Raja Fatash Abbasi
SP22 - BSE - 127

Project Title: Adventure Hunt Puzzle

Actors:
* Player
* Game System

Use Cases: * Resume Game
* Register Player
* Login Player
* Start Game
* Solve Puzzle
* Gent Hint
* View Score
* View Leaderboader
* Enit Game
* Pause Game
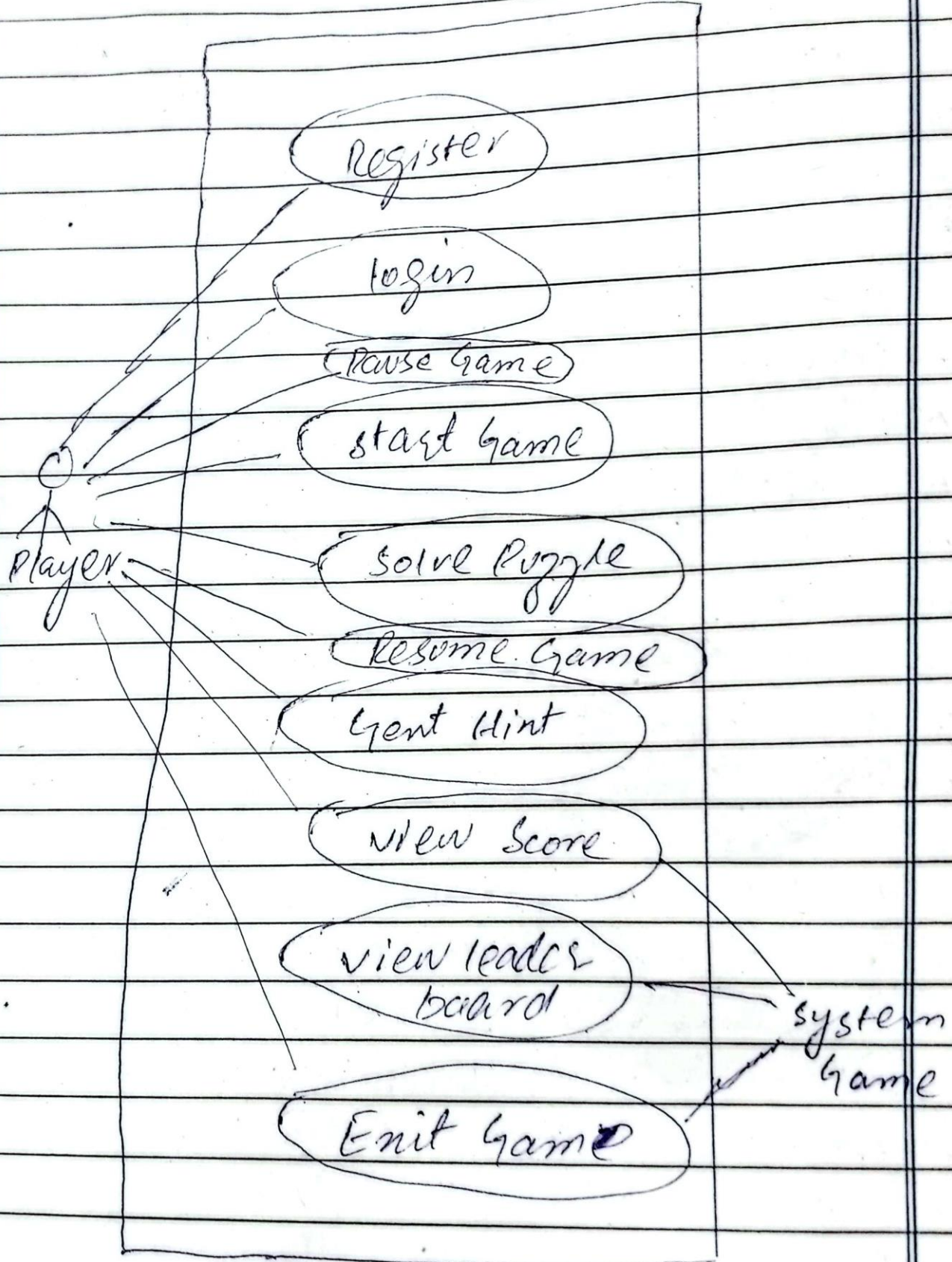
Relationships:
* Player → Register Player
* Player → login Player
* Player → Start Game
* Player → Solve Puzzle
* Player → Gent Hint
* Game System → View Score
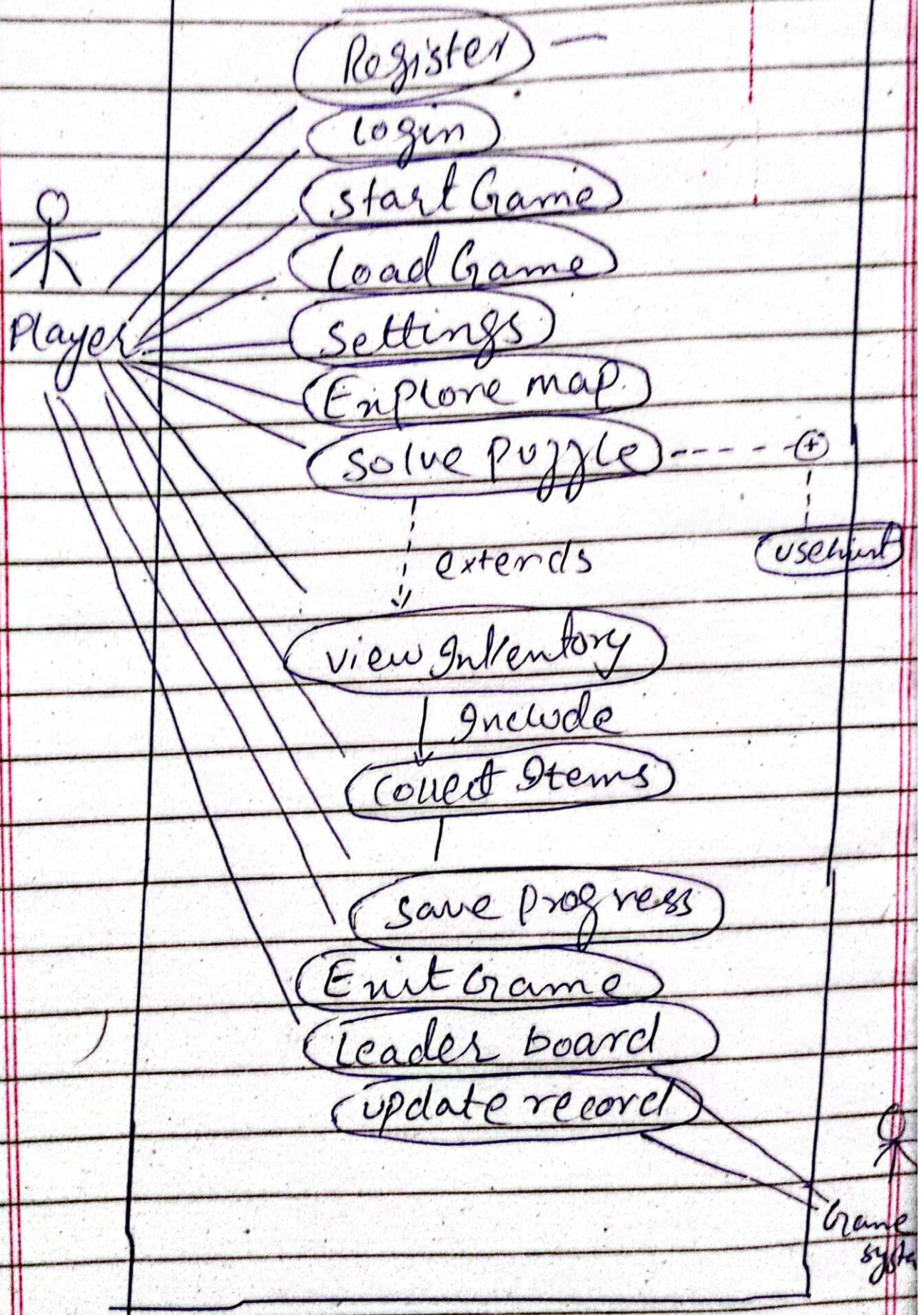* Game System → View leader board

