Expertise
and insight
for the future

Sudarshan Koirala

Content Management System Customization

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

14 November 2018

Author Title Number of Pages Date	Sudarshan Koirala Content Management System Customization 41 pages + 10 appendices 14 November 2018
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Software Engineering
Instructors	Ilkka Kylmäniemi, Senior Lecturer

The purpose of this final year project was to develop a fully functioning website with the implementation of user management tools to speed up new account creation and reduce the administrator's current workload. Along with that, making the website fully responsive and implementing a modern language version to translate the page into different languages also falls under the main objective of this project.

The skeleton of the website was developed using WordPress CMS, custom plugin, custom theme, and different already available WordPress plugins. The Ultimate Member plugin was used for the user management, WPML for the translation and Bootstrap was used for the responsiveness of the website.

Keywords	WordPress, theme, plugin, CMS, translation

Contents

Appendices

List of Abbreviations

CMS	Content Management System
HTML	Hypertext Mark-Up Language
PHP	Hypertext Pre-Processor
SQL	Structured Query Language
CSS	Cascading Style Sheet
SaaS	Software as a Service
RAM	Random Access Memory
SEO	Search Engine Optimization
WPML	WordPress Multilingual

1 Introduction

This thesis project is one of the Motiivi project supervised by Metropolia University of Applied Sciences and is developed for the young people through Finnish Settlement Association's NäytönPaikka service. The aim of this project is to transform the existing website to a completely new version and at the same time add more features to make it more user friendly.

Motiivi project is working for bringing the existing methods of future work for the young people, develop and evaluate the new methods of working. It is working in developing policies to support the young people aged 16 to 29 who are unemployed and out of work to identify their obstacles and help in their professional growth. Most importantly, during the development, the project will pay attention for the immigrant backgrounds to bring the equality among all and focus on the health issues. In order to achieve all this goal, the digital tools made during this project will be used. (Metropolia, 2018.)

NäytönPaikka service is a free of charge, customer oriented and resource-based tool which helps a person to reflect their life comprehensively from different perspectives. With the different tools available in the website, such as life situation mapping tool, a timeline tool and a network map, the users can have a closer look at their strengths, challenges, threats and possibilities. The service also provides workers with a complete and versatile way of customer oriented professional work against marginalization and in addition to that support inclusion, growth and empowerment. (NäytönPaikka, 2018.)

As the website covers all the solution, the web development needs to cover a wide range of existing fields as well as many new features. This report, therefore, focuses on differnet areas of the development and design processes which will be discussed in detail. The objective of this final year project is as to provide the case company with the fully functioning new website with the implementation of user managemnet tools so that it speeds up the creation of new accounts and reduce the administrator's current workload. Along with that, making the website fully responsive and implementating modern language version to translate the page into different language also falls under the main objective of this project.

The project is under the Motiivi project so in order to make the website user friendly and to achieve the objectives of the project, meetings were carried out with members of the Motiivi

project. The website is build in wordpress using all the available wordpress development tools alontg wth HTML, CSS, JavaScript, PHP, MySQL and Bootstrap.

The first section of this study describes the overall description Content Management System as the website for the case company is built on Wordpress. The second section deals with the general information of Wordpress and the reason behind choosing it. The third section focuses on custom design and implementation of the theme and plugin. After this, the user managemnet tools and the website translation will be explained. Finally, the results and success of this project are analysed.

2 Content Management System

A Content Management System (CMS) is a software which is designed for the users to create, edit, organize and publish digital content. It can be used in any size business operations, but small business can gain a lot from using the CMS technology to create and manage content without much technical knowledge or a hefty budget. (Pickard-Whitehead, 2018.) This section describes the histroy and differnet types of CMS. It also explains the advantages and disadvantages of CMS along with the difference between CMS and Hard Coding.

2.1 History

Back in mid-1990s, only dynamic pages were e-commerce sites going with Perl, Cold Fusion or whatever possible as getting HTML pages to display properly was a hard thing. After the PHP and other programming language evolve in late 1990s, people start figuring that it was a good idea to edit content on their own sites which eventually result in people writing CMSs. After writing CMSs for themselves, people start commercializing their CMSs for different businesses along with large magazines and newspapers. Around year 2000, the open source concept of CMSs evolved, and it was starting to be ready for the prime-time by about 2004. (Laminack, 2008.) After that, many CMSs started to evolve with their own unique features which fits for the user's preferences.

2.2 Types of CMS

Choosing CMS for your business depends upon the features, functions and pricing models of different CMS. There are mainly three broad types of CMS software: open source, proprietary and Software-as-a-Service CMS, including cloud-based solutions. (NIBUSINESSINFO, 2018.)

Open source CMS

Open source CMS is the most used CMS as it can be downloaded without any initial cost and no licence or upgrade fees is required. Some of the most common and widely used open source CMS are Wordpress, Joomla, Drupal, Magento and PrestaShop. This CMS can be

installed and managed in web server along with adding countless customisations according to the business needs such as adding plugins, installing search engine optimization tools, customisig the themes and layouts and many more. Although these softwares are free to download, there can be some charge for:

- Technical help during installation and set up
- Adding more features than the core offerings
- Customizing the templates, add-ons and plugins
- Training for the staffs
- Continuous support and software update

(NIBUSINESSINFO, 2018.)

Proprietary CMS

Proprietary CMS which is also known as commercial CMS software is built and managed by a single company and involves buying a licence fee to use software as well as paying annual or monthly fee for updates or support. Choosing this CMS depends upon the stage of the website where it is going to be implemented. New websites should choose the CMS which fulfill the current features and flexibility as well as future needs too whereas if the CMS is going to be implemented in the running website, CMS must be choosen with great care as it must meet all the requirements out of the box and extensive development work may be needed in the back-end part. Some of the examples of the popular Proprietary CMS are Kentico, Pulse CMS, Sitecore, Shopify, IBM Enterprise Content Management and Microsoft SharePoint. (NIBUSINESSINFO, 2018.)

Software as a Service (SaaS) CMS

SaaS CMS solutions comprises of web content managemnet software, web hosting and technical support from a single supplier which are hosted in the cloud and are based on a subscription model. The pricing can be usually per-user or per-site basis and is deterined accoridng to the amount of data transfer, storage and ongoing support. The cloud CMS can be fully cloud or partial cloud and it can be choosen according to the company need. Fully cloud is under the control of supplier and comes as a part of a package or service and isn't always possible to customise whereas partial cloud CMS is located in the companies own cloud web-sever which gives more flexibiltiy in modifiying the functionalities either with add-on modules or by altering the source code itself. (NIBUSINESSINFO, 2018.)

2.3 Advantage and Disadvantage of CMS

Before building a website, anyone must know whether to use CMS or not. Running websites can also implement CMS according to their necessity but need to know what they can benefit from it or the consequences they may have in the future.

Advantages of CMS

- CMS is centrally monitored portal, so it is easy to administer, and users should not necessarily know too many technicalities to upload content.
- Fewer employees needed in the IT department to handle the tasks which helps in cost reduction.
- The interface is user friendly so in less time more data can be updated which definitely saves time.
- Users can control the accessibility of content to the public.
- CMS offers extended measures of security.

(4MDesigners, 2014.)

Disadvantages of CMS

- Constant updates need to be done in order to function properly with CMS
- Not all the additional features and functionalities will be compatible with the existing CMS and a need of programmer may be needed to fix it.
- CMS may dominate the RAM space and resources of a computer, so one may need to operate on a fast processor.
- CMS hosting can be expensive as the web design companies may advice using a system depending upon how frequently the site needs to be updated.
- Direct support may not be available so need to rely on forums and existing documentations.

(4MDesigners, 2014.)

2.4 CMS vs Hard Coding

Building websites can be easy but the platform selection to build the website will be always complicated as it should fulfill the needs and one should consider about having possible problems in the future. The main question always pops up whether to use the traditional hard coding or opt for CMS.

Manual coding or hard coding involves the functional code in the basic language in which they will be compiled and helps in fulfilling almost all desires to fit for customers. It has better control as the web templates can be controlled by having only specific features which is a challenge in using CMS. Having a freedom in using shortcuts while coding from scratch also helps for the higher speed but it can be high priced as everything needs to be done from the beginning. Vast knowledge is also required in the specific programming language of coding and can take long hour in debugging too. (DeanInfotech, 2017.)

Content Management System in other hand is an uncomplicated system which allows to manage and control the features of the website without necessarily having technical skills. It falls under convenient programming and help manage multi-user with the ready-made template. It will help reduce the errors and save time as debugging is almost not necessary. Compared to hard coding, using CMS is far more secure as the provider will take care of security but the software must be kept up to date and may need lots of update. CMS has many choices of customization as it must fits to the need of many users so it can be more inconvenient in customizing the website as in hard coding. (DeanInfotech, 2017.)

3 WordPress

WordPress is the most popular open source CMS released in 2003. Initially, it was known as a blogging tool but currently it is also being used in building different kinds of websites ranging from normal website to online stores. Among the top 1 million sites, the CMS usage distribution is depicted in figure 1.

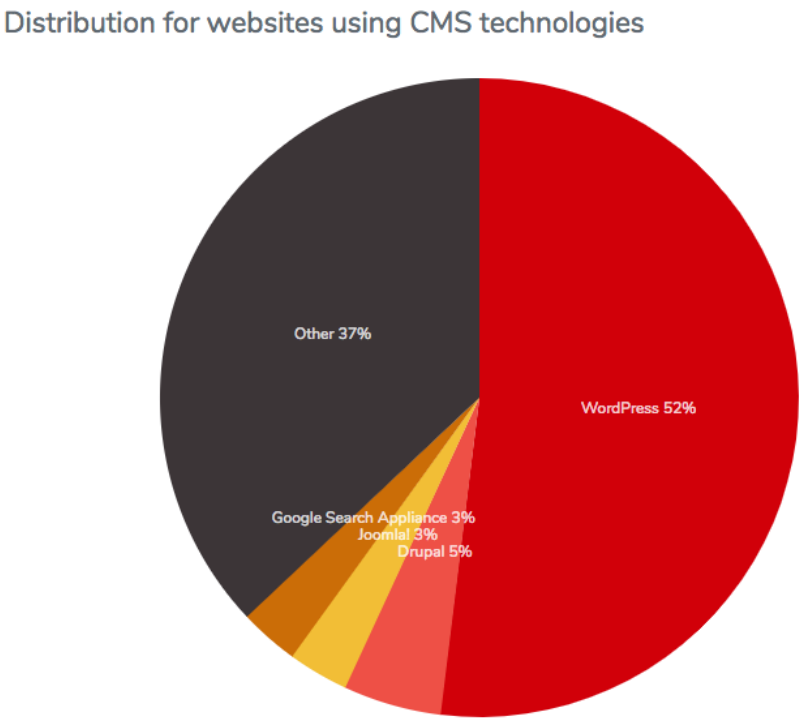

Figure 1: Distribution for websites using CMS technologies (BuiltWith, 2018).

As illustrated in figure 1, WordPress is the mostly used CMS which shows how popular it has been for users to build websites on it.

3.1 Comparison with similar CMS?

As from the figure 1, it is clear that WordPress is the leader among the other CMS. The main competitors out there are Drupal and Joomla, but both have significantly low usage. If you look at the stats from the Google trends on the top 3 CMSs in figure 2, WordPress is the only which have been in interest over time. The reason behind this trend can be that, WordPress has big

online community and the tools available online for web development are way more than other CMSs.

