

Syrian Private University
Faculty of Computer & Information
Engineering.
Department of Software & Information
Systems Engineering.



Title

**Language game for
children**

لعبة تعلم لغة للأطفال

Prepare By

هاجر الزعبي

سامح يحيى عرابي

Supervised by

المهندس: شادي بلدي

Academic Year 2026\2025

Supervisor Certification

I certify that the preparation of the project entitled

**.....
prepared by**

**.....
was made under my supervision at Department of Computer and
Information Engineering – Faculty of Computer & Communication
Engineering in partial fulfilment of the Requirements for the
Degree**

**of Bachelor of Software and information System Engineering.
Academic Rank:..... Name:.....**

Signature:

Date:

Abstract

This educational platform offers an engaging and motivating learning environment for children aged 8–14, significantly increasing their interaction and enthusiasm for learning while overcoming the boredom associated with traditional methods. It focuses on building a strong early foundation in the English language, particularly in vocabulary, pronunciation, and listening skills — essential competencies in the initial educational stages.

The system allows content administrators to easily add and update levels and games through an intuitive control panel, ensuring the content remains current and aligned with modern educational curricula. Parents and administrators benefit from detailed progress tracking features that identify strengths and weaknesses, facilitating personalized guidance. Finally, the platform provides a flexible, accessible solution across smart devices, expanding high-quality English language education to learners regardless of their geographical location.

الملخص

يُوفر هذا المنصة التعليمية بيئة تعلم محفزة وممتعة للأطفال من سن 8 إلى 14 عاماً، مما يرفع بشكل كبير من تفاعلهم ورغبتهم في التعلم ويُتغلب على الملل المرتبط بالطرق التقليدية. يركز المشروع على بناء أساس قوي ومبكر في اللغة الإنجليزية، خاصة في مجالات المفردات والنطق والاستماع، وهي مهارات أساسية في المراحل التعليمية الأولى. يتيح النظام لمدير المحتوى إضافة وتحديث المراحل والألعاب بسهولة من خلال لوحة تحكم مرنة، مما يضمن بقاء المحتوى محدثاً ومتواافقاً مع المناهج التعليمية الحديثة. كما تُمكّن لوحة التحكم الآباء والمشرفين من متابعة تقدم الطفل بدقة وتحديد نقاط القوة والضعف لتقديم توجيه شخصي فعال. وأخيراً، يقدم الحل تعليمًا مرناً ومتاحًا عبر مختلف الأجهزة الذكية، مما يوسع نطاق الوصول إلى تعلم اللغة الإنجليزية عالي الجودة بغض النظر عن الموضع الجغرافي.

Table of Contents

Chapter / Section	Title / Subtitle	Page
Chapter 1	Introduction and Definition of the Project	1
1	Solve The Problem	2
2	Aim of the Project	2
3	Documentation Description	3
4	Project's Management	3
4.1	Human Resource	3
4.2	Programming Resource	3
4.2.1	Bootstrap 5 (Front-End-Framework)	3
4.2.2	XAMPP Local Host	4
4.3	Technologies Used	4
4.3.1	HTML5	4
4.3.2	CSS	4
4.3.3	JavaScript	5
4.3.4	SMTP	5
4.3.5	PHP	5
4.4	Major Task (Incremental Model)	6
Chapter 2	Theoretical Study	8
	Introduction	9
1	What is a Gamified English Learning Platform?	9
2	Types of Gamified Learning Features in Our Project	9
2.1	Interactive Vocabulary & Matching Games	9
2.2	Pronunciation & Listening Challenges	9
2.3	Story-Based Interactive Adventures	9
2.4	Progress & Reward System	9
3	Benefits of Gamified English Learning Platforms	10
3.1	Boosted Engagement & Motivation	10

3.2	Enhanced Language Skills & Retention	10
3.3	Cognitive & Developmental Advantages	10
3.4	Accessibility & Long-Term Benefits	10
	Similar Systems	11
1.1	Flashcards-Based Arabic Educational Games	11
1.2	Lingokids	11
1.3	Duolingo	11
	In Summary — Why Our System Stands Out	12
	Comparison of similar Systems	13
Chapter 3	Analytical and Design Study	14
	Introduction	15
1	Requirements Gathering Approach	15
2	Project Requirements	15
2.1	Functional Requirements	15
2.2	Non-Functional Requirements	16
	Requirements Tracing Matrix (Table 2)	18
3	Analytical Chart of the Project	19
3.1	Use-Case Diagram (Figure 1 & 2)	19
3.2	Use-case Narrative (Tables 3–22)	21
3.3	Class Diagram (Figure 3)	28
3.4	ERD-Diagram (Figure 4)	29
3.5	Site Maps of The System (Figure 5)	30
3.6	Gantt Chart (Figure 6)	30
3.7	Activity Diagrams (Figures 7–11)	31
3.8	Sequence Diagrams (Figures 12–21)	34
4	Database (Figure 21 – Class Diagram)	39
Chapter 4	Implementation and Testing of Project	40
1	Log-In (Figure 22)	41
2	(Admin) Dashboard (Figure 23)	41
3	(Admin) Blacklist (Figure 24)	42

4	(Content Creator) Games (Figure 25)	42
5	(Content Creator) Levels (Figure 26)	43
	Testing (Table 23)	46
Chapter 5	Conclusion and Next Stages	48
	Conclusion	49
Chapter 6	References and Appendices	50
6	References	51
Appendix A	PHP Programming Language	52
Appendix B	MySQL Database Management System	52

Table of figure

Figure No.	Title / Description	Page
Figure 1	Use-Case Diagram high level	19
Figure 2	Use-Case Diagram low level	20
Figure 3	Class Diagram	28
Figure 4	ERD-Diagram	29
Figure 5	Site Map-Diagram	30
Figure 6	Gantt Chart	30
Figure 7	Activity Diagram (Use cases 1,2,3)	31
Figure 8	Activity Diagram (Use cases 4,5,6)	31
Figure 9	Activity Diagram (Use cases 7,8,9)	32
Figure 10	Activity Diagram (Use cases 10,11,12,13)	32
Figure 11	Activity Diagram (Use cases 14,15,16,17)	33
Figure 12	Activity Diagram (Use cases 18,19,20)	33
Figure 13	Sequence Diagram (use cases 1,2)	34
Figure 14	Sequence Diagram (use cases 3,4)	34
Figure 15	Sequence Diagram (use cases 5,6)	35
Figure 16	Sequence Diagram (use cases 7,8)	35
Figure 17	Sequence Diagram (use cases 9,10)	36
Figure 18	Sequence Diagram (use cases 11,12)	36
Figure 19	Sequence Diagram (use cases 13,14)	37
Figure 20	Sequence Diagram (use cases 15,16)	37
Figure 21	Sequence Diagram (use cases 17,18)	38
Figure 22	Sequence Diagram (use cases 19,20) + Class Diagram (Database)	38
Figure 23	Database	39
Figure 24	Log-In	41
Figure 25	Admin Dashboard	41
Figure 26	Admin Blacklist	42

Figure 27	Content Creator Games	42
Figure 28	Content Creator Levels	43
Figure 29	levels.php	43
Figure 30	Unauthorized Access Page	44
Figure 31	Error Page	44
Figure 32	autoload.php	44
Figure 33	conf.php	45
Figure 34	db.class.php	45

List of tables

Table No.	Title / Description	Page
Table 1	Comparison of Similar Systems	13
Table 2	Requirements Tracing Matrix (RTM)	18
Table 3–22	Use Case Narratives (each use case in separate table)	21–27
Table 23	Test Cases	46

Chapter1

Introduction and Definition of the Project

1) Solve The Problem:

In light of the rapid development in educational technologies and the increasing reliance on digital learning worldwide, many traditional methods of teaching English to children still depend on paper-based materials, textbooks, and conventional classroom activities. This creates several challenges: long preparation time for teachers, limited engagement for children (leading to boredom and reduced motivation), difficulty in providing personalized follow-up, and restricted access to updated, high-quality content — especially in different geographical locations.

The problem we aim to solve is to provide a modern, integrated, and fully digital alternative system for teaching English to children aged 8–14. A motivating, interactive, and automated platform that children, parents, teachers, content creators, and administrators can use easily, smoothly, and with accurate and up-to-date information, eliminating the limitations of traditional methods.

2) Aim of the Project:

The project aims to deliver a comprehensive, flexible, and accessible educational platform specialized in teaching English to children (8–14 years) through fun and interactive games.

The platform is designed to achieve the following key objectives:

- **Constant availability and accessibility** — allowing children, parents, teachers, and administrators to use the system anytime and anywhere using smartphones, tablets, or computers, without any time or geographical restrictions.
- **Exceptional ease of use and flexibility** — through:
 - Child-friendly, colourful, and intuitive interfaces specially designed for young learners
 - Professional, clear, and well-organized dashboards tailored for content creators, parents, teachers, and system administrators
- **Strong integration of education and entertainment (gamification)** — using animations, interactive games, points, rewards, levels, and visual progress tracking to dramatically increase children's motivation, reduce learning boredom, and enhance knowledge retention.
- **Effortless and continuous content renewal** — empowering content creators to add, modify, and publish new lessons, words, sentences, games, audio, and activities easily through a no-code content management interface.
- **Comprehensive progress monitoring & personalized follow-up** — providing parents and teachers with detailed, visual reports (progress percentages, completed stages, vocabulary mastery, time spent, strengths & weaknesses) along with periodic notifications to support effective personalized guidance.

Ultimately, the project seeks to build a complete, smart, sustainable, and scalable digital learning ecosystem that transforms English language learning into an enjoyable, effective, and truly accessible experience for every child — regardless of location, economic background, or access to traditional educational resources.

3) Documentation Description:

- 1) **Project Study:** Defining project objectives and development method.
- 2) **Theoretical study of the project:** The project was studied in theory and then tried to drop it on the practical side of the project.
- 3) **Analytical and Design Study of the Project:** Include detailed description of the project components and how to analyses, design, investigate and test these components.

4) Project's Management:

4.1 Human Recourse:

The working group include two students from Syrian Private University – Faculty of Computer and Informatics Engineering – Department of Software & Information System Engineering.

4.2 Programming Recourse:

4.2.1 Bootstrap5 (Front-End-Framework):

It is the most famous and powerful libraries that combine HTML, CSS and JS to build pages and applications, and it is mainly useful for building compatible sites with different screen sizes and sizes, including, cell phones, and it is a public and open-source library for everyone on the site.

We have relied on this platform in our project because:

- 1) Great flexibility and ease of use.
- 2) Designed to be compatible with all devices.
- 3) Compatible with all browsers.
- 4) Offers a stable and reusable design.
- 5) It can be learned easily and quickly.
- 6) Great scalability using JavaScript.
- 7) It has support for jQuery.
- 8) Support for the JavaScript API.
- 9) Can be used in any editor or programming environment.

4.2.2 XAMPP Local Host:

The Project was implemented using the local server (Apache Web Server) and has the following features:

1. Free and open source: the program has been well received and has built its own large community of users and developers, and received the necessary updates as quickly as possible, making it an ideal choice for many.
2. XAMPP supports many programs and services in embedded form, including the following:
 - 2.1 X: Supports Multiple Platforms.
 - 2.2 A: Supports Apache HTTP Server.
 - 2.3 M: Supports MariaDB(formerly MySQL rules).
 - 2.4 P: Supports the Programming Language PHP. 2.5 P: Supports the Programming Language Perl.
3. Multi-Platform: Supports(Windows, Linux, MacOS).
4. Easy to Install and Used: The program comes with all its contents within a single installation package, which makes it easy for the user to install and set it up automatically without the need to do any manual setup process.

4.3 Technologies Used:

4.3.1 HTML5:

HTML, or Hypertext Markup Language, allows web users to create and structure sections, paragraphs, and links using elements, tags, and attributes. However, it's worth noting that HTML is not considered a programming language as it can't create dynamic functionality.

4.3.2 CSS:

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable. CSS handles the look and feels part of a web page. Using CSS, you can control the colour of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colours are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects. CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

4.3.3 JAVA SCRIPT:

JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive.

Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user.

Common examples of JavaScript that you might use every day include the search box on Amazon, a news recap video embedded on the New York Times, or refreshing your Twitter feed.

Incorporating JavaScript improves the user experience of the web page by converting it from a static page into an interactive one. To recap, JavaScript adds behaviour to web pages.

4.3.4 SMTP:

The SMTP for Gmail is a feature you can use to send emails through different email clients using Gmail address or domain.

It was used when the employee accepts the sign-up request, after which an email is sent to the new client that his request has been accepted.

4.3.5 PHP:

PHP(short for Hypertext Preprocessor) is the most widely used open source and general-purpose server-side scripting language used mainly in web development to create dynamic websites and applications. It was developed in 1994 by

Rasmus Lerdorf. A survey by W3Tech shows that almost 79% of the websites in their data are developed using PHP. It is not only used to build the web apps of many tech giants like Facebook but is also used to build many CMS (Content Management System) like WordPress, Drupal, Shopify, WooCommerce etc. PHP can actually do anything related to server-side scripting or more popularly known as the backend of a website. For example, PHP can receive data from forms, generate dynamic page content, can work with databases, create sessions, send and receive cookies, send emails etc. There are also many hash functions available in PHP to encrypt user's data that makes PHP secure and reliable to be used as a server-side scripting language. So, these are some of the abilities of PHP that makes it suitable to be used as server-side scripting language. You will get to know more of these abilities in further tutorials.

4.4 Major Task:

The project was developed using the **Incremental Model**, where the system is built and delivered in small, functional increments (iterations). Each increment adds new features and improvements, allowing early delivery of working parts, continuous feedback, and progressive risk reduction.

The project was structured through several incremental phases, with the following key activities repeated and refined in each iteration:

1. Planning Phase (for each increment):

- Define the objectives and scope of the current increment
- Prioritize features to be implemented (e.g., core games → advanced games → dashboards → reporting)
- Estimate time, effort, and resources required for the increment
- Plan continuous communication and feedback sessions with stakeholders (client / educational experts / potential end-users)

2. Requirements Analysis & Specification (for each increment):

- Gather and refine detailed requirements for the features planned in this increment
- Create prototypes or mock-ups (especially for child interfaces and dashboards)
- Validate requirements with the client and potential users (teachers/parents/children)

3. Design & Development (Engineering):

- Design the architecture and user interfaces for the current increment
- Develop and implement the planned features (e.g., 3–5 interactive games per increment, parts of the admin/content dashboards, progress tracking modules)
- Write clean, modular, and testable code
- Perform unit testing and integration testing for the new features

4. Testing & Integration:

- Conduct thorough testing of the new increment (functional, usability, and performance testing)
- Integrate the new features with the previously delivered increments
- Perform regression testing to ensure existing functionality remains intact

5. Evaluation & Feedback:

- Deliver the working increment to the client, educational experts, and a sample group of children (aged 8–14)

- Collect detailed feedback on usability, engagement, learning effectiveness, and any issues
- Identify any emerging risks (e.g., performance issues, user experience problems, schedule delays)
- Document lessons learned and adjust the plan for the next increment

6. Deployment & Release (after selected increments):

- Deploy stable increments to production (e.g., initial version with basic games, followed by versions with parent dashboard, content management, full reporting)
- Provide early access to real users for continuous validation

Main Advantages Realized by Using the Incremental Model:

- Early and frequent delivery of usable parts of the system
- Continuous feedback from children, parents, and educators → better alignment with real learning needs
- Reduced risk through small, manageable increments
- Easier adaptation to changing requirements (new curriculum needs, user preferences)
- Higher user satisfaction due to visible progress throughout development

This incremental approach was particularly suitable for an educational game-based platform, as it allowed us to validate learning effectiveness, fun factor, and usability step-by-step with real children from the very beginning.

