AUT UNIVERSITY

Redefining the Use of Augmented Reality

Product Backlog

Version 1.4

19 October 2015

Jason Gerbes	Joshua Son	Paul Lee	Sean Young
1274664	1388288	1264218	1302108

Contents

0.0 Version History

Version 1.0

Version 1.0 is the initial version of the Product Backlog Document. This version of the document was created as part of Sprint Zero: Development Setup Phase. New versions of the document will be produced during each sprint to address the sign-off dates of each user story.

Version 1.1

Version 1.1 was created at the end of Sprint One. The sign-off dates of user stories completed in this sprint were added to the document.

Section	Overview of Changes	Date
Title Page	Updated version number and date.	07/09/15
3.0	Sign off dates have been added for the following user stories: • US 1: Retrieve location updates • US 2: Display current location • US 14: Retrieve device direction (heading) updates • US 15: Display the device's direction (heading)	07/09/15

Version 1.2

Version 1.1 was created at the end of Sprint Two. The sign-off dates of user stories completed in this sprint were added to the document.

Section	Overview of Changes	Date
Title Page	Updated version number and date.	21/09/15
3.0	Sign off dates have been added for the following user stories: • US 11: Modify a locally stored node • US 12: Remove a locally stored node • US 13: Insert a new node • US 16: Maintain a debug log	21/09/15

Version 1.3

Version 1.1 was created at the end of Sprint Three. The sign-off dates of user stories completed in this sprint were added to the document.

Section	Overview of Changes	Date
Title Page	Updated version number and date.	05/10/15
3.0	Sign off dates have been added for the following user stories: • US 3: Establish a connection • US 4: Load test nodes • US 5: Add test nodes to a local database • US 8: Display a list of locally-stored nodes	05/10/15

VERSION 1.4

Version 1.1 was created at the end of Sprint Four. The sign-off dates of user stories completed in this sprint were added to the document.

Section	Overview of Changes	Date
Title Page	Updated version number and date.	19/10/15
3.0	Sign off dates have been added for the following user stories: • US 6: Determine the distance of a node • US 7: Determine the direction (heading) of a node • US 9: Filter nodes by a given distance parameter • US 10: Display nodes within a given distance parameter	19/10/15

1.0 Introduction

This Product Backlog covers all of the user stories related to development of the Unity plugins and Testing Application. The Testing Application will be used to ensure the functionality of the Unity plugins, and will be referenced by Luminary when creating the Platform Application. The sole user of the testing application is the System Developer, as defined in 2.2.

2.0 Definitions

2.1 Testing Application

The Testing Application will be developed as part of the LuminAR project. The Testing Application will allow the LuminAR group and Luminary to ensure the functionality of the Unity plugins. The Testing Application will act as a proof-of-concept of the functionality of the Unity plugins.

The Unity plugins will be used by the Testing Application to:

- Retrieve the testing device's location updates.
- Display the current location in plaintext.
- Establish a connection with a test remote information server.
- Load a set of test nodes from the remote database.
- Add the loaded nodes to a locally stored SQLite database.
- Determine the distance of a node from the testing device's current location.
- Determine the direction (heading) of a node from the testing device's current location.
- Display a list of locally stored nodes.
- Filter nodes by a given distance parameter.
- Display a list of the nodes within a given distance parameter.
- Make changes to locally stored nodes.
- Remove locally stored nodes.
- Insert a new node into the local database.
- Retrieve the device's true north heading updates.
- Display the device's true north heading in plaintext.
- Maintain a debug log.

2.2 System Developer

The System Developer is the sole user of the Testing Application. The System Developer will use the Testing Application to ensure the functionality of the Unity plugins. The System Developer will develop the Platform Application following delivery of the Testing Application and Unity plugins.

2.3 Definition of Done/Sign-Off Criteria

The Definition of Done (DoD) or Sign-Off Criteria is the exit-criteria used to determine whether a User Story has been completed.

The following points are a checklist that determines the completion status of a User Story. All points must be satisfied to deem a User Story as completed.

1. Relevant acceptance tests created for the User Story.
2. Unit tests produced before code has been written (test-first programming).
3. Code has been produced via peer programming, or cross-checked by at least two developers.
4. Code can be compiled without errors.
5. Code has been commented correctly.
6. Correct naming conventions have been used.
7. All relevant unit tests pass.
8. All relevant functionality tests have been logged and have passed.
9. All acceptance tests have been signed off.
10. Source code documentation has been updated as required.

