|  |
| --- |
| Outbreak Smartphone App for iPhone  Use Case: Reward Achievement |

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 23-Jan-12 | 1.0 | First draft | Kevin Fauver |
| 28-Jan-12 | 1.1 | QA Formatting | Sean Marek |
| 2-Feb-12 | 1.2 | Group revision | GROUP |
|  |  |  |  |

Contents

1. Populate Store 4

1.1 Brief Description 4

1.2 Requirements Trace 4

1.3 Involved Actors 4

1.4 Preconditions 4

1.5 Post conditions 4

1.6 Invariants 4

2. Flow of Events 5

2.1 Basic Flow 5

3. Extension Points – None 5

4. Scenarios 5

4.1 Happy Day 5

4.2 Rainy Day 1 – Inventory doesn’t add item 6

4.3 Rainy Day 2 – Store allows purchase despite lock feature 6

4.4 Rainy Day 3 – Store allows purchase with less than required amount to purchase item 7

Use Case Reward Achievement

# Populate Store

## Brief Description

This use case rewards the iOS user with designed achievements that unlock special items in the store or looks for virus creation. The system needs to check that the achievement hasn’t already been awarded. Then it needs to check the corresponding reward and unlock it. Once that is done the user will be able to select it from the corresponding selection location.

## Requirements Trace

14, 15, 16

## Involved Actors

iOS User

## Preconditions

## Post conditions

* The iOS user has more awarded items/looks to choose from.

## Invariants

* iOS user cannot have negative currency

# Flow of Events

## Basic Flow

This use case starts when the iOS user attempts to purchase an item from the store.

* + 1. iOS user attempts to make a purchase from the store.
    2. System checks if the item is locked.
    3. System returns false if the item is not locked.
    4. System checks player’s currency to match with cost of item.
    5. System returns true if the amount of player’s currency matches the cost of item.
    6. System adds item (or multiplier) to iOS user’s inventory.
    7. iOS user notices item (or multiplier) added to inventory.

# Extension Points – None

# Scenarios

## Happy Day

Assumptions: iOS User – John

iOS User has positive correct currency

Item to be chosen is not locked

Steps:

* + 1. John attempts to make a purchase from the store.
    2. System checks if the item is locked.
    3. System returns false if the item is not locked.
    4. System checks player’s currency to match with cost of item.
    5. System returns true if the amount of player’s currency matches the cost of item.
    6. System adds item (or multiplier) to John’s inventory.
    7. John notices item (or multiplier) added to inventory.

## Rainy Day 1 – Inventory doesn’t add item

Assumptions: iOS User – John

iOS User has positive currency

Item to be chosen is not locked

Steps:

* + 1. John attempts to make a purchase from the store.
    2. System checks if the item is locked.
    3. System returns true if the item is locked.
    4. System checks player’s currency to match with cost of item.
    5. System returns true if the amount of player’s currency matches the cost of item.
    6. System fails to add item (or multiplier) to John’s inventory.
    7. John notices that the item (or multiplier) did not add to inventory.

## Rainy Day 2 – Store allows purchase despite lock feature

Assumptions: iOS User – John

iOS User has positive currency

Item to be chosen is locked

Steps:

* + 1. John attempts to purchase a locked item from the store.
    2. System checks if the item is locked.
    3. System returns true if the item is locked.
    4. System checks player’s currency to match with cost of item.
    5. System returns true if the amount of player’s currency matches the cost of item.
    6. System adds locked item (or multiplier) to John’s inventory.
    7. John notices that the locked item (or multiplier) added to inventory.

## Rainy Day 3 – Store allows purchase with less than required amount to purchase item

Assumptions: iOS User – John

iOS User has positive currency, but less than required amount needed

Steps:

* + 1. John attempts to purchase a locked item from the store.
    2. System checks if the item is locked.
    3. System returns true if the item is locked.
    4. System checks player’s currency to match with cost of item.
    5. System returns true if the amount of player’s currency does not match the cost of item.
    6. System adds item (or multiplier) to John’s inventory.
    7. John notices that the item (or multiplier) added to inventory.