Create and hold a reference to your FirebaseApp,
            // where app is a Firebase.FirebaseApp property of your application class.
            // app = Firebase.FirebaseApp.DefaultInstance;

            // Set a flag here to indicate whether Firebase is ready to use by your app.

            FirebaseAnalytics.SetAnalyticsCollectionEnabled(true);

            auth = Firebase.Auth.FirebaseAuth.DefaultInstance;

            FirebaseApp.DefaultInstance.SetEditorDatabaseUrl("https://done-ae5d8.firebaseio.com/");

            firebaseDatabaseReference = FirebaseDatabase.DefaultInstance.RootReference;

            loaderRef.LoadLoginScene (true);
        }
        else
        {
            UnityEngine.Debug.LogError(System.String.Format (
                "Could not resolve all Firebase dependencies: {0}", dependencyStatus));
            // Firebase Unity SDK is not safe to use here.

            loaderRef.DisplayErrorMessage("Could not resolve all Firebase dependencies:" + dependencyStatus);
        }
    });
}

```

Figure 7: Database Setup 2

4.2 Managing User Data

```
public void StoreUserDetails (string _userName, int _userAge, string _userOccupation, string _userEducation, bool _userHasAntiVirus)
{
    DataManager.instance.userAnswersInfo = new DataManager.UserAnswersInfo(firebaseUser.UserId, _userName);
    PlayerPrefs.SetString("UserName", _userName);

    Debug.Log("Creating user");
    DataManager.UserInfo userInfo = new DataManager.UserInfo(_userName, _userAge, _userOccupation, _userEducation, _userHasAntiVirus, firebaseUser.UserId);
    Debug.Log("Uploading data");
    Debug.Log(JsonUtility.ToJson(userInfo));

    firebaseDatabaseReference.Child("users").Child(firebaseUser.UserId).SetRawJsonValueAsync(JsonUtility.ToJson(userInfo)).ContinueWith(task =>
    {
        if (task.IsCanceled || task.IsFaulted)
        {
            LoginManager.instance.ShowErrorMessage("Error Creating User.");
            return;
        }
        //else if (task.IsCompleted)
        {
            Debug.Log("Load menu");
            LoginManager.instance.LoadMenuScene();
        }
    });
}
```

Figure 8: Managing User Data

This method is used to store the details of a user in the database.

public void StoreUserDetails (string _userName, int _userAge, string _userOccupation, string _userEducation, bool _userHasAntiVirus)

Is used to create an instance of a user and save their respective details to done-ae5d8.

In the FireBaseManager an instance of the DataManager is created to capture the information of the user.

4.3 Data Manager

```
public class UserAnswersInfo
{
    public string UserID;
    public string Name;

    public string NetworkSpoofing;
    public string Smishing;
    public string MaliciousApplication;
    public string Phishing;
    public string Malvertising;
    public string LostPhone;

    public int Score;
    2 references
    public UserAnswersInfo (string _userID, string _name)
    {
        this.UserID = _userID;
        this.Name = _name;
    }
    1 reference
    internal static T[] GetJsonInfoFromString<T>(string rawJsonValue)
    {
        if (rawJsonValue != null)
        {
            //Debug.Log("Response string = " + jsonString);
            //if (jsonString.Length > 69780)
            //Debug.LogError("Character at " + jsonString.Substring(69779));
            return JsonHelper.FromJson<T>(rawJsonValue);
        }
        else
            Debug.Log("Null response string");
        return null;
    }
}

public class UserInfo
{
    public string Name;
    public int Age;
    public string Occupation;
    public string Education;
    public bool HasAntiVirus;
    public string UserID;

    1 reference
    public UserInfo(string _name, int _age, string _occupation, string _education, bool _hasAntiVirus, string _userID)
    {
        this.Name = _name;
        this.Age = _age;
        this.Occupation = _occupation;
        this.Education = _education;
        this.HasAntiVirus = _hasAntiVirus;
        this.UserID = _userID;
    }
}

2 references
public string GetRandomString ()
{
    const string glyphs = "abcdefghijklmnopqrstuvwxyz0123456789";
    string randomString = "";

    int charAmount = UnityEngine.Random.Range(5, 9); //set those to the minimum and maximum length of your string

    for (int i = 0; i < charAmount; i++)
    {
        randomString += glyphs[UnityEngine.Random.Range(0, glyphs.Length)];
    }
}
```

Figure 9: Data Manger 1

```

using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

/// <summary>
/// Data manager
/// </summary>
83 references
public class DataManager : MonoBehaviour
{
    public static DataManager instance;

    // 1 - Free Wifi
    // 2 - Secure Wifi
    // 3 - Mobile Data
    public int selectedInternetOption;

    public int userScore;

    public UserAnswersInfo userAnswersInfo;

    0 references
    private void Start()
    {
        instance = this;
    }

    [System.Serializable]
    15 references

```

Figure 10: Data Manager 2

DataManager is used to capture the users selected answers and upload it on the FireBase. public int selectedInternetOption saves the users selected option as an integer.

Public int score counts the cumulative score of a user after every scenario.

Public userAnswersInfo() links a user's selected answer with the respective userID.

4.7 Scenarios

4.7.1 Menu

The Main Menu has 3 options:

Play: - Selecting the Play button initiates the game and transitions to the first scene.

Logout: - Selecting the logout button, allows the user to log out of the current account.

Quit: - Selecting the Quit button terminates the game.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

0 references
public class MainMenu : MonoBehaviour
{
    int counter = 0;

    0 references
    public void PlayGame()
    {
        SceneManager.LoadScene("InitialScene");
        DataManager.instance.userScore = 0;
    }

    0 references
    public void Logout ()
    {
        FirebaseManager.instance.SignOut();
        SceneManager.LoadScene("Login");
    }

    0 references
    public void QuitGame()
    {
        Debug.Log("QUIT!");
        Application.Quit();
    }
}
```

Figure 11: Menu Backend

Scenarios: - Every Scenario has a basic structure of a dialogue box pop up, followed by the actual scenario questions and ending with a feedback dialogue and a score tracker.

Each Scenario has a code to do the same and at the same time, record the user's selection and transition to the next scene.

Inumerator() initialises the dialogue box before each scene.

LoadQuestion() loads the question of the particular scene on the screen.

OnButtonClicked() record the answer against the respected user id.

LoadNextScene() initializes the transition from one scene to the other.

4.7.2 Network Spoofing

```

private void Start()
{
    hasDoneShowingDialogues = false;
    StartCoroutine(TypeSentence(dialogueString));
    scoreTextRef.text = DataManager.instance.userScore.ToString();

    //hasDoneShowingDialogues = true;
}

0 references
public void OnClick()
{
    Debug.Log("cafe2");
    //SceneManager.LoadScene("CafeScene2");
    if (hasDoneShowingDialogues)
    {
        Debug.Log("Showing");
        FadeEffectManager.instance.FadeIn();
        Invoke("LoadNextScene", 0.7f);
    }
}

0 references
private void LoadNextScene()
{
    SceneManager.LoadScene("MobileScene");
}

1 reference
IEnumerator TypeSentence(string sentence)
{
    yield return new WaitForSeconds(1);

    dialogueText.text = "";
    foreach (char letter in sentence.ToCharArray())
    {
        dialogueText.text += letter;
        yield return new WaitForSeconds(0.01f);
    }
    hasDoneShowingDialogues = true;
}

0 references
public class CafeScene2 : MonoBehaviour
{
    [SerializeField]
    string freeWifiString, secureWifiString, mobileDataString;

    [SerializeField]
    Text dialogueText;

    [SerializeField]
    Sprite goldTrophyRef, silverTrophyRef, bronzeTrophyRef;

    [SerializeField]
    Image trophyImageRef;

    [SerializeField]
    Text scoreTextRef;

    public bool hasDoneShowingDialogues;
    0 references
    private void Start()
    {
        hasDoneShowingDialogues = false;
        if (DataManager.instance.selectedInternetOption == 1)
        {
            trophyImageRef.sprite = bronzeTrophyRef;
            StartCoroutine(TypeSentence(freeWifiString));
        }
        else if (DataManager.instance.selectedInternetOption == 2)
        {
            StartCoroutine(TypeSentence(secureWifiString));
            trophyImageRef.sprite = silverTrophyRef;
        }
        else if (DataManager.instance.selectedInternetOption == 3)
        {
            StartCoroutine(TypeSentence(mobileDataString));
            trophyImageRef.sprite = goldTrophyRef;
        }

        scoreTextRef.text = "Your Score : " + DataManager.instance.userScore;

        //hasDoneShowingDialogues = true;
    }

    public void OnClick()
    {
        Debug.Log("cafe2");
        //SceneManager.LoadScene("CafeScene2");
        if (hasDoneShowingDialogues)
        {
            Debug.Log("Showing");
            FadeEffectManager.instance.FadeIn();
            Invoke("LoadNextScene", 0.7f);
        }
    }

    0 references
    private void LoadNextScene()
    {
        SceneManager.LoadScene("BusStopWalk");
    }

    3 references
    IEnumerator TypeSentence(string sentence)
    {
        yield return new WaitForSeconds(1);

        dialogueText.text = "";
        foreach (char letter in sentence.ToCharArray())
        {
            dialogueText.text += letter;
            yield return new WaitForSeconds(0.01f);
        }
        hasDoneShowingDialogues = true;
    }
}

