Strong Password and 2FA Awareness Mobile Game

Graduation Project Report

NAME	RAGHAD SAAD ALSAADI	FAREEDA MAHMOOD DAGHESTANI	Raoom Sami Mohammed	Kareemah Saleh Alharbi	Roua Abdullah Alnefai
ID	2111169	2116400	2110307	2110446	2111567

Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

0.1 Introduction

Despite advancements in cybersecurity measures, the age-old reliance on username-password combinations remains prevalent worldwide. Even two-factor authentication (2FA), often considered a robust security measure, frequently includes passwords as one of its components [1]. This reliance on passwords creates significant vulnerabilities, as studies show that many users opt for weak, easily memorable passwords due to a lack of awareness about creating stronger alternatives. This behavior dramatically increases the risk of cyberattacks, making it easier for malicious actors to breach accounts and access sensitive information [1].

The consequences of weak password practices and insufficient authentication methods have been stark. High-profile incidents, such as the 2017 Equifax data breach, where sensitive information of approximately 147 million individuals was compromised, highlight the dangers of inadequate password security. Similarly, the 2019 Capital One breach, which exposed the personal data of over 100 million customers, was traced back to a misconfigured firewall and the failure to implement robust security measures, including strong passwords and 2FA [2].

Globally, the statistics are alarming. Research indicates that many data breaches stem from weak or stolen passwords. Reports reveal that nearly 80% of hacking-related breaches involve compromised passwords, underscoring the urgent need for better practices [2]. In the corporate environment, attackers often target employees to steal passwords and gain unauthorized access to organizational data, leading to devastating financial and reputational damage [1].

Furthermore, organizational policies that enforce password expiration can inadvertently lead to insecure practices, such as writing down passwords or creating new passwords that are only marginally different from the old ones. For example, simply adding numbers or special characters to the end of a weak password does not significantly enhance security. Such practices contribute to a culture of complacency regarding password strength [2].

While password meters can assist users in creating stronger passwords, many provide only basic feedback and often rate passwords inconsistently. This lack of effective feedback highlights the need for greater awareness and education on password creation. Advanced training methods, particularly interactive games, are more effective in engaging users and imparting essential cybersecurity knowledge [1].

Given the global nature of these issues, it is imperative to foster a culture of cybersecurity awareness that extends beyond technical circles. By emphasizing the importance of strong passwords and 2FA, we can mitigate the risks posed by cyber threats. This report discusses the development of an innovative mobile video game designed to enhance cybersecurity awareness among users, particularly focusing on practical measures like creating strong passwords and implementing 2FA. Through engaging scenarios and interactive learning, the game aims to equip users with the tools they need to protect their digital identities and assets in an increasingly dangerous cyber landscape [3].

0.1.1 Problem Definition

In Saudi Arabia, the reliance on weak passwords and insufficient authentication methods poses a significant cybersecurity threat. As the Kingdom expands its digital services and embraces technology, cybersecurity becomes a crucial pillar for achieving Vision 2030 [4]. Many users often choose easily memorable passwords, which creates vulnerabilities that cybercriminals readily exploit. While two-factor authentication (2FA) is implemented to enhance security, it frequently still relies on passwords, making it susceptible to breaches. High-profile security incidents highlight the urgent need for stronger security practices, as a considerable proportion of data breaches in the region are linked to compromised passwords. Therefore, enhancing awareness and education about password security is essential to mitigate these risks and protect sensitive information.

0.1.2 Aims and Objectives

This project aims to develop a mobile video game specifically designed to educate Saudi Arabia's less tech-savvy population on essential aspects of cybersecurity. [5] Through interactive and engaging gameplay, the game will address key cybersecurity challenges, such as preventing unauthorized access to accounts, understanding privacy settings, and securing personal accounts using strong passwords and two-factor authentication (2FA). The ultimate goal is to foster a proactive mindset in users, encouraging them to adopt secure online practices as second nature.

- Create immersive gameplay that allows users to play real-life scenarios.
- Incorporate established cybersecurity frameworks such as NIST (National Institute of Standards and Technology), which outlines core cybersecurity practices, and ZTA (Zero Trust Architecture), which enforces a "never trust, always verify" model, to ensure the educational content is grounded in industry-leading best practices.
- Make cybersecurity education accessible and engaging for the target audience.
- Ensure the cybersecurity education is relevant to users' daily lives.
- Empower the target audience by reducing their vulnerability to cyber threats and enhancing their overall digital safety.

0.1.3 Proposed Solution

Our Proposed Solution focuses on developing a game that engages players by presenting realistic scenarios illustrating the severe consequences of poor cybersecurity practices, such as using weak passwords. By incorporating Protection Motivation Theory (PMT) as a framework, the game emphasizes the importance of secure practices by enhancing players' perceived severity of threats, vulnerability to risks, and the efficacy of protective measures. Through interactive elements where players must create strong passwords and enable two-factor authentication (2FA) to protect their virtual accounts, we will enhance their coping appraisal through practical experience. This hands-on learning will boost players' self-efficacy, empowering them to apply these security measures in their real lives. Additionally, the game will include rewards for successful implementation of these practices, reinforcing positive behavior changes and motivating players to share their newfound knowledge with peers. By aligning the game design with core principles of cybersecurity

0.1.4 Novelty and Contribution

Novelty: Our game is considered new and unique, as it is uncommon in our community to have games that combine entertainment and education. It is essential for individuals to understand security and authentication, but we do not see significant awareness in this area among Saudis, particularly teenagers. Therefore, it is important to create an app that increases their awareness in an engaging way and enhances their understanding through games. The game is designed based on the principles of Protection Motivation Theory (PMT), which emphasizes motivating individuals to adopt protective behaviors by highlighting the severity and vulnerability of cybersecurity threats and the effectiveness of protective actions.

Contribution: In this game, we aim to achieve one of the main goals of the National Cybersecurity Authority, which is to raise awareness among teenagers and young people about the importance of creating strong passwords and implementing two-factor authentication. By aligning with PMT, the game educates users on the consequences of poor cybersecurity practices while empowering them to take effective preventive measures. Our efforts focus on improving security, reducing the chances of attacks, and building a secure online environment to combat the significantly increasing cyberattacks in recent times.

0.1.5 Project Plan

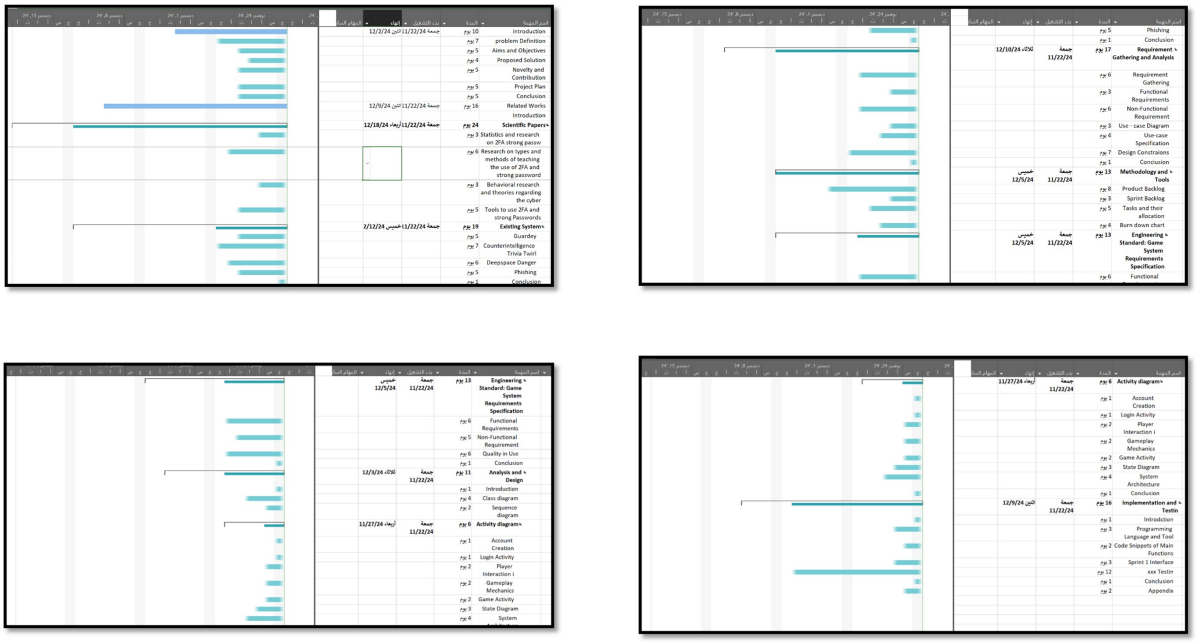


Figure 1: Project Gantt Chart for the Project Plan

0.1.6 Conclusion

Given the fast-expanding cybersecurity landscape and the rising sophistication of cyber threats, raising public understanding of digital security has never been more important. This project fills a large need in cybersecurity education, focusing on less tech-savvy persons in Saudi Arabia with a unique method that mixes entertainment and instruction. The proposed approach, which uses an interactive mobile video game, turns essential cybersecurity concepts like strong password creation and two-factor authentication (2FA) into practical, easily consumable solutions. The game creates a compelling, immersive experience by incorporating models such as Experiential Gaming and Conceptual-Procedural learning concepts, empowering users to protect their personal information.

CHAPTER 2: RELATED WORKS

0.2 Related Works

Understanding the current landscape of cybersecurity education, particularly in the context of gamification and interactive learning tools, is critical for determining the value and uniqueness of our proposed solution. This section reviews existing studies, instructional resources, and approaches for promoting cybersecurity awareness and training. By reviewing earlier research and related systems, we want to uncover both successful tactics and potential gaps in previous efforts, allowing our solution to build on these foundations while addressing unmet requirements in the Saudi Arabian context.

0.2.1 Scientific Papers

0.2.1.1 Secure Account and 2FA

Research emphasizes the importance of balancing usability and security [6] [7], particularly in areas like mHealth apps, smart devices, and wireless communication technologies. Advanced authentication approaches, such as a two-factor scheme using the modified Schnorr signature algorithm, provide hardware-free, robust solutions for access control, while multi-layered frameworks for cloud security leverage geolocation [8], browser verification, and AES encryption to prevent unauthorized access. Studies based on the Protection Motivation Theory (PMT) reveal that tailored messaging can improve security behaviors related to authentication and confidentiality [9]. As online activities like education, e-commerce, and financial transactions grow [10], promoting strong, hard-to-guess passwords and enhancing user authentication are critical to mitigating password-related attacks [11]and ensuring secure digital interactions.

0.2.1.2 Types and Methods of Teaching

Several studies have explored methods to improve user adoption of security features. Research on Duo 2FA among college students found that videos with human speakers were more effective than animated videos or no videos [12]. For the Internet of Things (IoT), the IoT-GP two-factor authentication scheme, combining alphanumeric and graphical passwords, improved security and usability [13]. Another study with 285 participants compared text- and video-based approaches to promote password manager adoption, finding videos more effective in influencing behavior [14]. Additionally, a study on messaging techniques for Duo 2FA [15] adoption tested five communication styles, with authoritarian and utilitarian approaches showing the most promise.

0.2.1.3 Behavioral Theories and models

The rise in cyber-attacks, particularly in Saudi Arabia [16], highlights the urgent need for a robust cybersecurity legal framework. While technical solutions are essential, [17]the human element plays a critical role in detecting and responding to security incidents. Low levels of cybersecurity awareness and insufficient understanding of user behavior emphasize [18]the importance of both legal and educational improvements to enhance overall security. Several behavioral theories provide valuable insights into this subject. The Theory of Planned Behavior (TPB), developed by Icek Ajzen, [19] predicts human behavior by emphasizing the role of three factors: attitude toward the behavior, subjective norms, and perceived behavioral control, all of which shape behavioral intention. Protection Motivation Theory (PMT) explains [20]how persuasive communication and cognitive mechanisms, such as fear appeals, influence protective behavior, particularly in response to perceived threats. Similarly, the Technology Threat Avoidance Theory (TTAT) asserts [21]that individuals' perceptions of their susceptibility to and the severity of technology threats drive their motivation and behavior to avoid these risks. These theories collectively offer a deeper understanding of user behavior, enabling more effective strategies to mitigate cybersecurity risks.

0.2.2 Existing System

0.2.2.1 Guardey

Protecting your network from cyber threats shouldn't feel like constantly pulling weeds from a garden. To keep your business safe, you need to know what you're doing, but mastering cybersecurity is too complicated and expensive to take on by yourself. With
Guardey, it is best for: IT/security agencies,Remoteteams, and SaaS.

Features

- GDPR-compliant
- Guardey is a plug-and-play cybersecurity solution that offers a business VPN, monitors data risks, and provides cyber awareness training.
- With Guardey, the organization can access a secure VPN tunnel to keep your data anonymous and private. This means your entire team can safely use workflow apps and tools anywhere in the world.

Business VPN Guardey provides the employees with access to a reliable business VPN. Its advanced monitoring capabilities allow Guardey to quickly identify virus and malware threats. It keeps your guard up with traffic insights and automatically scans for potential threats every 15 minutes, ensuring constant security and up-to-date protection. If suspicious activity is detected, you'll receive a notification about the security issue, which can be sent directly to your IT team.

Real-time Threat Monitoring Receive alerts about potential cybersecurity threats and resolve them immediately. No matter your industry, this tool gives you everything needed to protect your business and reputation. Since Guardey never logs any user activity, you can rest easy knowing your business data remains 100% confidential.

0.2.2.2 Counterintelligence Trivia Twirl

In this trivia twirl, you get to spin the wheel like way back in the day. Once you've landed on a category, you get a set of multiple-choice questions about it. The questions are of high quality, but the categories don't cover all the modern cyber threats. Also, the entire game can be played within 20 minutes — which doesn't make this a solution for the long term, but more like a nice way to freshen up your knowledge once a year. Aside from being able to spin the wheel — which is a nice touch — there are no significant gamification elements to engage users.

0.2.2.3 Deepspace Danger

The following cyber security awareness game for employees was created by Infosec: Deepspace Danger. The game takes place in outer space and is introduced by long animated videos. The animations are impressive but a little long-winded too. While your spacecraft is hit by a meteor, your colleague needs to leave to repair the hole that it made to prevent any further oxygen leakage. You are left alone to look after 'Pat', the computer that contains personal data for every being in the solar system. After every bit of video, you get a multiple-choice

question. It's an interesting way to learn and well put together, but it makes for a somewhat passive learning experience after a while. The outer space theme is creative, but it doesn't relate directly to real-life organizations. So we're not sure how effective it is to create lasting behavior change.
Infosec IQ training content library offers industry- and role-based training modules that personalize and contextualize education. Training modules prepare your workforce to defend against the cyber threats they're most likely to face. Training is mapped to one of nine core security behaviors outlined in the **NIST** security awareness and training guidelines.

0.2.2.4 CyberIsland-poste

Phishing is the practice of sending fraudulent communications that appear to come from a legitimate and reputable source, usually through email and text messaging. The attacker's goal is to steal money, gain access to sensitive data and login information, or install malware on the victim's device. Phishing is a dangerous, damaging, and increasingly common type of cyberattack.
To defend against phishing attacks leveraging cloud infrastructure, organizations must adopt a multi-layered approach to cyber security:
Cloud Security Posture Management (CSPM): Implement CSPM solutions to monitor and enforce security best practices across cloud environments. CSPM tools can detect misconfigurations, unauthorized access, and suspicious activity, helping organizations maintain a secure cloud posture. **User awareness training**: Educate employees about the risks associated with cloud-based phishing attacks and guide how to recognize and report suspicious emails or websites. Regular security awareness training can empower employees to remain vigilant and skeptical of unsolicited communications.**Email security gateways**: Deploy advanced email security gateways capable of detecting and blocking phishing emails originating from cloud services. These gateways leverage machine learning algorithms and threat intelligence to identify malicious content and prevent it from reaching end-users.**Endpoint protection**: Implement endpoint protection solutions equipped with cloud-based threat detection capabilities. These solutions can detect and block malicious activities initiated from compromised endpoints.

(a) Guardey

(b) Counterintelligence Trivia Twirl

(c) Deepspace Danger

(d) Cyberisland Poster

Figure 2: Existing Systems

0.2.3 Comparison of Existing Systems

Property	Guardey	Counterintelligence Trivia Twirl	Deepspace Danger	CyberIsland-poste	Haseen
Cross-Platform Gaming	✓	✓	✓	✓	✓
2FA Integration	×	×	×	×	✓
Password Manager Guidance	×	×	×	×	✓
Backup Feature	×	×	×	×	✓
Simulations	✓	✓	✓	✓	✓
Inclusivity	✓	×	×	×	✓
Following Theory	×	×	✓	✓	✓
Following Standards	✓	×	×	✓	✓

Table 1: Comparison of Existing Systems

Our game stands out from existing systems by offering a **holistic and interactive cybersecurity experience** that goes beyond simple concepts. Unlike other games, our game integrates multiple layers of cybersecurity education, such as **password creation, two-factor authentication (2FA), threat identification, and real-time defense simulations**, all within a progressive and engaging game map. Each stage builds on the previous one, allowing players to not only learn about protecting digital assets but also apply these skills against increasingly sophisticated threats.

