Abstract

This document is the Software Transfer Document (STD) for the SensUs Digital Platform. The SensUs Digital Platform is an online video streaming and data presentation service for the annual SensUs event - an international student competition on molecular biosensors for healthcare applications. With this document, SensUs will have a documented procedure to set up the SensUs Digital Platform. This document complies with the ESA software standard and is organized based on the IEEE table of contents.
# Contents

1 Introduction 6
   1.1 Purpose .................................................. 6
   1.2 Scope ..................................................... 6
   1.3 Definitions and Abbreviations .................. 7
      1.3.1 Definitions ......................................... 7
      1.3.2 Abbreviations ....................................... 8
   1.4 List of References ................................. 8
   1.5 Overview ............................................... 9

2 Build Procedure 10
   2.1 Production ............................................. 10
   2.2 Development .......................................... 10

3 Installation Procedure 11
   3.1 Deployment Server ................................. 11
      3.1.1 Database ........................................... 13
      3.1.2 Redis ................................................ 13
      3.1.3 Laravel Echo Server .......................... 13
   3.2 Development .......................................... 14

4 Configuration Item List 15
   4.1 Documentation ....................................... 15
   4.2 Test Plans ............................................. 15
   4.3 Software ............................................... 15

5 Acceptance Test Report 16

6 Software Problem Report 17
   6.1 Video Statistics ..................................... 17
   6.2 Real time updates .................................... 17
   6.3 Concurrent Administration ....................... 17
   6.4 Event Dashboard ...................................... 17

7 Software Change Requests 18

8 Software Modification Report 19
DOCUMENT STATUS SHEET

GENERAL

Document title: Software Transfer Document v1.0.0
Identification: STD/1.0.0
Authors: J.M.A. Boender, B.A.M. van Geffen
Document status: Final 1.0.0

DOCUMENT HISTORY

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author(s)</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1.0</td>
<td>20-06-2017</td>
<td>J.M.A. Boender</td>
<td>Document layout.</td>
</tr>
<tr>
<td>0.2.0</td>
<td>20-06-2017</td>
<td>J.M.A. Boender</td>
<td>All sections except 2 and 3 added.</td>
</tr>
<tr>
<td>0.3.0</td>
<td>20-06-2017</td>
<td>B.A.M. van Geffen</td>
<td>Sections 2 and 3 added.</td>
</tr>
<tr>
<td>0.4.0</td>
<td>20-06-2017</td>
<td>B.A.M. van Geffen</td>
<td>All sections revised.</td>
</tr>
<tr>
<td>1.0.0</td>
<td>20-06-2017</td>
<td>B.A.M. van Geffen</td>
<td>Feedback incorporated.</td>
</tr>
</tbody>
</table>
## DOCUMENT CHANGE RECORDS

<table>
<thead>
<tr>
<th>Section</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.0</td>
<td>Document created.</td>
</tr>
</tbody>
</table>
1 INTRODUCTION

1.1 PURPOSE

The Software Transfer Document will articulate exactly what steps are necessary to install and configure the SensUs Digital Platform. This will help the customer (or for any other party interested) to independently set up of the system. Furthermore, it describes what remaining software problems (or enhancements) have been reported.

1.2 SCOPE

Valedictorian is a group of computer science students working on behalf of the TU/e for SensUs, and is the group that made the SensUs Digital Platform.

The SensUs Digital Platform will provide online viewers with an experience as close as possible to actually attending the event. This will be realized by having live video streams and highlights of past live video streams of events available, along with additional information about the competition on the website. These streaming-related features would also be available on the mobile application. In this manner, online viewers will have access to the same information and ongoings of the events as attendees. Furthermore, for any attendee of the event, the website and mobile application will provide an additional source of information of ongoing activities, the participating teams, the different awards and the judges. Since it will be impossible to be everywhere at once, this will also allow attendees to catch up with any interesting missed activities. Finally, the website and mobile application will act as a medium to allow online viewers (and offer attendees an additional format) to engage with the participating teams. This engagement will be a mixture of asking the teams interesting questions, sharing videos of the event and voting for the teams on the different awards. Overall, the SensUs Digital Platform will provide another dimension of engagement for everyone interested in following the competition.