Diagram: w:

Adventure Hunt Puzzle

Register

Login

Pause Game

Start Game

Player

Solve Puzzle

Resume Game

Gent Hint

View Score

view leader board

System Game

Exit Game

Use Case Diagram

Fully Dressed Use Case

Register —

Login

start Game

load Game

Settings

Explore map

Solve Puzzle - - - - - ⊕ ---- use hint

extends

view Inventory

Include

Collect Items

save progress

Exit Game

leader board

update record

Player

Game system

SSD For "Start New Game"

Actor: Player:

'←press "start Game"'

'←display Game map'

'←select difficulty'

SSD For "Solve Puzzle"

'←tap on puzzle'

'←show puzzle interface'

'←enter solution'

SSD For "Use Hint ."

'←press hint'
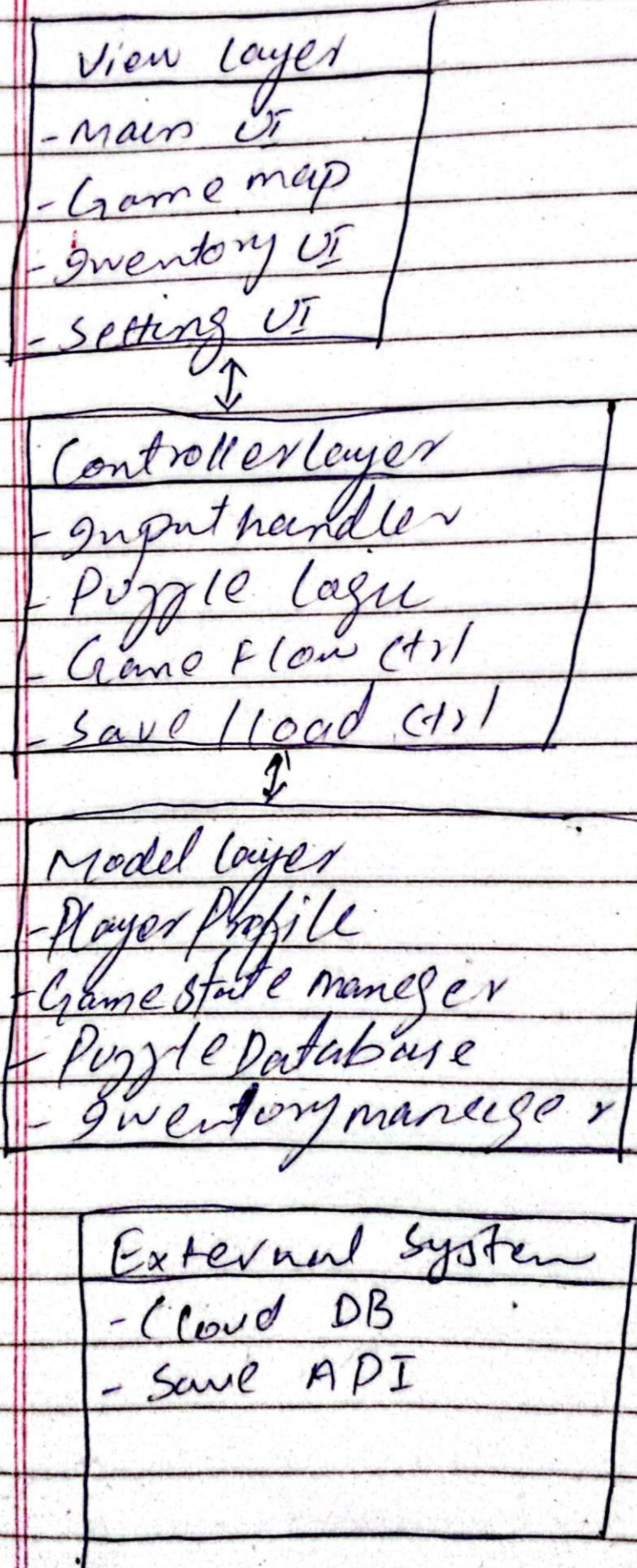
'←check hint availability'

'←show hint text/visual'

Model View Controller:

MVV architecture is ideal because

→ model handles game logics
→ view manges visual display
→ controller manges user input

