

Software Architecture

Exercise – Gaming Platform (iteration #2)

BSc

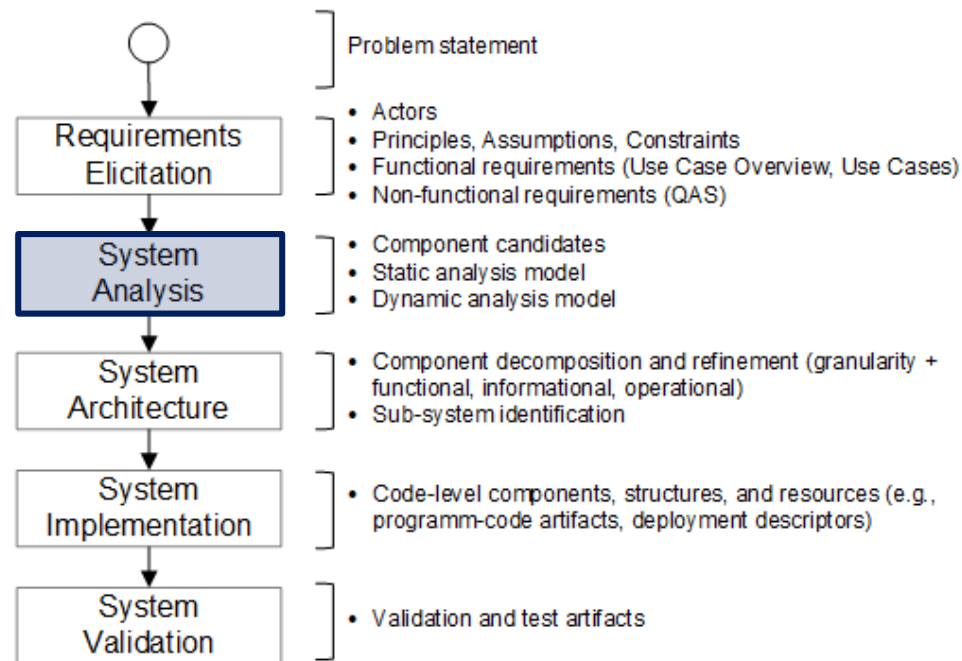


Ingo Arnold

Exercise Opening

Motivation

You **create an initial design** for a **gaming platform** by incrementally following the process outlined, below. Note that your design should focus less on the algorithmic and more on the structural solution aspects.



Exercise Opening

Motivation

Continue **refining your initial set of requirements** where this seems reasonable to you.

Next, analyze the requirements and **create an analysis model** of the component candidates that highlights their dynamic relationships accordingly (i.e., logical view).

Exercise Agenda



- Requirements Elicitation
- System Analysis
- System Architecture
- System Implementation

Requirements Elicitation

Use Case Overview and Use Cases

In this second iteration, you build on the results of the first iteration. First, examine whether the identified use cases are too coarsely grained, and whether they should be broken down into smaller use cases.

Next, create the flow of events section for any additional use cases.



Requirements Elicitation

Use Case Overview and Use Cases

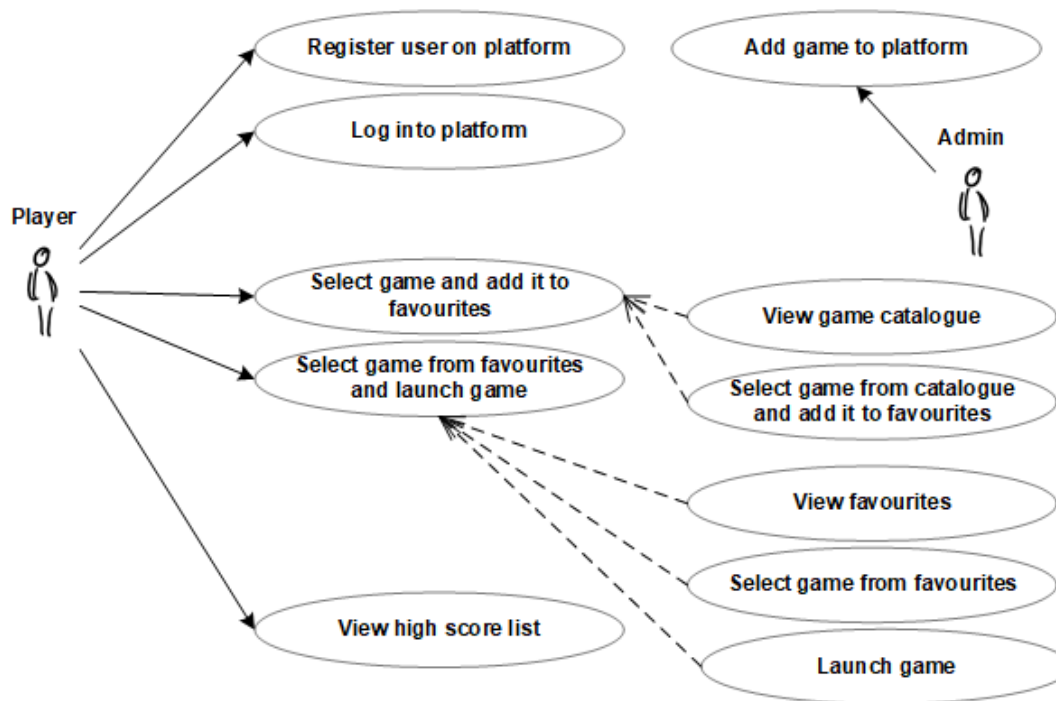
Create an **use case overview diagram** for the gaming platform that includes the refined set of use cases.