Chapter 2

Theoretical Study

Introduction:

Gone are the days when children had to sit through long, boring English lessons using only textbooks, rote memorization, and repetitive drills that often led to disengagement and frustration. With the rapid advancement of educational technology, learning English has become far more convenient, interactive, and enjoyable. Gamified digital platforms represent one of the most important innovations in modern childhood education, transforming language acquisition into an exciting and motivating experience.

1) What is a Gamified English Learning Platform?

A gamified English learning platform is an interactive digital system that integrates game design elements — such as points, rewards, levels, animations, badges, progress tracking, and fun challenges — into educational content. The goal is to make learning English feel like playing a game rather than studying, thereby boosting children's engagement, motivation, and retention while building essential language skills (vocabulary, pronunciation, listening, reading, and basic speaking) in a natural and enjoyable way.

2) Types of Gamified Learning Features in Our Project:

In our platform, specially designed for children aged 8–14, we incorporate several engaging game-based elements:

2.1 Interactive Vocabulary & Matching Games:

Fun matching, puzzles, and word association activities that help children build and reinforce vocabulary through repetition and immediate feedback.

2.2 Pronunciation & Listening Challenges:

Audio-based games with voice recording, sound matching, and interactive stories where children practice listening comprehension and clear pronunciation in a playful environment.

2.3 Story-Based Interactive Adventures:

Narrative-driven games and quests where children progress through levels by completing language tasks, earning rewards, and unlocking new content — combining storytelling with skill-building.

2.4 Progress & Reward System:

Points, badges, levels, daily streaks, and virtual rewards that motivate consistent practice and celebrate achievements, making learning addictive in a positive way.

3) Benefits of Gamified English Learning Platforms:

The advantages of this approach are numerous, especially for young learners. Here are some key benefits aligned with our project's focus:

3.1 Boosted Engagement & Motivation:

- Turns learning into play, reducing boredom and increasing children's willingness to practice regularly.
- Immediate rewards and progress visuals create a sense of achievement and fun.

3.2 Enhanced Language Skills & Retention:

- Improves vocabulary acquisition, pronunciation accuracy, and listening comprehension through repeated, enjoyable exposure.
- Supports better memory, cognitive flexibility, and problem-solving — skills strengthened by early bilingual experiences.

3.3 Cognitive & Developmental Advantages:

- Stimulates brain activity, creativity, attention, and critical thinking.
- Builds confidence and a positive attitude toward English, making children more open to cultural diversity and global opportunities.

3.4 Accessibility & Long-Term Benefits:

- Available anytime, anywhere on smart devices, removing barriers of time, location, or traditional classroom limitations.
- Prepares children for future academic success, better career prospects, and easier learning of additional languages in our globalized world.

In a Nutshell: Don't think of investing time in English learning apps or platforms as an unnecessary distraction or extra burden on children's schedules. Instead, view them as powerful tools that deliver real educational value through enjoyment.

Choose a gamified approach that matches your child's age and needs — it can help build strong language foundations, prevent future learning gaps, and open doors to endless opportunities without sacrificing fun or affecting their overall well-being. Early, engaging English learning is an investment in a brighter, more connected future!

Similar Systems:

1.1 Flashcards-Based Arabic Educational Games (e.g., ألعاب فلاش تعليمية / Digital Dialects & Similar Arabic Flash Tools):

We reviewed several popular Arabic educational flashcard and game platforms (such as Digital Dialects Arabic, Jarir/Alkitab flashcard sets, and various Arabic learning games sites). These are mostly simple, static tools focused on basic vocabulary, alphabet, and matching games, often in Arabic or bilingual Arabic-English.

Key observations:

- They are limited to predefined, non-updatable content (no easy way for educators to add new games or levels).
- No advanced admin/parental dashboards for progress tracking or personalized reports.
- Primarily offline or basic web-based, with little interactivity, gamification (rewards/levels), or focus on pronunciation/listening for English learners.
- No subscription model, but lack of continuous updates and scalability.

1.2 Lingokids (Playlearning™ App for Kids Aged 2–8):

Lingokids is a well-known, highly engaging app for younger children (primarily 2–8 years), using songs, videos, games, stories, and activities to teach English (and some other skills). It features colourful interfaces, progress tracking for parents, and a subscription model (e.g., monthly/yearly plans for full access).

Key features:

- Strong gamification with rewards, animations, and fun characters.
- Parental reports on activity completion and time spent.
- Continuous content updates by the company team.

Limitations compared to our system:

- Targeted at younger ages (less suitable for 8–14).
- No easy no-code content creation dashboard for teachers/educators to add custom games or align with specific curricula.
- Requires paid subscription for full features, no free unlimited access.
- Less emphasis on detailed individual progress analytics or admin-level control.

1.3 Duolingo (Including Duolingo Kids/ABC Features):

Duolingo is one of the most popular gamified language learning platforms worldwide, with bite-sized lessons, points, streaks, and rewards. Duolingo Kids/ABC focuses on

younger learners (3–8+) with visual icons, simple games, and reading/phonics emphasis, while the main app suits older kids/teens.

Key features:

- Highly engaging gamification and daily practice motivation.
- Free core access (with ads) or premium ad-free.
- Progress tracking and adaptive difficulty.

Limitations compared to our system:

- Content is fixed and updated only by Duolingo team — no user/admin dashboard to create or customize new games/levels easily.
- Less focus on pronunciation/listening for early stages; more translation/icon based.
- Limited parental/admin deep tracking (basic progress, no detailed reports or notifications).
- Not specifically tailored for Arabic-speaking children learning English with curriculum alignment.

In Summary — Why Our System Stands Out: our platform combines the best of these competitors (fun gamification, ease of use, updated content) while adding unique advantages:

- Full content creator dashboard for easy, no-code addition of new games, levels, audio, and activities (perfect for teachers/schools to keep aligned with modern curricula).
- Advanced admin/parental dashboards with detailed, visual progress reports, strengths/weaknesses identification, and periodic notifications.
- Focused on 8–14 age group with deeper English skills (vocabulary, pronunciation, listening).
- Flexible accessibility across devices, with potential for free core use and sustainable scaling — without heavy reliance on subscriptions or ads.

Comparison of Similar Systems:

Our System	ألعاب Flash التعليمية	Lingo kids	Duolingo	System Feature
				محتوى منظم
				تركيز على مهارة الاستماع
				سهولة الاستخدام
				لوحة تحكم للمحتوى
				اشتراك مكلف
				محتوى متعدد

Table (1) Comparison of System

Chapter 3

Analytical and Design Study

Introduction

This section details the functional and non-functional requirements of the gamified English learning platform for children aged 8–14. These requirements serve as the foundation for the entire project. Any errors or omissions at this stage may lead to significant issues in subsequent development, testing, and deployment phases.

The requirements were defined based on:

- Analysis of the target users' needs (children, parents, teachers, content creators, and administrators)
- Review of similar educational platforms
- Best practices in gamified learning and child-centered design
- Continuous stakeholder feedback during the incremental development process

1. Requirements Gathering Approach Requirements were collected through:

- Study of existing children's English learning apps (e.g., Lingokids, Duolingo Kids, Arabic educational games)
- Review of modern educational curricula and gamification principles
- Iterative refinement during each development increment

2. Project Requirements

2.1 Functional Requirements

The system supports multiple user roles with clearly defined responsibilities and privileges.

Actor: System Administrator

- REQ-1: Search for users (children, parents, teachers, content creators)
- REQ-2: Edit user account information
- REQ-3: Block/suspend user account
- REQ-4: Generate usage and performance reports (active users, average playtime, completion rates, etc.)
- REQ-5: Add new content creator
- REQ-6: Edit content creator account
- REQ-7: Delete content creator account
- REQ-8: Review, approve, or edit games and content created by content creators

Actor: Content Creator

- REQ-9: Add new learning stage/level
- REQ-10: Edit existing learning stage/level
- REQ-11: Delete learning stage/level
- REQ-12: Add new rewards/prizes/badges
- REQ-13: Edit existing reward/prize/badge
- REQ-14: Delete reward/prize/badge
- REQ-15: Generate detailed reports about games, stages, student progress, and content usage

Actor: User (Parent/Child)

- View child's detailed progress reports (vocabulary mastery, completed stages, time spent, strengths/weaknesses)
- Receive periodic notifications (weekly/monthly) about child's performance
- Edit own profile
- Link/unlink child accounts
- Register/login (with parental consent/supervision)
- Play interactive educational games (matching, puzzles, stories, pronunciation challenges, etc.)
- View personal progress, points, badges, levels, and rewards
- Receive immediate feedback and encouragement during gameplay

2.2 Non-Functional Requirements

2.2.1 Usability

- All interfaces (especially for children) must be colorful, intuitive, child-friendly, and easy to navigate
- Minimal text, large buttons, clear icons, voice guidance, and instant visual feedback
- Dashboards for adults (teachers, admins, content creators) must be clean, professional, and logically organized

2.2.2 Performance

- Response time: ≤ 2 seconds for most user interactions (game loading, answer submission, dashboard refresh)
- Support for at least 1,000 concurrent users without noticeable degradation
- High accuracy in progress tracking, scoring, and report generation

2.2.3 Availability

- The system must be accessible 24/7 via web and mobile devices
- Target uptime: $\geq 99.5\%$ (excluding planned maintenance)

2.2.4 Reliability

- The system must handle unexpected user inputs gracefully (no crashes)
- All critical data (progress, rewards, content) must be preserved after system restarts or errors

2.2.5 Implementation Constraints

- Frontend: Responsive design (Flutter / similar modern framework)
- Backend: RESTful API
- Database: Relational (MySQL) for structured data

2.2.6 Testing

- Unit testing for all backend services and game logic
- Integration testing for APIs and database operations
- Usability testing with real children (8–14 years) in each increment
- Security testing (OWASP top 10) including input validation and authentication checks

2.2.7 Security

- All user passwords stored using strong hashing MD5
- Role-based access control (RBAC) to restrict actions per user type
- Protection against SQL injection, XSS, CSRF using prepared statements, input sanitization, and security headers
- HTTPS enforced across the entire platform
- Parental consent/verification required for child accounts
- Secure file uploads (images) with size/type validation and virus scanning

2.2.8 Maintainability & Scalability

- Modular architecture allowing easy addition of new stages
- No-code/low-code content creation interface for non-technical educators
- Automated backups and easy rollback mechanisms

Requirements tracing matrix

REQ ID	Title	actor	Priority	Description	Testing
REQ-1	Search user	Admin	1	UC-1	TC-1
REQ-2	Edit user	Admin	1	UC-2	TC-2
REQ-3	Block user	Admin	1	UC-3	TC-3
REQ-4	Generate reports	Admin	3	UC-4	TC-4
REQ-5	Add content creator	Admin	1	UC-5	TC-5
REQ-6	Edit content creator	Admin	1	UC-6	TC-6
REQ-7	Delete content creator	Admin	1	UC-7	TC-7
REQ-8	Edit game	Admin	2	UC-8	TC-8
REQ-9	Add level	Content creator	2	UC-9	TC-9
REQ-10	Edit level	Content creator	2	UC-10	TC-10
REQ-11	Delete level	Content creator	2	UC-11	TC-11
REQ-12	Add rewords	Content creator	2	UC-12	TC-12
REQ-13	Edit reword	Content creator	2	UC-13	TC-13
REQ-14	Delete reword	Content creator	2	UC-14	TC-14
REQ-15	Generate report	Content creator	3	UC-15	TC-15
REQ-16	Select daily target	User	-	UC-16	TC-16
REQ-17	Manage account	User	-	UC-17	TC-17
REQ-18	Select game	User	-	UC-18	TC-18
REQ-19	Edit language level	User	-	UC-19	TC-19
REQ-20	View rewords	User	-	UC-20	TC-20

Table (2) RTM

3.Analytical Chart of the Project:

