

Requirements Document

Version 1.1

Martin-Springer Capstone Team

‘The Untitled Developers’

Herbie Duah

John Loudon

Michael Ortega

Luke Sanchez

Northern Arizona University

Fall 2015

Table of Contents

<i>1.0 - Introduction.....</i>	<i>3</i>
<i>2.0 - Functional Requirements.....</i>	<i>5</i>
2.1 - Historical Constancy	
2.2 - Front End Interactions	
<i>3.0 - Environmental Requirements.....</i>	<i>9</i>
3.1 - Server Hosting	
3.2 - Content Management System	
<i>4.0 - Non-functional Requirements.....</i>	<i>12</i>
4.1 - Performance	
4.2 - Maintainability	
<i>5.0 - Potential Risks.....</i>	<i>14</i>
5.1 - WordPress Plugins	
5.2 - Client changes to code	
5.3 - Map and Timeline crashing	
5.4 - Frameworks crashing	
5.5 - Server changes	
<i>6.0 - Project plan.....</i>	<i>15</i>
<i>7.0 - Requirements Appendix.....</i>	<i>17</i>

1. Introduction

The Untitled Developers are proud to present the first release of our capstone experience requirements document of Fall 2015. This abstract is intended to give you a thorough understanding of the system that is being developed by our team. This document is subject to change as we discover and reevaluate our requirements.

System Domain

The Martin-Springer Institute founded at N.A.U. attends to the experiences of the Holocaust in order to relate them to today's concerns and crises. With the help of a multi-disciplinary team of scholars and survivors of the Holocaust, a website was built as a supplement to a physical exhibit that travels from city to city as a monument to history.

Problem Statement

The site in its current structure is very primitive in terms of web technologies. Speed tests run on the website are extremely slow, there are redundant and distracting navigation links and the page will not adjust its user interface according to the device that is being used to view it. From a computer science perspective there are many changes that could be made to make the site more accessible to users and optimized for hardware speed. The system must also be flexible enough to allow the current director of the Martin-Springer Institute to update the site infrequently (once or twice a year).

Solution Statement

We are renovating the underlying architecture of this website to make it more modern, robust and usable. We will be ripping the content directly from the old website and will build an entirely new web system from the ground up. Our new design will make use of an array of web technologies discussed in our technology

feasibility document to create a more accessible and interactive platform that will be used to teach the public lessons learned from the history of the Holocaust in the context of the Bedzin Ghetto.

2. *Functional Requirements*

2.1 - Historical Constancy

The purpose of the website is to be a supplement to the history and stories of the exhibit *Through the Eyes of Youth: Life and Death in the Bedzin Ghetto*. The website will include all stories, terms and definitions currently found on the site¹. It will also include new content.

2.2 - Front End User Interactions

2.2.1 - Users will be able to access **Exhibit**, **History**, and **Resource** website directories.

2.2.2 - Users will be able to view an interactive timeline of World War II

2.2.3 - Users will be able to view three unique, interactive maps detailing events surrounding the town of Bedzin

2.2.4 - Users will be able to access educational materials related to the exhibit

2.2.5 - Users will be able to access Martin-Springer Institute contact information

2.3 - User Interface Specifications

2.3.1 - Users from around the world will be able to access and website through modern web browsers

(Chrome, Firefox, Safari, Internet Explorer and Microsoft Edge)

2.3.2 - The website interface will be responsive to mobile devices and reformat itself accordingly

2.3.3 - The website will be feature minimalistic like most modern websites

2.3.4 - The website will follow good placement concepts so the objects are aligned currently so it can appeal to the website.

2.3.5- The website will feature intuitive animations to sort of bring life to it when the user is navigating around it

¹ Bedzin Exhibit. Retrieved September 9, 2015 from http://bedzinexhibit.org/exhibit_home/

The purpose of this project is renovate website per our clients' request. Overall the client wants the website to be modern, user-friendly, editable, and sturdy. To make the site modern we are going to explore different web development technologies so the website has modern look and feel. Making user-friendly site requires as to study on user experience and do some real life user testing before the site is released. Our clients' require certain parts of the website to be editable by them so we have to look into technologies that will allow the website to be easily editable by them so they don't have to use go through code in order to make changes. In order to make the website sturdy, we will have to conduct performance testing and figure what tests we need to do in different cases. The system we are creating for our client should be modern, user-friendly, editable and sturdy so we are going describe how our system should be in order to satisfy the aforementioned requirements.

