Application Design Document (Part I)

specifications for the TUMap application

Adama Coulibaly
Willy Lulciuc
David Mason
Sean Moogan
Ben Walker

Date: October 22, 2012

Version 1.0

Contents

1. Document Overview

TUmap is a highly customized campus specific Android application that aims to ease navigation around campus for new students and guests. It will be Android based and levy Google Maps API for routing and an open source mapping toolkit for our custom map. The interface will include a simple drop down menu of buildings on campus. Once a building is selected, a user's location will be automatically polled and a route will be traced to their destination. As the user moves, the phone will update the GPS location and the route if necessary.The Application Design Document (Part I) describes the application architecture and how the requirements are mapped into the design. This document will be a combination of diagrams and text that describes what the diagrams are showing.

1.1 Purpose

The purpose of this document is to provide a detailed description of the TUMap application, developed for Temple University, in a holistic manner. It will serve not only as a source for those who wish to continue with the implementation of the application, but for those also focused on understanding the various object interactions, and how they are modeled, within the application.

1.2 Scope

The document includes much of the necessary architectural components of the application and the design of the internal database used to store campus locations. The description that follows will elaborate on the application's architecture with an array of diagrams (i.e. UML diagrams, state diagrams, sequence diagrams) for clarity.

1.3 Definitions, Acronyms, Abbreviations

Activity – An activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window for you in which you can place your UI with **setContentView**(**View**).

Context – Interface to global information about an application environment.

View – This class represents the basic building block for user interface components.

Intent – Intent is an abstract description of an operation to be performed. It can be used with **startActivity** to launch an **Activity**.

Content provider – A *content provider* manages a shared set of application data. You can store the data in the file system, an SQLite database, on the web, or any other persistent storage location your application can access. Through the content

provider, other applications can query or even modify the data (if the content provider allows it).

AndroidManifest.xml – The manifest presents essential information about the application to the Android system, information the system must have before it can run any of the application's code.

SQLiteDatabase – Wrapper class giving access to db and execution of SQL cmd's.

SQLiteOpenHelper – A utility used for db creation and version management (works with content providers).

Cursor – A structured object that encapsulates the query results.

SimpleCursorAdapter – used to go from data model to a view.

2. Design Overview

2.1 Introduction

This section acts as a brief overview of the applications design. This will place the applciations overall structure into prespective by illustratng the external, and internal, behaviors.

2.2 Application Architecture

2.2.1 System Design

Figure 1 depicts a set of objects and there relationships when the application is running. **Figure 2** illustrates the flow from activity to activty. **Figure 3** provides a state digram for the application

CampusMapActivity – starts the application; opens the database
CampusLocationsDbAdapter – stores the location records; provides search results from query
Location – object that represents a single location on the campus map
LocationAggregator – keeps a collection (i.e. **ArrayList**) of campus locations

