COMSATS University Islamabad,
Abbottabad Campus

SOFTWARE REQUIREMENTS SPECIFICATION

(SRS DOCUMENT)

for

Treasure Hunt Adventure

By

RAJA FATASH ABBASI **CIIT/SP22-BSE-127/ATD**

HUZAIFA TANVEER **CIIT/SP22-BSE-140/ATD**

Supervisor

Mr. Ahsan Khan

Bachelor of Science in Computer Science (2022-2026)

Table of Contents

Application Evaluation History

Comments (by committee) ***include the ones given at scope time both in doc and presentation**	**Action Taken**

Supervised by

<Supervisor's Name>

Signature_____

Introduction

Treasure Hunt Adventures is an innovative mobile application that blends augmented reality (AR) with puzzle-based adventure gaming to create an engaging and immersive experience for users. Built using Unity and AR Foundation, the game transforms real-world environments into interactive playgrounds where players follow GPS-based clues, interact with AR objects, and solve a variety of puzzles to discover hidden treasures. Unlike traditional AR games that often rely on simple object placement and minimal interaction, this game introduces cognitive challenges such as riddles, memory sequences, symbol-matching, and logical puzzles to keep users mentally engaged while encouraging physical exploration. With each successful task, players unlock rewards and progress through increasingly challenging levels, ultimately competing for top rankings on a global leaderboard powered by Firebase.

The system supports Android platforms and leverages essential smartphone features such as the rear camera, GPS module, and motion sensors to ensure dynamic and location-aware gameplay. The core architecture is designed for single-player interaction with future expansion potential for multiplayer modes and monetization features like in-app purchases and sponsored hunts. The project integrates Firebase for user authentication, real-time progress tracking, cloud-based data storage, and leaderboard management, ensuring both secure and scalable performance. *Treasure* Hunt Adventures not only aims to deliver a fun and gamified AR experience but also promotes outdoor activity and critical thinking, making it a unique and value-driven addition to the growing ecosystem of educational and adventure-based mobile games.

Purpose

The primary purpose of Treasure Hunt Adventures is to deliver a unique and immersive mobile gaming experience that combines Augmented Reality (AR), real-world exploration, and puzzle-solving mechanics. Designed for Android smartphones, the game motivates players to step outdoors and navigate real physical environments using GPS guidance, where they encounter virtual objects and hidden clues rendered through AR technology. Each treasure hunt involves a series of puzzles that must be solved using logic, observation, and interaction with digital elements overlaid onto the physical world. This fusion of mental challenge and physical activity not only entertains but also promotes critical thinking and healthy movement.

The game's core objective is to provide a gamified exploration platform that encourages users to engage with their surroundings in a meaningful way. By integrating Firebase-based authentication, cloud data synchronization, and a leaderboard system, *Treasure Hunt Adventures* also fosters community competition and progress tracking. Players can earn rewards, collect treasures, and climb the global leaderboard, creating an incentive to return and participate regularly. This experience supports not only casual players seeking fun but also puzzle enthusiasts and tech-savvy users interested in AR interaction. Overall, the game is designed to make technology-driven outdoor engagement both entertaining and intellectually stimulating.

Scope

Treasure Hunt Adventures is a mobile-based augmented reality (AR) game developed for Android smartphones that transforms the user's physical environment into an interactive treasure-hunting experience. The game's core concept revolves around encouraging users to explore real-world locations using their mobile device to discover hidden virtual objects, solve contextual puzzles, and complete treasure quests. By merging location-based navigation with AR interactions, the game aims to enhance user engagement, promote physical activity, and deliver an innovative form of entertainment that blends digital content with reality.

The scope of the project encompasses the complete development of a standalone AR gaming application, including its visual interface, game logic, and environment interaction. It covers all necessary gameplay stages—from account creation to session completion—and ensures that the player's journey is intuitive, immersive, and continuous. The application is intended for individual players but is designed in a scalable way that allows future enhancements, such as multiplayer functionality, seasonal challenges, and region-specific content. The project does not cover development for iOS platforms in its initial phase and focuses solely on a single-user experience using Unity and AR Foundation technologies.

Overall description

The Following shows the overall description of the system.

