

Camilla Lambrocco
CSCI 5448
11/9/2015

Use Case needed for selected use case

Use Case ID:	UC-011
Use Case Name:	Select game
Description:	The user can select either any game in the arcade.

Actors:	User		
Pre-condition:	User launched the application. User is logged in.		
Post-condition:	The user is start playing the selected game.		
Frequency of use:	Each instance of the application.		
Flow of events:			
	#	Actor action	System response
	1	Select game	Game name is stored and game interface is launched
Variations:	None.		
Notes and issues:			
Developer notes:			

Use Case ID:	UC-011
Use Case Name:	Select player mode
Description:	The user can select either single player mode or two player mode.

Actors:	User														
Pre-condition:	The user launched the application.														
Post-condition:	The user is either in single player mode or in two player mode.														
Frequency of use:	Each instance of the application.														
Flow of events:	<table><tr><th>#</th><th>Actor action</th><th>System response</th></tr><tr><td>1</td><td>User login in the application</td><td>Set up of arcade interface</td></tr><tr><td>2</td><td>Select game</td><td>Game name is stored and game interface is launched</td></tr><tr><td>3</td><td>Select mode</td><td>Set up platform response of the selected game depending on selection made</td></tr></table>			#	Actor action	System response	1	User login in the application	Set up of arcade interface	2	Select game	Game name is stored and game interface is launched	3	Select mode	Set up platform response of the selected game depending on selection made
#	Actor action	System response													
1	User login in the application	Set up of arcade interface													
2	Select game	Game name is stored and game interface is launched													
3	Select mode	Set up platform response of the selected game depending on selection made													
Variations:	None.														
Notes and issues:															
Developer notes:															

Use Case ID:	UC-021
Use Case Name:	Single Player Mode
Description:	The user can play any game in the arcade in single player mode (against the AI).

Actors:	User		
Pre-condition:	Application is started. User is logged in.		
Post-condition:	The user can play any game against the AI.		
Frequency of use:	Each instance the user selects single player mode and selects a game to play.		
Flow of events:			
	#	Actor action	System response
	1	Select game	Game name is stored and game interface is launched
	2	Select mode single player	AI is set up for the game selected
Variations:	None.		
Notes and issues:			
Developer notes:			

Use Case ID:	UC-022
Use Case Name:	Two Player Mode
Description:	The users can play any game in the arcade in two player mode (against each other).