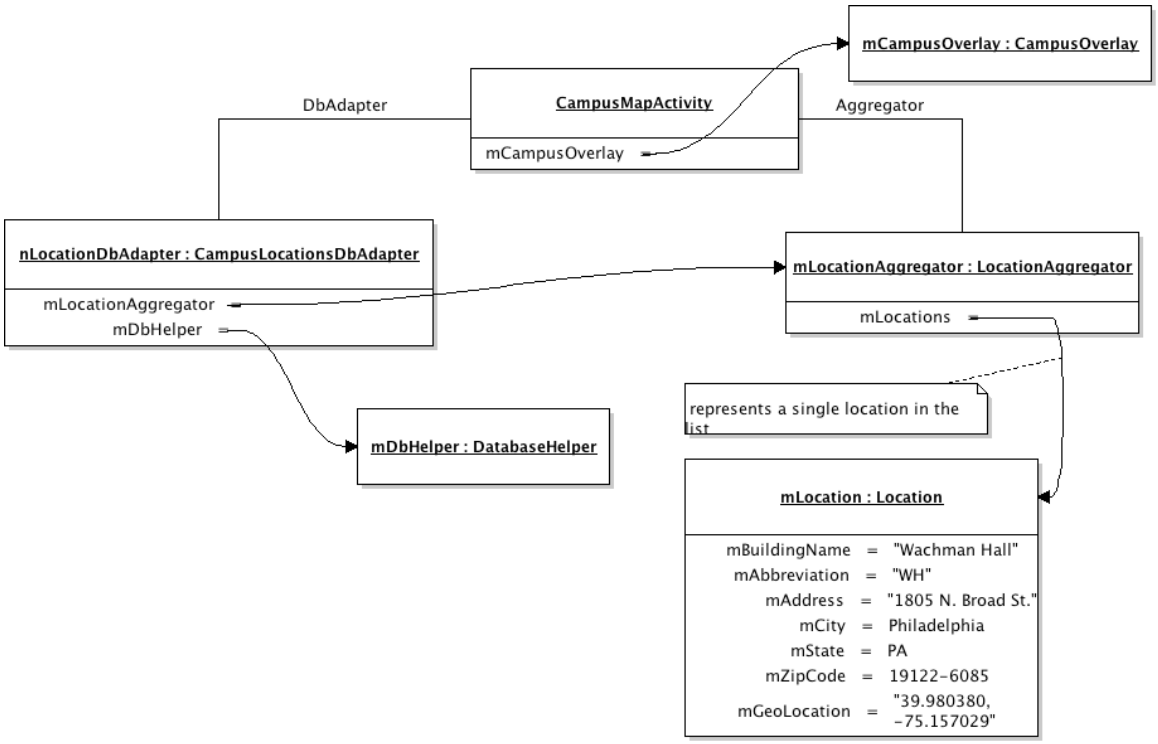

Figure 1: Object Diagram for TUMap

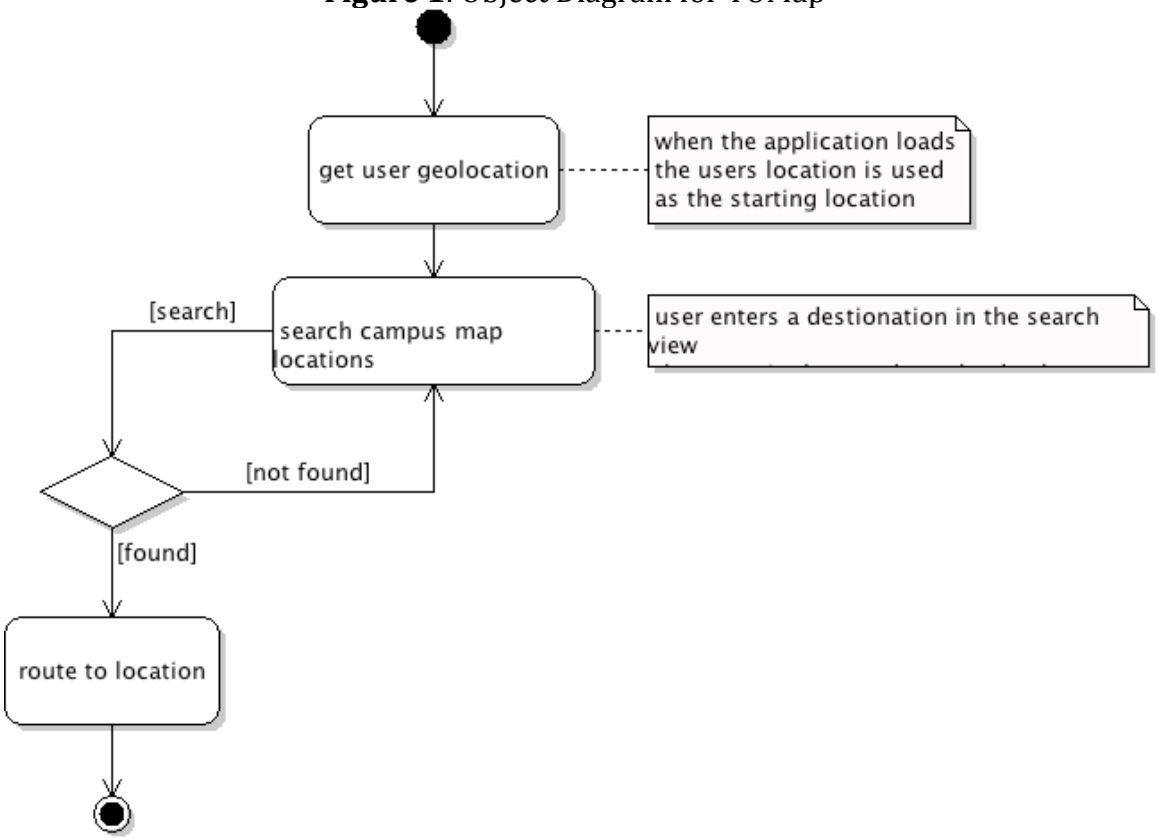

Figure 2: Activty Diagram for TUMap