Modern

The questions we need to ask ourselves are what is a modern website and how can we make our system a modern website? Do people determine how modern a website is by look of it? the feel of it? the underlying technology? or a mix of all of them? Fortunately our group consists of members that go online often to form a conclusion of what a modern site should look like. With a consensus between all of us we are going to create website that has a modern look and feel using modern technologies.

Look and Feel

Most modern websites look and feel minimalistic. Minimalistic goes along with the saying "Less is more." When it comes to designing the website, we need to only place the objects we need to place in order to avoid confusion. It's also easier to find mistakes in a clean, simple, minimal design.

Placement is also key when it comes to the look and feel of a website. Most websites in the past had some web objects placed all around the site. Modern websites almost have a very appealing placement of objects where all the objects are aligned nicely and perfectly parallel to another object and so on. When the website is created it is important, it should follow good placement so the objects are aligned currently so it can appeal to the website.

Smartphones and tablets have grown widely popular this day and age so we should assume users are not only going to visit the website from laptops and desktops but also smartphones and tablets. The look and feel of the website should not be different because of the size the screen. The look and feel should not be so different the user have to question if they are on the right website.

Animations are getting popular when visiting websites today. When going through websites today, one may notice objects sliding in, sliding out, fading, disappearing as you are moving around the page to make the page have a sense of life. Many of the objects the user interacts with is going to have animations and also assure the user they did indeed interact with the object.

Modern Technologies

There are more and more evolving web technologies every day and most of them are open source. With technologies like Git, it is very easy to download these technologies and implement them on your web application. They even offer examples of how those certain technologies work on their website so you can see it first hand. It is imperative we use modern technologies to make the site more current with the times.

User Experience

User experience is key when it comes to building a system with many different type of users interacting with it. The site is going to be built in a way that when users interact with it, it is going to lead to what they expected to go to. The site is going to

under testing by different users to see how they navigate around the site. The site should be usable from ages from ages 13 and up.

Editable

Our client require that the website is editable by him and his colleagues. We chose WordPress as a group because it is the most popular Content Management System. Most of the editable content on WordPress² is done via plug-ins. In order to make satisfy our clients request to edit certain parts of the site, our client should be able to go into WordPress and easily edit and make changes that will not cause the website to malfunctions.

Sturdy

A sturdy website derives from a lot good tests. A sturdy website means a user can navigate the site without it deviating from what is intended to do. Our website should not crash because of the user's interaction with it. Even after couple years, the website should be able to function properly even after new technologies are introduced.

² Wordpress. Retrieved September 12, 2015 from <http://www.wordpress.org/about>

3. Environmental Requirements

3.1 - Server Hosting

3.1.1 - Website will operate on an Apache server managed by ITS on the NAU Campus.

3.2 - Content Management System

3.2.1 - Website will be using a WordPress Content Management System (CMS) to manage client created content.

3.2.2 - Client will be able to manage assets (images and other files) through easy to use interface

3.2.3 - Client will be able to create and modify blog content through easy to use interface

A great deal of development is entirely dependant upon the hosting environment and the Content Management System (presuming that one is developing with a CMS in mind). In our case, our development is more focused on the CMS rather than the hosting server. This is the case since the selected CMS (WordPress) is dependant on a server that supports PHP and MySQL it was determined that hosting on an Apache server (which the WordPress documentation recommends) was our best option.

Server Hosting

The selection of the ITS operated JAN servers here on campus was determined by a number of factors. To clarify, the JAN servers are some servers here on campus that are available for faculty use much like how the CEFNS and DANA webspace is available for students. The first of which was that, for us and our client, it is free. Another important factor is that we know for a fact that ITS can and will give server support for WordPress sites since they currently maintain a fair number of them across the campus. All of this combined with the fact that they are fairly fast servers

with a speedy connection means that we will likely get improved performance from the current host.

Content Management System

The decision to use WordPress was made in part because the current site operates on it, and migration of data will be quick and easy. It will also allow for the client to be somewhat familiar with the environment, allowing for easier operation and management of content and assets by the client.

WordPress will allow for a simple interface by which the client can edit content on a single WYSIWYG (What You See Is What You Get) editorial field. This makes it very easy to fulfill the requirement that the client can create new content as well as maintain old content without the need for a working knowledge of coding or development.

