|  |
| --- |
| ­ |
| Moundville Interactive Experience |
|  |

|  |
| --- |
| Brian Connell, Matt Robbins, Darwin Witt  3/21/2012  C:\Users\Matt\Insync\matthew.r.robbins@gmail.com\cs\cs495\media\graphics\mountain_high_logo_black_emboss.png |

**Change History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Summary** | **Author** | **Date** |
| 0.1 | Initial Write-up | Darwin Witt | 2/6/2012 |
| 0.2 | Restructuring of Format, Listed Functional/Non-Functional Requirements | Darwin Witt | 2/8/2012 |
| 0.3 | Reformatting | Brian Connell | 2/13/2012 |
| 0.4 | Added UI mockups | Darwin Witt | 2/13/2012 |
| 0.5 | Added Use Cases, Activity Diagrams | Matt Robbins | 2/14/2012 |
| 0.6 | Added high-level Class Diagram | Matt Robbins | 2/20/2012 |
| 0.7 | Added diagram descriptions | Matt Robbins | 2/20/2012 |
| 0.8 | Reformatting | Brian Connell | 2/22/2012 |
| 1.0 | Added Design Section | Brian Connell | 3/10/2012 |
| 1.1 | Added Diagrams: Architecture, Sequence, Communication | Matt Robbins | 3/19/2012 |
| 1.2 | Added Diagram descriptions | Darwin Witt | 3/19/2012 |
| 1.3 | Modified original high-level class diagram to reflect architecture design | Matt Robbins | 3/20/2012 |

# Table of Contents

**1. Introduction**

1.1 Purpose

1.2 Scope

1.3 Definitions

**2. Overall Description**

2.1 Product Perspective

2.2 Product Features

2.3 Administrative Portal

2.4 Mobile Application

2.5 Exhibit Feature

2.6 Custom Map Feature

2.7 Proximity Feature

2.8 Constraints

2.9 Assumptions and Dependencies

**3. Functional Requirements**

3.1 Map

3.2 Barcode Scanning

3.3 Proximity Listing

3.4 Content Management Portal

**4. Nonfunctional Requirements**

4.1 Map

4.2 Barcode Scanning

4.3 Proximity Listing

4.4 Content Management Portal

4.5 Article Format

4.6 User Interface

4.7 Database Diagram

4.8 Maintainability

4.9 Modularity of Functions

**5. Requirements Diagrams**

5.1 Use Case Diagrams

5.2 User Activity Diagrams

5.3 High-level Class Diagram

**6. Design Diagrams**

6.1 Architecture Diagram

6.2 Sequence Diagram

6.3 Communication Diagram

# 1. Introduction

### Purpose

The purpose of this document is to outline the requirements specifications for the Moundville Interactive Experience Project. This document outlines the general features specified to address the issue. Furthermore, detailed outlines of functional and nonfunctional requirements are provided along with UML documentation of specific use cases, user activity or interaction flows, and an overarching class diagram to guide further development.

### 1.2 Scope

This proposed system will address the education concerns of the Moundville Archeological Park by embracing smartphone technology as a way to provide an innovative and engaging experience to visitors. Such an immersive educational experience will provide the museum with a modern outlet for educating the public and allowing the visitors to view archeology in a new and exciting way.

### 1.3 Definitions

### Business Domain:

*Sites* – A distinct area which contains a set of artifacts.

*Artifacts* – Specific objects of historical interest. Examples include: pottery, pottery shards, tools, etc.

*Points of Interest* – Markers of locations as specified by the Google Maps User Interface.

### Technology Domain:

3G – A cellular data connection type which is used to communicate with remote databases.

QR – Two-dimensional encoding of data which can be scanned by the camera on an Android smartphone and can link to content or issue application state changes.

# 2. Overall Description

### 2.1 Product Perspective

The Moundville Archeological Park is one of the greatest resources of Archeological data on Native Americans and as such has a wonderful collection of many various artifacts. In fact, there are so many artifacts that there is a great demand for the ability to display and disseminate information about the historical significance of this site to the general public. This coupled with the desire for an immersive educational experience has created a need for a new perspective on the museum experience, both for the indoor museum exhibits and the experience of the outdoor park.