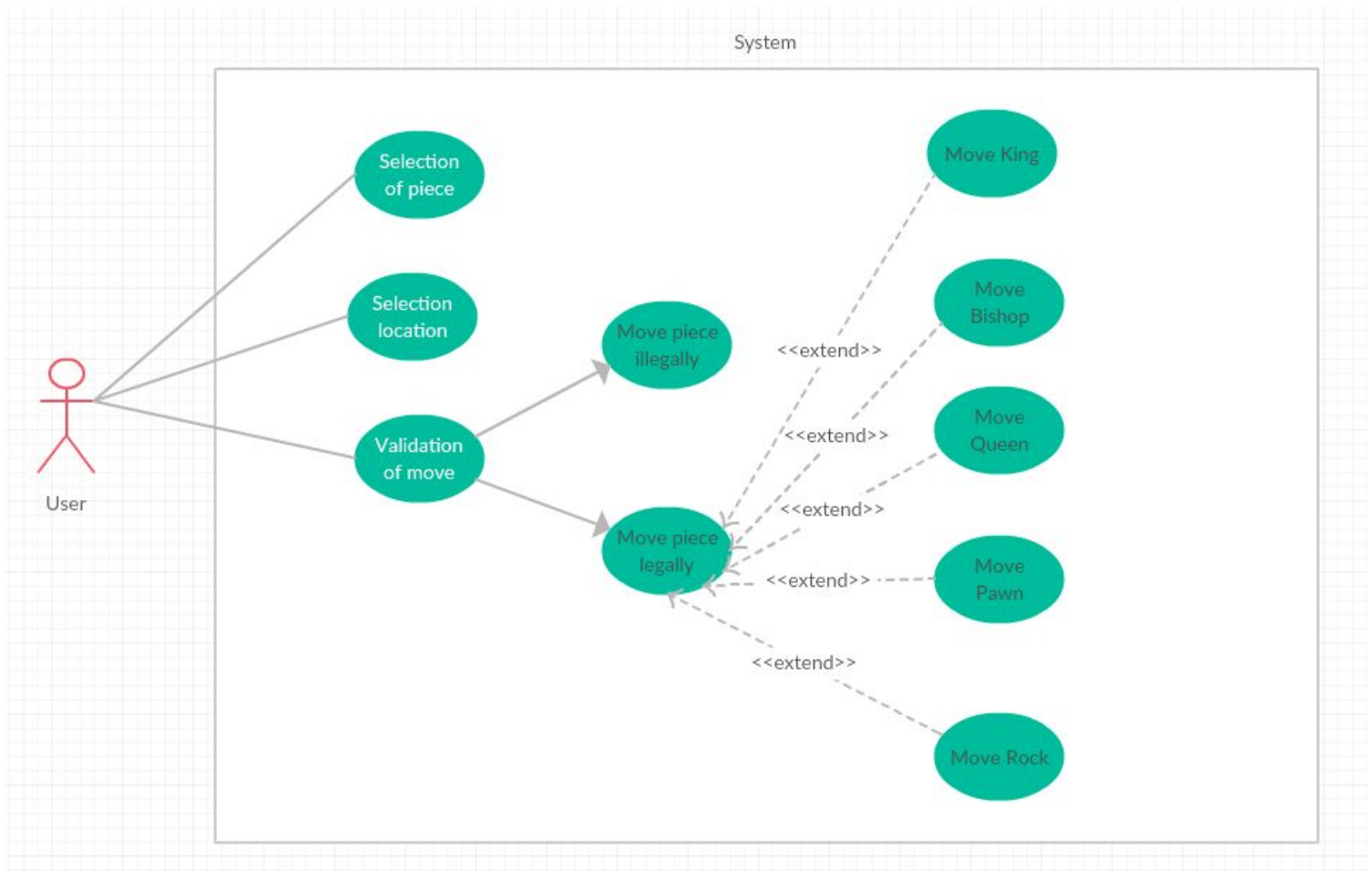
Actors:	User		
Pre-condition:	Application is started. User is logged in.		
Post-condition:	The users can play any game in the arcade in two player mode (against each other).		
Frequency of use:	Each instance the user selects two player mode and selects a game to play.		
Flow of events:			
	#	Actor action	System response
	1	Select game	Game name is stored and game interface is launched
	2	Select mode dual player	Interface is setup and system awaits a player to execute a move
Variations:	None.		
Notes and issues:			
Developer notes:			

