The Mathinator

The Mathinator
Software Requirements Specification
For The Android Application

Version 1.2

Revision History

Date	Version	Description	Author
23/10/16	1.0	Initial Description	Hug, Lamm, Saupp
1/11/16	1.1	Use Cases added	Hug, Lamm, Saupp
26/11/16	1.2	Redefined Scope, Added UCs	Hug, Lamm, Saupp
31/6/17	1.3	Update SRS	Hug,

Table of Contents

Software Requirements Specification

1. Introduction

This document describes the Software Requirements Specifications (SRS) for the Application “The Mathinator”.

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the application “The Mathinator”. It will cover the features in full detail. Furthermore important characteristics of this project will be specified.
This includes design and architectural decisions.

1.2 Scope

“The Mathinator” is an Android application designed to learn your handwriting via character recognition based on an Artificial Intelligence (AI) and providing the user with the solution.

1.3 Definitions, Acronyms, and Abbreviations

- **AI** - Artificial intelligence
- **Android** - A mobile operating system used primarily for smartphones and tablets

1.4 References

Document	Where to find?
Blog	https://mathinator.tobiaslamm.de
Github	https://github.com/SaschaHug/Mathinator
Use Case 1 “take a picture”	https://github.com/SaschaHug/Mathinator/blob/master/Use%20Cases/1_UC_Mathinator_Take_A_Picture.pdf
Use Case 2 “view history”	https://github.com/SaschaHug/Mathinator/blob/master/Use%20Cases/2_UC_Mathinator_View_History.pdf
Use Case 3 “show tour on first start”	https://github.com/SaschaHug/Mathinator/blob/master/Use%20Cases/3_UC_Mathinator_Delete_Entry.pdf
Use Case 4 “enable user to delete entries”	https://github.com/SaschaHug/Mathinator/blob/master/Use%20Cases/4_UC_Mathinator_Use_Manual_Calculator.pdf
Use Case 5 “do manual calculations”	https://github.com/SaschaHug/Mathinator/blob/master/Use%20Cases/5_UC_Mathinator_Show_tour.pdf
Use Case 6 “accept Picture”	https://github.com/SaschaHug/Mathinator/blob/master/Use%20Cases/6_UC_Mathinator_Accept_Picure.pdf

1.5 Overview

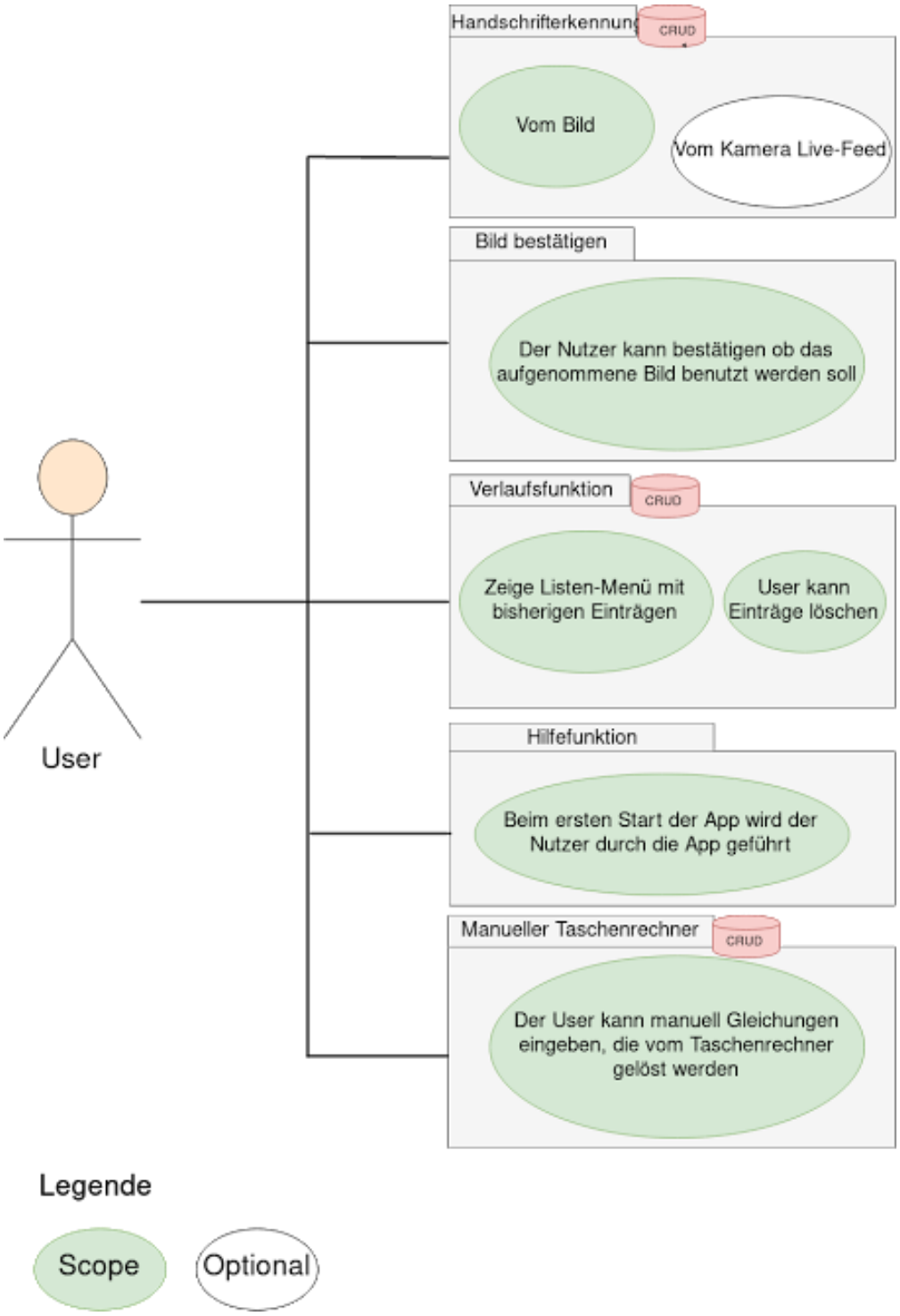
The rest of the document is separated into 3 different chapters.
Chapter 2 will cover our vision and Chapter 3 will cover the specific requirements needed to get there. Chapter 4 will provide additional Information.