What sets our game apart is its focus on **practical application and motivation**, driven by **Protection Motivation Theory (PMT)**. This ensures players understand the importance of cybersecurity while staying engaged through dynamic challenges, rewarding gameplay, and a narrative that connects learning with action. Additionally, our game emphasizes **inclusive design** and **standards adherence**, making it accessible and aligned with real-world cybersecurity practices, unlike other systems that may overlook these aspects.

0.2.4 Conclusion

The investigation into cybersecurity education through gamification reveals important opportunities and challenges, particularly in Saudi Arabia, where significant gaps in password security awareness persist. While applications like Neal Fun's Password Game and Password Guardian demonstrate effective user engagement, they fall short in areas such as two-factor authentication and comprehensive password management. Our proposed solution addresses these shortcomings by adapting successful strategies from existing systems to the local cultural context, emphasizing interactive content that promotes strong passwords and secure authentication practices. This approach aims to enhance cybersecurity understanding and foster a safer digital environment in the region.

CHAPTER 3: Requirement Gathering and Analysis

0.3 Requirement Gathering and Analysis

0.3.0.1 Introduction

The process of gathering requirements is one of the most important basic stages in the software development life cycle, as it plays a pivotal role in ensuring that the system is compatible with user needs and achieves project objectives. This chapter aims to clarify the importance of gathering requirements systematically and accurately, and how this affects the success of software projects. In addition to reviewing requirements gathering techniques and methods, the focus will be on analyzing practical examples to illustrate how to classify and document requirements effectively.

0.3.0.2 Requirements Gathering - Survey Results

The survey results provide insights into participants' cybersecurity practices and awareness. They reflect varying levels of understanding and application of essential security measures, such as using strong passwords, enabling two-factor authentication (2FA), and recognizing recommended security protocols. The data highlights key areas where users are informed, as well as gaps that can be addressed through targeted education and awareness campaigns. This analysis forms the foundation for improving digital safety practices within the target audience.

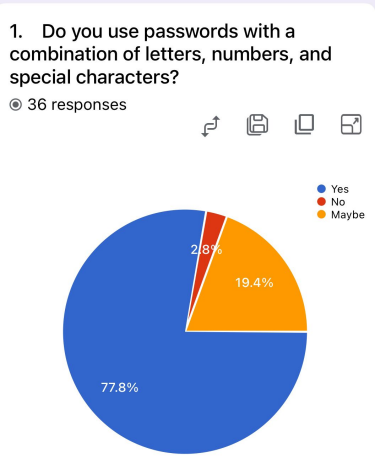


Figure 3: Survey Result 1

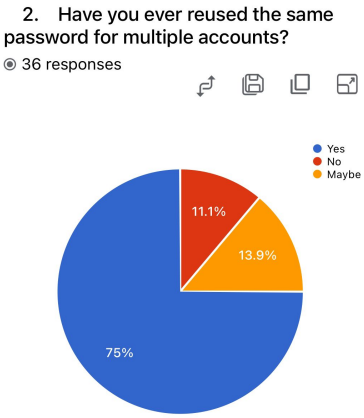


Figure 4: Survey Result 2

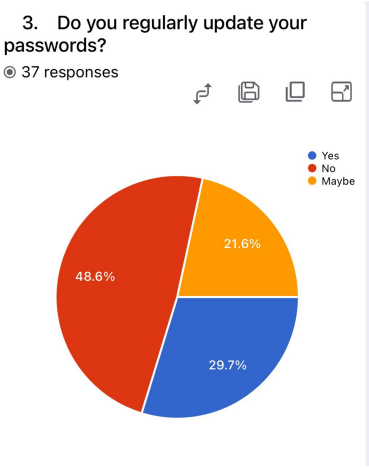


Figure 5: Survey Result 3

Figure 6: Survey Results

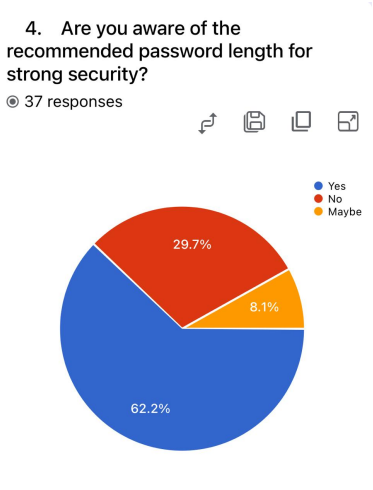


Figure 7: Survey Result 4

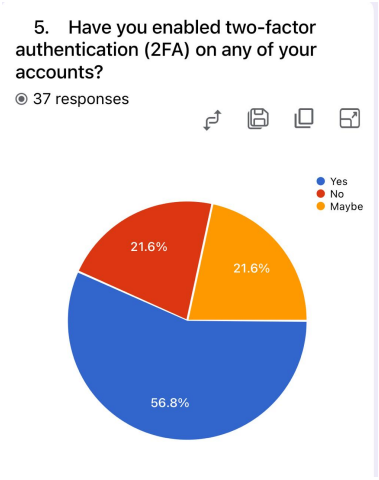


Figure 8: Survey Result 5

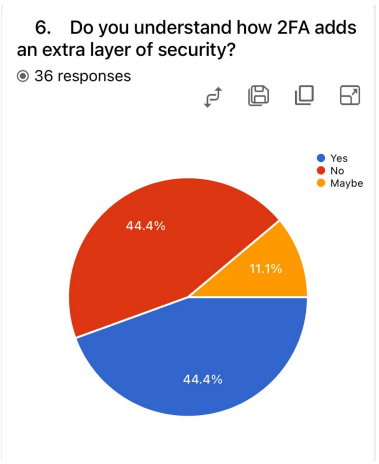


Figure 9: Survey Result 6

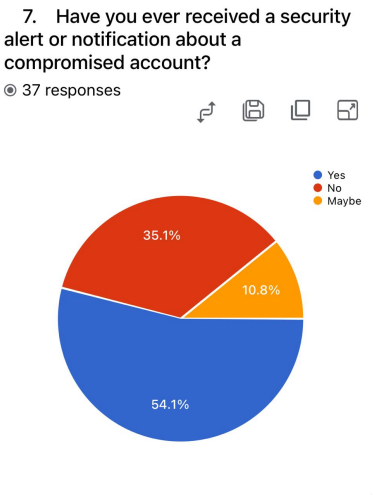


Figure 10: Survey Result 7

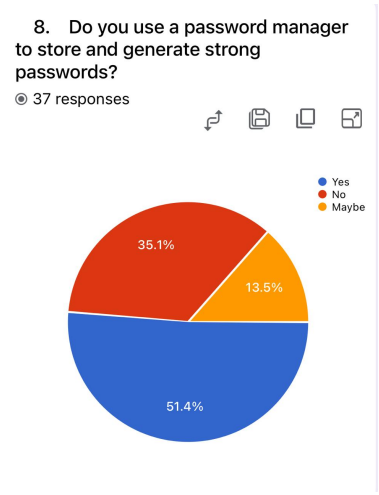


Figure 11: Survey Result 8

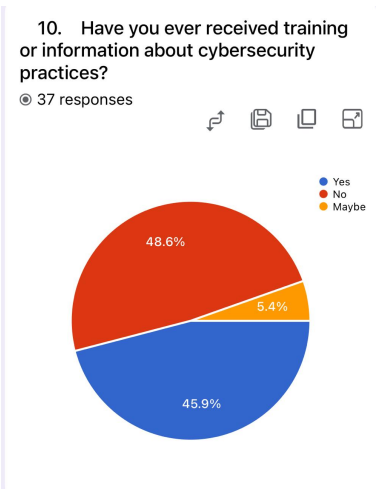


Figure 12: Survey Result 9

We conducted a survey to assess the community's awareness of the importance of strong passwords and two-factor authentication. Our target audience consisted of adults aged approximately 18–30. The following figures illustrate their responses, where we concluded that the majority have sufficient awareness. However, since our application targets children and the elderly, it may be necessary to conduct a more comprehensive study in the future.

0.3.0.3 Stakeholders

The primary stakeholders are less tech-savvy teenagers and adults in Saudi Arabia. They will use the game to improve their awareness of cybersecurity practices like strong passwords and two-factor authentication (2FA). Educational institutions may also use it to promote digital safety among students.

0.3.1 Functional Requirements

FR	Category	Requirement	Priority
1	Registration	The system must enforce password strength requirements, including minimum length and a mix of character types.	Medium
2	Registration	The system must allow the user to create an account.	High
3	Registration	The system must validate the email address format during registration.	Medium
4	Registration	The system must allow users to reset their password through a secure process, including email verification.	High
5	Registration	The system must ensure that usernames are unique and not already in use.	Low
6	Login	The system must allow users to log in using their registered email address and password.	Medium
7	Login	The system must provide feedback for successful and unsuccessful login attempts.	Low
8	Login	The system must lock the account after a specified number of unsuccessful login attempts to prevent unauthorized access.	High
9	Game Stages and Progression	The system should display the 3 stages of the game.	High
10	Game Stages and Progression	The user must successfully complete a certain number of tasks to pass the stage and move on to the remaining stages.	High
11	Game Stages and Progression	The user must complete the first stage (room) to move to the second stage.	High
12	Game Stages and Progression	The system must set a timer for each stage.	Medium
13	Game Stages and Progression	The system should store the progress of each user in the database.	Medium
14	Game Stages and Progression	The user should be able to quit the game.	Medium
15	Gameplay Mechanics	Players should be able to move characters in multiple directions.	Medium
16	Gameplay Mechanics	The user should be able to move by touching the screen.	High
17	Gameplay Mechanics	The user should be able to pause the game.	Medium
18	Gameplay Mechanics	The user should be able to quit the game.	Medium

19	Gameplay Mechanics	Provide in-game notifications for events such as completing quests or finding items.	Low
20	Gameplay Mechanics	Allow players to save their progress at specific points in the game.	Low
21	Gameplay Mechanics	Implement a character progression system that allows players to earn experience points and unlock new abilities.	Medium
22	Gameplay Mechanics	Include tutorials or help prompts to guide new players through the mechanics.	High
23	Player Interaction and Communication	The system should display a green color in the progress bar for successfully completed tasks.	Medium
24	Player Interaction and Communication	The system should display a red color in the progress bar for failed tasks.	Low
25	Player Interaction and Communication	The system should display the consequences if they choose a wrong action in the task.	High
26	Player Interaction and Communication	The user can view the results of the players.	High
27	Rewards and Achievements	The system must track player achievements and granting rewards for milestones.	Medium

Table 2: Functional Requirements

0.3.2 Non-Functional Requirements

NFR	Category	Requirement	Priority
1	Performance	The system should respond to user actions promptly for a more responsive experience.	High
2	Performance	The application shall support multiple users at the same time.	Medium
3	Performance	The user should be able to move between rooms smoothly.	High
4	Usability	The interfaces will be easy to use and user-friendly.	High
5	User Interface	The interface should strategically use colors and animations to direct the user's attention.	Low
6	User Interface	The interface should be simple and clear, using common UI elements of the game.	Medium
7	User Interface	The interface should avoid unnecessary elements.	Medium
8	Reliability	The system should perform its function with the required precision.	High
9	Reliability	The system must perform all its functions successfully.	High
10	Reliability	The system must return accurate accounts of the player's success in the room and their progress.	High
11	Availability	The system should be restarted after failure in less than 14 seconds 70% of the time.	Medium
12	Compatibility	The system shall operate on Android OS and iOS.	High
13	Compatibility	The application should be able to run on iOS version 16 and on Android version 10 and above.	High
14	Security and Privacy	The application must ensure that user confidential data cannot be accessed by unauthorized persons.	High
15	Security and Privacy	The system shall provide the user with 2 attempts to insert the passcode.	Medium
16	Security and Privacy	The system shall allow the user a maximum of 2 minutes to insert a passcode.	Medium

Table 3: Non-Functional Requirements

0.3.3 Use-Case Diagram

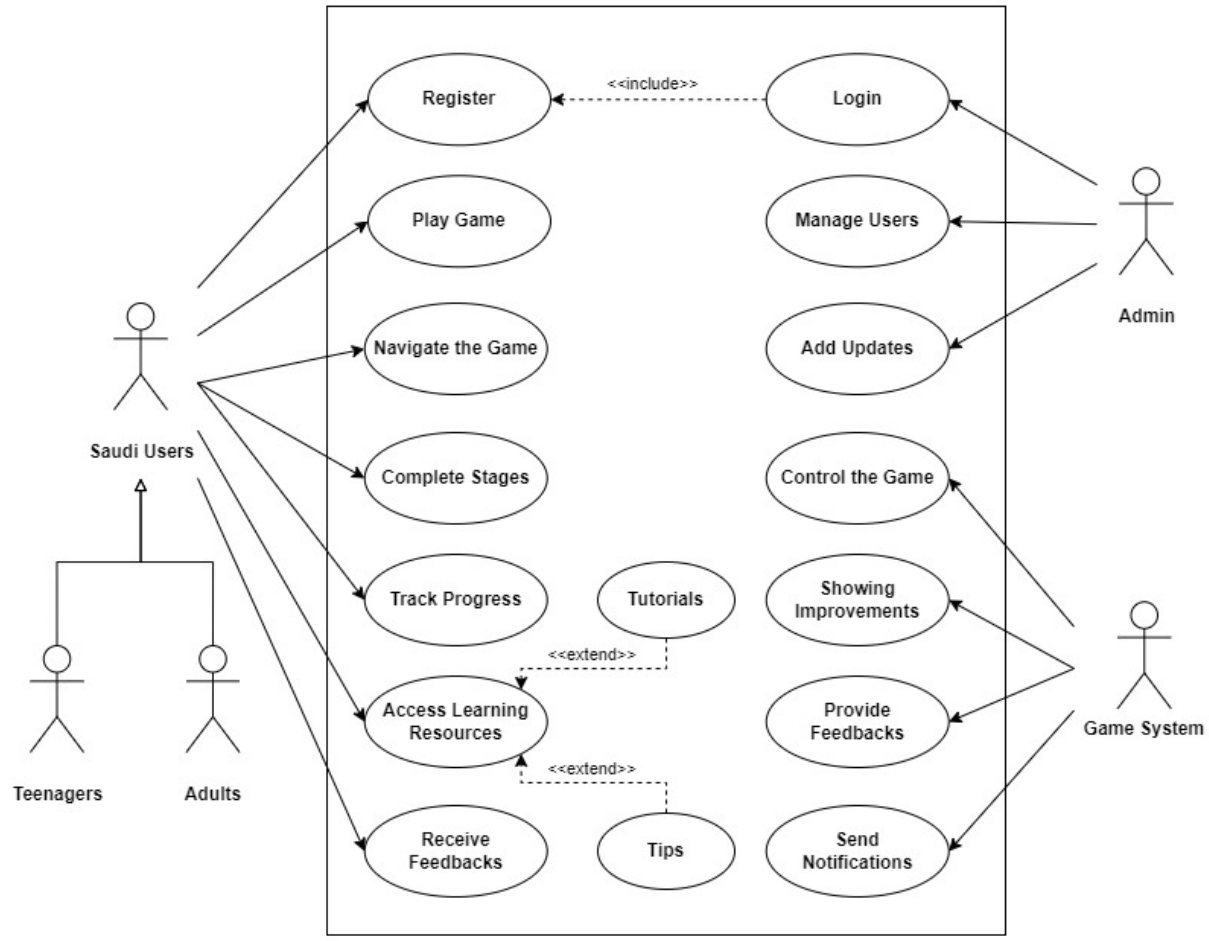


Figure 13: Use Case Diagram

0.3.4 Use-case Specifications

Name	**3.5.1 Use Case: Register**	
Actor	Saudi Users (Teenagers, Adults)	
Flow of Events	Actor	System
	1- Initiates the registration process. 2- Provides the required details and submits the form.	1- Prompts the user for personal details (e.g., username, email, password). 2- Validates the information and creates the user's account.
Exceptions	-Invalid data (e.g., already registered email or incorrect password format). -System failure during registration (e.g., database connection issues).	
Precondition	User must have internet access.	
Postcondition	The user is successfully registered and redirected to the login screen.	
Non-Functional Requirements	The system should handle the registration process within 2 seconds.	

Name	**3.5.2 Use Case: Play Game**	
Actor	Saudi Users	
Flow of Events	Actor	System
	1- Select the start to play a game. 2- Navigates through the game stages, completing tasks and challenges.	1- Loads the game interface and allows the user to interact with it. 2- Updates progress in real-time and stores game data.
Exceptions	-Game loading failure due to internet issues or system crashes. -User's progress not being saved correctly.	
Precondition	The user must be logged in and have a valid account.	
Postcondition	The game stage is completed or progress is saved.	
Non-Functional Requirements	Game performance should be smooth with no more than a 1-second delay in responsiveness.	

Figure 14: Use Cases: Register & Play Game

Name	3.5.3 Use Case: Manage Users	
Actor	Admin	
Flow of Events	Actor	System
	1- Logs into the admin panel. 2- Selects a user to edit, delete, or update their information.	1- Displays a list of users. 2- Processes the action (e.g., updates, deletes the selected user).
Exceptions	System rejects invalid inputs (e.g., incorrect user ID).	
Precondition	The admin must be logged in with elevated privileges.	
Postcondition	The selected user's information is updated or removed.	
Non-Functional Requirements	-Admin actions should reflect changes in the user base within 3 seconds. -The system should be secure and accessible only by authorized personnel.	