```

Figure 12: Network Spoofing Backend

4.7.3 Smishing

```

void Start()
{
    dialoguePanelRef.SetActive(true);
    resultsPanelRef.SetActive(false);

    hasDoneShowingDialogues = false;
    hasDoneShowingAnswerDialogue = false;

    previousScoreTextRef.text = DataManager.instance.userScore.ToString();

    questionsPanelRef.SetActive(false);
    StartCoroutine(TypeSentence(dialogueString));
}

3 references
IEnumerator TypeSentence(string sentence)
{
    yield return new WaitForSeconds(1);

    dialogueText.text = "";
    foreach (char letter in sentence.ToCharArray())
    {
        dialogueText.text += letter;
        yield return new WaitForSeconds(0.01f);
    }
    if (hasDoneShowingDialogues)
        hasDoneShowingAnswerDialogue = true;
    else
        hasDoneShowingDialogues = true;
}

0 references
public void LoadQuestion()
{
    if (hasDoneShowingDialogues)
    {
        questionsPanelRef.SetActive(true);
        dialoguePanelRef.SetActive(false);
        questionPanelAnimatorRef.SetBool("FadeIn", true);
        dialogueText.text = "";
    }
}

public void OnFollow()
{
    questionPanelAnimatorRef.SetBool("FadeIn", false);

    dialogueText = answerTextRef;
    trophyImageRef.sprite = wrongSprite;
    scoreTextRef.text = "Your Score : " + DataManager.instance.userScore;
    previousScoreTextRef.transform.parent.gameObject.SetActive(false);
    resultsPanelRef.SetActive(true);

    DataManager.instance.userAnswersInfo.Smishing = "Follow Link to Free Data";

    Invoke("HideQuestionsPanel", 0.5f);
    StartCoroutine(TypeSentence(followString));
}

0 references
public void OnDelete()
{
    DataManager.instance.userScore += 20;
    questionPanelAnimatorRef.SetBool("FadeIn", false);

    dialogueText = answerTextRef;
    trophyImageRef.sprite = goldSprite;
    scoreTextRef.text = "Your Score : " + DataManager.instance.userScore;
    previousScoreTextRef.transform.parent.gameObject.SetActive(false);
    resultsPanelRef.SetActive(true);

    DataManager.instance.userAnswersInfo.Smishing = "Delete Link to Free Data";

    Invoke("HideQuestionsPanel", 0.5f);
    StartCoroutine(TypeSentence(deleteString));
}

0 references
private void HideQuestionsPanel()
{
    questionsPanelRef.SetActive(false);
}

0 references
private void LoadNextScene()
{
    SceneManager.LoadScene("BusArrival");
}

```

Figure 13: Smishing Backend

4.7.4 Malicious Application

```
public void OnFreeButtonClick()
{
    questionPanelAnimatorRef.SetBool("FadeIn", false);

    dialogueText = answerTextRef;
    trophyImageRef.sprite = wrongOptionSprite;
    scoreTextRef.text = "Your Score : " + DataManager.instance.userScore;
    previousScoreTextRef.transform.parent.gameObject.SetActive(false);
    resultsPanelRef.SetActive(true);

    DataManager.instance.userAnswersInfo.MaliciousApplication = "Download Cracked Version";

    Invoke("HideQuestionsPanel", 0.5f);
    StartCoroutine(TypeSentence(freeString));
}

0 references
public void OnBuyButtonClick()
{
    DataManager.instance.userScore += 20;
    questionPanelAnimatorRef.SetBool("FadeIn", false);

    dialogueText = answerTextRef;
    trophyImageRef.sprite = goldSprite;
    scoreTextRef.text = "Your Score : " + DataManager.instance.userScore;
    previousScoreTextRef.transform.parent.gameObject.SetActive(false);
    resultsPanelRef.SetActive(true);

    DataManager.instance.userAnswersInfo.MaliciousApplication = "Buy Game";

    Invoke("HideQuestionsPanel", 0.5f);
    StartCoroutine(TypeSentence(buyString));
}

0 references
private void HideQuestionsPanel()
{
    questionsPanelRef.SetActive(false);
}

0 references
private void LoadNextScene()
{
    UnityEngine.SceneManagement.SceneManager.LoadScene("BusReachesOffice");
}
```

Figure 14: Malicious Application Backend

4.7.5 Phishing

```
public void LoadQuestion()
{
    if (hasDoneShowingDialogues)
    {
        questionsPanelRef1.SetActive(true);
        questionPanelAnimatorRef1.SetBool("FadeIn", true);
        dialogueText.text = "";
    }
}
0 references
private void HideQuestionsPanel2()
{
    questionsPanelRef2.SetActive(false);
}
0 references
public void OnOpenButtonClick()
{
    dialoguePanelRef.SetActive(false);
    questionPanelAnimatorRef1.SetBool("FadeIn", false);
    hasDoneShowingAnswerDialogue1 = true;
    Invoke("HideQuestionsPanel1", 0.5f);
    Invoke("LoadSecondQuestion", 0.5f);
}

0 references
public void OnDeleteButtonClick()
{
    DataManager.instance.userScore += 20;
    questionPanelAnimatorRef1.SetBool("FadeIn", false);

    dialoguePanelRef.SetActive(false);
    dialogueText = answerTextRef;
    trophyImageRef.sprite = goldSprite;
    scoreTextRef.text = "Your Score : " + DataManager.instance.userScore;
    previousScoreTextRef.transform.parent.gameObject.SetActive(false);
    resultsPanelRef.SetActive(true);

    DataManager.instance.userAnswersInfo.Phishing = "Deleted Lottery Mail";

    Invoke("HideQuestionsPanel1", 0.5f);
    StartCoroutine(TypeSentence(deleteString));
}
```

Figure 15: Phishing Backend 1

```

public void OnFillButtonClick()
{
    questionPanelAnimatorRef2.SetBool("FadeIn", false);

    dialoguePanelRef.SetActive(false);
    dialogueText = answerTextRef;
    trophyImageRef.sprite = wrongSprite;
    scoreTextRef.text = "Your Score : " + DataManager.instance.userScore;
    previousScoreTextRef.transform.parent.gameObject.SetActive(false);
    resultsPanelRef.SetActive(true);

    DataManager.instance.userAnswersInfo.Phishing = "Fill Lottery Mail";

    Invoke("HideQuestionsPanel2", 0.5f);
    StartCoroutine(TypeSentence(fillString));
}

0 references
public void OnExitButtonClick()
{
    DataManager.instance.userScore += 10;
    questionPanelAnimatorRef2.SetBool("FadeIn", false);

    dialoguePanelRef.SetActive(false);
    dialogueText = answerTextRef;
    trophyImageRef.sprite = silverSprite;
    scoreTextRef.text = "Your Score : " + DataManager.instance.userScore;
    previousScoreTextRef.transform.parent.gameObject.SetActive(false);
    resultsPanelRef.SetActive(true);

    DataManager.instance.userAnswersInfo.Phishing = "Exit Lottery Mail";

    Invoke("HideQuestionsPanel2", 0.5f);
    StartCoroutine(TypeSentence(exitString));
}

0 references
private void HideQuestionsPanel1()
{
    questionsPanelRef1.SetActive(false);
}

0 references
private void LoadSecondQuestion ()
{
    if (hasDoneShowingAnswerDialogue1)
    {
        questionsPanelRef2.SetActive(true);
        questionPanelAnimatorRef2.SetBool("FadeIn", true);
        dialogueText.text = "";
    }
}

0 references
public void OnNext()
{
    if (hasDoneShowingAnswerDialogue2)
    {
        FadeEffectManager.instance.FadeIn();
        Invoke("LoadNextScene", 0.7f);
    }
}

0 references
private void LoadNextScene()
{
    UnityEngine.SceneManagement.SceneManager.LoadScene("WaitingForBusAtOffice");
}