Product perspective

Treasure Hunt Adventures is a standalone mobile application developed from the ground up, leveraging existing technologies such as Unity, AR Foundation, and Firebase. It is not an extension of an existing product or system but rather a novel implementation that combines elements of AR navigation, puzzle-solving, and GPS-based exploration into a unified game experience. The application falls under the category of single-player AR adventure games, with a unique focus on real-world interaction, cognitive engagement through diverse puzzle types, and competitive motivation via a cloud-based leaderboard.

The product functions independently but integrates with various third-party services for enhanced functionality. Unity serves as the core game engine, offering visual rendering, physics, and scene management. AR Foundation provides cross-platform AR capabilities, enabling surface detection, camera tracking, and object placement in real-world environments. Firebase plays a critical role in managing backend operations, including user authentication, real-time database storage, leaderboard management, and progress tracking. The product is designed to operate on Android smartphones with ARCore support, and it interacts with onboard sensors such as GPS, accelerometer, and gyroscope to enable a context-aware gaming experience. The modular design of the system allows for future enhancements, including multiplayer features, monetization strategies, and compatibility with iOS platforms or AR headsets.

Operating environment

The *Treasure Hunt Adventures* application shall operate on Android smartphones that support ARCore technology. These devices must include a functional rear-facing camera, GPS module, gyroscope, and accelerometer to support the augmented reality and location-based functionalities of the game. The application will run on Android OS version 10.0 or higher.

The development environment includes Unity 2022 or later for game development, AR Foundation for cross-platform AR capabilities, and Android Studio for device testing and deployment. Firebase will be used as the backend platform to manage user authentication, real-time database operations, and cloud-based storage.

The system requires a stable internet connection (Wi-Fi or mobile data) for functionalities such as user login, leaderboard updates, and saving progress to the cloud. The application is designed for outdoor and indoor environments, though performance may vary depending on GPS signal strength, camera quality, and lighting conditions.

OE-1: The application shall run on Android smartphones with ARCore support.
OE-2: The system shall require internet access to connect to Firebase services.
OE-3: The application shall utilize onboard sensors including GPS, camera, gyroscope, and accelerometer for full functionality.
OE-4: The development and testing environments shall include Unity, Firebase Console, and Android Studio.

Design and implementation constraints

DC-1: No Guest Access
The game does not allow unregistered users to play. All players must authenticate via email and OTP using Firebase before accessing any gameplay.

DC-2: No Offline Mode
The game restricts any core functionality when offline. Players cannot start sessions, collect treasure, or sync progress without an active internet connection.

DC-3: No Multiple Concurrent Sessions
A single user cannot start multiple game sessions in parallel. Only one active session is allowed per user until the current one ends.

DC-4: No Fake Location Access
The app disallows the use of mock location apps. Fake GPS coordinates are blocked to maintain fairness in treasure collection.

Requirement identifying technique

The requirements for Treasure Hunt Adventures have been identified using the Use Case technique, which is suitable for interactive, user-driven applications such as mobile games. This technique helps in modeling the system's functional behavior from the perspective of its users and provides a clear understanding of how the system will interact with external entities.

Use cases describe the primary interactions between the player and the system, such as signing in, navigating using AR, solving puzzles, collecting virtual treasures, and viewing leaderboard standings. These scenarios help uncover detailed system functionalities, user actions, and expected outcomes.

Additionally, the use case approach supports early validation of system requirements by stakeholders, ensuring that all critical interactions are captured before design and development begin. By breaking down the system into specific user-goal-driven scenarios, developers can prioritize features, identify dependencies, and design intuitive user flows. This also aids in testing and validation, as each use case can be directly translated into test cases to verify that the application behaves as expected in real-world usage. Ultimately, the use case model serves as a bridge between requirement gathering and implementation, promoting clarity, consistency, and user-centric development throughout the lifecycle of Treasure Hunt Adventures.