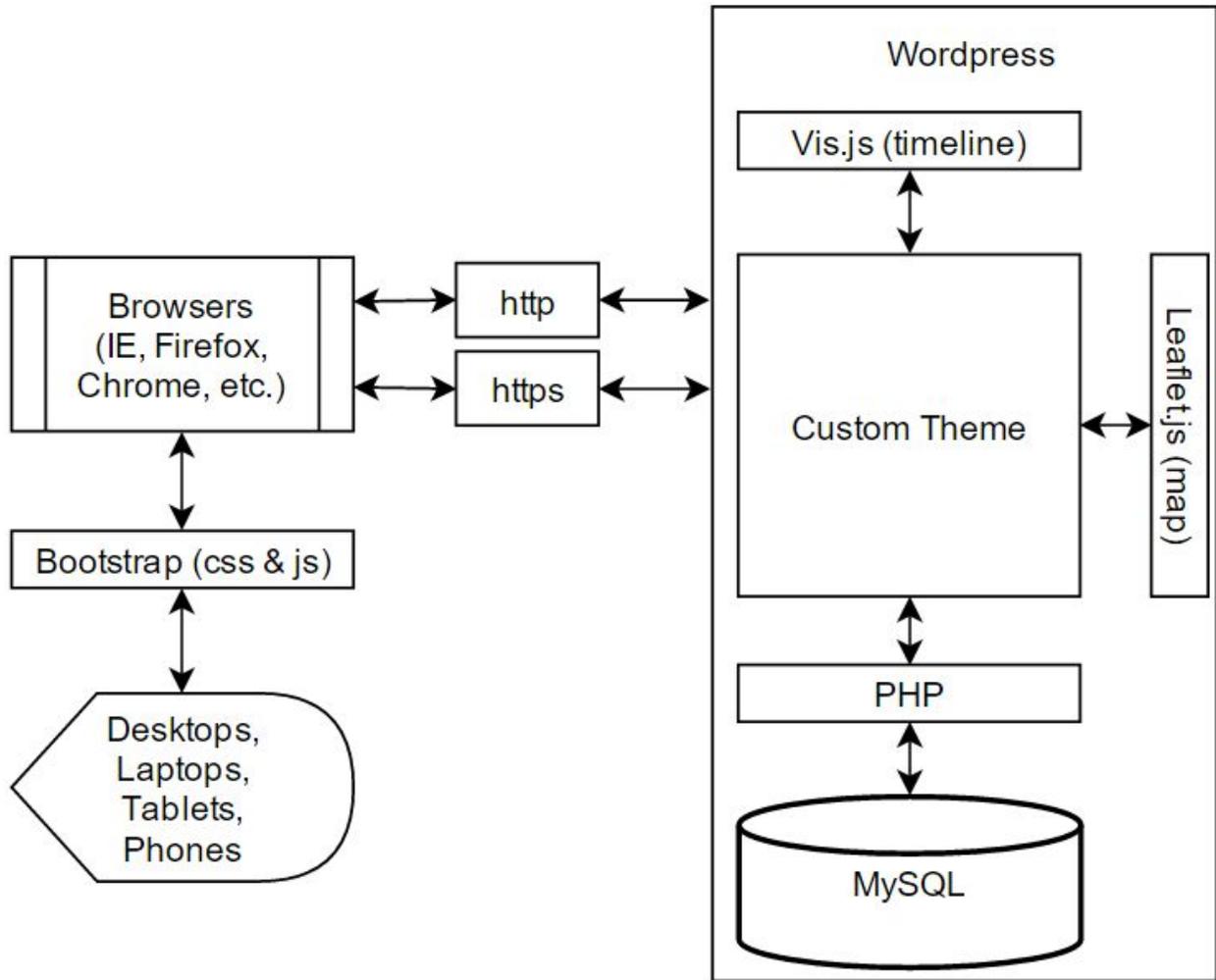


Figure 1: Technology Stack

4. Non-Functional Requirements

4.1 - Performance

4.1.1 - Pages will load quickly on both desktop, mobile, and everything inbetween

4.1.2 - Images will be optimized throughout the entire site for web delivery in terms of both speed and quality

4.1.3 - Videos will be hosted externally by free services such as YouTube for efficiency in web delivery that we cannot provide ourselves

4.1.4 - Resources (images, scripts, styles, etc.) will be compressed for faster transfer

4.1.5 - Scripts, Styles, and HTML will be minified (remove whitespace and a few other tricks to minimize the amount of data to be transferred)

4.2 - Maintainability

4.2.1 - Directions for client use will be well documented and explained in detail face-to-face

4.2.2 - Documentation will be available as both hard copy and digital documents for ease of use

4.2.3 - Development process and information related to how to continue to develop will be well documented

4.2.4 - All code developed by our team will be thoroughly commented

4.2.5 - All code developed by our team will be consistently formatted (naming conventions, indentation, etc.)

