REVOLUTION THEORY SOFTWORKS

Snowfall Design Documentation

Anthony Haddox

Version 1.3

Changelog

Date	Version	Description	Author
4 April 2017	1.0	Document Creation	Anthony Haddox
20 April 2017	1.1	Updated Technology Overview; Added Activity Overview; Added Model Overview	Anthony Haddox
23 April 2017	1.2	Updated Model Overview	Anthony Haddox
24 April 2017	1.3	Updated Model Overview	Anthony Haddox

Contents

Overview

I. Introduction

This document will outline the technical requirements and projected direction of the Snowfall project. This includes but is not limited to:

- Target user base
- Target platform
- Required technologies
- Module implementation
- UI/UX specifications
- Business strategies

The contents of this document are subject to change; any changes will be noted in the changelog.

II. Project Overview

The Snowfall app is a mobile application targeted at education professionals who conduct student behavior observations. Snowfall will be developed for the Android operating system and target mobile phone and tablet devices.

III. Project Name

The developmental name of the project is Snowfall. When a release candidate name is decided upon the documentation and code repository will be updated to reflect this change.

IV. Target User Base

The target user base for Snowfall is individuals in the professional education sector. Ages may range from 21-55 and experience in the sector may range from 1-30 years. Although this demographic is the main target of Snowfall, concerned parents may utilize Snowfall which suggests we should take care to simplify application operations to be friendly to non-industry users.

V. Target Platform

The target platform for Snowfall is the Android operating system. The minimum SDK supported is SDK 21 (Android Lollipop) however the target platform is SDK 25 (Android Nougat) or higher. The UI will be designed to consistently conform to a variety of mobile phone and tablet screen sizes and resolutions.

Technology Overview

I. Introduction

This section will outline the technologies required to fully implement Snowfall.

II. Development Toolkit

Snowfall development will be done using Android Studio v2.3.* or higher. The development environment is the default environment set up by Android Studio during installation; the visual theme of Android Studio can be altered at the discretion of the developer.

Snowfall will use mainly use features from Android API Level 25, however to maintain backwards compatibility Snowfall may use features from Android API Levels 21-24.

To help secure sections of user data, Snowfall will utilize parts of the Java™ Cryptography Architecture.

UI prototyping will be done using the free tier of Fluid UI. While there is a limit to the number of views the user can prototype using the free tier, some of the Snowfall activity changes are similar which negates the need for extra prototyping.

Data visualization will use the open source library Android GraphView and MPAndroidChart. See external libraries.

Source control will make use of GitHub; the developer may choose to use the command line tools or desktop GUI.

III. Development Devices

Snowfall testing will be conducted on a Samsung Galaxy S5 handset. The relevant device specifications are as follows:

- Operating System: Android 5.1.1 (SDK 22)
- Resolution: 1920x1080
- Aspect Ratio: 16:9
- Pixel Density: 432 ppi
- Processor: Qualcomm Snapdragon 801 | Quad-core | 2500 MHz
- Graphics Processor: Adreno 330
- Memory: 2GB
- Storage: 32GB

