

Appendix 1

Test cases against the system's functional requirements

Empty template:

Test Case ID:	Test Purpose:		
Environment:			
Preconditions:			
Test Case Steps:			
Step Number:	Procedure:	Expected Response:	Pass/Fail:
Comments:			
Checker:			

1. Requirement: Deliver content to the user through the mobile application when their physical location is within a certain distance of the iBeacon.
 - Description: The mobile application must deliver content to the user when certain conditions. These conditions would include entering the radius of an iBeacon.
 - Acceptance criteria:
 - 1a) When the user enters a two metre radius of an iBeacon the user should be notified via the application.

Test Case ID: 1a	Test Purpose: Deliver content to the user through the mobile application when their physical location is within a certain distance of the iBeacon. <div>● When the user enters a two metre radius of an iBeacon the user should be notified via the application.</div>		
Environment: iPhone 5S: iOS 8.3, 1136 x 640 Screen Resolution, 16GB Built-in Memory, Bluetooth 4.0, Apple A7 Chipset, Dual-core 1.3GHz Cyclone (ARM v8-based) CPU, PowerVR G6430 (quad-core graphics), 1GB DDR3 RAM.			
Preconditions: The mobile application has been installed on the user's smartphone. Bluetooth LE 4.0 is enabled.			
Test Case Steps:			
Step Number:	Procedure:	Expected Response:	Pass/Fail:
1	User opens up application.	The mobile application’s home page loads up.	Pass
2	User opens up the story page.	Mobile application starts searching for iBeacons and lists available iBeacons on the screen telling the user its Major, Minor, RSSI, Distance and colour.	Pass
3	User enters the two radius of iBeacon 1.	The mobile application notifies the user via the application by outputting “Outer zone 1” at the top of the screen.	Pass
Comments:			
Checker: Julian Sykes			

2. Requirement: Deliver content to the user through the system management interface when users' physical locations are within a certain distance of the iBeacon.
- Description: The system management interface must also be used to deliver content to users. The content would be delivered when a user enters the radius of an iBeacon, enters the radius the iBeacons in a specific order or when a specific number of users are inside the radius.
 - Acceptance criteria:
 - 2a) When the user enters a two metre radius of an iBeacon content should be displayed on a monitor through the system management interface.
 - 2b) Content for beacon two is only displayed after the user enters the beacon one radius and content for beacon three is only displayed after the user enters the beacon one and two radius.
 - 2c) The content only plays if one user is inside the radius. If more than one user is inside the radius it should reset as if there were none.

Test Case ID: 2a	Test Purpose: Deliver content to the user through the system management interface when users physical locations are within a certain distance of the iBeacon: <ul style="list-style-type: none">● When the user enters a two metre radius of an iBeacon content should be displayed on a monitor through the system management interface.		
Environment: iPhone 5S: iOS 8.3, 1136 x 640 Screen Resolution, 16GB Built-in Memory, Bluetooth 4.0, Apple A7 Chipset, Dual-core 1.3GHz Cyclone (ARM v8-based) CPU, PowerVR G6430 (quad-core graphics), 1GB DDR3 RAM. Macbook Air 13-inch, Mid 2013: OSX Yosemite version 10.10.3, 1.7 GHz Intel Core i7 Processor, 8GB 1600 MHz DDR3 RAM, Intel HD Graphics 5000 1536 MB Graphics. Google Chrome Browser: Version 42.0.2311.90 (64-bit).			
Preconditions: The mobile application has been installed on the user's smartphone. Bluetooth LE 4.0 is enabled and a network connection is available on the mobile device. The system management system has access to the internet and has been set up so the mobile application user can see the monitor page.			
Test Case Steps:			
Step Number:	Procedure:	Expected Response:	Pass/Fail:
1	User opens up application.	The mobile application's home page loads up.	Pass
2	User opens up the	Mobile application starts	Pass

	story page.	searching for iBeacons and lists available iBeacons on the screen telling the user its Major, Minor, RSSI, Distance and colour.	
3	User enters the two meter radius of iBeacon 2.	Data is sent to the database stored on a remote server containing the BeaconID, UUID hash value, Timestamp and enter Status.	Pass
4	System management interface detects the user entered the radius of iBeacon 2 as it is querying the database every second.	The system management interface displays the video for iBeacon two showing a highlight from a football match.	Pass
Comments:			
Checker: Julian Sykes			