Use case diagram

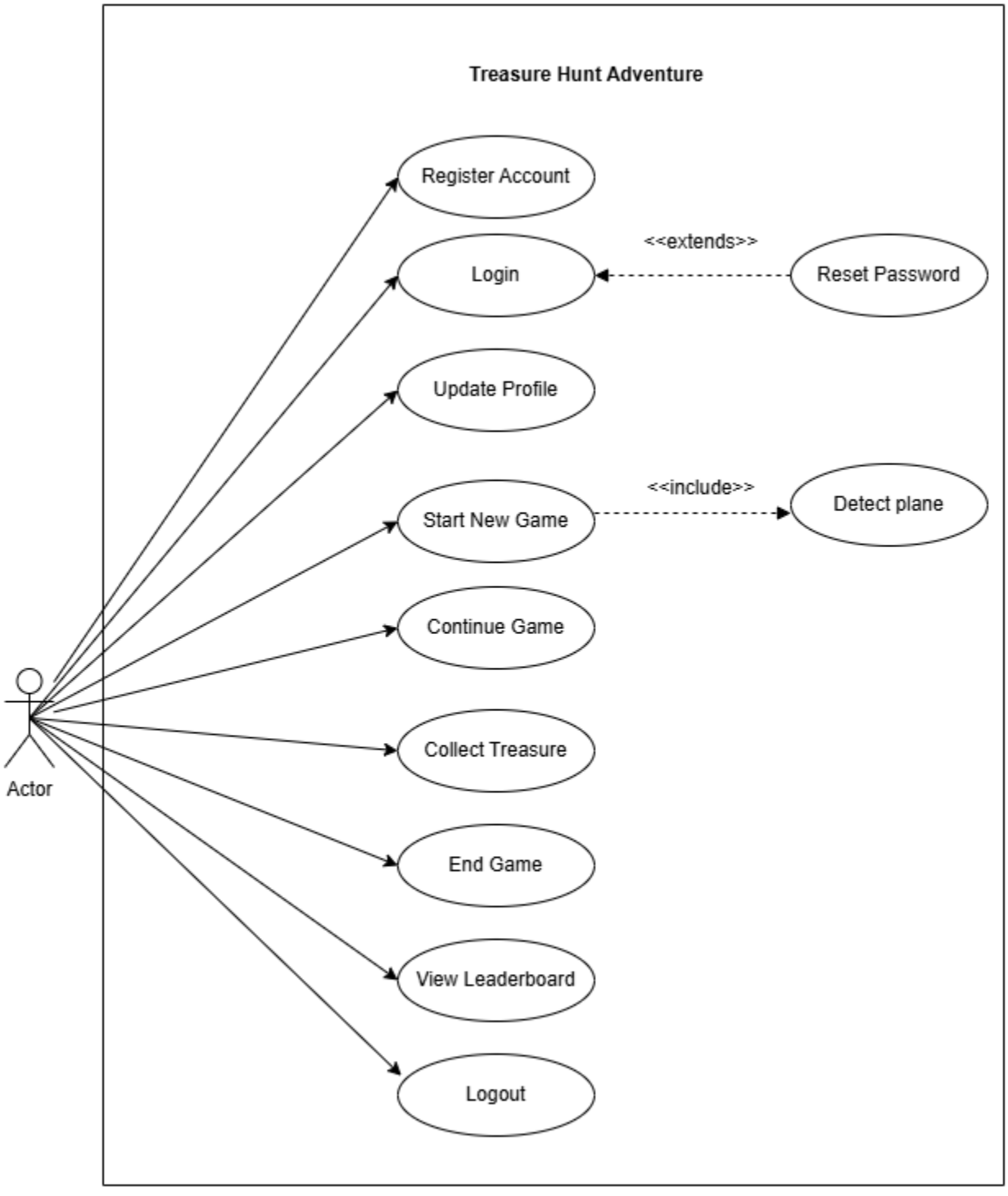


Fig 01. Use Case Diagram for Treasure Hunt Adventure

Use case description

This use case diagram provides a detailed functional overview of the Treasure Hunt Adventure System, showcasing how various actors interact with the system across different modules. The central actor, the Player, engages with all primary features including authentication, AR-based gameplay, profile and inventory management, social sharing, and system-level operations. The diagram groups use cases into six logical categories: Authentication, AR Gaming, Game Management, Social, System, *and* External Services. It highlights how users register, log in, reset passwords, and manage profiles via Firebase Services, while AR interactions such as detecting planes, placing, and collecting treasures rely on GPS and ARCore Services. The system also supports social features like viewing leaderboards and challenging friends, and management features like checking achievements, viewing inventory, and statistics. Backend processes such as data synchronization, notification sending, and analytics tracking are handled by the System actor. Relationships like `<<include>>` and `<<extend>>` are used to show dependencies and optional enhancements, creating a clear, modular view of how system functionalities are structured and interconnected.