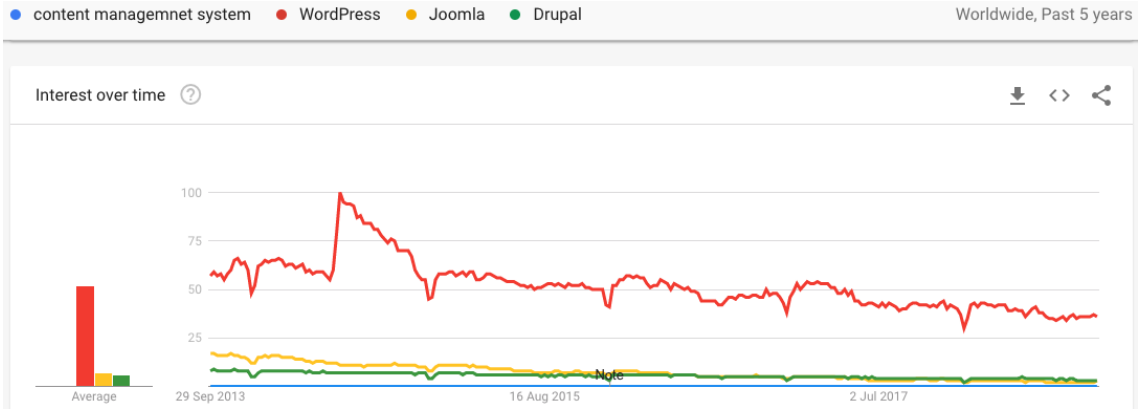

Figure 2: Popular CMS Comparison

The statistics in figure 2 clearly show that Drupal and Joomla are significantly losing the ground which means that more development will eventually go into WordPress platform. In the last 5 years as shown in the figure above, WordPress has maintained to somehow stay strong among others and it is also expected to carry on as it is also being used by more people every day including big companies like Facebook, CNN, TED, TIME, NBC Sports, TechCrunch, etc (Elgameel, 2018).

3.2 Why WordPress?

WordPress is the most used CMS as shown in figure 2 and it is easy to maintain and update. It is versatile as we can implement more visual appearance and implement many features and integrate contents from different systems. In this project, this CMS was chosen as it is easy in creating custom themes and plugins and the technical as well as programming skills is not needed to handle the website. Along with this, website translation is also the main field of this project and WordPress is more convenient in using translation plugin. Lastly, WordPress has good support and materials are found easily online as well as it is convenient for Search Engine Optimization (SEO) too.

4 Custom Design and its Implementation

The objective of custom design is to keep the interface simple and be purposeful for the page layout as well as meet the exact needs of the users. This section describes the design and implementation of custom theme and custom plugin with the reason behind choosing it despite having many ready-made plugins and themes.

4.1 Custom Theme

WordPress theme provides the control over the look and the material presentation on the website with the collection of files that work together. It modifies the way a site is displayed without modifying the software using the different customized template files, images files, style sheets, custom pages along with the code files. With the help of different customized template files which can be called to generate a WordPress page as shown in figure 3, the theme can be designed according to the user's expectation. (WordPress, 2018.)

Figure 3: WordPress template hierarchy (WordPress, 2018)

As illustrated in figure 3, WordPress needs to follow a certain hierarchy of template files and a clear understanding of this is needed to customize the theme.

Why Custom theme?

WordPress theme is the combination of different files that create the design and functionality of the website. In order to completely transform an existing website to new WordPress website, the customization of theme is chosen rather than ready-made themes. Apart from that, it's an opportunity to learn more about CSS, HTML and PHP and put the expertise with these tools in work and have fun coding (WordPress, 2018). Although, creating a custom theme is time consuming and needs more budget, following points prove why a custom theme is needed.

- Lack of theme which suits the vision, goals, requirements and contents.
- Time consuming for choosing the right theme
- No need to be limited in design options
- Want to have an original, unique design that stands out from the crowd.
- Lightweight code with no feature bloat for things not necessary

  (cre8d, 2016.)

Theme Design and Implementation (page, post, categories)

WordPress theme live in the subdirectory of WordPress themes directory (wp-content/themes) by default. The subdirectory holds all of the theme's stylesheet files, template files, and optional functions file (functions.php), JavaScript files and images. (WordPress, 2018.) For example, in this project, the theme npadmin resides in the directory wp-content/themes/npadmin.

The stylesheet controls the presentation of the website pages as all the visual design and layout is controlled through this file. Template files are created according to the necessity of the project and controls the way the site pages generate the information from the database and display it on the site. Along with this functions file (functions.php) also considered as the root file acts like a plugin and adds features, functionalities to the WordPress site through php code. (WordPress, 2018.)

Custom theme creation in WordPress basically needs two files in the beginning for the software to recognize that it is theme. Index.php acts as a main template and style.css the main stylesheet. The style.css used in this project is shown below.

```
/*
Theme Name: Naytonpaikka Theme
Theme URI: https://www.naytonpaikka.fi
Author: Naytonpaikka
Author URI: https://www.naytonpaikka.fi
Description: Theme for Naytonpaikka
Version: 1.0
Text Domain: nptheme
*/
/*
 * Globals
 */
```

Figure 4: theme stylesheet (style.css) file

As illustrated in figure 4, style.css provides detail information of the theme in the form of comments. It is important to note that the details must always be in comments and no two themes are allowed to have the same details as this will lead to problems in the theme selection dialog in admin panel.

The basic file structure or subdirectories of the theme for successful website creation includes the files as shown in figure 5.

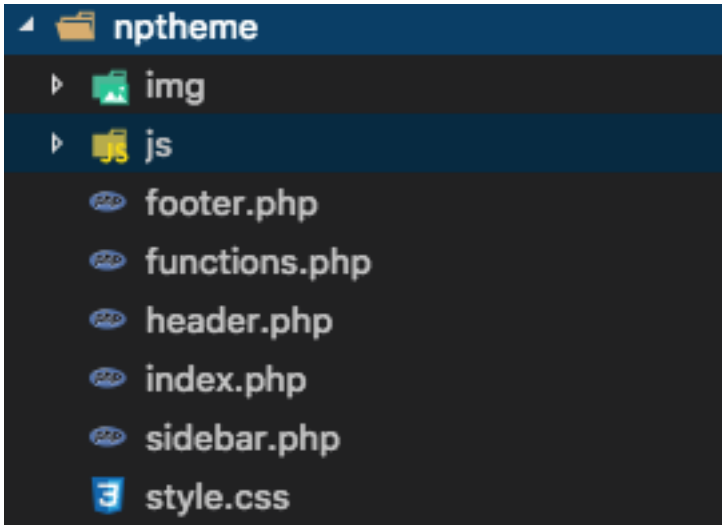

Figure 5: Basic subdirectories of theme

Figure 5 illustrates the basic file structure of the theme for the site creation. Along with these files, in this project other custom page templates were also created to generate the information in the specific page for the users.

Header.php

The WordPress header by default is a simple piece of code but the header file contains all the necessary code for the header section of the theme. As being the main section of the website people first look after opening the website in the browser, it contains the navigation menu with sub-menus along with other necessary links to navigate in the website. For this website, as it contains three different page layout, one the front page without login while the rest two are for the users and employees after login.

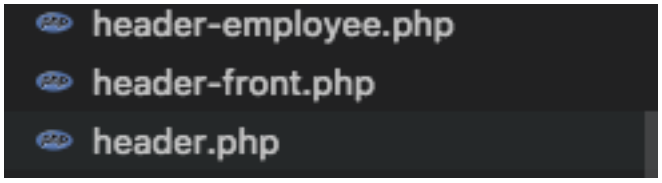

Figure 6: Name of three different header files created

As illustrated in the figure 6, the header-front.php contains the code for the header file for the front page of the website which can be viewed by everyone landing on the website. The header.php file contains the code for the users after being logged in to the website and the header-employee.php is for the employee page after being logged in.

By creating these three different headers, it also makes coding smooth as get_header can be used to render all the headers where necessary.

<?php get_header(); ?>
This php code can be used to render the simple header file in any page.

<?php get_header($name); ?>
This php code can be used to render the header file with any name associated with it. In the website, $name is replaced either by front or employee to implement header-front.php or header-employee.php where needed.

Bootstrap navbar was used in all the three header files in order to make the header responsive and easy to use in the smaller devices. Along with this the header file contains the code to display the breadcrumbs using the Yoast SEO which can also be used to make a site as search engine-friendly as possible.

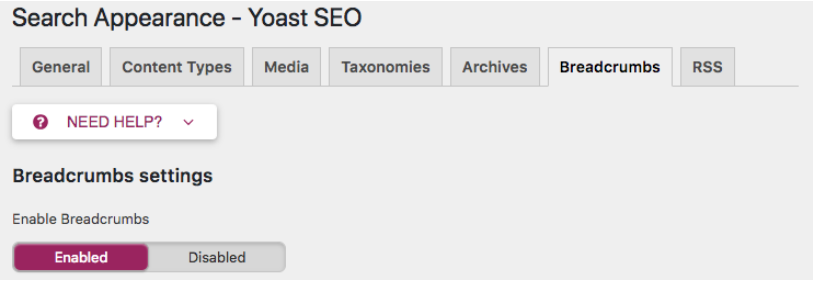

Figure 7: Enabling the breadcrumbs inside Yoast SEO

Figure 7 shows that the breadcrumbs are being enabled in the Yoast SEO plugin.

Sidebar.php

The WordPress Sidebars are used to display widgets inside the theme and can be used anywhere if needed. The website contains three different sidebars and are created in the functions.php file of the theme. After that, the sidebars appear on the widgets area of the theme and the available widgets can be included inside the sidebars. In order to render the custom sidebar in different page, three different php files are created and codes are written inside it.

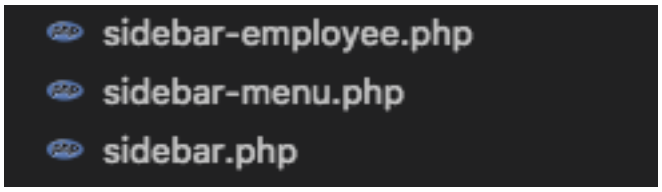

Figure 8: Name of different sidebars created

Figure 8 shows the list of different sidebars created in order to insert the widgets in necessary places. For simplicity, in this website three menus apart from the main header menu are created so that the menus can be inserted as a widget inside the sidebar.

Sidebars in WordPress works exactly as header file.
<?php get_sidebar(); ?>
This php code can be used to render the simple sidebar in any page.
<?php get_sidebar($name); ?>

This php code can be used to render the sidebar with any name associated with it. In the website, $name is replaced either by menu or employee to implement sidebar-menu.php or sidebar-employee.php where needed.

The menu with name Sidebar Front Page is being used as a widget for the normal Sidebar while the menu with name Sidebar Menu is being used inside Sidebar-Menu. Similarly, the menu named Sidebar Employee is used inside Sidebar-Employee as shown in figure 9 below.