```

Figure 16: Phishing Backend 2

4.7.6 Malvertising

```
public void LoadQuestion ()
{
    if (hasDoneShowingDialogues)
    {
        questionsPanelRef.SetActive(true);
        questionPanelAnimatorRef.SetBool("FadeIn", true);
        dialogueText.text = "";
    }
}

0 references
public void OnNext()
{
    if (hasDoneShowingAnswerDialogue)
    {
        FadeEffectManager.instance.FadeIn();
        Invoke("LoadNextScene", 0.7f);
    }
}

0 references
public void OnFollowButtonClick()
{
    questionPanelAnimatorRef.SetBool("FadeIn", false);

    dialoguePanelRef.SetActive(false);
    dialogueText = answerTextRef;
    trophyImageRef.sprite = wrongSprite;
    scoreTextRef.text = "Your Score : " + DataManager.instance.userScore;
    previousScoreTextRef.transform.parent.gameObject.SetActive(false);
    resultsPanelRef.SetActive(true);

    DataManager.instance.userAnswersInfo.Malvertising = "Follow to Download AntiVirus";

    Invoke("HideQuestionsPanel", 0.5f);
    StartCoroutine(TypeSentence(followString));
}
```

Figure 17: Malvertising Backend

4.7.7 Lost Phone

```
private void EnableBus()
{
    playerArrivalAnimation.enabled = true;
}

1 reference
public void PlayerReachedHome ()
{
    StartCoroutine(TypeSentence(dialogueString));
    dialoguePanelRef.SetActive(true);
}
// -----

5 references
IEnumerator TypeSentence(string sentence)
{
    yield return new WaitForSeconds(1);

    dialogueText.text = "";
    foreach (char letter in sentence.ToCharArray())
    {
        dialogueText.text += letter;
        yield return new WaitForSeconds(0.01f);
    }
    if (hasDoneShowingDialogues)
        hasDoneShowingAnswerDialogue = true;
    else
        hasDoneShowingDialogues = true;
}

0 references
public void LoadQuestion()
{
    if (hasDoneShowingDialogues)
    {
        questionsPanelRef.SetActive(true);
        questionPanelAnimatorRef.SetBool("FadeIn", true);
        dialogueText.text = "";
    }
}

0 references
public void OnIgnoreButtonClick()
{
    DataManager.instance.userScore += 20;
    questionPanelAnimatorRef.SetBool("FadeIn", false);

    dialoguePanelRef.SetActive(false);
    dialogueText = answerTextRef;
    trophyImageRef.sprite = goldSprite;
    scoreTextRef.text = "Your Score : " + DataManager.instance.userScore;
    previousScoreTextRef.transform.parent.gameObject.SetActive(false);
    resultsPanelRef.SetActive(true);

    DataManager.instance.userAnswersInfo.Malvertising = "Ignore to Download AntiVirus";

    Invoke("HideQuestionsPanel", 0.5f);
    StartCoroutine(TypeSentence(ignoreString));
}

0 references
private void HideQuestionsPanel()
{
    questionsPanelRef.SetActive(false);
}

0 references
private void LoadNextScene()
{
    UnityEngine.SceneManagement.SceneManager.LoadScene("HomeScene");
}
```

Figure 18: Lost Phone Backend 1


```

public void OnReportToPoliceButtonClick()
{
    DataManager.instance.userScore += 20;
    questionPanelAnimatorRef.SetBool("FadeIn", false);

    dialoguePanelRef.SetActive(false);
    dialogueText = answerTextRef;
    trophyImageRef.sprite = bronzeSprite;
    scoreTextRef.text = "Your Score : " + DataManager.instance.userScore;
    previousScoreTextRef.transform.parent.gameObject.SetActive(false);
    resultsPanelRef.SetActive(true);

    DataManager.instance.userAnswersInfo.LostPhone = "Report To Police";

    Invoke("HideQuestionsPanel", 0.5f);
    StartCoroutine(TypeSentence(reportToPoliceString));
}

```

0 references

```

public void OnRetraceStepsButtonClick()
{
    DataManager.instance.userScore += 10;
    questionPanelAnimatorRef.SetBool("FadeIn", false);

    dialoguePanelRef.SetActive(false);
    dialogueText = answerTextRef;
    trophyImageRef.sprite = wrongSprite;
    scoreTextRef.text = "Your Score : " + DataManager.instance.userScore;
    previousScoreTextRef.transform.parent.gameObject.SetActive(false);
    resultsPanelRef.SetActive(true);

    DataManager.instance.userAnswersInfo.LostPhone = "Retrace Steps";

    Invoke("HideQuestionsPanel", 0.5f);
    StartCoroutine(TypeSentence(retraceStepsString));
}

```

Figure 19: Lost Phone Backend 2

4.7.8 Final Result

The Final Result Screen displays the cumulative score of the user and displays it on a leader board as compared to other users of the game.

AssignTrophy() is used to assign the user a trophy by comparing the user's marks with the pre-determined levels.

ShowLeaderboardData(), SortResults() and DisplayLeaderBoard() are used together to update the leader board after every instance of the game.

```
void Start()
{
    //string responseString = "{\"Items\":" + jsonString + "}";
    //DataManager.UserAnswersInfo[] allUserAnswersInfo = DataManager.UserAnswersInfo.GetJsonInfoFromString<DataManager.UserAnswersInfo>(responseString);
    //ShowLeaderboardData(allUserAnswersInfo);

    leaderboardPlayer1TextRef.text = "";
    leaderboardPlayer2TextRef.text = "";
    leaderboardPlayer3TextRef.text = "";
    infoTextRef.text = "";

    playerScoreTextRef.text = "Your Score : " + DataManager.instance.userScore;
    AssignTrophy();

    DataManager.instance.userAnswersInfo.Score = DataManager.instance.userScore;
    AnimateLoader();
    FirebaseManager.instance.FetchUserResults(this);
}

1 reference
private void AssignTrophy()
{
    int score = DataManager.instance.userScore;
    if (score >= 120)
        trophyImageRef.sprite = goldTrophyRef;
    else if (score >= 90)
        trophyImageRef.sprite = silverTrophyRef;
    else if (score >= 60)
        trophyImageRef.sprite = bronzeTrophyRef;
    else
        trophyImageRef.sprite = wrongImageRef;
}

public void ShowLeaderboardData(DataManager.UserAnswersInfo[] allUserAnswersInfo)
{
    Debug.Log("ShowLeaderboardData");
    if (allUserAnswersInfo == null)
    {
        leaderboardPlayer1TextRef.text = "You : " + DataManager.instance.userScore;

        FirebaseManager.instance.UploadUserAnswers (0, this);
    }
    else
    {
        SortResults(allUserAnswersInfo);
    }
}

5 references
public void ShowErrorMessage(string message)
{
    infoTextRef.text = message;
    loaderRef.SetActive(false);
    homeButtonRef.SetActive(true);
    CancelInvoke("AnimateLoader");
}

1 reference
public void OnSuccess()
{
    infoTextRef.text = "";
    loaderRef.SetActive(false);
    homeButtonRef.SetActive(true);
    CancelInvoke("AnimateLoader");
}

0 references
public void OnHome()
{
    UnityEngine.SceneManagement.SceneManager.LoadSceneAsync("Menu");
}

1 reference
private void AnimateLoader()
{
    loaderRef.transform.localEulerAngles = new Vector3(0, 0, loaderRef.transform.localEulerAngles.z - 45);

    Invoke("AnimateLoader", 0.1f);
}
```

