

A Project Report On

EPIC RPG

Submitted in partial fulfillment of the requirement for the
award of the degree
**MASTER OF COMPUTER
APPLICATIONS**
(M.C.A.)

Academic Year 2025 - 26

Adil Belim (92400584032)
Krupali Shah (92400584055)
Niyati Bhanani (92400584032)

Internal Guide

Dr Ashwin Dobariya



Marwadi
University
Marwadi Chandarana Group





Marwadi
University
Marwadi Chandarana Group



Faculty of Computer Applications (FCA)

Certificate

*This is to certify that the project work entitled
EPIC RPG*

*submitted in partial fulfillment of the requirement for
the award of the degree of*

**MASTER OF COMPUTER APPLICATIONS
(M.C.A.)**

of the

Marwadi University

is a result of the bonafide work carried out by

Adil Belim (92400584032)

Krupali Shah (92400584055)

Niyati Bhanani (92400584019)

during the academic year 2025-26

Faculty Guide

HOD

Dean

Certificate

DECLARATION

We hereby declare that this project work entitled **EPIC RPG** is a record done by us.

We also declare that the matter embodied in this project is genuine work done by us and has not been submitted whether to this University or to any other University / Institute for the fulfillment of any course of study.

Place:

Date:

Adil Belim (92400584032) Signature:

Krupali Shah (92400584055) Signature:

Niyati Bhanani (92400584019) Signature:

ACKNOWLEDGEMENT

It is indeed a great pleasure to express our thanks and gratitude to all those who helped us. No serious and lasting achievement one can ever achieve without the help of friendly guidance and co-operation of so many people involved in the work.

We are very thankful to our faculty guide **Dr Ashwin Dobariya** the person who makes us follow the right steps during project work. we express our deep sense of gratitude to his guidance, suggestions and expertise at every stage. Apart from that his valuable and expertise suggestion during documentation of our report indeed helped us a lot.

Thanks to our friends who have been a source of inspiration and motivation that helped us during my/our project work. We are heartily thankful to our external guide **Mr Anas Basha** for providing us an opportunity to work over this report and for their endless and great support. We are also thankful to our Head **Dr.Sunil Bajeja** and Dean **Dr Rijwan Khan** for their support as and when required. we would like to extend our gratitude towards all who directly or indirectly supported and helped us to accomplish my/our project work.

Adil Belim (92400584032)
Krupali Shah (92400584055)
Niyati Bhanani (92400584019)

Signature :
Signature :
Signature :-

COMPANY PROFILE

GALMINE AI

Galamine AI is a technology-driven company specializing in Artificial Intelligence solutions that help organizations automate processes, improve customer engagement, and enhance operational efficiency. The company develops AI-powered chatbots, virtual assistants, and customized digital solutions to support business growth and innovation.

Galamine AI operates under Amri Foundation, an organization committed to innovation, social impact, and sustainable development through technology and education.

Vision and Mission

Vision

To become a leading provider of intelligent AI solutions that empower businesses through innovation and automation.

Mission

- To design and develop AI-driven software solutions for real-world business problems.
- To deliver reliable, scalable, and user-friendly digital products.
- To promote the use of artificial intelligence for improving operational efficiency.

Services Offered by Galamine AI

Galamine AI provides a wide range of IT and AI-based services, including:

- **Artificial Intelligence Solutions**
Development of AI-powered tools such as chatbots, virtual assistants, and intelligent agents.
- **Custom Software Development**
Designing and developing customized web and application-based solutions according to business requirements.
- **Web Application Development**
Creation of responsive and user-friendly websites and web applications.
- **CRM and Business Automation Tools**
Development of customer relationship management systems and automation tools for managing business operations.
- **AI Integration Services**
Integration of AI technologies into existing systems to enhance functionality and performance.

CONTENTS

Chapters	Particulars	Page No.
1	SYNOPSIS	1
2	PREAMBLE 2.1 General Introduction 2.2 Statement of Problem 2.3 Objective and Scope of the Study 2.4 Module Description with functionality 2.5 Methodology 2.6 Feasibility Study	2
3	REVIEW OF LITERATURE 3.1 Review of Literature :- Basic Theory of the Game 3.2 Study of Existing Similar Systems :- Zork	4
4	TECHNICAL DESCRIPTION 4.1 Hardware Requirement 4.2 Software Requirement	5
5	SYSTEM DESIGN AND DEVELOPMENT 5.1 Flowchart 5.2 Data Flow Diagram (DFD) 5.3 DFD Level 0 5.4 DFD Level 1 5.5 Use Case Diagram 5.6 Sequence Diagram 5.7 Activity Diagram 5.8 Menu Design 5.9 Screen Design	6
6	SYSTEM TESTING 6.1 Testing and Implementation 6.2 Testing Methodology 6.3 Test Case Design 6.4 System Implementation Strategy	13
7	CONCLUSION	18
8	LEARNING DURING PROJECT WORK	19
9	BIBLIOGRAPHY Online References	20

FIGURE INDEX

Sr. No.	Figure No.	Particulars	Page No.
1	Figure 5.1	Flow chart	6
2	Figure 5.2	DFD Level 0	7
3	Figure 5.3	DFD Level 1	7
4	Figure 5.4	Use Case Diagram	8
5	Figure 5.5	Sequence Diagram	9
6	Figure 5.6	Activity Diagram	10
7	Figure 5.8	Screen Design	12
8	Figure 5.9	Hunt	13
9	Figure 5.10	Adventure	13
10	Figure 5.11	Inventory	14
11	Figure 5.12	Profile	14
12	Figure 5.13	Store	15
13	Figure 5.14	Chop	15
14	Figure 5.15	Reward	15
15	Figure 5.16	Craft	15