Originally, the SensUs Digital Platform was intended to consist both of an SensUs Digital mobile application and a SensUs Digital website. Due to time constraints of the project, Valedictorian was not able to implement the SensUs Digital mobile application portion of the SensUs Digital Platform, meaning that the SensUs Digital Platform consists entirely of the SensUs Digital website. Thus, this Software Transfer Document will focus entirely on how to set up the SensUs Digital website and all testing done on the SensUs Digital website.
### 1.3 DEFINITIONS AND ABBREVIATIONS

#### 1.3.1 DEFINITIONS

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Composer</td>
<td>A dependency manager for PHP.</td>
</tr>
<tr>
<td>Event Attendee</td>
<td>A user of the SensUs Digital website who is physically present at the SensUs competition event.</td>
</tr>
<tr>
<td>Git</td>
<td>A version control system for tracking files and coordinating work among a group of people.</td>
</tr>
<tr>
<td>JavaScript</td>
<td>Client side scripting language used to provide interactive views on our platform.</td>
</tr>
<tr>
<td>Laravel</td>
<td>Open-source PHP web framework for developing MVC web application.</td>
</tr>
<tr>
<td>NPM</td>
<td>Package manager for JavaScript (dependencies).</td>
</tr>
<tr>
<td>Online Viewer</td>
<td>A user of the SensUs Digital website who is not physically present at the SensUs competition event.</td>
</tr>
<tr>
<td>React</td>
<td>Open-source JavaScript library for building user interfaces.</td>
</tr>
<tr>
<td>Redis</td>
<td>Open-source in-memory data store used for caching.</td>
</tr>
<tr>
<td>Dashboard application</td>
<td>A part of the SensUs Digital website where SensUs Personnel and Web administrators can edit content of the website.</td>
</tr>
<tr>
<td>SensUs Personnel</td>
<td>A role of the SensUs Digital website, that has a certain level of permissions allowing specific set of actions on the SensUs Digital website.</td>
</tr>
<tr>
<td>Web Administrator</td>
<td>A role of the SensUs Digital website, that has a certain level of permissions allowing specific set of actions on the SensUs Digital website.</td>
</tr>
</tbody>
</table>
1.3.2 ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATP</td>
<td>Acceptance Test Plan</td>
</tr>
<tr>
<td>DDD</td>
<td>Detailed Design Document</td>
</tr>
<tr>
<td>ESA</td>
<td>European Space Agency</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronic Engineers</td>
</tr>
<tr>
<td>SUM</td>
<td>Software User Manual</td>
</tr>
<tr>
<td>TU/e</td>
<td>Eindhoven University of Technology</td>
</tr>
<tr>
<td>UTP</td>
<td>Unit Test Plan</td>
</tr>
<tr>
<td>URD</td>
<td>User Requirements Document</td>
</tr>
</tbody>
</table>

1.4 LIST OF REFERENCES

1.5 OVERVIEW

The document is structured as follows:
Section 2 describes the build procedure of the SensUs Digital Platform. Note that this procedure is short as it is only required to build some of the front-end assets (e.g. because of using React). Next, Section 3 describes in detail the installation procedure to set up the deployment environment, and recommendations for setting up a development environment. Section 4 provides an overview of all the deliverables (documents and other artifacts) that are being transferred. Then, Section 5 reports the results of the acceptance test specified in the ATP [3]. Section 6 describes software problems (ranging from bugs to enhancements) that are currently in the system. Section 7 describes software change requests. Finally, Section 8 reports software modifications. Note that both Section 7 and Section 8 are not applicable, and as such left empty.
2 BUILD PROCEDURE

This section describes how to build the system. In particular, the steps required to compile the web applications (which are then served by the server application) are stated. Note that all steps apply only for the server which hosts the SensUs Digital Platform, either in production or a development environment.

2.1 PRODUCTION

1. We enable maintenance mode for the server, to avoid issues during the build process.
   
   $ php artisan down

2. We compile, copy and/or process the (front-end) assets.
   
   $ npm run prod

3. We disable maintenance mode for the server.
   
   $ php artisan up

2.2 DEVELOPMENT

For the development environment, building the front-end application requires running:

   $ npm run dev

During development, it is more efficient to run:

   $ npm run watch

The latter will watch all assets, and quickly compile assets that were changed.
3 INSTALLATION PROCEDURE