3.1 Use-Case Diagram:

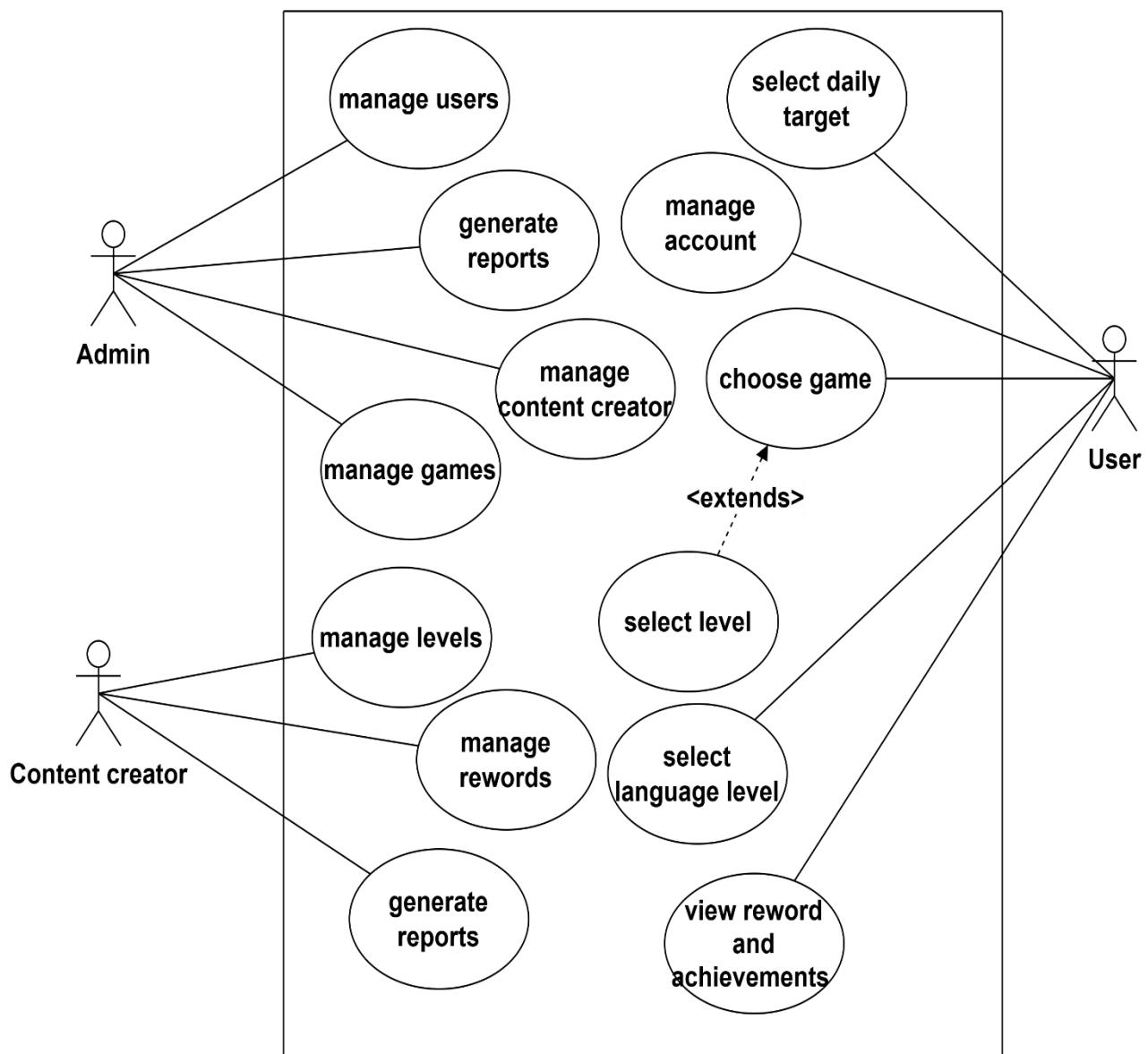


Figure (1) Use-Case Diagram high level

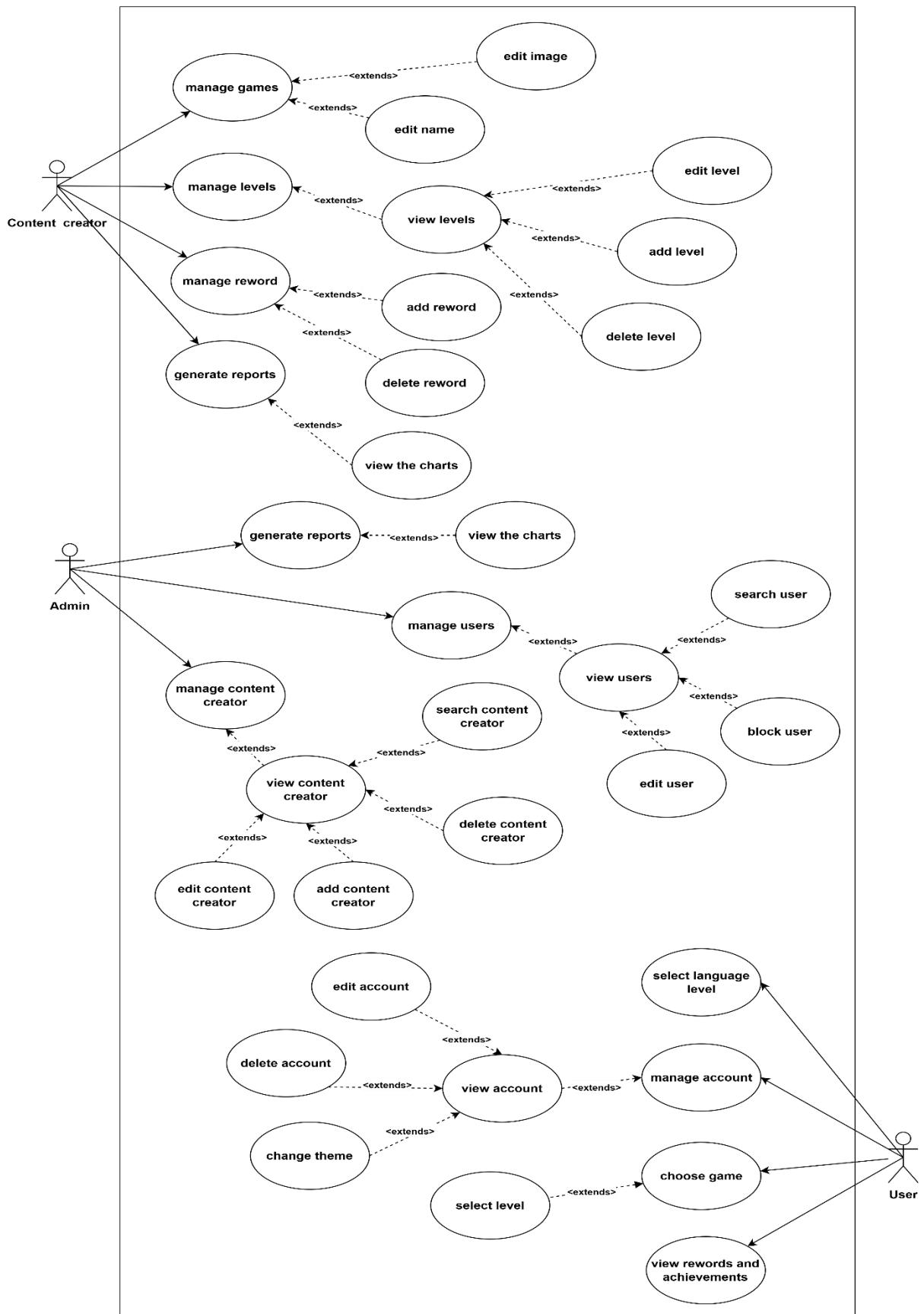


Figure (2) Use-Case Diagram low level

3.2 Use-case Narrative:

Field	Content
Use Case Name	Search for User
Use Case ID	UC-01
Actor	Admin
Description	The Admin searches for a user.
Preconditions	The Admin is logged in with full permissions.
Postconditions	The user is found and their data is displayed based on the name search.
Main Flow	1. Access the control panel 2. Select "Search for User" 3. Search by name 4. Display user data
Alternative Flow	If no data exists → Nothing is displayed
Related Requirement	REQ-1
Priority	1 (High)

Table (3) Use case 1

Field	Content
Use Case Name	Edit User Account
Use Case ID	UC-02
Actor	Admin
Description	Edit the data of an existing user account.
Preconditions	The user exists and the Admin is logged in.
Postconditions	The account data is updated.
Main Flow	1. Search for the user 2. Select "Edit" 3. Modify the fields 4. Save
Alternative Flow	Input error → Retry
Related Requirement	REQ-2
Priority	1 (High)

Table (4) Use case 2

Field	Content
Use Case Name	Block User
Use Case ID	UC-03
Actor	Admin
Description	Temporarily or permanently block a user account.
Preconditions	The user exists and the Admin is logged in.
Postconditions	Account status changed to "Blocked" and login is prevented.
Main Flow	1. Select the user 2. Choose "Block" 3. Specify block type 4. Confirm
Alternative Flow	Cancel → No change

Related Requirement	REQ-3
Priority	1 (High)

Table (5) Use case 3

Field	Content
Use Case Name	Generate Usage Reports
Use Case ID	UC-04
Actor	Admin
Description	View and export statistical reports on application usage.
Preconditions	Admin is logged in.
Postconditions	Reports are displayed or exported (PDF/Excel).
Main Flow	1. Access the reports section 2. Select period/type 3. View/export
Alternative Flow	No data → Message "No data available"
Related Requirement	REQ-4
Priority	3 (Low)

Table (6) Use case 4

Field	Content
Use Case Name	Add Content Manager
Use Case ID	UC-05
Actor	Admin
Description	Add a new content manager account.
Preconditions	Admin is logged in.
Postconditions	New account created with content manager permissions.
Main Flow	1. Select "Add Content Manager" 2. Fill in the data 3. Save
Alternative Flow	Duplicate data → Error
Related Requirement	REQ-5
Priority	1 (High)

Table (7) Use case 5

Field	Content
Use Case Name	Edit Content Manager
Use Case ID	UC-06
Actor	Admin
Description	Edit data or permissions of a content manager.
Preconditions	Content manager exists.
Postconditions	Data is updated.
Main Flow	1. Select the content manager 2. Edit fields 3. Save

Alternative Flow	Error → Retry
Related Requirement	REQ-6
Priority	1 (High)

Table (8) Use case 6

Field	Content
Use Case Name	Delete Content Manager
Use Case ID	UC-07
Actor	Admin
Description	Delete a content manager account.
Preconditions	Content manager exists.
Postconditions	Account is deleted.
Main Flow	1. Select content manager 2. Confirm deletion 3. Execute
Alternative Flow	Cancel → No deletion
Related Requirement	REQ-7
Priority	1 (High)

Table (9) Use case 7

Field	Content
Use Case Name	Edit Games
Use Case ID	UC-08
Actor	Admin
Description	Edit general game properties (e.g., difficulty, design).
Preconditions	Games exist in the system.
Postconditions	Games updated for all users.
Main Flow	1. Select the game 2. Edit settings 3. Save
Alternative Flow	Error → Retry
Related Requirement	REQ-8
Priority	2 (Medium)

Table (10) Use case 8

Field	Content
Use Case Name	Add Stage
Use Case ID	UC-09
Actor	Content Manager
Description	Add a new educational stage with language content.
Preconditions	Content manager is logged in.
Postconditions	New stage available to users.
Main Flow	1. Select "Add Stage" 2. Fill in content (words, exercises) 3. Save
Alternative Flow	Incomplete content → Error

Related Requirement	REQ-9
Priority	2 (Medium)

Table (11) Use case 9

Field	Content
Use Case Name	Edit Stage
Use Case ID	UC-10
Actor	Content Manager
Description	Edit content of an existing stage.
Preconditions	Stage exists.
Postconditions	Stage is updated.
Main Flow	1. Select the stage 2. Edit content 3. Save
Alternative Flow	Error → Retry
Related Requirement	REQ-10
Priority	2 (Medium)

Table (12) Use case 10

Field	Content
Use Case Name	Delete Stage
Use Case ID	UC-11
Actor	Content Manager
Description	Delete an educational stage.
Preconditions	Stage exists.
Postconditions	Stage removed from the system.
Main Flow	1. Select the stage 2. Confirm deletion 3. Execute
Alternative Flow	Cancel → No deletion
Related Requirement	REQ-11
Priority	2 (Medium)

Table (13) Use case 11

Field	Content
Use Case Name	Add Rewards
Use Case ID	UC-12
Actor	Content Manager
Description	Add new rewards (badges, points).
Preconditions	Content manager is logged in.
Postconditions	Rewards available to users.
Main Flow	1. Select "Add Reward" 2. Upload image and set conditions 3. Save
Alternative Flow	Unsupported file → Error
Related Requirement	REQ-12
Priority	2 (Medium)

Table (14) Use case 12

Field	Content
Use Case Name	Edit Reward
Use Case ID	UC-13
Actor	Content Manager
Description	Edit conditions or appearance of an existing reward.
Preconditions	Reward exists.
Postconditions	Reward is updated.
Main Flow	1. Select the reward 2. Edit properties 3. Save
Alternative Flow	Error → Retry
Related Requirement	REQ-13
Priority	2 (Medium)

Table (15) Use case 13

Field	Content
Use Case Name	Delete Reward
Use Case ID	UC-14
Actor	Content Manager
Description	Delete an unused reward.
Preconditions	Reward exists.
Postconditions	Reward is removed.
Main Flow	1. Select the reward 2. Confirm deletion 3. Execute
Alternative Flow	Cancel → No deletion
Related Requirement	REQ-14
Priority	2 (Medium)

Table (16) Use case 14

Field	Content
Use Case Name	Generate Reports on Games and Stages
Use Case ID	UC-15
Actor	Content Manager
Description	View statistics on usage of games and stages.
Preconditions	Content manager is logged in.
Postconditions	Reports are displayed.
Main Flow	1. Access reports section 2. Select period 3. View/export
Alternative Flow	No data → Empty message
Related Requirement	REQ-15
Priority	3 (Low)

Table (17) Use case 15

Field	Content
Use Case Name	Set Daily Goal
Use Case ID	UC-16
Actor	User (Child or Parent)
Description	Set a daily learning goal (number of words, minutes).
Preconditions	User is logged in.
Postconditions	Goal saved and tracked daily.
Main Flow	1. Access account settings 2. Set the goal 3. Save
Alternative Flow	Unrealistic goal → Suggest default values
Related Requirement	REQ-16
Priority	Not specified

Table (18) Use case 16

Field	Content
Use Case Name	Manage Account
Use Case ID	UC-17
Actor	User
Description	Edit personal data or password.
Preconditions	User is logged in.
Postconditions	Account data is updated.
Main Flow	1. Access profile 2. Edit fields 3. Save
Alternative Flow	Wrong password → Error
Related Requirement	REQ-17
Priority	Not specified

Table (19) Use case 17

Field	Content
Use Case Name	Select Game
Use Case ID	UC-18
Actor	User
Description	Choose and start a game from the list.
Preconditions	User is logged in.
Postconditions	Game starts and progress is recorded.
Main Flow	1. Display games list 2. Select a game 3. Start playing
Alternative Flow	Game not available (locked) → Message "Complete the previous stage"
Related Requirement	REQ-18
Priority	Not specified

Table (20) Use case 18

Field	Content
Use Case Name	Adjust Language Level
Use Case ID	UC-19
Actor	User
Description	Change the language difficulty level.
Preconditions	User is logged in.
Postconditions	New level applied to games.
Main Flow	1. Access settings 2. Select level 3. Save
Alternative Flow	No change → Current level remains
Related Requirement	REQ-19
Priority	Not specified

Table (21) Use case 19

Field	Content
Use Case Name	Monitor Rewards and Progress
Use Case ID	UC-20
Actor	User
Description	View earned rewards and progress in stages.
Preconditions	User is logged in.
Postconditions	Display only (no changes).
Main Flow	1. Access progress page 2. View charts and rewards
Alternative Flow	No progress → Motivational message "Start playing now!"
Related Requirement	REQ-20
Priority	Not specified

