A close-up of a logo

Description automatically generated**Usability Testing and Verification**

Lukas Klöck - 11014902

Assignment 1: **Test Plan**

**Table of Contents**

[1 Introduction 2](#_Toc159855337)

[2 Objectives and Tasks 2](#_Toc159855338)

[3 Scope 2](#_Toc159855339)

[4 Testing Strategy 2](#_Toc159855340)

[5 Hardware Requirements 2](#_Toc159855341)

[6 Environment Requirements 2](#_Toc159855342)

[7 Test Schedule 2](#_Toc159855343)

[8 Control Procedures 2](#_Toc159855344)

[9 Features to be Tested 2](#_Toc159855345)

[10 Features not to be Tested 2](#_Toc159855346)

[11 Resources/Roles & Responsibilities 2](#_Toc159855347)

[12 Schedules 2](#_Toc159855348)

[13 Significantly Impacted Departments (SIDs) 2](#_Toc159855349)

[14 Dependencies 2](#_Toc159855350)

[15 Risks/Assumptions 3](#_Toc159855351)

[16 Tools 3](#_Toc159855352)

[17 Approvals 3](#_Toc159855353)

# Introduction

In the following report, the procedure of testing of the function **‘Array Shift’** will be described. The tests are limited to the different specifications which follow the objective of extensively checking its functions regarding *suitability, usability, and reliability* as well as the general absence of errors and flaws. The overall functionality is tested with the goal of getting a correct return value which depends on different varying numerical input criteria.

# Objectives and Tasks

The main objective of the tests should be

# Scope

The above mentioned function is required to fulfil three different practices. For the input, there must be two different input criteria, firstly a numerical array and secondly a specific value which is expected to be contained within the array. In case the function works as desired three different outputs should be provided, depending on the given value to be analyzed:

1. If the value is contained in the array, the output should be the next sequential value of the array
2. If the value is not contained in the array, the output should be the first element of the array.
3. If the value is the last element of the array, the output should be -1.

# Testing Strategy

## Unit Testing

**Definition:** To test the desired functionalities, a simple unit test will be implemented that is capable of testing for correctness of all three required in- and output cases.

**Participants:** The developer himself is responsible for the testing.

**Methodology:** The developer is responsible for providing the required testing script that sufficiently fulfils the above-mentioned functionalities.

# Hardware Requirements

The only hardware related requirement that must be fulfilled, so that the function can be tested, is, that the hardware should be able to run any preferred IDE for the required JavaScript environment, as further described in Chapter 6.

# Environment Requirements

To conduct the tests a runtime environment is required, that does in include an IDE that can run JavaScript and a needed testing framework, for example “Jest”.

# Test Schedule

The function is to be tested directly after the implementation of

# Control Procedures

# Features to be Tested

# Features not to be Tested

# Resources/Roles & Responsibilities

# Schedules

# Significantly Impacted Departments (SIDs)

# Dependencies

# Risks/Assumptions

# Tools

# Approvals

const assert = require("assert");

function arrayShift(array, value) {

let index = array.findIndex((element) => element === value);

if (index < 0) return array[0];

if (index == array.length -1) return -1;

return array[index + 1]; };

//const testarr = [1,2, 3, 4, 5]

//const testval = 1;

//console.log(arrayShift(testarr, testval));

assert.equal(arrayShift(data, 3), 7);

assert.equal(arrayShift(data, 5), 2);

assert.equal(arrayShift(data, 5), -1);

console.log("Finished");

Listing 1: Array Shift Function Source code