



Universitatea Tehnică a Moldovei

**PROIECTAREA SITE-URILOR GUVERNAMENTALE A
REPUBLICII MOLDOVA PE PLATFORMA DRUPAL
DESIGNING MOLDAVIAN GOVERNMENT WEBSITES
ON DRUPAL PLATFORM**

Student:

Sîrbu Ștefan

Conducător:

lector universitar Melnic Radu

Chișinău 2017

Abstract

The thesis named **Designing Moldavian Government Websites on Drupal Platform**, presented by student Sîrbu Ștefan as a Bachelor project, was developed at the Technical University of Moldova. It is written in English and contains 57 pages, 7 tables, 8 figures, 16 listings and 13 references. The thesis consists of a list of figures, list of tables, introduction, four chapters, conclusion, and references list.

Developing a website from zero is difficult job that requires a lot of time, but using a CMS simplifies your job as web developer. The main objective of the current document is to save developers time by using a Drupal distribution that will meet Moldavian Government website guidelines. This distribution will save your time and effort by allowing to move quickly to create a seamless website experience with distribution's ready to go features. From downloading to hosting, from transferring content to customising and organising your Government website, you'll quickly speed ahead. This distribution was refined to meet the specific requirements of the Moldavian Government, from perspectives of features this project is unique. Focus your time and effort on creating engaging content and customised features. Because your Government Drupal site comes fully loaded with a core set of user interface elements, functionality and features these can be reused as the basis for any new Moldavian Government website. For small Government websites, the majority of requirements have been met. Once it's installed, you're ready to go. For larger, more complex websites, you will have all the tools you need for customising and enhancing the features on your website.

The four chapters that compose the current report include the analysis of the encountered problem, the possible solutions for it and some additional hints for new features implementation. There are several addressed points of view that describe the current proposed solution. A careful review was performed, in order to emphasize all the strong sides of the applications and the components that need additional attention. Also, considering the current functionality provided by the application, some future directions for development were established. In order to ensure a plausible business plan for the stage when the product is released on the market, an economical research was accomplished. Analyzing the product from economical point of view offered a general idea about several financial indicators, which are crucial for the lifetime of the application.

This document is intended for readers with technical background.

Rezumat

Teza de licență cu numele **Proiectarea Site-urilor Guvernamentale a Republicii Moldova pe Platforma Drupal**, prezentată de studenta Sîrbu Ștefan în calitate de proiect de Licență, a fost dezvoltată în cadrul Universității Tehnice din Moldova. Aceasta este scrisă în limba engleză și conține 57 pagini, 7 tabele, 8 figuri, 16 listări de cod și 13 referințe. Teza consistă dintr-o listă de figuri, o listă de tabele, introducere, patru capitole, concluzie și bibliografie.

Dezvoltarea site-urilor web de la zero este un lucru dificil care necesită mult timp, doar că utilizând un CMS simplifică lucrul tău ca dezvoltator web. Obiectivul principal al lucrării curente este economisirea timpului dezvoltatorului prin utilizarea a distribuției Drupal care va satisface cerințele a Guvernului Republicii Moldova pentru site-urile web. Această distribuție vă va economisi timpul și efortul, permițându-vă să vă mișcați rapid pentru a crea o experiență fără probleme a site-ului cu funcțiile disponibile pentru distribuție. De la descărcarea la găzduire, de la transferarea conținutului la personalizarea și organizarea site-ului web al Guvernului, veți accelera repede. Această distribuție a fost îmbunătățită pentru a îndeplini cerințele specifice ale Guvernului Republicii Moldova, din perspectiva caracteristicilor acest proiect este unic. Concentrați-vă timpul și efortul asupra creării unui conținut captivant și a unor funcții personalizate. Deoarece site-ul dvs. guvernamental Drupal este complet încărcat cu un set de elemente de interfață cu utilizatorul, funcționalități și caracteristici, acestea pot fi refolosite ca bază pentru orice nou site Web al Guvernului Moldovei. Pentru site-urile guvernamentale mici, majoritatea cerințelor au fost îndeplinite. Odată ce este instalat, sunteți gata să plecați. Pentru site-uri mai mari, mai complexe, veți avea toate instrumentele de care aveți nevoie pentru personalizarea și îmbunătățirea funcțiilor pe site-ul dvs. web.