### 2.2 Product Features

The general structure of the system has two main domains, a mobile application for the visitors of the park and a web portal for the administrative purposes of updating and maintaining the data made available to the visitors. Both domains focus on the end goal of providing visitors access to information about artifacts written in concise articles with photographs or artistic renderings. The mobile application can also be thought of as three main methods or experiences with which visitors can interact with the park.

### 2.3 Administrative Portal

The main focus of the administrative portal is as a method of updating and maintaining a catalog of articles on major sites or artifacts. The administrator would access this portal through a web browser and use it to view current articles, add new articles, edit existing articles, and delete articles if needed. Other additional uses of the portal will be to update locations of specific sites or where artifacts were discovered by interacting with a map of the park and printing out QR codes for articles to be placed at exhibit locations.

### 2.4 Mobile Application

A mobile application will be developed for visitors to download onto their android smartphone by scanning a QR code displayed in the museum. This application will open onto a “splash screen” which will display a general welcome message and give the user a menu of three experiences to choose from. These experiences are listed as Exhibit Information, Map of Park, and Explore the Park.

### 2.5 Exhibit Feature

A simple method for interacting with exhibits, this experience will allow a visitor to scan a QR code which will bring up the article associated with that exhibit.

### 2.6 Custom Map Feature

A custom Google Map, this experience will pull up the map of the park on the visitor’s smartphone. Pins placed on the map through the Administrator Portal will populate the map. Visitors can then hover over the pins to display titles of the locations and tap on the pins to bring up the article associated with the site.

### 2.7 Proximity Feature

While outside viewing the mounds, a visitor can use this experience to pull up a list of articles related to their current location. This experience will populate a list of artifacts, sites, and related photographs or artistic renderings for the visitor to browse through. When a visitor taps on an item in the list, this action will bring up the associated article. This list will show only artifacts and sites associated with the visitor’s immediate location, within a predetermined area.

### 2.8 Constraints

The amount of data transferable to the device at one time will be constrained to the amount of available memory space on the device.

### 2.9 Assumptions and Dependencies

The minimum SDK version the application will support is Android 2.1 and the target SDK version is Android 4.0.

The application will require data connection (3G) in order to update maps and database listings in the field.

# 3. Functional Requirements

### 3.1 Map

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Priority** |
| Points of Interest | Points of Interest are set across the park map to indicate to the user locations where artifacts were discovered. Users can tap on these location points to bring up articles showing the information related to these artifacts. | High Priority |
| Region Overlays | Region overlays are composed of either topological overlays from the University of North Carolina dataset or custom-made colored polygonal regions lay over the map. These overlays are meant to highlight major site regions located throughout the park. | Low Priority |

### 3.2 Barcode Scanning

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Priority** |
| Interface with Camera | System call necessary to poll camera data for information encoded on QR code. The associated QR code will have a reference key encoded to direct the application to bring up an article associated with that artifact’s primary key listed in the database. | High Priority |

### 3.3 Proximity Listing

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Priority** |
| Pull GPS Location from phone | System call necessary to poll GPS receiver in phone for coordinates. This is required to search database for all artifacts and sites with geo-locations nearby the User. | High Priority |
| Database Search for locations | Search function necessary to search against the database for artifacts and sites with geo-locations nearby the GPS coordinates of the cell phone. | High Priority |
| Preset Max Number of Items Shown | Listing preset for a maximum number of artifacts and sites returned by the database. This could be useful to minimize computational overhead of the system. | Low Priority |

### 3.4 Content Management Portal

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Priority** |
| Data Entry | A standard form must be specified for the content manager to update records, enter new records, and delete records from the database. | High Priority |
| Geo-Location for records | A Google Map interface would be very helpful for the content manager to edit geo-location data for each record. This allows for a more intuitive interaction with the database for the content manager as well as helps minimize geo-location data entry errors. | Medium Priority |
| Barcode Generating and Print Function | Functionality should be included in the Content Management Portal to allow the content manager to generate and print out QR codes for each record in the database. | Medium Priority |

