**Software Requirements Specification**

Version 2.0

April 15, 2015

Rohan Gadad

MS Candidate, Information Technology

Rensselaer Polytechnic Institute

Prepared for

ITWS\_6940, Software Engineering

Instructor: Ingrid Liu, Ph.D.

Spring 2015

Table of Contents

[**1.** **Introduction** 3](#_Toc419671522)

[**1.1** **Purpose** 3](#_Toc419671523)

[**1.2** **Scope** 3](#_Toc419671524)

[**1.3** **Out of Scope** 3](#_Toc419671525)

[**1.4** **Project Deliverables** 3](#_Toc419671526)

[**1.5** **Project Management Plan Updates** 4](#_Toc419671527)

[**1.6** **Definitions and Acronyms** 4](#_Toc419671528)

[**2.** **Management Process** 5](#_Toc419671529)

[**2.1** **Risk Management** 5](#_Toc419671530)

[**2.2** **Monitoring and Control Mechanisms** 5](#_Toc419671531)

[**3.** **Configuration Management** 6](#_Toc419671532)

[**3.1** **Android Studio system requirements for development/testing** 6](#_Toc419671533)

[**3.2** **Mobile device configuration** 7](#_Toc419671534)

[**4.** **Application Architecture** 8](#_Toc419671535)

[**4.1** **Use Case Diagram** 8](#_Toc419671536)

[**5.** **Progress** 9](#_Toc419671537)

[**5.1** **Gantt Chart** 9](#_Toc419671538)

[**5.2** **Burndown Chart** 9](#_Toc419671539)

# **Introduction**

This Software Design Document (SDD) establishes the overall design of the system to be used during the development of the Notelity app, which is a note taking app for the Android operating system. The app is being built using modern UI principles and ease of use, along with power features for advanced users.

## **Purpose**

The aim of this project is to develop a productivity app on the Android platform. The app is called Notelity - the basic purpose of the app is to allow the users to take notes. After the development of the app, it will be deployed on the Google Play Store.

## **Scope**

1. View the titles of the notes
2. Modify existing notes or create new notes and store them locally in a SQL/XML based persistent storage
3. Each note will be able to store text
4. Real time search functionality to search for titles and content of the notes
5. Social sharing functionality to share the note on social media sites or email/text messages

## **Out of Scope**

1. Maintenance of the application after the deployment
2. The app will not be expected to work on mobile devices which do not support Android platform

## **Project Deliverables**

Below is a list of artifacts to be delivered

1. **Project documentation**: A complete and up-to-date documentation of the project design and implementation
2. **Source code**: The final working source code of the project
3. **APK file**: The final Android APK file for installation on any Android based smartphone

## **Project Management Plan Updates**

The Project Management Plan will be created and maintained using standard version control processes. Page 3 of this document outlines the Document History, Distribution List, and Plan Approvers.

## **Definitions and Acronyms**

**SDK:** Software Development Kit

**API:** Application Programming Interface

**APK:** Android Application Package

**Android SDK:** The Android SDK includes sample projects with source code, development tools, an emulator, and required libraries to build Android applications. Applications are written using the Java programming language and run on Dalvik, a custom virtual machine designed for embedded use which runs on top of a Linux kernel.

**Android API Level:** API Level is an integer value that uniquely identifies the framework API revision offered by a version of the Android platform. The framework API consists of a core set of packages and classes.

**Shared Preferences:** The SharedPreferences class provides a general framework that allows to save and retrieve persistent key-value pairs of primitive data types.

# **Management Process**

## **Risk Management**

The main issue will be managing time as some of the features of the project may take more time than expected. However, a clearer picture will arise after the 1st prototype will be ready.

The level of difficulty for the development of some of the features is unknown. This may lead to more time being used or changes in the features themselves to simplify the development process.

Due to the relatively flexible requirements of the project, new features may be added or UI design might change due to ease of use or other issues. All this will have to be accounted during the design phase.

## **Monitoring and Control Mechanisms**

The project will make use of Trello. Trello is a cloud based software which can be used to create boards. This can be used to monitor and control the project. Also Trello provides numerous plugins for development – such as created project Gantt charts, burndown charts etc. It is also simple to use for small projects.

Trello will help to follow Kanban for tracking the SDLC. Kanban is a type of Agile. Visualization is an important aspect of Kanban as it allows to understand the work and the workflow. The Kanban board will be maintained on Trello.