Name	3.5.4 Use Case: Provide Feedback	
Actor	Saudi Users	
Flow of Events	Actor	System
	1- Accesses the feedback section after completing a game. 2- Submits feedback.	1- Prompts the user to input feedback (e.g., rating, comments). 2- Stores the feedback and may notify the admin.
Exceptions	-Feedback submission fails due to server issues. -The system rejects feedback if certain fields are incomplete.	
Precondition	The user must have played the game.	
Postcondition	Feedback is successfully stored in the system	
Non-Functional Requirements	Feedback submission should be immediate with confirmation shown to the user.	

Figure 15: Use Cases: Manage Users & Provide Feedback

0.3.5 Design Constraints

- **User Accessibility:** The system must provide an intuitive and user-friendly interface that accommodates both teenagers and adults, ensuring ease of use for individuals with varying levels of technical proficiency.
- **Performance Requirements:** The registration process must be completed within 2 seconds, and game performance should ensure a maximum response delay of 1 second to maintain a smooth and engaging user experience.
- **Security Protocols:** All user data must be handled securely during registration, gameplay, and feedback submission, with strong authentication and authorization mechanisms to prevent unauthorized access.
- **Error Handling and Feedback:** The system must effectively handle exceptions, providing clear, immediate feedback to users regarding any errors encountered during registration, gameplay, or feedback submission.
- **Data Persistence and Integrity:** User progress, feedback, and other critical data must be reliably stored to prevent loss in case of system failures, ensuring that users do not lose their progress or submitted feedback.

0.3.6 Conclusion

This chapter covered the class diagram, requirements (functional and non-functional), use case specification, and design constraints. The game system presented in this chapter caters to both teenage and adult Saudi users, providing a user-friendly and secure experience. The important features include seamless registration, engaging gameplay, progress tracking, and a feedback mechanism. Overall, the game system is designed to deliver a high-quality interactive experience that meets the needs of the Saudi user population.

CHAPTER 4: Methodology and Tools

0.4 Methodology and Tools

Adopting effective methodologies is essential for delivering high-quality products that meet user needs. This project embraces *Agile methodology* to promote flexibility, collaboration, and iterative progress throughout the development lifecycle. By emphasizing continuous feedback and adaptive planning, Agile allows our team to respond swiftly to changing requirements, ensuring the final product aligns closely with user expectations. To enhance these Agile practices, we utilize Teamwork Project , a powerful project management tool that streamlines task tracking, sprint planning, and team collaboration. With its customizable workflows and real-time reporting capabilities, Teamwork fosters transparency and accountability, keeping all team members aligned and informed about project developments.

As we move into the implementation phase, we are integrating GitHub Repository to facilitate version control and collaborative coding. GitHub provides a robust platform for code management, enabling our development team to work together effectively while maintaining a clear history of changes. Utilizing features such as pull requests and code reviews not only enhances code quality but also encourages knowledge sharing among team members. By combining Agile methodologies with the tools offered by Teamwork and GitHub, we are positioned to deliver a responsive and adaptive development process, ultimately leading to a successful software project that meets the needs of our users.

0.4.1 Product Backlog

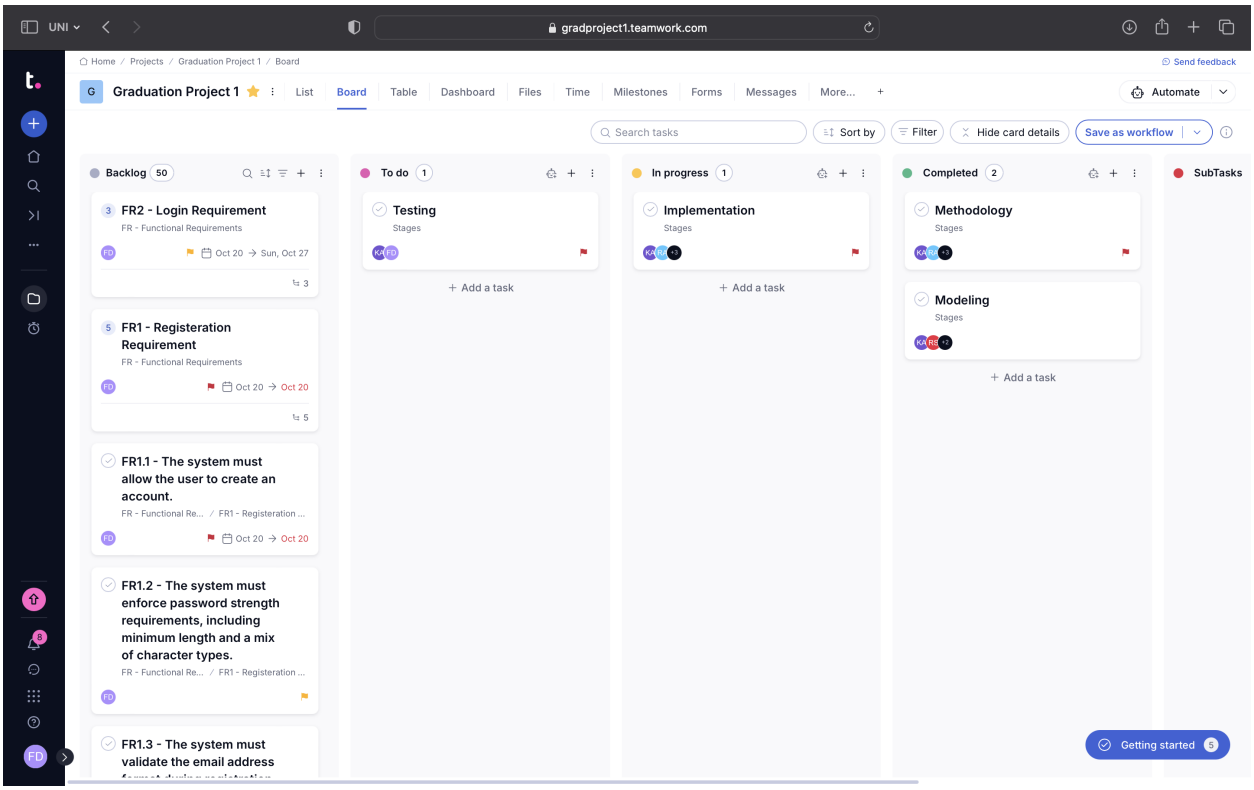
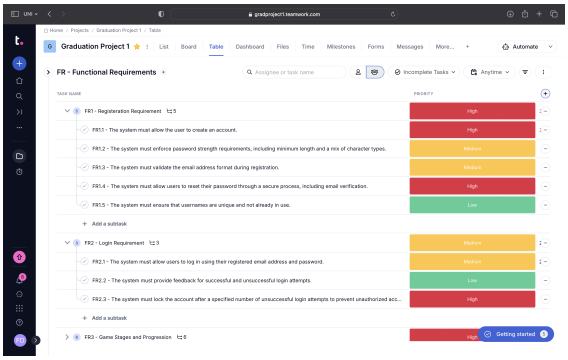
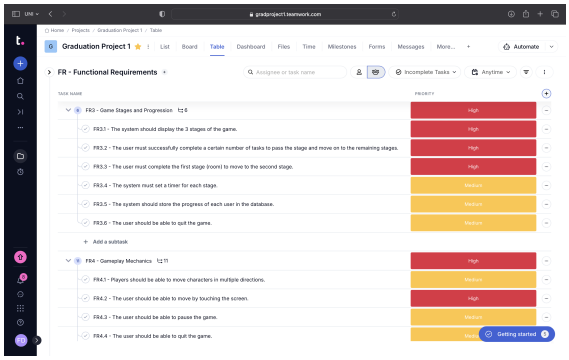


Figure 16: Enter Caption

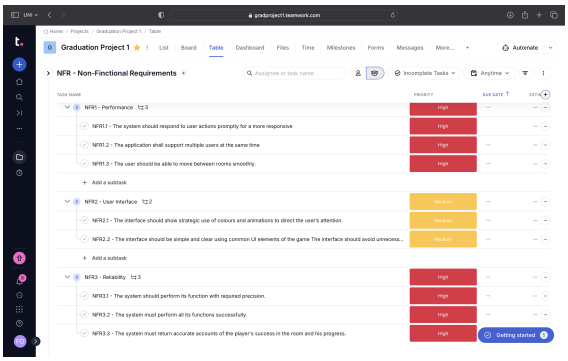
0.4.2 Sprint Backlog



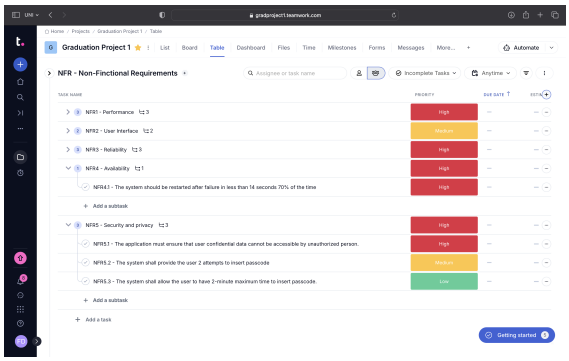
(a) FR1 1/2



(b) FR2 1/2



(c) NFR 1/2



(d) NFR2 2/2

Figure 17: Overall Sprint Backlog

0.4.3 Tasks and their allocation

Task	Start Date	Due Date	Responsible	Created By	Priority
Methodology			karema alharbi, Roua Alnefai, Raoom Sami, Raghad Alsaadi, Fareeda Daghestani	Fareeda Daghestani	High
Implementation			karema alharbi, Roua Alnefai, Raoom Sami, Raghad Alsaadi, Fareeda Daghestani	Fareeda Daghestani	High
Testing			karema alharbi, Fareeda Daghestani	Fareeda Daghestani	High
Modeling			karema alharbi, Roua Alnefai, Raoom Sami, Raghad Alsaadi, Fareeda Daghestani	Fareeda Daghestani	High

Figure 18: Tasks and their allocation

0.4.4 Burn down chart

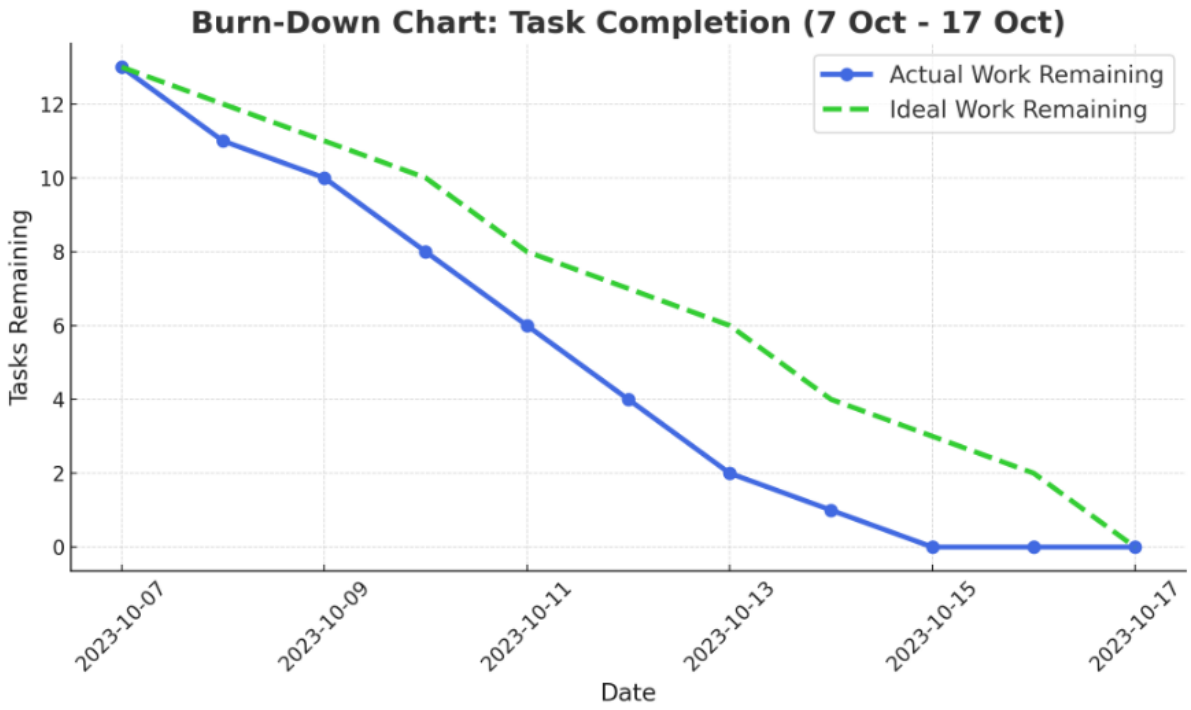


Figure 19: Burn down chart

0.4.5 Game Mechanism

Protection Motivation Theory (PMT) serves as a foundational framework for the design of this game. The theory emphasizes individuals' motivation to protect themselves based on their perceived severity and vulnerability to threats, as well as the efficacy of the protective actions they can take. Each stage of the game is carefully designed to align with PMT, reinforcing the importance of cybersecurity practices. It is designed to help individuals understand and adopt protective behaviors against perceived threats by evaluating the severity of the threat, their vulnerability, and the effectiveness of the protective actions they can take. This theory aligns perfectly with your project as it focuses on raising awareness and changing behaviors to protect users from cybersecurity risks.

0.4.5.1 Stage 1: Protecting Digital Assets

Focus: Creating and storing strong, unique passwords for each type of asset (e.g., treasure rooms, royal documents, palace gates).

Mini-Game 1: Password Creation Challenge

- **Objective:** Help the player create strong, secure passwords to protect key assets of the kingdom.
- **Gameplay:** The player is tasked with securing different assets by creating a password for each. Each asset has rules for password complexity, such as length, inclusion of

numbers, symbols, and mixed case letters. Players interact with a password creation interface to drag and drop characters into a "password generator" box. A strength bar provides real-time feedback on password complexity.

- **Learning Outcome:** This mini-game demonstrates the importance of complex passwords, emphasizing random, long, and varied character types.

Mini-Game 2: Password Maze Challenge

- **Objective:** Teach the player to recognize the importance of adding complexity to passwords by navigating through a maze.
- **Gameplay:** The player navigates a maze where each path represents a different password component (letters, symbols, numbers). Dead-ends represent weak passwords, while correct paths lead to strong, complex passwords.
- **Learning Outcome:** Players understand the significance of using diverse characters and avoiding easily guessable patterns.

0.4.5.2 Stage 2: Knowing Adversaries

Focus: Identifying attacks that bypass or weaken password security and understanding the role of Two-Factor Authentication (2FA) in countering them.

Mini-Game 1: 2FA Defense Challenge

- **Objective:** Help players defend their assets from cyber-attacks using appropriate 2FA methods.
- **Gameplay:** Players face adversaries (e.g., password-guessing marauders, phishing knights, brute-force bots) and select suitable 2FA methods to block attacks. Correct choices deploy defenses like biometric authentication, app-based 2FA, or time-based one-time passwords.
- **Learning Outcome:** Players learn to identify cyber-attacks and apply effective 2FA solutions.

Mini-Game 2: Guessing Goblin Quiz

- **Objective:** Educate players about common password vulnerabilities and defenses.
- **Gameplay:** Players answer rapid-fire questions about password vulnerabilities and countermeasures posed by a goblin character. Correct answers reinforce security, while incorrect answers weaken defenses.
- **Learning Outcome:** This quiz emphasizes common password mistakes and the importance of 2FA.

0.4.5.3 Stage 3: Preparing to Defend

Focus: Testing password and 2FA strength against simulated attacks.

Mini-Game 1: Battle Simulator

- **Objective:** Test and refine password and 2FA defenses through simulated attacks.
- **Gameplay:** Players face waves of cyber-attacks (e.g., brute force, credential stuffing) and must adapt passwords and 2FA settings to block them.
- **Learning Outcome:** Players practice strengthening passwords and using dynamic defenses.

Mini-Game 2: Password Storm

- **Objective:** Strengthen defenses against random attack waves.
- **Gameplay:** Players create and upgrade passwords by dragging characters into stronger combinations to repel attack storms.
- **Learning Outcome:** Players understand the ongoing nature of cybersecurity threats and the need for dynamic defenses.

0.4.5.4 Stage 4: Building a Resilient Defense

Focus: Reinforcing long-term defenses with password best practices and multi-layered 2FA.

Mini-Game 1: Resilience Upgrade

- **Objective:** Improve long-term security by upgrading passwords and 2FA.
- **Gameplay:** Players assess assets' defenses and decide on improvements, such as strengthening passwords or enabling additional 2FA layers.
- **Learning Outcome:** Players learn the importance of continuous vigilance and upgrades.

Mini-Game 2: The Final Lockdown

- **Objective:** Finalize defenses by applying learned practices to repel final waves of attacks.
- **Gameplay:** Players must secure all assets by combining strong passwords, multi-layered 2FA, and recovery options.
- **Learning Outcome:** Reinforces the value of proactive defense and comprehensive cybersecurity measures.