Table Index

Sr No.	Table No.	Particular	Page No.
1	Table 6.4	Taste case	16

1. SYNOPSIS:

- The Text-Based Python RPG Game is a command-line game developed using Python, where players can register, log in, create their character, explore different areas, fight enemies, collect items, and gain experience as the game progresses.
- The gameplay is intentionally kept simple so that both playing the game and understanding how it is built feels easy and enjoyable.
- The project is structured in a modular way, with different files handling specific parts of the game such as user management, game logic, inventory system, store operations, and utility functions. This approach keeps the code clean and well-organized.
- Player data like username, password, character stats, experience points, coins, and inventory is stored using JSON files, allowing the game to save progress and continue smoothly.
- The main motive behind this project was to make gaming easy and to show that game development does not always require complex graphics or advanced tools. Through this project, it is demonstrated that a complete and engaging game can be developed using simple logic and basic Python concepts.
- The focus was not only on the final game but also on making the process of game development understandable and enjoyable.
- Being a text-based game, it can run on any system with Python installed and does not require special hardware. This project clearly shows how fundamental programming concepts such as functions, loops, conditional statements, file handling, and modular coding can be used to build a real-world application.
- Overall, the project helped strengthen logical thinking, creativity, and confidence in developing games and applications in a simple yet effective way.

2. PREAMBLE

2.1 General Introduction:

- Text-based RPGs (Role-Playing Games) are one of the earliest forms of computer games, built entirely through text prompts and user choices. These games are simple yet highly interactive, relying on decision-making rather than graphics.

2.2 Statement of Problem:

- Many beginner programmers find it difficult to understand how real games manage player progress, combat, and rewards. Most examples available are either too simple to be engaging or too complex to understand. This project aims to solve that problem by creating a simple, text-based RPG using basic Python, where core game mechanics like combat, crafting, inventory, and progression are easy to follow, modify, and learn from

2.3 Objective and Scope of the study:

- Implement user registration and login functionality.
- Created a system for character level progression and attribute management (health, XP, coins).
- Design and developed various game modules, including hunting, adventure, inventory management, a store, and character profiles.
- Incorporate file-based storage for user data persistence.
- Provide a user-friendly text-based interface with clear instructions and feedback.

• Scope:

- The project will include core RPG elements such as character development, combat, and inventory.
- The game will be text-based, focusing on gameplay mechanics rather than graphical presentation.
- User data will be stored in a JSON file.
- The game will be developed using Python 3.x.

2.4 Module Description with functionality:

2.4.1 User Management Module:

- Description: This module handles user registration, login, and deletion.
- Functions:
 - `register()`: Creates a new user account, storing the username and password.
 - `login()`: Authenticates existing users and retrieves their data.
 - `delete_user()`: Removes a user account from the system.
 - `load_user_data()`: Loads user data from the JSON file.
 - `save_user_data()`: Saves user data to the JSON file.

2.4.2 Game Mechanics Module:

- Description: This module contains the core gameplay functions, including hunting,
- adventure, chopping, and reward systems.
- Functions:
- `rpg_hunt()`: Simulates a hunting expedition where the player encounters creatures, gains rewards, and loses health.
- `rpg_adventure()`: Presents a more challenging adventure scenario with a single, powerful creature.
- `rpg_chop()`: Allows the player to chop wood and gather resources.
- `reward()`: Provides random rewards to the player.
- `get_level()`: Calculates the player's level based on their XP.
- `check_level_up()`: Checks if the player has leveled up and updates their stats

2.4.3 Inventory Module:

- Description: Manages the player's items.
- Function:
- `view_inventory()`: Displays the items currently in the player's inventory.

2.4.4 Profile Module:

- Description: Displays the player's character information.
- Function:
- `view_profile()`: Shows the player's username, level, stats (attack, defense, health), coins, and XP.

2.4.5 Store Module:

- Description: Allows the player to buy items from a store.
- Function:
- `store()`: Displays the store's inventory and handles purchases.

2.4.6 Main Game Module:

- Description: The main entry point of the game, orchestrating the game loop and user interaction.
- Function:
- `start_game()`: Manages the main game loop, user input, and calls other modules.

2.4.7 Utils Module:

- Description: Provides utility functions and data structures used across other modules.
- Functions/Data:
- `get_level()`: Calculates the player's level.
- `item_emojis`: A dictionary mapping item names to emojis for display.

3. Review of Literature

3.1 Review of Literature :- Basic Theory of the Game

- Role-playing games work on the idea that a player starts with limited abilities and improves over time by playing the game. As the player performs actions like fighting enemies or collecting items, they gain experience, level up, and become stronger.
- Most RPGs use concepts such as health, experience points, inventory, and equipment to show player progress. This project follows the same basic idea but keeps everything text-based so the logic behind the game is easy to understand and learn

3.2 Study of Existing Similar Systems :- Zork

- One of the earliest examples of a similar game is Zork, a popular text-based adventure game released in the late 1970s. In Zork, players explored a fantasy world by typing commands, solving puzzles, and collecting items.
- It did not use graphics, but still managed to be engaging and immersive. This project takes inspiration from games like Zork by using text interaction, while also adding modern RPG features such as combat, levelling, crafting, and inventory management. The focus is on simplicity and learning rather than complex graphics.