Test Case ID: 2b	Test Purpose: Deliver content to the user through the system management interface when users physical locations are within a certain distance of the iBeacon: <ul style="list-style-type: none">● Content for beacon two is only displayed after the user enters the beacon one radius and content for beacon three is only displayed after the user enters the beacon one and two radius.		
Environment: iPhone 5S: iOS 8.3, 1136 x 640 Screen Resolution, 16GB Built-in Memory, Bluetooth 4.0, Apple A7 Chipset, Dual-core 1.3GHz Cyclone (ARM v8-based) CPU, PowerVR G6430 (quad-core graphics), 1GB DDR3 RAM. Macbook Air 13-inch, Mid 2013: OSX Yosemite version 10.10.3, 1.7 GHz Intel Core i7 Processor, 8GB 1600 MHz DDR3 RAM, Intel HD Graphics 5000 1536 MB Graphics. Google Chrome Browser: Version 42.0.2311.90 (64-bit).			
Preconditions: The mobile application has been installed on the user's smartphone. Bluetooth LE 4.0 is enabled and a network connection is available on the mobile device. The system management system has access to the internet and has been set up so the mobile application user can see the behaviour page.			
Test Case Steps:			
Step Number:	Procedure:	Expected Response:	Pass/Fail:
1	User opens up application.	The mobile application's home page loads up.	Pass
2	User opens up the story page.	Mobile application starts searching for iBeacons and lists available iBeacons on the screen telling the user its Major, Minor, RSSI, Distance and colour.	Pass
3	User enters the two meter radius of iBeacon 2.	Data is sent to the database stored on a remote server containing the BeaconID, UUID hash value, Timestamp and enter Status.	Pass
4	System management interface detects the user entered the radius as it is querying the database every second and checks to see if they have visited iBeacon 1	The system management interface tells the user they need to visit beacon 1 first.	Pass

	in the last five minutes.		
5	User exits the two meter radius of iBeacon 2.	Data is sent to the database stored on a remote server containing the BeaconID, UUID hash value, Timestamp and exit Status.	Pass
6	User enters the two meter radius of iBeacon 1.	Data is sent to the database stored on a remote server containing the BeaconID, UUID hash value, Timestamp and enter Status.	Pass
7	System management interface detects the user entered the radius of iBeacon 1 as it is querying the database every second.	The system management interface displays the lego video for iBeacon 1.	Pass
8	User exits the two meter radius of iBeacon 1.	Data is sent to the database stored on a remote server containing the BeaconID, UUID hash value, Timestamp and exit Status.	Pass
9	User enters the two meter radius of iBeacon 2.	Data is sent to the database stored on a remote server containing the BeaconID, UUID hash value, Timestamp and enter Status.	Pass
10	System management interface detects the user entered the iBeacon 2 radius as it is querying the database every second and checks to see if they have visited iBeacon 1 in the last five	The system management interface displays a rabbit video for iBeacon 2 as they have previously visited iBeacon 1.	Pass

	minutes.		
11	User exits the two meter radius of iBeacon 2.	Data is sent to the database stored on a remote server containing the BeaconID, UUID hash value, Timestamp and exit Status.	Pass
12	User enters the two meter radius of iBeacon 3.	Data is sent to the database stored on a remote server containing the BeaconID, UUID hash value, Timestamp and enter Status.	Pass
13	System management interface detects the user entered the radius as it is querying the database every second and checks to see if they have visited iBeacon 1 and 2 in the last five minutes.	The system management interface displays the pool video for iBeacon 3 as they have previously visited iBeacon 1 and 2.	Pass
Comments:			
Checker: Julian Sykes			