Package Diagram:

```
┌─────────────────────┬──────────┐
│ View Layer          │          │
│ - Main UI           │          │
│ - Game map          │          │
│ - Inventory UI      │          │
│ - Setting UI        │          │
└─────────────────────┴──────────┘
            ↕
┌─────────────────────┬──────────┐
│ Controller Layer    │          │
│ - Input handler     │          │
│ - Puzzle logic      │          │
│ - Game Flow Ctrl    │          │
│ - Save/load Ctrl    │          │
└─────────────────────┴──────────┘
            ↕
┌─────────────────────┬──────────┐
│ Model Layer         │          │
│ - Player Profile    │          │
│ - Game state manager│          │
│ - Puzzle Database   │          │
│ - Inventory manager │          │
└─────────────────────┴──────────┘

┌─────────────────────┬──────────┐
│ External System     │          │
│ - Cloud DB          │          │
│ - Save API          │          │
│                     │          │
└─────────────────────┴──────────┘
```

System Events:

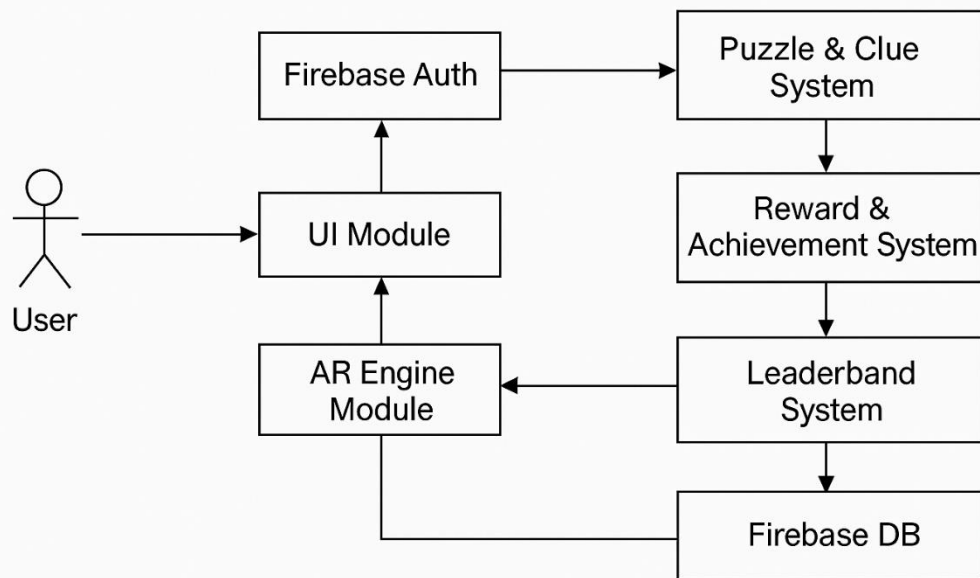| UseCase	(Event) Actor Action	System Resp
Register	Enter user detail	Validate
Login	Login (username)	Validate Credi
Start Game	start New Game()	initialize Gane
Solve Puzzle	Submit solution	check correctness
Use hint	request init	fetch & displa
collect items	collect items	add items
Save Game	load save Game	Persist current
load Game	update save Game	fetch last save
Adjust	update settings	apply new settin

Class Diagram:

Game Manager

- current level : init
- game state : Gamestate

+ Start Game()

+ Save Game()

+ load Game()

Puzzle Manager

- Puzzle list : list

+ check Answer()

+ getHint()

uses →

↓ manages

PlayerProfile

- user Name : string
- Inventory : List
- Progress : int

+ addItem()

+ get Progress

uses →

Puzzle

- Puzzle_id : init
- question : string
- answer : string
- hint : string

+ validate Answer()

+ getHint()

Items

- itemID : int
- name : string
- description : string

+ use Item()

settings

- volume : int
- difficulty : string
- language : string

+ update setting

Fig SEQUENCE DIAGRAM

COMMUNICATION DIAGRAM

FULLY DRESSED USE CASE

Use Case	Play Game
Primary Actor	Player
Stakeholders and Interests	Player: Wants to engage in and complete a treasure hunt
Preconditions	The player is authenticated
Main Success Scenario	1. Open the app. 2. Start the game. 4. Navigate to ⟩a⟨location at locations. 5. View awarded earned rewards. 6. View updated leaderboard
Extensions	2a. Player cancels starting the game 4a. Player fails to solve a puzzle 5a. Cannot navigate to iocalt. location
Frequency of Use	Multiple times a day
Open Issues	None