2. Overall Description

2.1 Vision

Studying mathematics can be a frustrating endeavour at times. Our application aims to aid the user in these troubled situations by providing solutions to certain problems, so the user can check whether they correctly solved the given equation. People using our app can take pictures of equations and are provided with a solution.

2.2 Use Case Diagram



3. Specific Requirements

3.1 Functionality - Android App

3.1.1 Take a Picture

The User is able to take a picture with the smartphone camera.

3.1.2 Solve Math Equation

The app provides the solution of the the mathematical equation on the screen after a picture was taken.

3.1.3 Go through History

The user is able to go through previously taken pictures and review the results.

3.1.4 Delete Entries

The user is able to delete previous entries by long-clicking on them from the history menu.

3.1.5 Show Tour on First Startup

The first time the app is started, the user is being given tips regarding the functions of each menu.

3.2 Usability

We assume that the user is capable of installing an app via the official Play Store provided by Google. We will additionally provide an installation guide.

The app itself is self-explanatory, though the user will be guided by popups which explain the apps functions the first time the app is used.

3.3 Reliability

To be determined.

3.4 Performance

To be determined.

3.5 Supportability

3.5.1 Languages and platforms

We will use the following languages and platforms, which will also be supported in future versions:

- Java EE 8
- Android Version 6.0 (Marshmallow)

3.6 Design Constraints

All information about the architectural design of our application can be found in our software architecture document (Yet to be done).
In the following chapter you can read about general decisions.

3.6.1 Backend in Java

The Backend of this application is written in Java. It consists of 2 elementary parts: The AI that recognizes input and the calculation unit that processes the aforementioned input.

3.7 Online User Documentation and Help System Requirements

The app itself is designed to be intuitive. Additional help prompts are implemented to guide the user on first startup. Should there still be questions about the use of the app, users can contact us on our blog.

3.8 Purchased Components

None

3.9 Interfaces

3.9.1 User Interfaces

To Be Determined

3.9.2 Hardware Interfaces

N/A

3.9.3 Software Interfaces

To Be Determined

3.9.4 Communications Interfaces

N/A

3.10 Licensing Requirements

To be Determined

3.11 Legal, Copyright, and Other Notices

To be Determined

3.12 Applicable Standards

To be Determined

4. Supporting Information

Visit our github for more information (https://github.com/SaschaHug/Mathinator) and our

The Mathinator

blog for the current status of the project (https://mathinator.tobiaslamm.de).