
Cheetah

Cheetah
Software Requirements Specification

Version 1.2

Revision History

Date	Version	Description	Author
26/10/2016	1.0	Initial Version	Damian Pistorius, Lior Olszewski, Sara Olszewski
26/10/2016	1.1	UML	Sara Olszewski
01/11/2016	1.2	Use Case	Sara Olszewski

Table of Contents

1. Introduction.....	4
1.1 Purpose.....	4
1.2 Scope.....	4
1.3 Definitions, Acronyms, and Abbreviations.....	4
1.4 References.....	4
1.5 Overview.....	4
2. Overall Description.....	5
3. Specific Requirements.....	6
3.1 Functionality.....	6
3.1.1 Use-Structure.....	6
3.1.2 Communications.....	6
3.1.3 Create/Upload Data.....	6
3.1.4 Search Data.....	6
3.2 Usability.....	7
3.2.1 Training Time.....	7
3.2.2 Hardware Requirements.....	7
3.2.3 Software Requirements.....	7
3.3 Reliability.....	7
3.3.1 Availability.....	7
3.3.2 Bug or Defect Rate.....	7
3.4 Performance.....	7
3.4.1 Response Time.....	7
3.4.2 Throughput.....	7
3.4.3 Capacity.....	7
3.4.4 Resource utilization.....	7
3.5 Supportability.....	7
3.6 Design Constraints.....	7
3.6.1 Software Language.....	7
3.7 On-line User Documentation and Help System Requirements.....	7
3.8 Purchased Components.....	7
3.9 Interfaces.....	8
3.9.1 User Interfaces.....	8
3.9.2 Hardware Interfaces.....	8
3.9.3 Software Interfaces.....	8
3.9.4 Communications Interfaces.....	8
3.10 Licensing Requirements.....	8
3.11 Legal, Copyright, and Other Notices.....	8
3.12 Applicable Standards.....	8
4. Supporting Information.....	8

Software Requirements Specification

1. Introduction

Project Cheetah will help students to learn from earlier test that were uploaded with the memories of the students.

1.1 Purpose

The purpose of this document is to present a more detailed description of the Cheetah project.

1.2 Scope

This SRS applies to the entire Cheetah project. This project will be realized as an Web-Application. The overview of the features and subsystems are documented in the Use-Case model shown in Overall Description.

1.3 Definitions, Acronyms, and Abbreviations

SRS	Software Requirements Specification

1.4 References

- [GitHub](#)
- [Blog](#)
- [Use Case - Create Text](#)
- [Use Case - Add Institution](#)

1.5 Overview

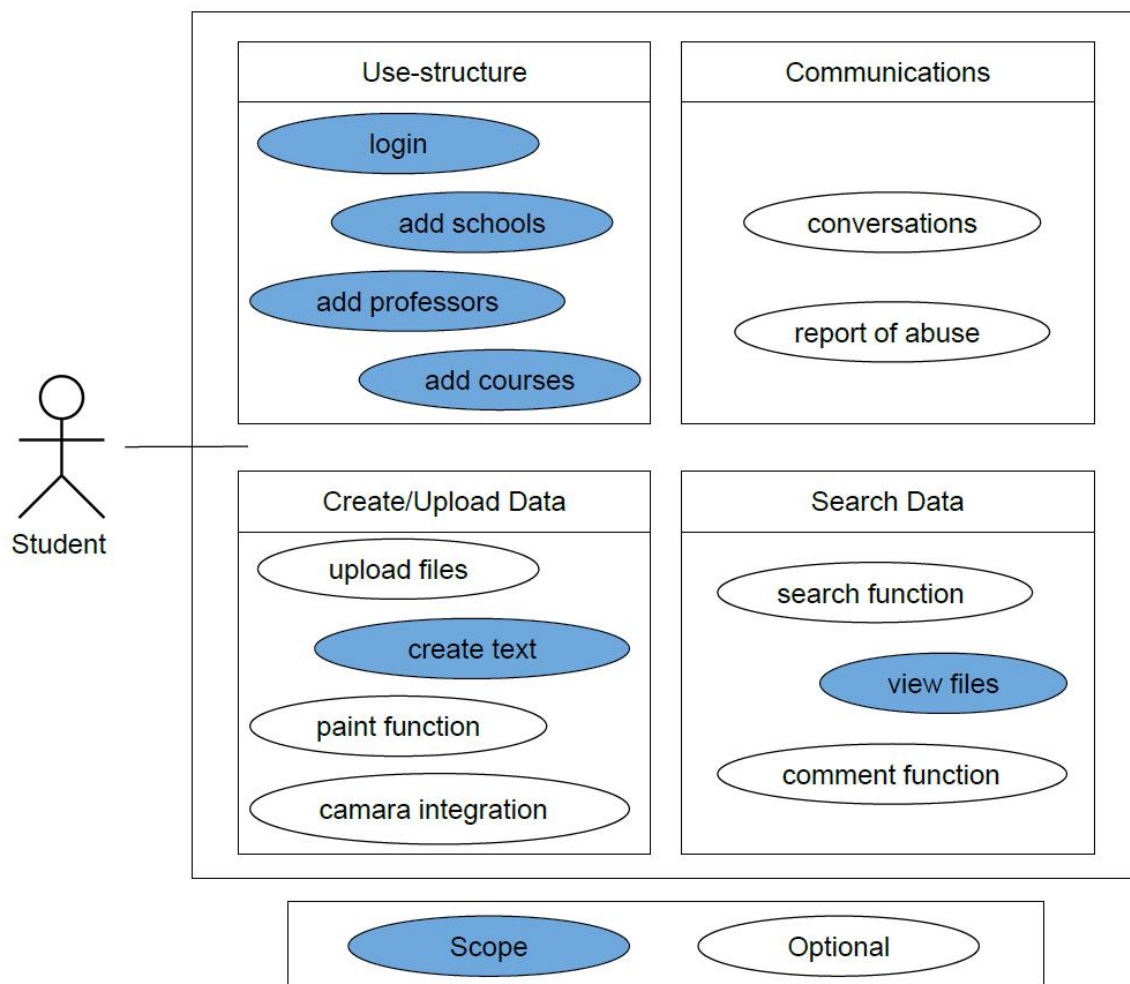
At most of the Universities there are already existing pools of earlier tests. Just at the DHBW not. Cheetah, created from “cheater”, will give the opportunity to the students of DHBW to upload test from their memories and also look on others to get prepared for their tests.