3.0 User Stories

The User Stories below have been prioritised by the group. The estimated effort is measured in story points, where one story point = one hour of work. The estimated effort was calculated using PlanningPoker (see 4.0).

Title:	Retrieve location updates	US:	1
Modification Date:	24/08/15	Sign-Off Date:	29/08/15
User Story	As the System Developer, I want to retrieve the test device's current GPS location updates so that I can create a location-aware Platform Application.		
Priority	High		
Estimated Effort	3 story points		
Acceptance Tests	1. The Testing Application requests permission to access the test device's location. 2. The Testing Application delegates location updates from the test device's location manager.		

Title:	Display current location	US:	2
Modification Date:	24/08/15	Sign-Off Date:	28/08/15
User Story	As the System Developer, I want the test device's current location to be displayed on the screen so that I can ensure the accuracy of the information.		
Priority	High		
Estimated Effort	2 story points		
Acceptance Tests	1. The test device's current longitude and latitude is displayed in plaintext by the Testing Application. 2. The displayed location information is updated when a location update has been received.		

Title:	Establish a connection	US:	3
Modification Date:	24/08/15	Sign-Off Date:	26/10/15
User Story	As the System Developer, I want to establish a connection with a remote information server so that I can transmit information to the device.		
Priority	Low		
Estimated Effort	5 story points		
Acceptance Tests	1. Required internet access permissions are requested/granted. 2. A connection to the information server can be successfully established.		

Title:	Load test nodes	US:	4
Modification Date:	24/08/15	Sign-Off Date:	26/10/15
User Story	As the System Developer, I want to load a set of test nodes from the remote database so that I can confirm the functionality of the remote database, local database and transmission protocols.		
Priority	Low		
Estimated Effort	3 story points		
Acceptance Tests	1. An SQL database file can be generated by the remote database. 2. The database file can be transmitted to the testing device. 3. The database file is stored on the testing device.		

Title:	Add test nodes in local database	US:	5
Modification Date:	24/08/15	Sign-Off Date:	02/10/15
User Story	As the System Developer, I want the loaded test nodes to be added to the local SQLite database so that I can interact with the information.		
Priority	Low		
Estimated Effort	5 story points		
Acceptance Tests	1. Test nodes contained within the downloaded database are added to the local SQLite database. 2. Duplicate nodes are not added to the SQLite database. 3. The downloaded database file is deleted after the nodes have been loaded into the local SQLite database.		

Title:	Determine the distance of a node	US:	6
Modification Date:	24/08/15	Sign-Off Date:	05/10/15
User Story	As the System Developer, I want to determine the distance of the locally stored nodes from the test device's current location so that the nodes can be filtered by a distance parameter.		
Priority	High		
Estimated Effort	2 story points		
Acceptance Tests	1. The distance between two GPS coordinates can be calculated. 2. The distance between the test device's current location and each node is calculated. 3. The distance of each node from the test device's current location is stored in a HashMap with a 'node : distance' format (with the node being the key).		

Title:	Determine the direction (heading) of a node	US:	7
Modification Date:	24/08/15	Sign-Off Date:	07/10/15
User Story	As the System Developer, I want to determine the direction (heading) of the locally stored nodes from the test device's current location so that nodes can be displayed based on the device's current heading within the Platform Application.		
Priority	High		
Estimated Effort	3 story points		
Acceptance Tests	1. The heading between two GPS coordinates can be calculated. 2. The heading between the test device's current location and each node is calculated. 3. The heading of each node from the test device's current location is stored in a HashMap with a 'node : heading' format (with the node being the key).		

Title:	Display a list of locally-stored nodes	US:	8
Modification Date:	24/08/15	Sign-Off Date:	03/10/15
User Story	As the System Developer, I want the list of locally-stored nodes to be displayed on the screen so that I can visualise the contents of the local SQLite database.		
Priority	Medium		
Estimated Effort	2 story points		
Acceptance Tests	1. A 'show nodes' button is visible in the Testing Application. 2. Tapping the 'show nodes' button displays a list of nodes stored within the SQLite database		

Title:	Filter nodes by a given distance parameter	US:	9
Modification Date:	24/08/15	Sign-Off Date:	10/10/15
User Story	As the System Developer, I want to be able to filter nodes by a given distance parameter so that I can reduce the number of nodes to those within a nearby radius.		
Priority	High		
Estimated Effort	2 story points		
Acceptance Tests	1. Nodes outside of a given distance parameter are removed from the 'node : distance' HashMap. 2. The 'node : distance' HashMap has been ordered by distance value in an ascending order (closest nodes first).		

