Software Requirements Specification

for

The Planimeter

Version 0.1

Prepared by

Group Name: PS17-04(Group 4)

KAGAME STEVENLUC	**<Reg No #>**	**<e-mail>**
LUKYAMUZI BENON	**<Reg No #>**	**<e-mail>**
KARUHANGA LINCOLN	**16/U/5465/PS**	**karuhanga475@gmail.com**
OLUMA RODERICK	**<Reg No #>**	**<e-mail>**

Mentor:

Course: **CSC 1304 Practical Skills Development**

Date: **14th July 2017**

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
#0.1	**Kagame** Stevenluc **Lukyamuzi** Benon **Karuhanga** Lincoln **Oluma** Roderick	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	14/07/17

Version	Primary Author(s)	Description of Version	Date Completed
		Initial analysis of feasibility, possible implementation methods, advantages, tradeoffs and difficulty of implementation for each implementation method	

<In this template you will find text bounded by the "<>" symbols. This text appears in italics and is intended to guide you through the template and provide explanations regarding the different sections in this document. There are two types of comments in this document. These comments that are in black are intended specifically for that course. These comments that are in blue are more general and apply to any SRS. Please, make sure to delete all of the comments before submitting the document.

The explanations provided below, do not cover all of the material, but merely, the general nature of the information you would usually find in SRS documents. It is based on the IEEE requirements and was adapted specifically for the needs of CSC 1304 Practical Skills Development course. Most of the sections in this template are required sections, i.e. you must include them in your version of the document. Failure to do so will result in marks deductions. Optional sections will be explicitly marked as optional.

1 Introduction

<TO DO: Please provide a brief introduction to your project and a brief overview of what the reader will find in this section.>

1.1 Document Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.
TO DO: Write 1-2 paragraphs describing the purpose of this document as explained above.>

1.2 Product Scope

The Planimeter is a mobile application that when completed will provide the functionality required to estimate the area of a plot of land. Currently available methods require an individual to hire land surveyors that use specialized equipment like Theodolites to make recordings and later calculations based on the recordings. These methods are tedious, especially considering the fact that the majority of people seek only an estimate of the size of a plot to make quick decisions. The application seeks to provide an intuitive, affordable, quick, efficient and reasonably accurate method to help users make this estimate.

1.3 Intended Audience and Document Overview

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers (In your case it would probably be the "client" and the course facilitator). Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

1.4 Definitions, Acronyms and Abbreviations

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.
TO DO: Please provide a list of all abbreviations and acronyms used in this document sorted in alphabetical order.>
GPS- Global Positioning System

1.5 Document Conventions

<In general this document follows the IEEE formatting requirements. Use Arial font size 11, or 12 throughout the document for text. Use italics for comments. Document text should be single spaced and maintain the 1" margins found in this template. For Section and Subsection titles please follow the template.

TO DO: Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. Sometimes, it is useful to divide this section to several sections, e.g., Formatting Conventions, Naming Conventions, etc.>

1.6 References and Acknowledgments

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document.

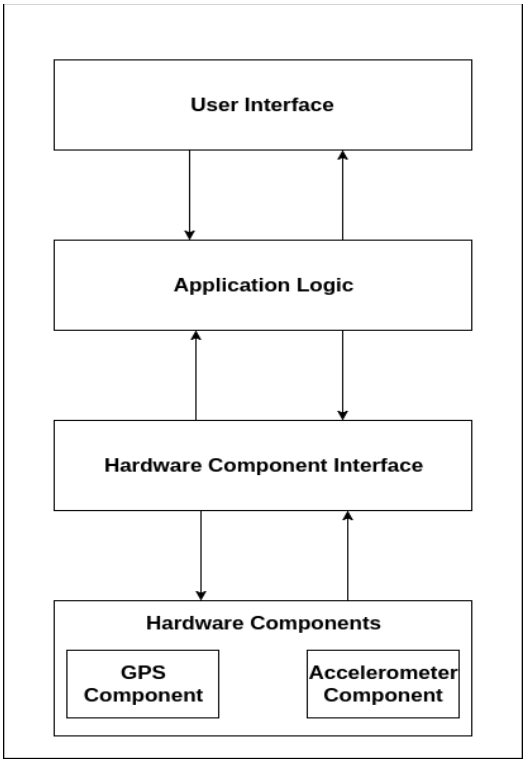
TO DO: Use the standard IEEE citation guide for this section. An example citation guide can be found at this URL http://www.ieee.org/documents/ieeecitationref.pdf .>
<Wikipedia links for smart phone and mobile app>
<Github links for used projects>
<Material Design Guidelines>