Activity Overview

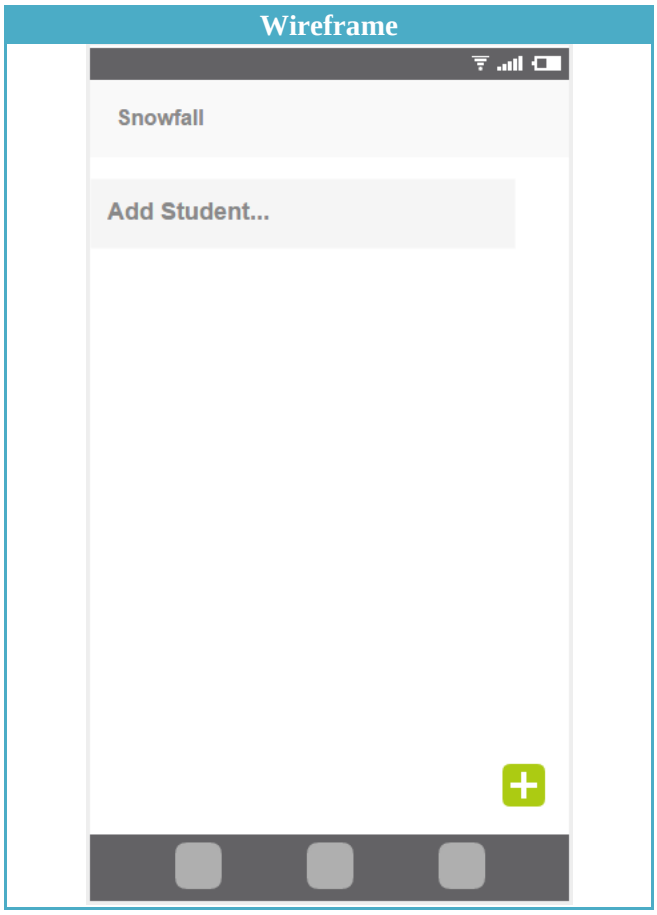
I. Introduction

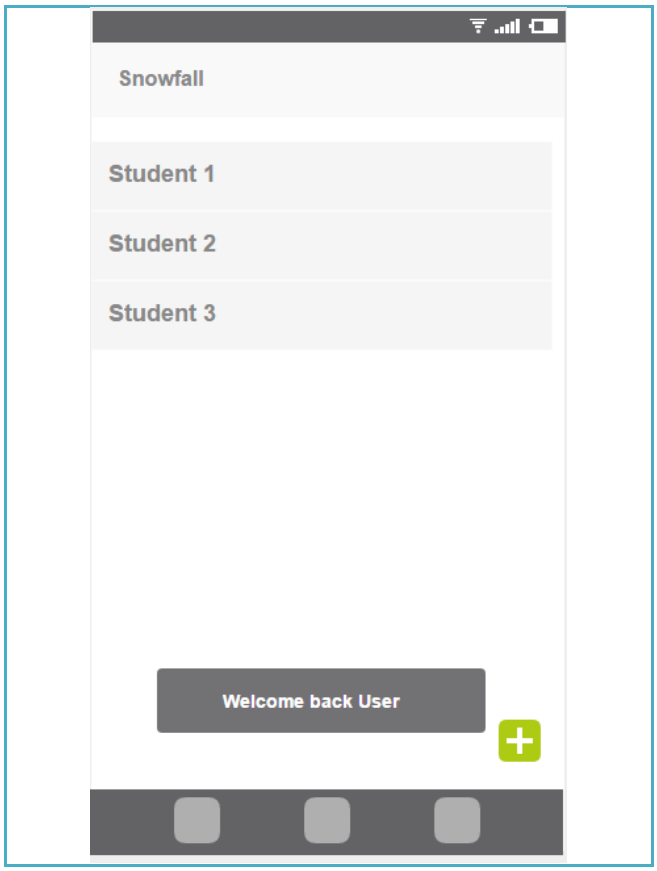
This section will outline the various Android Activities that will be implemented in Snowfall. Note: The wireframe contents may differ from actual implementation.

II. Main Activity

Source File: MainActivity.java

Layout File: activity_main.xml





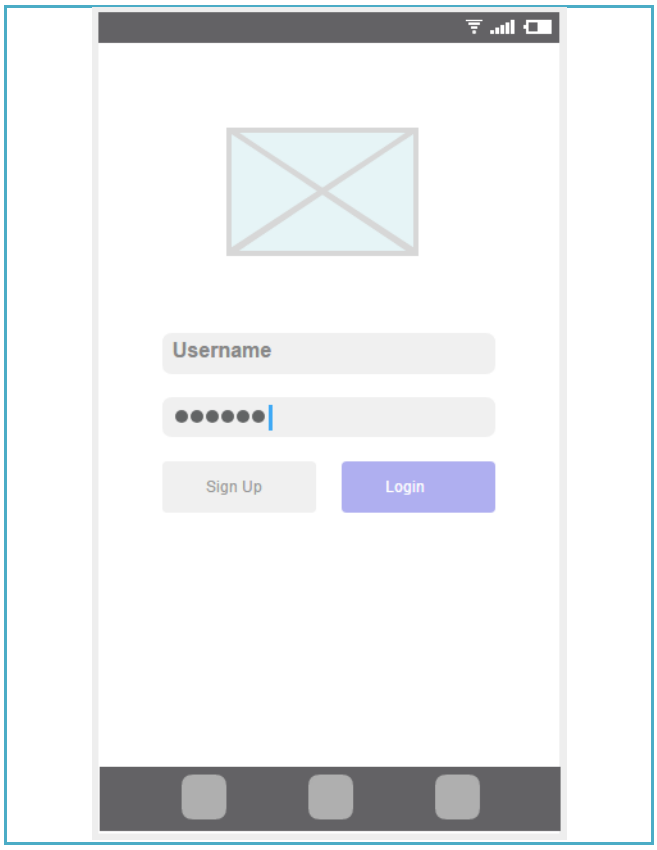
Main Activity is the launch activity of Snowfall. When the user initially launches Snowfall the Main Activity will launch the Login Activity. After a successful login the Main Activity will display a list of Students added by the user. If the user wants to add a new student, they will press the FAB located in the lower right corner of the screen; the FAB will launch the Add Student Activity. If the user selects a student, by pressing on the relevant list item, this will launch the Student Activity. The Main Activity will contain a List View of the added students; this will require an adapter.