0.4.5.5 Following the Framework of PMT

The game aligns with **Protection Motivation Theory (PMT)** by:

- **Perceived Severity:** Players see the consequences of weak security through breaches and lost assets.
- **Perceived Vulnerability:** Visual feedback and attack scenarios demonstrate how easily weak defenses can be compromised.
- **Response Efficacy:** Gameplay reinforces that strong passwords and 2FA significantly enhance protection.
- **Self-Efficacy:** Players gain confidence through hands-on learning and immediate feedback.
- **Rewards:** Security points, unlockable boosters, and visual progress indicators motivate players to adopt secure behaviors.

The stages and mini-games are designed to educate players on cybersecurity best practices while making the learning experience engaging and interactive.

The game leverages **Protection Motivation Theory (PMT)** to educate and motivate users toward secure online behaviors. Players grasp the **perceived severity** of weak security through simulated breaches and visualize **perceived vulnerability** via attack scenarios. **Response efficacy** is highlighted by demonstrating how strong passwords and 2FA bolster protection, while hands-on gameplay fosters **self-efficacy**, building confidence. Motivational elements like rewards and progress indicators encourage consistent engagement with best practices, ensuring players recognize and mitigate cyber threats effectively.

PMT is central to this project as it emphasizes the importance of motivating individuals to adopt protective measures against threats. By illustrating risks, vulnerabilities, and effective defenses, the project ensures that users gain practical knowledge and actionable skills to enhance their cybersecurity posture.

Through interactive learning, users are empowered to create strong passwords, enable 2FA, recognize phishing attempts, and adopt proactive measures like regular updates and monitoring account activity. These steps reduce risks and instill confidence in managing digital security, creating a strong defense system.

By integrating awareness, practical steps, and engaging activities, the project offers a comprehensive approach to cybersecurity. This ensures users develop lasting habits, enhancing their ability to protect themselves from evolving digital threats.

0.4.6 Engineering Standards

Game System Requirements Specification (Based on ISO 25000)
We extracted what we think would be the most suitable for our application based on the standards in the ISO/IEC 25000 model (also known as SQuaRE - System and Software Quality Requirements and Evaluation).

0.4.6.1 Functional Requirements (ISO/IEC 25010 - Functional Suitability)

The system must meet the following functional requirements to ensure its suitability for use by end users:

User Registration and Authentication (ISO/IEC 25010 - Functional Completeness)

- Users must be able to register an account by providing their name and group name.
- The system must send a verification email after registration to confirm the account.
- Users must be able to log in to their accounts using valid credentials.
- The system should provide an option to reset forgotten passwords.
- The system must allow users to delete or modify their accounts.

Gameplay Progression and Feedback (ISO/IEC 25010 - Functional Correctness)

- The game will present three stages, with users required to complete tasks in each stage to advance.
- A timer must be set for each stage to track the player's performance.
- A green progress bar must indicate completed tasks, while a red progress bar indicates failed tasks.
- The system must store and track each user's progress in the database.
- The system must notify users of consequences when they make wrong decisions.

Player Interaction and Movement (ISO/IEC 25010 - Functional Appropriateness)

- Players must be able to move characters in multiple directions and interact with the environment by touching the screen.
- The system should allow players to pause or quit the game.
- The game must include a character progression system with experience points and unlockable abilities.

Player Communication and Support Features

- The system must include text or voice chat for player communication.
- Players must be able to view the results and achievements of other players.
- The game must provide tutorials and help prompts to guide new players.

0.4.6.2 Non-Functional Requirements (ISO/IEC 25010 - Quality Characteristics)

Performance Efficiency (ISO/IEC 25010 - Time Behavior and Resource Utilization)

- The system must respond promptly to user inputs for an interactive experience.
- The application should support multiple users concurrently without degradation of performance.
- The transition between game stages or rooms must be smooth, with minimal lag.

Usability (ISO/IEC 25010 - Operability)

- The user interface must be easy to navigate, intuitive, and user-friendly.
- The system must offer customizable text size options and alternative color schemes to support accessibility (e.g., for colorblind players).
- Strategic use of colors and animations must be employed to enhance the user's focus and understanding.

Reliability (ISO/IEC 25010 - Maturity and Fault Tolerance)

- The system must perform all its intended functions accurately and consistently.
- The system must record and report the player's success rate and track progress accurately.
- In case of system failure, it should recover within 14 seconds, at least 70% of the time.

Compatibility (ISO/IEC 25010 - Interoperability)

- The system must be compatible with iOS version 16 and Android version 10 and above.
- The game must function properly on both Android and iOS devices without significant performance differences.

Security and Privacy (ISO/IEC 25010 - Confidentiality, Integrity)

- User data (including personal information and game progress) must be secured and protected from unauthorized access.
- The system must require two attempts for passcode entry and allow a maximum of two minutes for passcode input.
- Proper encryption and data protection mechanisms must be employed to secure confidential user data.

Maintainability (ISO/IEC 25010 - Modularity and Modifiability)

- The system architecture should be modular to allow easy updates or fixes without impacting the entire system.
- Developers must be able to modify or extend the system with minimal disruptions to existing functionality.

0.4.6.3 Quality in Use (ISO/IEC 25010 - Effectiveness and Satisfaction)

The overall goal is to deliver an engaging and enjoyable experience for users:

Effectiveness

- Users should be able to achieve their objectives (completing levels, progressing characters, etc.) without difficulty.

Efficiency

- The application must minimize time and resource usage while providing a high-quality gaming experience.

Satisfaction

- The game must be designed to be engaging and enjoyable, encouraging users to continue playing and exploring different features.

This engineering standard outlines the critical functional and non-functional requirements aligned with ISO/IEC 25000 (SQuaRE) standards. By adhering to these guidelines, the game system can ensure quality, usability, and security, leading to a high-performing, reliable, and accessible gaming experience.

0.4.7 Conclusion

finally, when we adopt Agile methodologies and utilize powerful tools such as Teamwork and GitHub, this project will ensure a flexible, collaborative, and efficient development process. The integration of ISO/IEC 25000 (SQuaRE) standards further enhances the quality of the game system by addressing both functional and non-functional requirements. With a strong focus on user registration, gameplay progression, and player interaction, as well as performance, usability, reliability, and security, the game is designed to deliver an engaging, accessible, and high-performing experience. This comprehensive approach positions the project for success in meeting user expectations and achieving long-term satisfaction.

CHAPTER 5: Analysis and Design

0.5 Analysis and Design

This chapter focuses on implementing the requirements outlined in the previous chapter for the two-factor authentication (2FA) awareness game. It includes a class diagram to illustrate the relationships between game components, as well as sequence and activity diagrams to depict the flow and interactions during gameplay. Additionally, a state diagram shows the system's key states as players progress through levels, reinforcing 2FA concepts. These diagrams provide a comprehensive view of the game's core functionality and educational objectives.

