|  |
| --- |
| Outbreak Smartphone App for iPhone  Use Case: Populate Store |

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 21-Jan-12 | 1.0 | First draft | Kevin Fauver |
| 28-Jan-12 | 1.1 | QA Formatting | Sean Marek |
|  |  |  |  |

Contents

[1. Populate Store 4](#_Toc315529469)

[1.1 Brief Description 4](#_Toc315529470)

[1.2 Requirements Trace\*\* 4](#_Toc315529471)

[1.3 Involved Actors 4](#_Toc315529472)

[1.4 Preconditions 4](#_Toc315529473)

[1.5 Post conditions 4](#_Toc315529474)

[1.6 Invariants 4](#_Toc315529475)

[2. Flow of Events 5](#_Toc315529476)

[2.1 Basic Flow 5](#_Toc315529477)

[3. Extension Points - None 5](#_Toc315529478)

[4. Scenarios 5](#_Toc315529479)

[4.1 Happy Day 5](#_Toc315529480)

[4.2 Rainy Day 1 – Items do not reflect locked parameter 6](#_Toc315529481)

[4.3 Rainy Day 2 – Items do not load to screen 6](#_Toc315529482)

Use Case: Populate Store

# Populate Store

## Brief Description

This use case has the system populate the store when opened by the iOS user. The system needs to check the database for unlocked and locked items. Once the system reads in the information it will show the iOS user what is available for purchase for the offline game.

## Requirements Trace

7

## Involved Actors

iOS User

## Preconditions

The system interface is at the store screen

## Post conditions

Unlocked items will show a non-greyed graphic

Locked items will show ITEM\_LOCK by a greyed out graphic

## Invariants

Store must have items

# Flow of Events

## Basic Flow

This use case starts when the iOS user wants to make a purchase from the store.

* + 1. iOS user enters the store from the main menu.
    2. System loads item graphics to the screen.
    3. System checks database for ITEM\_LOCK parameters.
    4. System reflects ITEM\_LOCK parameters on item graphics.
    5. System locks ITEM\_LOCK graphics from use.
    6. iOS user notices store populated with items.

# Extension Points - None

# Scenarios

## Happy Day

Assumptions: iOS User – John

There are both locked and unlocked items

Steps:

* + 1. John enters “Store” from the main menu.
    2. System loads item graphics to the screen.
    3. System checks database for ITEM\_LOCK parameters.
    4. System reflects ITEM\_LOCK parameter on associated item graphics.
    5. System locks ITEM\_LOCK item from purchase.
    6. John notices store populated with correctly reflected items.

## Rainy Day 1 – Items do not reflect locked parameter

Assumptions: iOS User – John

There are both locked and unlocked items

Steps:

* + 1. John enters “Store” from the main menu.
    2. System loads item graphics to the screen.
    3. System checks database for ITEM\_LOCK parameters.
    4. System fails to reflect ITEM\_LOCK parameter on associated item graphics.
    5. System locks ITEM\_LOCK item from purchase.
    6. John notices store populated with no locked items.

## Rainy Day 2 – Items do not load to screen

Assumptions: iOS User – John

There are both locked and unlocked items

Steps:

* + 1. John enters “Store” from the main menu.
    2. System fails to load item graphics to the screen.
    3. System checks database for ITEM\_LOCK parameters.
    4. System attempts to reflect ITEM\_LOCK parameter on associated item graphics.
    5. System attempts to locks ITEM\_LOCK item from purchase.
    6. John notices no items populated in the store.