# 4. Nonfunctional Requirements

### 4.1 Map

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Priority** |
| Google Maps API | This API is required to set up points of interest on the Google Map view as well is to set up information presented in the Popup Bubbles for each point of interest. | High Priority |

### 4.2 Barcode Scanning

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Priority** |
| Barcode Scanning API | This API is required to interpret the QR codes and pull the associated numerical keys that will be used for pulling up the articles associated with the QR codes. | High Priority |

### 4.3 Proximity Listing

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Priority** |
| GPS Services | This Service is required to poll for location information from the GPS receiver embedded in the cell phone. | High Priority |

### 4.4 Content Management Portal

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Priority** |
| Google Maps for setting Location Points | This API or script is required to allow the User to visually select geo-locations for records. | High Priority |

### 4.5 Article Format

The following is the Article Format currently agreed upon:

* Title
* Photograph
* Category
* Style
* Information
* Time Period

### 4.6 User Interface

### Splash Screen

Splash Screen



Map Experience Button – Button Tap sends User to Map Screen



Proximity Experience Button – Button Tap sends User to Proximity Screen



Barcode Experience Button – Button Tap sends User to Barcode Screen

### Map Screen

* Points of Interest markers – Highlights different sites on the map
* Information Bubble
  + Contents – Site Name, Image, More Info Link
  + Appears when User taps on Point of Interest markers
  + Allows User to interact with data associated with site location
* More Info Link – Link Tap sends User to Site Article Screen for that Site
* Map Interaction – Other interactions based on Google Maps UI
  + Includes Zoom In and Zoom Out
  + Includes Different Views – Satellite and Street-only views

Map Screen

### Proximity Screen

* Screen based on Google Android UI standard List View
* Items are included in List based on distance to User Device
  + Ranked in Shortest to Longest Distance within a range limit
* Item Listings
  + Contain Item name and Primary Image
  + Are both Sites and Artifacts
* Tapping on an Item Listing sends User to either Site Article Screen or Artifact Article Screen

Proximity Screen



### Barcode Screen

* Basic Barcode Screen UI in accordance with the Google Android UI specifications. No changes are expected to be made to this format for this application.

Barcode Screen

### Site Article Screen

* Content of the Article includes the following fields
  + Site Name
  + Site Image
  + Site Information
* Site Map Button – Button Tap sends User to Site-Specific Map Screen
* Artifact List Button – Button Tap sends User to Screen based on Proximity Screen UI format, but Items listed are artifacts tagged as associated with the site.

Site Article Screen

### Site-Specific Map Screen

* Points of Interest markers – Highlights different artifact locations on the map
* Information Bubble
  + Contents – Artifact Name, Image, More Info Link
  + Appears when User taps on Point of Interest markers
  + Allows User to interact with data associated with artifact location
* More Info Link – Link Tap sends User to Artifact Article Screen for that Artifact
* Map Interaction – Other interactions based on Google Maps UI
  + Includes Zoom In and Zoom Out

Site-Specific Map Screen

* + Includes Different Views – Satellite and Street-only views

### Artifact Article Screen

* Content of the Article includes the following fields
  + Artifact Name
  + Time Period of Artifact
  + Artifact Image
  + Artifact Information
  + Category Tags – Tags stored in database relating broad categories of artifacts
  + Style Tags – Tags stored in database relating specific artistic or other similarly classifications of artifacts
* Category Tags – Button Tap sends User to Screen based on Proximity Screen UI format, but Items listed are artifacts tagged within the category.

Artifact Article Screen

* Style Tags – Button Tap sends User to Screen based on Proximity Screen UI format, but Items listed are artifacts tagged as having the same style.
* Time Period of Artifact Button- Button Tap sends User to Screen based on Proximity Screen UI format, but Items listed are artifacts tagged as within the time period.