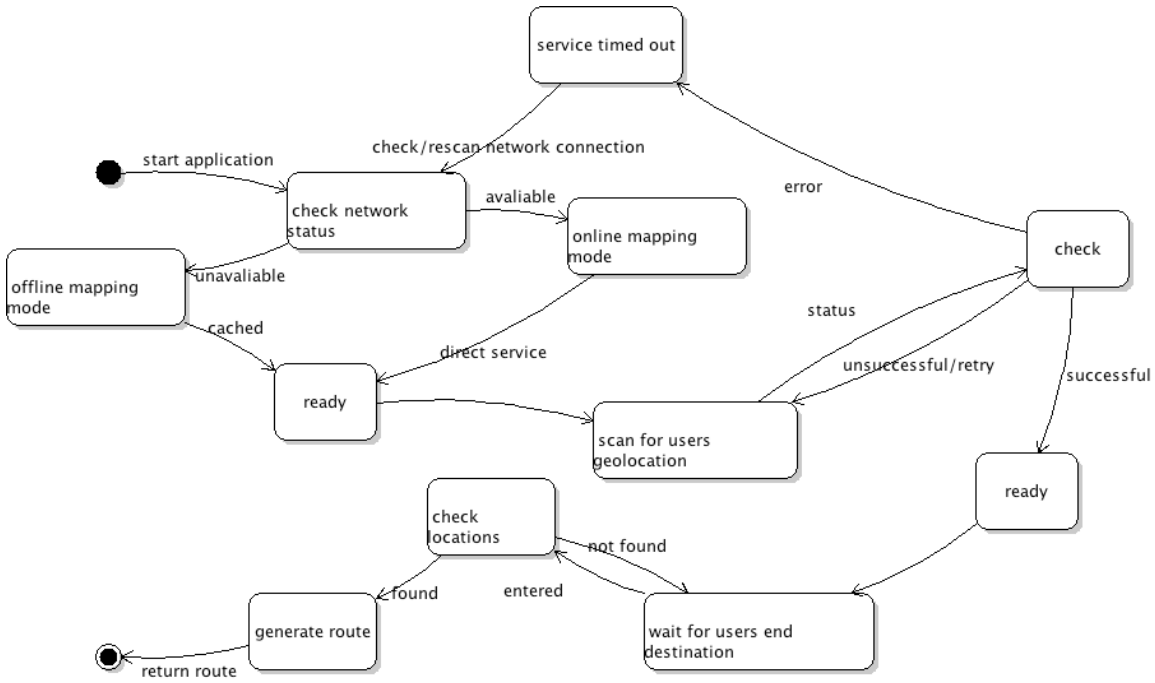

Figure 3: State Diagram for TUMap

2.2.2 System Operations

Figure 4 captures the interaction between parts of the application.