Table (22) Use case 20

3.3 Class Diagram:

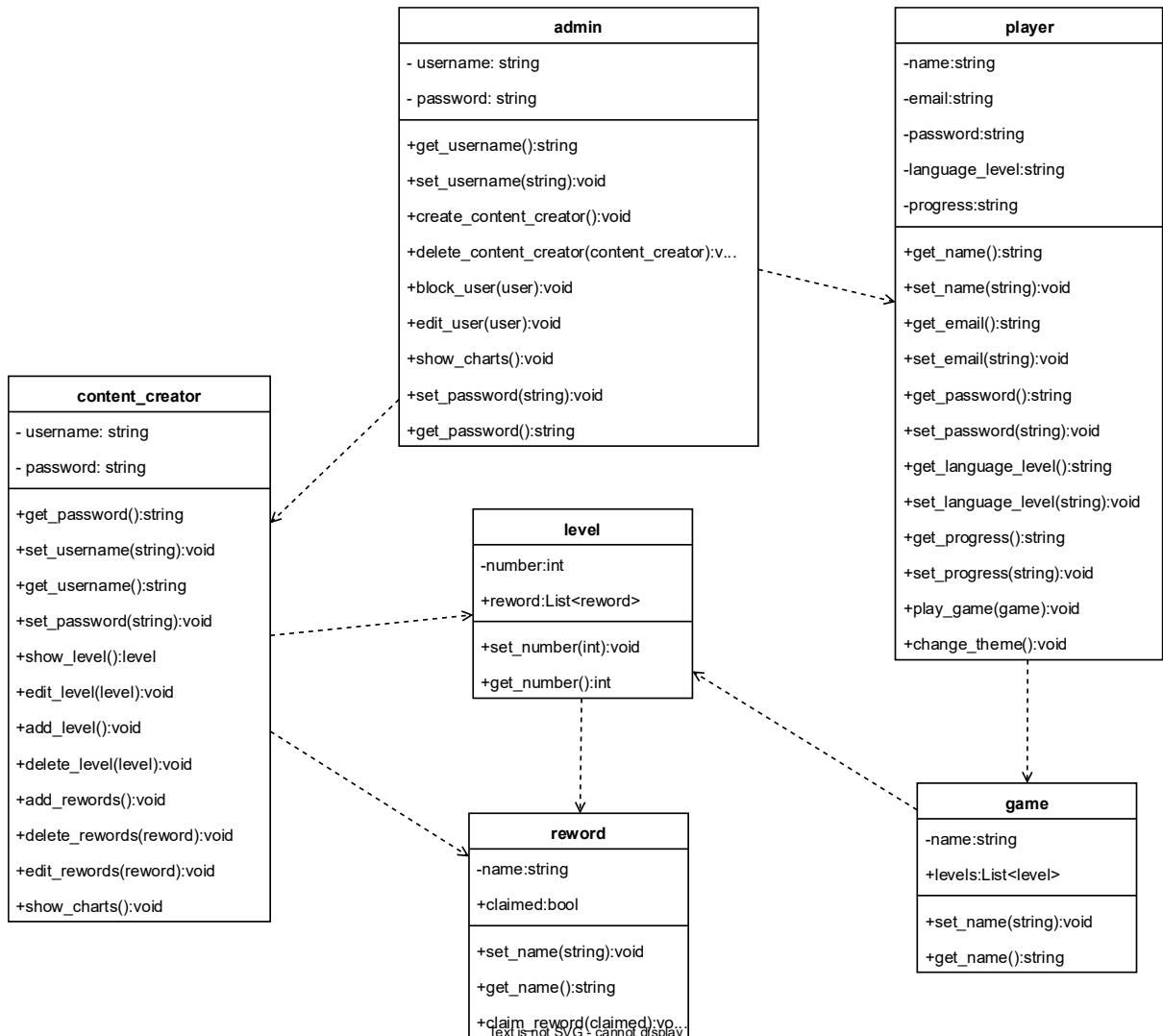


Figure (3) Class-Diagram

3.4 ERD-Diagram:

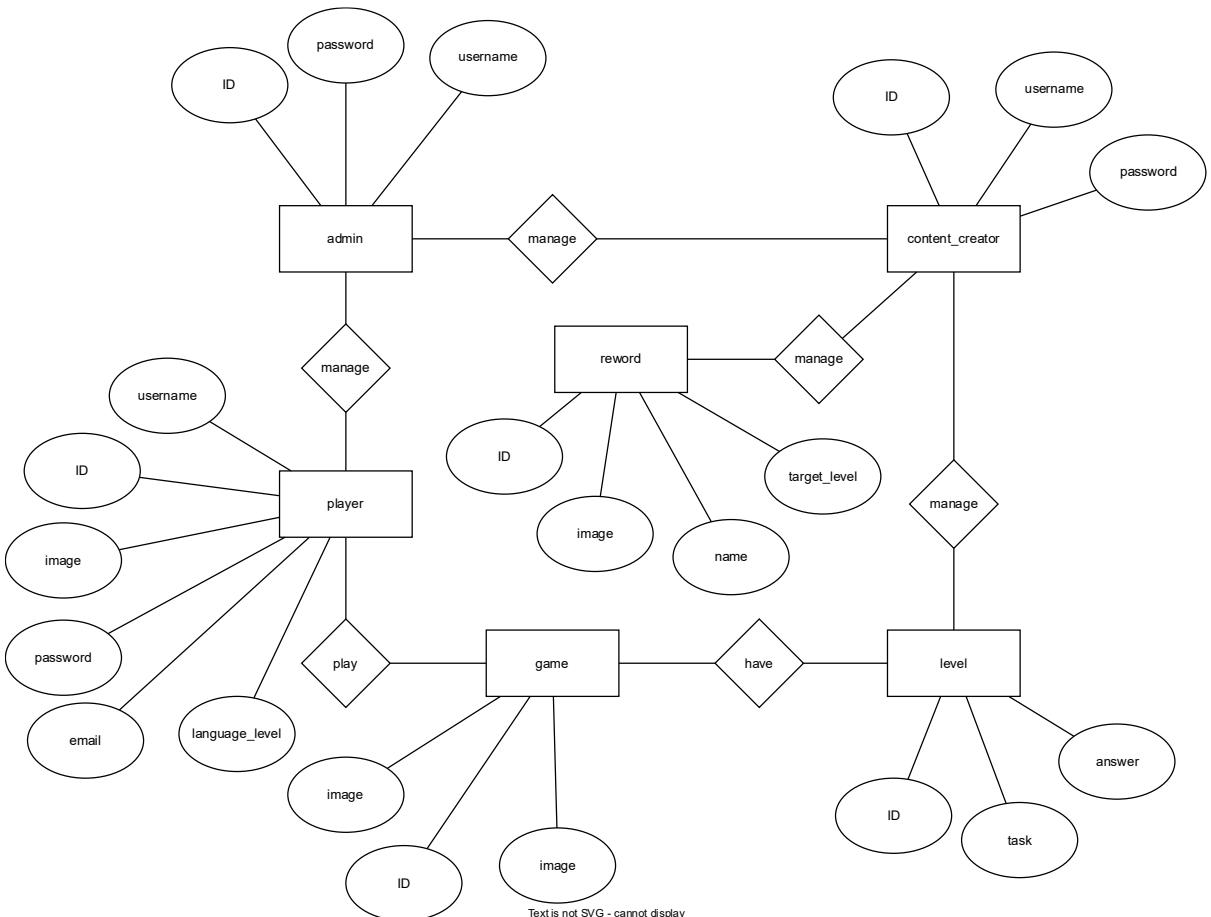


Figure (4) ERD-Diagram

3.5 Site Maps of The System:

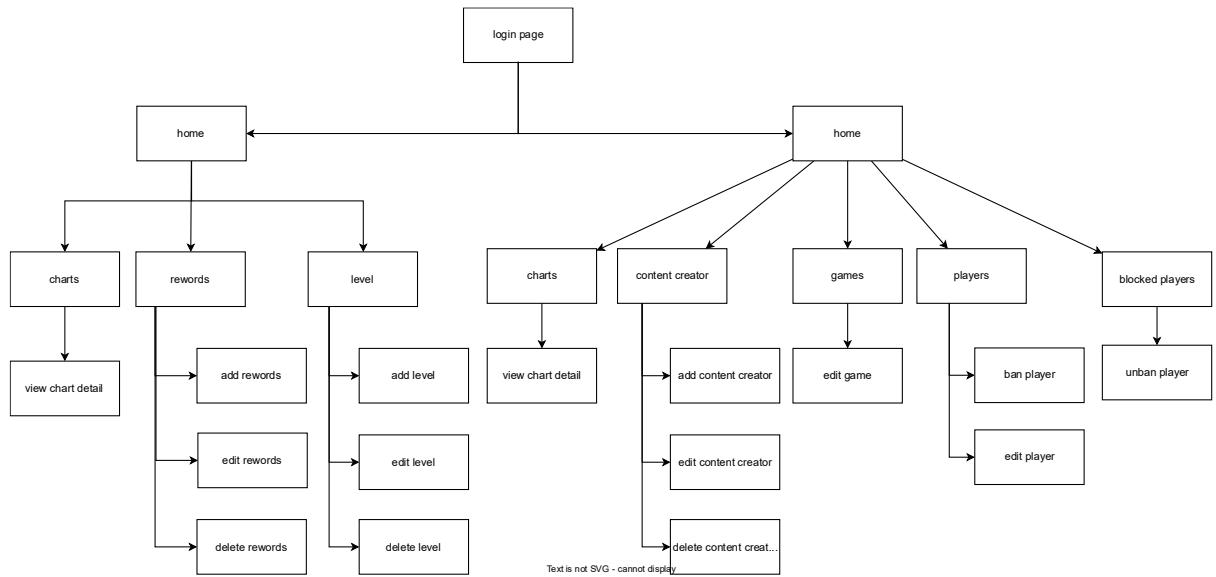


Figure (5) Site Map-Diagram

3.6 Gant Chart:

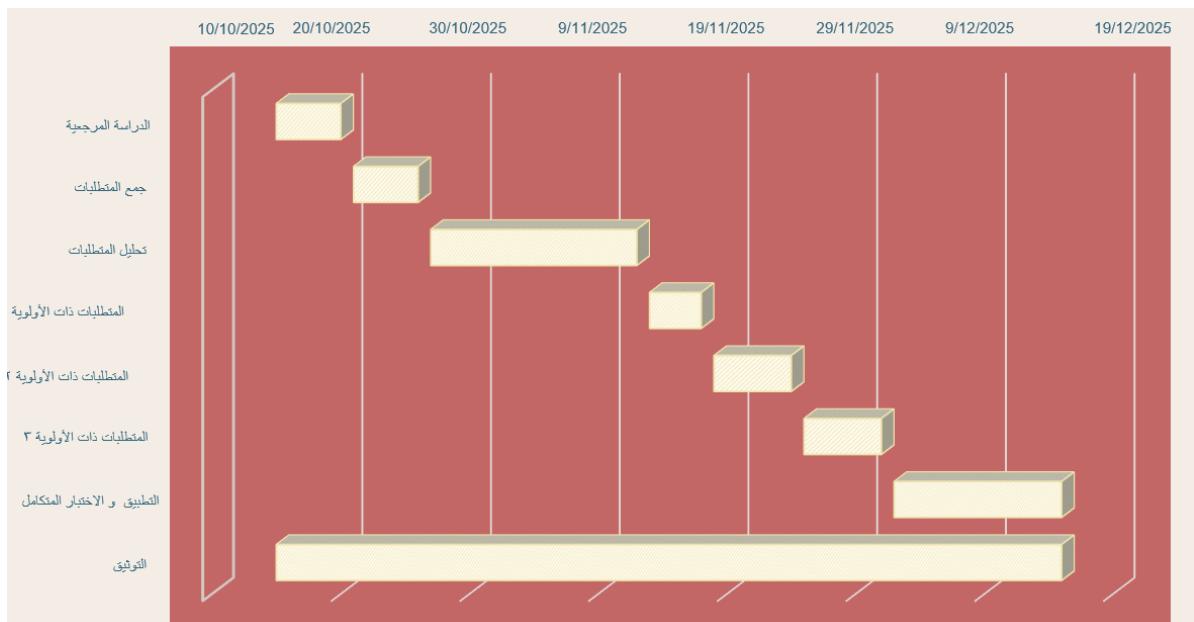


Figure (6) Gant Chart

3.7 Activity-Diagram:

3.7.1 Use cases 1,2,3

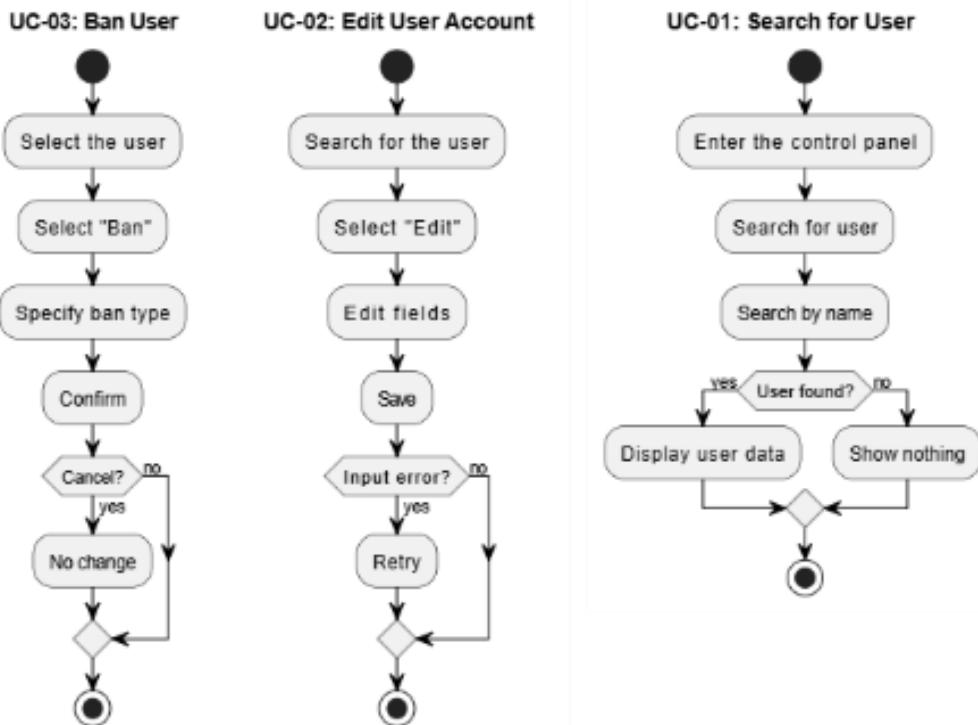


Figure (7) Activity -Diagram (1,2,3)

3.7.2 Use cases 4,5,6

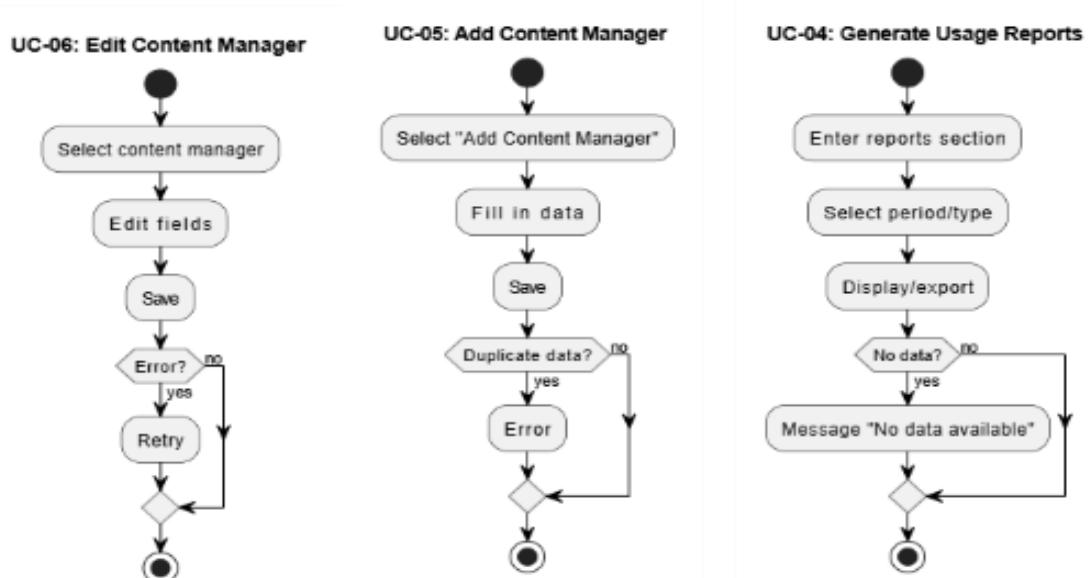


Figure (8) Activity -Diagram (4,5,6)