4. Technical Description

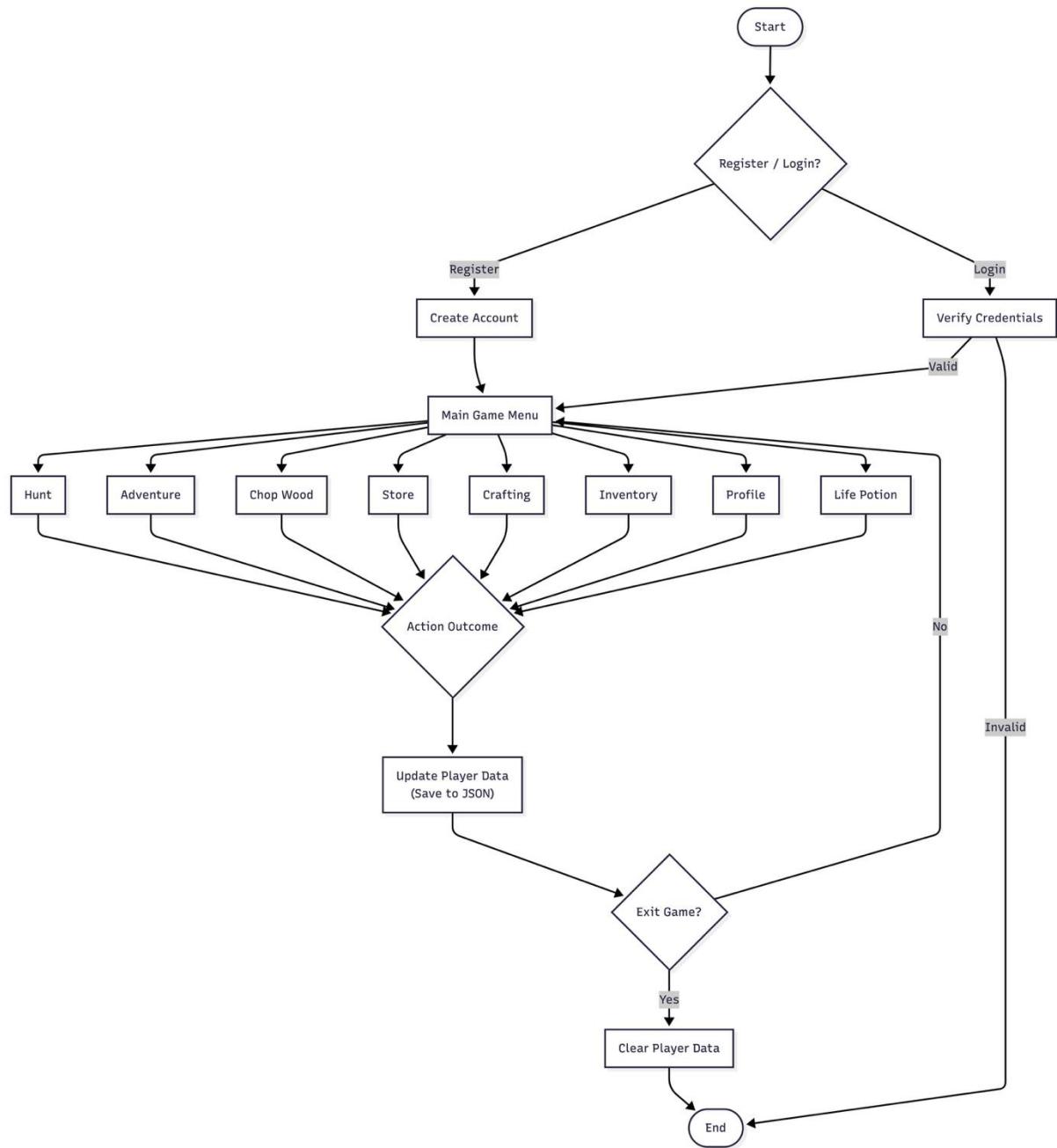
4.1 Hardware Requirements

- Processor: Intel / AMD processor or equivalent
- RAM: Minimum 2 GB (4 GB or more recommended)
- Storage: 200–300 MB free disk space to support future game data, save files, logs, and feature expansions
- Display: Terminal or console window
- Input Device: Keyboard