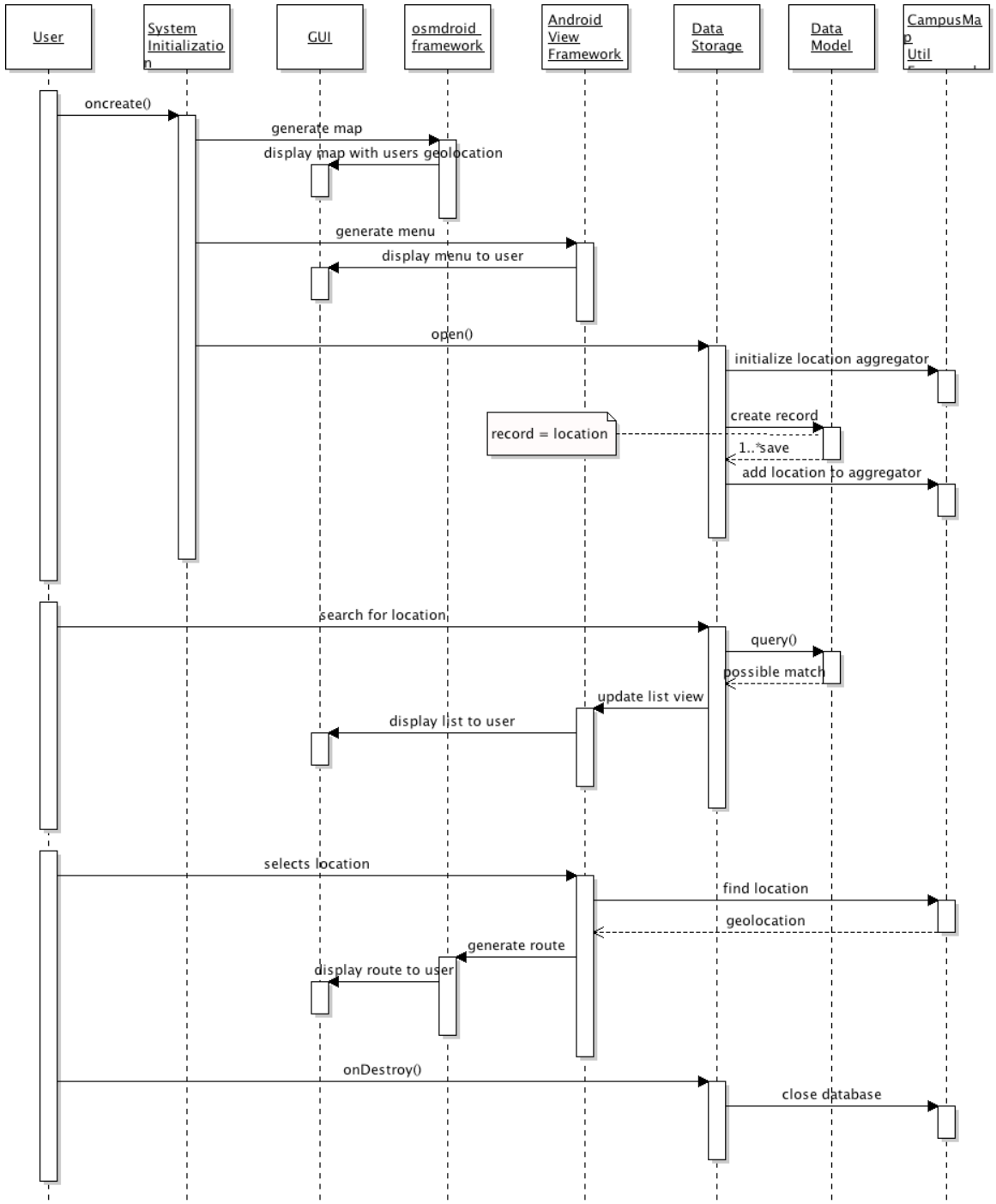

Figure 4: Sequence Diagram for TUMap

2.3 Application Interfaces

Figure 5 depicts the UML model for the main application interface that the user can interact with. The **CampusMapAcivity** class takes care of creating the window in which the user can ineract with. The **CampusMapActivityDemo** activity will create a **Screen** object and will hold a reference to that object; the **Screen** object is more of a conceptual representation of what is occuring during the applications initial rendering of the UI. **Figure 6** depicts the UML diagram for the TUMap application.