Figure 20: Final Result Backend 1

```

private void SortResults(DataManager.UserAnswersInfo[] allUserAnswersInfo)
{
    Debug.Log("SortResults " + allUserAnswersInfo.Length);
    bool valueHasCurrentUser = false;
    for (int index = 0; index < allUserAnswersInfo.Length; index++)
    {
        Debug.Log("ID Compare : " + FirebaseManager.instance.firebaseUser.UserId + " - " + allUserAnswersInfo[index].UserID);
        if (FirebaseManager.instance.firebaseUser.UserId.Equals(allUserAnswersInfo[index].UserID))
        {
            Debug.Log("Has id : ");
            valueHasCurrentUser = true;
            break;
        }
    }
    Debug.Log("Done Sorting");
    if (valueHasCurrentUser)
    {
        for (int index = 0; index < allUserAnswersInfo.Length - 1; index++)
        {
            for (int nextIndex = index + 1; nextIndex < allUserAnswersInfo.Length; nextIndex++)
            {
                if (allUserAnswersInfo[index].Score < allUserAnswersInfo[nextIndex].Score)
                {
                    DataManager.UserAnswersInfo usersAns = allUserAnswersInfo[index];
                    allUserAnswersInfo[index] = allUserAnswersInfo[nextIndex];
                    allUserAnswersInfo[nextIndex] = usersAns;
                }
            }
        }
        DisplayLeaderboard(allUserAnswersInfo, false, 0);
    }
    else
    {
        Debug.Log("User is not in list " + FirebaseManager.instance.firebaseUser.UserId);
        DataManager.UserAnswersInfo[] newResults = new DataManager.UserAnswersInfo[allUserAnswersInfo.Length + 1];
        for (int index = 0; index < allUserAnswersInfo.Length; index++)
        {
            newResults[index] = allUserAnswersInfo[index];
        }
        newResults[allUserAnswersInfo.Length] = DataManager.instance.userAnswersInfo;
        for (int index = 0; index < newResults.Length - 1; index++)
        {
            for (int nextIndex = index + 1; nextIndex < newResults.Length; nextIndex++)
            {
                if (newResults[index].Score < newResults[nextIndex].Score)
                {
                    DataManager.UserAnswersInfo usersAns = newResults[index];
                    newResults[index] = newResults[nextIndex];
                    newResults[nextIndex] = usersAns;
                }
            }
        }
        DisplayLeaderboard(newResults, true, allUserAnswersInfo.Length);
    }
}

2 references
private void DisplayLeaderboard(DataManager.UserAnswersInfo[] allUserAnswersInfo, bool isToUploadAnswer, int userInfoCount)
{
    Debug.Log("DisplayLeaderboard : " + allUserAnswersInfo.Length);
    bool foundRecords = false;
    for (int index = 0; index < allUserAnswersInfo.Length; index++)
    {
        Debug.Log(index + " User : " + allUserAnswersInfo[index].Name);
        if (index == 0)
        {
            foundRecords = true;
            leaderboardPlayer1TextRef.text = allUserAnswersInfo[index].Name + " : " + allUserAnswersInfo[index].Score;
            Debug.Log("One player");
        }
        else if (index == 1)
        {
            leaderboardPlayer2TextRef.text = allUserAnswersInfo[index].Name + " : " + allUserAnswersInfo[index].Score;
            Debug.Log("Two player");
        }
        else if (index == 2)
        {
            leaderboardPlayer3TextRef.text = allUserAnswersInfo[index].Name + " : " + allUserAnswersInfo[index].Score;
            Debug.Log("Three player");
        }
        else
        {
            Debug.Log("Breaking loop");
            break;
        }
    }

    if (isToUploadAnswer)
    {
        FirebaseManager.instance.UploadUserAnswers(userInfoCount, this);
    }
    else
    {
        if (foundRecords)
        {
            Debug.Log("Found record");
            homeButtonRef.SetActive(true);
            loaderRef.SetActive(false);
            CancelInvoke("AnimateLoader");
        }
        else
        {
            Debug.Log("Error message");
            ShowErrorMessage("No Records Found");
        }
    }
}
}