Requirements Elicitation

Use Case Overview and Use Cases

Create an **use case overview diagram** for the gaming platform that includes the refined set of use cases.





Requirements Elicitation

Use Case Overview and Use Cases

Create **use case descriptions** for all additional use cases (focus on *flow of events*).

Use Case Name

Actor

Flow of Events

Precondition

Postcondition

Qualities



Requirements Elicitation

Use Case Overview and Use Cases

Use case **view game catalogue**.

Use Case Name	View game catalogue
Actor	Player, Admin
Flow of Events	<ol style="list-style-type: none">1. Actor opens the game catalogue.2. Actor views the game catalogue contents.
Precondition	Player is successfully logged into platform. Game catalogue is not empty.
Postcondition	Game catalogue was viewed by an actor.
Qualities	



Requirements Elicitation

Use Case Overview and Use Cases

Use case **select game from game catalogue and add it to favourites.**

Use Case Name	Select game from catalogue and add it to favourites
Actor	Player
Flow of Events	<ol style="list-style-type: none"> 1. Actor selects game from game catalogue. 2. Selected game is added to actor's favourites list.
Precondition	Player is successfully logged into platform. Game catalogue is not empty and selected game is not yet in the player's favourites list.
Postcondition	Game is added to actor's favourites list.
Qualities	

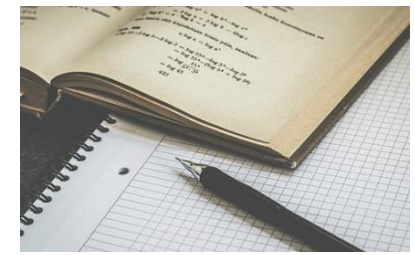


Requirements Elicitation

Use Case Overview and Use Cases

Use case **view favourites**.

Use Case Name	View favourites
Actor	Player
Flow of Events	<ol style="list-style-type: none">1. Actor opens the favourites list.2. Actor views the favourites list contents.
Precondition	Player is successfully logged into platform. favourites list is not empty.
Postcondition	Favourites list was viewed by an actor.
Qualities	

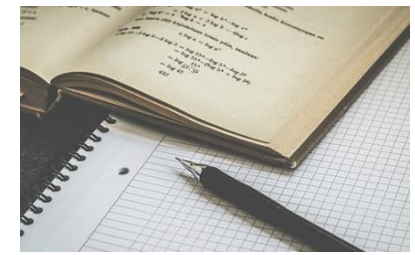


Requirements Elicitation

Use Case Overview and Use Cases

Use case **select game from favourites**.

Use Case Name	Select game from favourites
Actor	Player
Flow of Events	<ol style="list-style-type: none">1. Actor opens favourites list.2. Actor selects game from favourites list.
Precondition	Player is successfully logged into platform. Favourites list is not empty.
Postcondition	Game is selected by player.
Qualities	



Requirements Elicitation

Use Case Overview and Use Cases

Use case **launch game**.

