Stone Identification App
Software Requirements Specification

Genevieve Okon (okong), Sydney Lieng(liengsn),
Niko Savas(savasn), Nick Lago(lagond),
Eric Le Fort(leforte)

March 6, 2016

Contents

1 Introduction **3**
- 1.1 Purpose . . . 3
- 1.2 Scope . . . 3
- 1.3 Definitions, Acronyms & Abbreviations . . . 3
- 1.4 References . . . 3
- 1.5 Overview . . . 3

2 Overall Description **3**
- 2.1 Product Perspective . . . 3
  - 2.1.1 Product Functions . . . 4
  - 2.1.2 User Characteristics . . . 4
  - 2.1.3 Constraints . . . 5
  - 2.1.4 Assumptions and Dependencies . . . 5
  - 2.1.5 Apportioning of Requirements . . . 5

3 Functional Requirements **6**
- 3.1 Business Event: User wants to identify a rock . . . 6
  - 3.1.1 Viewpoint: Everyday User . . . 6
  - 3.1.2 Viewpoint: Developer . . . 6
  - 3.1.3 Viewpoint: Security Expert . . . 6
- 3.2 Business Event: User wants to update the data store . . . 6
  - 3.2.1 Viewpoint: Everyday User . . . 6
  - 3.2.2 Viewpoint: Developer . . . 6
  - 3.2.3 Viewpoint: Security Expert . . . 6
- 3.3 Business Event: User wants to save a rock they've found . . . 6
  - 3.3.1 Viewpoint: Everyday User . . . 6
  - 3.3.2 Viewpoint: Developer . . . 6
  - 3.3.3 Viewpoint: Security Expert . . . 6
- 3.4 Business Event: User wants to identify a rock . . . 6
  - 3.4.1 Viewpoint: Everyday User . . . 6
  - 3.4.2 Viewpoint: Developer . . . 6
  - 3.4.3 Viewpoint: Security Expert . . . 7
- 3.5 Business Event: User wants to learn more about a rock . . . 7
  - 3.5.1 Viewpoint: Everyday User . . . 7
  - 3.5.2 Viewpoint: Developer . . . 7
  - 3.5.3 Viewpoint: Security Expert . . . 7

4 Non-Functional Requirements **7**
- 4.1 Look & Feel Requirements . . . 7
  - 4.1.1 Appearance Requirements . . . 7
  - 4.1.2 Style Requirements . . . 7
- 4.2 Usability & Humanity Requirements . . . 7
  - 4.2.1 Ease of Use Requirements . . . 7
  - 4.2.2 Personalization & Internalization Requirements . . . 7
  - 4.2.3 Learning Requirements . . . 7
  - 4.2.4 Understandability & Politeness Requirements . . . 8
  - 4.2.5 Accessibility Requirements . . . 8
- 4.3 Performance Requirements . . . 8
  - 4.3.1 Speed & Latency Requirements . . . 8
  - 4.3.2 Safety Critical Requirements . . . 8
  - 4.3.3 Precision & Accuracy Requirements . . . 8
  - 4.3.4 Reliability and Availability Requirements . . . 8
  - 4.3.5 Robustness or Fault-Tolerance Requirements . . . 8

4.3.6 Capacity Requirements . . . 8
4.3.7 Scalability or Extensibility Requirements . . . 8
4.3.8 Longevity Requirements . . . 8
4.4 Operational & Environmental Requirements . . . 9
4.4.1 Expected Physical Environment . . . 9
4.4.2 Requirements for Interfacing with Adjacent Systems . . . 9
4.4.3 Productization Requirements . . . 9
4.4.4 Release Requirements . . . 9
4.5 Maintainability & Support Requirements . . . 9
4.5.1 Maintenance Requirements . . . 9
4.5.2 Supportability Requirements . . . 9
4.5.3 Adaptability Requirements . . . 9
4.6 Security Requirements . . . 9
4.6.1 Access Requirements . . . 9
4.6.2 Integrity Requirements . . . 9
4.6.3 Privacy Requirements . . . 9
4.6.4 Audit Requirements . . . 9
4.6.5 Immunity Requirements . . . 10
4.7 Cultural & Political Requirements . . . 10
4.7.1 Cultural Requirements . . . 10
4.7.2 Political Requirements . . . 10
4.8 Legal Requirements . . . 10
4.8.1 Compliance Requirements . . . 10
4.8.2 Standards Requirements . . . 10

A Division of Labour **11**

1 Introduction

1.1 Purpose

The following Software Requirements Specification (SRS) document will provide an overview of the behaviour of the Stone Identification App that will be developed. This document will include a full description of the application and will outline the functional and non-functional requirements that the system needs to fulfill. To gain a thorough understanding of this application, this SRS is geared towards those who are interested in developing and funding this application. It will give them insight to what the software will do and how it will be expected to perform.

