Unifying SDK SRS

Software Requirements Specification

V1.0

7 March 2017

Yossi Cohn

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Title** | **Comments** |
| 2017-03-07 | Initial | Yossi Cohn | SW Team Leader | First Revision |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  |  | SW Team Leader |  |
|  |  | VP R&D |  |
|  |  | Product Manager / Head of Products |  |

Table of Contents

*Revision History* 2

*Document Approval* 2

1.1 Introduction 5

1.1.1 Purpose 5

1.1.2 Definitions, Acronyms, and Abbreviations 5

1.1.3 References 5

1.1.4 Feature to SRS Functionality 5

1.2 General Description 6

1.2.1 Product Perspective 6

1.2.2 Product Functions 6

1.2.3 Constraints 6

1.2.3.1 SDK Performance 6

1.2.3.2 SDK Functionality 6

1.2.4 Assumptions and Dependencies 6

1.2.4.1 The SDK will include the functionality of the Real-Time and the Optitrack 6

1.3 Specific Requirements 6

1.3.1 External Interface Requirements 6

1.3.1.1 User Interfaces 6

1.3.1.2 Software Interfaces 6

1.3.1.3 Communications Interfaces 6

1.3.2 Functional Requirements 7

1.3.2.1 Easy SDK Versions Deployment and Upgrade 7

1.3.2.2 SDK Logging Mode for Debug 7

1.3.2.3 Support Optitrack Functionality 7

1.3.2.4 Support Real-Time Functionality 8

1.3.2.5 Unify Calls to Real-Time and Tracked events 8

1.3.2.6 Support SDK usage by the GTM 9

1.3.3 Use Cases 11

1.3.3.1 Use Case #1 11

1.3.3.2 UC #2 11

1.3.4 Non-Functional Requirements 11

1.3.4.1 Performance 11

1.3.4.2 Reliability 11

1.3.4.3 Accessibility 11

1.3.4.4 Inverse Requirements 11

1.3.4.5 Design Constraints 12

1.3.4.6 Logical Database Requirements 12

1.3.4.7 State Saving Requirements 12

1.3.4.8 Other Requirements 12

1.4 Analysis Models 12

1.4.1 Sequence Diagrams 12

1.4.2 Data Flow Diagrams (DFD) 12

1.4.3 State-Transition Diagrams (STD) 12

1.5 Appendixes 12

1.5.1 Appendix 1 12

1.5.2 Appendix 2 12

# Introduction

Optimove deploy and provide several SDK ‘s to Clients side.

To make some order and provide a single SDK to the User/Client we would like to Unify these SDK’s

## Purpose

Here we will write the details of requirements for the DSK Unifications*.*

We will write the Specific functionality of the SDK and how it is implementing the different functionalities of today’s existing SDK’s.

The Steak holders are the Customers and the SDK Providing parties.

## Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Name | Definition |
| SDK | Software Development Kit |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## References

*Documents that this* document *references.*

*Must include the general SRS document*

|  |  |
| --- | --- |
| Reference | Detail |
| *<XXXX>* General SRS.docx | The general application SRS. |
|  |  |
|  |  |

## Feature to SRS Functionality

*Here we define the connection R&D feature/Task to the* Source Document *(*Feature Source + ID by the CM System (TFS etc.)*)*

|  |  |
| --- | --- |
| Feature | Source Document |
| The Feature SRS title + Sys ID. | The SRS Definition. |
|  |  |
|  |  |

# General Description

## Product Perspective

*.*

## Product Functions

## Constraints

### SDK Performance

The SDK Performance will not change as consequence of the new SDK unification.

### SDK Functionality

The SDK Functionality will not be harmed or changed.

## Assumptions and Dependencies

### The SDK will include the functionality of the Real-Time and the Optitrack

*Assumption details*

# Specific Requirements

## External Interface Requirements

### User Interfaces

*Describe all UI requirements.*

### Software Interfaces

*Describe all software interfaces including 3rd party (external libraries), OS, etc.*

* Piwik
* Real-time

### Communications Interfaces

*Describe any needed communication interface (WR, multicast, video, other).*

*If there is no communication interface – write NONE.*

## Functional Requirements

*Functional requirements are all the actual feature requirements. Make sure to describe everything*

### Easy SDK Versions Deployment and Upgrade

* Feature Source + ID by the CM System (TFS etc.) as mapped by the table at section 1.1.4.
* Priority: Essential
* Effort: Days
* Risk: Safe

##### Introduction

*We would like to deploy and upgrade the SDK Versions in a fast and easy manner.*

*The Upgrade should be as transparent as possible to the client.*

##### Inputs

*SDK Versions sources*

##### Processing

*The exact processing flow*

1. *Upload the SDK Version to Client*

##### Outputs

*SDK Version is upgraded*

### SDK Logging Mode for Debug

* Feature Source + ID by the CM System (TFS etc.) as mapped by the table at section 1.1.4.
* Priority: Essential
* Effort: Days
* Risk: Safe

##### Introduction

Support debug logs for SDK debugging needs.