Use Case Name	Launch game
Actor	Player
Flow of Events	<ol style="list-style-type: none">1. Actor launches a selected game.2. Actor plays the game till it ends.3. Game results are added to the game's high score list.
Precondition	Player is successfully logged into platform. A game was previously selected (e.g. From the favourites list).
Postcondition	Game results are added to high score list.
Qualities	

Exercise Agenda

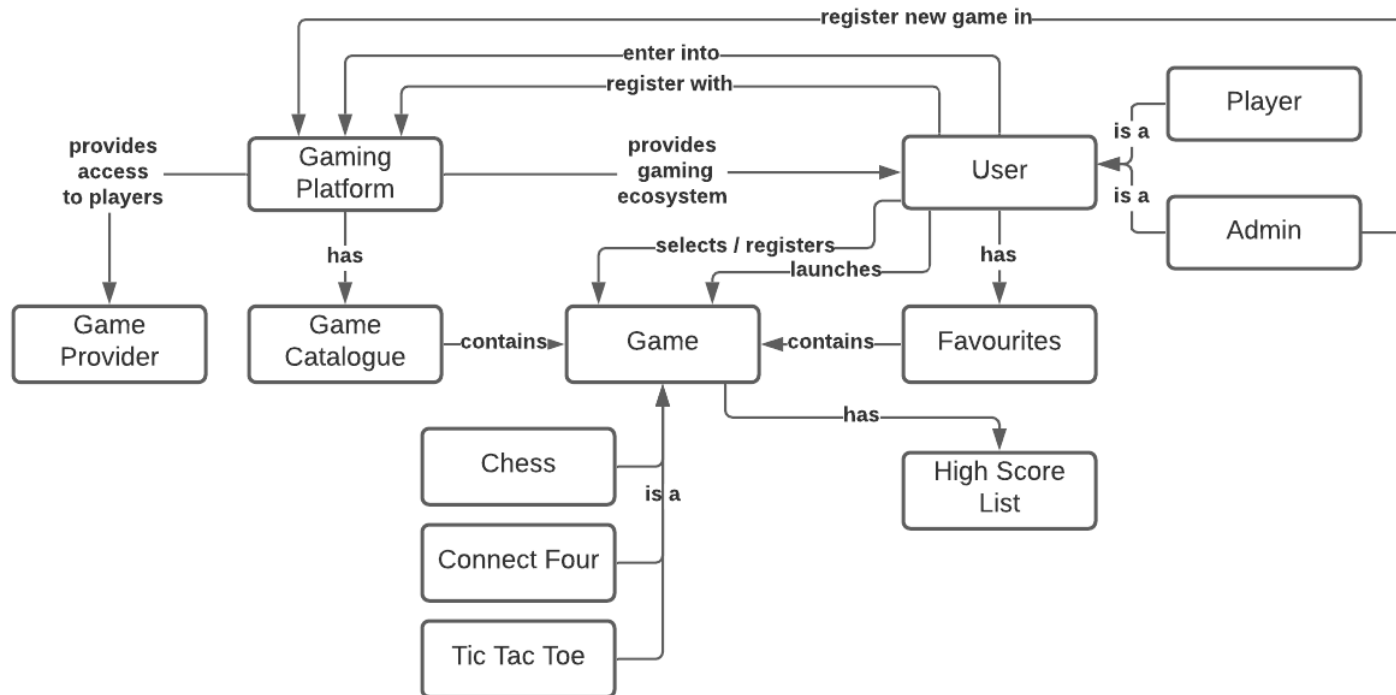


- Requirements Elicitation
- System Analysis
- System Architecture
- System Implementation

System Analysis

Domain Model

Recap: the **domain model for the gaming platform** as it was developed in the previous exercise.





System Analysis

Analysis Model (dynamic)

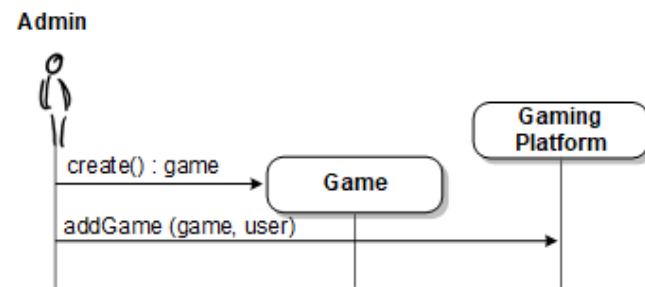
Develop *use case realizations* for all use cases identified to date. Use components that you identified in the domain model or, alternatively, those that you intuitively derive from a respective use case's realization.



System Analysis

Analysis Model (logical view)

Realisation of use case **add game to platform**.