1.2 Scope

The Stone Identification App will assist users with the classification of rocks. The system will be able to take in descriptions, such as colour, texture, and size, to recognize the type of rock. When searching through the database, the application will display images of the possible matches, a description, and how much it will cost per gram. In addition, the application will access Google maps API to help narrow the search by location, as well as show others in the area who have also have found similar rocks. Users will also be able to search for rock names and see a description, value and images of the selected rock. The goals of this application is to allow those who are interested in rocks to have a quick access to a database and search through its resources. This application is also used to encourage those who do not know much about rocks to learn more about their qualities.

1.3 Definitions, Acronyms & Abbreviations

Software Requirements Specification (SRS): a complete description of the behaviour of a system to be developed. It includes a set of use cases that describe interaction betweeen the system and the user/environment.
Layman's Terms: describe a complex or technical issue using words and terms that the average individual (someone without professional training in the subject area) can understand, so that they may comprehend the issue to some degree.
Crash: when a software stops functioning properly
Read-only: cannot write or change the file's contents

1.4 References

Andrew LeClair.
SE 3A04: Requirements Templates
Department of Computing and Software, McMaster University, January 27/28, 2016

1.5 Overview

The remaining SRS document will include the overall description, functional requirements, and non-functional requirements. It will provide a description of the general factors that affect the product and its requirements in the Overall Description section. The Functional Requirements section will contain all the software requirements to a level of detail. The last section, Non-Functional Requirements, will outline the how the system is supposed to be, such as the look and feel, usability and humanity, performance, etc.

2 Overall Description

The following section will outline the factors that affect our product and its requirements.

2.1 Product Perspective

The stone identification app will provide similar functions to rock identification sites, presented in an app to be more accessible and portable. The application will also implement some extra features such as value approximation and marketplaces. The application will work by asking multiple questions to refine the list of rocks suggested by the application. The product will make use of google maps and therefore will not be completely self contained. There is no larger system

that will be interacting with our application.