Test Case ID: 2c	Test Purpose: Deliver content to the user through the system management interface when users physical locations are within a certain distance of the iBeacon: <ul style="list-style-type: none">● The content only plays if one user is inside the radius. If more than one user is inside the radius it should reset as if there were none.		
Environment: User A = iPhone 5S: iOS 8.3, 1136 x 640 Screen Resolution, 16GB Built-in Memory, Bluetooth 4.0, Apple A7 Chipset, Dual-core 1.3GHz Cyclone (ARM v8-based) CPU, PowerVR G6430 (quad-core graphics), 1GB DDR3 RAM. User B = iPhone 5: iOS 8.3, 1136 x 640 Screen Resolution, 16GB Built-in Memory, Bluetooth 4.0, Apple A6 Chipset, Dual-core 1.3GHz Swift (ARM v7-based) CPU, PowerVR SGX 543MP3 (triple-core graphics), 1GB DDR2 RAM. Macbook Air 13-inch, Mid 2013: OSX Yosemite version 10.10.3, 1.7 GHz Intel Core i7 Processor, 8GB 1600 MHz DDR3 RAM, Intel HD Graphics 5000 1536 MB Graphics. Google Chrome Browser: Version 42.0.2311.90 (64-bit).			
Preconditions: The mobile application has been installed on two user's smartphone. Bluetooth LE 4.0 is enabled and a network connection is available on the two mobile devices. The system management system has access to the internet and has been set up so the mobile application user can see the monitor page.			
Test Case Steps:			
Step Number:	Procedure:	Expected Response:	Pass/Fail:
1	User A opens up application.	The mobile application's home page loads up.	Pass
2	User A opens up the story page.	Mobile application starts searching for iBeacons and lists available iBeacons on the screen telling the user its Major, Minor, RSSI, Distance and colour.	Pass
3	User A enters the one meter radius of iBeacon 4.	Data is sent to the database stored on a remote server containing the BeaconID, UUID hash value, Timestamp and enter Status.	Pass
4	System management interface detects the user entered the radius as it is querying the database every second and checks	The system management interface plays the horror frame video as only one user is inside the radius.	Pass

	to see how many users are inside the radius.		
5	User B opens up application.	The mobile application's home page loads up.	Pass
6	User B opens up the story page.	Mobile application starts searching for iBeacons and lists available iBeacons on the screen telling the user its Major, Minor, RSSI, Distance and colour.	Pass
7	User B enters the one meter radius of iBeacon 4.	Data is sent to the database stored on a remote server containing the BeaconID, UUID hash value, Timestamp and enter Status.	Pass
8	System management interface detects the user entered the radius as it is querying the database every second and checks to see how many users are inside the radius.	The system management interface pauses and restarts the horror frame video as two users are inside the radius.	Pass
Comments:			
Checker: Julian Sykes			

3. Requirement: Each user must be uniquely identified anonymously.

○ Description: To avoid privacy issues regarding the device UUID this will need to be changed into an anonymous value which can still identify each user uniquely.

○ Acceptance criteria:

■ 3a) Each user can be uniquely identified without using their device UUID.

Test Case ID: 3a	Test Purpose: Each user must be uniquely identified anonymously. ● Each user can be uniquely identified without using their device UUID.		
Environment: User A = iPhone 5S: iOS 8.3, 1136 x 640 Screen Resolution, 16GB Built-in Memory, Bluetooth 4.0, Apple A7 Chipset, Dual-core 1.3GHz Cyclone (ARM v8-based) CPU, PowerVR G6430 (quad-core graphics), 1GB DDR3 RAM. User B = iPhone 5: iOS 8.3, 1136 x 640 Screen Resolution, 16GB Built-in Memory, Bluetooth 4.0, Apple A6 Chipset, Dual-core 1.3GHz Swift (ARM v7-based) CPU, PowerVR SGX 543MP3 (triple-core graphics), 1GB DDR2 RAM. Macbook Air 13-inch, Mid 2013: OSX Yosemite version 10.10.3, 1.7 GHz Intel Core i7 Processor, 8GB 1600 MHz DDR3 RAM, Intel HD Graphics 5000 1536 MB Graphics. Google Chrome Browser: Version 42.0.2311.90 (64-bit).			
Preconditions: The mobile application has been installed on two user's smartphone. Bluetooth LE 4.0 is enabled and a network connection is available on the two mobile devices. The system management system has access to the internet and has been set up so the Yellobrick employee can see the view all data page.			
Test Case Steps:			
Step Number:	Procedure:	Expected Response:	Pass/Fail:
1	User A opens up application.	The mobile application’s home page loads up.	Pass
2	User A opens up the story page.	Mobile application starts searching for iBeacons and lists available iBeacons on the screen telling the user its Major, Minor, RSSI, Distance and colour.	Pass
3	User A enters the one meter radius of iBeacon 4.	Data is sent to the database stored on a remote server containing the BeaconID, UUID hash value, Timestamp and enter	Pass