The goal of this project is to improve on the current website in many different ways, Two of those ways are in performance and maintainability. In the realm of performance we would like to improve on the dramatically slow 15 second load time, and in the area of maintainability we would like to heavily document both the end user processes as well as our own codebase.

Performance

In terms of performance, the current homepage is very very slow. According to the developer tools' Network tab on Google Chrome, the page load took 14-16 seconds. This is brutally slow. Our goal for this project will be to get that load time down to less than 1 second. We will do this through several means; possibly including, but not limited to, changing hosting environments, optimizing images, compressing resources, minifying text-based resources, leveraging browser caching, and eliminating render blocking for "above-the-fold" content. The purpose in all of this is to improve user experience by making the overall activity of accessing the site to be more enjoyable and relaxing.

Maintainability

As of right now, the current site has a single document concerning how to maintain the site. It consists mostly of directions for the end user to create and edit content in WordPress. While this is important, it is not the entirety of what is needed to really maintain the site. Our team will document not only how to create and edit content in WordPress, but how to manage any plugins that we create from an end-user perspective.

It is also important to note that there will likely be certain pages and sections of content that will be difficult to implement from within the content management system itself. For those sections of the website we will need documentation concerning how to modify plugins from a developer perspective along with how to modify any themes or child themes that we create. Similarly, we will ensure that all code adheres to pre-defined standards for naming conventions, indentation, code decoration, and commenting procedures. We will also include our "style guides" for what all of the standards for these things are in the event that the client grows tired of certain aspects of the site and wishes to change them after the project is over.

5. Potential Risks

5.1 - WordPress Plugins Compatibility

5.1.1 - Plugins are can become out-of-date or obsolete

5.1.2 - Plugins are can cause compatibility issues when WordPress is updated

5.2 - Client changes to code

5.2.1 - The client has no knowledge of programming and may stray from documentation and break something

5.3 - Map and Timeline Performance Verification

5.3.1 - The Map and Timeline are untested and unknown technologies

5.3.2 - If these technologies fail another will have to take their place which will be just as untested and unknown

5.3.3 - As of this writing the Map technology is still in beta version

5.4 - Frameworks Failing

5.4.1 - Should WordPress fail the client will need to seek outside help to fix it

5.4.2 - The next version of Bootstrap is being released meaning support for the version used might not be around much longer

5.5 - Server changes

5.5.1 - In the next few years it's likely that server structure will change and this will take the site down temporarily

5.5.2 - If in the next several years the JAN servers are taken down the entire site will have to be rehosted on whatever is then available

6. Project Plan

As was outlined in our *'Team Inventory'* document we will be taking an agile approach to developing our system. The project will have three main phases:

- Requirements elicitation
- Research and development
- Deployment and fine tuning

These phases are each continuous processes that will be pursued throughout the entire capstone experience. Below is a rough waterfall outline of the milestones that we will pursue in completing the system:

- A. Rough Draft Formal Requirements Documents
- B. Design Overview Powerpoint (this presentation)
- C. Final Formal Requirements Documents
- D. Website UML Architectural Documents
- E. Functional Prototype

While these milestones are listed in waterfall order, we are still making use of an *agile* approach to development. Milestones D and E are repeatable in the sense that as we develop our system and learn more details regarding implementation intricacies, we will be able to continuously update and upgrade both our formal architectural documents and our functioning prototype. The result of a full semester's worth of development and refinement will be a complete website system fulfilling all the requirements outlined in this document. The diagram on the next page describes the current schedule for our system.