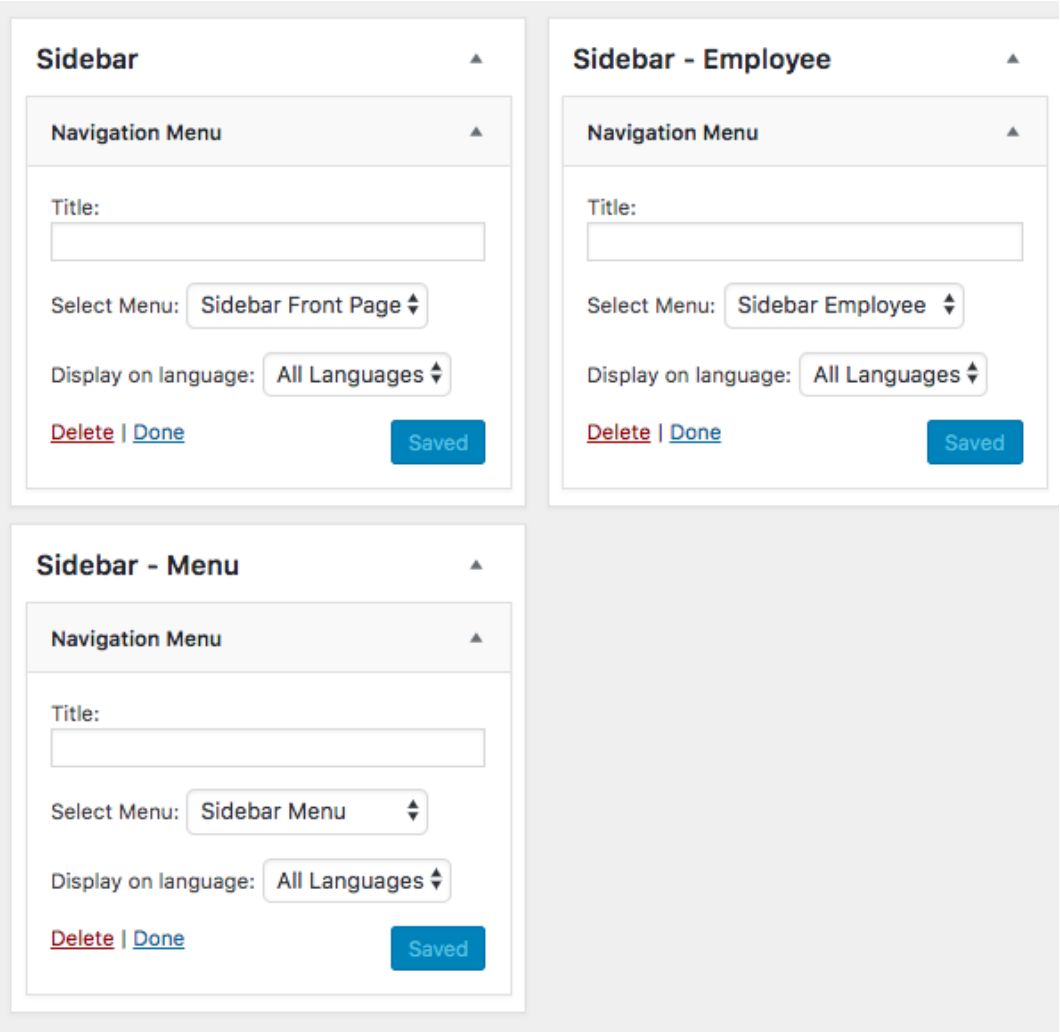

Figure 9: Screenshot of three different Sidebars with menu as Widget

As can be seen in Figure 9, three different menus are being used as a widget for three different sidebars so that it can contain different links to navigate inside the website conveniently.

Functions.php

The functions.php file of WordPress theme behaves like a WordPress plugin which adds features and functionality to a WordPress site. It is mainly used to hook into the core functions of the WordPress to make that particular theme more modular, extensible and functional. It is stored in the theme's subdirectory in wp-content/themes and requires no unique header text and executes only when that particular theme is activated. As this file can have numerous blocks of code used for different purposes, there is no need to be worried about the loss of any code in custom made theme as no update is needed as other third-party themes. (WordPress, 2018.)

Some of the uses of functions file are listed below.

- Enqueueing theme scripts and stylesheets.
- Enabling theme features such as sidebars, navigation menus, post thumnails, post formats and others.
- Defining functions that are used in several templates of the theme.

(WordPress, 2018.)

The following code from appendix 2 shows how the header navigaton menu is enabled in the theme.

```
function nptheme_register_header_menu()
{
   register_nav_menu('header-menu', __('Header Menu'));
}
add_action('init', 'nptheme_register_header_menu');
```

Listing 1: Registering header menu

Listing 1 shows how a header menu can be registered in the WordPress theme through functions.php.

Index.php

Index file is the default template file for rendering the website. If any content is not being asked to execute, WordPress will automatically fall back to this file and execute it. This is the file

which needs to be created in order for the theme to appear in the theme section of the WordPress dashboard to be activated. The code of this file is included in appendix 1. If no specific page or category templates are assigned or found, WordPress will automatically default back to using the theme's index file to render the pages or categories. So, this file can be taken as a last rescue for displaying the pages in the website.

Style.css

Style.css provides detail information of the theme in the form of comments. Apart from that, this template file is very crucial for the custom theme as the styling of the website is done completely from this file and it controls the visual design and layout of the website pages. Instead of modifying the theme directly, using custom CSS ensures that the modification is preserved.

The above described file structure or the sub-directories of the theme are used for each and every theme development. Apart from these files, other many custom files are created in the theme in order to display the content as needed which cannot be displayed through WordPress pages, categories and posts. In order to display the content of the corresponding pages, categories and posts, php files must be created through will the specific content can be extracted and displayed.

During the theme development, all the categories and pages have the corresponding php files with the same name as created in the WordPress dashboard so that the contents can be displayed with the php codes written in those php files. The created page and category php files with the code inside it and how those codes extract the content and displays in the website is described below.

Category.php

WordPress theme can be created with different templates for different sections of the website as WordPress has a powerful templating system. For instance, to display a category page, it looks the template in the order as shown below.

Category-slug.php –> category-id.php –> category.php –>archive.php –>index.php

At first, WordPress will look for the category that uses the category slug, for example, category-diary.php template will be used to display 'Diary' category. If it does not find the category-slug template then WordPress will look at the template with the category id, for example category-1.php. If this is also not present, then it looks for the general category template, i.e. category.php and if this is not available too, it looks for general archive template, i.e. archive.php. Finally, it will use the index.php template to display the category. (wpbeginner, 2015.) Furthermore, in order to display the category, there must be atleast one post associated with that category.

Moreover, Categories images WordPress plugin was used so that the image can be assigned to that particular category for visual identification too. Along with this, for custom sorting the categories and identifiying the parent and child categories just by viewing it, a plugin named Taxonomy Terms Order was used.

While creating the theme for this website, first the normal category.php template was created as shown in appendix 3. As the new category were being created in the dashboard with certain name, the category-slug.php template file was created to display this category differently than others. For example, a category called "My diary" with the category slug "my-diary" was created and the template file with name category-my-diary.php was created as shown in appendix 4 which eventually displays only the posts related to that category if the posts are assigned to that category. This process of creating the category-slug.php and showing the content of that specific category was done for all the category created in this theme. Altogether, six different posts were assigned to the category My Diary, so the outcome in the website under the category My Diary is shown as below.

Figure 10: Screenshot of the posts related to category My Diary

As illustrated in the figure 10, the category template My Diary displays only those posts which are assigned to that category.

Page.php

WordPress by default allows to create posts and pages but with the help of a template file called page.php as shown in appendix 5, the theme can control the appearance of the pages. Before making the custom page template for the theme, it should be taken in consideration whether the page template will be for one specific page or for any page and what type of user control can be implemented for the template (WordPress, 2018).

Apart from this main custom template, two different template files are created to display the different content in different template. FrontPageTemplate is used to display the content in the front page of the website and the code is available in appendix 6. EmployeePageTemplate displays the content of the employee page after the employees are being logged into the web-site and appendix 7 contains the code for this template. These extra template files are created by adding the template name on the header section of that page. With just adding the template name, WordPress recognizes these pages as the template page and it can be choosen by any page to use that template.

The WordPress looks for template files in the following order.

Page Template –> page-slug.php –> page-id.php –> page.php –> singular.php –> index.php

At first, WordPress will look if the page has a custom template assigned it. If not, it looks for the page that uses the page slug, for example, page-mysection.php template will be used to display 'My Section' page. If it does not find the page-slug template then WordPress will look at the template with the page id, for example page-1.php. If this is also not present, then it looks for the general page template, i.e. page.php and if this is not available too, it looks for theme's template used for a single post, irregardless of the post types, i.e. singular.php. Finally, if no specific page templates are assigned, WordPress defaults back to the theme's index file, i.e. index.php. (WordPress, 2018.)

During the theme development, pages and posts were created in the WordPress and to make the page content different from one another php files of the same pages were created with the slug. For example, a page with My Section was created in the WordPress as it allows to create page by default. Although, the content can be described inside the Wordpress itself, for including more precise content, a file was created with the name page-mysection.php with the php codes as shown in appendix 8 inside it to display the specific content to that page only. In this theme, the page My Section is defined in such a way that it displays all the child categories of the parent category My Seciton. The outcome of this page in this website is shown as below.

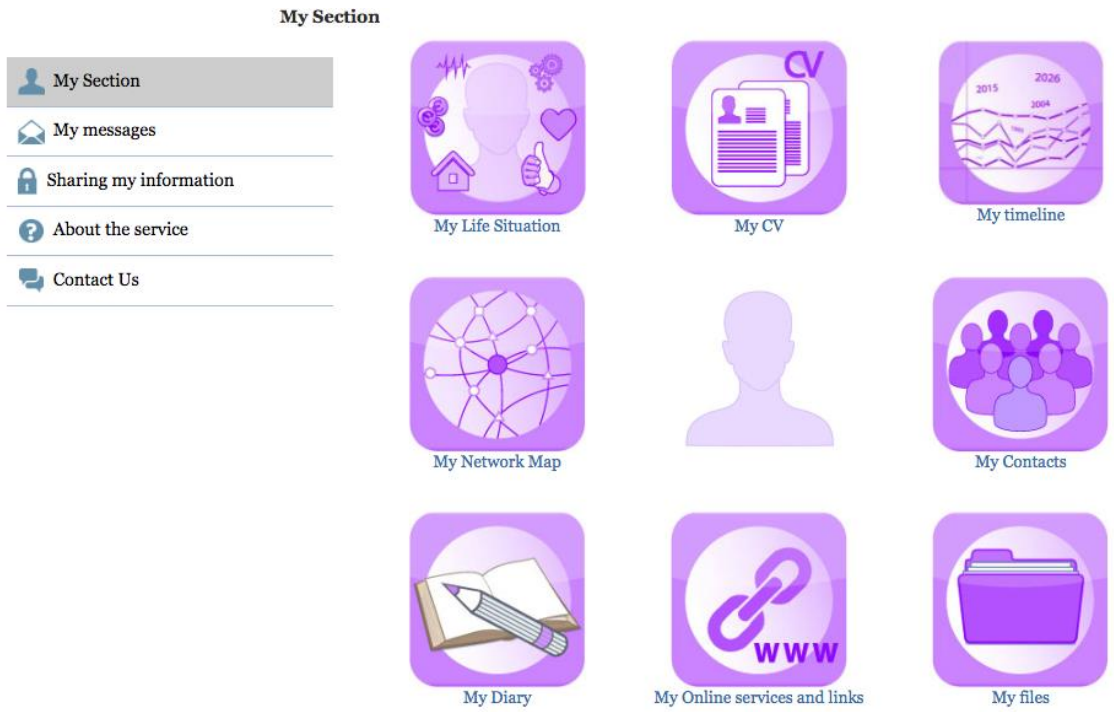

Figure 11: Screenshot of the content of page My Section

As illustrated in the figure 11, the page My Section displays all the child categories of the parent categories My Section.

4.2 Custom Plugin

WordPress plugin is a program which is written in PHP scripting language and allows to add a specific set of features to the WordPress site. Along with this, it allows easily to modify, customize and enhance the site by adding functionalities rather than changing the core program code of WordPress. (WORDPRESS.ORG, 2018.)

Why Custom Plugin

Many plugins are available to implement in the WordPress site but finding the right one which suits the basic need is always a hassle. The basic idea of making the custom plugin is extending the functionality of the website without touching WordPress core itself and simple interface. As the core files will be overwritten during the WordPress updates to new version, having own plugin is preferred.

The website for this project contains many forms and a questionnaire for the users, so for the simplicity of creating form and adding questionnaire in the website, a simple plugin with name NPForms was created with simple interface so that admins can also create and modify the content in the future if needed.