0.5.1 Class diagram

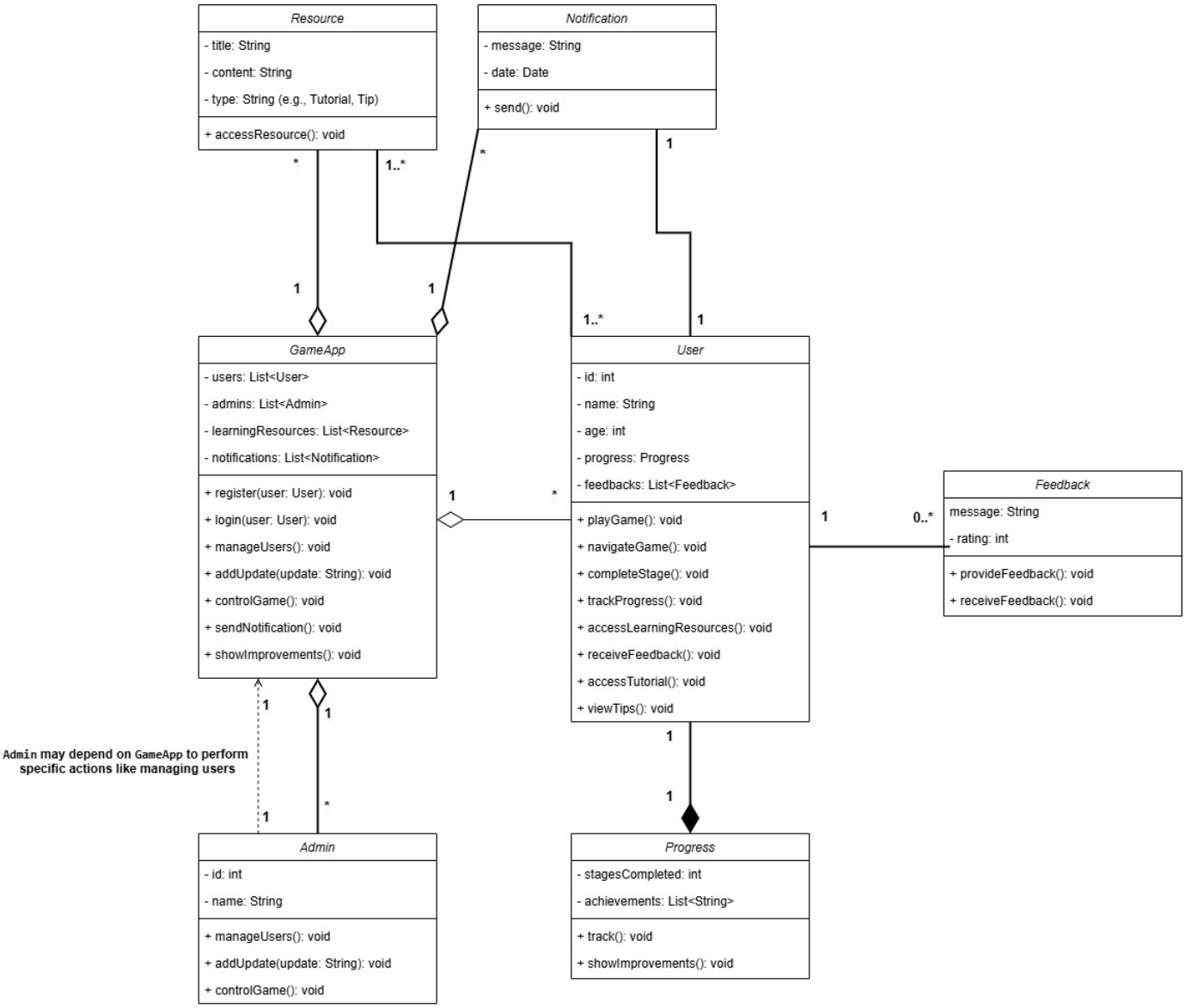


Figure 20: Class Diagram

0.5.2 Sequence Diagram

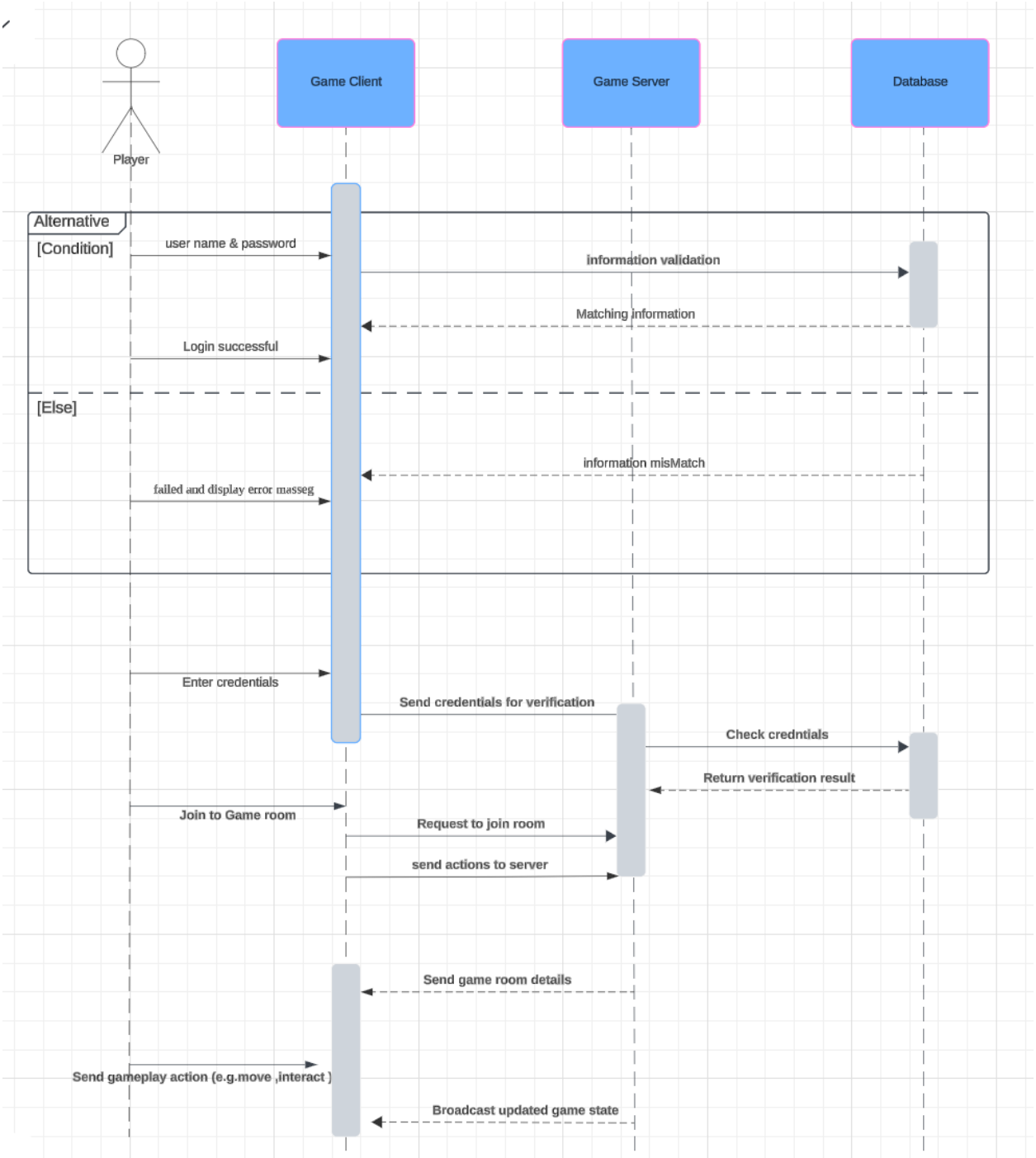


Figure 21: Sequence diagram

0.5.3 Activity Diagram

0.5.3.1 Account Creation

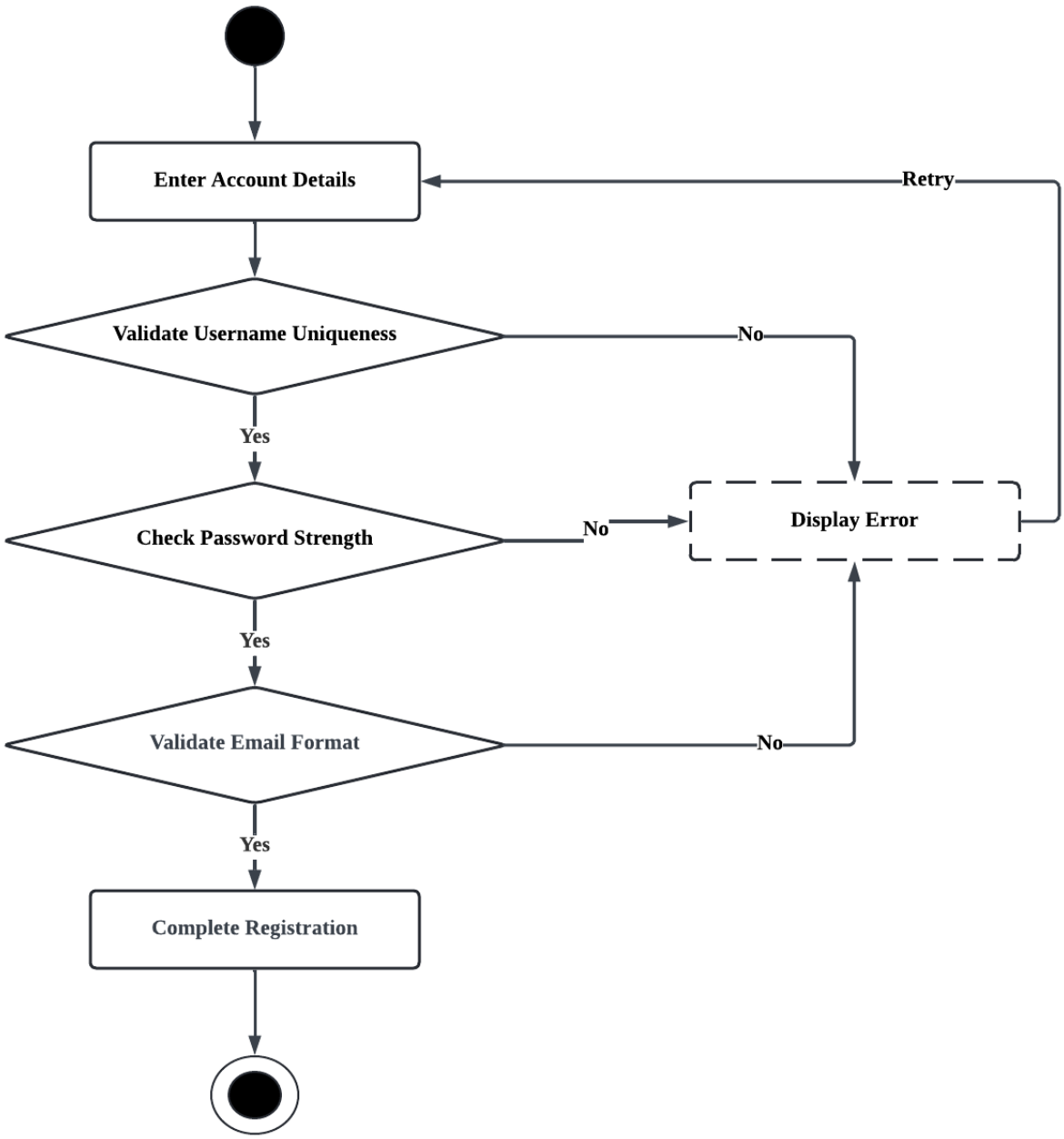


Figure 22: Activity Diagram for Account Creation

0.5.3.2 Login Activity

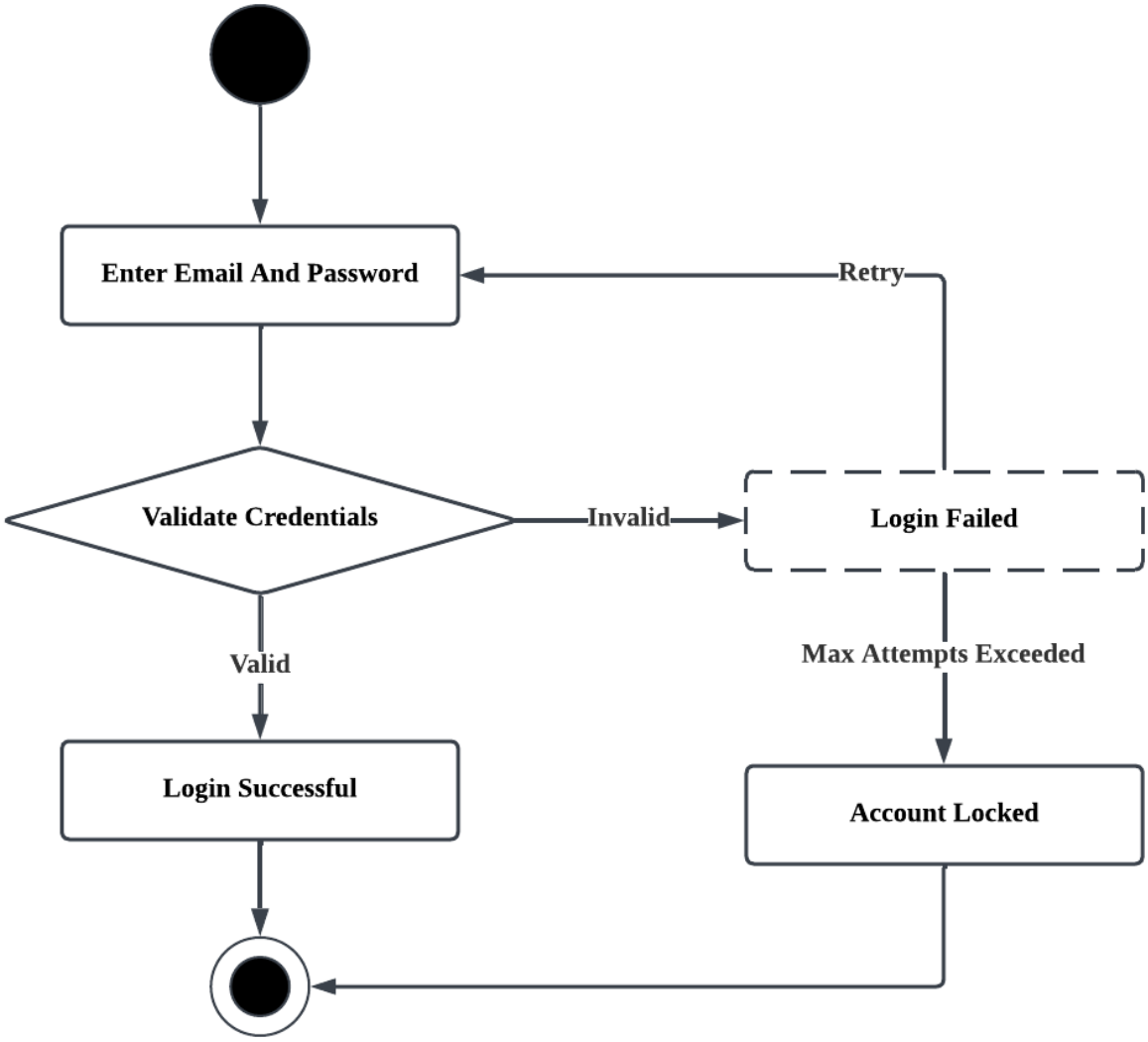


Figure 23: Activity Diagram for Login Activity

0.5.3.3 Player Interaction

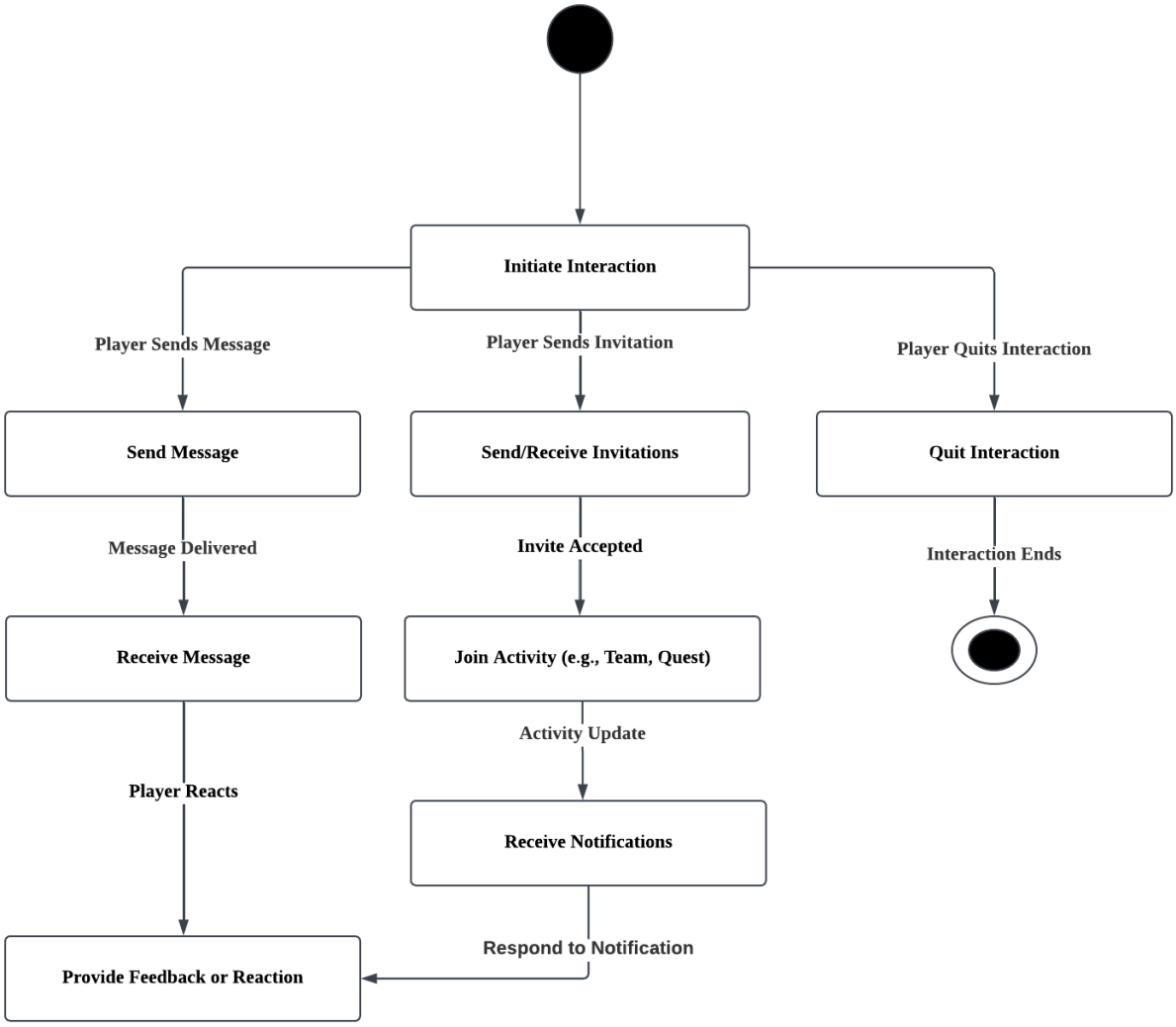


Figure 24: Activity Diagram for Player Interaction

0.5.3.4 Gameplay Mechanics

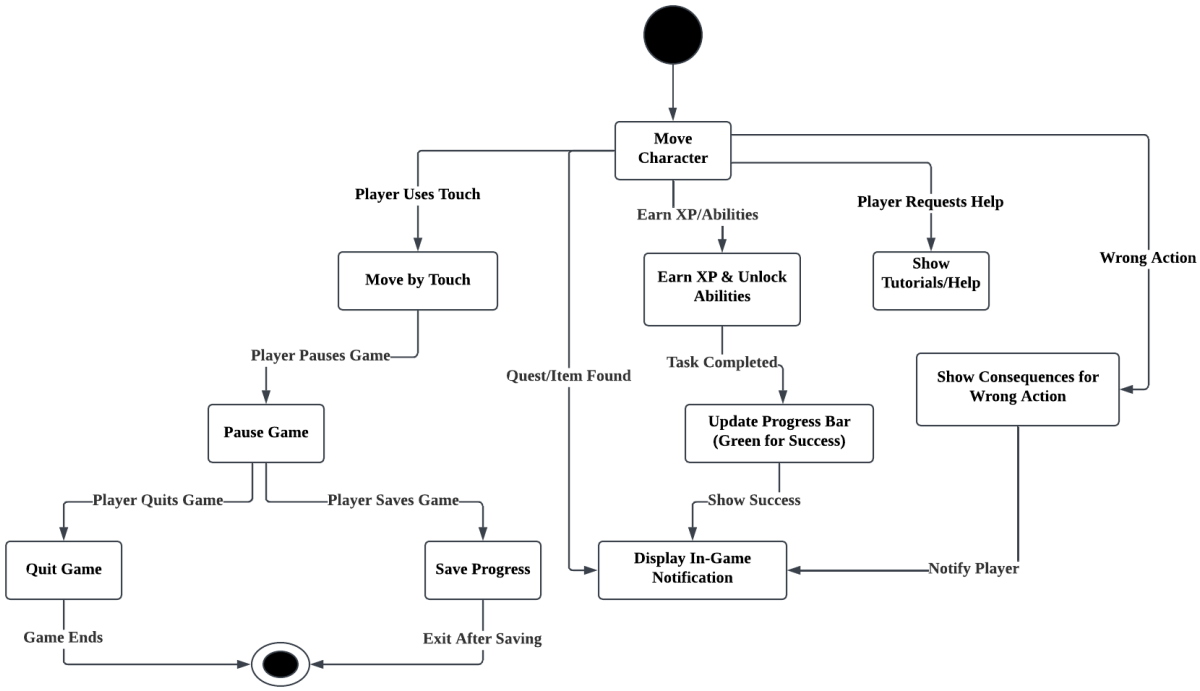


Figure 25: Activity Diagram for Gameplay Mechanics

0.5.3.5 Game Activity

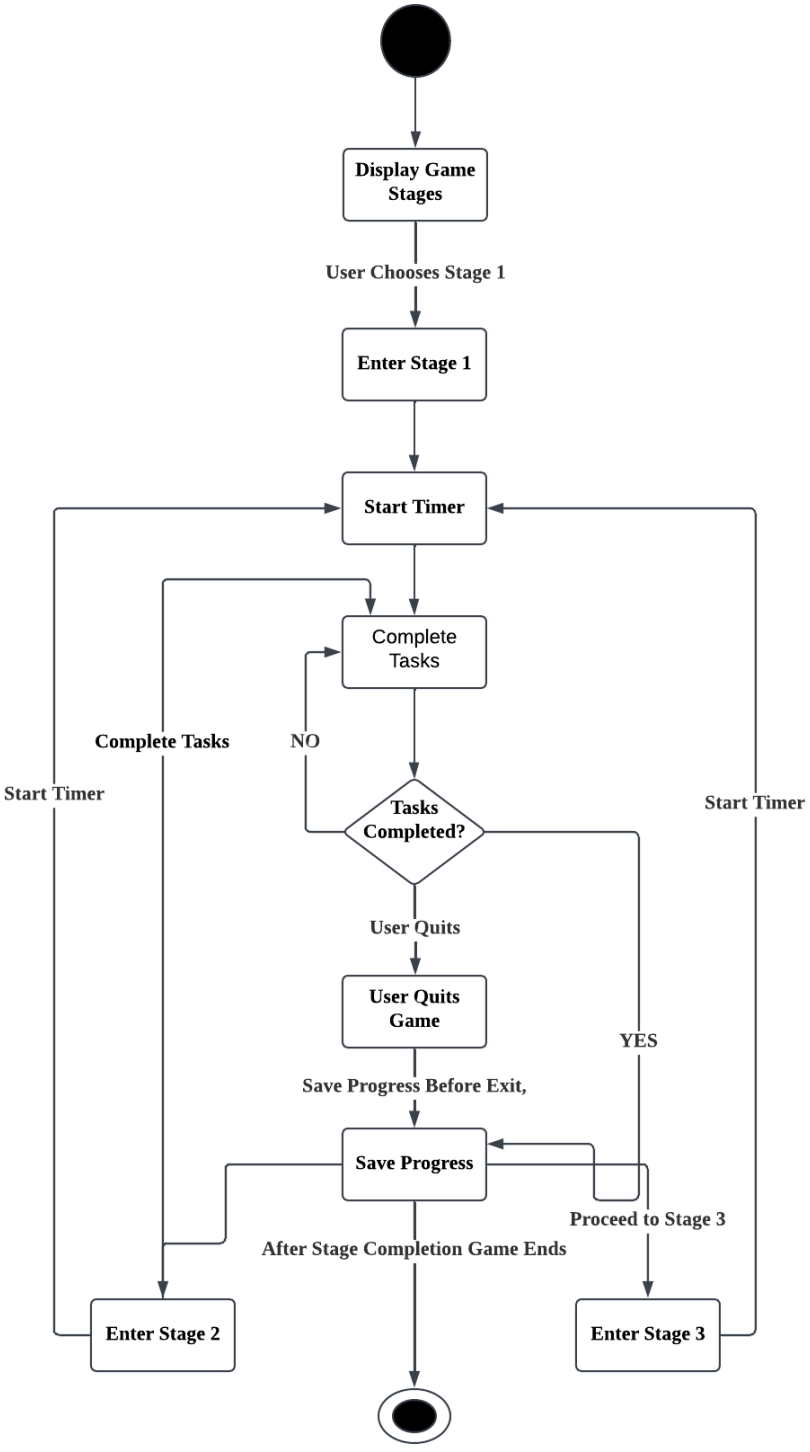


Figure 26: Activity Diagram for Game Activity

0.5.4 State Diagram

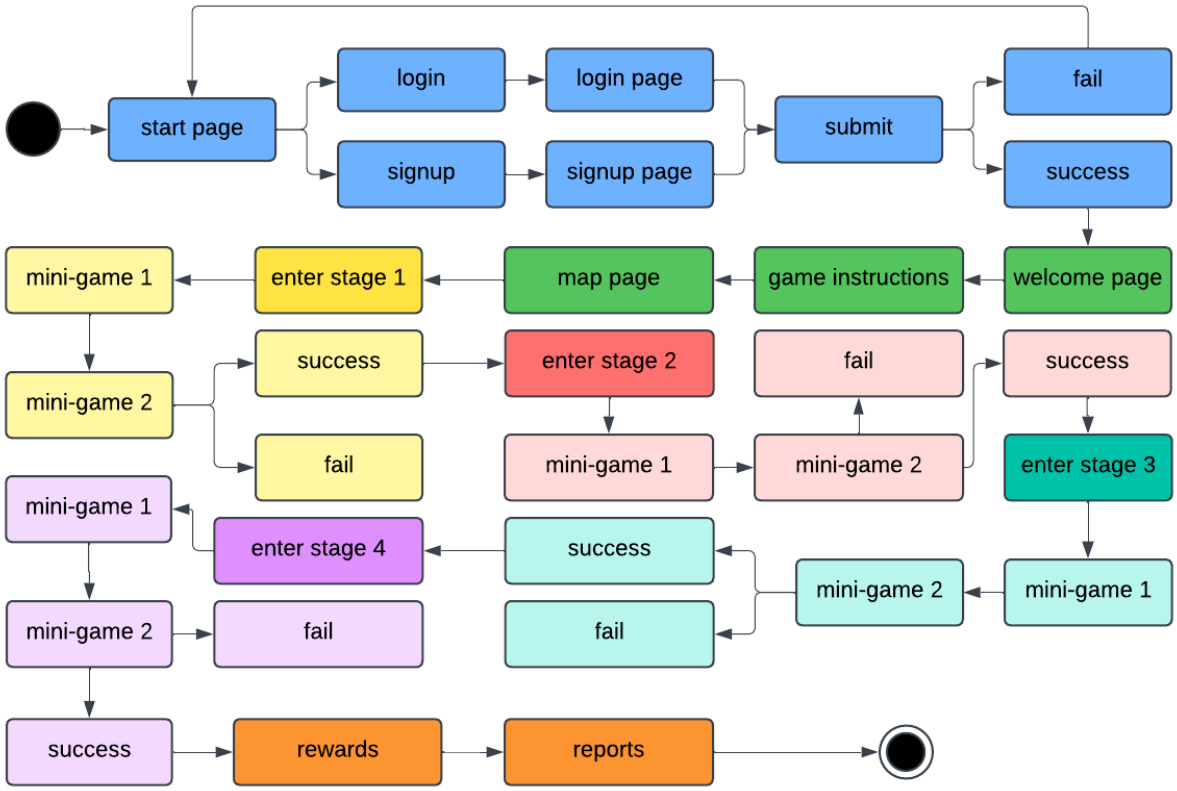


Figure 27: State Diagram

0.5.5 System Architecture

This section describes the system architecture, illustrating the main components and interactions that constitute the design of the system. The architecture ensures effective communication between various modules and provides a scalable foundation to enhance the application's cybersecurity capabilities.