III. Login Activity

Source File: LoginActivity.java

Layout File: activity_login.xml

Wireframe



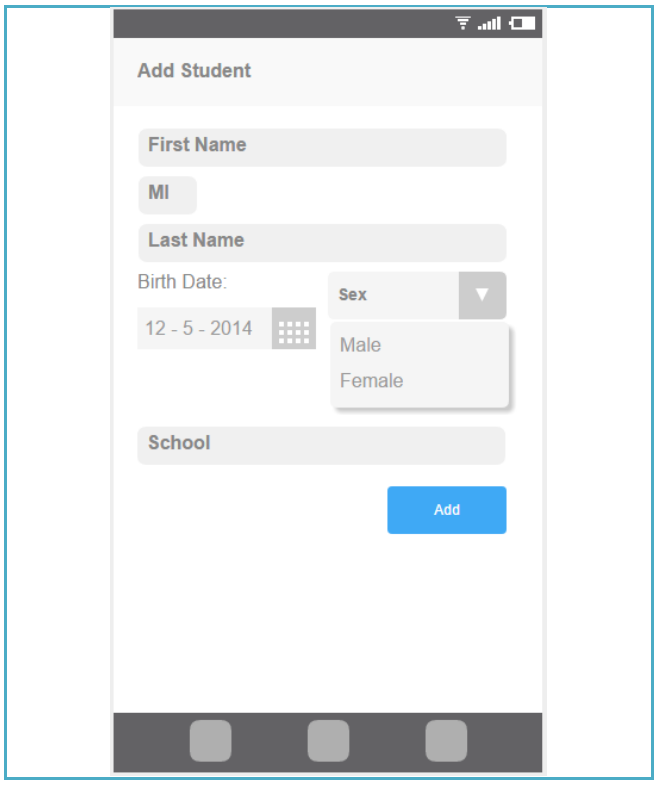
The Login Activity is launched from the Main Activity. The user will be able to enter the credentials, press the Login button, and be returned to the Main Activity upon successful validation. If the user needs to register new credentials they can do so by entering a username and password and pressing the Sign Up button. If the username exists the user will be prompted. The Login Activity contains no provisions to recover lost credentials.