3.7.3 Use cases 7,8,9

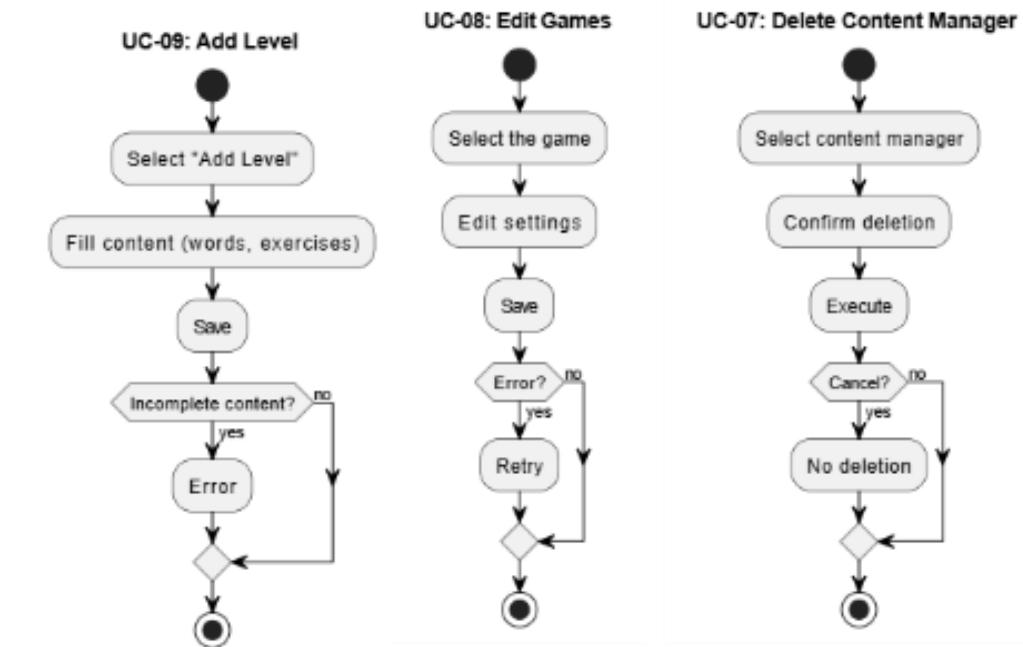


Figure (9) Activity -Diagram (7,8,9)

3.7.4 Use cases 10,11,12,13

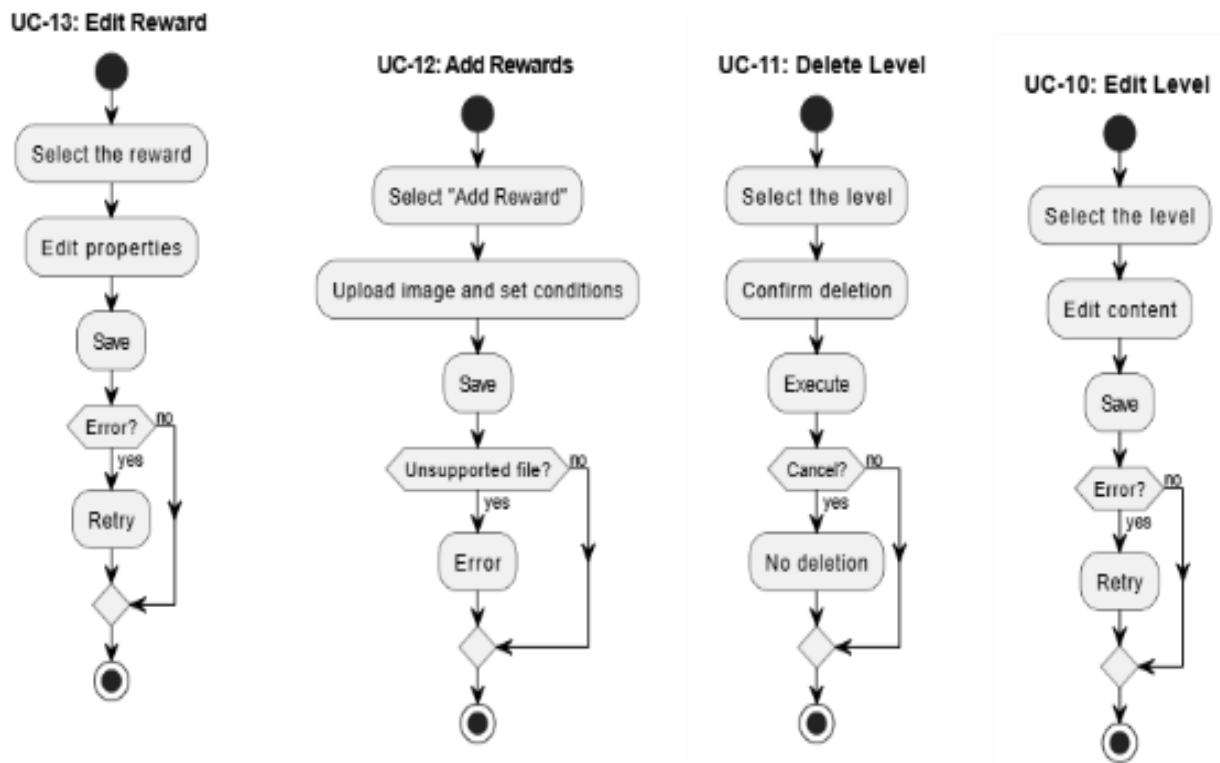


Figure (10) Activity-Diagram (10,11,12,13)

3.7.5 Use cases 14,15,16,17

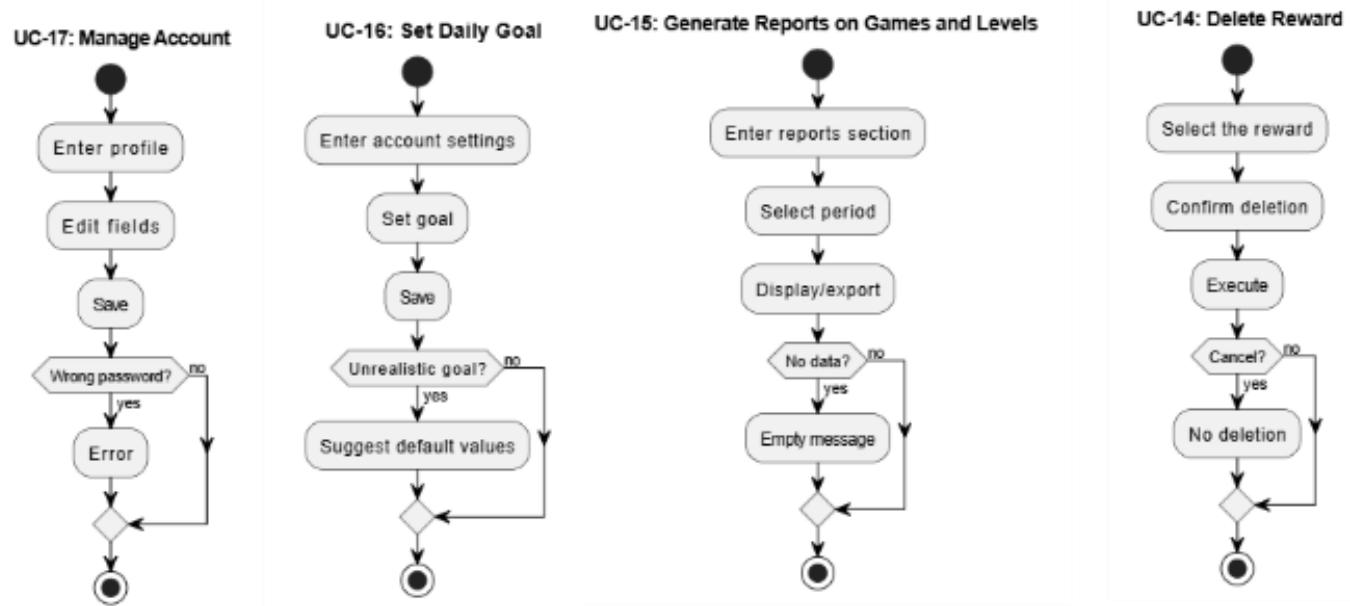


Figure (11) Activity-Diagram (14,15,16,17)

3.7.6 Use cases 18,19,20

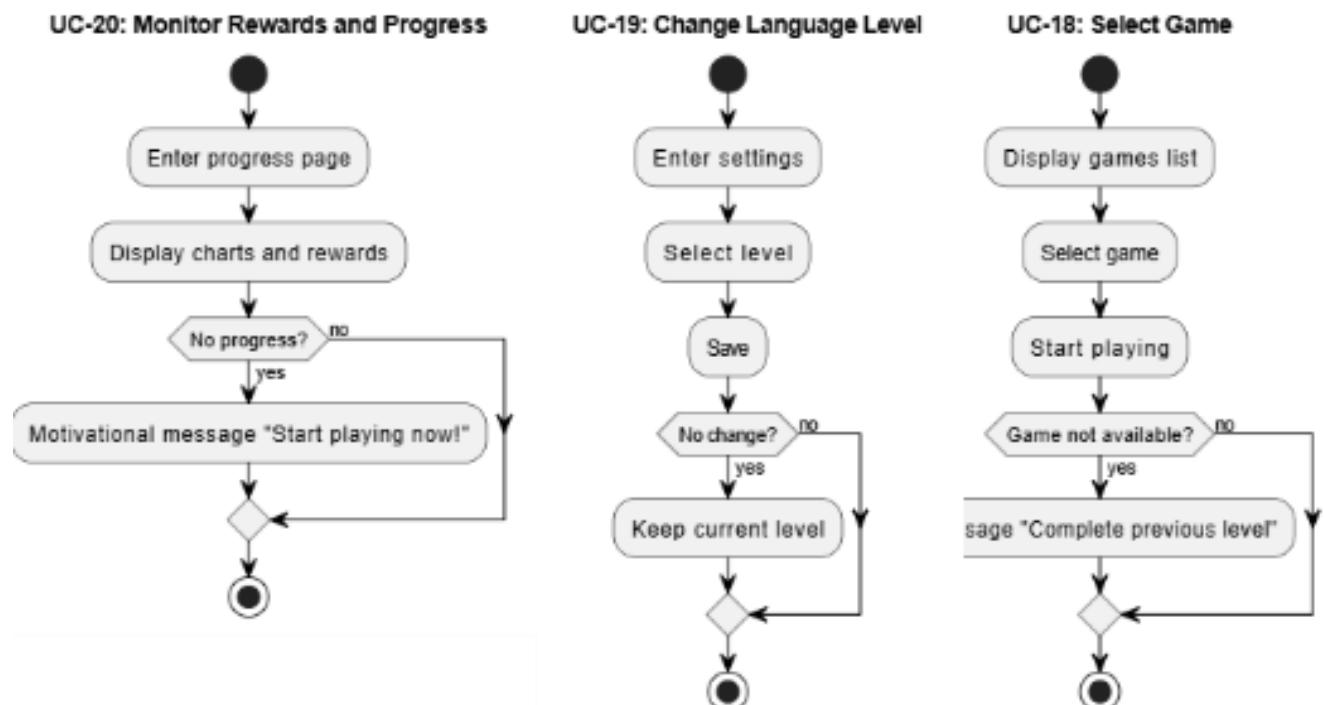


Figure (12) Activity-Diagram (18,19,20)