```

Figure 21: Final Result Backend 2

Chapter 5: Project Screenshots



Figure 22: Login/Signup

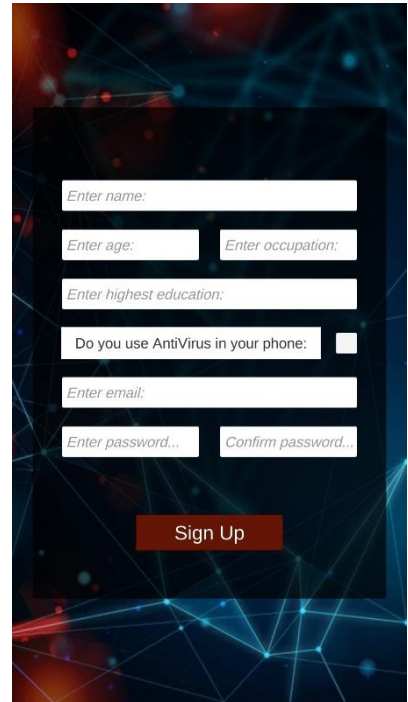


Figure 23: Sign up

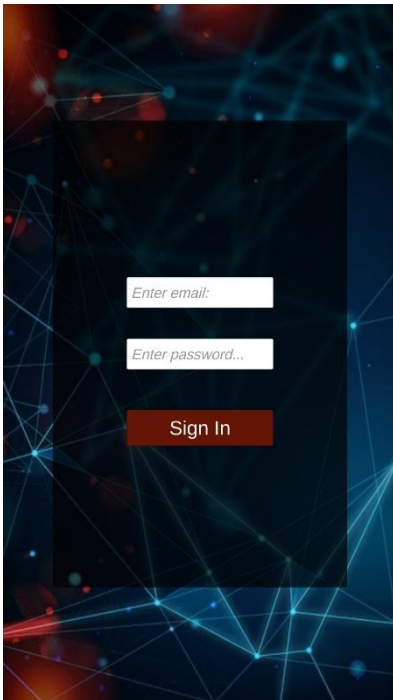


Figure 24: Sign in

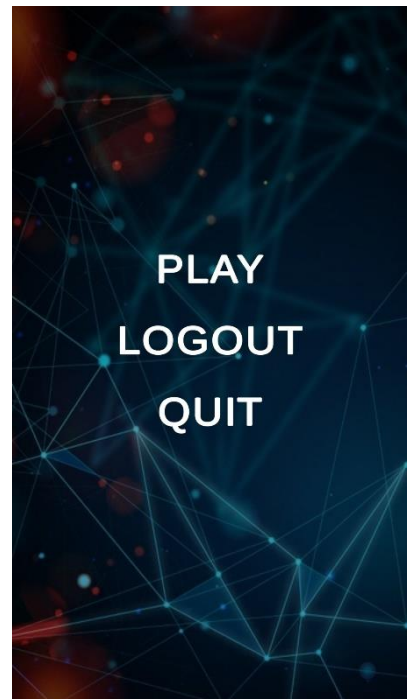


Figure 25: Main Menu

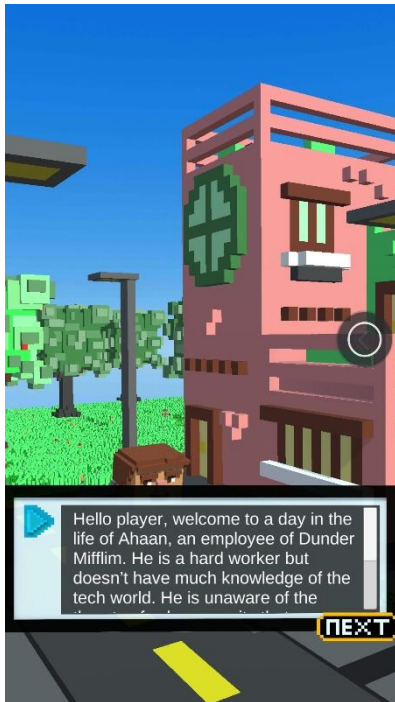


Figure 26: Introduction Scene 1

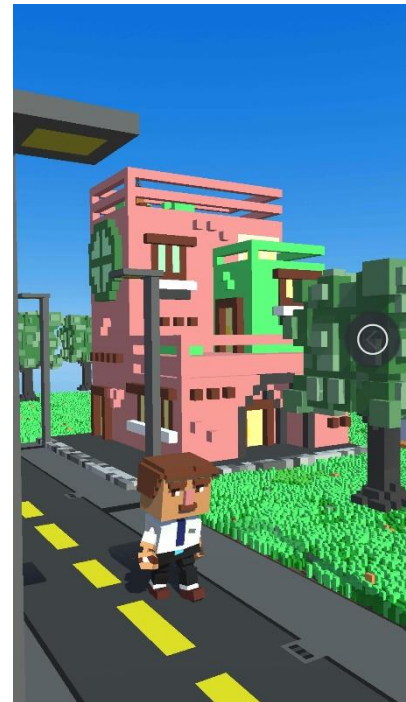


Figure 27: Introduction Scene 2



Figure 28: Network Spoofing Screen 1

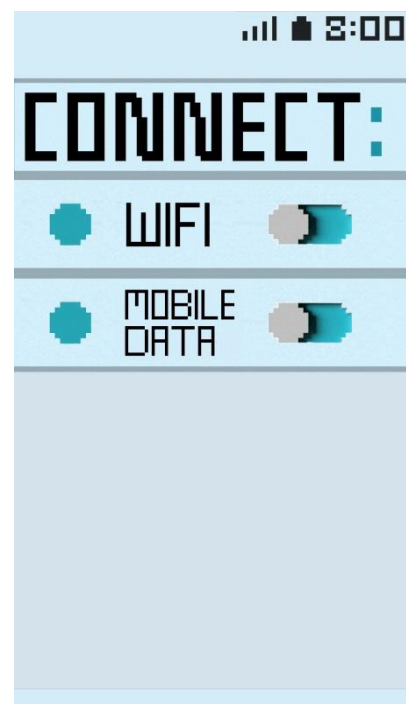


Figure 29: Network Spoofing Screen 2

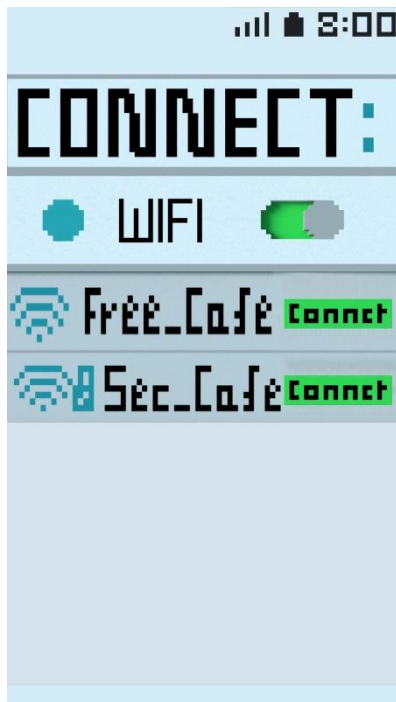


Figure 30: Network Spoofing Screen 3



Figure 31: Network Spoofing Score Screen



Figure 32: Smishing Screen 1

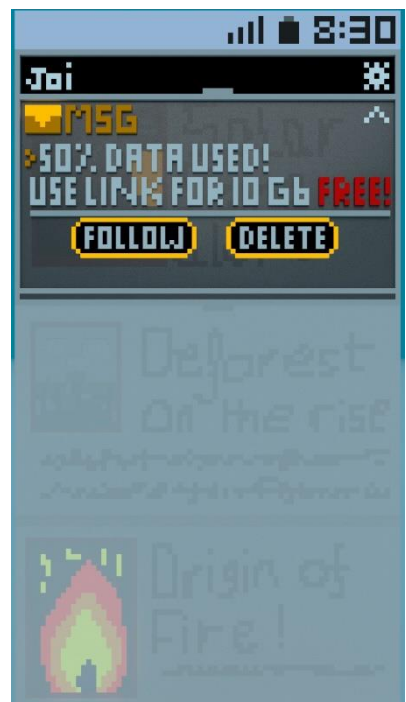


Figure 33: Smishing Screen 2

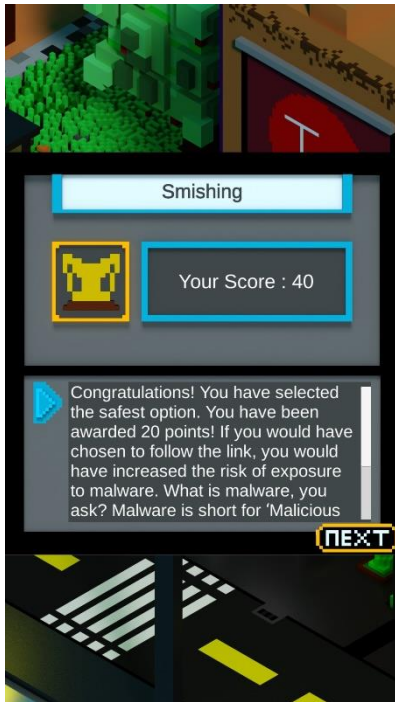


Figure 34: Smishing Score Screen



Figure 35: Malicious App Screen 1



Figure 36: Malicious App Screen 2



Figure 37: Malicious App Screen 3



Figure 38: Malicious App Score Screen

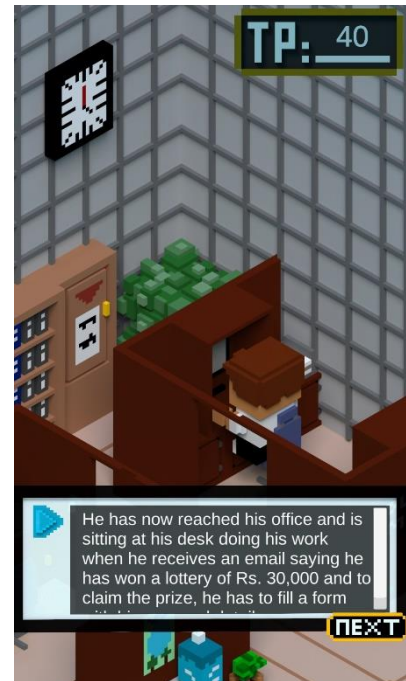


Figure 39: Phishing Screen 1



Figure 40: Phishing Screen 2



Figure 41: Phishing Screen 3



Figure 42: Phishing Score Screen

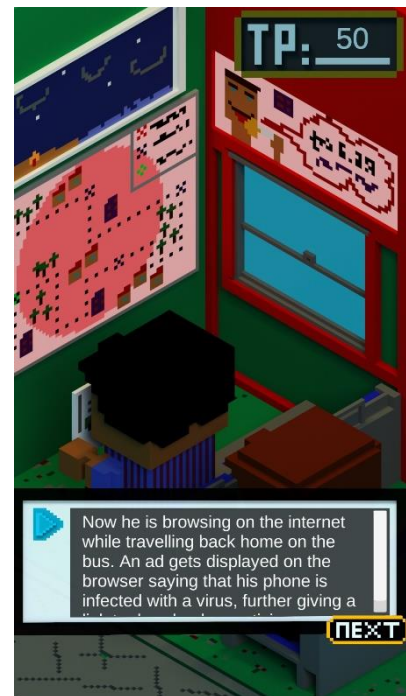


Figure 43: Malvertising Screen 1

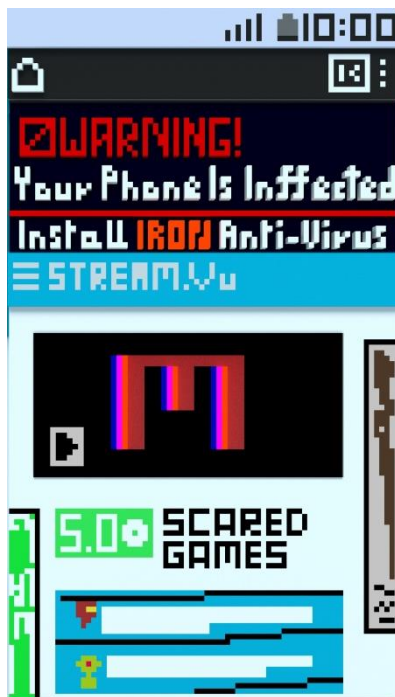


Figure 44: Malvertising Screen 2

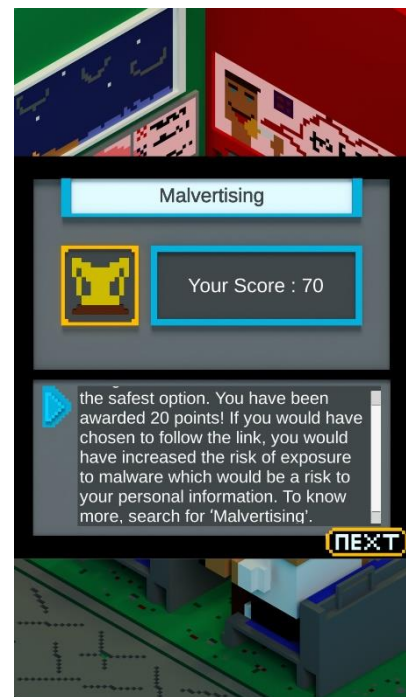


Figure 45: Malvertising Score Screen

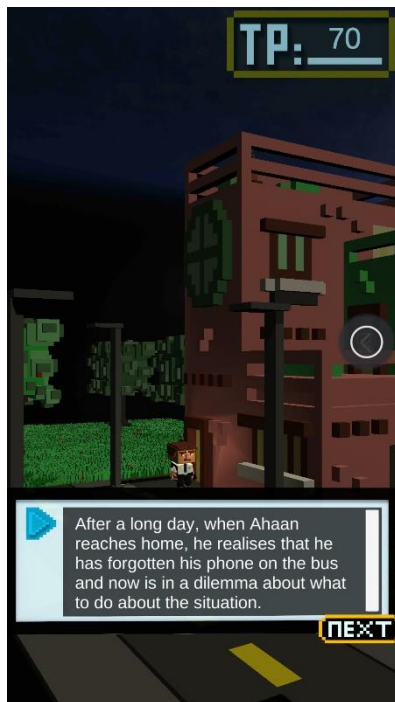


Figure 46: Phone Lost Screen 1

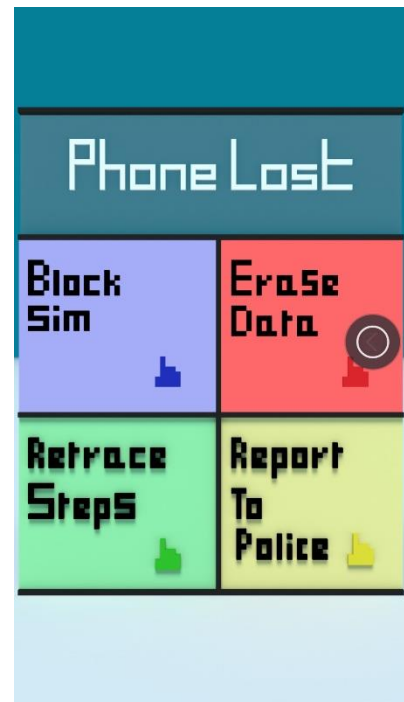


Figure 47: Phone Lost Screen 2

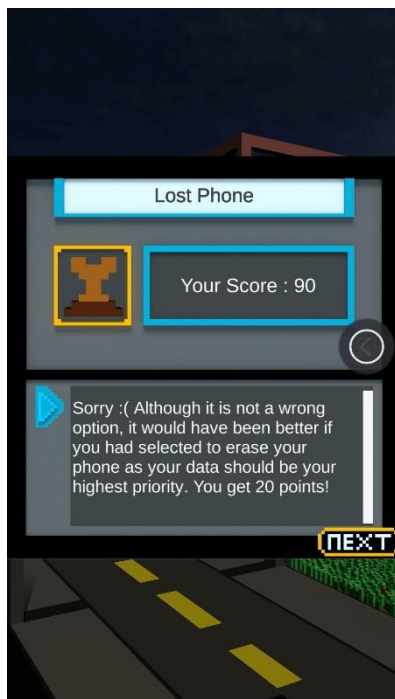


Figure 48: Phone Lost Score Screen

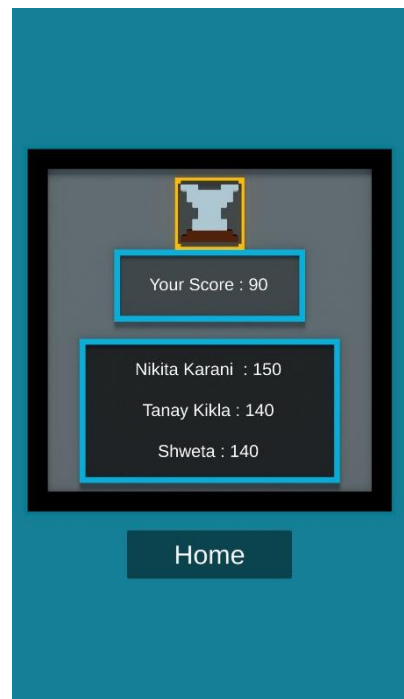


Figure 49: Leaderboard Screen

Chapter 6: Testing Report

Software testing refers to the process of validation and verification of software programs aimed at discovering and rectifying errors as well as ensuring that requirements specified at both the business and the technical level are properly implemented. The various types of testing are shown below.

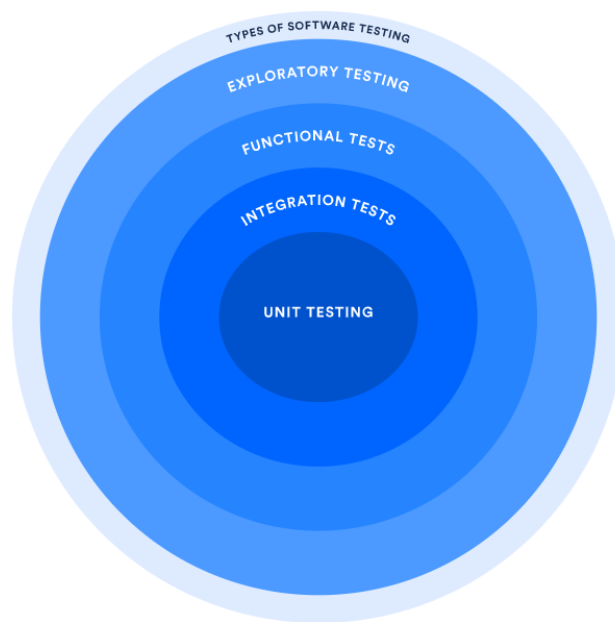


Figure 50: Types of Software Testing

6.1 Unit Testing

Unit testing refers to the individual testing of the smallest testable components of an application. It involves providing the requisite input parameters and checking if the outputs match the expected requirement.

Unit testing for the project was conducted by creation and validation of the following test cases.

User Profile/Login

Table III: Unit Testing User Profile/Login

Test case	Test Scenario	Test Steps	Inputs/Data	Expected Results	Actual Results	Pass/Fail
1	Login	- User enters username - User enters password - User clicks 'Login' button - Entered credentials are compared with stored values	Username Password	Entered username and password match with the stored values	Entered username and password match with the stored values	Pass
2	Sign Up	- User enters <ul style="list-style-type: none"> • Name • Age • Occupation • Education • If they have antivirus • Email • Password - User clicks 'Sign Up' - All the entered data gets stored in the database	Name Age Occupation Education If they have antivirus Email Password	The database must be updated with the entered details	The database is updated with the entered details	Pass
3	Home Screen	- User logs in successfully - User can see three options – i) Play ii) Logout iii) Quit	Valid login details	After successful login, user must be able to see the home screen with three options	After successful login, user is able to see the home screen with three options	Pass
4	Pressing one of the three options on the home screen	- User clicks either 'Play', 'Logout' or 'Quit'	Valid login details	- After clicking 'Play', the game should start - After clicking 'Logout', the user should get logged out.	- After clicking 'Play', the game starts - After clicking 'Logout', the user gets logged out. - After clicking	Pass

				- After clicking 'Quit', the application should close	'Quit', the application closes	
--	--	--	--	---	--------------------------------	--