4.2 Software Requirement

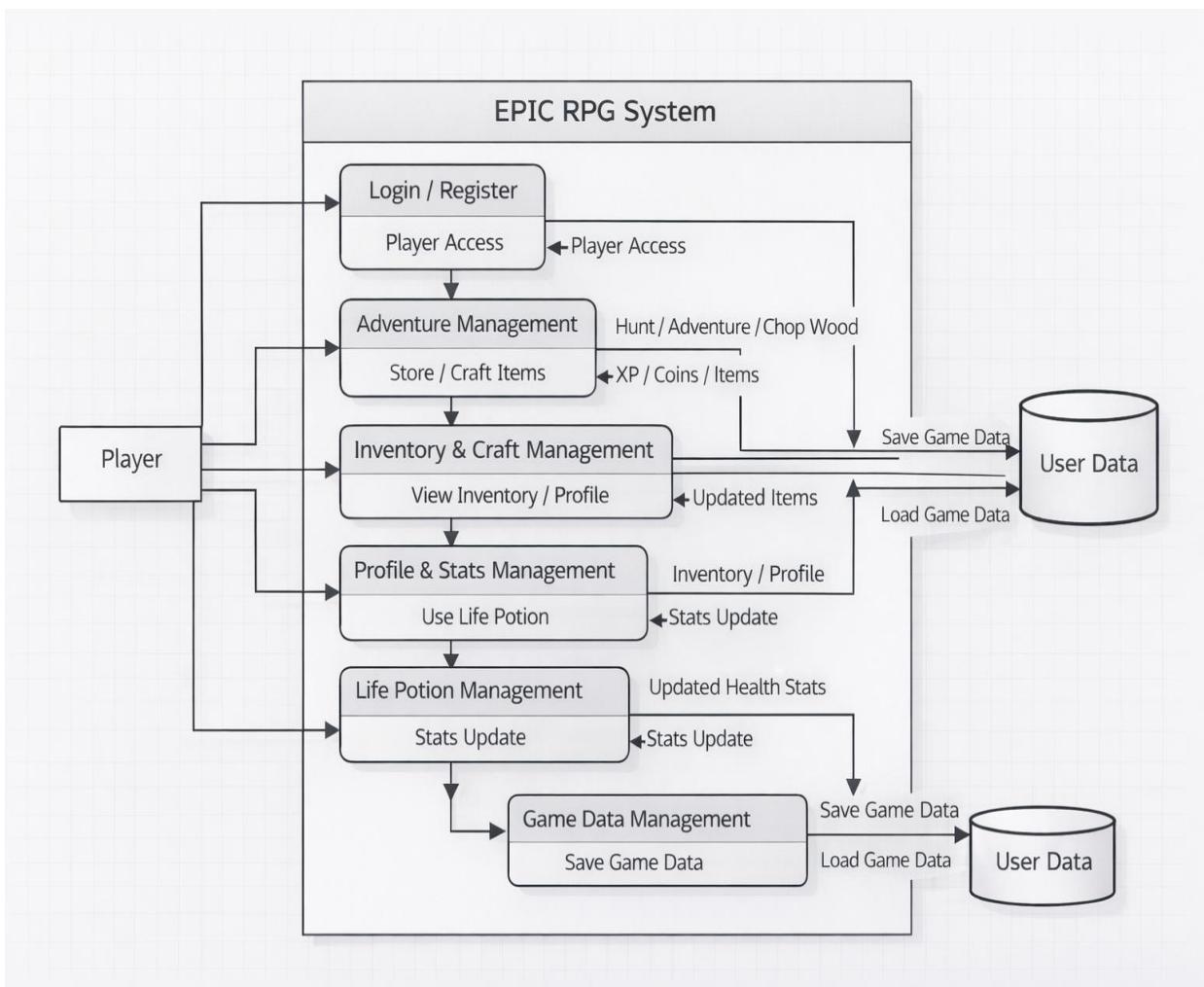
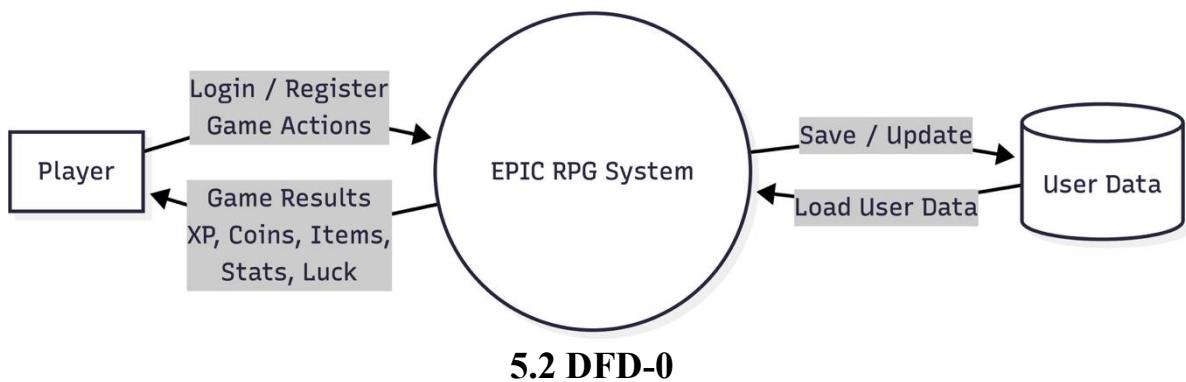
- Windows: Windows 10 or later
- Linux: Modern Linux distributions
- macOS: macOS 11 or later