# **Configuration Management**

## **Android Studio system requirements for development/testing**

**Windows**

* Microsoft® Windows® 8/7/Vista/2003 (32 or 64-bit)
* 2 GB RAM minimum, 4 GB RAM recommended
* 400 MB hard disk space
* At least 1 GB for Android SDK, emulator system images, and caches
* 1280 x 800 minimum screen resolution
* Java Development Kit (JDK) 7
* Optional for accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality

**Mac OS X**

* Mac® OS X® 10.8.5 or higher, up to 10.9 (Mavericks)
* 2 GB RAM minimum, 4 GB RAM recommended
* 400 MB hard disk space
* At least 1 GB for Android SDK, emulator system images, and caches
* 1280 x 800 minimum screen resolution
* Java Runtime Environment (JRE) 6
* Java Development Kit (JDK) 7
* Optional for accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality

**Linux**

* GNOME or KDE desktop
* GNU C Library (glibc) 2.15 or later
* 2 GB RAM minimum, 4 GB RAM recommended
* 400 MB hard disk space
* At least 1 GB for Android SDK, emulator system images, and caches
* 1280 x 800 minimum screen resolution
* Oracle® Java Development Kit (JDK) 7
* Tested on Ubuntu® 14.04, Trusty Tahr (64-bit distribution capable of running 32-bit applications).

## **Mobile device configuration**

Use smartphone running minimum **Android 4.0.3** version (Android Ice Cream Sandwich) and maximum Android 5.0.1 version (Android Lollipop).

Follow the steps to test source code on the chosen Android smartphone:

* Verify that your application is "**debuggable**" in your manifest or build.gradle file
* Set up your system to detect your device.

If you're developing on **Windows**, you need to install a USB driver for adb. For an installation guide and links to OEM drivers, see the OEM USB Drivers document.

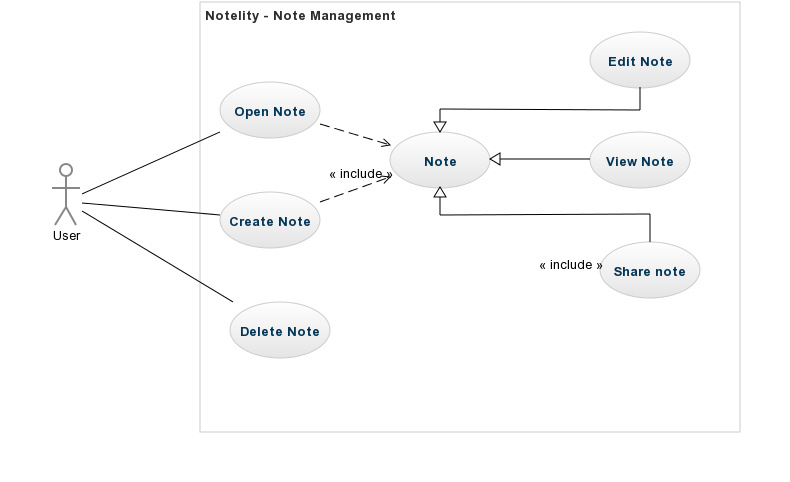
If you're developing on **Mac OS X**, it just works. Skip this step.

If you're developing on Ubuntu Linux, you need to add an udev rules file that contains a USB configuration for each type of device you want to use for development. In the rules file, each device manufacturer is identified by a unique vendor ID, as specified by the ATTR{idVendor} property.

* Enable **USB debugging** on your device. It's in **Settings > Developer options**.

# **Application Architecture**

## **Use Case Diagram**

******

# **Progress**

## **Gantt Chart**



Figure 3: Notelity - Gantt Chart

## **Burndown Chart**

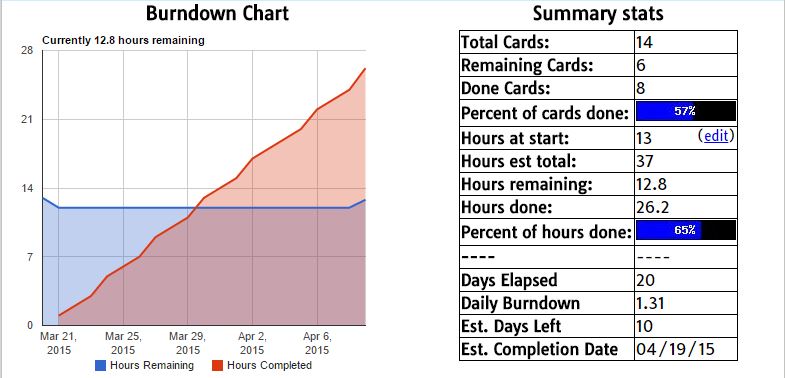


Figure 4: Notelity - Burndown Chart