IV. Add Student Activity

Source File: AddStudentActivity.java

Layout File: activity_add_student.xml

Wireframe



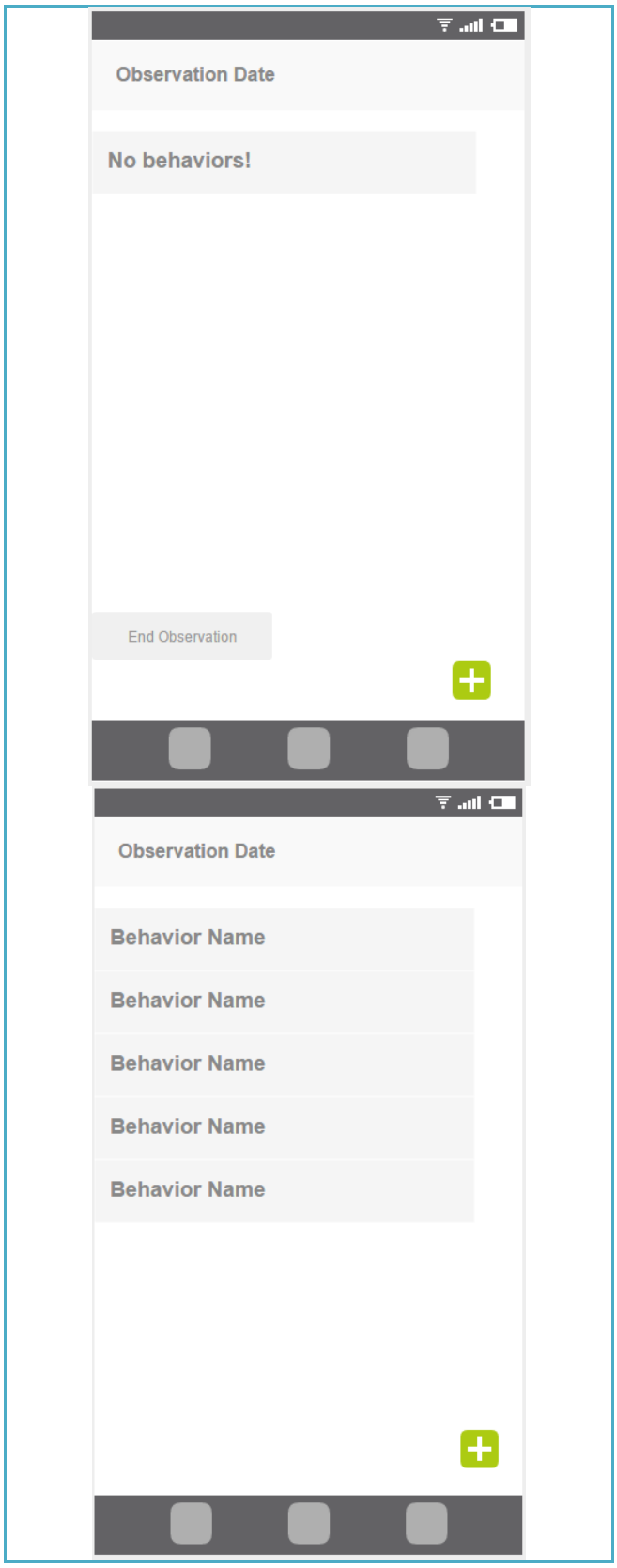
The Add Student Activity allows the user to add the demographic information of a student. The activity contains EditText views for the student's first and last names, middle initial, and school. The middle initial EditText view is constrained to allow a maximum length of one character. The gender of the student will be selected using a ComboBox dropdown [TODO: Find an alternative - Android does not provide one]. The student's date of birth will be chosen using a DatePicker view; this may be launched in a Dialog screen [TODO: Research how to create this screen]. Validation on the date of birth will ensure that the selection is at least a year prior to the date of student creation. When the user clicks the Add button they will be returned to the Main Activity.