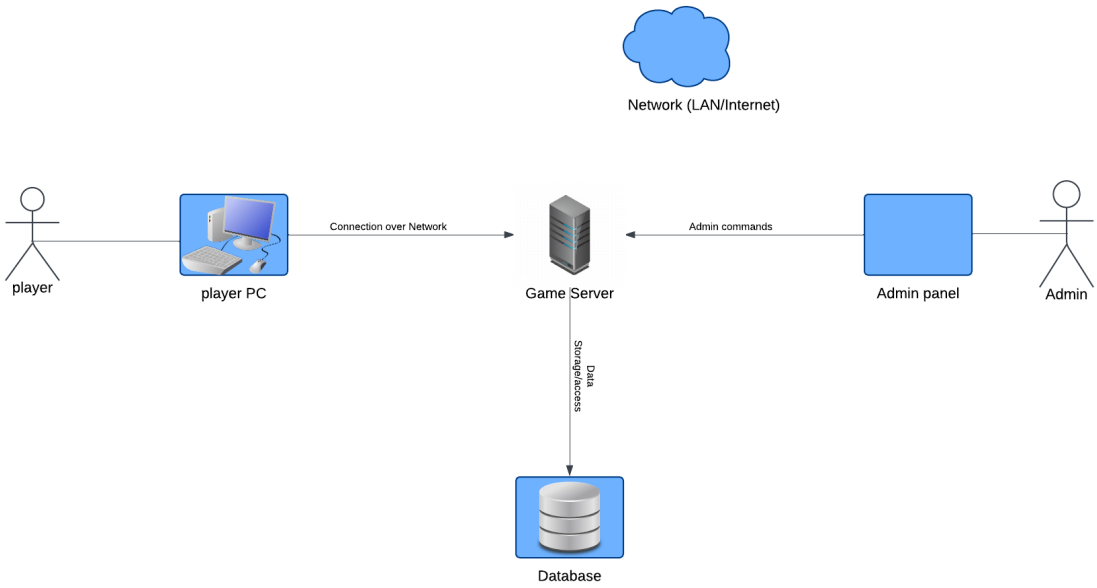


Figure 28: System Architecture Diagram

0.5.6 Conclusion

This chapter presented the implementation framework for the 2FA awareness game, emphasizing its design and educational goals. The class diagram outlined the relationships between core game components, while sequence and activity diagrams demonstrated the flow of user interactions and system processes. The state diagram depicted how players transition through various stages, reinforcing key cybersecurity concepts such as strong password creation and 2FA. Through an engaging and interactive gameplay experience, the game aims to address the global challenges of password vulnerabilities and improve user awareness, ultimately fostering better cybersecurity practices.

CHAPTER 6: Implementation and Testing (Sprint 1)

0.6 Implementation and Testing (Sprint 1)

0.6.1 Introduction

This chapter concentrates on the Flutter implementation and preliminary testing procedures that were completed during Sprint 1 of our project. This sprint's main objective was to create the interactive game's fundamental elements, such as the splash, login, and registration screens, while maintaining a smooth and aesthetically pleasing user experience. To confirm that these functionalities work and that the program operates as intended, basic testing was done. The foundation for more complex development and testing in subsequent sprints is laid by this first stage.

0.6.2 Programming Language and Tools

Our project's main programming language was Dart because of its effectiveness and compatibility with the cross-platform application development framework Flutter.
Our primary code editor was Visual Studio Code, which we used to improve the development process by taking use of its extensive extensions and debugging features. To further test and simulate our application, we used Android Studio.
In order to facilitate development and guarantee a seamless user experience, we used Firebase to manage crucial backend services tasks like authentication and data management. These tools were crucial in creating the application's core functionalities, properly debugging it, and producing an excellent end product.
We want to switch to Unity in the future to add the essential gameplay components, taking use of its robust game production capabilities.

0.6.3 Code Snippets of Main Functions

0.6.3.1 Welcome Screen

```
        ],
      ), // Column
    ), // Expanded
    // الجزء السفلي: الأزرار
    Row(
      children: [
        // زر تسجيل الدخول
        Expanded(
          child: Container(
            decoration: const BoxDecoration(
              color: Colors.white,
              borderRadius: BorderRadius.only(
                topRight: Radius.circular(30),
              ), // BorderRadius.only
            ), // BoxDecoration
            child: ElevatedButton(
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(
                    builder: (context) => const LoginScreen(),
                  ), // MaterialPageRoute
                );
              },
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.white,
                foregroundColor: Colors.green,
                elevation: 0,
                padding: const EdgeInsets.symmetric(vertical: 15.0),
                shape: const RoundedRectangleBorder(
                  borderRadius: BorderRadius.only(
                    topRight: Radius.circular(30),
                  ), // BorderRadius.only
                ), // RoundedRectangleBorder
              ),
              child: Row(
                mainAxisAlignment: MainAxisAlignment.center,
                children: const [
                  Icon(Icons.login),
                  SizedBox(width: 10),
                  Text(
                    'Login',
                    style: TextStyle(
                      fontSize: 18,
                      fontWeight: FontWeight.bold,
                    ), // TextStyle
                  ), // Text
                ],
              ), // Row
            ), // ElevatedButton
          ), // Container
```

Figure 29: Welcome Screen 1/2:Combines the login and register options into one screen. Acts as a gateway for users to choose between logging in or creating a new account. Displays the app's logo and introductory text alongside navigation options.

```
login_register_screen.dart
lib > login_register_screen.dart > LoginRegisterScreen > build
class LoginRegisterScreen extends StatelessWidget {
  Widget build(BuildContext context) {
            ), // Expanded
            // زر التسجيل
            Expanded(
              child: Container(
                decoration: const BoxDecoration(
                  color: Colors.green,
                  borderRadius: BorderRadius.only(
                    topLeft: Radius.circular(30),
                  ), // BorderRadius.only
                ), // BoxDecoration
                child: ElevatedButton(
                  onPressed: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(
                        builder: (context) => const RegisterScreen(),
                      ), // MaterialPageRoute
                    );
                  },
                  style: ElevatedButton.styleFrom(
                    backgroundColor: Colors.green,
                    foregroundColor: Colors.white,
                    elevation: 0,
                    padding: const EdgeInsets.symmetric(vertical: 15.0),
                    shape: const RoundedRectangleBorder(
                      borderRadius: BorderRadius.only(
                        topLeft: Radius.circular(30),
                      ), // BorderRadius.only
                    ), // RoundedRectangleBorder
                  ),
                  child: Row(
                    mainAxisAlignment: MainAxisAlignment.center,
                    children: const [
                      Icon(Icons.app_registration),
                      SizedBox(width: 10),
                      Text(
                        'Register',
                        style: TextStyle(
                          fontSize: 18,
                          fontWeight: FontWeight.bold,
                        ), // TextStyle
                      ), // Text
                    ],
                  ), // Row
                ), // ElevatedButton
              ), // Container
            ), // Expanded
          ],
        ), // Row
      ], // Column
    );
```

Figure 30: Welcome Screen 2/2: Provides navigation buttons for login and registration, serving as the main entry point for users.

0.6.3.2 Login Screen

```
105                 decoration: InputDecoration(
106                   labelText: 'Password',
107                   hintText: 'Enter your password',
108                   border: OutlineInputBorder(
109                     borderRadius: BorderRadius.circular(10),
110                   ), // OutlineInputBorder
111                   filled: true,
112                   fillColor: Colors.blue[50],
113                 ), // InputDecoration
114                 validator: (value) {
115                   if (value == null || value.isEmpty) {
116                     return 'Please enter your password';
117                   } else if (value.length < 6) {
118                     return 'Password must be at least 6 characters long';
119                   }
120                   return null;
121                 },
122               ), // TextFormField
123               const SizedBox(height: 20),
124               // خيار "تذكرني" و "نسيت كلمة المرور"
125               Row(
126                 children: [
127                   Checkbox(
128                     value: rememberMe,
129                     onChanged: (value) {
130                       setState(() {
131                         rememberMe = value ?? false;
132                       });
133                     },
134                     activeColor: Colors.blue,
135                   ), // Checkbox
136                   const Text(
137                     'Remember Me',
138                     style: TextStyle(color: Colors.black),
139                   ), // Text
140                 ],
141               ), // Row
142               Align(
143                 alignment: Alignment.centerRight,
144                 child: GestureDetector(
145                   onTap: () {
146                     Navigator.push(
147                       context,
148                       MaterialPageRoute(
149                         builder: (context) =>
150                             const ForgotPasswordScreen(),
151                       ), // MaterialPageRoute
152                     );
153                   },
154                   child: const Text(
155                     'Forgot Password?',
```

Figure 31: Login Screen 1/2: Manages the user login interface. Provides fields for entering an email and password. Includes a "Remember Me" option and a link to reset forgotten passwords.

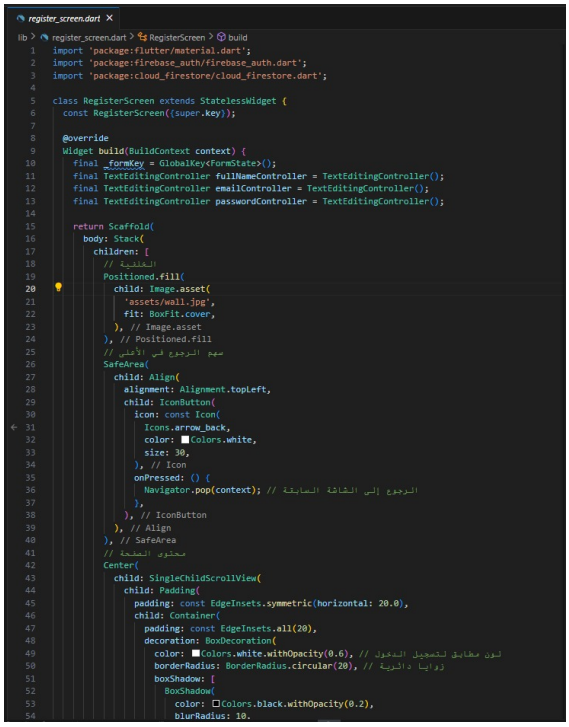
```
54                   color: Colors.white.withOpacity(0.6),
55                   borderRadius: BorderRadius.circular(20),
56                   boxShadow: [
57                     BoxShadow(
58                       color: Colors.black.withOpacity(0.2),
59                       blurRadius: 10,
60                       offset: const Offset(0, 4),
61                     ), // BoxShadow
62                   ],
63                 ), // BoxDecoration
64                 child: Form(
65                   key: _formKey,
66                   child: Column(
67                     mainAxisSize: MainAxisSize.min,
68                     children: [
69                       // عنوان
70                       Image.asset(
71                         'assets/تسجيل الدخول.png',
72                         width: 200,
73                         height: 100,
74                         fit: BoxFit.contain,
75                       ), // Image.asset
76                       const SizedBox(height: 20),
77                       // حقل البريد الإلكتروني
78                       TextFormField(
79                         controller: emailController,
80                         decoration: InputDecoration(
81                           labelText: 'Email',
82                           hintText: 'Enter your email',
83                           border: OutlineInputBorder(
84                             borderRadius: BorderRadius.circular(10),
85                           ), // OutlineInputBorder
86                           filled: true,
87                           fillColor: Colors.blue[50],
88                         ), // InputDecoration
89                         validator: (value) {
90                           if (value == null || value.isEmpty) {
91                             return 'Please enter your email';
92                           } else if (!RegExp(
93                                   r"^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$") // RegExp
94                               .hasMatch(value)) {
95                             return 'Please enter a valid email address';
96                           }
97                           return null;
98                         },
99                       ), // TextFormField
100                      const SizedBox(height: 15),
101                      // حقل كلمة المرور
102                      TextFormField(
103                        controller: passwordController,
104                        obscureText: true,
```

Figure 32: Login Screen 2/2: Validates user inputs and communicates with Firebase for authentication, ensuring secure login.

0.6.3.3 Signup Screen

```
                offset: const Offset(0, 4),
              ), // BoxShadow
            ],
          ), // BoxDecoration
          child: Form(
            key: _formKey,
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.center,
              children: [
                // اضيف الصورة بالعنوان
                Padding(
                  padding: const EdgeInsets.only(bottom: 20),
                  child: Image.asset(
                    'assets/الشعار حساب.png', // مسار الصورة
                    width: 200,
                    height: 100,
                  ), // Image.asset
                ), // Padding
                // حقل الاسم الكامل
                TextFormField(
                  controller: fullNameController,
                  decoration: InputDecoration(
                    labelText: 'Full Name',
                    hintText: 'Enter your full name',
                    border: OutlineInputBorder(
                      borderRadius: BorderRadius.circular(10),
                    ), // OutlineInputBorder
                    filled: true,
                    fillColor: Colors.grey.withOpacity(0.1),
                  ), // InputDecoration
                  validator: (value) {
                    if (value == null || value.isEmpty) {
                      return 'Please enter your full name';
                    }
                    return null;
                  },
                ), // TextFormField
                const SizedBox(height: 15),
                // حقل البريد الإلكتروني
                TextFormField(
                  controller: emailController,
                  keyboardType: TextInputType.emailAddress,
                  decoration: InputDecoration(
                    labelText: 'Email',
                    hintText: 'Enter your email',
                    border: OutlineInputBorder(
                      borderRadius: BorderRadius.circular(10),
                    ), // OutlineInputBorder
                    filled: true,
                    fillColor: Colors.grey.withOpacity(0.1),
                  ), // InputDecoration
```

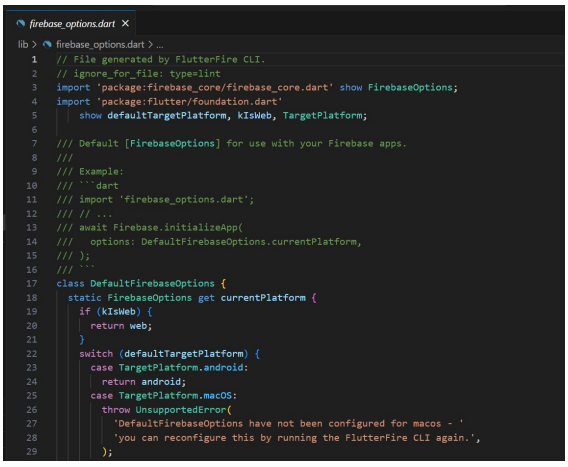
Figure 33: Signup Screen 1/2: Handles the user registration process. Collects user details such as full name, email, and password.



```
register_screen.dart
lib > register_screen.dart > RegisterScreen > build
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:cloud_firestore/cloud_firestore.dart';

class RegisterScreen extends StatelessWidget {
  const RegisterScreen({super.key});

  @override
  Widget build(BuildContext context) {
    final _formKey = GlobalKey<FormState>();
    final TextEditingController fullNameController = TextEditingController();
    final TextEditingController emailController = TextEditingController();
    final TextEditingController passwordController = TextEditingController();

    return Scaffold(
      body: Stack(
        children: [
          // الخلفية
          Positioned.fill(
            child: Image.asset(
              'assets/wall.jpg',
              fit: BoxFit.cover,
            ), // Image.asset
          ), // Positioned.fill
          // سهم الرجوع في الأعلى
          SafeArea(
            child: Align(
              alignment: Alignment.topLeft,
              child: IconButton(
                icon: const Icon(
                  Icons.arrow_back,
                  color: Colors.white,
                  size: 30,
                ), // Icon
                onPressed: () {
                  Navigator.pop(context); // الرجوع إلى الشاشة السابقة
                },
              ), // IconButton
            ), // Align
          ), // SafeArea
          // محتوى الصفحة
          Center(
            child: SingleChildScrollView(
              child: Padding(
                padding: const EdgeInsets.symmetric(horizontal: 20.0),
                child: Container(
                  padding: const EdgeInsets.all(20),
                  decoration: BoxDecoration(
                    color: Colors.white.withOpacity(0.6), // لون مطابق لتسجيل الدخول
                    borderRadius: BorderRadius.circular(20), // زوايا دائرية
                    boxShadow: [
                      BoxShadow(
                        color: Colors.black.withOpacity(0.2),
                        blurRadius: 10.
```

Figure 34: Signup Screen 2/2: Stores the user information in Firebase Authentication and Firestore. Provides error feedback for invalid inputs.