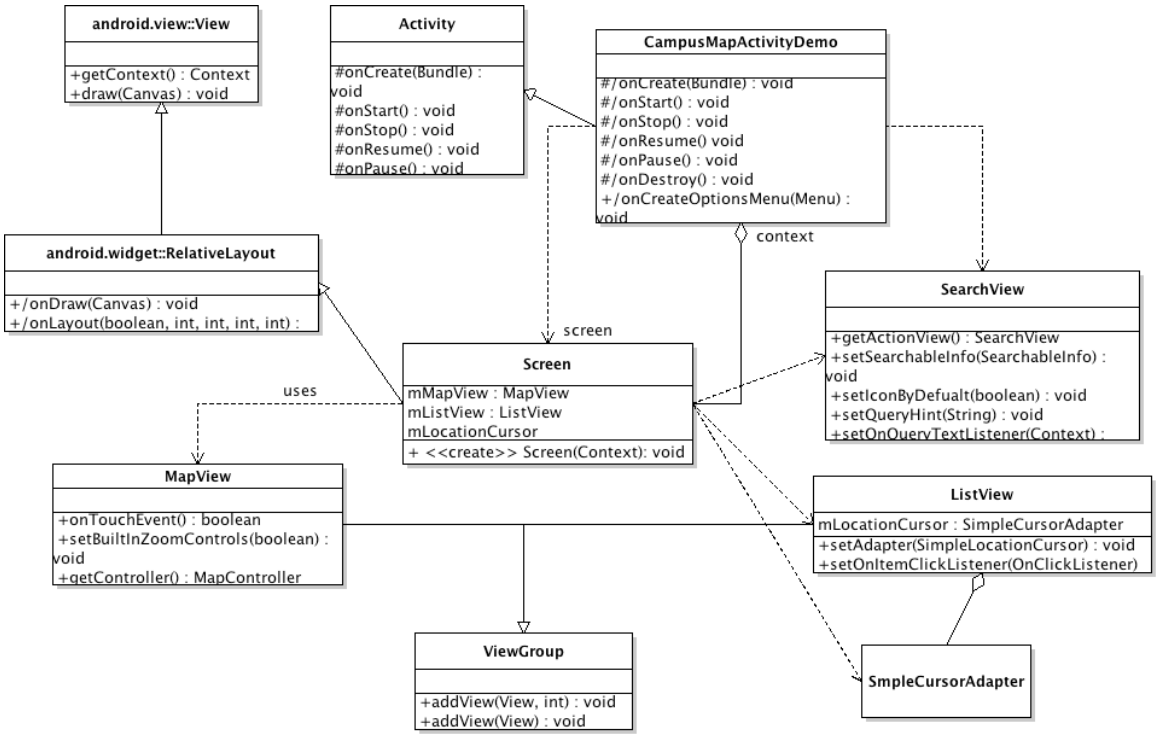

Figure 5: Main Screen Interface

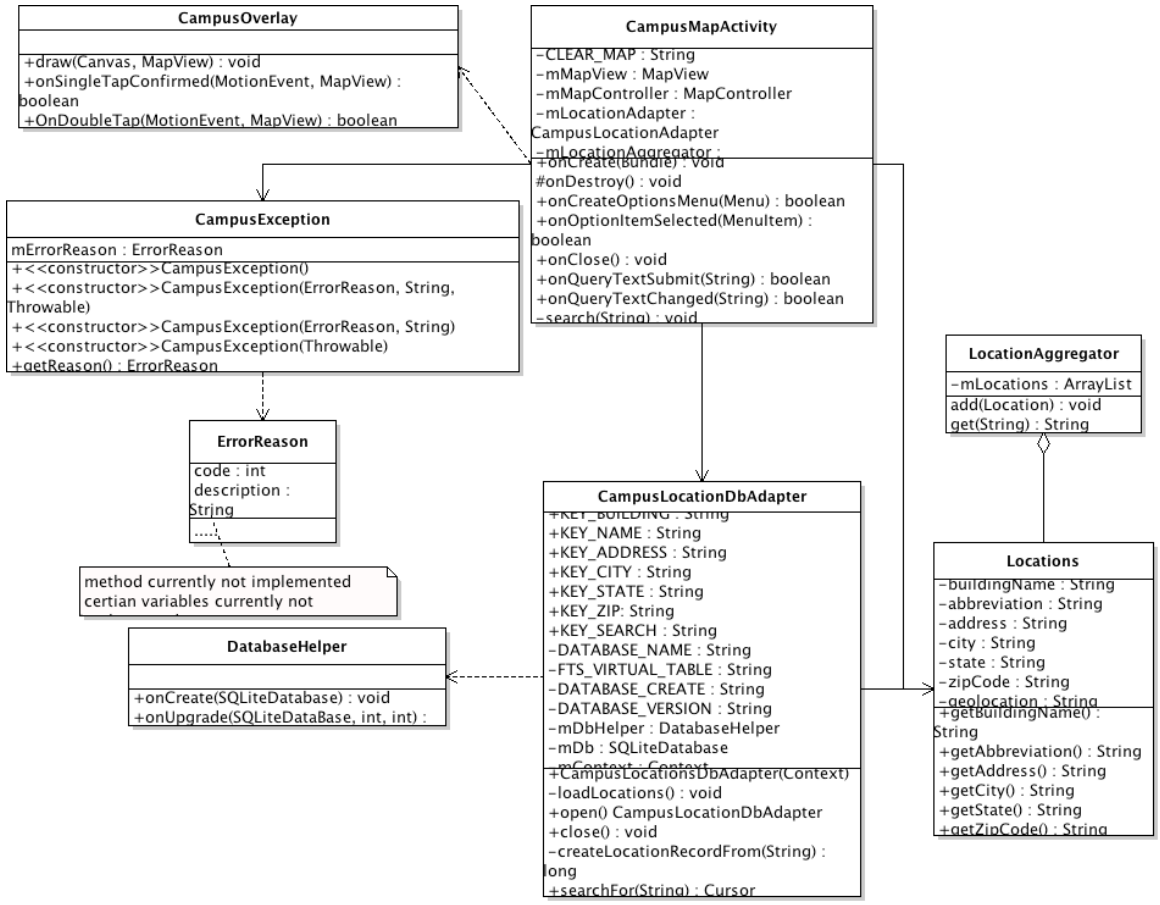

Figure 6: TUMap implementation class level UML

3. Data Model and Storage

The Data Model (**Figure 7**) and Storage (**Figure 8**) classes control behaviors assocated with user search results and information needed for routing from one destination to another. The Data Model holds the desired locations (i.e. address, building name, etc.) while the Data Storage (implemented using SQLite) makes it possible to save the locations for later retrival.

3.1 Data Model Design

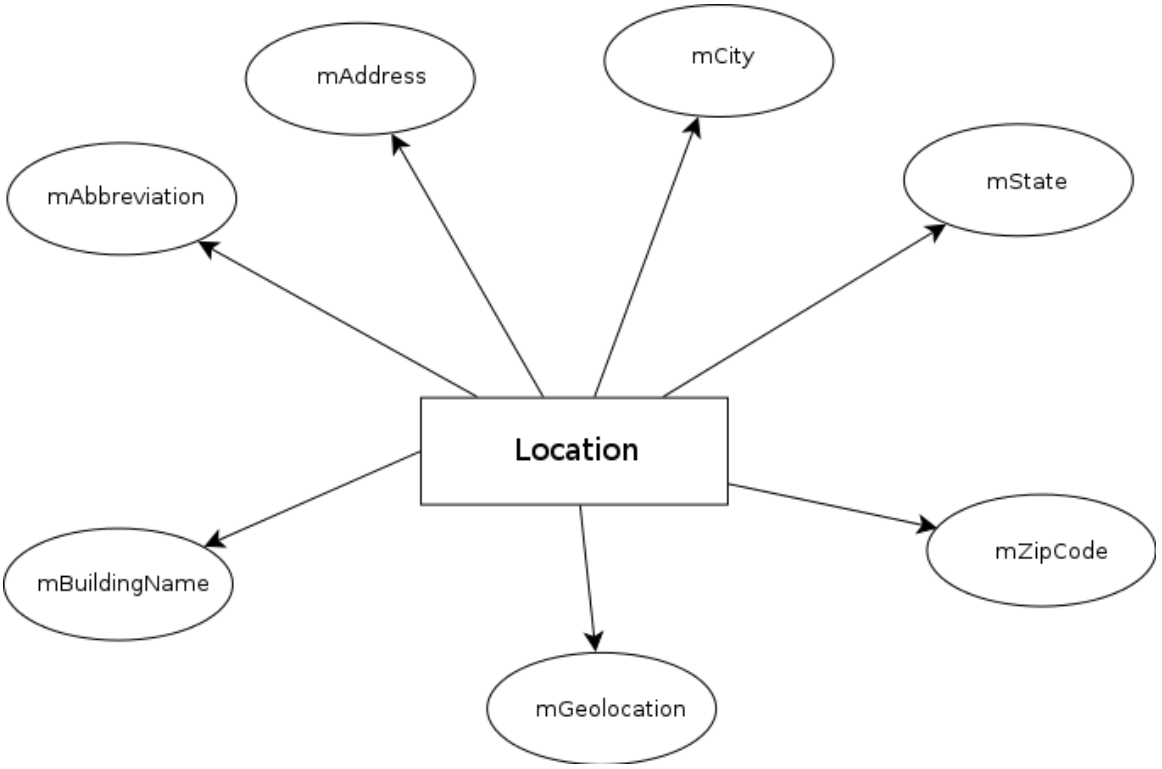

Figure 7: Location Data Model

3.1.1 Data Model Attributes

Name	Type	Description
mBuildingName	String	The name of the builing
mAbbreviation	String	The abbreviation of the building
mAddress	String	The address of the building
mCity	String	The city of the building
mState	String	The state of the building
mZipCode	String	The zip code of the building
mGeolocation	String	The geographical location of the building

3.2 Data Storage Design

Figure 8 illustrates the **SQLite** database model which communicates to the **SearchView** the results of possible matches to the query. Each release of the application will see a number of revisions to the data.