2 Overall Description

2.1 Product Perspective

The system will consist of a newly developed, single, standalone mobile application that will be installed on the user's device. It will make use of two of the device's sensors, the GPS component, which will track a user's position at various points along the boundaries of the plot and the Accelerometer which will provide data that will be used to approximate movement, that is footsteps. These components are included in nearly all smart phones currently available on the market.

In as much as the technology utilized by smart phones has come a long way, the devices still don't have the capability to understand their environment which hinders functionality like independently using the devices to take measurements and track movement. The system will therefore utilize the data from the above mentioned components to simulate this kind of behaviour and achieve the objective without the need for pre-callibration and other types of pre-sets. The system will make use of data from both components to improve accuracy of the results provided.



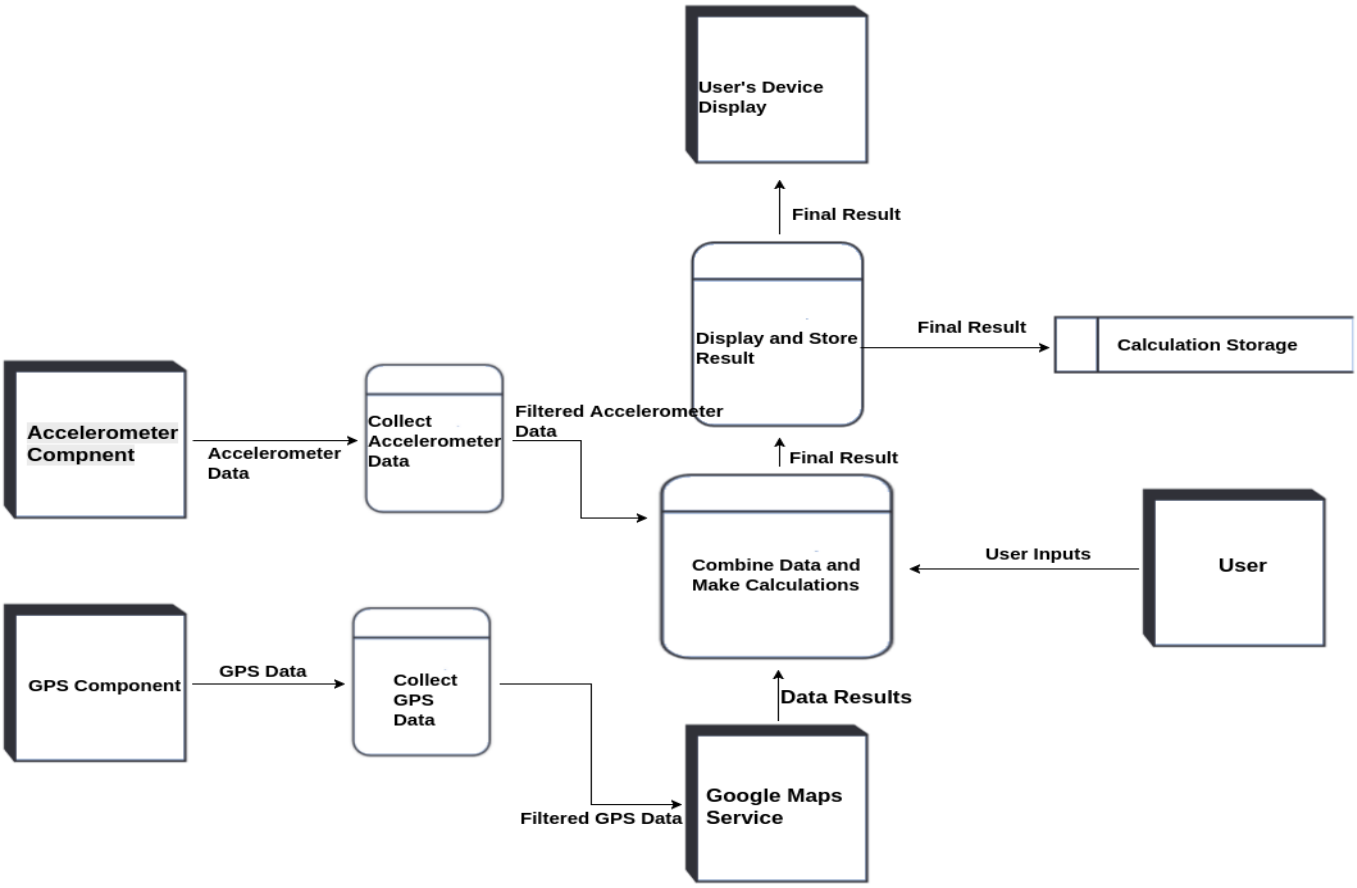
2.1- Operating Environment for the System