Use case to be analyzed: Execute Move in Chess Game

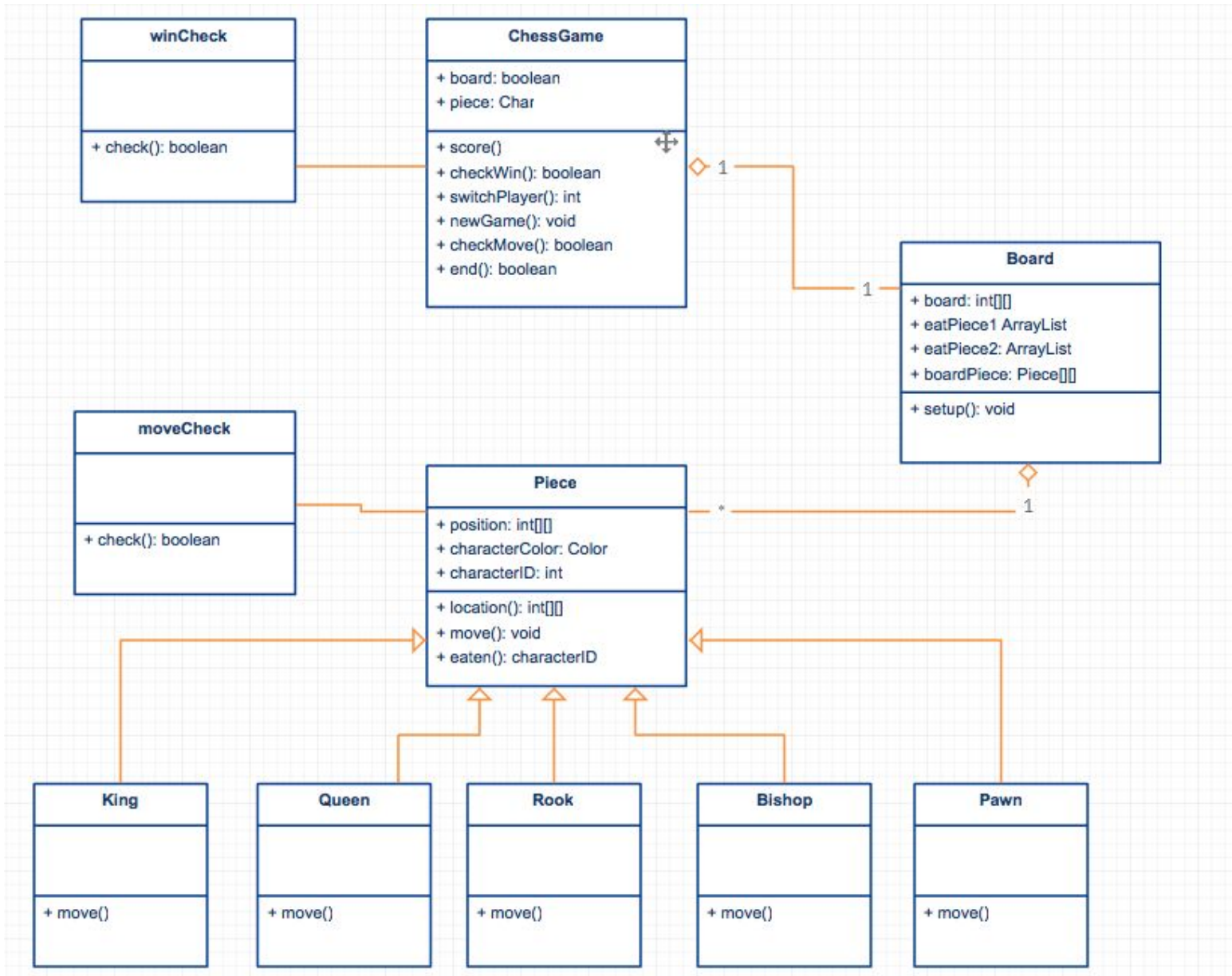
Use Case ID:	UC-041
Use Case Name:	Execute Move Chess
Description:	The user executes a move in the game of chess.

Actors:	User														
Pre-condition:	A game of chess has been launched in either single or two player mode. Player has been switched														
Post-condition:	A user's desired move is executed.														
Frequency of use:	When player is switched and the game has not yet been won or tied.														
Flow of events:	<table><tr><th>#</th><th>Actor action</th><th>System response</th></tr><tr><td>1</td><td>Current player select piece to move</td><td>Piece is selected and returned to user</td></tr><tr><td>2</td><td>Destination of selected piece is chosen</td><td>The location of the piece is identified and stored</td></tr><tr><td>3</td><td>Confirmation of the move of the piece</td><td>Piece is moved to stored location</td></tr></table>			#	Actor action	System response	1	Current player select piece to move	Piece is selected and returned to user	2	Destination of selected piece is chosen	The location of the piece is identified and stored	3	Confirmation of the move of the piece	Piece is moved to stored location
#	Actor action	System response													
1	Current player select piece to move	Piece is selected and returned to user													
2	Destination of selected piece is chosen	The location of the piece is identified and stored													
3	Confirmation of the move of the piece	Piece is moved to stored location													
Variations:	If an invalid move is selected, the move is not executed.														
Notes and issues:															
Developer notes:															