Scenarios

Table IV: Unit Testing Scenarios

Test case	Test Scenario	Test Steps	Inputs/Data	Expected Results	Actual Results	Pass/Fail
1	Network Spoofing	<ul style="list-style-type: none"> - User clicks 'Play' on the home screen - User sees first scenario with narration - User clicks either 'WiFi' or 'Mobile Data' 	Answer chosen by user	<ul style="list-style-type: none"> - The answer chosen by the user should get stored in the database in the correct field - According to the answer chosen, user should get points 	<ul style="list-style-type: none"> - The answer chosen by the user gets stored in the database in the correct field - User gets points according to answer chosen 	Pass
2	Smishing	<ul style="list-style-type: none"> - User sees second scenario with narration - User clicks either 'Follow' or 'Delete' 	Answer chosen by user	<ul style="list-style-type: none"> - The answer chosen by the user should get stored in the database in the correct field - According to the answer chosen, user should get points 	<ul style="list-style-type: none"> - The answer chosen by the user gets stored in the database in the correct field - User gets points according to answer chosen 	Pass
3	Malicious Application	<ul style="list-style-type: none"> - User sees second scenario with narration - User clicks either 'Buy' or 'Download Free APK' 	Answer chosen by user	<ul style="list-style-type: none"> - The answer chosen by the user should get stored in the database in the correct field - According to the answer chosen, user should get points 	<ul style="list-style-type: none"> - The answer chosen by the user gets stored in the database in the correct field - User gets points according to answer chosen 	Pass

4	Phishing	<ul style="list-style-type: none"> - User sees second scenario with narration - User clicks either 'Fill' or 'Delete' 	Answer chosen by user	<ul style="list-style-type: none"> - The answer chosen by the user should get stored in the database in the correct field - According to the answer chosen, user should get points 	<ul style="list-style-type: none"> - The answer chosen by the user gets stored in the database in the correct field - User gets points according to answer chosen 	Pass
5	Malvertising	<ul style="list-style-type: none"> - User sees second scenario with narration - User clicks either 'Download Iron Antivirus' or 'Continue with previous work' 	Answer chosen by user	<ul style="list-style-type: none"> - The answer chosen by the user should get stored in the database in the correct field - According to the answer chosen, user should get points 	<ul style="list-style-type: none"> - The answer chosen by the user gets stored in the database in the correct field - User gets points according to answer chosen 	Pass
6	Lost Phone	<ul style="list-style-type: none"> - User sees second scenario with narration - User clicks either 'Block Sim', 'Erase Phone', 'Report to Police' or 'Retrace Steps' 	Answer chosen by user	<ul style="list-style-type: none"> - The answer chosen by the user should get stored in the database in the correct field - According to the answer chosen, user should get points 	<ul style="list-style-type: none"> - The answer chosen by the user gets stored in the database in the correct field - User gets points according to answer chosen 	Pass
7	Game Completion	<ul style="list-style-type: none"> - User finishes the game - User sees the leaderboard - User clicks 'Home' 	No input required	<ul style="list-style-type: none"> - Leaderboard with three highest scores should be visible - On clicking 'Home', user should be redirected to home screen 	<ul style="list-style-type: none"> - Leaderboard with three highest scores is visible - On clicking 'Home', user gets redirected to home screen 	Pass

6.2 System Testing

System testing refers to a complete testing of the entire software application once all the modules have been integrated into it and ensuring compliance with the specified requirements. System testing for the mobile application was performed once all modules were completed and integrated.