2.2 Product Functionality

The application must let the user measure the length of a plot boundary and calculate the area of a plot. This process will include;

- Taking GPS readings and measuring distance between two recorded points
- Take Accelerometer readings and approximate walking behaviour
- Generate approximation of plot layout
- Calculate the area of a plot based on this recorded data

- Store saved calculations



2.2- Level 1 Data Flow Diagram for the System

2.3 Users and Characteristics

The system is designed and developed targeted towards users from any background that have basic knowledge of how to utilize a smart phone. The product does not come with any need for technical expertise, as that is the problem it seeks to solve.

Any individual with the need to measure the area of an expanse of land can therefore easily make use of the application.

It will however be of frequent use to people whose work regularly requires this use case. These may include real estate agents, brokers, building contractors, prospective land owners and real estate investors.

2.4 Operating Environment

The rule for selecting hardware and software is that components of the application must be functionally efficient, capable of interfacing with other software if required to in the future and easy to maintain.

The following table displays a summary of the system's operating environment.

Specification	Requirement
Hardware Platform	Smart phones or Tablet devices that fit the following criteria; • Run the Android OS • Have inbuilt GPS and/or Accelerometer components
Device Operating System	Android OS, minimum Android 4.1- *Kitkat*

As will be indicated in the specification details, the system will not have any other dependencies and is also not expected to interfere with the smooth running of any of the other software on the device.

The minimum software requirements chosen are influenced by the fact that low end devices running this version had the first widely adopted support for GPS in Android.

Please refer to the diagram as in 2.1- Product Perspective

2.5 Design and Implementation Constraints

The biggest challenge the project is faced with is the fact that low-end android devices which are the kind commonly used on the continent lack the more advanced sensors such as the step sensor. This has therefore necessitated the creative use of the available sensors, which may not produce results as accurate as the above mentioned sensors.

In addition to this, the team faces the following implementation constraints;

1. The availability of devices with both GPS and Accelerometer hardware components is not guaranteed. This necessitates the design of each system independently and later combining the results of data from both components, such that in the event that one of the components is not available for any reason, the other can function independently, albeit not as accurately.

2. Difficulty of access to a wide range of android devices for testing the system, coupled with lack of appropriate devices to run the required emulators.
3. The system will make use of the GPS component that is widely known for its power drain especially on low end devices. Developers will therefore have to ensure the efficient use of this component, especially in light of the fact that both hardware components will be in use at the same time if available.
4. In order to ensure accurate GPS readings, devices require internet access which is not a guarantee in some areas.
5. Design for minimal future maintenance. In order to reduce technical debt that might be incurred in the future, the team will need to ensure that all components included the system are resilient and easily improvable and/or interchangeable if the need arises.
6. Where feasible, developers are expected to follow Android design guidelines, which include but are not limited to following material design principles and efficient memory use.

2.6 User Documentation

As previously mentioned, the system is developed with non technical and non tech-savvy people in mind. The system is targeted towards smart phone owners who have the knowledge of basic application usage.
However, for the totally unacquainted, the team can be reached on social media, email and telephone. Wikipedia also offers excellent resources in this case, the links of which are provided in the appendix.

