University of Pretoria

COS 301 - Software Engineering

The Savage Ru's

Software Requirements Specification and Technology Neutral Process Design

Author(s):	*Student number(s):*
Jodan Alberts	14395283
Mark Klingenberg	14020272
Una Rambani	14004489
Ruan Klinkert	14022282

May 24, 2016

Contents

1 Introduction **2**

2 Vision **3**

3 Background **4**

4 Architecture Requirements **5**
4.1 Architectural Scope 5
4.2 Access Channel Requirements 5
4.3 Quality Requirements 5
4.3.1 Performance 5
4.3.2 Reliability 5
4.3.3 Scalability 5
4.3.4 Usability 5
4.3.5 Auditability 5
4.3.6 Security 5
4.4 Integration Requirements 5
4.5 Architecture Constraints 5
4.6 Use case prioritization 5
4.6.1 Critical 5
4.6.2 Important 5
4.6.3 Nice to Have 6
4.7 Use case/Services contracts 6
4.8 Required functionality 6
4.9 Process specifications 6
4.10 Domain Model 6

5 Software Architecture **7**
5.1 Architectural Patterns or Styles 7
5.2 Architectural Tactics or Strategies 7
5.3 Use of Reference Architectures and Frameworks 7
5.3.1 Web 2.0 Reference Architecture 7
5.4 Access and Integration Channels 7
5.5 Technologies 7

6 Open Issues **8**

1 Introduction

This is the software requirements specification for the vizARD Augmented Reality application being developed for EPI-USE Labs by The Savage Ru's.

2 Vision

EPI-USE Labs (henceforth referred to as "the client") intends for the VizARD application to be used by a large variety of mobile device users across both Android and iOS platforms. VizARD helps to simplify the analysis of numerical data through visualization, in the form of automatically generated 3D graphs.

Fundamentally, the system will allow a user to take a picture of a table of numerical data which he/she may need to interpret. The application will then use OCR (Optical Character Recognition) to read the data from the picture. It will then decide on an appropriate graph for the type of data and generate a graph for the data. After the graph is generated, it will project a 3D model of the graph onto the image (or, ideally, onto a live stream of the paper) for the user to view.

Additionally, the system will allow users to send images (or screen captures) of generated graphs to other devices via popular social media channels.

Typically usage will be as follows:

- The user (possibly a businessman) finds tabular data he/she would like to analyse more easily.
- The user opens the app.
- Once the app is open and loaded, the user takes a picture of the table he/she would like to analyse.
- The user receives a notification that the graph has been generated and the generated graph is displayed on the screen (mapped onto the paper).
- The user taps on the "Share" button and is presented with several options through which he/she can share the graph.
- An option is selected and an image of the graph is sent to the other user.

3 Background

4 Architecture Requirements

4.1 Architectural Scope

4.2 Access Channel Requirements

4.3 Quality Requirements

4.3.1 Performance

4.3.2 Reliability

4.3.3 Scalability

4.3.4 Usability

4.3.5 Auditability

4.3.6 Security

4.4 Integration Requirements

4.5 Architecture Constraints

4.6 Use case prioritization

4.6.1 Critical

- Taking a picture
- OCR (Optical Character Recognition)
- Automatic Graph Suggestion Algorithm
- Graph Generation
- Mapping Graph to Page

4.6.2 Important

- Live Augmented Reality Mapping
- Editing Graphs
- iOS Application
- Social Media Sharing

4.6.3 Nice to Have

- Opening Previous Graph

4.7 Use case/Services contracts

4.8 Required functionality

4.9 Process specifications

4.10 Domain Model

5 Software Architecture

5.1 Architectural Patterns or Styles

5.2 Architectural Tactics or Strategies

5.3 Use of Reference Architectures and Frameworks

5.3.1 Web 2.0 Reference Architecture

5.4 Access and Integration Channels

5.5 Technologies

6 Open Issues