V. Observation Activity

Source File: ObservationActivity.java

Layout File: activity_observation.xml

Wireframe

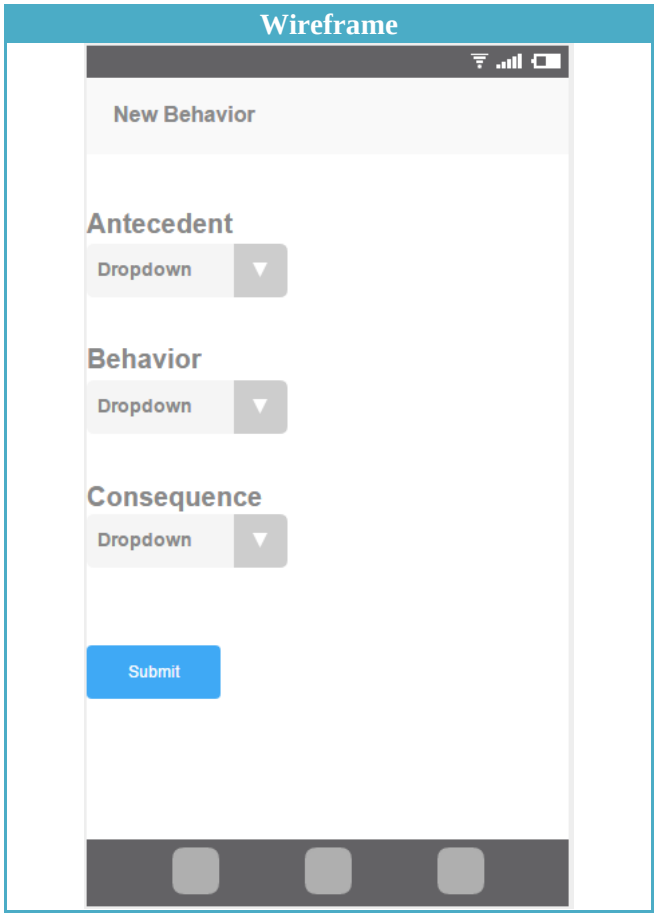

Observation Date
No behaviors!
End Observation
Observation Date
Behavior Name
Behavior Name
Behavior Name
Behavior Name
Behavior Name

The Observation Activity is launched by the user through the Student Activity. When the Observation Activity is initially launched, Snowfall will populate the date of the observation and prompt the user to enter the observation location. The user will then be able to add ActivityBundles, using a FAB in the lower right corner of the device, which log observation metrics or end the observation session. An Observation does not require an ActivityBundle which represents an observation where a student exhibits no behaviors.

VI. Add Behavior Activity

Source File: AddBehaviorActivity.java

Layout File: activity_add_behavior.xml

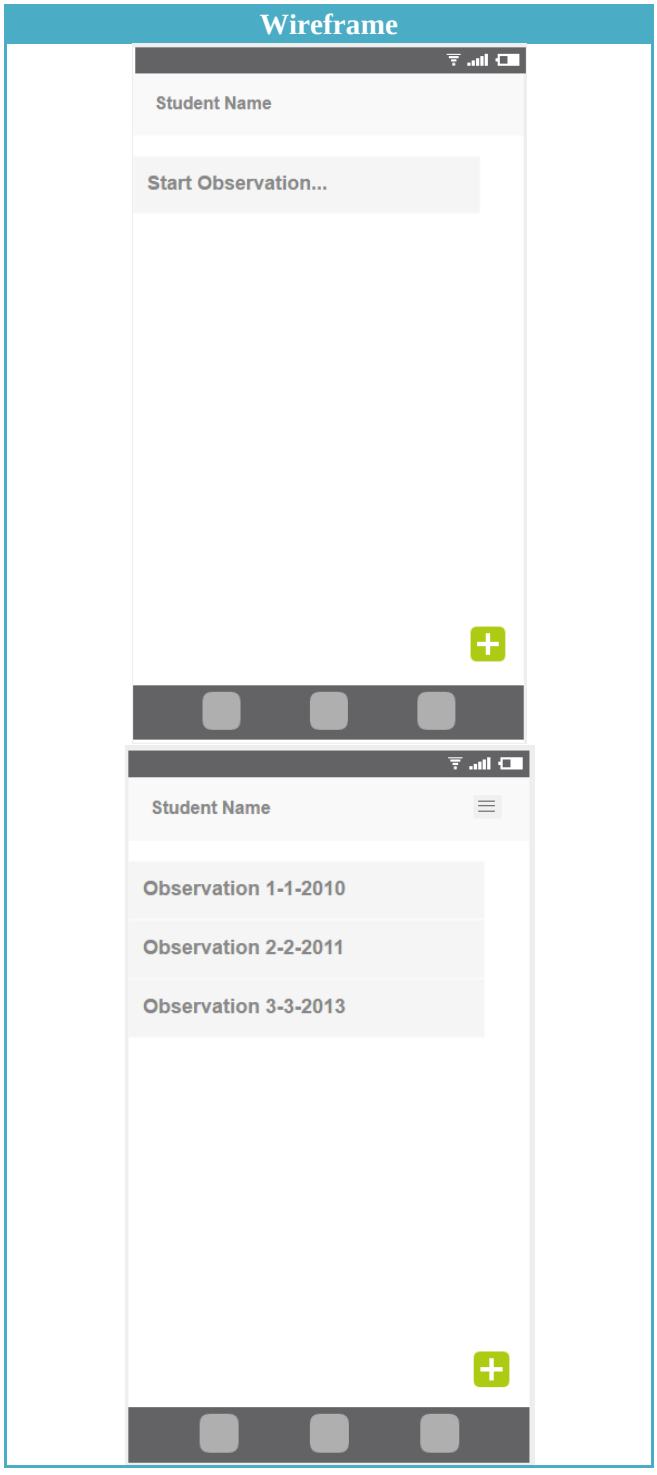


The Add Behavior Activity is launched by the user from the Observation Activity. From a series of dropdown menus the user will select an Activity, Antecedent, Behavior, and Consequence. The available options will correspond to the ActivityBundle Enums. There will be an EditText view for the user to enter the topic of the activity bundle. Snowfall will automatically populate the time of the Activity. When the user clicks Submit they will be returned to the Observation Activity.