2.7 Assumptions and Dependencies

The following are the most sensitive assumptions, whose change would have significant impact;

- As mentioned in the software requirements, the project will make use of two, non proprietary software components. Since these are open source projects, they are subject to change at ay time ad there is no guarantee that functionality will still be maintained after the change.

- The project's use of the GPS component also necessitates the availability of a reliable internet connection during recordings to achieve the best accuracy possible.
- Readings taken from the user's device can only be as good as the sensors embedded in the device. Inaccurate readings due to faulty sensors or otherwise may in turn produce inaccurate results.
- The software will depend on the availability of both sensors for utmost accuracy. Absence of either may significantly reduce the accuracy of readings.
- Lastly, user cooperation is expected, especially because their input will be required when taking accelerometer readings. Inaccurate input will in turn produce inaccurate results.

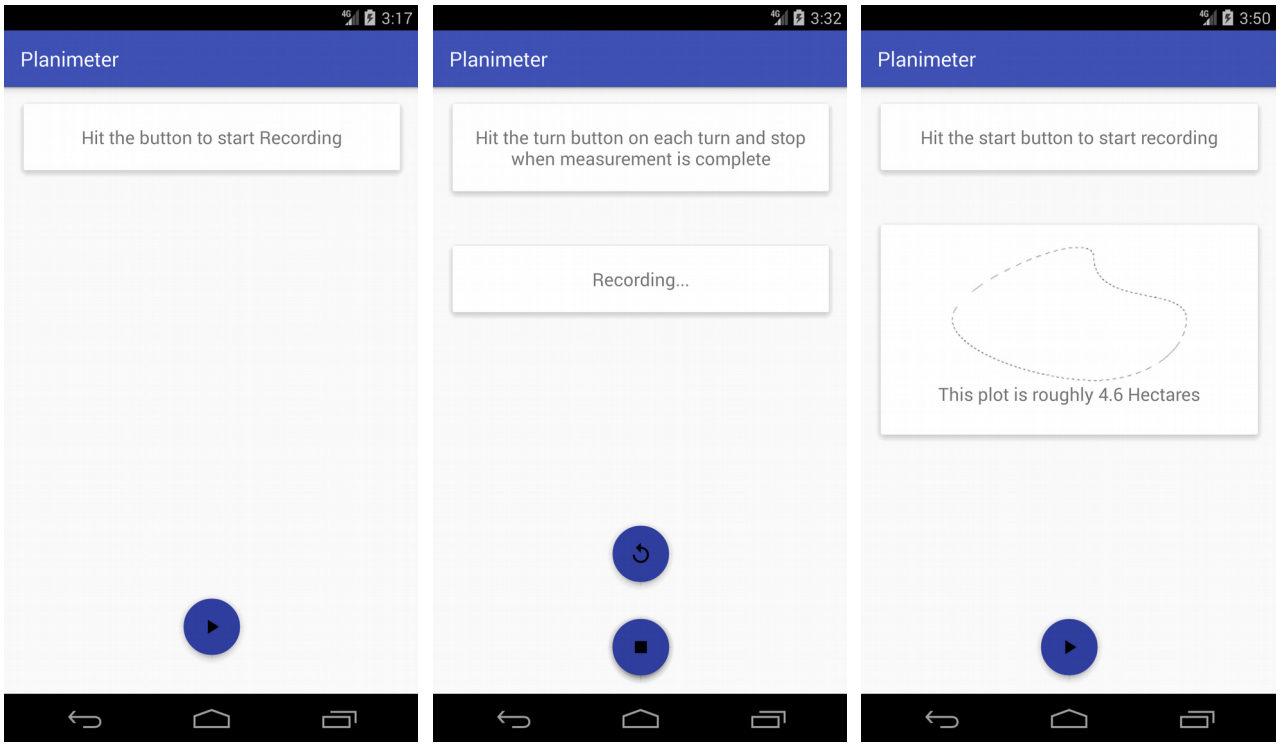
3 Specific Requirements

3.1 External Interface Requirements

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware and software interfaces for the application. In addition to these, the section provides basic prototypes of the proposed user interface for the application. The application is a standalone system and therefore there is no need for a standard definition of a communication interface specification.