Furthermore, the diagram effectively demonstrates the system's modular design by categorizing use cases, which aids in both development and maintenance. The use of external actors such as Firebase Service, GPS Service, and ARCore Service emphasizes the system's reliance on third-party components for critical functionalities like user authentication, location tracking, and augmented reality rendering. The inclusion relationships, such as between "Start AR Session" and "Detect Planes," indicate mandatory sub-processes, while extension relationships, like those in social features, highlight optional or conditional interactions that enhance user experience. This structured approach not only ensures comprehensive coverage of user interactions but also provides a foundation for prioritizing features during implementation and testing phases.

Use Case ID: UC-01 – Login

Table 1.1 Fully dressed UC Login

Field	**Description**
Use Case ID	UC-01
Use Case Name	User Login
Primary Actor	Player
Secondary Actor	Firebase
Description	The player logs into the game using email/password credentials.
Trigger	Player opens the app and selects the login option.
Preconditions	PRE-1. Player has a registered account. PRE-2. Internet connection is available.
Postconditions	POST-1. User is logged in. POST-2. Player profile and progress are loaded.
Normal Flow	1. Player opens app 2. Enters credentials 3. System sends request to Firebase 4. Firebase validates credentials 5. Player is logged in and navigated to main menu

Alternative Flows	AF-1. Invalid credentials → Show error and retry option AF-2. Firebase timeout → Show retry
Exceptions	EX-1. No internet → Prompt offline mode or retry EX-2. Firebase error → Show alert

Use Case ID: UC-02 – Logout

Table 1.2 Fully dressed UC Logout

Field	**Description**
Use Case ID	UC-03
Use Case Name	Logout
Primary Actor	Player
Secondary Actor	Firebase
Description	The player logs out from their current session.
Trigger	Player selects "Logout" from settings menu.
Preconditions	PRE-1. Player must be logged in.
Postconditions	POST-1. Session is cleared. POST-2. Returned to login screen.
Normal Flow	1. Player opens settings 2. Selects Logout 3. App clears session data 4. Navigates to login screen
Alternative Flows	None
Exceptions	EX-1. Logout request fails due to backend error → Retry/logout locally

Use Case ID: UC-03 – Reset a Password

Table 1.3 Fully dressed UC Reset Password

Field	**Description**
Use Case ID	UC-03
Use Case Name	Reset Password
Primary Actor	Player
Secondary Actor	Firebase
Description	Allows a player to reset their password through email verification.
Trigger	Player selects 'Forgot Password' on login screen.
Preconditions	PRE-1. Player has an account. PRE-2. Email must be valid.
Postconditions	POST-1. Password reset link sent. POST-2. Player receives instructions.
Normal Flow	1. Player clicks 'Forgot Password'. 2. Enters email. 3. Firebase sends reset link. 4. Player resets password.
Alternative Flows	AF-1. Email not found → Show error. AF-2. Timeout → Retry.
Exceptions	EX-1. Firebase error → Show message. EX-2. No internet → Prompt retry.
Use Case ID	UC-04

Use Case ID: UC-04 - Start New Game

Table 1.4 Fully dressed UC Start New Game

Field	Description
Use Case ID	UC-04
Use Case Name	Start New Game
Primary Actor	Player
Secondary Actor	ARCore Service
Description	Initializes the AR environment for gameplay.
Trigger	Player selects 'Start AR Hunt' option.
Preconditions	PRE-1. AR permissions granted. PRE-2. Device supports AR.
Postconditions	POST-1. AR session active. POST-2. Camera feed initiated.
Normal Flow	1. Player starts AR session. 2. System requests camera access. 3. ARCore initializes AR.
Alternative Flows	AF-1. Permissions denied → Prompt request. AF-2. AR not supported → Show alert.
Exceptions	EX-1. Camera failure → Abort. EX-2. ARCore crash → Retry or exit.