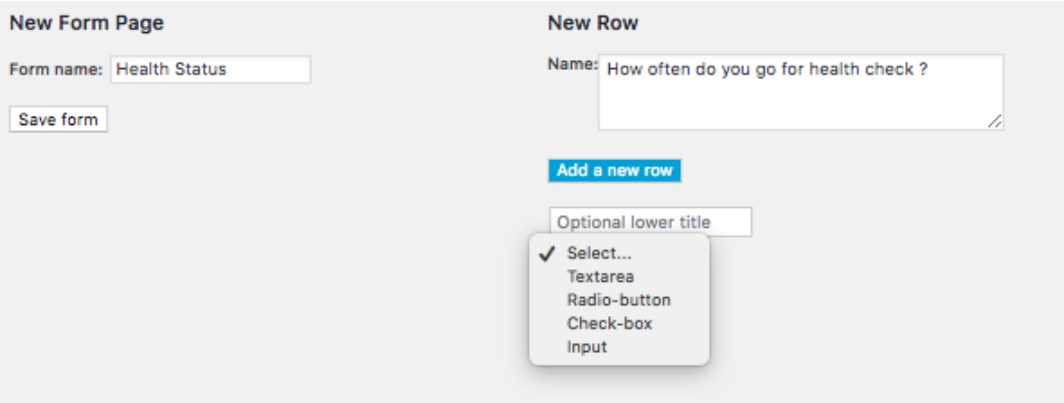

Figure 12: Screenshot of NPForms simple interface

The figure 12 illustrates the simple plugin interface where on the left side the name of the form can be written and on the right-side questions for the form can be added and submitted to the database.

The plugin just needs the text area, radio-button, check-box and input to be filled in the forms, so the custom plugin comes in handy as the plugin can be created with just the needed functionality for the website. After the needed questions for the forms being loaded, the form can be submitted to the database. After submitting the form, shortcodes are created which can be then implemented in the WordPress website to display the form directly instead hard coding the form.

4.3 Responsiveness of the Website

Today, majority of the user's website visit starts on the mobile device because of which optimizing the website accessibility and user experience on tablets, smartphones and every device is becoming necessity rather than choice. It is therefore, to attract users, websites are designed responsively. Generally, for molding website design and user experience to user's device, whether desktop, tablet or mobile, the concept of responsive design is evolved.

(Ricotta, 2018.) From many ways to make the website responsive, for NäytönPaikka, bootstrap is choosen as it is one of the most used, intuitive and powerful front-end frameworks and is created for developing responsive websites with main focus on mobile devices.

Bootstrap has the extensive and thorough documentation as well as being open-source project with extensive development and continuous maintenance, it is choosen for responsiveness design. It deals with grids, utilities and media queries which helps free development time and can be used to quickly create mobile-first and mobile-optimized WordPress theme without reinventing the wheel. Specific project requirements were easily meet as Bootstrap can be easily customized after having enough knowledge of the available classes. (Bouchefra, 2018.)

Bootstrap uses the grid system which uses a series of containers, rows and columns to layout and align content which is built with flexbox and fully responsive (BootStrap, 2018). The use of this grid system is used in all the code written for this website to make the website responsive. The main navigation menu for this website is also created with the help of Bootstrap so that the navigation bar can extend or collapse depending on the screen size and the code is included in appendix 9.

For integrating bootstrap navigation with the WordPress menu, a php file named wp-bootstrap-navwalker.php was saved in the root directory of the theme and that php file was included in the functions.php with the require_once() statement which is shown in appendix 2. The bootstrap stylesheet and JavaScript files were also enqueued in the functions.php file. Along with this some media queries were used for some custom responsiveness. Some examples are shown in figure 13.

```
@media (max-width: 600px) {
  .container {
    width: 100%;
  }
  .carousel-caption {
    display: none;
  }
  .breadcrumbholder,
  .scroll-up {
    display: none;
```

Figure 13: Screenshot of media queries example

As illustrated in figure 13, the media queries to make the container width 100% in devices having max-width of 600px as well as not displaying the carousel-caption and breadcrumbs for that width devices.

After implementing the bootstrap for responsiveness, the NäytönPaikka website was responsive in different devices as shown below.

Figure 14: Website responsive in different devices

As illustrated in figure 14, the website is responsive in various devices.

5 User Management System

User management describes the ability to manage different user access to various resources in the system. WordPress allows to assign different user roles which include administrator, author, contributor, subscriber and regular user and this can be assigned to many members of the team. The built-in user management system which comes with WordPress does not allow the full control over the users so many user management plugins are available. (Silkalns, 2018.)

5.1 Ultimate member plugin

Ultimate member is the user profile and membership plugin for WordPress which makes it a breeze for users to sign-up and become member of the website. It is lightweight, responsive and highly extendible plugin which will enable to create almost on any website where users can join and become member with absolute ease. Having more paid extensions and not requiring having any coding knowledge or manually build shortcodes makes it more promising plugin. Social login can also be implemented making user registration and login simple. The most prominent features of this plugin are as follows.

- Front-end user profiles
- Front-end user registration
- Front-end user login
- Custom form fields
- Drag and drop form builder
- User account page
- User emails
- Content restriction

(UltimateMember, 2018.)

5.2 Complexity of using it

Ultimate member plugin has the simple layout and having able to use shortcodes and no need of any programming knowledge needed makes it easy to use. The paid version of the plugin was used for good support from the plugin team and has good documentation for all the basic use cases. In this website, as we have mainly two different user roles customer and employee with administrator being the default one having access to all the user, this plugin comes handy in managing the user roles. Subscriber is taken as customer and author as employee for managing user roles. Some of the most prominent features of this plugin implemented in this website is described below.

Registration

The default registration form with this plugin helps users easily register and become member with users to be auto-approved, require email activation or be manually approved by the admin. Along with the pre-defined registration fields, custom fields are assigned from the available field manager. The custom user roles can be assigned to the form so that anyone who registers using that form will automatically be assigned that specific role. But in this website, we have provided only one user role (subscriber role) to the form so that later admin can change the role as required.

Login

The ultimate member login feature allows users to effortlessly login from the front-end of the site and provides a forgot password link for the password rest if necessary. Along with the pre-defined fields, the form is fully customizable including: hiding the registration from the login, changing button text, applying custom styling and much more. The login form is easy to use with having anti-spam measures in it which comes with triple anti-spam protection which includes: a hidden honeypot field, time delay on submit button and WordPress nonces. (UltimateMember, 2018.)

User Roles

Ultimate member can be used create multiple user roles and each roles can be given its own permission and capabilities which makes the plugin perfect choice for site having more than one user like this website. As in this website, customers and employees are directed to two different landing page, the user restriction on pages is crucial and this plugin solves this problem with wase. Along with that the front page content is only accessible for the users which are not logged in to the website which was also done with the help of this plugin. User roles features in this website are as follows.

- The user role feature was used to determine which users can have the adminstrative permissions and general permissions.
- Profile access and page access according to user roles
- Redirecting to different landing page after user registraion, login, logout and account deletion.

(UltimateMember, 2018.)

Nav menu Visibility

As the pages in WordPress can be customized with specific user roles with the help of this plugin, displaying the content in menu was done accordingly. The following screenshot from the WordPress menu settings shows how it can be done.

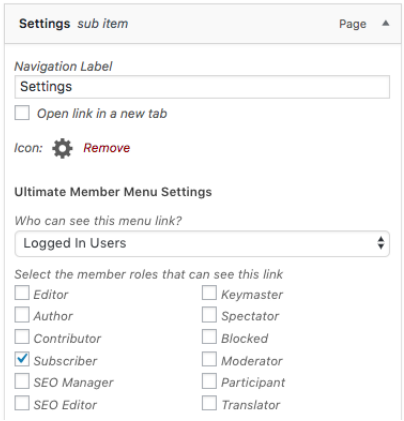

Figure 15: Nav menu visibility according to user roles

As illustrated in the figure 15, the nav menu named “Settings” is only shown to the logged in users and only for subscriber.

Custom Fields

Collecting a wide range of data from users using the form field is quite simple by using this plugin as it has a collection of pre-defined fields that come automatically installed with the plugin to create forms faster. Apart from that custom fields can be created from the field manager to make forms handy.

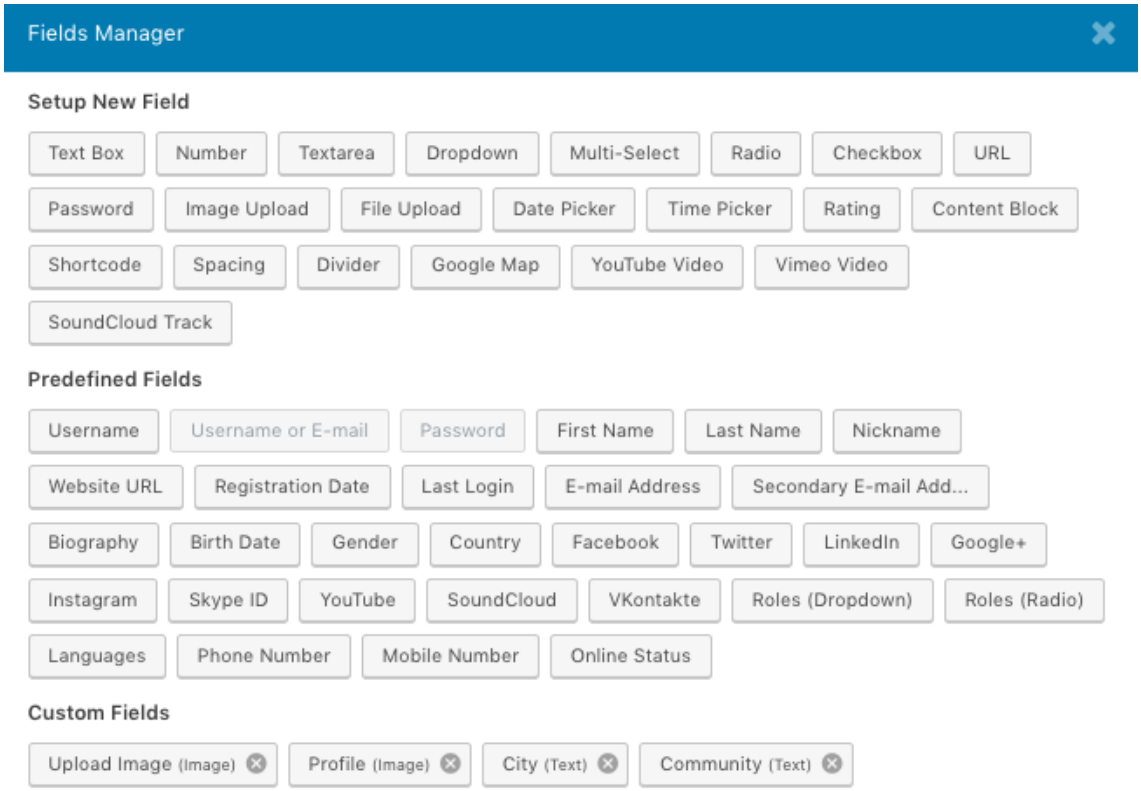

Figure 16: Screenshot of the field manager in Ultimate member plugin

As the figure 16 clearly illustrates the fields available in the plugin to make the form easily by either using the predefined fields or creating the custom fields. Using this field manager, it was quite easy to make fields as well as customize them and eventually saves lot of time.

6 Website Translation

In today's digital world, website is the main tool for any business or organization to run effectively and display the content for the end users. So, in order to reach to the target audience, it is almost compulsory for displaying the website in different languages. In simple words, Website translation is the process of modifying an existing website to make it accessible, usable and culturally suitable to a target audience (Language Scientific, 2018).

According to Forrester Research, visitors stay for twice as long if the website is in their own language and more than one third of all internet users are non-native English speakers (Language Scientific, 2018). It is therefore, for any organisation in order to reach large group of audience, website translation is essential.