Title:	Display nodes within a given distance parameter	US:	10
Modification Date:	24/08/15	Sign-Off Date:	12/10/15
User Story	As the System Developer, I want the Testing Application to display a list of nodes, along with their distances, within a given distance parameter so that I can verify the functionality of the Unity plugins.		
Priority	Medium		
Estimated Effort	2 story points		
Acceptance Tests	1. The filtered HashMap of 'node : distance' values is displayed in plaintext on the Testing Application as a list within a text box. 2. The list of nodes can be scrolled.		

Title:	Modify a locally stored node	US:	11
Modification Date:	24/08/15	Sign-Off Date:	15/09/15
User Story	As the System Developer, I want to modify a locally-stored node so that I can verify the functionality of the local SQLite database.		
Priority	Low		
Estimated Effort	2 story points		
Acceptance Tests	1. An 'update' button is visible in the Testing Application. 2. A test node is dedicated as the 'modifiable node'. 3. The updated coordinates of the modifiable node can be typed into a text box. 4. The updated description of the modifiable node can be typed into a text box. 5. Tapping the 'update' button updates the values of the modifiable node in the local SQLite database.		

Title:	Remove a locally stored node	US:	12
Modification Date:	24/08/15	Sign-Off Date:	10/09/15
User Story	As the System Developer, I want to remove a locally-stored node so that I can verify the functionality of the local SQLite database.		
Priority	Low		
Estimated Effort	2 story points		
Acceptance Tests	1. A 'remove' button is visible in the Testing Application. 2. Tapping the 'remove' button removes the modifiable node from the local SQLite database.		

Title:	Insert a new node	US:	13
Modification Date:	24/08/15	Sign-Off Date:	10/09/15
User Story	As the System Developer, I want to insert a new node into the local SQLite database to verify the functionality of the database.		
Priority	Low		
Estimated Effort	2 story points		
Acceptance Tests	1. The coordinates of the new node can be typed into a text box. 2. The description of the new node can be typed into a text box. 3. Tapping an 'insert' button add the new node to the local SQLite database.		

Title:	Retrieve device direction (heading) updates	US:	14
Modification Date:	24/08/15	Sign-Off Date:	04/09/15
User Story	As the System Developer, I want to retrieve device heading updates so that I can create a direction-based Platform Application.		
Priority	High		
Estimated Effort	5 story points		
Acceptance Tests	1. The Testing Application requests permission to access the test device's heading. 2. The Testing Application delegates heading updates from the test device.		

Title:	Display the device's direction (heading)	US:	15
Modification Date:	24/08/15	Sign-Off Date:	02/09/15
User Story	As the System Developer, I want the test device's current location to be displayed on the screen so that I can ensure the accuracy of the information.		
Priority	High		
Estimated Effort	3 story points		
Acceptance Tests	1. The test device's current heading is displayed by the Testing Application. 2. The displayed heading is updated when a heading update has been received.		

Title:	Maintain a debug log	US:	16
Modification Date:	24/08/15	Sign-Off Date:	09/09/15
User Story	As a System Developer, I want the Testing Application to maintain a debug log so that I can track any issues that may occur.		
Priority	High		
Estimated Effort	3 story points		
Acceptance Tests	1. An automatic debug log is created by the Testing Application. 2. The debug log can be opened at any time.		

LuminAR Planning Poker

Planning poker to estimate the user stories of the LuminAR project.

Story	Story Title	Score
1	US 1: Retrieve location updates	3
2	US 2: Display current location	2
3	US 3: Establish a connection	5
4	US 4: Load test nodes	3
5	US 5: Add test nodes to local database	5
6	US 6: Determine the distance of a node	2
7	US 7: Determine the direction (heading) of a node	3
8	US 8: Display a list of locally-stored nodes	2
9	US 9: Filter nodes by a given distance parameter	2
10	US 10: Display nodes within a given distance parameter	2
11	US 11: Modify a locally stored node	2
12	US 12: Remove a locally stored node	2
13	US 13: Insert a new node	2
14	US 14: Retrieve device direction (heading) updates	5
15	US 15: Display the device's direction (heading)	3
16	US 16: Maintain a debug log	3
	TOTAL:	**46**