5. System Design and Development

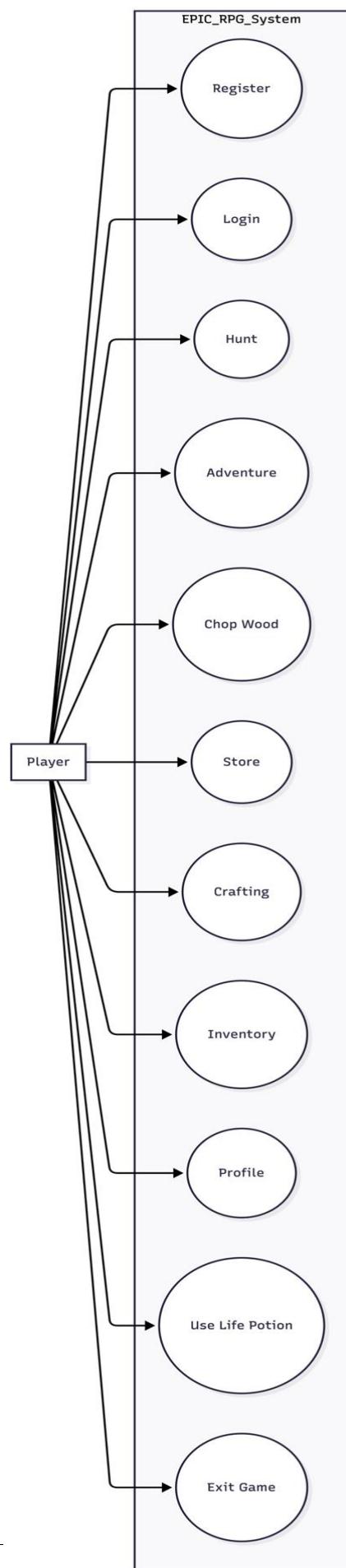


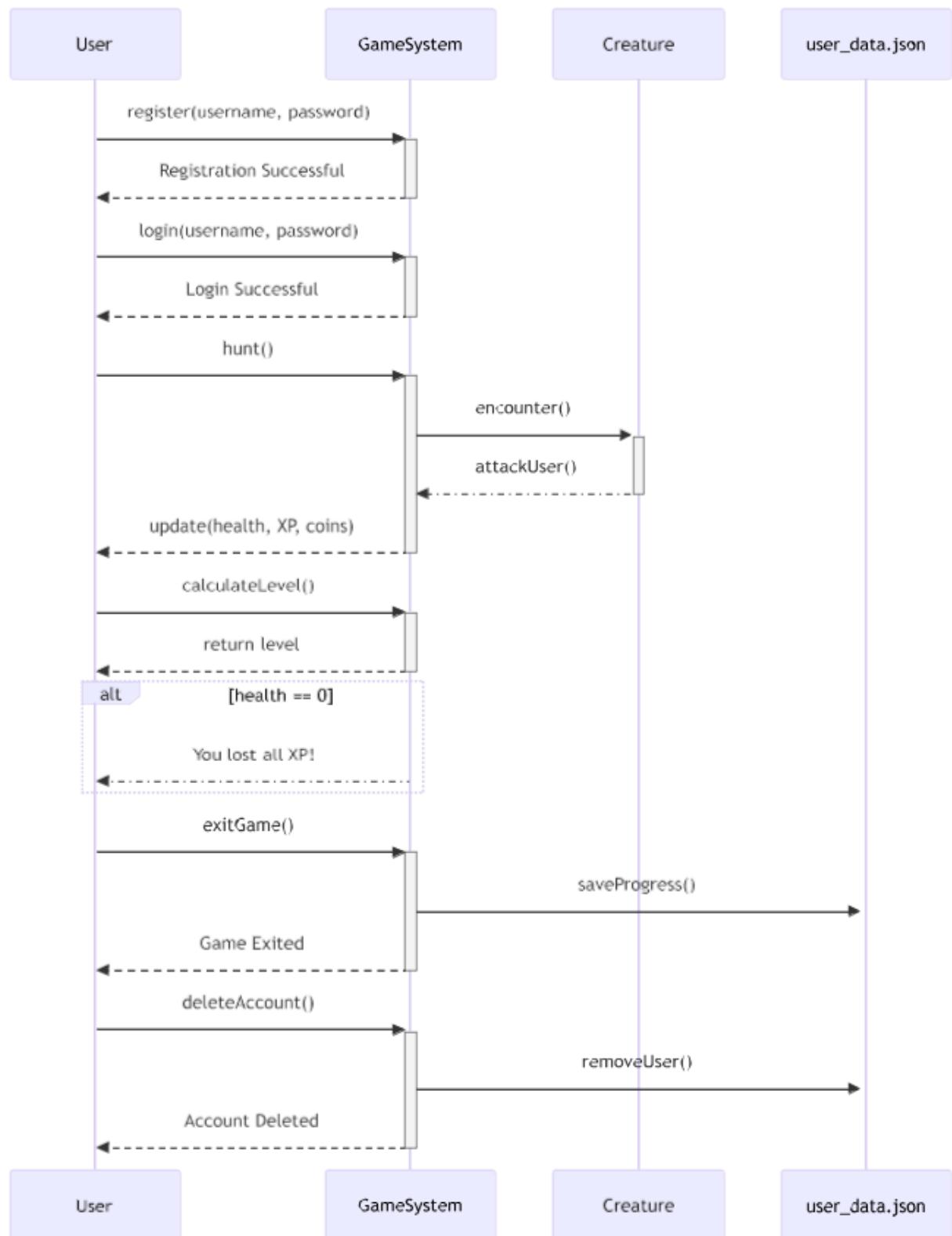
5.1 Flowchart:

Data flow diagram:



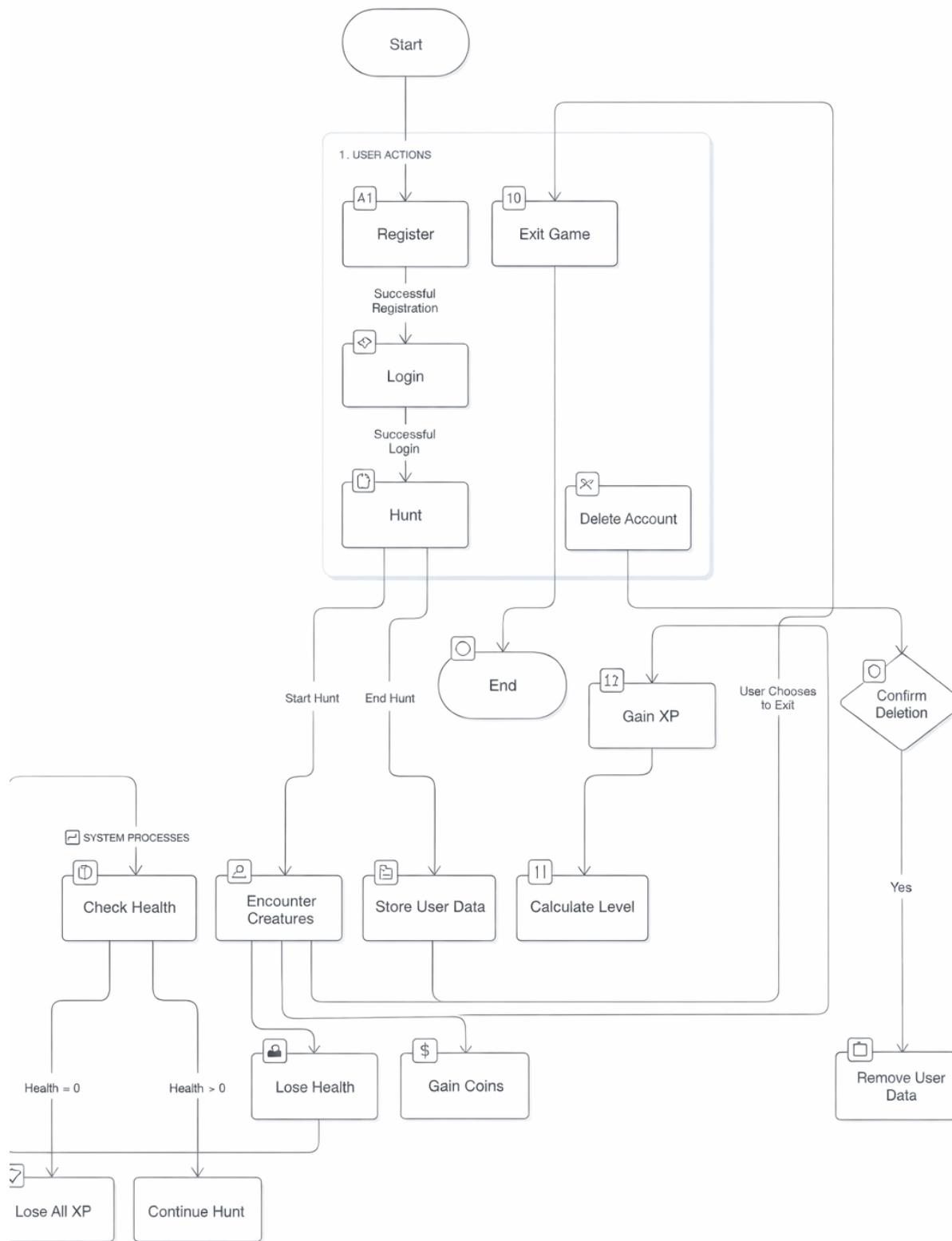
5.4 Use case diagram





5.5 Sequential diagram

RPG Game System Flowchart



5.6 Activity diagram:

5.7 Menu Design

- The menu is the main place where the player controls the game. It shows all the available actions in a simple numbered list so the player can easily decide what to do next.
- The player just enters the number of the option they want, and the game performs that action. After the action is completed, the menu is shown again so the player can continue playing without any confusion.

MENU

1.  Hunt
2.  Adventure
3.  Heal using Life Potion
4.  View Inventory
5.  View Profile
6.  Store
7.  Chop Wood
8.  Reward
9.  Craft
10.  Delete User
11.  Exit

5.8 Screen Design

```
adil@192 RPG % python3 main.py
```

Menu:

1. Register
2. Login

```
Enter your choice: 2
```

```
Enter your username: a
```

```
Enter your password: a
```

```
Login successful! Welcome, a!
```

Menu:

1. Hunt
2. Adventure
3. Heal using Life Potion
4. View Inventory
5. View Profile
6. Store
7. Chop Wood
8. Reward
9. Craft
10. Delete User
11. Exit

```
Enter your choice: █
```

Figure 5.9

Hunt:

- During gameplay, the player hunts a deer enemy. The system updates health, XP, coins, and inventory in real time. Successful encounters contribute to player progression.



Figure 5.10 Hunt

Adventure:

- The player fights a Demonic Hydra, takes damage, and defeats the enemy. Rewards include XP, coins, and a level increase.