6.1 Different types of Translation plugins

For attracting truly international audience, multilingual content is a must these days. Having the content in different content will help spike in visits and engagement in the website from different language speaking visitors. The best way to do this in WordPress website is to install a multilingual plugin that suits the need for the specific website. (Quarton, 2018.) Auto-translate and self-translate are two different types of multilingual plugins available which is explained briefly below.

Auto-translate

Auto-translate plugin are the plugins which rely on online translation services which help translate content into different language with just a click of a button, but this is not fully reliable as this automated language service has not fully cracked the nuances of modern language. (Quarton, 2018.)

Self-Translate

Self-Translate plugin are the plugins which require someone to translate the content of the website. Although this translation process is more time-consuming than the auto-translate, it produces the highest quality tranlations. With the help of these plugins, the contents can be

translated by the admins themselves or hire someone else to do the translations. (Quarton, 2018.)

Before choosing the plugin for the website, clear understanding of the translation plugin is crucial, and a research is needed whether a language pack is needed or not and whether the manual or automatic translation must be implemented. Apart from having multilingual website, if the translation plugin is not optimized for search engines, the marketing efforts will backfire, so the plugin must be SEO friendly. (isitwp, 2018.)

There are a lot of translation plugin available but according to isitwp (2018), the 11 best WordPress translation plugin are as follows.

1. WPML
2. Polylang
3. Multilingual Press
4. Xili-language
5. Google Language Translator
6. GTranslate
7. Loco Translate
8. Multilanguage
9. Goo Translate Widget
10. Google Website Translator
11. Lingotek

Among the above-mentioned translation plugin, WPML is choosen for this project as it completely fits in the necessity of the website.

6.2 Why WPML?

The WordPress Multilingual Plugin (WPML) is the most popular premium WordPress plugin for turning the site multilingual with over 600,000 sites running WPML. Translation can be done by the site admins themselves or can be done with a team of translators without any coding. WPML was choosen as it allows to translate all the post, pages, custom post types, theme strings, plugins and menus into any language with easy interface and great support along with easy to go documentation. (WPML, 2018.)

WPML Features

As being the most powerful premium plugin, it has many features which can be implemtned in the website as needed. In this website, some of the important features are being used to make the site multilingual which are described briefly below.

- One WordPress Installation – Multiple Languages

In a single WordPress installation, WPML makes it easy to run a multilingual website by choosing language for the site and translating the content (WPML, 2018). The simple WPML setup is needed in the beginning before translation into another langague.

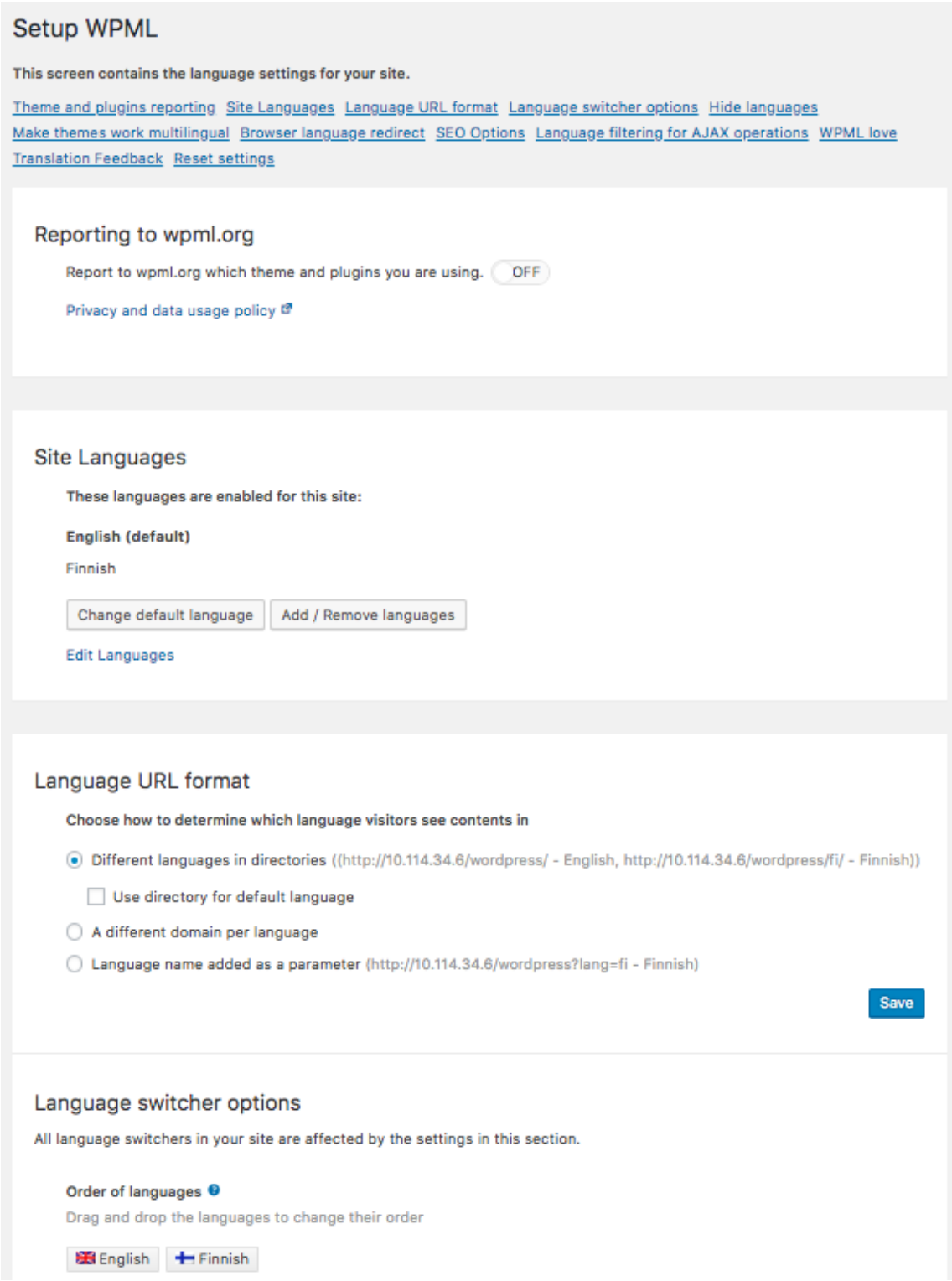

Figure 17: WPML Setup page

The figure 17 illustrates the site's multilingual settings on the WPML Setup page. At first the site's default language was choosen as English and Finnish language was added as the

second language. After that, the language URL format was choosen for both languages and the language order was determined.

After adjusting the setting for the site's multilingual settings, the WordPress interface allows to translate the content easily. The pages, posts and categories can be conveniently translated to another language.

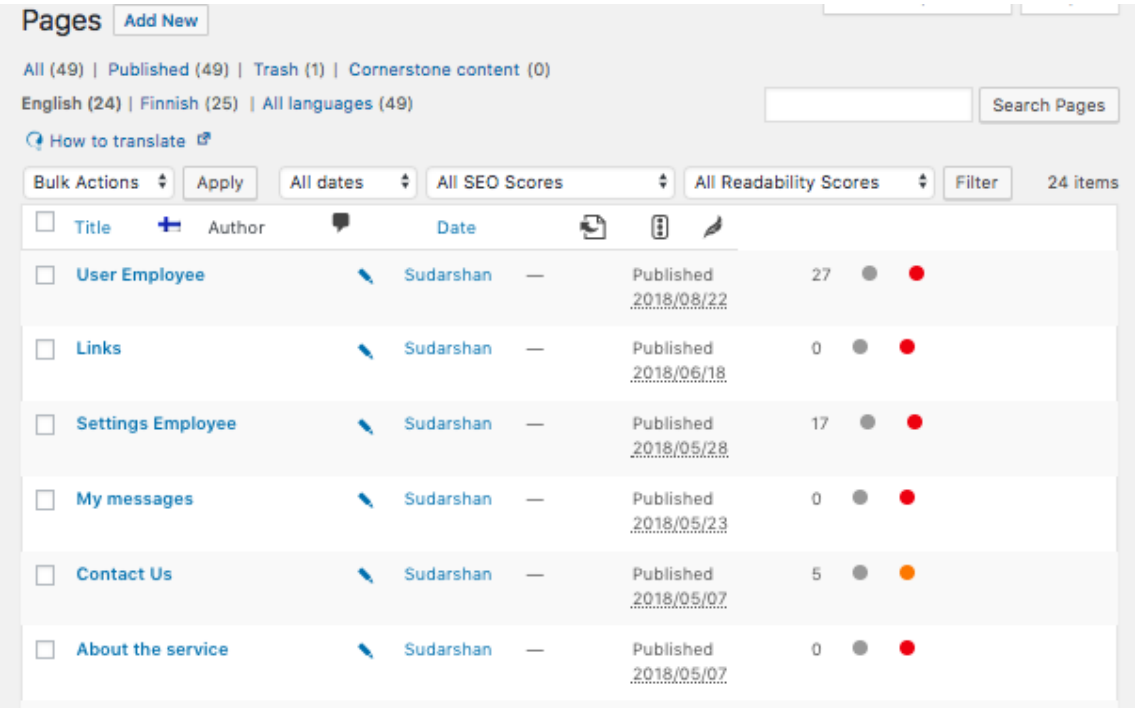

Figure 18: List of pages

As figure 18 illustrates the list of pages which can be easily translated after the WPML settings as shown in figure 17.

❖ Powerful Translation Management

WPML has the state-of-the-art translation management where the site's admin can be as translator or can assign to specific users who can access the specific translation jobs assigned to them (WPML, 2018). Apart from this, WPML's powerful translation management can be connected to translation service of choice.

