Fully Dressed UC1:

Player Ends Game

High Risk

# Table of Contents

[**Table of Contents**](#_snfbekw2pr0s) **1**

[**Use Case**](#_evhhxcpx2amh) **2**

[**Scope**](#_6y1d169bged6) **2**

[**Level**](#_3n4jagl0e698) **2**

[**Primary Actor**](#_upfc3o6ly5a6) **2**

[**Stakeholders and Interests**](#_hvz7dbz2iivd) **2**

[**Preconditions**](#_u9nbw7fey1e) **2**

[**Success Guarantee**](#_gg1yvkx0vdvi) **2**

[**Main Success Scenario**](#_ucov0batc4zy) **2**

[**Extensions**](#_md73gkwn9x44) **3**

[**Special Requirements**](#_36dj0qx99gua) **3**

[**Technology and Data Variations List**](#_ikdqkdod0ub6) **3**

[**Frequency of Occurrence**](#_w9cw1nlggd34) **3**

[**Miscellaneous**](#_ud06lnaabx8) **3**

[**System Sequence Diagram**](#_7tth1yi9t8rn) **4**

[**Alternate Scenario**](#_yxalm4oiphal) **4**

# 

\*highlights indicate changes

# 

# 

# 

# 

# 

# 

# 

# 

# Use Case

Player Ends Game

# 

# Scope

CityBuilder Game Application

# 

# Level

user-goal

# 

# Primary Actor

Player

# 

# Stakeholders and Interests

-Player: Wants to win the game by building five structures the fastest.

-Administrator: Wants to make sure the win is delivered to the right person. Wants to make sure

the win is legitimate.

-Developer: Wants to make sure the system works perfectly. Wants to make sure players can experience a game with no shortcomings.

-Evaluator: Wants to make sure our game design is good.

# 

# Preconditions

System is running, Players and Administrators are logged in, One player has enough buildings and resources

# 

# Success Guarantee

Player must build five structures before any other player.

# 

# Main Success Scenario

1. Administrator starts the system
2. Players are logged in and authenticated
3. System administrator starts the the game.
4. Players gather resources.
5. Players builds structures.
6. Play passes to next player
7. System gives control to the next player

*Steps 4 - 7 is repeated until a player successfully builds 5 structures*

1. System recognizes player has win condition
2. System displays victory screen with player name
3. System updates Player’s account with victory.
4. System closes the game.

# 

# Extensions

Player is not able to win yet because another player has satisfied win condition (build 5 structures)

* If two or more players finish five structures in the same turn, the player with more victory points will win the game.
  + If two or more players have the same amount of victory points in that turn, the player with more leftover resources will win the game.
    - If there is still a tie, the player who wins the coin flip wins the game.

System Administrator may extend the game if win is deemed unfair

* Administrator changes win condition
* System resumes game

# 

# Special Requirements

* System works in both Linux and C++
* Shows all options in English
* System outputs victory screen in under 5 seconds

# 

# Technology and Data Variations List

* System Administrator Override by entering user name and password

# Frequency of Occurrence

Only once per game.

# 

# Miscellaneous

What is the amount of buildings needed for a win?

What happens if two players tie in every resource?

Can there be more than one winner?

# 

# 

# 

# 

# 

# 

# 

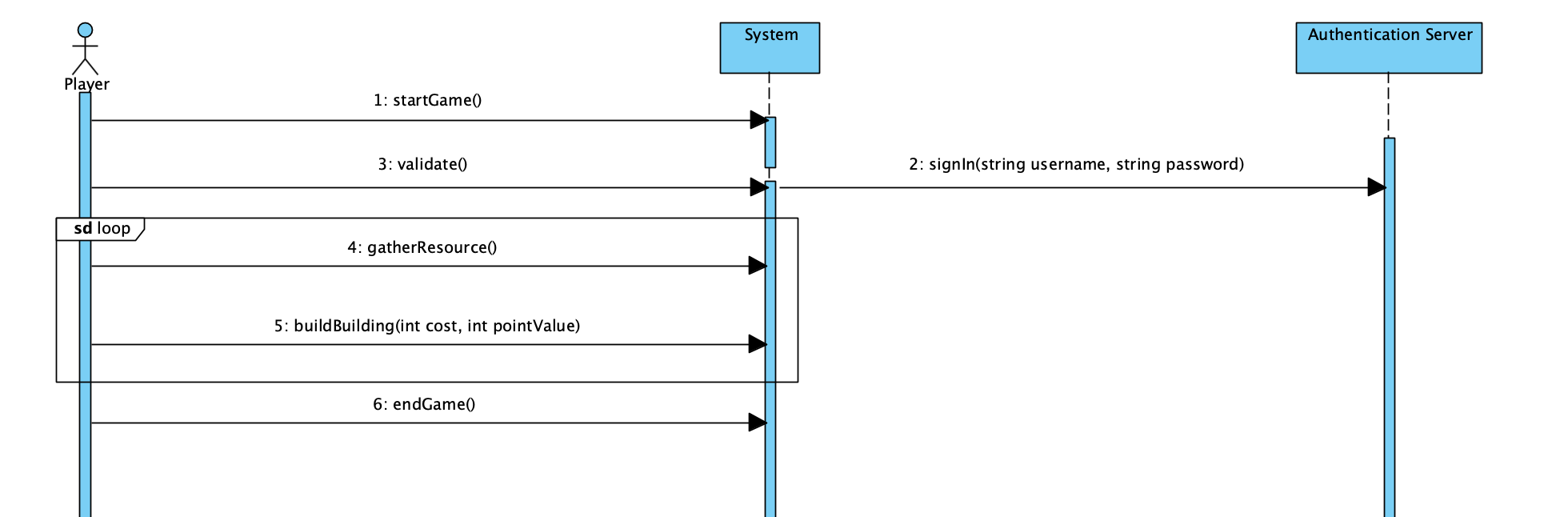
# 

# 

# 

# System Sequence Diagram

In this use case, the player will login into the system. The system will successfully authenticate the player, and the player will start the game. The player(s) will take turns gathering resources and building buildings through gatherResource() and buildBuilding(cost, pointvalue). After one player has finished successfully building five buildings, the player will try to end the game with endGame(). The system will then determine that the player is able to win and return PlayerWins() to the player, which will display the victory to that player. In this iteration, the login is uncompleted and is very basic at this point.



# Alternate Scenario

In this scenario, the player will login into the system. The system will successfully authenticate the player, and the player will start the game. The player(s) will take turns gathering resources and building buildings through gatherResource() and buildBuilding(cost, pointvalue). After one player has finished successfully building five buildings, the player will try to end the game with endGame(). The system will then determine that the player is unable to win the game because there is another player with the same victory condition but with more victory points. The system then returns to the player that he or she has lost with PlayerLoses(). In this iteration, the login is uncompleted, and very basic.