		Status.	
4	User B opens up application.	The mobile application's home page loads up.	Pass
5	User B opens up the story page.	Mobile application starts searching for iBeacons and lists available iBeacons on the screen telling the user its Major, Minor, RSSI, Distance and colour.	Pass
6	User B enters the one meter radius of iBeacon 4.	Data is sent to the database stored on a remote server containing the BeaconID, UUID hash value, Timestamp and enter Status.	Pass
7	Yellobrick employee opens up the view all data in System management interface.	The system management interface displays all the users data. Each user will have a different UUID hash value meaning they can be uniquely identified, but are still anonymous as you have no idea who they are.	Pass
Comments:			
Checker: Julian Sykes			

4. Requirement: User data must be sent to a remote server when they enter a specific radius of an iBeacon and have access to the internet.
- Description: User data including Beacon ID, User ID, Location status, date and time needs to be sent to the server if a network is available. If there is no network connection the data will need to be stored locally on the device and sent to the server when a network connection is next available.
 - Acceptance criteria:
 - 4a) Beacon ID, User ID, Location status, date and time is sent to the remote server when the user enters a specific radius of an iBeacon.
 - 4b) The mobile application stores user data in local storage when there is no network connection available. The data is then sent to the remote database.

Test Case ID: 4a	Test Purpose: User data must be sent to a remote server when they enter a specific radius of an iBeacon and have access to the internet. ● Beacon ID, User ID, Location status, date and time is sent to the remote server when the user enters a specific radius of an iBeacon.		
Environment: iPhone 5S: iOS 8.3, 1136 x 640 Screen Resolution, 16GB Built-in Memory, Bluetooth 4.0, Apple A7 Chipset, Dual-core 1.3GHz Cyclone (ARM v8-based) CPU, PowerVR G6430 (quad-core graphics), 1GB DDR3 RAM. Macbook Air 13-inch, Mid 2013: OSX Yosemite version 10.10.3, 1.7 GHz Intel Core i7 Processor, 8GB 1600 MHz DDR3 RAM, Intel HD Graphics 5000 1536 MB Graphics. Google Chrome Browser: Version 42.0.2311.90 (64-bit).			
Preconditions: The mobile application has been installed on the user's smartphone. Bluetooth LE 4.0 is enabled and a network connection is available on the mobile device. The system management system has access to the internet and has been set up so the Yellobrick employee can see the view all data page.			
Test Case Steps:			
Step Number:	Procedure:	Expected Response:	Pass/Fail:
1	User opens up application.	The mobile application’s home page loads up.	Pass
2	User opens up the story page.	Mobile application starts searching for iBeacons and lists available iBeacons on the screen telling the user its Major, Minor, RSSI, Distance and colour.	Pass
3	User enters the	Data is sent to the database	Pass

	two meter radius of iBeacon 2.	stored on a remote server containing the BeaconID, UUID hash value, Timestamp and enter Status.	
4	Yellobrick employee opens up the view all data in System management interface.	The system management interface displays the user entering the radius of iBeacon 2.	Pass
Comments:			
Checker: Julian Sykes			

Test Case ID: 4b	Test Purpose: User data must be sent to a remote server when they enter a specific radius of an iBeacon and have access to the internet. <ul style="list-style-type: none">● The mobile application stores user data in local storage when there is no network connection available. The data is then sent to the remote database.		
Environment: iPhone 5S: iOS 8.3, 1136 x 640 Screen Resolution, 16GB Built-in Memory, Bluetooth 4.0, Apple A7 Chipset, Dual-core 1.3GHz Cyclone (ARM v8-based) CPU, PowerVR G6430 (quad-core graphics), 1GB DDR3 RAM. Macbook Air 13-inch, Mid 2013: OSX Yosemite version 10.10.3, 1.7 GHz Intel Core i7 Processor, 8GB 1600 MHz DDR3 RAM, Intel HD Graphics 5000 1536 MB Graphics. Google Chrome Browser: Version 42.0.2311.90 (64-bit).			
Preconditions: The mobile application has been installed on the user's smartphone. Bluetooth LE 4.0 is enabled and no network connection is available on the mobile device. The system management system has access to the internet and has been set up so the Yellobrick employee can see the view all data page.			
Test Case Steps:			
Step Number:	Procedure:	Expected Response:	Pass/Fail:
1	Yellobrick employee opens up the view all data in System management interface.	All the data will be shown.	Pass
2	User opens up application.	The mobile application’s home page loads up.	Pass
3	User opens up the story page.	Mobile application starts searching for iBeacons and lists available iBeacons on the screen telling the user its Major, Minor, RSSI, Distance and colour. A message will also be output saying “This device is not connected to the internet. Please connect to the internet to enjoy the full experience”.	Pass
4	User enters the two meter radius of iBeacon 2.	Data is stored in local storage containing the BeaconID, UUID hash value, Timestamp and enter Status.	Pass
5	User exits the two	Data is stored in local storage	Pass

