|  |
| --- |
| Outbreak Smartphone App for iPhone  Use Case: Proximity Detection |

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 1-Jan-12 | 1.0 | Original draft | Sean Marek |
| 28-Jan-12 | 1.1 | QA Formatting | Sean Marek |
|  |  |  |  |

Contents

1. Proximity Detection 4

1.1 Brief Description 4

1.2 Requirements trace 4

1.3 Involved actors 4

1.4 Preconditions 4

1.5 Postconditions 4

1.6 Invariants 4

2. FLOW EVENTS 4

2.1 Basic flow 4

3. Extentions Points – None 5

4. Scenarios 5

4.1 Happy day 5

4.2 Rainy Day 5

Use Case: Proximity Detection

# Proximity Detection

## Brief Description

This use case allows the user to gather the people around them from stored GPS coordinates of all online players and populate a table showing how far away each player is

## Requirements trace

12, 13

## Involved actors

iPhone User

## Preconditions

The system interface is at the instant spread screen

## Postconditions

Phone Displays a table of nearby users

## Invariants

The distance at which players are selected (yet to be determined)

# FLOW EVENTS

## Basic flow

This use case starts when the user wants to get the distances of the other users around them

* + 1. User presses interface button on the iPhone to get local players
    2. The iPhone connects to the web server
    3. The iPhone sends the current location of user to the web server
    4. The web server stores the location on the database
    5. The system runs the GPS distance calculation of each online player in the database
    6. The database returns a list of each online player within a certain distance to the web server
    7. The web server packs up the information and sends it to the iPhone
    8. The iPhone populates a table of nearby users from received list
    9. The user sees the table with usernames and distances from current location

# Extentions Points – None

# Scenarios

## Happy day

Assumptions: User – Sean

Steps:

* + 1. Sean Presses the interface button on the iPhone to get nearby players
    2. The iPhone connects to the web server
    3. The iPhone sends the current location of Sean to the web server
    4. The web server stores the location on the database
    5. The system runs the GPS distance calculation of each online player in the database
    6. The database returns a list of each online player within a certain distance from Sean to the web server
    7. The web server packs up the information and sends it to the iPhone
    8. The iPhone populates a table of nearby players from received list
    9. Sean sees the table with usernames and distances from his current location

## Rainy Day

Assumptions: User – Sean

Steps:

* + 1. Sean Presses the interface button to get nearby players
    2. The iPhone connects to the web server
    3. The connection cannot be established
    4. The app displays signal strength recommends relocation