### 4.7 Database Diagram



### 4.8 Maintainability

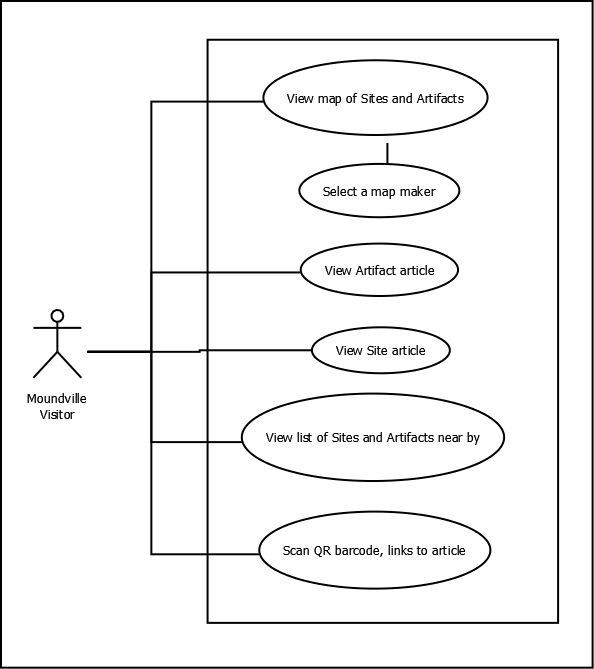
This application is expected to be transferrable to the University of Alabama Museums system after completion to be used in their normal production environment. Therefore, the application design and documentation must be conducted with future maintenance and support in mind.

### 4.9 Modularity of Functions

The expectation for future use of this application includes the desire of function flexibility. Therefore, the design of the application must consider modularity of the different functions in order to maintain their portability as well as to ensure the ability to add further experience functions later in the application’s lifetime.