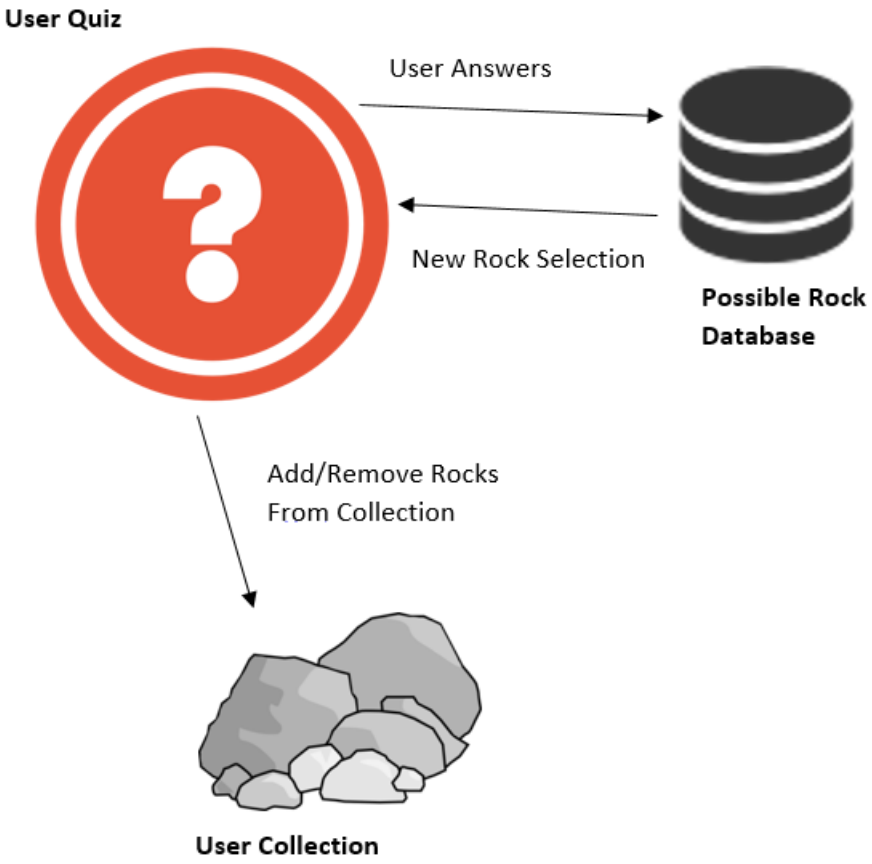
2.1.1 Product Functions

- If the user finds a rock, the application will display a list of all possible rocks for a match
- Each rock possibility is accompanied by a picture, a description and an approximate value
- The application will display a set of questions as a form that the user can answer to refine the selection of possible rocks
- Once a match has been found, the customer has the option to save rock information to their r̈ocks foundl̈ist.
- Customers will be able to remove rocks from their collection

2.1.2 User Characteristics

User restrictions may include:

- No educational restrictions during the identification process, as the use of pictures make this very intuitive
- Some descriptions, and questions about rocks may require a slight knowledge of geology
- Previous experience with phone applications is an asset when using this product
- No technical expertise needed, however users may take some time at first to learn all the features



2.1.3 Constraints

- Possible matches are limited to storage space

2.1.4 Assumptions and Dependencies

- User has enabled GPS function in phone
- Device with application is an android phone

2.1.5 Apportioning of Requirements

Some requirements that may be delayed until future versions:

- Interacting with other users rock collections, such as viewing, and requesting to purchase
- Messaging system between collectors

3 Functional Requirements

3.1 Business Event: User wants to identify a rock

3.1.1 Viewpoint: Everyday User

- The application will allow the user to input information about texture, colour, or size of a rock.
- The application will estimate the type of rock based on user-inputted texture, colour, and/or size.
- The application will suggest rocks based on the user's location.

3.1.2 Viewpoint: Developer

3.1.3 Viewpoint: Security Expert

3.2 Business Event: User wants to update the data store

3.2.1 Viewpoint: Everyday User

- The every day user will not be able to modify the data store.

3.2.2 Viewpoint: Developer

- The developer will be able to update the data store with new information

3.2.3 Viewpoint: Security Expert

- The security expert must be able to manage developer access to the data store

3.3 Business Event: User wants to save a rock they've found

3.3.1 Viewpoint: Everyday User

- The user will be able to add a rock from their matched list to a F̈ound Rocksl̈ist.

3.3.2 Viewpoint: Developer

- The developer will not be able to access an individual user's found rock list.

3.3.3 Viewpoint: Security Expert

3.4 Business Event: User wants to identify a rock

3.4.1 Viewpoint: Everyday User

- The application will be able to recieve user input based on the available experts
- The application will display possible matches based on user input
- The application will allow experts to dictate which rocks from the data store are displayed
- The application will display the image of all rocks matching the user input

3.4.2 Viewpoint: Developer

- The developer will not be able to view user specific searches

3.4.3 Viewpoint: Security Expert

3.5 Business Event: User wants to learn more about a rock

3.5.1 Viewpoint: Everyday User

- The user will be able to view information about the rock including Colour, Size, Texture (Depends upon available experts)
- The user will be able to view an image of the rock
- The user will not be able to modify information about a rock

3.5.2 Viewpoint: Developer

- The developer cannot access user rock searches
- The developer can add and remove experts

3.5.3 Viewpoint: Security Expert

- The security expert must be able to manage developer access to the experts

4 Non-Functional Requirements

The following section will outline the various non-functional requirements that pertain to the Rock Identification app.

4.1 Look & Feel Requirements

These requirements dictate how the final product will appear to the user. The overall goal will be to make the app's look & feel in a way that promotes simple and efficient use by the average user.

4.1.1 Appearance Requirements

Ap1: The program shall use colour schemes appealing to the expected users determined by the designers.

Ap2: The program's appearance shall be designed in a way that it does not distract from usage of the system.

Ap3: The program shall be easy to see even in environments with high background light (i.e. outdoors).

4.1.2 Style Requirements

S1: The program shall uphold a simple design with each page displaying only the necessary information.

4.2 Usability & Humanity Requirements

4.2.1 Ease of Use Requirements

EOF1: The program shall be usable by any user between the ages of 8 and 90 with basic knowledge of how apps function.

4.2.2 Personalization & Internalization Requirements

PI1: The program shall allow the user to choose whether they would like to use location services.