The following chapters of this document will help you to understand the requirements and specifications of Project Cheetah.

2. Overall Description

Cheetah will be a web portal, where users can see earlier tests that were written in the years before. Students can upload or create the tests from their memories. These will be categorized that the searcher will find them easier.

The Server Backend application will be written in Ruby using the Ruby on Rails Framework.



3. Specific Requirements

3.1 Functionality

3.1.1 Use-Structure

Login

For the first login the student needs to register by providing an username, password and email address and confirm this.

After that he will sign in with the username and password.

Add Schools

The student should be able to add his school, if it isn't existing yet. This will be used to categorize the tests.

Add Professors

The student should be able to add professors, if they aren't existing yet. This will be used to categorize the tests.

Add courses

The student should be able to add the specific course, if it isn't existing yet. This will be used to categorize the tests.

3.1.2 Communications

Conversations

The students should be able to connect over the system, to talk to older students about the tests, without knowing them.

Report of abuse

If anybody sees something wrong there should be the possibility to report it, from where the admin is going to correct it.

3.1.3 Create/Upload Data

Upload Files

The user should be able to upload files (pdf/jpg).

Create Text

The user should be able to create a text in the application.

Paint function

To paint (circuit diagrams) fast from the memories of the test, the student should be able to draw in the application over a browser on the cellphone.

Camera Integration

The application through a cellphone browser should include a camera, to make photos from the test.

3.1.4 Search Data

Search Function

The Search Function should include search in the categories or a text search.

View Files

The files should be shown in the correct format, so the students can read everything.

Comment Function

Students could write comments on other students entries, to complete or correct something.

3.2 Usability

3.2.1 Training Time

The application will be easy to learn. With minimal designs and labels, it will be easy for the students to work with the application.

3.2.2 Hardware Requirements

The student needs a computer + peripherals (mouse, keyboard) or a cellphone. An internet connection is required to search and upload tests.

3.2.3 Software Requirements

The computer or the cellphone needs a working internet browser to open the web application.

3.3 Reliability

3.3.1 Availability

The web application will be always available.

3.3.2 Bug or Defect Rate

All functions should be available without any bugs.

3.4 Performance

3.4.1 Response Time

(n/a)

3.4.2 Throughput

(n/a)

3.4.3 Capacity

(n/a)

3.4.4 Resource utilization

(n/a)

3.5 Supportability

(n/a)

3.6 Design Constraints

3.6.1 Software Language

Ruby with the Ruby on Rails Framework

3.7 On-line User Documentation and Help System Requirements

The manual has to be read while signing up.

3.8 Purchased Components

(n/a)

3.9 Interfaces

3.9.1 User Interfaces

(n/a)

3.9.2 Hardware Interfaces

(n/a)

3.9.3 Software Interfaces

(n/a)

3.9.4 Communications Interfaces

(n/a)

3.10 Licensing Requirements

(n/a)

3.11 Legal, Copyright, and Other Notices

We are developing this application open source. The code will be accessible via GitHub using:
<https://github.com/Damian1234523/Cheetah>

3.12 Applicable Standards

(n/a)

4. Supporting Information

(n/a)