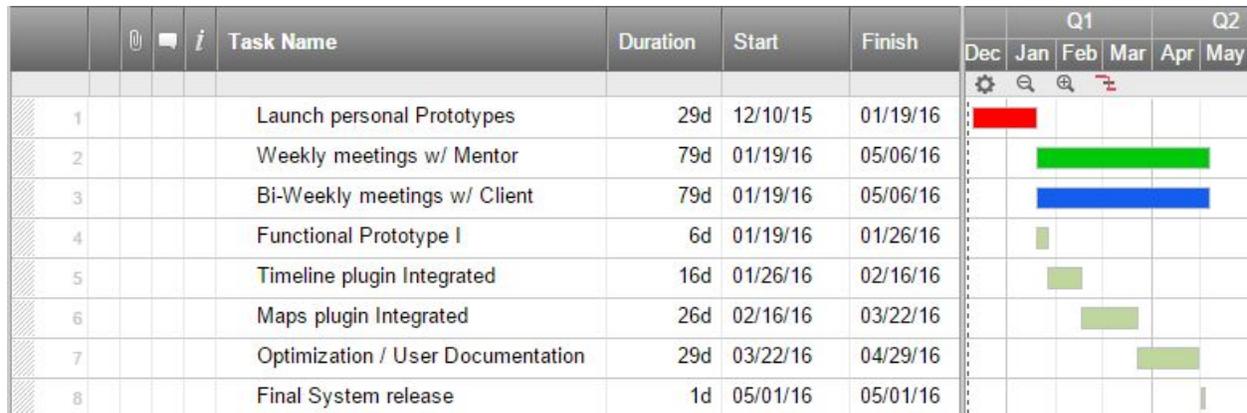


Figure 2: Gantt Schedule

The project schedule will be updated *at least twice* during the Spring semester once we have had more exposure in the development environment and have a greater understanding of our work throughput.

7. Requirements Appendix

This appendix is a review/consolidation of all key system requirements outlined by the previous pages of this document. Requirements are preceded by N.m where N and m are sections/subsections of this document.

2.1 - Historical Constancy

The website will include all stories, terms and definitions currently found on the site. It will also include new content.

2.2 - Front End User Interactions

2.2.1 - Users will be able to access **Exhibit**, **History**, and **Resource** website directories.

2.2.2 - Users will be able to view an interactive timeline of World War II

2.2.3 - Users will be able to view three unique, interactive maps detailing events surrounding the town of Bedzin

2.2.4 - Users will be able to access educational materials related to the exhibit

2.2.5 - Users will be able to access Martin-Springer Institute contact information

2.3 - User Interface Specifications

2.3.1 - Users from around the world will be able to access and website through modern web browsers

(Chrome, Firefox, Safari, Internet Explorer and Microsoft Edge)

2.3.2 - The website interface will be responsive to mobile devices and reformat itself accordingly

2.3.3 - The website will be feature minimalistic like most modern websites

2.3.4 - The website will follow good placement concepts so the objects are aligned currently so it can appeal to the website.

2.3.5- The website will feature intuitive animations to sort of bring life to it when the user is navigating around it

3.1 - Server Hosting

3.1.1 - Website will operate on an Apache server managed by ITS on the NAU Campus.

3.2 - Content Management System

3.2.1 - Website will be using a WordPress Content Management System (CMS) to manage client created content.

3.2.2 - Client will be able to manage assets (images and other files) through easy to use interface

3.2.3 - Client will be able to create and modify blog content through easy to use interface

4.1 - Performance

4.1.1 - Pages will load quickly on both desktop, mobile, and everything inbetween

4.1.2 - Images will be optimized throughout the entire site for web delivery in terms of both speed and quality

4.1.3 - Videos will be hosted externally by free services such as YouTube for efficiency in web delivery that we cannot provide ourselves

4.1.4 - Resources (images, scripts, styles, etc.) will be compressed for faster transfer

4.1.5 - Scripts, Styles, and HTML will be minified (remove whitespace and a few other tricks to minimize the amount of data to be transferred)

4.2 - Maintainability

4.2.1 - Directions for client use will be well documented and explained in detail face-to-face

4.2.2 - Documentation will be available as both hard copy and digital documents for ease of use

4.2.3 - Development process and information related to how to continue to develop will be well documented

4.2.4 - All code developed by our team will be thoroughly commented

4.2.5 - All code developed by our team will be consistently formatted (naming conventions, indentation, etc.)

6.0 - Project Plan

6 - The project schedule will be updated at least twice during the Spring semester once we have had more exposure in the development environment and have a greater understanding of our work throughput.