3.8 Sequence Diagram:

3.8.1 use cases 1,2

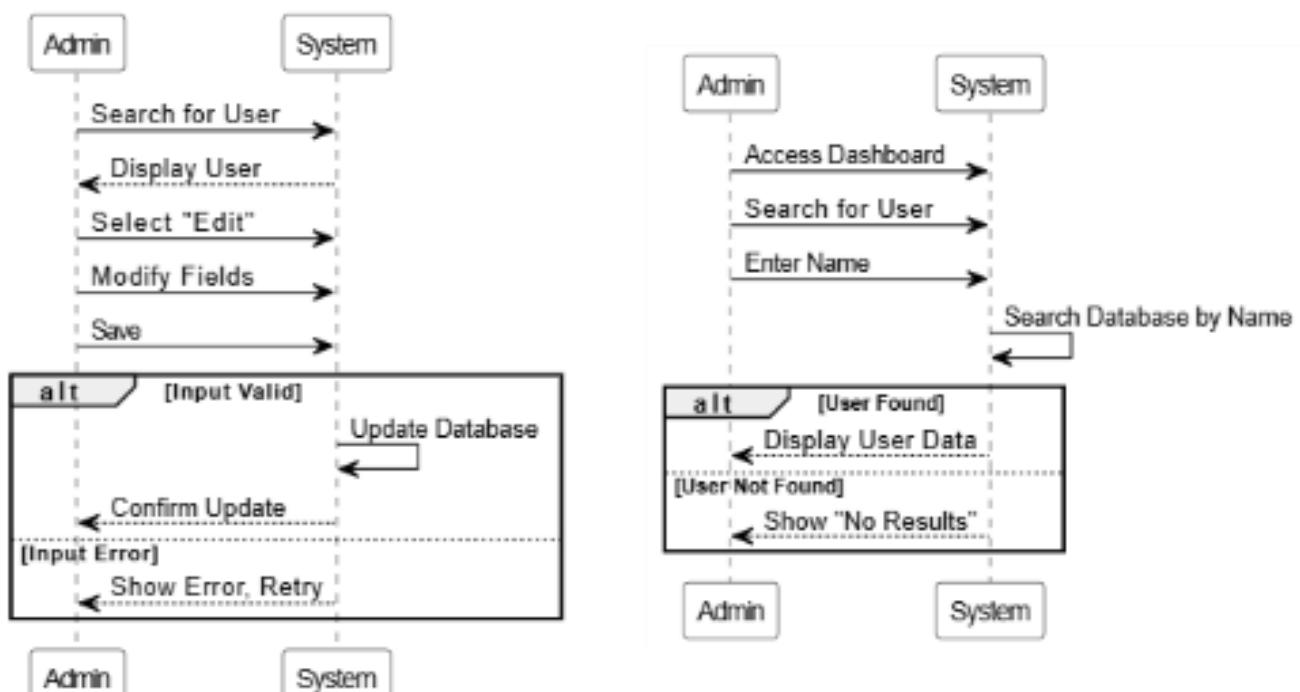


Figure (13) use cases 1,2

3.8.2 use cases 3,4

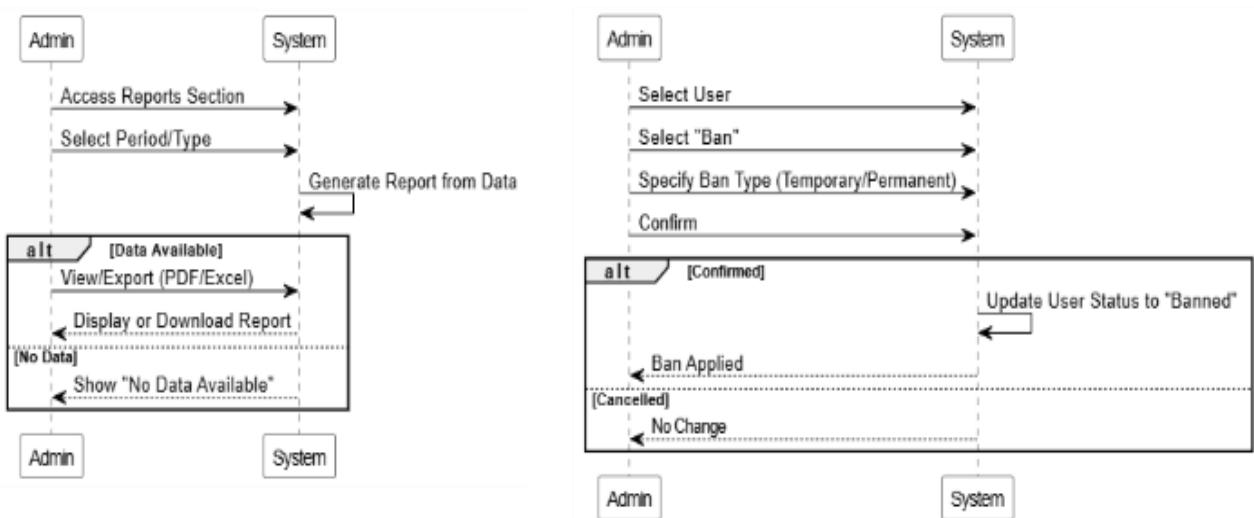


Figure (14) use cases 3,4

3.8.3 use cases 5,6

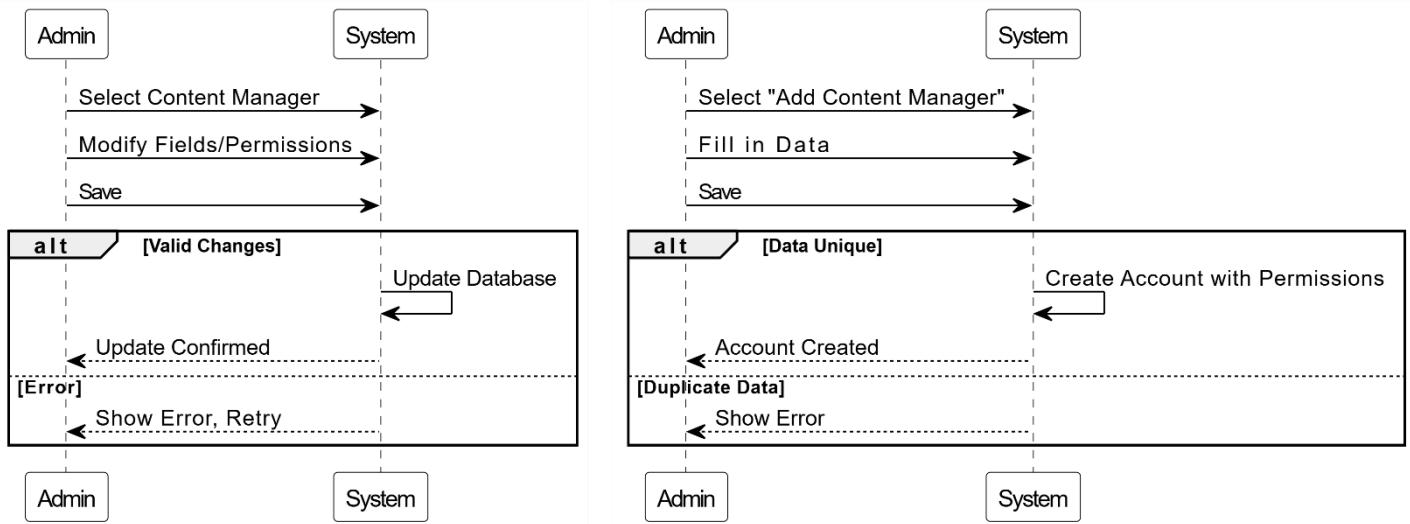


Figure (15) use cases 5,6

3.8.4 use cases 7,8

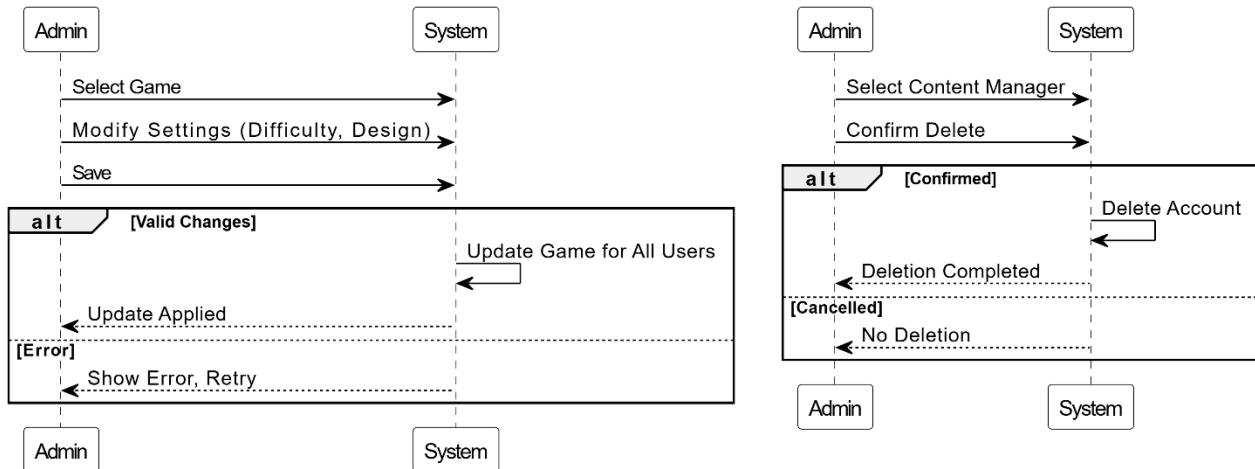


Figure (16) use cases 7,8

3.8.5 use cases 9,10

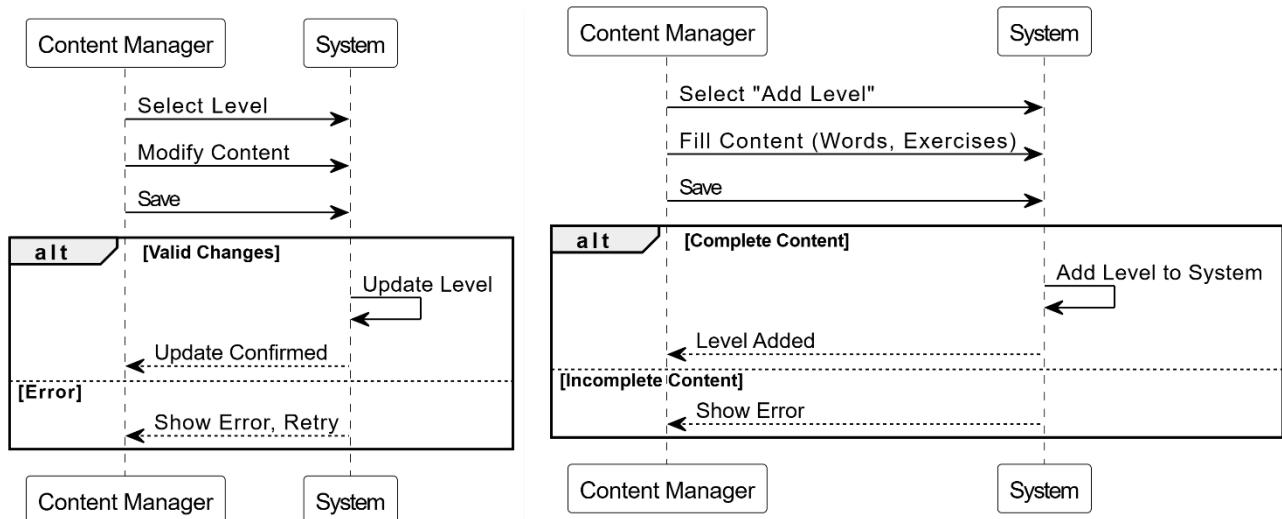


Figure (17) use cases 9,10

3.8.6 use cases 11,12

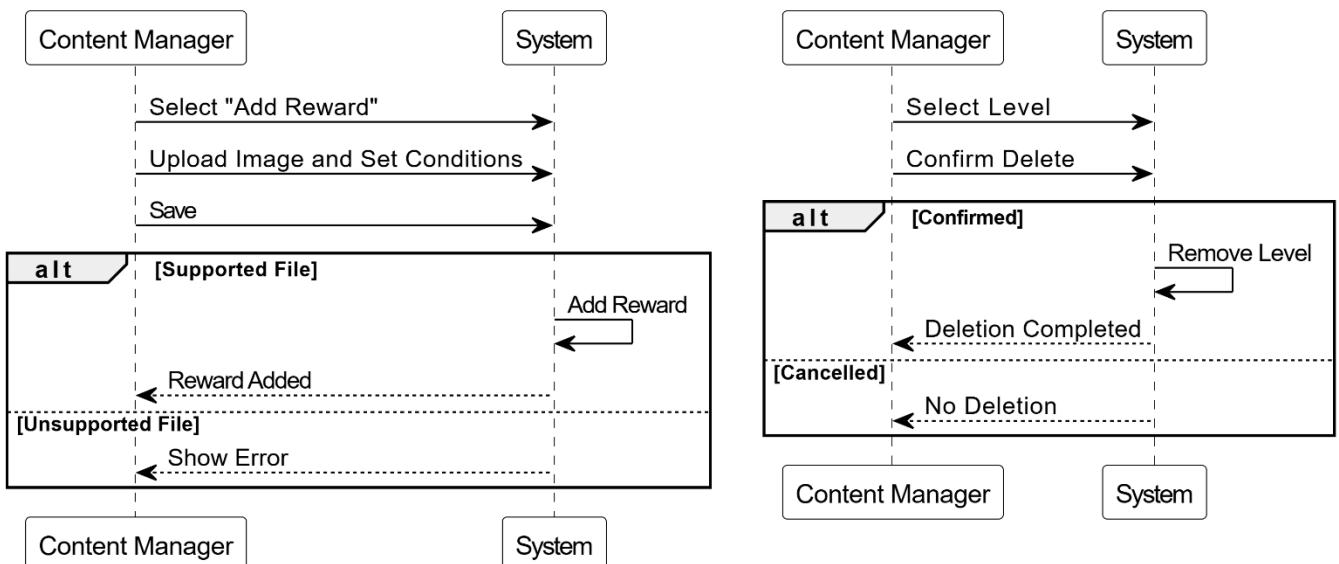


Figure (18) use cases 11,12

3.8.7 use cases 13,14

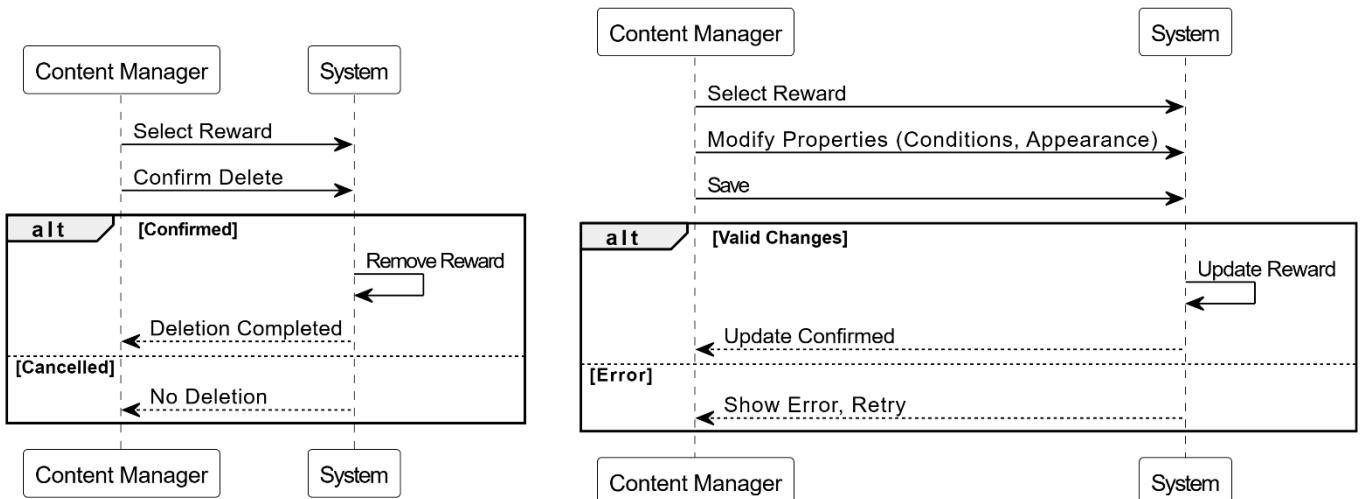


Figure (19) use cases 13,14

3.8.8 use cases 15,16

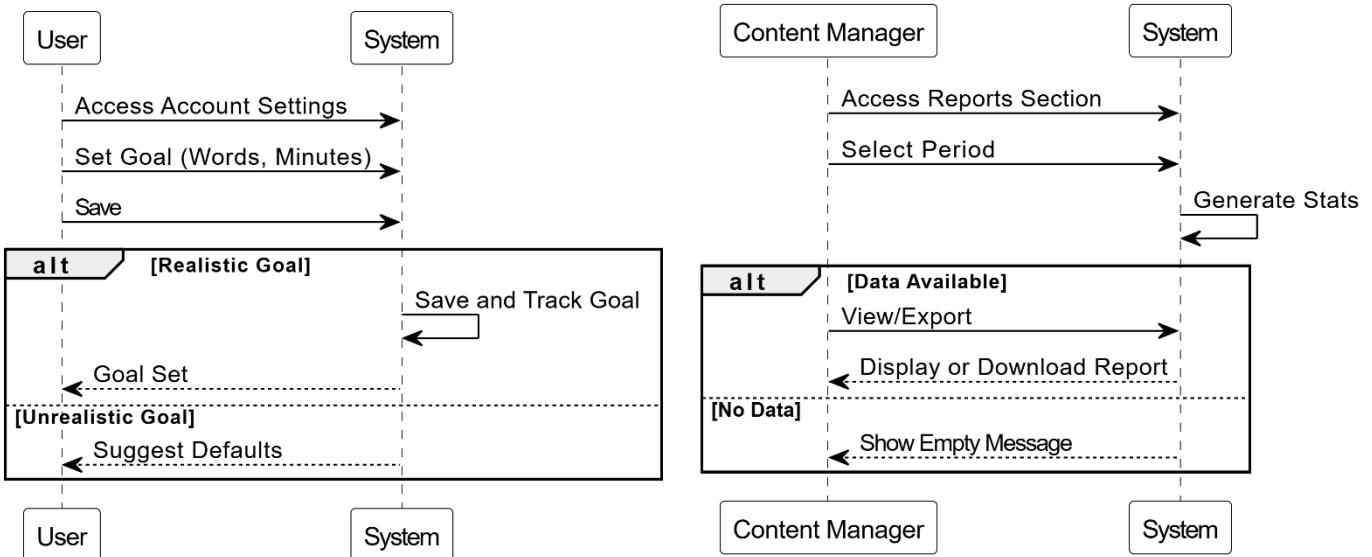


Figure (20) use cases 15,16

3.8.9 use cases 17,18

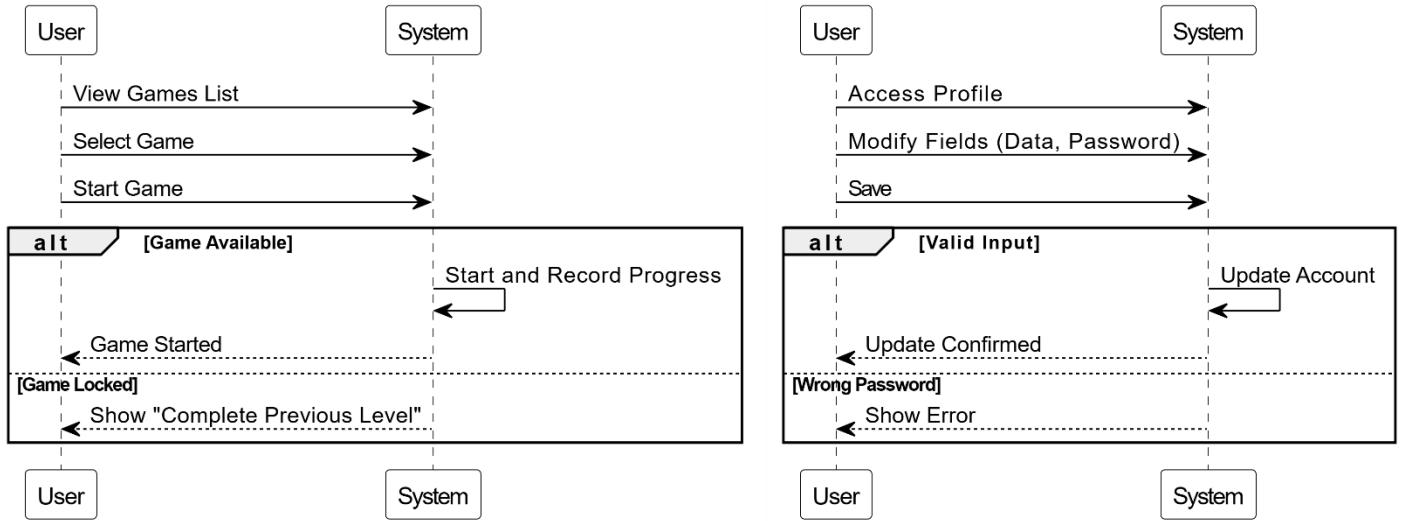


Figure (21) use cases 17,18

3.9.10 use cases 19,20

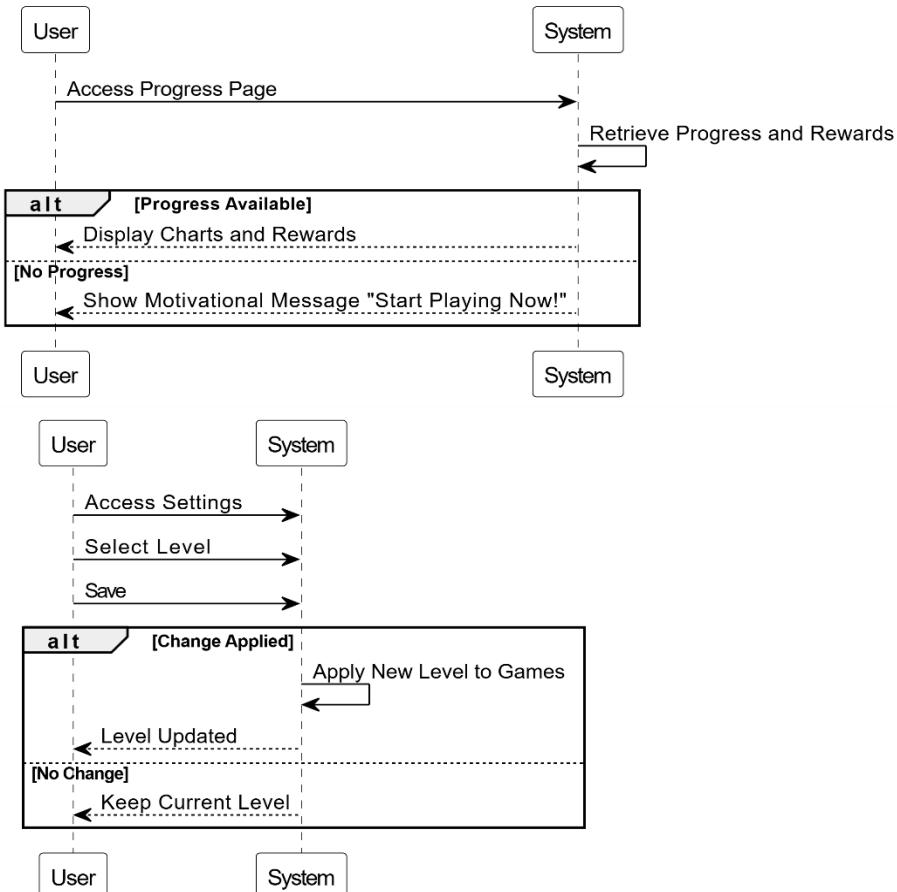


Figure (22) use cases 19,20

4. Database:

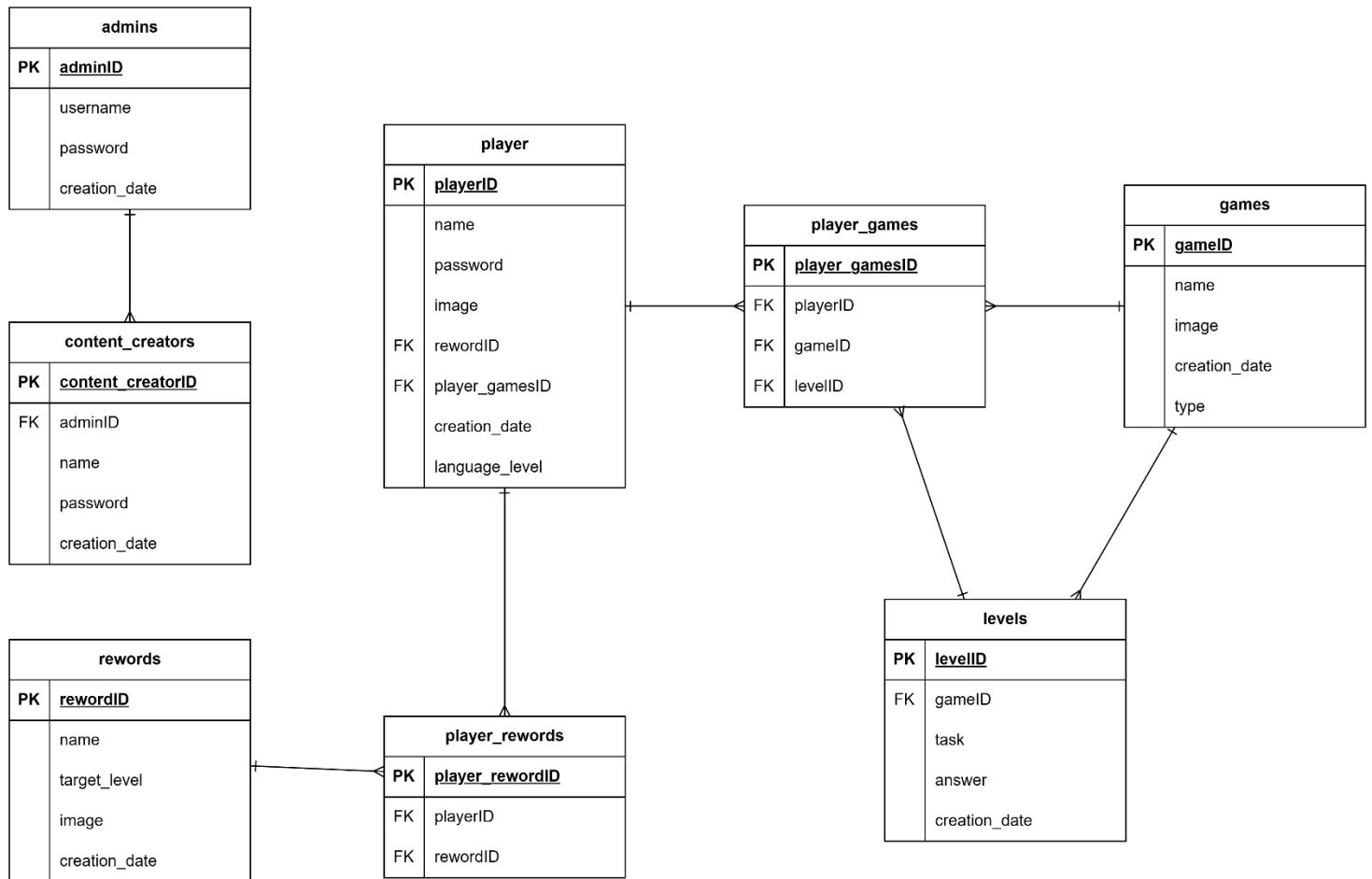


Figure (23) database

Chapter4

Implementation and Testing of Project

1.Log-In:

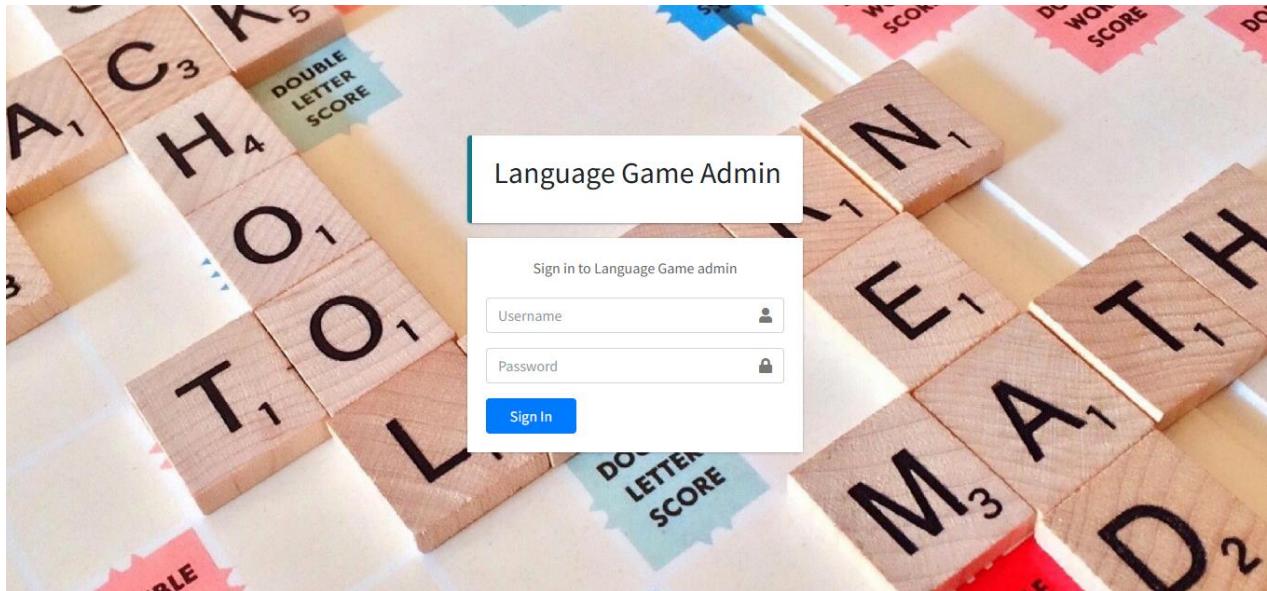


Figure (24) Log-In

On this page, the Admin and the Content creator enters his user's name and password, and then the site verifies the entered user name and password, and if they are correct, then it will move it to the main interface, but if the user's name or password is wrong, then it will show a message that there is no one with this information entered.

2.(Admin) Dashboard:

ID	Name	Image	Age	Gender	Status	Creation Date	Operations
1	karam		13	male	●	2023-05-09 12:05:10	<button>Edit</button> <button>block</button>
2	hala		14	female	●	2023-05-09 12:05:10	<button>Edit</button> <button>block</button>

Figure (25) dashboard

On this page, the site displays all the information about the system, which include Content creator, Players, Blacklist, charts and from any of them he can show details.

3.(Admin) Blacklist:

The screenshot shows the 'BLACK_LIST' page. On the left, a sidebar menu includes 'Content Creator', 'Players', 'Black List' (which is highlighted in blue), and 'Charts'. The main content area is titled 'BLACK_LIST' and contains a table with one row. The table columns are: Id, Name, Image, Age, Gender, Status, Creation Date, and Operations. The single entry is for a player named 'hala' (Id: 2), who is female, 14 years old, and has a red status icon. The creation date is 2023-05-09 12:05:10. A green button labeled 'Unblock' with a checkmark icon is visible in the 'Operations' column.

ID	Name	Image	Age	Gender	Status	Creation Date	Operations
2	hala		14	female	●	2023-05-09 12:05:10	<button>Unblock</button>

Figure (26) blacklist

On this page, the admin managers the players status to unblock them which allow them to access the game and play it again

4.(Content Creator) Games:

The screenshot shows the 'GAMES' page. On the left, a sidebar menu includes 'Games' (which is highlighted in blue), 'Reword', 'Guess the Picture', 'Translate', 'Answer the quistion', 'Match', and 'Charts'. The main content area is titled 'GAMES' and contains a table with three rows. The table columns are: Id, Image, Title Ar, Title En, Type, Creation date, and Operations. The three games listed are: 1. Guess the Picture (Arabic title: احجز الصورة), 2. Translate (Arabic title: ترجم), and 3. Answer (Arabic title: ابحث عن الجواب). Each game has a green 'Edit' button in the 'Operations' column.

ID	Image	Title Ar	Title En	Type	Creation date	Operations
1		احجز الصورة	Guess the Picture	guess	2023-05-09 12:05:10	<button>Edit</button>
2		ترجم	Translate	translate	2023-05-09 12:05:10	<button>Edit</button>
3		ابحث عن الجواب	Answer	answer	2023-05-09 12:05:10	<button>Edit</button>

Figure (27) games

On this page, the content creator can modify the games info

5.(Content creator) Levels:

ID	Task	Answer 1	Answer 2	Answer 3	Answer 4	Image	Creation date	Operations
2	guess the fruit	banana	apple	manga	orange		2025-12-22 12:59:40	<button>Edit</button> <button>Delete</button>

Figure (28) levels

On this page, the site shows the Content Creator information about the levels, he can modify it.

The Dashboard was created based on the previously explained interface diagrams, and the two types of users who will use the panel were implemented. A file was built containing a permissions determination matrix that determines what pages the user is allowed to access and prevents him from accessing pages that are outside his permissions.