VII. Student Activity

Source File: StudentActivity.java

Layout File: activity_student.xml



The Student Activity contains a list of all of the observations of a particular student. From the Student Activity the user can launch either the Observation Activity using the FAB in the lower right corner or the

Report Activity by using the menu in the top right corner of the screen. The Student Activity will require an adapter to list all of the observations.

Model Overview

I. Introduction

Snowfall will make use of several data models to store and manipulate user-supplied data.

II. Student

Source File: Student.java

Fields		
Name	**Data Type**	**Description**
id	Integer	The id of the student (used for ORM)
firstName	String	The first name of the student.
lastName	String	The last name of the student.
dateOfBirth	Date	The student's date of birth.
gradeLevel	Enum - GradeLevel	The grade level of the student.
school	String	The school the student attends.
teacherName	String	The student's instructor.
parentName	String	The parent or legal guardian of the student.

Enum - GradeLevel
TK
K
G1
G2
G3
G4
G5
G6
G7
G8
G9
G10
G11
G12

Methods	
Method Stub	**Description**

III. ActivityBundle

Source File: ActivityBundle.java

Fields		
Name	**Data Type**	**Description**
id	Integer	The id of the activity bundle (used for ORM)
studentID	Integer	The id of the student (used for ORM)
topic	String	The topic of the activity bundle
time	Date	The time that the activity occurs.
activity	Enum-Activity	The activity or task the student is participating in.
antecedent	Enum-Antecedent	The antecedent which triggers the student's behavior.
behavior	Enum-Behavior	The behavioral category exhibited by the student.
consequence	Enum-Consequence	The outcome or consequence of the student's behavior.

Enum - Activity
Large_Group_Desk
Large_Group_Carpet
Small_Group
Centers
Independent_Work
Unstructured_Time

Enum - Antecedent
Given_Instruction
Alone
Peer_Provoke
Engaged_In_Preferred_Activity
Preferred_Activity_Removed
Transition
Sensory_Conflict

Enum - Behavior
Talking_Out
Off_Task
Physical_Aggression
Verbal_Aggression
Elopement
Tantrum
Theft

Harassment
Property_Damage
Crawling_On_Floor
Wandering_Room
Eating_NonFood_Items
Noncompliance

Enum - Consequence
Adult_Attention
Peer_Attention
Got_Preferred_Activity_Or_Item
Sensory_Needs_Met
Ignored
Avoided_Task_Or_Activity
Required_To_Continue
Other

Methods	
Method Stub	**Description**

IV. Observation

Source File: Observation.java

Fields		
Name	**Data Type**	**Description**
id	Integer	Observation id (used for ORM)
studentID	Integer	Student id (used for ORM)
activityBundleList	List<ActivityBundle>	A list containing the ActivityBundles of the observation session.
location	String	The location of the observation.
observationStart	Date	The start time of the observation.
observationEnd	Date	The end time of the observation.

Methods	
Method Stub	**Description**

External Libraries

I. Introduction

Snowfall will utilize external libraries not included with the standard distribution of the Android SDK. This section details the function of these libraries in no discernible order.

II. Android GraphView

From the website:

> GraphView is a library for Android to programmatically create flexible and nice-looking diagrams. It is **easy** to understand, to integrate and to customize. Create Line Graphs, Bar Graphs, Point Graphs, or implement your own custom types.

Link to website: http://www.android-graphview.org/

III. MPAndrioidChart

From the website:

> MPAndroidChart is a powerful and easy to use chart library for Android. It runs on API level 8 and upwards.

Link to website: https://github.com/PhilJay/MPAndroidChart

IV. ORMLite

From the website:

> Object Relational Mapping Lite (ORM Lite) provides some lightweight functionality for persisting Java objects to SQL databases while avoiding the complexity and overhead of more standard ORM packages.

Link to website: http://ormlite.com/sqlite_java_android_orm.shtml