0.6.3.4 Firebase



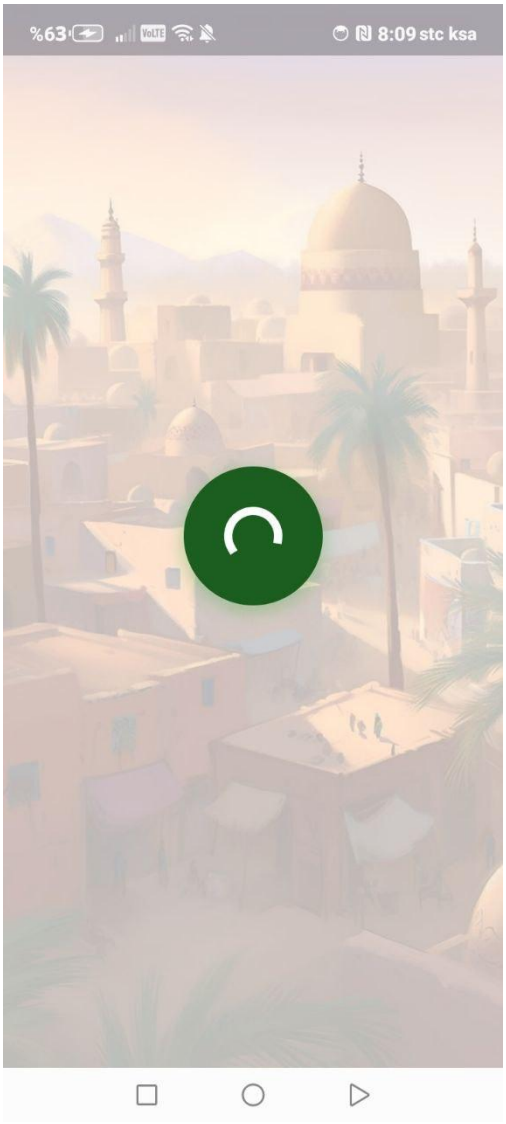
```
firebase_options.dart
lib > firebase_options.dart > ...
// File generated by FlutterFire CLI.
// ignore_for_file: type=lint
import 'package:firebase_core/firebase_core.dart' show FirebaseOptions;
import 'package:flutter/foundation.dart'
    show defaultTargetPlatform, kIsWeb, TargetPlatform;

/// Default [FirebaseOptions] for use with your Firebase apps.
///
/// Example:
/// ```dart
/// import 'firebase_options.dart';
/// // ...
/// await Firebase.initializeApp(
///   options: DefaultFirebaseOptions.currentPlatform,
/// );
/// ```
class DefaultFirebaseOptions {
  static FirebaseOptions get currentPlatform {
    if (kIsWeb) {
      return web;
    }
    switch (defaultTargetPlatform) {
      case TargetPlatform.android:
        return android;
      case TargetPlatform.macOS:
        throw UnsupportedError(
          'DefaultFirebaseOptions have not been configured for macos - '
          'you can reconfigure this by running the FlutterFire CLI again.',
        );
```

Figure 35: Firebase Implementation 1/2: Configures Firebase settings for the app. Determines platform-specific Firebase configurations (e.g., Android or Web).

```
      case TargetPlatform.windows:
        throw UnsupportedError(
          'DefaultFirebaseOptions have not been configured for windows - '
          'you can reconfigure this by running the FlutterFire CLI again.',
        );
      case TargetPlatform.linux:
        throw UnsupportedError(
          'DefaultFirebaseOptions have not been configured for linux - '
          'you can reconfigure this by running the FlutterFire CLI again.',
        );
      default:
        throw UnsupportedError(
          'DefaultFirebaseOptions are not supported for this platform.',
        );
    }
  }

  static const FirebaseOptions android = FirebaseOptions(
    apiKey: 'AIzaSyC72K-GRYnObTb-YzDA0bvBnAxXeowguwI',
    appId: '1:445891354343:android:67bc7db6a285cd7b194c42',
    messagingSenderId: '445891354343',
    projectId: 'saudigard-403d8',
    storageBucket: 'saudigard-403d8.firebasestorage.app',
  );
  static const FirebaseOptions web = FirebaseOptions(
    apiKey: "AIzaSyD7vknnFKL1Cgu0_wmFwFpmzeb631oy-uI",
    authDomain: "saudigard-403d8.firebaseapp.com",
```

Figure 36: Firebase Implementation 2/2: Provides boilerplate code to initialize Firebase based on the target platform. Enables Firebase services like authentication and database management.

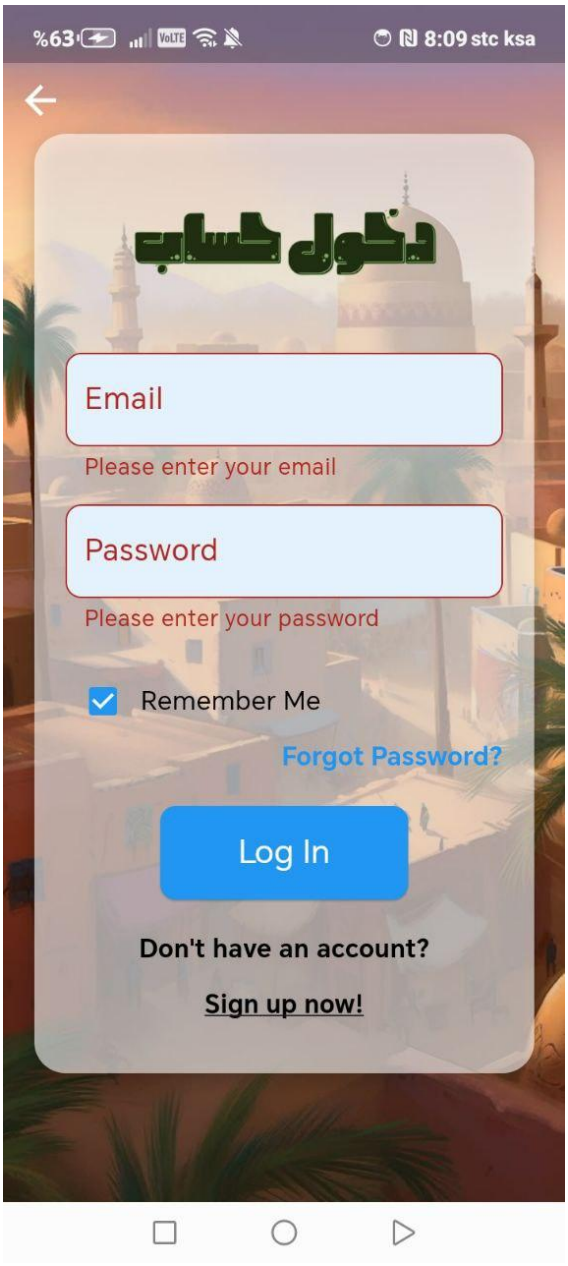
0.6.4 Sprint 1 Interfaces



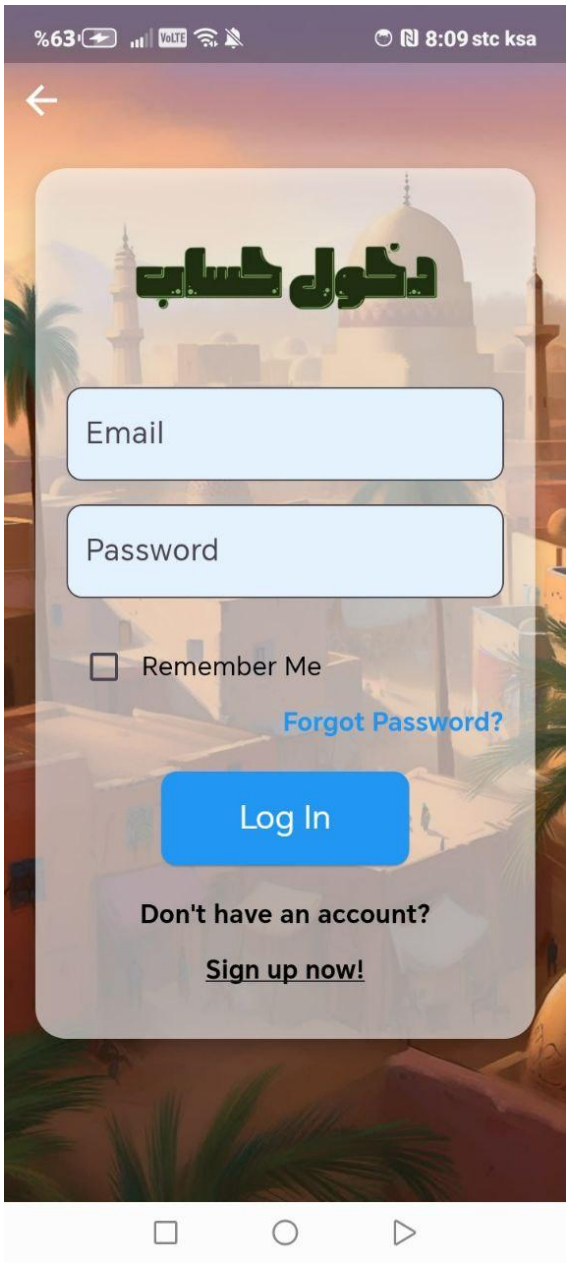
(a) Splash Screen - Welcoming the user with an Arabian-inspired cityscape and a loading indicator.

(b) Home Screen - Displays "" and introduces cybersecurity awareness with strong passwords and 2FA.

Figure 37: Sprint 1 Interfaces: Splash Screen and Home Screen.

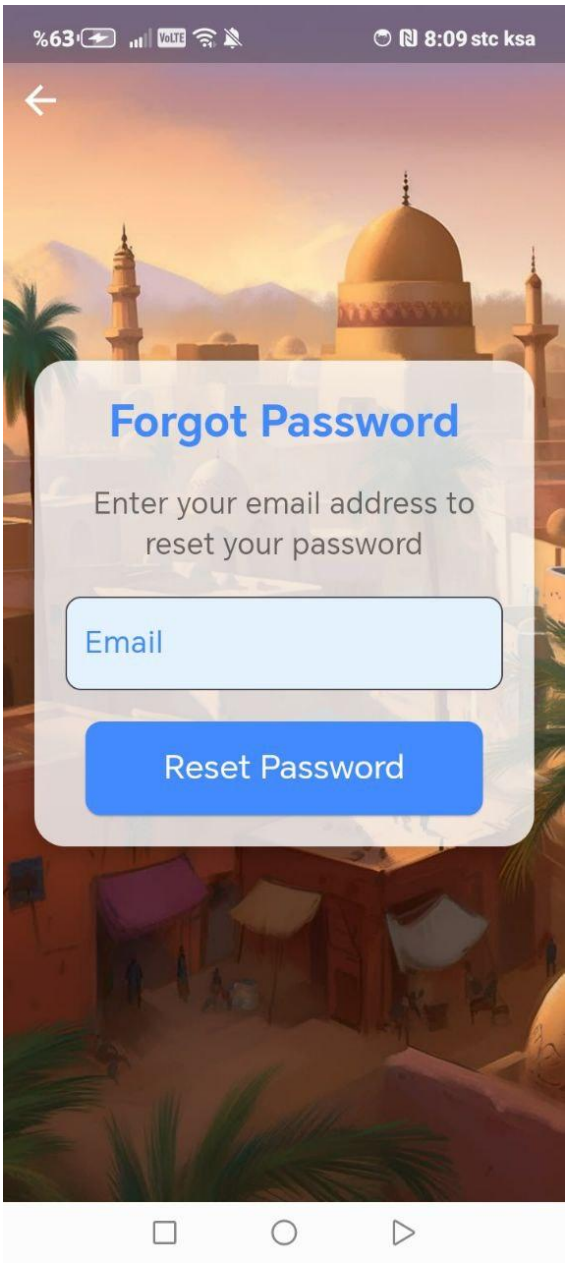


(a) Login Screen - A login form where users can enter their email and password to access their accounts. The design includes a "Remember Me" checkbox and a "Forgot Password" link for account recovery.

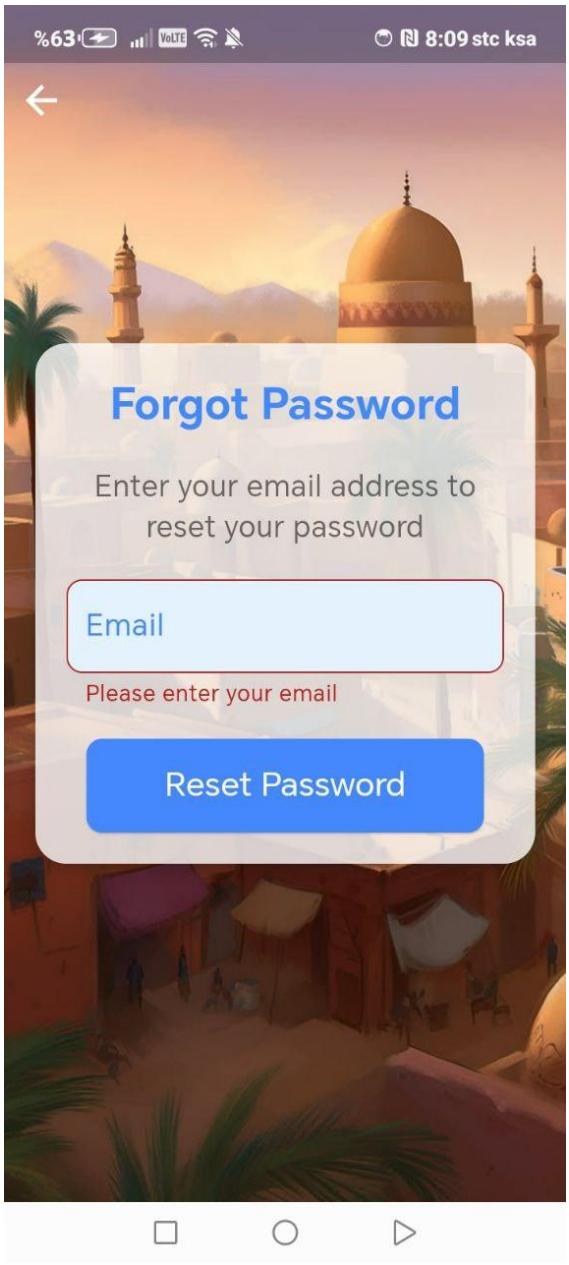


(b) Invalid Login - A login form that highlights errors in red, indicating that the email or password entered is incorrect. Prompts the user to re-enter valid credentials.

Figure 38: Sprint 1 Interfaces: Login screen and invalid login screen.

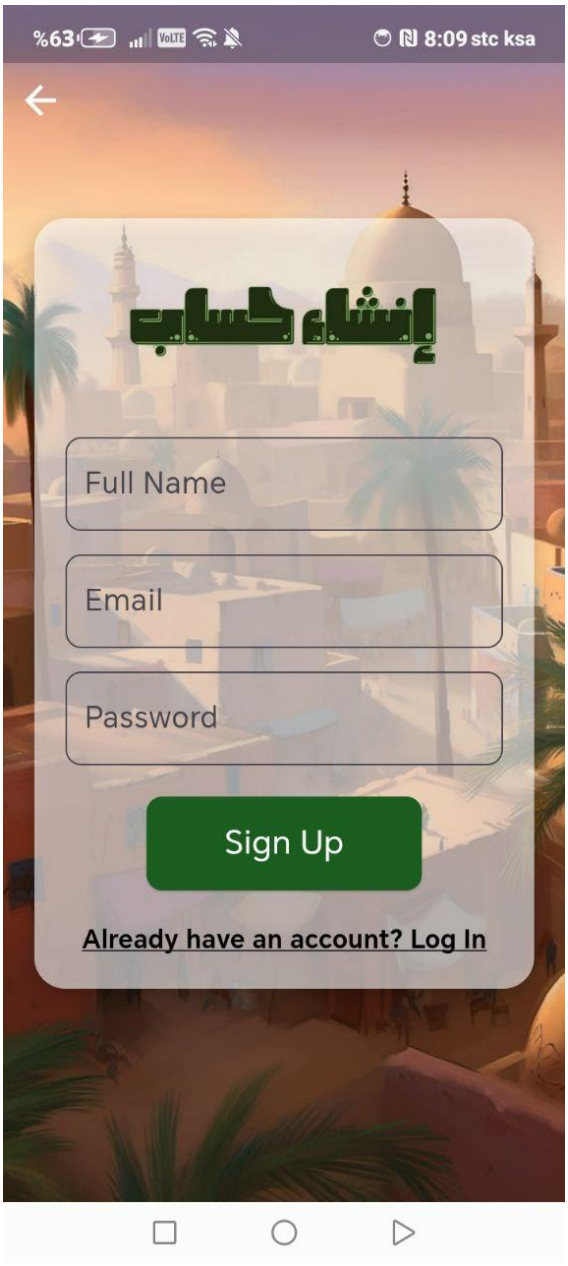


(a) Forgot Password - A simple screen allowing users to reset their password by entering their email address. Includes a “Reset Password” button for initiating the recovery process.