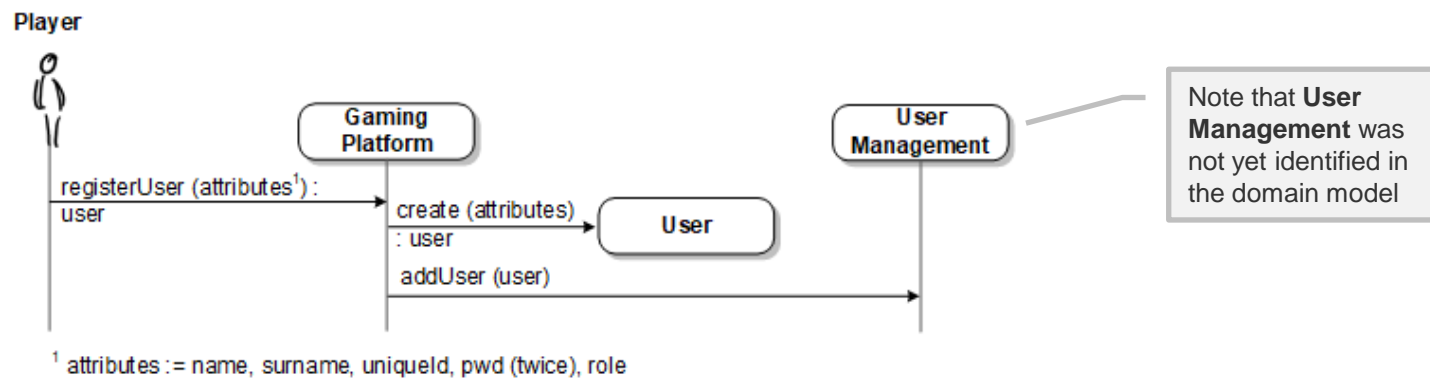




System Analysis

Analysis Model (logical view)

Realisation of use case **register user on platform**.

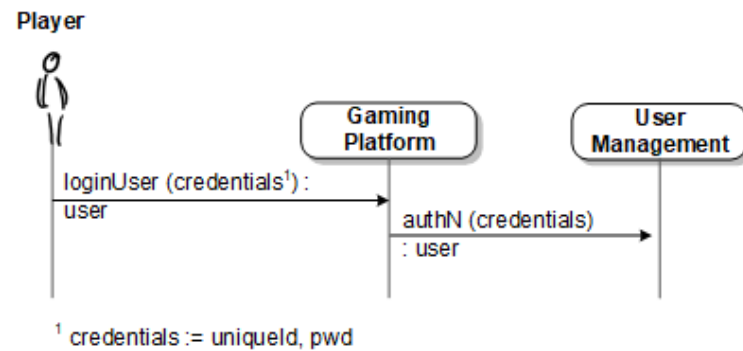


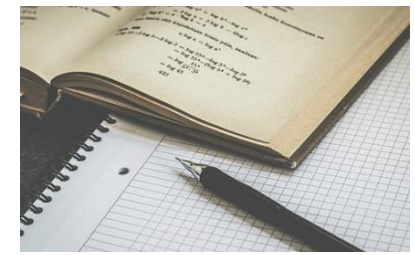


System Analysis

Analysis Model (logical view)

Realisation of use case **log into platform**.