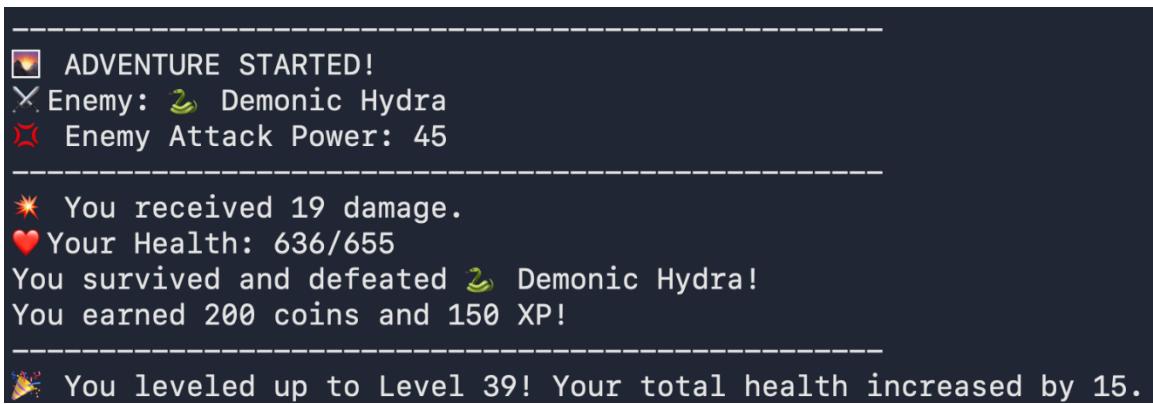


Figure 5.11 Adventure

Inventory:

- The inventory module maintains a list of all items obtained from battles and exploration.
Each item is stored with quantity values and used for healing, crafting, or combat enhancement.

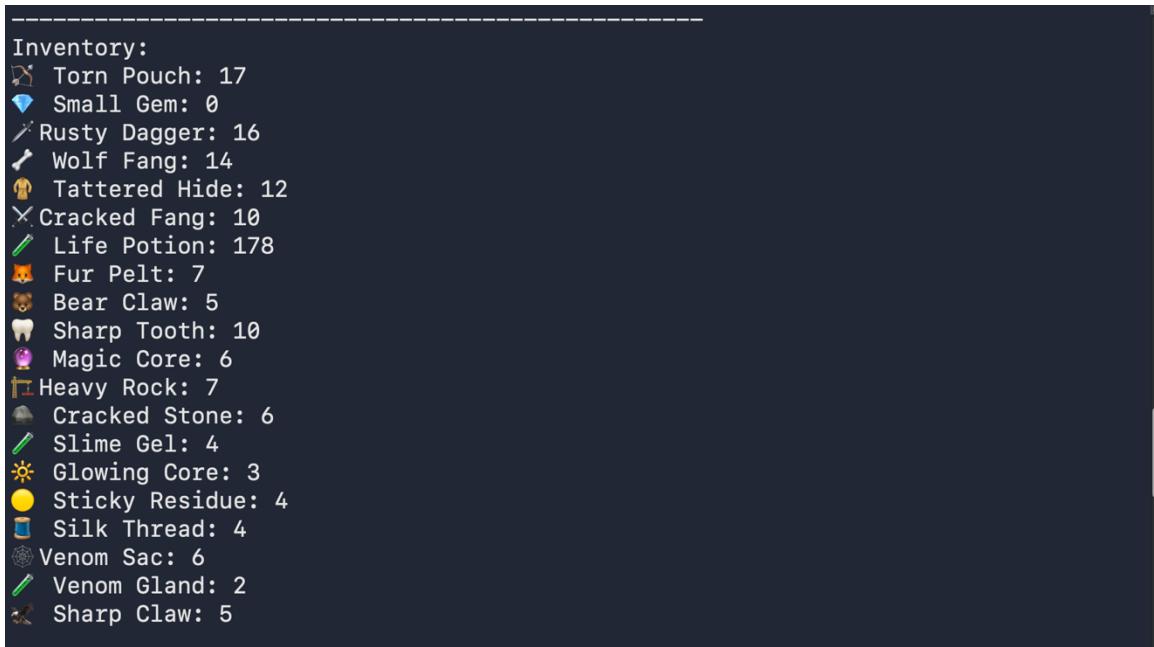


Figure 5.12 Inventory

Profile:

- The player profile module maintains real-time character attributes such as combat stats, equipment, and resources. All values are dynamically updated based on gameplay actions.

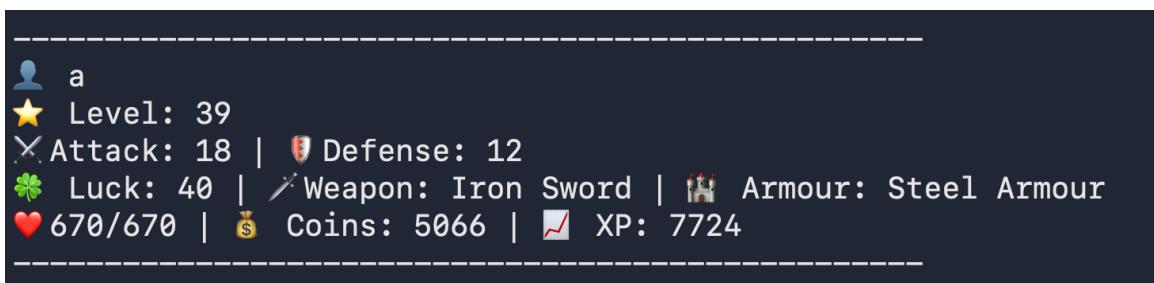


Figure 5.13 Profile

Store:

- The player can buy weapons, armour, and life potions from the store. Coins are deducted based on the selected item.