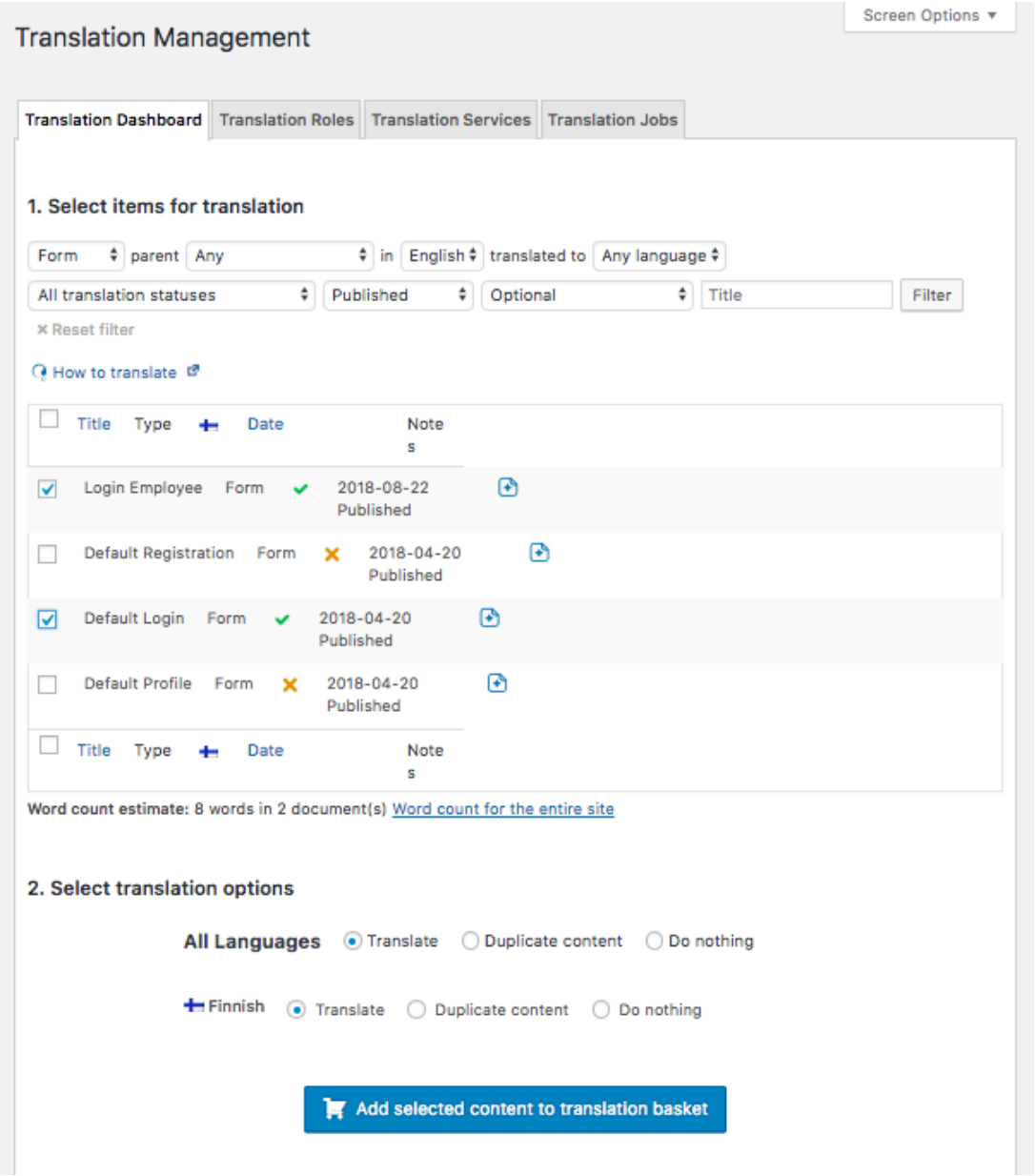

Figure 19: Screenshot of Translation Management with translation dashboard overview

The figure 19 clearly illustrates the translation dashboard where the form is selected for translation and the translation options is choosen as Finnish which is only the second language of the site.

❖ Translation for Theme and Plugin Texts

WPML is handy in translating the texts of theme and different plugins. It frees the hassle of editing PO files and uploading MO files as text can be translated directly from the String Translation interface and this setting can be done in theme and plugins localization.

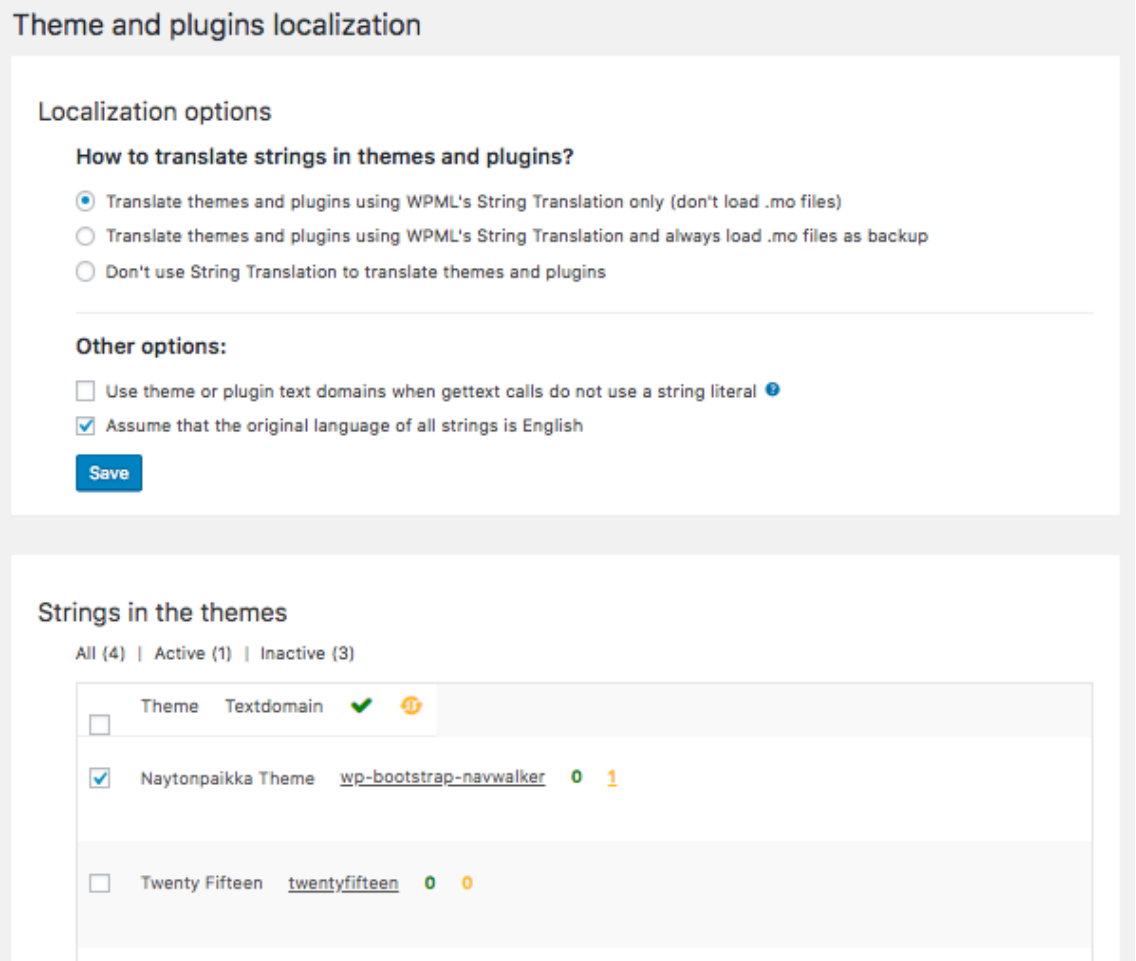

Figure 20: Settings for theme and plugin localization

Figure 20 shows that in this site, the themes and plugins texts are translated using WPML's String Translation only without loading the .mo files. Along with this the option for original language of all the strings being English was choosen so that only the English strings will appear to translate.

The string translation is one of the main features where all the strings appear of the theme and plugins and can be translated by the site admins without any hassle. Along with this the specific strings within certain domains can be selected.

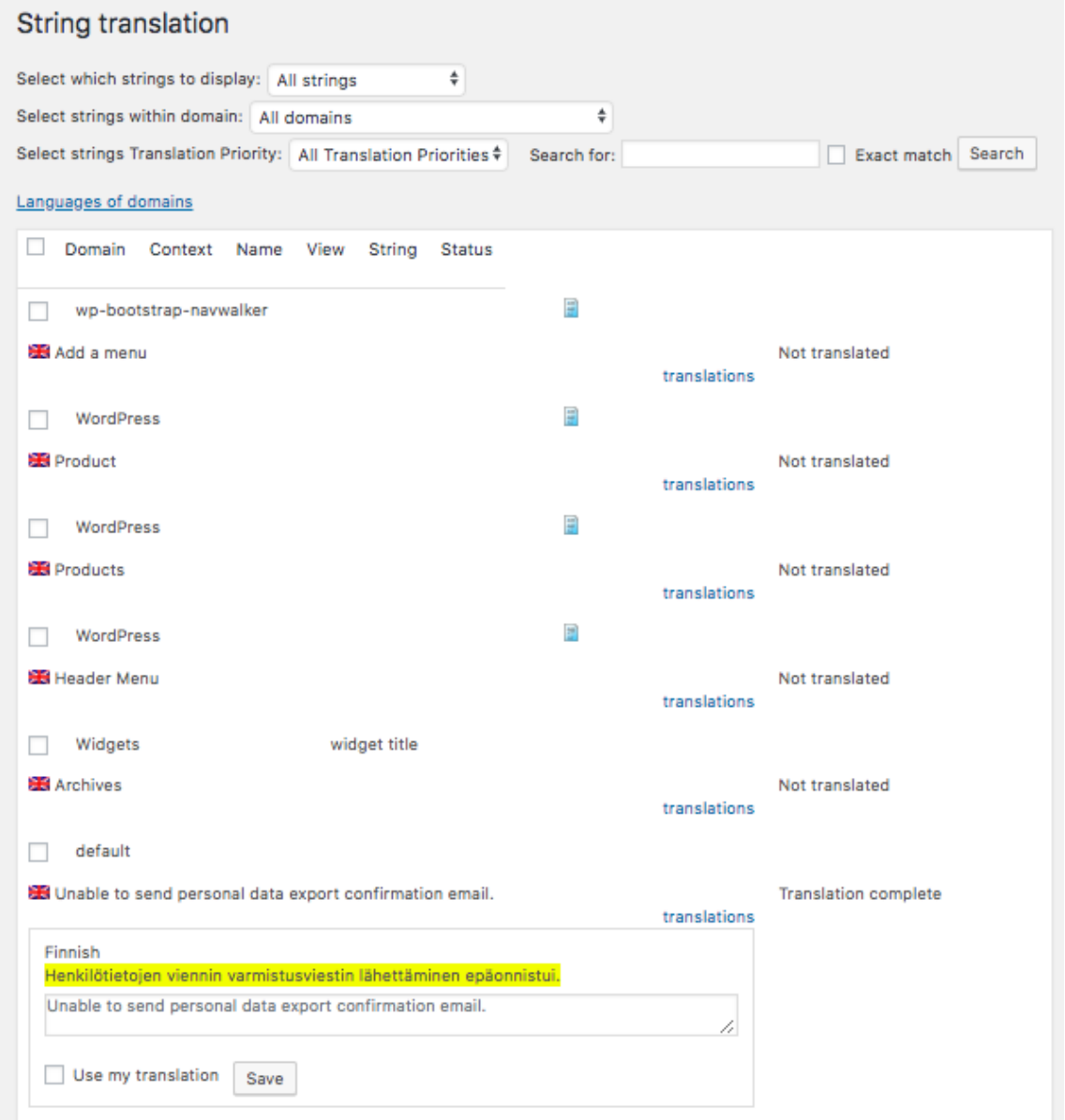

Figure 21: Translation texts coming from different domains

The figure 21 illustrates the strings for translation. The strings from all the domains are selected and it shows the translated as well as not translated strings. The yellow highlighted strings show the translated text.

❖ Works with Most WordPress Plugins

Many plugins are compatible with WPML, so it is easy to translate the strings of plugins to create multilingual site without any hassle.

7 Result and Discussion

The new version of the NäytönPaikka website was created in reference to the previous website and all the necessary steps are completed. Currently, the website is in the local Metropolia server for development as the final version of the website is not yet completed. The skeleton of the website with all the CMS, plugins, custom theme, user management tools and translation tools are created as expected which makes the further development of the website convenient.

Although WordPress has become the easy and convenient tool for developing websites, a few problems were faced during the project. Due to the php versions being different between WordPress and plugin, the whole website needs to be transferred to new server and was time consuming but new techniques were learnt during this process too. Along with this, as the server being in Centos, removing MySQL properly and installing MariaDB was also challenging as in the initial phase there was not any indication of malfunction.

WordPress theme was created from scratch without any prior knowledge so being familiar with the working environment was challenging in the beginning but got acquainted after going through documentation and related projects. Along with this, choosing the right user management tool was challenging and time consuming as it needs to fit the site's exact need and easy to use. Choosing right translation plugin was also challenging as it must be compatible with other plugins and testing needs to be done for all the plugin selected.

Overall, the project helped to develop skills in different fields and get familiarized with a new CMS tool. The help from my fellow colleague and supervisor helped the project to be successful and learned that by using CMS, a well manageable website can be developed from scratch with the right selection of tools. Hard coding can be crucial in some projects but from this project it is well known that with the right selection of CMS, plugins and custom development, any projects can be done without hard coding.

