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](#_Toc161081122)

[2 Objectives and Tasks 2](#_Toc161081123)

[3 Scope 2](#_Toc161081124)

[3.1 General 2](#_Toc161081125)

[3.2 Tactics 2](#_Toc161081126)

[4 Testing Strategy 3](#_Toc161081127)

[4.1 Unit Testing 3](#_Toc161081128)

[5 Hardware Requirements 3](#_Toc161081129)

[6 Environment Requirements 3](#_Toc161081130)

[7 Test Schedule 3](#_Toc161081131)

[8 Control Procedures 3](#_Toc161081132)

[9 Features to be Tested 4](#_Toc161081133)

[10 Features not to be Tested 4](#_Toc161081134)

[11 Resources/Roles & Responsibilities 4](#_Toc161081135)

[12 Schedules 4](#_Toc161081136)

[13 Significantly Impacted Departments (SIDs) 4](#_Toc161081137)

[14 Dependencies 4](#_Toc161081138)

[15 Risks/Assumptions 4](#_Toc161081139)

[16 Tools 5](#_Toc161081140)

[17 Approvals 5](#_Toc161081141)

Appendix

# 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 cases of a numerical input.

# Objectives and Tasks

The main objectives which are set out by this test plan are limited to the few tasks, which include the description of the test sequence, the explanation of required hard- and software tools, as well as all responsible and required personal who are involved in fulfilment of the entire creation and testing process.

# Scope

## General

The above mentioned function is required to fulfil three different practices. For the input, two different inputs will be taken, 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 within the array, the output should be the next sequential value of such.
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.

## Tactics

For the testing tactics a simple unit test must be provided, that covers the entire functionality of the array shift function and is able to verify all the above mentioned requirements. Due to this simplicity, a single developer will be assigned to execute the whole creation and testing process.

# Testing Strategy

## Unit Testing

**Definition:** To test the desired functionalities, a singular unit test will be implemented that is capable of testing for correctness of all three output cases, depending on the different inputs.

**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/Testing 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 any arbitrary testing framework in which those tests can be create to test the implementation, for example “Jest”.

# Test Schedule

The function is to be tested directly after the implementation of the actual Array-Shift function. The test procedure should only take a small amount of time to complete, therefore no further detailed milestones are required for the process.

# Control Procedures

Problem reporting or changes are solely conducted by the responsible developer which implements and tests the code. No further documentation or reports need to be created or provided due to the profound simplicity of the task.

# Features to be Tested

To be tested is the correct output of specific numerical values which must be found, with the help of the function, within an array and output depending on the position and condition of this input value:

1. If the number is contained, the next sequential digit of the array should be returned.
2. If it is not present, the first element should be returned.
3. If the number is at the last position of the array, the value -1 should be returned.

# Features not to be Tested

Apart from the in Chapter 9 mentioned functions that are to be teste, no further test requirements have been set forth, which means that once these cases (Chapter 9) are completed the process is considered to be finished.

# Resources/Roles & Responsibilities

All members involved in the process are limited to one single developer who should have the expertise to conduct all the necessary steps. This includes implementation, resolving of issues and of course have knowledge about the operation of the required hard and software.

# Schedules

Except this required test plan documents, no further reports or other documents are to be scheduled for the implementation and testing.

# Significantly Impacted Departments (SIDs)

No SIDs are dependent on the testing process.

# Dependencies

The process is not dependent on additional specific dependencies.

# Risks/Assumptions

Due to the simplicity of the function itself as well as the here mentioned associated testing task, it is estimated that the entire scope of work can be managed by a single person in a very short period of working time, therefore there are no further noteworthy coherent risks.

# Tools

The tools which are required to develop the necessary code and testing scripts, as well as conduct the tests are limited to the hard- and software tools already mentioned in the Chapters 5 and 6.

# Approvals

For the approval of the testing plan, a revision by the developer, which has the mandatory level of expertise, is required which is followed by a subsequent final approval by his own efforts.

Appendix

Reference code provided from external repository:

<https://github.com/UTV-2024-02/02_testing> (Commit 3abf8dd)

A screenshot of a computer code

Description automatically generated

Figure 1: File someFunction\_data.js

A screenshot of a computer program

Description automatically generated

Figure 2: File someFunction.test.js

A screenshot of a computer program

Description automatically generated

Figure 3: File someFunction.js

A screenshot of a computer code

Description automatically generated

Figure 4: File someFunction\_solved.js