# 5. Requirements Diagrams

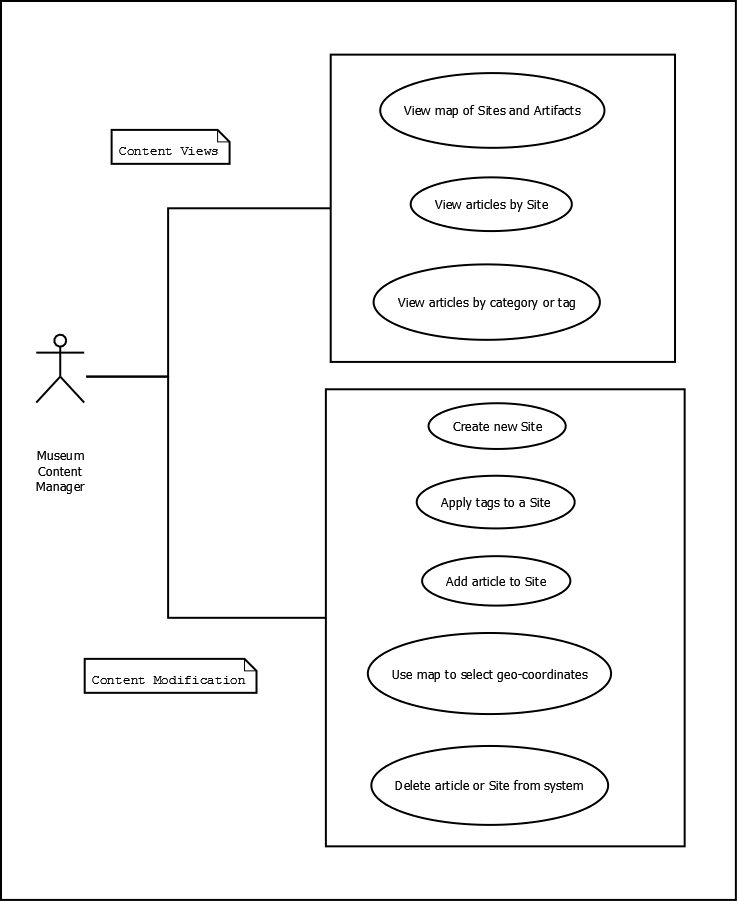
### 5.1 Use Case Diagrams



Use Case - Android Application

|  |  |
| --- | --- |
| **Identifier:** | UC5 |
| **Name:** | “Articles by Proximity” |
| **Description:** | User Experience that allows Users to explore the artifact and site information based on their present location |
| **Actors:** | User |
| **Preconditions:** | User is currently on the Splash Screen |
| **Postconditions:** | User loads an article of interest; Users can then explore related articles based on Artifact Category, Artifact Style, Artifact Time Period, Site Map, and Site Artifacts List associations. |
|  |  |
| **Basic Course of Actions:** | 1. User taps on Proximity Experience Button  2. The Proximity Screen loads  3. User scrolls through Proximity Screen listings  4. User selects site/artifact list item  5.The Site/Artifact Article Screen loads |
| **Alternative Course of Actions A:** | 1. User can press the “Back” button on their device to return to any previous screen |

Use Case Scenario - Android App - UC5



Use Case - Content Management Portal

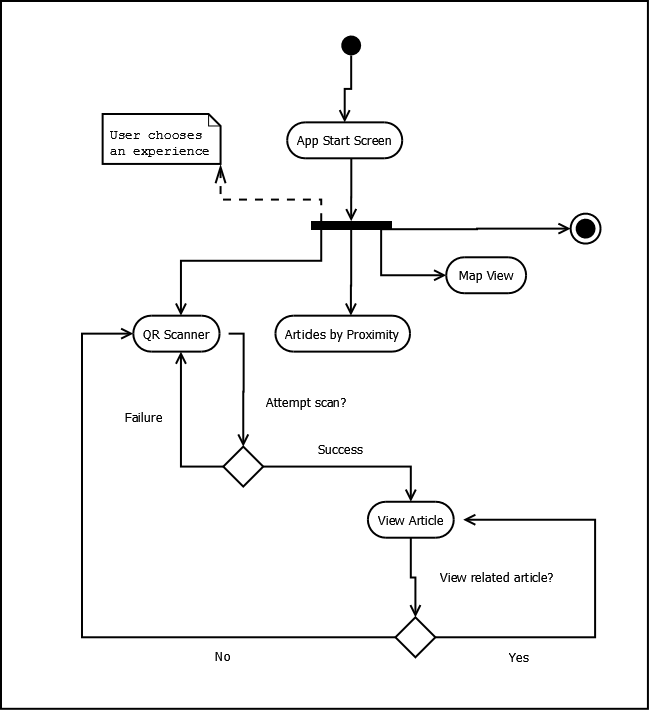
This diagram shows the typical use cases or a user of the database interface. There are two distinct categories of use for this system, viewing the current database and making modification to the database. Each is represented with a box around the associated use cases.