This section describes how the server application should be installed and configured. We expect a system with GNU/Linux 4.4.0 (or later) installed (we recommend Ubuntu 16.04 LTS). Whenever brackets (“< >”) are used, the brackets and contents are supposed to be replaced by an appropriate value. Commands prefixed with `$ sudo` suggest that a user with sudo rights is required. Note that such a user can be created by running, as a root user:

```
$ sudo usermod -aG sudo <username>
```

### 3.1 DEPLOYMENT SERVER

1. Start a terminal.
2. Start by setting up a basic firewall.
   ```
   $ sudo ufw allow ssh
   $ sudo ufw allow http
   $ sudo ufw allow https
   $ sudo ufw enable
   ```
3. We proceed by installing the LEMP stack (Linux, NGINX, MariaDB, PHP). When running the secure installation for MariaDB (`mysql_secure_installation`) ensure you set a strong root password, remove anonymous users, disallow remote root login and remove the test database. All of this is carefully explained in the installation instructions.
   ```
   $ sudo apt-get update
   $ sudo apt-get install nginx
   $ sudo apt-get install mariadb-server
   $ sudo mysql_secure_installation
   $ sudo apt-get install php-fpm php-mysql php-cli php-mcrypt php- mbstring
   $ sudo phpmodule mcrypt
   ```
4. We now configure the PHP processor. Open the php-fpm config file by running:
   ```
   $ sudo nano /etc/php/7.0/fpm/php.ini
   ```
   Find the commented (line prefixed with a semi-colon) parameter “cgi.fix_pathinfo”. Ensure that this line is uncommented (remove the semi-colon) and is set such that “cgi.fix_pathinfo=0”.
   Restart the PHP processor by running:
   ```
   $ sudo systemctl restart php7.0-fpm
   ```
5. We now configure NGINX to use the PHP processor. Open the default site configuration by running
   ```
   $ sudo nano /etc/nginx/sites-available/default
   ```
   Replace the contents with the following:
fastcgi_cache_path /etc/nginx/cache levels=1:2
    keys_zone=MYAPP:100m inactive=60m;
fastcgi_cache_key "$scheme$request_method$host$request_uri";
add_header X-Cache $upstream_cache_status;
server {
    listen 80 default_server;
    listen [:]:80 default_server;
    root <site root directory>/sensus/public;
    index index.php;
    server_name <domain name (or IP)>;

    location / {
        try_files $uri $uri/ =404 /index.php?$query_string;
    }

    location ~ \.\.(\d+)\.php \$ {
        try_files $uri /index.php =404;
        fastcgi_split_path_info ~(\+\.(\d+))(\./.*)$;
        fastcgi_pass unix:/run/php/php7.0-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
        fastcgi_cache MYAPP;
        fastcgi_cache_valid 200 60m;
    }
}

Restart NGINX by running $ sudo systemctl reload nginx.

6. We continue with installing and using Git in order to retrieve the system.

$ sudo -apt-get install git
$ git clone https://github.com/rolfmorel/sepgroup7.git <site root directory>
$ cd <site root directory>
$ git checkout master

7. Composer is required to install the back-end dependencies, including Laravel.

$ cd ~
$ curl -sS https://getcomposer.org/installer | php
$ sudo mv composer.phar /usr/local/bin/composer

8. Similarly, we require Node.js with NPM to install front-end dependencies.

$ curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -
$ sudo apt-get install nodejs build-essential

9. Next, install all dependencies using NPM and Composer.

$ cd <site root directory>/sensus
$ composer install --no-dev
$ npm install --production

N.B.: To avoid access to the website at this point, it is recommended to run:

$ php artisan down

This will put the website in maintenance mode.

10. Finally, we configure the Laravel application.

$ cd <site root directory>/sensus
$ cp .env.example .env
$ php artisan key:generate
$ nano .env

This will open the “.env” file, which contains the private configuration for the server application. In particular, ensure that DB_PASSWORD and DB_USERNAME are set to a MariaDB user that has rights on the “sensus” database. We strongly recommend against using the “root” MariaDB user, and instead creating an user account with local access. [7].

### 3.1.1 DATABASE

Before the server is deployed (i.e. built) it is necessary to set up the database.

1. Create the “sensus” database (empty).

$ mysql -u <mariadb username> -p

You are prompted for the MariaDB password, after entering it a SQL prompt will start.

```
CREATE DATABASE sensus;
```

2. We now create the database tables and run the seeders to insert rows.

$ php artisan migrate --seed

### 3.1.2 REDIS

A naive way to install Redis is stated. However, we recommended following a detailed guide which can be followed without any hassle. [8] The result is a much more secure installation for multiple reasons, as well as a better configuration.