	meter radius of iBeacon 2.	containing the BeaconID, UUID hash value, Timestamp and exit Status.	
6	Yellobrick employee opens up the view all data in System management interface.	All the user behavior data will be shown as before. There will be no new data added to the table.	Pass
7	User turns connects to the internet.	The message telling the user “This device is not connected to the internet. Please connect to the internet to enjoy the full experience” will disappear.	Pass
8	User enters the two meter radius of iBeacon 3.	The system checks through local storage and sends that data before the current data to the remote server. Each set of data includes the BeaconID, UUID hash value, Timestamp and enter Status.	Pass
9	Yellobrick employee opens up the view all data in System management interface.	Three new records will be added to the database. The enter and exit data from when the network connection wasn’t available and the current data.	Pass
Comments:			
Checker: Julian Sykes			

5. Requirement: Yellobrick staff must be able to track user “dwell times” of all users.

- Description: The system management interface must show all user’s behaviour history which can be used to track where users have been, how long they spend at each location and the order they visited each location.

- Acceptance criteria:

- 5a) User data including Beacon ID, User ID, Location status, date and time is available to view on the system management interface.

Test Case ID: 5a	Test Purpose: Yellobrick staff must be able to track user “dwell times” of all users. <div>● User data including Beacon ID, User ID, Location status, date and time is available to view on the system management interface.</div>		
Environment: Macbook Air 13-inch, Mid 2013: OSX Yosemite version 10.10.3, 1.7 GHz Intel Core i7 Processor, 8GB 1600 MHz DDR3 RAM, Intel HD Graphics 5000 1536 MB Graphics. Google Chrome Browser: Version 42.0.2311.90 (64-bit).			
Preconditions: Users have previously entered or exited beacon radii and data has been sent to the server. The system management system has access to the internet and has been set up so the mobile application user can see view all data page.			
Test Case Steps:			
Step Number:	Procedure:	Expected Response:	Pass/Fail:
1	Yellobrick employee opens up the view all data in System management interface.	The system management interface displays the complete user history of every user including the BeaconID, UUID, Timestamp, Status, RoomName, LocationName, Longitude and Latitude.	Pass
Comments:			
Checker: Julian Sykes			

6. Requirement: Data must be stored on a remote server in a database so the mobile application and system management interface uses the same data.
- Description: A database will need to be set up on a remote system so the user behaviour data inside the mobile application can be used by system management interface.
 - Acceptance criteria:
 - 6a) A database has been set up and stores important user behaviour data including BeaconID, UUID, Timestamp and Status.

Test Case ID: 6a	Test Purpose: Data must be stored on a remote server in a database. ● A database has been set up and stores important user behaviour data including BeaconID, UUID, Timestamp and Status.		
Environment: Macbook Air 13-inch, Mid 2013: OSX Yosemite version 10.10.3, 1.7 GHz Intel Core i7 Processor, 8GB 1600 MHz DDR3 RAM, Intel HD Graphics 5000 1536 MB Graphics. Google Chrome Browser: Version 42.0.2311.90 (64-bit).			
Preconditions: Users have previously entered or exited beacon radii and data has been sent to the server. The system management system has access to the internet and has been set up so the mobile application user can see view all data page.			
Test Case Steps:			
Step Number:	Procedure:	Expected Response:	Pass/Fail:
1	Yellobrick employee opens up the view all data in System management interface.	The system management interface displays data stored in the remote database.	Pass
Comments:			
Checker: Julian Sykes			