We would like to have a function which will be used to turn Off/On this debugging capability.

##### Inputs

*Function setLogMode(mode = [0/1])*

##### Outputs

Shoe Logs

### Support Optitrack Functionality

* Feature Source + ID by the CM System (TFS etc.) as mapped by the table at section 1.1.4.
* Priority: Essential
* Effort: Days
* Risk: 1-risk

##### Introduction

The SDK Should support the current existing Optitrack Functionality, using the Piwik 3rd Party.

This Includes:

1. Log Sessions.
2. Log Routing Events.
3. Log Tracking Event (Piwik trackevent functionality).
4. Support setUserId.
5. Support setUserEmail.
6. Support setCustomDimension.
7. Support Visitor Life Cycle Needs i.e. record Session Custom Dimension for Original VisitorID
8. Support the SPA site with the Piwik trackPagView solution.

##### Inputs

##### Processing

##### Outputs

##### Status Validation

### Support Real-Time Functionality

* Feature Source + ID by the CM System (TFS etc.) as mapped by the table at section 1.1.4.
* Priority: Essential
* Effort: Days
* Risk: 1-risk

##### Introduction

The SDK Should support the current existing Real-Time Functionality, Log Sessions.

1. Log Real-Time Events.

##### Inputs

##### Processing

##### Outputs

##### Status Validation

### Unify Calls to Real-Time and Tracked events

* Feature Source + ID by the CM System (TFS etc.) as mapped by the table at section 1.1.4.
* Priority: Essential
* Effort: Days
* Risk: 1-risk

##### Introduction

We would like the user to call a single function instead of several with the appropriate parameters in order to execute a message to the Piwik server and to the Real-Time Server

##### Inputs

##### Processing

##### Outputs

##### Status Validation

### Support SDK usage by the GTM

* Feature Source + ID by the CM System (TFS etc.) as mapped by the table at section 1.1.4.
* Priority: Essential
* Effort: Days
* Risk: 1-risk

##### Introduction

We would like the user to be able to use the GTM in order to implement the different logic for the event calls.

##### Inputs

##### Processing

##### Outputs

##### Status Validation

## Use Cases

### Use Case #1

*Fully describe the UC flow*

*Including any notes, comments and insights you might have*

*The UC’s should include only the UC requirements (not the implementation!)*

*You are encouraged to use the following structure:*

1. Summary: <UC summary 1-2 sentances>
2. Priority: < Essential | Expected | Desired | Optional >
3. Use frequency: < Always | Often | Sometimes | Rarely | Once >
4. Prerequisits: <if any>
   1. Cond
   2. Cond
   3. cond
5. Main success scenario:
   1. <Describe here>
   2. <the success flow>
   3. <step by step>
6. Alternative Scenarios:
   1. If CONDITION, then ALTERNATIVE STEPS.
      1. NOTES or DETAILS.
   2. If CONDITION, then ALTERNATIVE STEPS.
      1. NOTES or DETAILS.
7. Notes:
   1. Note
   2. Note
   3. Open question
   4. Open question

### UC #2

## Non-Functional Requirements

### Performance

*Write any performance requirements and or implications*

### Reliability

*Write the reliability requirement(s)*

### Accessibility

*Write the accessibility priority of the feature, as seen ‘in the eyes’ of the end user (ie – is this feature highly accessible? Or maybe it can be hidden in a 3rd level menu?)*

### Inverse Requirements

Write any inverse requirements – ie what the software SHOULD NOT do

### Design Constraints

Describe any requirement for design constraint.

### Logical Database Requirements

State all the data entities this feature need to store in persistent storage (Cassandra)

### State Saving Requirements

State all the things that needs to be saved in the application state, in order to return to the same state in case of application crash.

### Other Requirements

Catchall section for any additional requirements.

# Analysis Models

List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS’s requirements.

## Sequence Diagrams

## Data Flow Diagrams (DFD)

## State-Transition Diagrams (STD)

# Appendixes

Appendixes may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS’s overall set of requirements.

*Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.*

## Appendix 1

## Appendix 2