### 5.2 User Activity Diagrams

### 

Activity Diagram - Android App - Map View

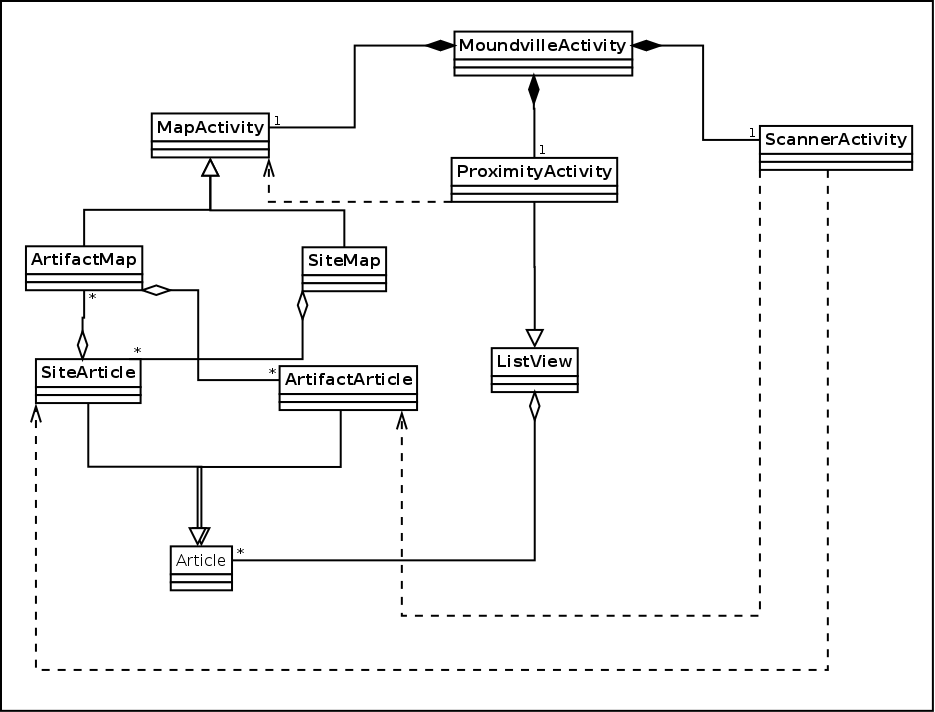
This diagram walks through one of the three major application activities, the *Map View*. The user’s view will change depending on if their location is available or not. In either case, they will see a plot of points of interest overlaid over a Google Map. The user will be able to click on these markers and see a popup containing summary information. From this point they can either return to the map or click the popup to read the full article. Note: The *QR Scanner* and *Article by Proximity* are major activities that are shown here as stubs.



Activity Diagram - Android App - QR Scanner

This diagram walks through one of the three major application activities, the *QR Scanner*. Users will be able to scan display barcodes which will link to a related exhibit article. Once the article appears, the user may return to the scanner activity or view related articles, which may be linked on the article page. Note: The *Article by Proximity* and *Map View* are major activities that are shown here as stubs.