Use Case ID: UC-05 - Update Profile

Table 1.5 Fully dressed UC Update Profile

Field	Description
Use Case ID	UC-05
Use Case Name	Update Profile
Primary Actor	Player
Secondary Actor	Firebase
Description	The player updates profile details such as username or avatar.
Trigger	Player accesses profile settings and modifies details.
Preconditions	PRE-1. Player is logged in.
Postconditions	POST-1. Profile updated in Firebase.
Normal Flow	1. Open profile settings 2. Modify details 3. Save changes 4. Data updated
Alternative Flows	AF-1. Empty fields → Prompt to fill in required info
Exceptions	EX-1. Update fails → Show retry option
Use Case ID	UC-04

Use Case ID: UC-06 – Continue Game

Table 1.6 Fully dressed UC Continue Game

Field	Description
Use Case ID	UC-06
Use Case Name	Continue Game

Primary Actor	Player
Secondary Actor	Firebase
Description	The player resumes a previously started game session.
Trigger	Player selects 'Continue Game' from the menu.
Preconditions	PRE-1. Player must be logged in. PRE-2. Saved progress must exist.
Postconditions	POST-1. Previous game session is restored.
Normal Flow	1. Player selects 'Continue Game'. 2. System loads saved data. 3. Resumes AR session at last checkpoint.
Alternative Flows	AF-1. No saved data → Prompt player to start new game.
Exceptions	EX-1. Firebase sync failure → Show error and retry option.

Use Case ID: UC-07 - Solve a Puzzle

Table 1.7 Fully dressed UC Solve a Puzzle

Field	**Description**
Use Case ID	UC-07
Use Case Name	Solve a Puzzle
Primary Actor	Player
Secondary Actor	Firebase
Description	A player explores a real-world location using GPS and AR. When arriving at the correct location, a puzzle appears in AR. The player solves it to receive a reward.
Trigger	Player reaches a geolocation-linked AR marker and opens the app to initiate puzzle display.
Preconditions	PRE-1. Player must be logged in. PRE-2. GPS and AR permissions must be granted. PRE-3. AR surface must be detected.
Postconditions	POST-1. Puzzle marked as solved. POST-2. Reward added to player profile. POST-3. Firebase saves player progress.
Normal Flow	1. Player signs in. 2. Game shows nearby puzzle locations. 3. Player navigates to location. 4. Puzzle appears. 5. Player solves puzzle. 6. Reward issued. 7. Leaderboard updated.
Alternative Flows	AF-1. Puzzle Failed → Retry prompt. AF-2. AR Surface Not Found → Prompt to rescan or move.
Exceptions	EX-1. Internet Disconnected → Save locally, sync later. EX-2. Firebase Error → Show retry option.

Use Case ID: UC-08 – Collect Treasure

Table 1.8 Fully dressed UC Collect Treasure

Field	Description
Use Case ID	UC-08
Use Case Name	Collect Treasure
Primary Actor	Player
Secondary Actor	Firebase
Description	The player collects a virtual treasure during an AR session.
Trigger	Player interacts with AR treasure object.
Preconditions	PRE-1. AR session must be active. PRE-2. Player must be near the treasure.
Postconditions	POST-1. Treasure is marked as collected. POST-2. Player's inventory is updated.
Normal Flow	1. Player finds a treasure. 2. Taps on treasure in AR. 3. System updates inventory and gives reward.
Alternative Flows	AF-1. Treasure already collected → Show message.
Exceptions	EX-1. Sync failure with Firebase → Store offline and sync later.

Use Case ID: UC-09 – End Game

Table 1.9 Fully dressed UC End Game

Field	Description
Use Case ID	UC-09
Use Case Name	End Game
Primary Actor	Player
Secondary Actor	Firebase
Description	The player ends the current game session.
Trigger	Player selects 'End Game' in menu or completes all objectives.
Preconditions	PRE-1. Player must be in a game session.
Postconditions	POST-1. Game state saved. POST-2. Player is redirected to menu.
Normal Flow	1. Player pauses or completes the game. 2. Selects 'End Game'. 3. System saves progress and returns to menu.
Alternative Flows	None
Exceptions	EX-1. Save failure → Show error and offer retry.