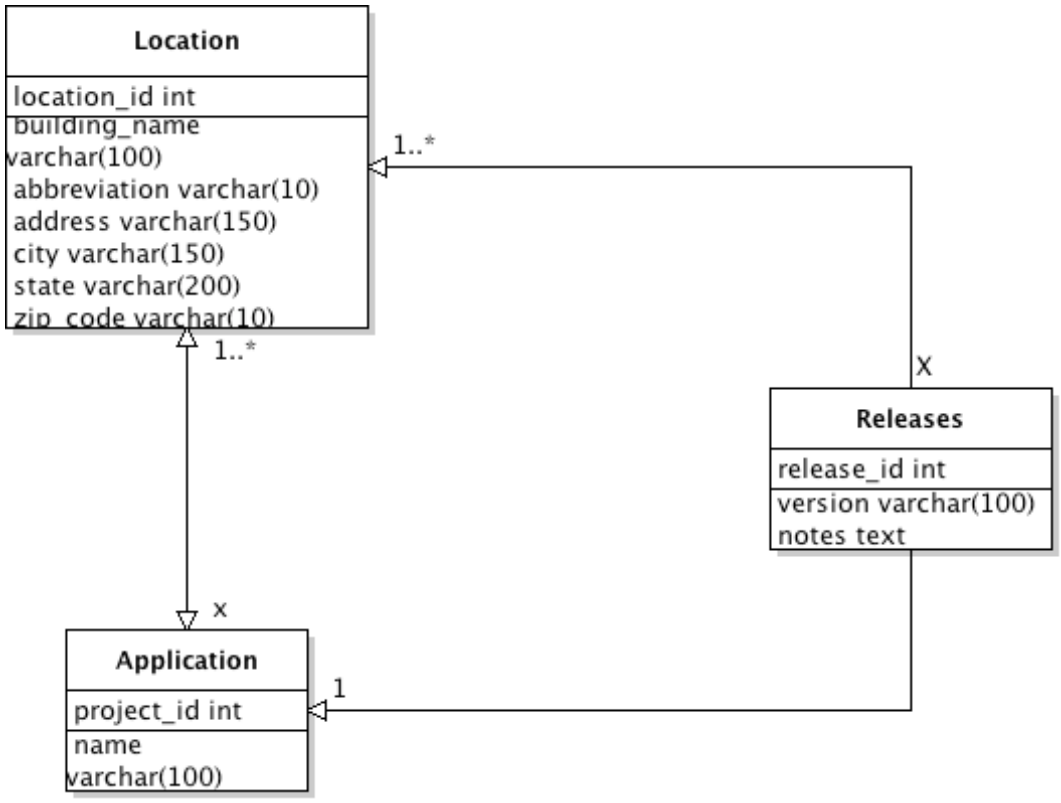

Figure 8: Data Storage (Table) Model

4. Reference

http://developer.android.com/reference/android/app/Activity.html
http://developer.android.com/reference/android/view/View.html
http://developer.android.com/guide/components/fundamentals.html
http://developer.android.com/reference/android/content/Intent.html
http://developer.android.com/guide/topics/manifest/manifest-intro.html
http://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html
http://developer.android.com/reference/android/database/Cursor.html
http://developer.android.com/reference/android/widget/SimpleCursorAdapter.html
http://developer.android.com/reference/android/database/sqlite/package-summary.html
http://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html
http://developer.android.com/guide/topics/data/index.html
http://developer.android.com/guide/topics/data/data-storage.html#db

4.1 Tool Used to Create Diagrams

4.2.1 UML Modeling Tool

Violet – Version 0.21.1, http://horstmann.com/violet/

4.2.2 Entity Relationship Diagramming Tool

Dia – Version 0.95, http://live.gnome.org/Dia