8 Conclusion

The goal of this thesis was to provide the case company with a fully functioning new website with the implementation of user managemnet tools so that it speeds up the creation of new accounts and reduces the administrator's current workload. Along with that, making the website fully responsive and implementating a modern language version to translate the page into different languages.

For achieving these goals, a custom theme was created in order to make contents precise and easier to implement into it. Ultimate member plugin was used for the user managemnet tool which is simple to use and creates new accounts reducing the administrator's workload. Boostrap was used for the responsiveness of the website and WPML plugin was used for making the translation of the website in different languages because of its compatibility with various plugins.

The key to have a successful website needs good content and continuos work on it to make the improvements. WordPress has good solutions for developing and maintaining websites. With custom theme and a strong user management tool along with translation available in differnet langauges, this website has good web development tools to excel in the future.

References

4MDesigners, 2014. *4MDesigners.* [Online]
Available at: http://www.4mdesigners.com/articles/advantages-and-disadvantage-of-cms/
[Accessed 26 September 2018].

BootStrap, 2018. *BootStrap.* [Online]
Available at: https://getbootstrap.com/docs/4.0/layout/grid/
[Accessed 11 October 2018].

Bouchefra, A., 2018. *sitepoint.* [Online]
Available at: https://www.sitepoint.com/bootstrap-wordpress-theme-integration/
[Accessed 11 October 2018].

BuiltWith, 2018. *builtwith.* [Online]
Available at: https://trends.builtwith.com/cms
[Accessed 27 September 2018].

cre8d, 2016. *cre8d.* [Online]
Available at: https://www.cre8d-design.com/2016/04/7-reasons-get-a-custom-wordpress-theme/
[Accessed 4 October 2018].

DeanInfotech, 2017. *Dean Infotech.* [Online]
Available at: https://www.deaninfotech.com/blog/cms-vs-hand-coding-right-choice-build-website/
[Accessed 27 September 2018].

Elgameel, A., 2018. *WP-ME.COM.* [Online]
Available at: https://wp-me.com/wordpress-best-content-management-system/
[Accessed 27 September 2018].

isitwp, 2018. *isitwp.* [Online]
Available at: https://www.isitwp.com/best-wordpress-translation-plugins-compared/
[Accessed 22 October 2018].

Laminack, I. B., 2008. *Brent Laminack's Personal site.* [Online]
Available at: http://www.laminack.com/index.php/technology/11-a-brief-history-of-content-management-systems
[Accessed 20 September 2018].

Language Scientific, 2018. *LANGUAGE SCIENTIFIC.* [Online]
Available at: https://www.languagescientific.com/website-localization-and-website-translation-what-is-involved/
[Accessed 11 Oct 2018].

Metropolia, 2018. *MOTIVES: New Methods for Young People in Future Work.* [Online]
Available at: https://www.metropolia.fi/tutkimus-kehittaminen-ja-innovaatiot/hankkeet/motiivi/
[Accessed 19 September 2018].

NäytönPaikka, 2018. *What is NäytönPaikka service ?.* [Online]
Available at: https://www.naytonpaikka.fi/tietoapalvelusta/
[Accessed 20 September 2018].

NIBUSINESSINFO, 2018. *NIBUSINESSINFO.CO.UK.* [Online]
Available at: https://www.nibusinessinfo.co.uk/content/different-types-content-management-systems
[Accessed 25 September 2018].

Pickard-Whitehead, G., 2018. *Small Business TRENDS.* [Online]
Available at: https://smallbiztrends.com/2018/04/what-is-a-content-management-system.html
[Accessed 20 September 2018].

Quarton, S., 2018. *WinningWP.* [Online]
Available at: https://winningwp.com/best-multilingual-translation-plugins-for-wordpress/
[Accessed 22 October 2018].

Ricotta, M., 2018. *BLUE FOUNTAIN MEDIA.* [Online]
Available at: https://www.bluefountainmedia.com/blog/what-is-a-responsive-web-design
[Accessed 11 October 2018].

Silkalns, A., 2018. *colorlib.* [Online]
Available at: https://colorlib.com/wp/wordpress-user-management-plugins/
[Accessed 11 October 2018].

UltimateMember, 2018. *UltimateMember.* [Online]
Available at: https://ultimatemember.com/features/
[Accessed 12 October 2018].

WORDPRESS.ORG, 2018. *WORDPRESS.ORG.* [Online]
Available at: https://codex.wordpress.org/Writing_a_Plugin
[Accessed 25 October 2018].

WordPress, 2018. *WORDPRESS.ORG.* [Online]
Available at: https://codex.wordpress.org/Using_Themes
[Accessed 4 October 2018].

WordPress, 2018. *WORDPRESS.ORG.* [Online]
Available at: https://developer.wordpress.org/themes/basics/template-hierarchy/
[Accessed 4 October 2018].

wpbeginner, 2015. *wpbeginner.* [Online]
Available at: https://www.wpbeginner.com/wp-themes/how-to-create-category-templates-in-wordpress/
[Accessed 10 October 2018].

WPML, 2018. *WPML.* [Online]
Available at: https://wpml.org/home/about-us/
[Accessed 23 October 2018].

WPML, 2018. *WPML.* [Online]
Available at: https://wpml.org/features/
[Accessed 23 October 2018].

Code of index.php file

```
<?php get_header(); ?>
<?php get_sidebar('menu'); ?>

 <div class="col-sm-8 np-main">

   <?php
   if ( have_posts() ) {
     while ( have_posts() ) : the_post();
       ?>
       <div class="np-post">
         <h2 class="np-post-title"><?php //the_title(); ?></h2>
         <p class="np-post-meta"><?php //the_date(); ?> <?php //by?> <?php //the_author();
?></p>
         <?php the_content(); ?>
       </div><!-- /.np-post -->
     <?php
     endwhile;
   }
   ?>

 </div><!-- /.np-main -->

<?php get_footer(); ?>
```

Code of functions.php file of nptheme

```
<?php
//Style enqueue
function nptheme_enqueue_styles()
{
    wp_register_style('bootstrap', get_template_directory_uri() . '/bootstrap/css/boot-
strap.min.css');
    $dependencies = array('bootstrap');
    wp_enqueue_style('nptheme-style', get_stylesheet_uri(), $dependencies);
}

//Scripts enqueue
function nptheme_enqueue_scripts()
{
    $dependencies = array('jquery');
    wp_enqueue_script('bootstrap', get_template_directory_uri() . '/bootstrap/js/boot-
strap.min.js', $dependencies, '', true);
    wp_enqueue_script('script', get_template_directory_uri() . '/js/nptheme.js', array('jquery'),
1.1, true);
}

add_action('wp_enqueue_scripts', 'nptheme_enqueue_styles');
add_action('wp_enqueue_scripts', 'nptheme_enqueue_scripts');

//Enqueueing the fontawesome
function wmpudev_enqueue_icon_stylesheet()
{
    wp_register_style('fontawesome', 'http:////maxcdn.bootstrapcdn.com/font-awe-
some/4.3.0/css/font-awesome.min.css');
    wp_enqueue_style('fontawesome');
}

add_action('wp_enqueue_scripts', 'wmpudev_enqueue_icon_stylesheet');

//Setting title-tag and post-thumbnails
function nptheme_wp_setup()
{
    add_theme_support('title-tag');
    add_theme_support('post-thumbnails');
```

```
}

add_action('after_setup_theme', 'nptheme_wp_setup');

//Header registration
function nptheme_register_header_menu()
{
    register_nav_menu('header-menu', __('Header Menu'));
}

add_action('init', 'nptheme_register_header_menu');

//Sidebar registration
function nptheme_widgets_init()
{
    register_sidebar(array(
        'name' => 'Footer - Copyright Text',
        'id' => 'footer-copyright-text',
        'before_widget' => '<div class="footer_copyright_text">',
        'after_widget' => '</div>',
        'before_title' => '<h4>',
        'after_title' => '</h4>',
    ));

    register_sidebar(array(
        'name' => 'Sidebar - Inset',
        'id' => 'sidebar-1',
        'before_widget' => '<div class="sidebar-module sidebar-module-inset">',
        'after_widget' => '</div>',
        'before_title' => '<h4>',
        'after_title' => '</h4>',
    ));

    register_sidebar(array(
        'name' => 'Sidebar - Default',
        'id' => 'sidebar-2',
        'before_widget' => '<div class="sidebar-module sidebar-module-default">',
        'after_widget' => '</div>',
        'before_title' => '<h4>',
        'after_title' => '</h4>',
```

```
    ));

    register_sidebar(array(
        'name' => 'Sidebar - Employee',
        'id' => 'sidebar-3',
        'before_widget' => '<div class="sidebar-module sidebar-module-employee">',
        'after_widget' => '</div>',
        'before_title' => '<h4>',
        'after_title' => '</h4>',
    ));
}

add_action('widgets_init', 'nptheme_widgets_init');

require_once('wp-bootstrap-navwalker.php');

// Bootstrap navigation
function bootstrap_nav()
{
    wp_nav_menu(array(
            'theme_location' => 'header-menu',
            'depth' => 2,
            'container' => 'false',
            'menu_class' => 'nav navbar-nav',
            'fallback_cb' => 'wp_bootstrap_navwalker::fallback',
            'walker' => new wp_bootstrap_navwalker())
    );
}

// Makes <br> tag working in excerpts so that line breaks work
function custom_trim_excerpt($text)
{
    if ('' == $text) {
        // Add the original wp_trim text code here if auto-generated excerpt is needed
    } else {
        $text = apply_filters('the_content', $text);
        $text = strip_tags($text, '<br>');
    }
    return $text;
}
```

```
remove_filter('get_the_excerpt', 'wp_trim_excerpt');
add_filter('get_the_excerpt', 'custom_trim_excerpt');


@ini_set('upload_max_size', '64M');
@ini_set('post_max_size', '64M');
@ini_set('max_execution_time', '300');

?>
```

Code of category.php file to display the categories

```
<?php get_header(); ?>
<div class="content-row">
  <main>
    <h2><?php echo get_queried_object()->name; ?></h2>

    <?php
    $id = get_queried_object()->term_id;
    $artikkelit = get_posts( array('category' => $id, 'numberposts' => 999));

    foreach ($artikkelit as $artikkeli):
      ?>
      <article>
        <a href="<?php echo get_permalink( $artikkeli->ID ); ?>">
          <?php echo get_the_post_thumbnail( $artikkeli->ID, 'thumbnail'); ?>
          <h4><?php echo $artikkeli->post_title; ?></h4>
        </a>
      </article>
    <?php endforeach; ?>
  </main>
</div>
<?php get_footer(); ?>
```

Code of category-my-diary.php to display post related to this category

```
<?php get_header(); ?>
<?php get_sidebar('menu'); ?>

 <section class="page-content col-sm-8">
  <h2 class="h2-categorypage"><?php echo get_queried_object()->name; ?></h2>

  <?php
  $id = get_queried_object()->term_id;
  $cat_id = get_cat_ID("elamantilanteeni");
  $artikkelit = get_posts( array('category' => $id, 'numberposts' => 999));
  get_cat_name($cat_id);
  foreach ($artikkelit as $artikkeli):
    ?>
    <article class="category-categorypage">
      <div class="article-content">
        <a href="<?php echo get_permalink( $artikkeli->ID ); ?>">
          <?php echo get_the_post_thumbnail( $artikkeli->ID, 'thumbnail'); ?>
          <h4 class="h4-categorypage"><?php echo $artikkeli->post_title; ?></h4>
        </a>
      </div>
    </article>
  <?php endforeach; ?>
 </section>
<?php get_footer(); ?>
```