At this stage, system testing was conducted by testing the flow of the mobile application. The application was run from the registration page. Once the user was able to sign up, each scenario was tested in the following order:

1. Network Spoofing
2. Smishing
3. Malicious Application
4. Phishing
5. Malvertising
6. Lost Phone

At each stage, it was checked if the user response was stored in the database in the appropriate field.

The system was tested based on usability and system response time.

Chapter 7: Results and Discussions

We tested the game by giving it out to 50 individuals of different age groups, education backgrounds and occupations to play in order to analyze the data generated by them. We sent the game via WhatsApp to a few people requesting them to play the game. Since this game isn't bilingual yet, for the people who cannot read and understand English like security guards, house helpers etc. we helped them understand the scenarios and the questions posed in the game.

We gave the game to individuals of different age groups, the lowest being 10 and the highest being about 53. The occupations of these individuals ranged from being in the service sector like house helpers, to being students to business owners. Their education levels were between those who did not pass 10th grade to a graduate degree to a Phd.

The results are as follows:

7.1 The Highest and Lowest Number of Choices

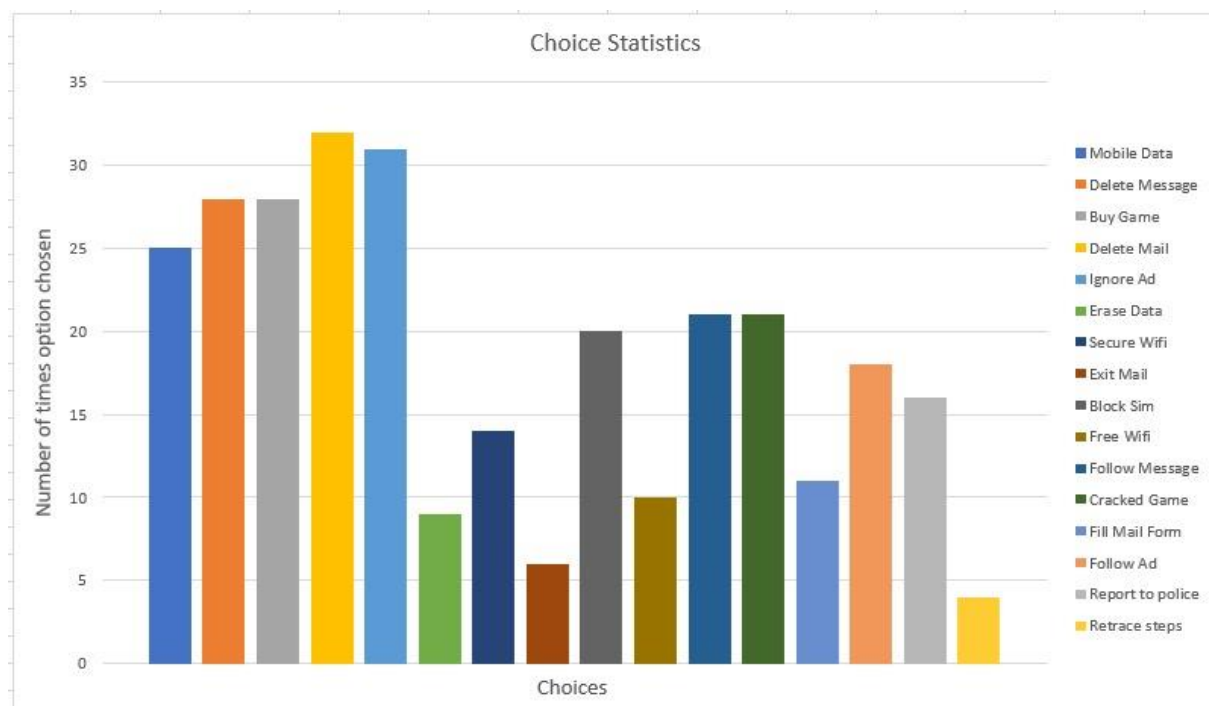


Figure 51: Choices Statistics

The bar graph above shows the choices and the number of times each choice was chosen by the users. It can be clearly seen that the choice chosen the most times was the 'Delete Mail' choice, which suggests that most users in the test are aware of phishing scams happening around

the world. They know that any unverified links are not to be opened without verifying them first.

It can also be seen that the choice chosen the least number of times, i.e. about 4 times, is 'Retrace Steps'. This data indicates that individuals are alert when they lose their phones and know that retracing steps is not the most appropriate option. But at the same time, it is also observed that not a lot of people have chosen to erase their data, the number being between 5 and 10. This shows that although people are aware that losing phone is a major threat, they aren't aware that remotely erasing the data on their phones is the first thing they should do.

7.2 Lost Phone Statistics

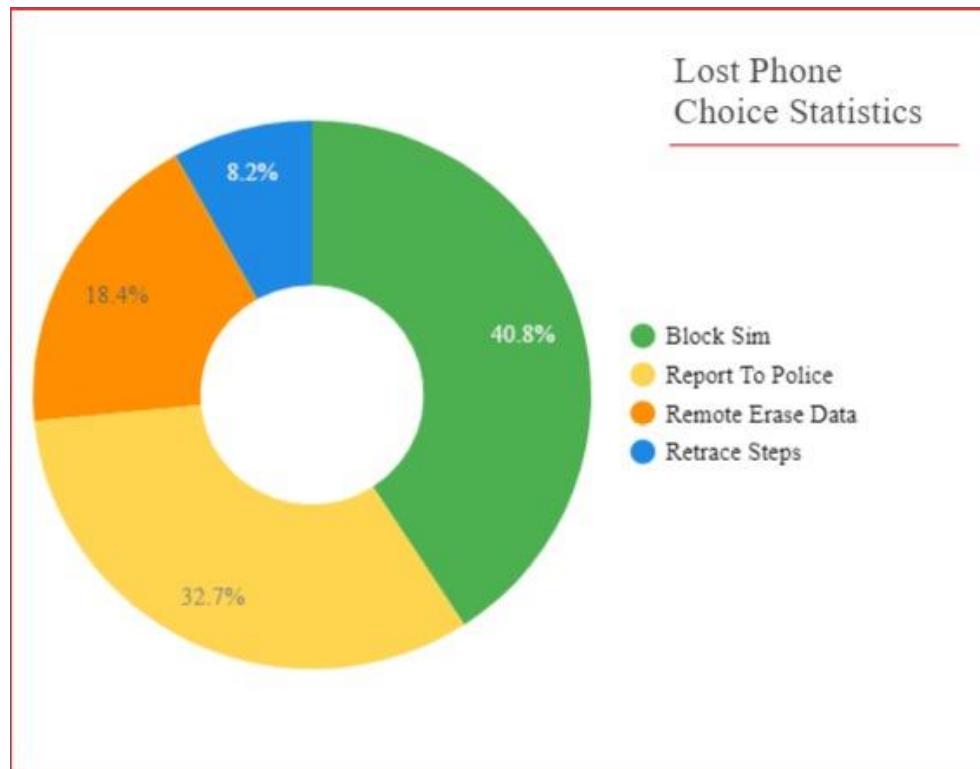


Figure 52: Lost Phone Statistics

The pie chart above indicates the Lost Phone Choice Statistics. It can be observed that most people selected the 'Block Sim' option which tells us about their uninformed nature. Though blocking sim is the right thing to do but the foremost thing to be done is remotely erase data. The next most chosen option is 'Report to Police' followed by 'Remote Erase Data' and 'Retrace Steps'. Most of the people are wise enough to not retrace steps in case such a scenario occurs as it can be seen from the statistics.