Use Case ID: UC-10 – View Leaderboard

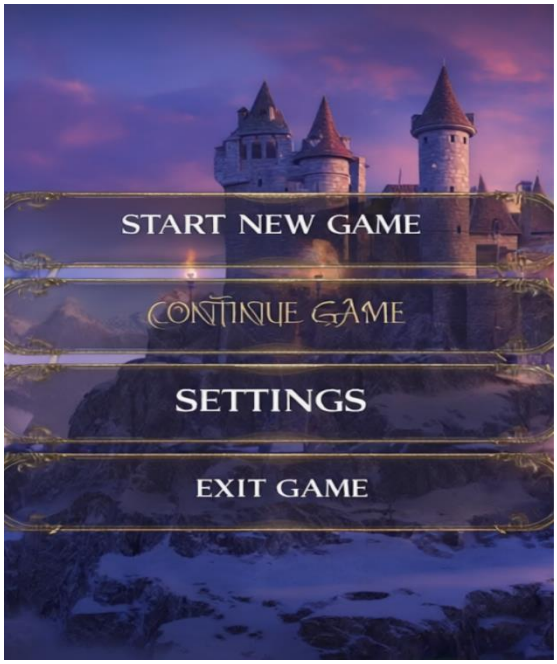
Table 1.10 Fully dressed UC View Leaderboard

Field	Description
Use Case ID	UC-10
Use Case Name	View Leaderboard
Primary Actor	Player
Secondary Actor	Firebase
Description	The player views the ranking and scores of other players.
Trigger	Player selects 'Leaderboard' from menu.

Preconditions	PRE-1. Player must be logged in. PRE-2. Internet connection must be available.
Postconditions	POST-1. Leaderboard data is displayed.
Normal Flow	1. Player opens menu. 2. Selects 'Leaderboard'. 3. System fetches and displays player rankings.
Alternative Flows	None
Exceptions	EX-1. Data fetch failure → Show retry option.

Requirement Identification technique:

Storyboarding:




THE PLAYER ENTERS THE THE GAME WORLDTHROUGH THROUGH THE
AR CAMERA, SEACHINGFOR THE FIRST CLUE
MOVE CLOSER TO SCARN
SCAN
0/5 CLUES FOUND


THE PLAYER ENCOUNTERS THEIR FIRST PUZZLE
SOLVE THE RIDDLE OR FIND HIDDEN SYMBOLS


PLAYER FOLLOWS CLUES
TOWARD THE TREASURE


THE PLAYER ENCOUNTERS AN OBST CLE
PLAYER MUST OVERCOME OBSTACLE
TO MOVE FORVARD


VICTORY!
VIEW REWARDS
RESTART
The player finds the tresure. A victtory mesage appsars. , alomg to the
optio wih view their rewards or restat.


LEADERBOARD
RAJA FATASH ABBASI – 30
HUZAIFA TANVEER – 30
RESTART
MAIN MENU

Functional Requirements

This section outlines the functional requirements of the system based on its core features and user interactions. These requirements define what the system must do to support the treasure hunt gameplay, including AR interactions, puzzle logic, rewards, and backend services. Each requirement is written in clear, measurable terms and follows a structured format that includes rationale, source, dependencies, business rules, and priority.

The requirements are identified using the Use Case technique, which is suitable for interactive applications like mobile games. These functional requirements are organized by feature and described in the following detailed table, which includes:

- **Requirement ID**
- **Title**
- **Detailed Requirement Statement**
- **Source**
- **Rationale**
- **Business Rules**
- **Dependencies**
- **Priority**

The table below presents all primary functional requirements for the Treasure Hunt Adventure project in row-to-column format to enhance clarity and traceability.