Code of page.php file to display the pages

```
<?php
/**
 * The template for displaying pages
 *
 * This is the template that displays all pages by default.
 * Please note that this is the WordPress construct of pages and that
 * other "pages" on your WordPress site will use a different template.
 *
 * @package WordPress
 * @subpackage nptheme
 */
?>
<?php get_header(); ?>
<?php get_sidebar('menu');?>
<div class="col-sm-8 np-main">

  <?php
  if ( have_posts() ) {
    while ( have_posts() ) : the_post();
      ?>
          <div class="np-post">
             <h2 class="np-post-title"><?php //the_title(); ?></h2>
             <p class="np-post-meta"><?php //the_date(); ?> <?php //by?> <?php //the_au-
thor(); ?></p>
        <?php the_content(); ?>
          </div><!-- /.np-post -->
    <?php
    endwhile;
  }
  ?>
</div><!-- /.np-main -->
<?php get_footer(); ?>
```

Code of FrontPageTemplate.php file

```
<?php
/**
 * Template Name: FrontPageTemplate
 *
 * This is the template that displays all pages by default.
 * Please note that this is the WordPress construct of pages and that
 * other "pages" on your WordPress site will use a different template.
 *
 * @package WordPress
 * @subpackage nptheme
 */

?>
<?php get_header('front'); ?>
<div class="col-sm-8 np-main">

 <?php
 if ( have_posts() ) {
   while ( have_posts() ) : the_post();
     ?>
     <div class="np-post">
       <h2 class="np-post-title"><?php //the_title(); ?></h2>
       <p class="np-post-meta"><?php //the_date(); ?> <?php //by?> <?php //the_author();
?></p>
       <?php the_content(); ?>
     </div><!-- /.np-post -->
   <?php
   endwhile;
 }
 ?>

</div><!-- /.np-main -->
<?php get_sidebar()?>
<?php get_footer(); ?>
```

Code of EmployeePageTemplate.php file

```
<?php
/**
 * Template Name: EmployeeTemplate
 *
 * This is the template that displays all pages by default.
 * Please note that this is the WordPress construct of pages and that
 * other "pages" on your WordPress site will use a different template.
 *
 * @package WordPress
 * @subpackage nptheme
 */

?>
<?php get_header('employee'); ?>
<?php get_sidebar('employee');?>
<div class="col-sm-9 np-main">

  <?php
  if ( have_posts() ) {
    while ( have_posts() ) : the_post();
      ?>
          <div class="np-post">
            <h2 class="np-post-title"><?php //the_title(); ?></h2>
            <p class="np-post-meta"><?php //the_date(); ?> <?php //by?> <?php //the_au-
thor(); ?></p>
        <?php the_content(); ?>
          </div><!-- /.np-post -->
    <?php
    endwhile;
  }
  ?>
</div><!-- /.np-main -->
<?php get_footer(); ?>
```

Code of page-mysection.php to display the contents of mysection page

```
<?php get_header(); ?>
<?php get_sidebar('menu');?>
<div class="col-sm-8 np-main text-center container-fluid">

   <?php
   $args = array('child_of' => 32);
   $categories = get_categories( $args );


   $i = 1;
   $k = 1;
   foreach($categories as $category) :

      if($i === 1) {
         ?> <div class="row">
      <?php }
      if($k === 2 && $i === 2){
         ?>
         <div class="col-sm-4 section-category">
            <!-- The commented line of code below is for getting an image by id from media
library -->
            <a href="#"><?php  echo wp_get_attachment_image( 420, array('160', '160'), '',
array( 'class' => '' ));  ?></a>
            <?php //echo get_avatar( 3, 160 ); ?>
         </div>
         <?php
         $i++;
      }
      ?>

      <div class="col-sm-4 section-category">
         <a href="<?php echo get_category_link($category->term_id); ?>">
            <img src="<?php echo z_taxonomy_image_url($category->term_id); ?>" /><br>
            <?php echo $category->cat_name; ?></a>
      </div>
      <?php if($i === 3){
      $k++;   ?>
      </div>
```

```
        <?php
        $i = 0;
    }
        $i++;
    endforeach;

    ?>
</div>
 
<?php get_footer(); ?>
```

Code of header.php file to display the header section of the site

```
<!DOCTYPE html>
   <html <?php language_attributes(); ?>>
   <head>
      <meta charset="<?php bloginfo('charset'); ?>">
      <meta http-equiv="X-UA-Compatible" content="IE=edge">
      <meta name="viewport" content="width=device-width, initial-scale=1">
      <link href="<?php echo get_stylesheet_uri(); ?>" rel="stylesheet">
      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awe-
some/4.7.0/css/font-awesome.min.css">

      <?php wp_head(); ?>
   </head>
   <body <?php body_class(); ?>>
    <div class="container">
         <nav class="navbar navbar-default">
            <div class="container-fluid">
         <div class="navbar-header">

            <button type="button" class="navbar-toggle" data-toggle="collapse"
                  data-target="#myNavbar">
               <span class="icon-bar"></span>
               <span class="icon-bar"></span>
               <span class="icon-bar"></span>
            </button>
           <a class="navbar-brand" href="http://10.114.34.9/wordpress/mysection/"><img
src="<?php echo get_template_directory_uri();
            ?>/img/npimage-header.png"></a>
         </div>

         <div class="collapse navbar-collapse" id="myNavbar">
            <ul class="nav navbar-nav navbar right">
            <?php bootstrap_nav();?>
         </ul>
         </div>
      </div>
   </nav>

      <div class="row">
```

```
    <div class="breadcrumbholder">
    <?php if ( function_exists('yoast_breadcrumb') )
{yoast_breadcrumb('<p id="breadcrumbs">','</p>');} ?>
</div>
```

Code of style.css used for styling the website

```
   /*
Theme Name: Naytonpaikka Theme
Theme URI: https://www.naytonpaikka.fi
Author: Sudarshan
Author URI: https://www.naytonpaikka.fi
Description: Theme for Naytonpaikka
Version: 1.0
Text Domain: nptheme
*/

/*
 * Globals
 */

 html,
 body {
  font-family: Georgia, "Times New Roman", Times, sans-serif;
  color: #333;
  background-color: #cfdfe8;
  height: 100%;
}

.container {
  background-color: white;
  width: 80%;
  display: flex;
  flex-direction: column;
  min-height: 100%;
}

h1 h2 h3 h4 h5 h6,
.h1 .h2 .h3 .h4 .h5 .h6 {
  margin-top: 0;
  font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
  font-weight: normal;
  color: red;
}
```

```
/*
 * Override Bootstrap's default container.
 *Setting up different widths for the media breakpoints in Bootstrap.
 */

 @media (max-width: 600px) {
  .fa {
    font-size: 20px;
    padding: 10px;
  }
  .navbar-brand {
    margin-left: 0px;
  }
}

@media (max-width: 600px) {
  .container {
    width: 100%;
  }
  .carousel-caption {
    display: none;
  }
  .breadcrumbholder,
  .scroll-up {
    display: none;
  }
  #topbanner {
    display: none;
  }
  #icon {
    max-width: 150px;
    color: black;
  }
  #myCarousel {
    display: none;
  }
  h2 {
    font-size: 1.5em;
  }
}
```

```
#topbanner {
  background-color: gray;
  height: 40px;
}

.sininen {
  background-color: lightblue;
}

.search-block form {
  float: right;
}

/*
 * Navbar
 */

 .navbar {
  margin-bottom: 0;
  border-radius: 0;
  padding: 1% 0;
  font-size: 1.1em;
  border: 0;
  background-color: mintcream;
}

.navbar-brand {
  float: left;
  min-height: 55px;
  padding: 5px 15px 5px;
}

.navbar-nav li a {
  color: black;
  font-size: medium;
}

.myNavbar {
  float: right;
```

```
}

.translate {
  text-align: center;
  padding-top: 8px;
  list-style: none;
}

.breadcrumbholder {
  margin-left: 26%;
  font-size: medium;
}

ul{
  list-style: none;
}

/*
 * np name and description
 */

 .np-header {
  padding-top: 7%;
  padding-bottom: 10px;
}

.np-title {
  margin-top: 20px;
  margin-bottom: 0;
  font-size: 50px;
  font-weight: normal;
}

.np-description {
  font-size: 20px;
}


/*
 * Main column and sidebar layout
```

```
 */

 .sidebar-module-default {
  /*background-color: none;*/
}

.np-main {
  font-size: 15px;
  line-height: 1.2;
}

/* Sidebar modules for boxing content */

.sidebar-module {
  padding: 15px;
}

.np-sidebar-front {
  border-left: lightsteelblue 1px solid;
}

.np-sidebar-menu li,
.np-sidebar-employee li {
  border-top: lightsteelblue 1px solid;
}

.np-sidebar-menu li:last-child,
.np-sidebar-employee li:last-child {
  border-bottom: lightsteelblue 1px solid;
}

/*
 * np posts
 */

 .np-post {
  margin-bottom: 60px;
}

.np-post-title {
```

```
  margin-bottom: 5px;
  font-size: 30px;
}

.np-post-meta {
  margin-bottom: 20px;
  color: #999;
}

/*
 * Footer
 */

  .np-footer {
    flex: 0 0 60px;
    /*or just height:50px;*/
    margin-top: auto;
  }

  hr {
    display: block;
    height: 1px;
    border: 0;
    border-top: 1px solid #ccc;
    margin: 1em 0;
    padding: 0;
  }

 /*** npthemes - custom styling ***/

 ul.np-nav {
  list-style: none;
  display: inline;
}

/* Nav links */

.menu-item a {
  position: relative;
  display: inline-block;
```

```
  padding: 10px;
  font-weight: 500;
  color: black;
}

.menu-item a:hover,
.menu-item a:focus {
  background-color: lightgray;
  text-decoration: none;
  width: 100%;
}

/* Active state gets a caret at the bottom */

.menu-item.current-menu-item a {
  background-color: lightgray;
  width: 100%;
}

/* Carousal */

#myCarousel {
  margin-bottom: 1%;
}

.carousel-caption {
  top: 50%;
  transform: translateY(-50%);
  text-transform: uppercase;
}

.btn {
  font-size: 18px;
  color: #FFF;
  padding: 12px 22px;
  background: #5E4485;
  border: 2px solid #FFF;
}

.carousel-inner>.item>img,
```

```
.carousel-inner>.item>source,
.carousel-inner>.item>a>img {
  display: block;
  height: 400px;
  max-width: 100%;
  line-height: 1;
  margin: auto;
  width: 100%;
}

  /* Sidebar */
  .sidebar-module ul {
    list-style: none;
    padding-left: 0;
  }

  li a {
    display: block;
    padding: 8px 16px;
    text-decoration: none;
    font-size: 16px;
    line-height: 0.8em;
  }

  #logodiv {
    padding-top: 25px;
  }

  .fa {
    padding: 12px;
    font-size: 25px;
    color: grey;
  }

  .fa:hover {
    color: black;
    text-decoration: none;
  }

  @media (max-width: 600px) {
```

```
  .fa {
    font-size: 20px;
    padding: 10px;
  }
}

@media (max-width: 600px) {
  .carousel-caption {
    display: none;
  }
  #icon {
    max-width: 150px;
  }
  h2 {
    font-size: 1.5em;
  }
}

.section-category {
  padding-bottom: 5%;
}

.category-categorypage {
  border-top: solid #6e9ab2 1px;
}

.category-categorypage:hover {
  background-color: lightgrey;
}

.h4-categorypage {
  display: inline-block;
}

.category-categorypageemployee {
  border-top: solid #6e9ab2 1px;
}

.category-categorypageemployee:hover {
  background-color: lightgrey;
```

```
}

.category-categorypageemployee:last-child {
  border-bottom: solid #6e9ab2 1px;
}

.h2-categorypage {
  padding-bottom: 5%;
}

.article-content {
  padding-top: 1em;
  padding-bottom: 1em;
  border: none;
}

.forms-to-share {
  display: none;
}
```