Store:

💰 Coins: 5066
1. 🪵 Wooden Sword: 100 coins
2. ✕ Iron Sword: 300 coins
3. 🛡 Basic Armour: 200 coins
4. 🛡 Steel Armour: 400 coins
5. 🧡 Life Potion: 20 coins

Figure 5.14 Store

Chop:

- The player gathers resources by interacting with the environment. Obtained items are added to the inventory.

🪓 You chopped down a tree and obtained: 🪵 Pine Log

Figure 5.15 Chop

Reward:

- Reward system showing loot distribution after successful completion of an action.

🎁 You received a reward!
💰 Coins: 396
🧪 Life Potions: 6

Figure 5.16 Reward

Craft:

The player crafts new items using collected resources. Each item requires specific materials.

Crafting:

1. Wooden Sword (needs 5 🪵 Oak Logs)
2. Lucky Charm (needs 3 💎 Small Gems)
3. Exit

Figure 5.8.9 Craft

6. System Testing

6.1 Testing and Implementation

- Testing was done throughout the development of the project by running the game in the terminal after adding new features. Each time a feature was implemented, the game was played to check if it was working properly.
- When errors occurred, such as crashes, wrong outputs, or logical mistakes, the code was reviewed and fixed immediately. This ongoing testing helped keep the game stable while new features were being added.

6.2 Testing Methodology

- Manual testing was followed for this project. Since the game runs entirely in the command line, testing was done by selecting different menu options and observing how the system responds.
- Various situations were tested, including normal gameplay, invalid inputs, missing inventory items, cooldown limits, and login failures. This helped ensure that the game handles user actions correctly and does not break during gameplay.

6.3 Test Case Design

- Test cases were created based on the options available in the game menu. Each test case focused on checking one feature at a time and observing how player data such as health, experience points, inventory items, and overall statistics were updated during gameplay.

Test Case ID	Menu Option	Description	Expected Result
TC-01	Register	Create a new user account	User account created successfully
TC-02	Login	Login with valid credentials	User logged in
TC-03	Login	Login with invalid credentials	Access denied
TC-04	Hunt	Perform hunt action	XP, coins, and items updated
TC-05	Adventure	Perform adventure	Health reduced or rewards gained
TC-06	Chop Wood	Chop wood	Wood logs added to inventory
TC-07	Store	Buy weapon or item	Inventory updated, stats affected
TC-08	Craft	Craft Wooden Sword	Weapon crafted, attack increased
TC-09	Craft	Craft Lucky Charm	Luck increased, cooldown applied
TC-10	Heal	Use Life Potion	Health restored
TC-11	Inventory	View inventory	Items displayed correctly
TC-12	Profile	View profile	Player stats displayed
TC-13	Reward	Claim reward	XP or coins added
TC-14	Delete User	Delete account	User data removed
TC-15	Exit	Exit game	Program terminates safely

Table No. 6.4

6.5 System Implementation Strategy

- The system was implemented in a step-by-step manner to keep the development process simple and manageable. Basic features such as user registration, login, and menu navigation were implemented first.
- Once these core features were stable, gameplay functions like hunting, adventure, inventory handling, and profile viewing were added.
- Advanced features such as the store, crafting system, cooldown mechanisms, and luck attribute were implemented later. After adding each feature, the system was tested through the terminal to ensure correct behaviour.
- This gradual implementation approach made it easier to identify errors, apply fixes, and maintain overall stability of the game.

7. Conclusion

- This project helped in understanding how a role-playing game works at a basic level using Python. By building a text-based RPG, important programming concepts such as functions, data handling, file storage, and logic flow were learned and applied in a practical way.
- Features like combat, inventory management, crafting, and player progression were implemented step by step, keeping the system simple and easy to understand.
- Working on this project also improved problem-solving skills, as many errors were encountered during development and fixed through testing and debugging. Overall, the project successfully meets its objective of creating a simple yet functional RPG system while providing a good learning experience in Python programming

8. Learning During Project Work

- Working on this project was a valuable learning experience. It helped in understanding how to break a large problem into smaller parts and implement them step by step.
- Through this project, practical knowledge of Python programming was gained, especially in areas like file handling, functions, dictionaries, and program flow control.
- The project also improved problem-solving skills, as many errors and logical issues were faced during development and resolved through testing and debugging. Additionally, it provided experience in designing a system, planning features, and improving code gradually.
- Overall, this project helped build confidence in programming and gave a better understanding of how real applications are developed.

9. Online References

1. **W3Schools, Python Tutorial.**
 - ✓ Available at: <https://www.w3schools.com/python/>
 - ✓ Referred for understanding Python basics such as functions, dictionaries, loops, and file handling
2. **EPIC RPG Wiki, EPIC RPG Game Guide.**
 - ✓ Available at: https://epic-rpg.fandom.com/wiki/EPIC_RPG_Wiki
 - ✓ Referenced for understanding RPG game concepts, items, progression, and gameplay mechanics

This project marks the end of an important phase of my academic journey. Building this game was not just about writing code, but about learning, experimenting, failing, fixing mistakes, and improving step by step. It reflects the effort, patience, and growth I

experienced throughout my final year, making this project a memorable and meaningful conclusion to my studies.