3.1.1 User Interfaces

Team members will follow material design principles and guidelines wile developing the application. Consistency should be maintained across devices. Samples of major interfaces are provided below.



3.1.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware. You are not required to specify what protocols you will be using to communicate with the hardware, but it will be usually included in this part as well.

TO DO: Please provide a short description of the different hardware interfaces. If you will be using some special libraries to communicate with your software mention them here. In case you have more than one hardware interface divide this section into subsections.>

The proposed mobile application does not have any designated hardware required for its operation and it therefore does not have any need for direct hardware interfaces.

However, the application will interact with the device's physical GPS and Accelerometer components which are both natively managed by default by software components built into the android platform.

3.1.2.1 Interaction with the GPS component

Interaction with the GPS which involves making location data requests and receiving data readings from the GPS component. The android platform provides a software component, *Location Manager to* to allow developers access the GPS component.

3.1.2.2 Interaction with the Accelerometer

Interaction with the accelerometer involves making repeated calls to the devices accelerometer sensor and get readings which are then run through algorithms to put them into context and then take actions based on the readings. Access to this component is done via the *Sensor Manager which allows developers access this data.*

3.1.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems (Android? iOS? Windows? Linux? Etc…), tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.

TO DO: The previous part illustrates some of the information you would usually include in this part of the SRS document. To make things simpler, you are only required to describe the specific interface with the operating system.>

As previously mentioned, the application will be developed for the android platform. The team will make use of the default recommended software components for the android platform.

In addition to these, the project will make use of two software components previously implemented and hosted on *github.com* under a free to use license by Daniel Murray and It Kovian, the links of which have been provided in the appendix. Both projects implement unique ways of taking accelerometer readings and applying filtering algorithms that have a final result of isolating *noise,* that is non required recordings from the accelerometer

component and therefore make it easier to identify movement, i.e *footsteps,* which is the intended use for the Accelerometer in the project.

3.2 Functional Requirements

< Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. This section is the direct continuation of section 2.2 where you have specified the general functional requirements. Here, you should list in detail the different product functions with specific explanations regarding every function.

TO DO: Break the functional requirements to several functional areas and divide this section into subsections accordingly. Provide a detailed list of all product operations related to these functional areas.

3.3 Behaviour Requirements

3.3.1 Use Case View

<A use case defines a goal-oriented set of interactions between external actors and the system under consideration. Since sometimes we will not be able to specify completely the behaviour of the system by just State Diagrams, we use use-cases to complete what we have already started in section 3.3.1.

TO DO: Provide a use case diagram which will encapsulate the entire system and all possible actors. Do not include detailed use case descriptions, but make sure to include a short description of what every use-case is, who are the actors in your diagram. For more information please refer to Wikipedia article http://en.wikipedia.org/wiki/Use_case .>

4 Other Non-functional Requirements

4.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.
TODO: Provide at least 5 different performance requirements based on the information you collected from the client. For example you can say "1. Any mobile money transaction will not take more than 10 seconds, etc…>

4.2 Safety and Security Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied. Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements.
TODO:

- *Provide at least 3 different safety requirements based on your interview with the client or, on your research, and again you need to be creative here.*
- *Describe briefly what level of security is expected from this product by your client and provide a bulleted (or numbered) list of the major security requirements.>*

4.3 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

TODO: Use subsections (e.g., 4.3.1 Reliability, 4.3.2 Portability, etc…) provide requirements related to the different software quality attributes. Base the information you include in these subsections on the material you have learned in the class. Make sure, that you do not just write "This software shall be maintainable…" Indicate how you plan to achieve it, & etc…Do not forget to include such attributes as the design for change. Please note that you need to include at least 2 quality attributes, but it is the mere minimum and it will not receive the full marks.>

5 Other Requirements

*<This section is **Optional**. Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal*

requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

5.1 Appendix B - Group Log

<Please include here all the minutes from your group meetings and meetings with users, your group activities, and any other relevant information that will assist the Course Facilitator to determine the effort put forth to produce this document>