### 5.3 High-level Class Diagram

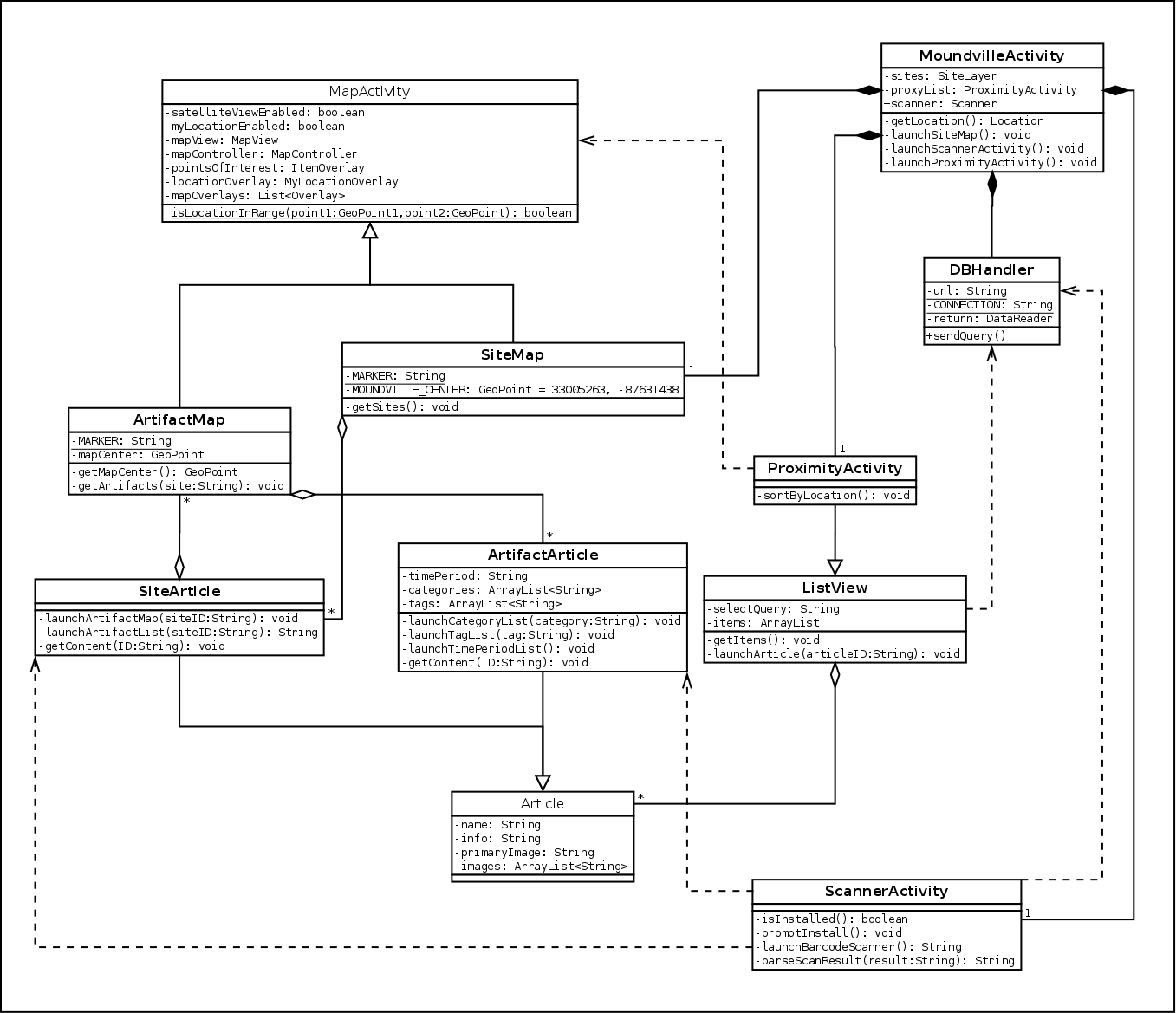


High-level Class Diagram – Android App

This diagram shows the interaction between high-level classes for the Android application component. Three main activities, *MapActivity*, *ProximityActivity*, and *ScannerActivity*, are composite members of *MoundvilleActivity*, which will render the application initial screen.

# 6. Design Diagrams

### 6.1 Architecture Diagram



Architecture Diagram – Android App

This architecture diagram shows the interaction between high-level classes for the Android application component. Three main activities, *SiteMap*, *ProximityActivity*, and *ScannerActivity*, are composite members of *MoundvilleActivity*, which will render the application initial screen.