1. If instead the naive way is followed, we simply install redis.

$ sudo apt-get install redis-server

### 3.1.3 LARAVEL ECHO SERVER

Finally, we explain the installation procedure for the Laravel Echo Server.

1. Install the Laravel Echo Server (globally) via NPM.
$ sudo npm install -g laravel-echo-server
$ laravel-echo-server start

3.2 DEVELOPMENT

The installation procedure for development is similar to the installation procedure of the Deployment Server. The main difference is in step 9.

9. Instead, install all (develop) dependencies using NPM and Composer.

$ cd <site root directory>/sensus
$ composer install --dev
$ npm install

Furthermore, step 2 can be skipped (we recommend not allowing any remote traffic to ports, since only local access is required).

Alternatively, we provide an in-depth guide for Windows user on GitHub, which includes some completely optional steps to enhance PhpStorm. [9]
4 CONFIGURATION ITEM LIST

Here we give a list of items that Valedictorian will be delivering. All documents will be given in a pdf format (both documents and test plans). We believe pdf will the best easiest format for SensUs to read them. And the source files of the software will be the items handed over (as the SensUs might be interested in expanding the project).

4.1 DOCUMENTATION

- URD [1]
- SRD [2]
- DDD [4]
- SUM [5]
- STD (this document)

4.2 TEST PLANS

- ATP [3]
- UTP [6]

4.3 SOFTWARE

- SensUs Digital Platform, consisting of:
  - SensUs Digital Server Application
  - SensUs Digital main web application
  - SensUs Digital dashboard web application
5 ACCEPTANCE TEST REPORT

The first acceptance test, as described in the Acceptance Test Plan version 0.0.1, was performed on June 23rd 2017, between 9:45 and 12:00. This acceptance test was done as a trial run to point out the problems with the product for the team as well as for the client. Therefore, a substantial amount of tests did not pass. These tests are listed below:


The final acceptance test, as described in the Acceptance Test Plan version 1.0.0 [3], was performed on July 4th 2017, between 21:15 and 23:00. For this test the client, supervisor and 3 team members were present. All the tests in this document passed with the exception of test ATD-29 and ATM-17. The issues causing these failures were small and fixed almost immediately. On Juli 6th 2017, between 9:20 and 9:30, tests ATD-29 and ATM-17 were performed by the client, with two team members present, and succeeded.
6 SOFTWARE PROBLEM REPORT

6.1 VIDEO STATISTICS

Due to the limited time available for the project, Valedictorian was not able to implement features to enable users to view statistics on videos (and possibly other media). For example, the number of views, likes and dislikes of a video could be shown on the explore page. Ideally, these statistics should be managed in a dashboard to enable/disable statistics of videos. Even though this has not been implemented, the API calls to YouTube are already used to show these statistics (and some other information) on the control dashboard.

6.2 REAL TIME UPDATES

The SensUs Digital Platform supports and uses real time updates using web sockets with Laravel Echo Server. In particular, the total number of votes (i.e. vote count) is updated for all users that currently view the vote count whenever a new vote is cast. However, due to time constraints we have not implemented real time updates for other features such as measurement data, or, for example, on approval of a video. As such, extending this functionality to other features is a desirable change to the software. Alternatively, removing the web sockets and Laravel Echo server and instead using periodic pulling for the vote statistics (similar to the measurement statistics) is an option.

6.3 CONCURRENT ADMINISTRATION

The SensUs Digital Platform allows multiple administrators or SensUs personnel to use the same dashboard simultaneously. As a result, it is possible for changes to be overwritten accidentally as a result of concurrent work. It is desirable to protect some dashboards from this, for example by allowing only one user to work on it at the same time, or alternatively, by notifying other editors.

6.4 EVENT DASHBOARD

The dashboard web application contains an event dashboard which is used to showcase information regarding the event, such as live graphs, social media feeds and submitted media. The event dashboard is in place and contains for example the possibility to show vote statistics (per team) and a Twitter feed. Nevertheless, more options are desirable such as graphs with live measurement data.
7 SOFTWARE CHANGE REQUESTS

Not applicable.
8 SOFTWARE MODIFICATION REPORT

Not applicable.