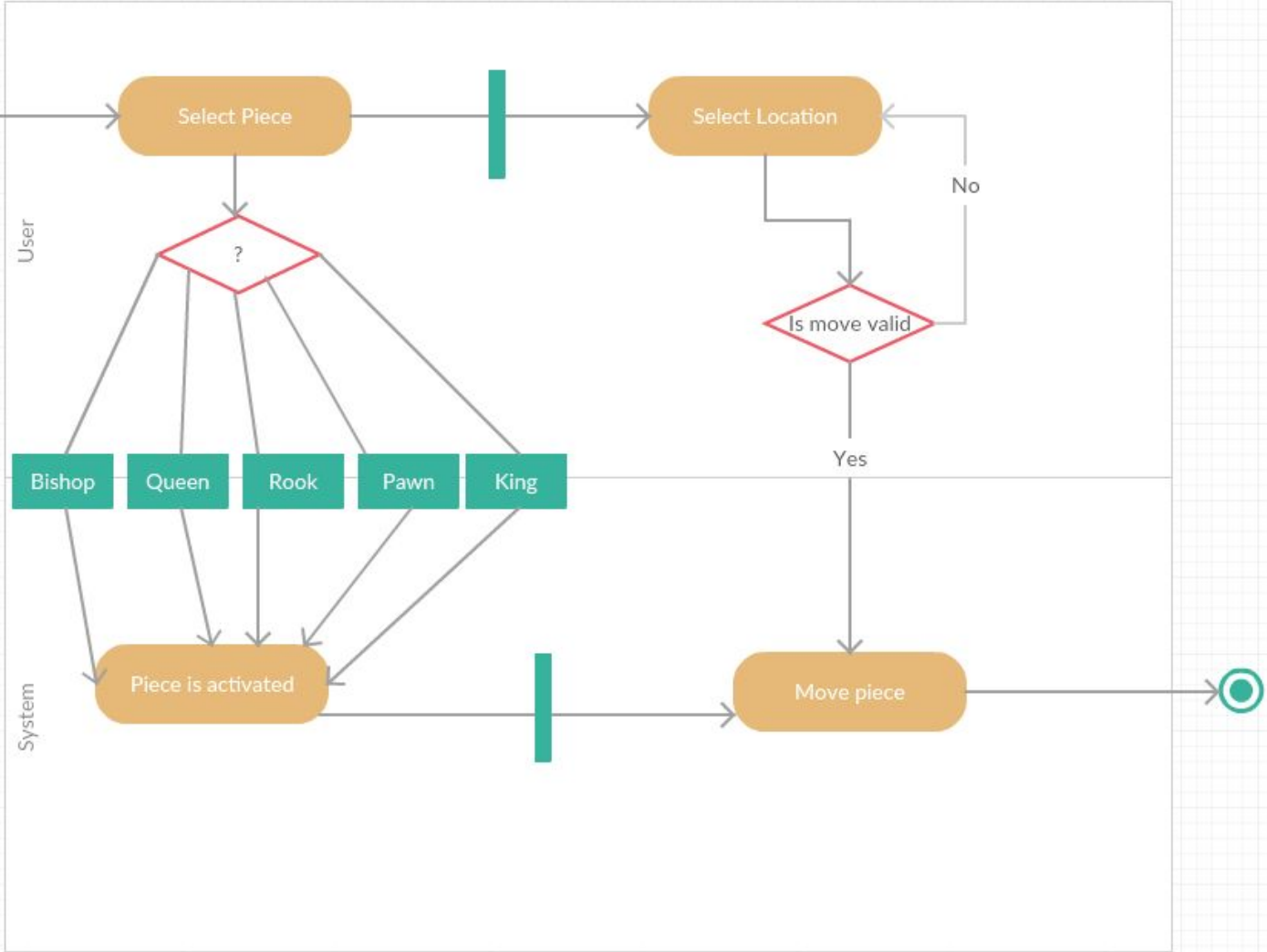
Use Case Diagram for UC-041



Class Diagram for UC-041



Activity Diagram UC-041



Sequence Diagram UC-041