4.2.3 Learning Requirements

Le1: The user shall require training in order to understand usage of basic qualifying characteristics of rocks.

4.2.4 Understandability & Politeness Requirements

UP1: The program shall only assume prior knowledge of the basics of how to use an app and simple defining characteristics of rocks. Functions that require knowledge other than this will not be included.

UP2: The program shall include information in *Layman's Terms* to support the more technical information.

4.2.5 Accessibility Requirements

Ac1: The program shall utilize the largest text possible while maintaining it's aesthetic appeal.

4.3 Performance Requirements

4.3.1 Speed & Latency Requirements

SL1: Time computing possible matches and displaying those results shall never exceed 1 second.

SL2: Time to start the program shall never exceed 3 seconds.

4.3.2 Safety Critical Requirements

N/A

4.3.3 Precision & Accuracy Requirements

PA1: The program shall display the correct rock that matches the descriptors (assuming they are correct) 95% of the time.

PA2: The program shall display the correct rock that matches the descriptors (assuming small mistakes by user being possible) 80% of the time.

4.3.4 Reliability and Availability Requirements

RA1: This system is available on the user's machine and does not rely on other factors to function. This program should be available 99% of the time.

RA2: The program shall *crash* no more than an average of once per hour of runtime.

4.3.5 Robustness or Fault-Tolerance Requirements

RFT1: The program shall be designed in a way that faulty inputs are not possible to select.

4.3.6 Capacity Requirements

No requirements necessary from this section since the program will run independently on each piece of hardware it is running on. So the maximum capacity is only 1.

4.3.7 Scalability or Extensibility Requirements

SE2: The program shall be designed in a way that adding new rocks to the database is possible after deployment without major changes.

4.3.8 Longevity Requirements

Lo1: The product shall be designed in a modular way so as to allow future anticipated change.

4.4 Operational & Environmental Requirements

4.4.1 Expected Physical Environment

EP1: The program will primarily be operated outdoors. The program shall be designed such that non-extreme environments will not affect usability or performance.

4.4.2 Requirements for Interfacing with Adjacent Systems

RIA1: The program shall interface with Android Jelly Bean (4.1.x) and higher.

4.4.3 Productization Requirements

Pro1: The program shall come packaged with all required libraries necessary to readily interface with Android Jelly Bean (4.1.x) and higher.

4.4.4 Release Requirements

R1: The program shall be ready for release of version 1.0 by April 3^{rd}, 2016.

4.5 Maintainability & Support Requirements

4.5.1 Maintenance Requirements

MS1: The product shall store its data in such a way that it will be able to restore to a previous state as necessary.

4.5.2 Supportability Requirements

Su1: The program shall follow a modular design pattern. No more than 10% of modules will be so rigid they can only operate in the exact implementation of the current system.

4.5.3 Adaptability Requirements

Ad1: The program shall be able to integrate new types of rocks into its identification system without affecting any other component's operation.

4.6 Security Requirements

4.6.1 Access Requirements

AR1: Accessing the identification feature will not require credentials, however, accessing the previous rocks discovered or adding new rocks will require the user entering their password.

4.6.2 Integrity Requirements

In1: The rock data stored locally will be encrypted and compared against fixed values as needed to verify there has not been any tampering.

In2: The user's password will be padded and stored encrypted with two different methods of encoding. Both methods of encoding must return the same password to ensure there has not been any tampering.

4.6.3 Privacy Requirements

Pri1: The program will not disclose any information about the user to third-parties without permission.

4.6.4 Audit Requirements

N/A

4.6.5 Immunity Requirements

Im1: The program shall make all files that do not change due to normal program operation *read-only* in the file system.

Im2: The program shall make an announcement to the user if any files hold unexpected data.

4.7 Cultural & Political Requirements

4.7.1 Cultural Requirements

Cu1: This program shall not include any symbols, statements or anything of that nature that may be found offensive to anyone's culture.

4.7.2 Political Requirements

Po1: This program shall not include any symbols, statements or anything of that nature that are offensive in any region in a political sense.

4.8 Legal Requirements

4.8.1 Compliance Requirements

C1: The program shall not retain sensitive data about the user without the user's permission.

4.8.2 Standards Requirements

N/A

A Division of Labour

Name	Contribution	Signature
Genevieve Okon	Functional Requirements	
Eric Le Fort	Non-Functional Requirements	
Nick Lago	Overall Description	
Niko Savas	Functional Requirements	
Sydney Lieng	Introduction	