7.3 User Age v Score

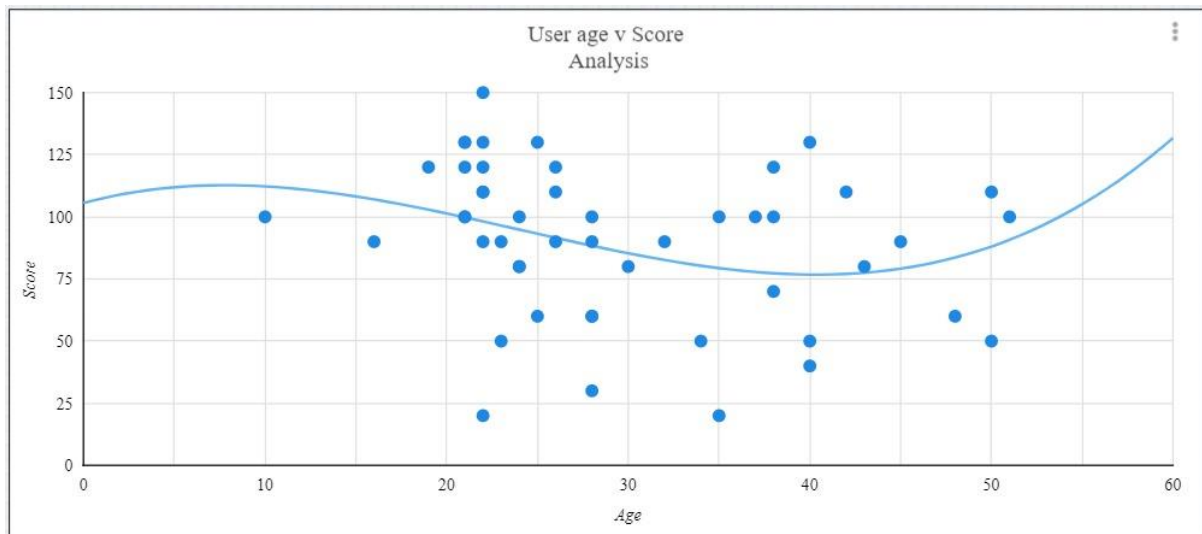


Figure 53: User Age v Score

It is evident that the most points are scored by people between the ages of 20 and 30 years. The highest score is 150 points which is scored by only one user of the 50 chosen users, indicating that there is still a need to educate the mass about the threats they face.

7.4 Education v Score

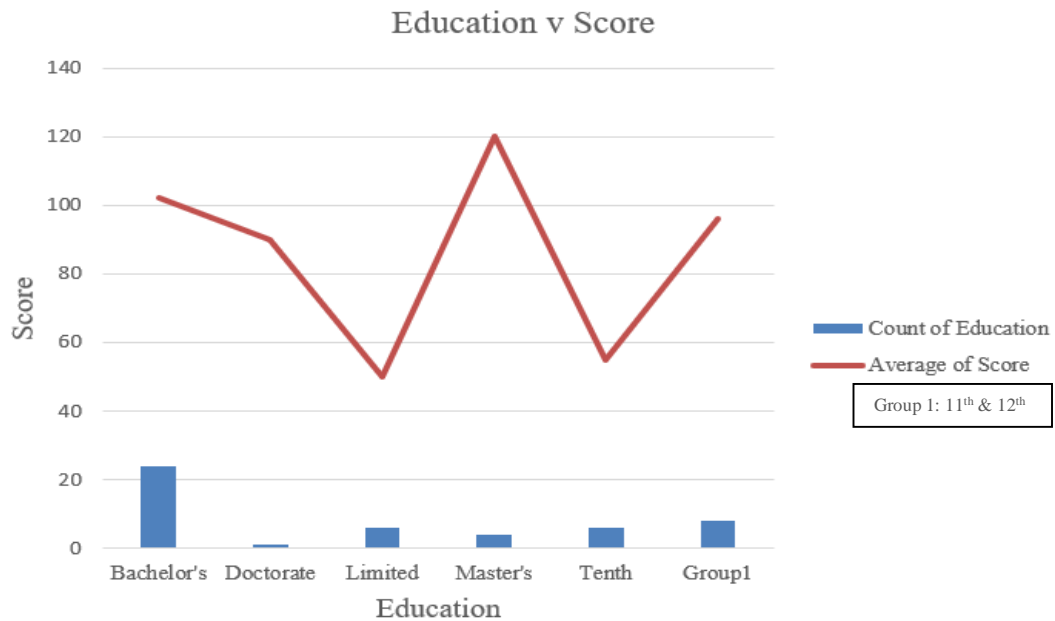


Figure 54: Education v Score

In the graph given below it is evident that the level of education an individual possesses does affect their awareness of cybersecurity threats. It is notable to see the jump in score when moving from just 10th pass to individuals who have attended higher school education. As expected, it is also clear that the population with limited formal education are the most at risk of being exploited.

7.5 Assumptions

- The player plays the game once and is unaware of the questions and answers.
- The player has an active internet connection on their mobile phones
- Assuming the player owns and regularly uses a smartphone and understand basic smartphone terminologies.

7.6 Future Work

- The game was developed in only one language restricting the amount of use. Making the game bilingual and multilingual is a future development plan to expand the reach and scope of the game. Regional languages like Hindi, Marathi, etc, would be included in future updates.
- Only 6 of the threats are covered in the game as of now. The game only focuses on 6 major threats faced by users but these are not all of the threats. In future updates more of such common threats faced by users will be covered.
- Push Notifications and reminders. Push Notifications and Reminders are an essential part of keeping the user up to date about the threats around them and at the same time making them aware of their personal device settings that need to be changed and maintained in order to be protected from various threats.

Chapter 8: Conclusions

8.1 Conclusion

The aim of this project was to develop a game that can be used as a medium to share information about cybersecurity and as a means to test the user's pre-existing knowledge in the field. An additional objective was to test the effectiveness of games as a medium for the previously mentioned purposes. To develop a game that would be more interactive and easy to use than those that came before it, multiple research papers were studied, and similar existing games analysed.

The data collected through the game was used to analyse users of different backgrounds on their knowledge on cybersecurity and the threats they face on a daily basis. The data generated can help identify which cyber-threats the people of India are more vulnerable to and thus help improve awareness programs.

8.2 Future Avenues

At present, the game has one environment and storyline i.e. a character going from his home to his office and back to his home again and facing cybersecurity threats on the way. It also have only one male character. In addition to this, there are six scenarios that a user has to go through, and these are the cyber-threats covered by the game.

In the near future, the gameplay could be refined and made even more engaging. This can be done by introducing interesting ways to interact with the game world and by embedding videos in the gameplay so the user can retain more information. An increase in the number of scenarios and variety of situations will help researchers better analyze the understanding of cyber security the mass population possess. Other improvements can additional environments and characters to the game, like a character going to college and coming back home.

REFERENCES

- [1] <https://www.fbi.gov/news/stories/ic3-releases-2018-internet-crime-report-042219>
- [2] <https://www.youthkiawaaz.com/2016/06/cyber-crime-rate-in-india/>
- [3] <https://www.thehindubusinessline.com/info-tech/mobile-cyber-threats-on-the-rise-but-awareness-still-low-experts/article27265655.ece>
- [4] <https://www.techinsurance.com/blog/cyber-liability/9-data-breach-mistakes/>
- [5] <https://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html>
- [6] Abawajy, Jemal & Kim, Tai-Hoon. (2010). Performance Analysis of Cyber Security
- [7] Awareness Delivery Methods. 122. 142-148. 10.1007/978-3-642-17610-4_16.
<https://keeptraditionsecure.tamu.edu/>
- [8] Benjamin D. Cone, Cynthia E. Irvine, Michael F. Thompson, Thuy D. Nguyen, A video game for cyber security training and awareness, Computers & Security, Volume 26, Issue 1, 2007, Pages 63-72, ISSN 0167-4048.
- [9] <https://www.pwc.co.uk/issues/cyber-security-data-privacy/services/game-of-threats.html>
- [10] https://cyberbowl.its.fsu.edu/game/#gf_8
- [11] Labuschagne, W.A. & Veerasamy, Namosha & Burke, Ivan & Eloff, Mm. (2011). Design of cyber security awareness game utilizing a social media framework. 2011 Information Security for SouthAfrica - Proceedings of the ISSA 2011 Conference.1-9. 10.1109/ISSA.2011.6027538
- [12] <https://www.isdecisions.com/user-security-awareness-game/>
- [13] <https://www.pbs.org/wgbh/nova/labs/lab/cyber/>
- [14] <https://iatraining.disa.mil/eta/cyber-awareness-challenge/index.html>
- [15] <http://targetedattacks.trendmicro.com/cyoa/en/>
- [16] F. Alotaibi, S. Furnell, I. Stengel and M. Papadaki, "Enhancing cyber security awareness with mobile games," 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST), Cambridge, 2017, pp. 129-134.
- [17] F. Giannakas, G. Kambourakis and S. Gritzalis, "CyberAware: A mobile game-based app for cybersecurity education and awareness," 2015 International Conference on Interactive Mobile Communication Technologies and Learning (IMCL), Thessaloniki, 2015, pp. 54-58.
- [18] M. Salazar, J. Gaviria, C. Laorden and P. G. Bringas, "Enhancing cybersecurity learning through an augmented reality-based serious game," 2013 IEEE Global Engineering Education Conference (EDUCON), Berlin, 2013, pp. 602-607.

- [19] <https://edtechdigest.blog/2018/02/22/the-heart-of-educational-game-design/>
- [20] <https://blog.netwrix.com/2018/05/15/top-10-most-common-types-of-cyber-attacks/>
- [21] <https://www.stardewvalley.net/>
- [22] <https://us.diablo3.com/en/>
- [23] <https://fallout.bethesda.net/en/games/fallout-tactics>