Table 1 Show the functional requirement template

Requirement ID	FR-01	FR-02	FR-03	FR-04	FR-05	FR-06	FR-07	FR-08
Title	User Authentication	AR Puzzle Activation	Puzzle Solving	Reward Issuance	Leaderboard System	Offline Progress Saving	AR Surface Detection	Puzzle Retry Mechanism
Requirement	The player shall be able to register and sign in using Firebase Authentication.	The system shall activate AR puzzle display when the player reaches a valid GPS location.	The player shall solve puzzles presented in AR and receive feedback.	The system shall issue virtual rewards for correct puzzle completions.	The system shall update/display leaderboards using puzzle/treasure data.	The system shall save game progress locally and sync to Firebase once reconnected.	The system shall detect flat surfaces using ARCore to place virtual content.	The system shall allow retrying failed puzzles with optional hints.
Rationale	To save and restore user data securely.	Ensures real-world integration and immersion.	Core interactive gameplay mechanic.	Enhances player motivation and progression.	Adds competition and replay value.	Prevents data loss during internet outages.	Required for correct object placement in real world.	Ensures player engagement and prevents frustration.
Business Rule	None	BR-01: Puzzle unlocks only at valid GPS spot	None	BR-02: Each reward linked to unique puzzle ID	None	None	None	None
Dependencies	None	FR-01	FR-02	FR-03	FR-04	FR-01	FR-02	FR-03
Priority	High	High	High	Medium	Medium	Medium	High	Medium

Non Functional Requirements

1. Scalability

- Support up to 10,000 concurrent users via Firebase backend.

2. Reliability

- Ensure 99.5% uptime.
- Auto-save progress locally if disconnected, and sync when reconnected.

3. Security

- Protect all access via Firebase Authentication.
- Two Factor Authentication.

4. Usability

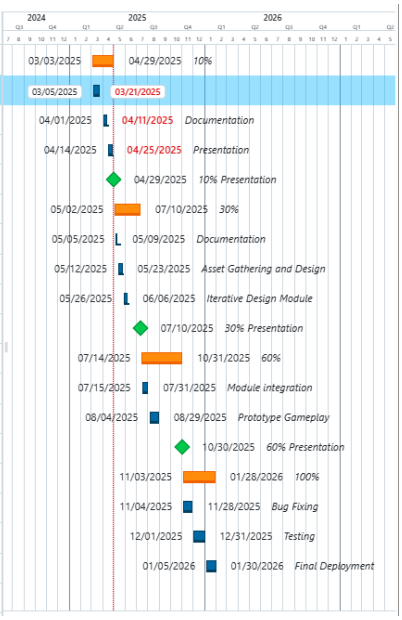
- Designed for users aged 10+ with simple 1–2 tap interactions.

5. Data Integrity

- Ensure safe syncing and backup of game progress using Firebase Firestore.

Gannt Chart

ID	TYPE	SUBJECT	STATUS	START DATE	FINISH DATE	DURATION
37	SUMMARY TASK	10%	New	03/03/2025	04/29/2025	42 days
38	TASK	Project Approval	New	03/05/2025	03/21/2025	13 days
39	TASK	Documentation	New	04/01/2025	04/11/2025	9 days
40	TASK	Presentation	New	04/14/2025	04/25/2025	10 days
41	MILESTONE	10% Presentation	New	04/29/2025	04/29/2025	
42	SUMMARY TASK	30%	New	05/02/2025	07/10/2025	50 days
43	TASK	Documentation	New	05/05/2025	05/09/2025	5 days
44	TASK	Asset Gathering and Design	New	05/12/2025	05/23/2025	10 days
45	TASK	Iterative Design Module	New	05/26/2025	06/06/2025	10 days
46	MILESTONE	30% Presentation	New	07/10/2025	07/10/2025	
47	SUMMARY TASK	60%	New	07/14/2025	10/31/2025	80 days
48	TASK	Module integration	New	07/15/2025	07/31/2025	13 days
49	TASK	Prototype Gameplay	New	08/04/2025	08/29/2025	20 days
50	MILESTONE	60% Presentation	New	10/30/2025	10/30/2025	
51	SUMMARY TASK	100%	New	11/03/2025	01/28/2026	63 days
52	TASK	Bug Fixing	New	11/04/2025	11/28/2025	19 days
53	TASK	Testing	New	12/01/2025	12/31/2025	23 days
54	TASK	Final Deployment	New	01/05/2026	01/30/2026	20 days
55	MILESTONE	100% Presntation	New	-	-	



References

- Feasibility Report: Treasure Hunt Adventure
- Final Year Proposal Document
- Unity AR Foundation Docs – https://unity.com/solutions/ar
- Firebase Docs – https://firebase.google.com/docs
- Google ARCore – https://developers.google.com/ar