This way we ensure that no one has access to data outside their authority

```
$tabs=array("Content_Creator"=>1,"Players"=>1,"Reward"=>1,"games"=>1,"Match"=>1,"Answer"=>1,"Translate"=>1,
"Guess"=>1,"charts"=>1,"black_list"=>1,);
if($admin_arrays[0]['admin_level']==2)
{
    $tabs['Content_Creator']=0;//
    $tabs['Players']=0;//
    $tabs['Reward']=1;//
    $tabs['games']=1;//
    $tabs['Match']=1;
    $tabs['Answer']=1;//
    $tabs['Translate']=1;//
    $tabs['Guess']=1;//
    $tabs['charts']=1;//
    $tabs['black_list']=0;
}
else if($admin_arrays[0]['admin_level']==1)
{
    $tabs['Content_Creator']=1;
    $tabs['Players']=1;
    $tabs['Reward']=0;
    $tabs['games']=0;
    $tabs['Match']=0;
    $tabs['Answer']=0;
    $tabs['Translate']=0;
    $tabs['Guess']=0;
```

Figure (29) levels.php

If the user accesses a page that is outside his scope of authority, he will be directed to the unauthorized access page

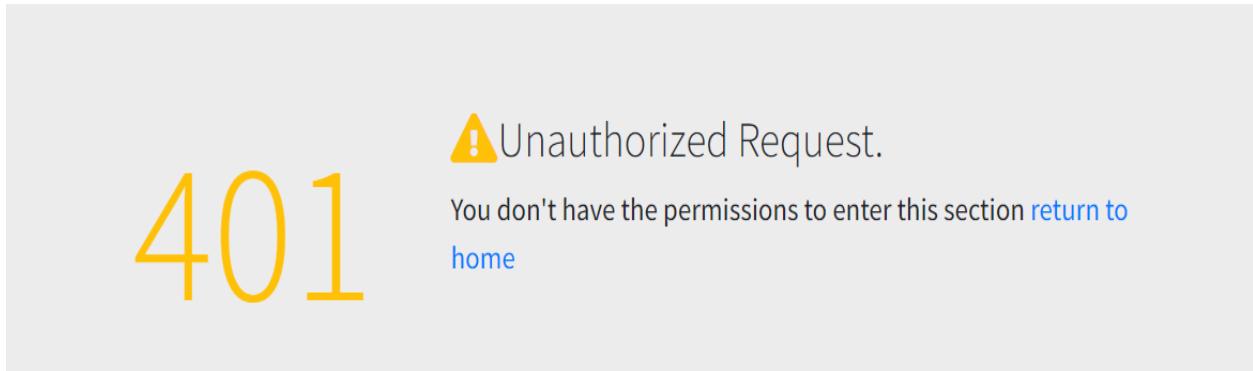


Figure (30) unauthorized

or he will be directed to the error page if he accesses any page in a wrong way

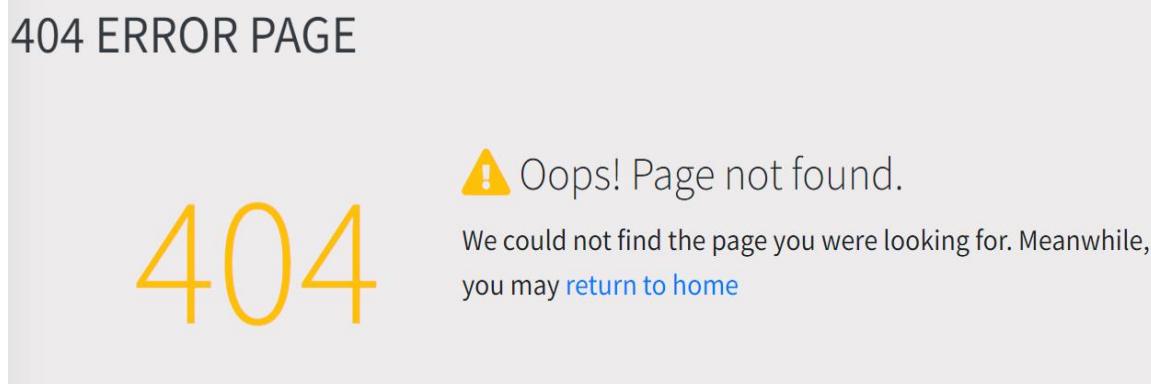


Figure (31) error

The autoload.php file has been added, which compares the user's level with his permissions within the level permissions matrix in the level.php file. It also downloads the JavaScript files related to the loaded page.

```
require('functions/levels.php');
if(isset($_GET['m']) && isset($_GET['a'])){
    if($_GET['m'] != '' && $_GET['a'] != ''){
        $m = $_GET['m'];
        $a = $_GET['a'];
        if(file_exists('modules/'.$m)){
            if(file_exists('modules/'.$m.'/'.$a.'.php')){
                $html_require = 'modules/'.$m.'/'.$a.'.php';
                $html_header = strtoupper($m);
                if(file_exists('modules/'.$m.'/script.js')){
                    $html_require_script = 'modules/'.$m().'/script.js';
                }
            }
        }
    }
}
```

Figure (32) autoload.php

The conf.php file was created and most of the variables used in most of the files were installed in order to achieve the principle of reuse in order to reduce the amount of code written and achieve the same result.

```
<?php
ini_set('memory_limit', '500M');
error_reporting();
date_default_timezone_set("Asia/Damascus");
require("db.class.php");
$full_site_url = '';
$user = 'root';
$password = '';
$server = 'localhost';
$dbname = 'language_game';
$db = new db;
$db->connect($dbname , $server , $user , $password);
```

Figure (33) conf.php

Then we created the functions that will allow us to connect to the database to bring the data from it in a class which will allow us to combine the front end and the back end together while using the conf.php information to link to the right data base

```
1 <?php
2 class db{
3     var $dbConnect;
4     function connect($db , $server = "localhost" , $user ="root" ,
5                     $pass ="" ){-
6         }
7         function select($query){-
8             }
9             function select_where($query , $where = 1){-
10                }
11                function update($table , $values , $where = 1){-
12                    }
13                    function insert($table , $values ){-
14                        }
15                        function delete($table , $where){-
16                            }
```

Figure (34) db.class.php

Testing:

Test Case	Test Case Name	Req ID	Title	Description	Test Steps	Expected Result	Actual Result	Status
TC-01	User Search Test	REQ-1	User Search	Verify the ability to search for users based on specific criteria	1. Log in as Admin 2. Go to search page 3. Enter criteria (name/email) 4. Click search	The system displays a list of users matching the criteria	Success	Executed
TC-02	User Account Edit Test	REQ-2	User Account Edit	Verify editing of an existing user account data	1. Log in as Admin 2. Search for the user 3. Click Edit 4. Change a field 5. Save	The data is updated and a success message appears	Success	Executed
TC-03	User Block Test	REQ-3	User Block	Verify blocking a user and preventing their login	1. Log in as Admin 2. Select the user 3. Click Block 4. Confirm 5. Attempt login as blocked user	The user is blocked, and the system rejects the login	Success	Executed
TC-04	Usage Reports Generation Test	REQ-4	Usage Reports Generation	Verify displaying and exporting usage reports	1. Log in as Admin 2. Go to reports section 3. Select period 4. View report 5. Export as PDF	The report displays correct data and exports successfully	Success	Executed
TC-05	Add Content Manager Test	REQ-5	Add Content Manager	Verify adding a new content manager account	1. Log in as Admin 2. Add new content manager 3. Fill in data 4. Save	The account is created with content manager permissions	Success	Executed
TC-06	Edit Content Manager Test	REQ-6	Edit Content Manager	Verify editing content manager data	1. Select content manager 2. Edit email or name 3. Save	The data is updated successfully	Success	Executed
TC-07	Delete Content Manager Test	REQ-7	Delete Content Manager	Verify deleting a content manager account	1. Select content manager 2. Click Delete 3. Confirm 4. Verify absence	The account is deleted and no longer appears in the list	Success	Executed
TC-08	Edit Games Test	REQ-8	Edit Games	Verify editing general game properties	1. Select game 2. Edit difficulty or design 3. Save	The changes are applied to the game	Success	Executed
TC-09	Add Stage Test	REQ-9	Add Stage	Verify adding a new educational stage	1. Go to Add Stage 2. Fill in content 3. Save	The stage appears in the list and becomes available	Success	Executed
TC-10	Edit Stage Test	REQ-10	Edit Stage	Verify editing stage content	1. Select stage 2. Edit words or exercises 3. Save	The stage is updated	Success	Executed

TC-11	Delete Stage Test	REQ-11	Delete Stage	Verify deleting a stage	1. Select stage 2. Confirm deletion 3. Verify absence	The stage is deleted	Success	Executed
TC-12	Add Rewards Test	REQ-12	Add Rewards	Verify adding a new reward	1. Add reward 2. Upload image and set conditions 3. Save	The reward appears in the system	Success	Executed
TC-13	Edit Reward Test	REQ-13	Edit Reward	Verify editing an existing reward	1. Select reward 2. Edit image or conditions 3. Save	The reward is updated	Success	Executed
TC-14	Delete Reward Test	REQ-14	Delete Reward	Verify deleting a reward	1. Select reward 2. Confirm deletion 3. Verify absence	The reward is deleted	Success	Executed
TC-15	Games and Stages Reports Generation Test	REQ-15	Generate Reports on Games and Stages	Verify displaying statistics for games and stages	1. Go to reports 2. Select period 3. View report	Correct data is displayed	Success	Executed
TC-16	Set Daily Goal Test	REQ-16	Set Daily Goal	Verify setting and tracking a daily goal	1. Go to settings 2. Set goal (e.g., 20 minutes) 3. Save	The goal is saved and appears on the home page	-	-
TC-17	Account Management Test	REQ-17	Account Management	Verify editing account data	1. Go to profile 2. Edit name or password 3. Save	The data is updated	-	-
TC-18	Select Game Test	REQ-18	Select Game	Verify starting the selected game	1. Display games list 2. Select game 3. Start playing	The game starts and the session is recorded	-	-
TC-19	Language Level Edit Test	REQ-19	Edit Language Level	Verify changing difficulty level	1. Go to settings 2. Select new level 3. Save	Game content changes according to the new level	-	-
TC-20	Monitor Rewards and Progress Test	REQ-20	Monitor Rewards and Progress	Verify displaying progress and rewards	1. Go to progress page 2. View charts and rewards	Correct data is displayed (points, rewards, completed stages)	-	-

Table (23) test cases

Chapter 5

Conclusion and Next Stages

Conclusion

In today's world, technology has become an essential part of daily life, especially in education. Children and parents increasingly seek modern, engaging, fast, and effective ways to learn — ways that turn education into an enjoyable experience rather than a chore. Traditional methods of teaching English often lead to boredom, low motivation, and limited progress, particularly for children aged 8–14.

This project was designed to meet these evolving needs by creating an interactive, gamified digital platform for learning English. Through our system:

- 1-Children experience fun, motivating, and rewarding learning through colourful games, animations, points, badges, levels, and interactive stories.
- 2-Parents and teachers gain powerful tools to track progress in detail, identify strengths and weaknesses, and provide personalized support.
- 3-Teachers and content creators benefit from an easy-to-use, no-code dashboard that allows continuous addition and updating of new stages, games, vocabulary, audio, and activities — keeping the content fresh, curriculum-aligned, and relevant.
- 4-The platform ensures accessibility anytime and anywhere across smart devices, breaking down geographical and time barriers.

We believe this project has successfully achieved its core goals: transforming English language learning into an enjoyable, effective, and truly child-centered experience that increases engagement, accelerates skill development (vocabulary, pronunciation, listening), and builds long-term confidence in using the language.

Future Enhancements (Senior Project 2 / Next Phase):

- 1.Implement daily login limits and streak rewards to encourage consistent practice without burnout.
- 2.Further refine and beautify the user interfaces (especially child-facing screens) to make them even more attractive, intuitive, and age-appropriate.
- 3.Add full Arabic language support for instructions, menus, and content to better serve Arabic-speaking learners.
- 4.Develop a collaborative mode for group/classroom participation (e.g., multiplayer challenges, teacher-led sessions).
- 5.Create a native Android mobile application to improve performance, offline capabilities, and push notifications for daily goals and progress updates.

With this foundation in place, we are confident that the platform has the potential to become a leading tool in the field of gamified English language learning for children in the Arab world and beyond. The journey continues — toward even more engaging, personalized, and impactful learning experiences!

Chapter6

References and Appendices:

References

- [1] Alan Dennis, Systems Analysis and Design,2012
- [2] Joel Murach, Murach's PHP and MySQL,2010.
- [3] Luke Welling, PHP and MySQL Web Development, Second Edition,2003.
- [4] Alan Forbes, The Joy of PHP: A Beginner's Guide to Programming Interactive Web Applications with PHP and MySQL,2012.
- [5] Jake Spurlock, Bootstrap: Responsive Web Development ,2013.
- [6] Matt Lambert., Learning Bootstrap, 2014.
- [7] Marijn Haverbeke, Eloquent JavaScript, 3rd Edition: A Modern Introduction to Programming, 2018.

Appendices:

Appendix A: PHP Programming Language

PHP (which stands for **PHP: Hypertext Preprocessor**) is a widely used, open-source, general-purpose scripting language especially suited for **web development**. It was originally created by Rasmus Lerdorf in 1994 as a set of Perl scripts called "Personal Home Pages" (PHP/FI). Over time, it evolved into a full-fledged programming language.

Key milestones in its history:

- In 1995, the first public release appeared.
- PHP 3 (1998) introduced a major rewrite, added object-oriented programming basics, and redefined PHP as "Hypertext Preprocessor" — a recursive acronym.
- Subsequent versions (PHP 4, 5, 7, and 8 series) brought significant improvements in performance (Zend Engine), security, modern syntax (e.g., JIT compiler in PHP 8), typed properties, attributes, enums, and more.
- As of January 2026, the latest stable versions are from the **PHP 8.5** series (released late 2025), with PHP 8.6 under active development for release by the end of 2026.

Main Advantages of PHP (relevant to our educational platform):

1. **Open-source and free** — full access to source code, allowing unlimited modifications and community-driven improvements.
2. **Easy to learn and simple syntax** — ideal for rapid development of interactive web applications like our gamified learning platform.
3. **Cross-platform** — runs seamlessly on Windows, Linux, macOS, and various web servers (Apache, Nginx, etc.).
4. **Native HTML integration** — PHP code can be embedded directly into HTML, making it perfect for creating dynamic, child-friendly interfaces.
5. **High flexibility and security** — strong support for modern security practices, sessions, authentication, and data validation — crucial for protecting children's progress data.
6. **Extensive ecosystem** — vast libraries, frameworks (Laravel, Symfony), and tools for building scalable web apps, including our content management dashboards.
7. **Strong community and documentation** — official PHP.net site and global forums provide excellent support.

In our project, PHP serves as the backend language powering the API, user authentication, content management, progress tracking, and game logic — ensuring a fast, reliable, and maintainable system.

Appendix B: MySQL Database Management System

MySQL is one of the world's most popular open-source **relational database management systems (RDBMS)**. It uses **Structured Query Language (SQL)** for managing data and organizes information in tables with defined relationships.

History and Ownership:

- Created in 1995 by Michael "Monty" Widenius (named after his daughter "My").
- Initially developed by MySQL AB (Sweden).
- Acquired by Sun Microsystems in 2008, then by Oracle Corporation in 2010 (current owner).
- MySQL remains open-source under the GNU General Public License (GPL), with optional commercial editions from Oracle.
- As of January 2026, MySQL follows a dual-track release model: **Innovation releases** (new features) and **Long-Term Support (LTS)** releases (stable, long-lived bug/security fixes). MySQL 8.0 reaches End-of-Life in April 2026; recommended production versions are **MySQL 8.4 LTS** and newer **Innovation releases** (e.g., 9.x series).

Key Features of MySQL:

- Extremely fast performance — multi-threaded server design enables quick queries, even with large datasets.
- Easy integration — works seamlessly with PHP (via PDO or mysqli), making it the ideal database for our platform's backend.
- Robust security features — user authentication, encryption, role-based access control, and protection against SQL injection.
- Support for multiple storage engines (e.g., InnoDB for transactions and ACID compliance).
- Graphical user interfaces (GUI) — free tools like **phpMyAdmin**, **MySQL Workbench**, and **Adminer** allow visual management of databases, tables, and queries — very useful for administrators maintaining our educational content.
- Scalability and reliability — handles high concurrency, replication, and backups — perfect for growing user bases of children, parents, and teachers. In our gamified English learning platform, MySQL stores all critical data: user accounts, game progress, vocabulary/stages, rewards, reports, and analytics — providing a secure, efficient, and scalable foundation for the entire system. These appendices provide technical background on the core technologies used in the development of the platform, highlighting why PHP and MySQL were chosen for their proven reliability, ease of use, performance, and strong community support in building modern educational web applications.