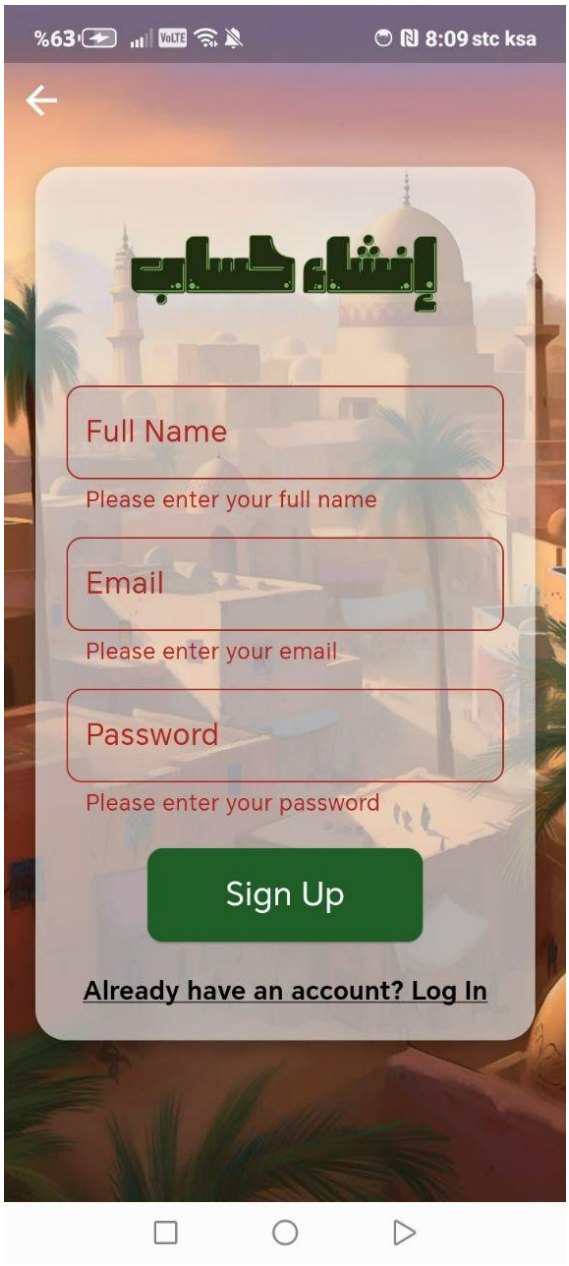


(b) Invalid Forgot Password - A version of the ”Forgot Password” screen with an error prompt in red, notifying the user to enter a valid email address.

Figure 39: Sprint 1 Interfaces: Forgot password and invalid forgot password screens.



(a) Sign-Up Screen - A registration form where new users can create an account by entering their full name, email, and password. Includes a "Sign Up" button and a link to log in if they already have an account.



(b) Invalid Sign-Up - The registration form highlights fields in red when incomplete or incorrect information is provided. Prompts the user to enter valid data in all fields before submitting.

Figure 40: Sprint 1 Interfaces: Sign-up screen and invalid sign-up screen.

0.6.5 Testing

0.6.5.1 Unit Testing

Test Case ID	Test Case Description	Test Steps and Data	Expected Results	Actual Results	Pass/Fail
TC01	Verify user can register with valid inputs	1- Enter full name and email 2- Enter password and confirm 3- Submit. **Data:** Name: John Doe, Email: john@example.com	Registration is successful, and the user is redirected to the login page.	Registration completed successfully, and the user was redirected to the login page.	Pass
TC02	Verify password strength validation works	1- Enter weak password 2- Submit. **Data:** Password: 1234	An error message is displayed: "Password is too weak."	Error message displayed correctly: "Password is too weak."	Pass
TC03	Verify the 2FA code generation	1- Log in 2- Enter code sent via email 3- Submit. **Data:** Code: 987654	Login completes successfully, and the user is redirected to the dashboard.	Login successful, and user was redirected to the dashboard.	Pass

Table 4: Unit Testing Table with Actual Results

0.6.5.2 Acceptance Testing

Test Case No.	Test Case Description	Test Steps	Expected Results	Actual Results	Pass/Fail
TC1	Login with valid credentials	1- Open the website 2- Enter email and password 3- Click login 4- Enter valid 2FA code.	The user should log in successfully and access the game dashboard.	User successfully logged in and accessed the dashboard.	Pass
TC2	Register with valid inputs	1- Open the website 2- Fill in full name, email, and password 3- Confirm password 4- Click "Register".	The user should register successfully and be redirected to the login page.	User registered successfully and redirected to the login page.	Pass
TC3	Complete a stage successfully	1- Log in to the game 2- Complete all tasks for a specific stage 3- Submit all achievements.	The stage should be marked as complete, and the next stage should unlock.	Stage marked as complete, and next stage unlocked successfully.	Pass
TC4	Test password reset process	1- Open the "Forgot Password" page 2- Enter registered email 3- Submit request 4- Reset password via email link.	A reset email is sent, and the user should reset their password successfully.	Password reset email received and password updated successfully.	Pass
TC5	Verify rewards allocation after completion	1- Complete all tasks for a stage 2- Check the rewards screen for unlocked rewards.	The user should see the unlocked rewards.	Unlocked rewards displayed successfully after stage completion.	Pass

Table 5: Acceptance Testing (TC1 to TC5)

0.6.5.3 Integration Testing

Test Case No.	Scenario	Status
1	Verify UI Rendering → Load the 'LoginRegisterScreen' → Verify that the logo image is visible → Verify the welcome text is displayed → Verify the two buttons ('Login' and 'Register') are visible → The logo and the text should be visible on the screen → The buttons for "Login" and "Register" should be present. **Objective:** Ensure the UI is rendered correctly with the logo, text, and buttons.	Success
2	Verify Login Button Navigation → Tap the "Login" button → Wait for the navigation to complete → Verify that the 'LoginScreen' is shown → After pressing the "Login" button, the app should navigate to the 'LoginScreen'. **Objective:** Test that pressing the "Login" button navigates to the 'LoginScreen'.	Success
3	Verify Register Button Navigation → Tap the "Register" button → Wait for the navigation to complete → Verify that the 'RegisterScreen' is shown → After pressing the "Register" button, the app should navigate to the 'RegisterScreen'. **Objective:** Test that pressing the "Register" button navigates to the 'RegisterScreen'.	Success
4	Verify UI Rendering → Load the 'ForgotPasswordScreen' → Verify that the background image is displayed → Verify that the title "Forgot Password" is visible → Verify that the email input field is displayed → Verify that the "Reset Password" button is displayed → The background image should be visible → The title "Forgot Password" and description should be displayed → The email input field and "Reset Password" button should be visible. **Objective:** Ensure that the 'ForgotPasswordScreen' renders correctly with the expected UI components.	Success
5	Verify Back Button Navigation → Tap the back arrow button in the top left → Verify that the app navigates back to the previous screen → Tapping the back arrow should navigate back to the previous screen. **Objective:** Test that pressing the back button (arrow icon) navigates back to the previous screen.	Success
6	Verify Empty Email Validation → Leave the email field empty → Tap the "Reset Password" button → Verify that a validation message appears → If the email field is empty, the app should show a validation message. **Objective:** Test that the form correctly validates an empty email field.	Success

7	Verify UI Rendering → Load the 'LoginScreen' → Verify that the background is displayed → Verify that the "Email" and "Password" input fields are visible → Verify that the "Log In" button is visible → Verify that the "Remember Me" checkbox and "Forgot Password?" text are present → Verify that the "Don't have an account?" text and "Sign up now!" button are visible → The background image should be visible → The email and password fields, login button, remember me checkbox, and other relevant components should be visible. **Objective:** Ensure that the 'LoginScreen' UI components are rendered correctly.	Success
8	Verify Back Button Navigation → Tap the back arrow button in the top left → Verify that the app navigates back to the previous screen → Tapping the back button should navigate back to the previous screen. **Objective:** Test that pressing the back button navigates to the previous screen.	Success
9	Verify Empty Email Validation → Leave the email field empty → Tap the "Log In" button → Verify that a validation message appears. **Objective:** Test that the form correctly validates an empty email field.	Success
10	Successful Registration → Open the RegisterScreen → Enter valid input into the Full Name, Email, and Password fields → Tap the Sign Up button → Verify that the user is created in Firebase Authentication (check Firebase console), the user's details are stored in Firestore, and a success message is displayed via a SnackBar → The registration process is successful, and the user can navigate to the login screen. **Objective:** Ensure the user can successfully register and their data is saved to Firebase Authentication and Firestore.	Success
11	Invalid Email Format → Open the RegisterScreen → Enter an invalid email format → Enter any valid Full Name and Password → Tap the Sign Up button → Verify that the app shows an error message → The registration form is not submitted, and the error message appears next to the email field. **Objective:** Ensure the user receives an error message when entering an invalid email format.	Success
12	Short Password → Open the RegisterScreen → Enter a valid Full Name and Email → Enter a short password → Tap the Sign Up button → Verify that the app shows an error message → The registration form is not submitted, and the error message appears next to the password field. **Objective:** Ensure the app prevents registration if the password is too short (less than 6 characters).	Success

Table 6: Integration Testing

0.6.5.4 System Testing

Test Case No.	Scenario	Status
1	Navigating to Login Screen → Open the LoginRegisterScreen → Verify that the Login button is visible and labeled "Login" → Tap the Login button → Ensure that the app navigates to the LoginScreen → Verify the title or key elements of the LoginScreen (e.g., email and password fields, login button) → Confirm the LoginScreen is displayed after tapping the Login button. **Objective:** Ensure that when the user presses the Login button, the app navigates to the LoginScreen.	Pass
2	Navigating to Register Screen → Open the LoginRegisterScreen → Verify that the Register button is visible and labeled "Register" → Tap the Register button → Ensure that the app navigates to the RegisterScreen → Verify the title or key elements of the RegisterScreen (e.g., full name, email, and password fields) → Confirm the RegisterScreen is displayed after tapping the Register button. **Objective:** Ensure that when the user presses the Register button, the app navigates to the RegisterScreen.	Pass
3	UI Elements Display → Open the LoginRegisterScreen → Verify that the background image (‘wall.jpg‘) is properly displayed and covers the full screen → Verify that the logo image is displayed → Verify that the text below the logo is visible and properly aligned → Verify that the Login and Register buttons are visible and labeled correctly. **Objective:** Ensure all UI elements are properly displayed, including the background image, logo, and buttons.	Pass
4	Valid Email Input → Enter a valid email → Press the "Reset Password" button → Observe the SnackBar showing a success message → Confirm the user is returned to the previous screen. **Objective:** A user enters a valid email address and clicks the "Reset Password" button.	Pass
5	Invalid Email Format → Enter an invalid email → Press the "Reset Password" button → Observe the validation error message under the email field. **Objective:** A user enters an invalid email format and tries to submit the form.	Pass
6	Empty Email Field → Leave the email field empty → Press the "Reset Password" button → Observe the validation error message → Confirm the password reset request is not sent. **Objective:** A user tries to submit the form without entering any email address.	Pass

7	Successful Registration → Enter a valid full name → Enter a valid email → Enter a valid password → Press the "Sign Up" button → Observe the success message and verify data in Firebase. **Objective:** A user enters valid information in all fields and successfully registers.	Pass
8	Empty Fields → Leave all fields empty → Press the "Sign Up" button → Observe the validation error messages for each field. **Objective:** The user clicks "Sign Up" without filling in any fields.	Pass
9	Invalid Email Format → Enter a valid full name → Enter an invalid email → Enter a valid password → Press the "Sign Up" button → Observe the validation error for the email field. **Objective:** The user enters an invalid email address.	Pass
10	Short Password → Enter a valid full name and email → Enter a short password → Press the "Sign Up" button → Observe the validation error for the password field. **Objective:** The user enters a password shorter than 6 characters.	Pass

Table 7: System Testing

0.6.6 Conclusion

This chapter covers the Flutter implementation and initial testing completed during Sprint 1 of the project. The main goal of this sprint was to develop the core features of the interactive game, including the splash screen, login, and registration functionalities, while ensuring a smooth and engaging user experience. Basic testing confirmed that these components work as intended.

We used Dart with Flutter for cross-platform development, Visual Studio Code for coding, and Android Studio for testing. Firebase was integrated for backend services like authentication and data management.

Looking ahead, we plan to transition to Unity in future sprints to implement the core gameplay elements, leveraging its game development capabilities. The initial testing has provided a solid foundation for further development, and the groundwork laid in Sprint 1 will support more advanced features in subsequent stages.

Bibliography

[1] J. Howarth, "50+ password statistics: The state of password security in 2024," *Exploding Topics*, 2024.

[2] Norton, "139 password statistics to help you stay safe in 2024," 2024.

[3] N. Dekker, "Two-factor authentication statistics: First line of defence," *Eftsure*, 2022.

[4] N. C. Authority, "National cybersecurity strategy," 2023.

[5] N. Alhalafi and P. Veeraraghavan, "Cybersecurity policy framework in saudi arabia: Literature review," *Frontiers in Computer Science*, vol. 3, p. 736874, 2021.

[6] B. Aljedaani, A. Ahmad, M. Zahedi, and M. A. Babar, "End-users' knowledge and perception about security of mobile health apps: a case study with two saudi arabian mhealth providers," *arXiv preprint arXiv:2101.10412*, 2021.

[7] K. Kovalan, S. Z. Omar, L. Tang, J. Bolong, R. Abdullah, A. H. A. Ghazali, and M. A. Pitchan, "A systematic literature review of the types of authentication safety practices among internet users," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, 2021.

[8] A. M. Mostafa, M. Ezz, M. K. Elbashir, M. Alruily, E. Hamouda, M. Alsarhani, and W. Said, "Strengthening cloud security: an innovative multi-factor multi-layer authentication framework for cloud user authentication," *Applied Sciences*, vol. 13, no. 19, p. 10871, 2023.

[9] E. Al Qahtani, *Evaluating Risk Appeal Approaches Based on PMT Towards Making Secure Decisions*. PhD thesis, The University of North Carolina at Charlotte, 2023.

[10] H. Albazar, A. Abdel-Wahab, M. Alshar'e, and A. Abualkishik, "An adaptive two-factor authentication scheme based on the usage of schnorr signcryption algorithm," *Informatica*, vol. 47, no. 5, 2023.

[11] M. A. M. Ataelfadiel, "E-authentication system using qr code & otp," *International Research Journal of Innovations in Engineering and Technology*, vol. 6, no. 9, p. 75, 2022.

[12] E. Al Qahtani, L. Sahoo, Y. Javed, and M. Shehab, "" why would someone hack me out of thousands of students": Video presenter's impact on motivating users to adopt 2fa," in *Proceedings of the 27th ACM on Symposium on Access Control Models and Technologies*, pp. 139–150, 2022.

[13] F. S. Alshahrani and M. Abdullah, "Graphical-based password for user authentication in internet of things," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 28, no. 2, pp. 1139–1146, 2022.

[14] Y. Albayram, J. Liu, and S. Cangonj, "Comparing the effectiveness of text-based and video-based delivery in motivating users to adopt a password manager," in *Proceedings of the 2021 European Symposium on Usable Security*, pp. 89–104, 2021.

[15] E. Al Qahtani, L. Sahoo, and M. Shehab, "The effectiveness of video messaging campaigns to use 2fa," in *International Conference on Human-Computer Interaction*, pp. 369–390, Springer, 2021.

[16] H. P. Singh and T. S. Alshammari, "An institutional theory perspective on developing a cyber security legal framework: a case of saudi arabia," *Beijing L. Rev.*, vol. 11, p. 637, 2020.

[17] A. Alzubaidi, "Measuring the level of cyber-security awareness for cybercrime in saudi arabia," *Heliyon*, vol. 7, no. 1, 2021.

[18] R. A. Alsharida, B. A. S. Al-rimy, M. Al-Emran, and A. Zainal, "A systematic review of multi perspectives on human cybersecurity behavior," *Technology in society*, vol. 73, p. 102258, 2023.

[19] M. Asare, "Using the theory of planned behavior to determine the condom use behavior among college students," *American journal of health studies*, vol. 30, no. 1, p. 43, 2015.

[20] D. Marikyan and S. Papagiannidis, "Protection motivation theory: A review," *Theory-Hub Book: This handbook is based on the online theory resource: TheoryHub*, pp. 78–93, 2023.

[21] D. Carpenter, D. K. Young, P. Barrett, and A. J. McLeod, "Refining technology threat avoidance theory," *Communications of the Association for Information Systems*, vol. 44, 2019.