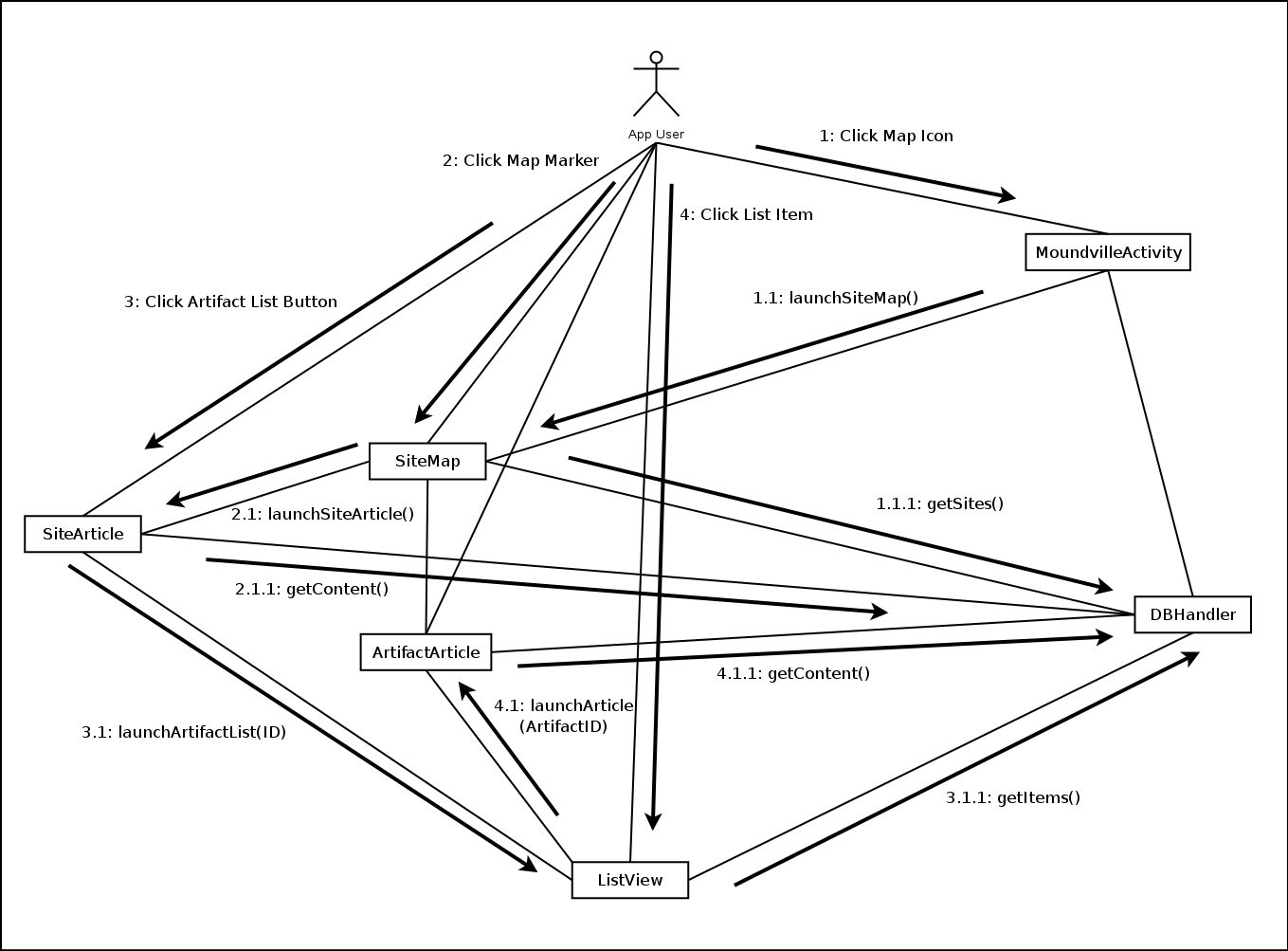
6.2 Sequence Diagram

### C:\Users\Brian.CONNELL\git\moundville\docs\Sequence_Android_App.png

Sequence Diagram - Android App

This diagram shows the sequence of method calls from when a user clicks a map activity through the viewing of an artifact article. For each activity, the user must make a decision which corresponds to a method for launching subsequent activities. At each step, the user can also return to the previous activity (not shown) by using the built-in Android *back* functionality. The diagram also shows how various classes, such as *SiteArticle*, *ListView*, and *ArtifactArticle* must communicate with the *DBHandler* to request content from the remote database.

### 6.3 Communication Diagram



Communication Diagram – Android App

This diagram abstracts the communication relationships among classes in the Architecture Diagram during the specific interaction sequence displayed in the Sequence Diagram. The boxes represent classes defined in the Architecture Diagram while the lines represent the abstract relationships among said classes. Arrows depicting the interaction sequences displayed in the Sequence Diagram are mapped onto these relationships. Labels for these interactions are numbered according to how the actions cascade through the graph of classes, starting with the actor.

Three major behavioral aspects of the class interactions are highlighted through this diagram: degree of dependency on a particular class, degree of flexibility for a specific class, and degree of coupling. Dependencies on classes are related to the number of total interactions the class is called upon. Looking at this particular Communication Diagram, it can be seen that the class with the greatest degree of dependency is the DBHandler. The degree of coupling for a class can be seen as the total number of interactions both leaving a class and acting upon a class. The degree of flexibility is the inverse of the degree of coupling. These behavioral aspects can be used later to determine the priority of security and maintenance tasks for each class.