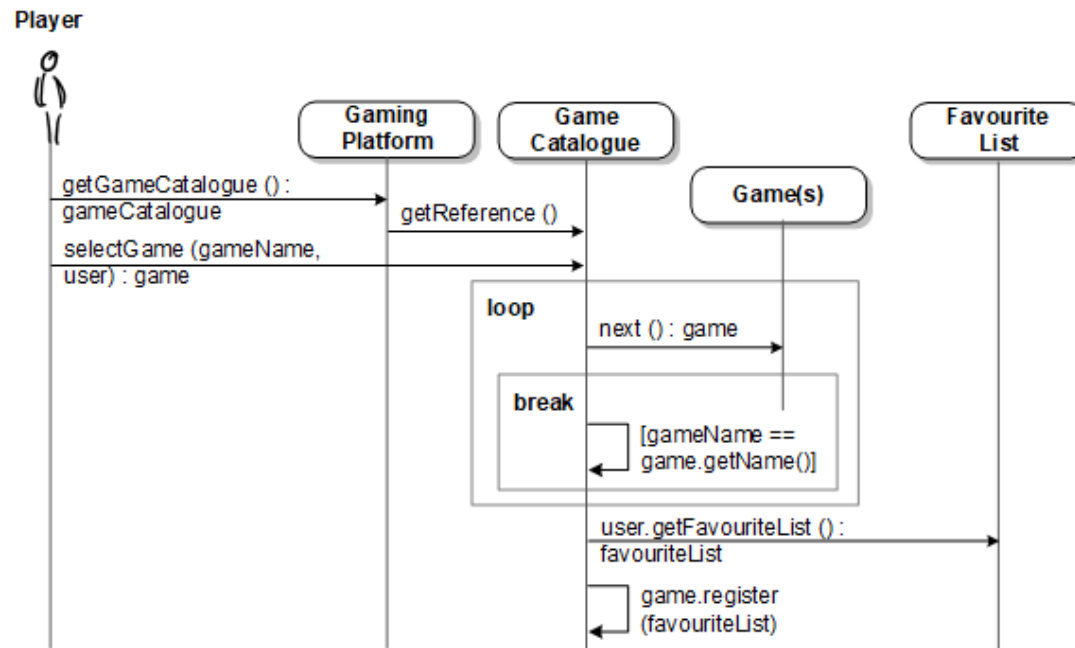




System Analysis

Analysis Model (logical view)

Realisation of use case **select game from catalogue and add it to favourites.**

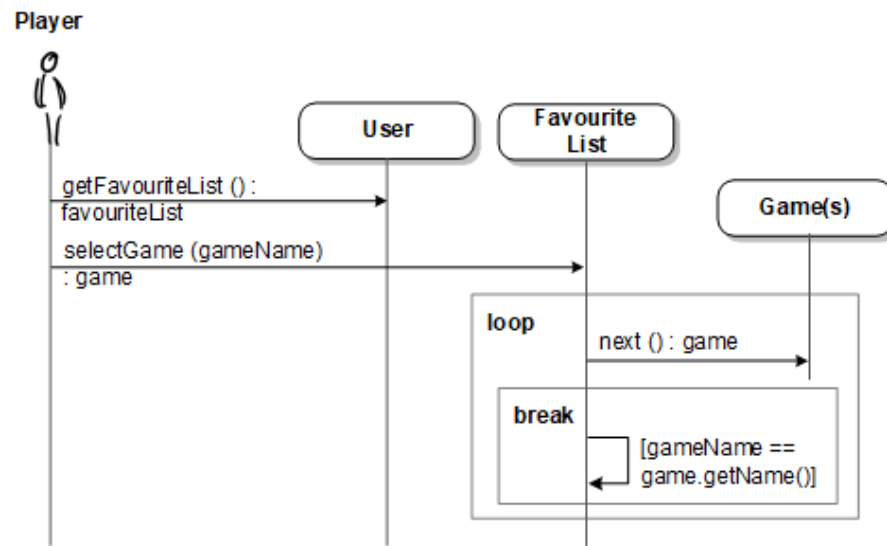




System Analysis

Analysis Model (logical view)

Realisation of use case **select game from favourites**.

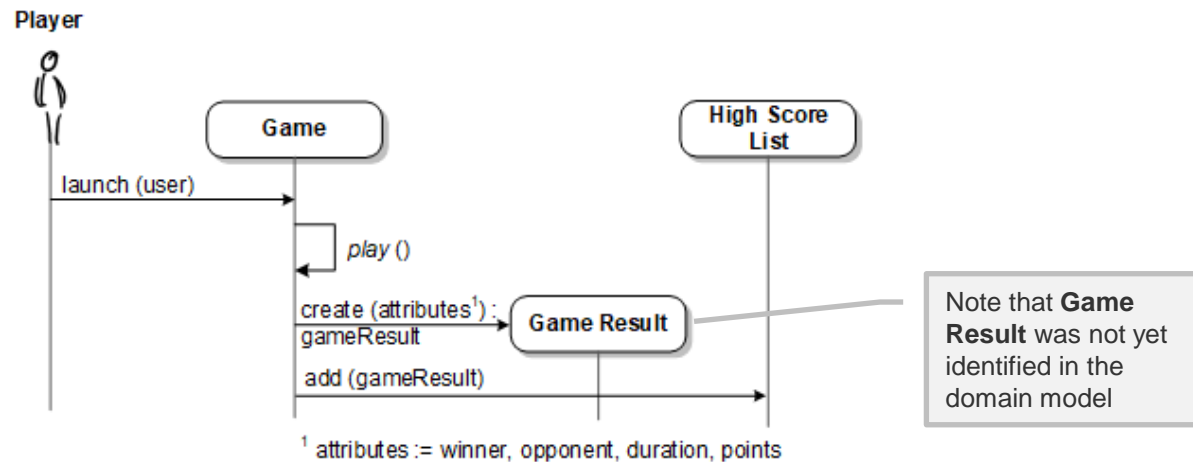




System Analysis

Analysis Model (logical view)

Realisation of use case **launch game**.



Questions