Cele patru capitole care compun raportul curent includ analiza problemei întâmpinate, soluții posibile pentru aceasta și câteva sugestii adiționale pentru implementarea unor noi caracteristici ale aplicației. Deasemenea, aplicația este analizată din câteva puncte de vedere distincte și concrete, menite să descrie soluția curent propusă. O revizie minuțioasă a fost îndeplinită pentru a sublinia toate punctele forte ale aplicației și, totodată, componentele care necesită atenție adițională. Deasemenea, luând în considerație funcționalitatea curentă a aplicației, câteva direcții pentru viitoarea dezvoltare au fost stabilite. Pentru a asigura un business plan plauzibil pentru momentul în care produsul va fi lansat pe piață, o investigare economică a fost realizată. Analizând produsul din punct de vedere economic, a fost posibilă structurarea unei idei generale asupra câtorva indicatori financiari, care sunt cruciali pentru durata de viață a proiectului.

Acest document este destinat cititorilor specializați în domeniul tehnic.

Table of contents

List of tables	10
List of figures	11
Listings	12
Introduction	13
1 Description and Analysis of the Domain	14
1.1 The purpose and objectives of the project	14
1.2 Analysis of this field	14
1.2.1 Planning	17
1.3 Project Analysis	25
1.3.1 Drupal	26
1.3.2 mGov	27
1.4 Comparing with other systems	28
1.4.1 Similarities and differences	29
2 Designing the System	30
2.1 Themes	30
2.1.1 The Bootstrap Process	32
2.1.2 Initialize Configuration	32
2.1.3 Early Page Cache	32
2.1.4 Initialize Database	32
2.1.5 Hostname/IP-Based Access Control	32
2.1.6 Initialize Session Handling	32
2.1.7 Late Page Cache	32
2.1.8 Language Determination	33
2.1.9 Path	33
2.1.10 Full	33
2.1.11 Processing a Request	33
2.1.12 Theming the Data	33
2.2 User Permissions - Use Case Model	34
2.3 Blocks	34
2.4 Modules - Sequence Model	35
2.5 Nodes	35

	2.5.1	State Machine Model			UTM 526.2.XXX ME	35
	2.5.2	Class Model				37
Mod.	Coala	Nr. document	Semnăt.	Data		
Elaborat	Sirbu S.				Proiectarea Site-urilor Guvernamentale a Republicii Moldova pe Platforma Drupal	39
Conducător	Melnic R.					39
Consultant	Sava N.					
Aprobat	Ciorbă D.					
					UTM FCIM FAF– 131	

3.1.1	How Drupal works?	39
3.1.2	Nodes	40
3.1.3	The Core	40
3.1.4	Fast and Flexible	41
3.1.5	How it flows	41
3.2	Distribution	43
3.2.1	What is a distribution?	43
3.2.2	Distributions and installation profiles	44
3.2.3	When we need distributions?	44
3.2.4	Creating Drupal Installation Profile	45
3.3	Custom Modules	47
3.3.1	Object Oriented Programming	48
3.3.2	Symfony 2	48
4	Business Plan	49
4.1	Schedule	49
4.1.1	Objectives	49
4.1.2	SWOT Analysis	49
4.1.3	The time meant for the accomplishment of the project and the schedule	50
4.2	Economical Proofs	52
4.2.1	Material and non-material expenses	52
4.2.2	Salary expenses	53
4.2.3	Indirect expenses	53
4.2.4	Wear calculation	54
4.3	Project Cost	54
4.4	Financial results	54
4.5	Conclusion	54
Conclusions		55
References		57

					UTM 526.2.XXX ME	Coala
Mod	Coala	Nr. document	Semnăt.	Data		10

List of Tables

4.1 SWOT analysis 50

4.2 Projects Schedule 51

4.3 The expenses on the materials used in the project development on long term 52

4.4 Non-material expenses 52

4.5 Direct materials costs 52

4.6 Salary expenses 53

4.7 Indirect expenses 54

List of Figures

1.1	CMS learning curve	27
2.1	Drupal Architecture Layers	30
2.2	Drupal Page Serving Mechanism	31
2.3	Use Case	34
2.4	Sequence Diagram	35
2.5	State	36
2.6	Class Diagram	37
3.1	Drupal Flow	42

Listings

1	.info.yml file	45
2	.install.yml file	46
3	.profile.yml file	47
4	.info.yml file	47

Introduction

In our technology era, Internet is one of the most useful tools where big amounts of data circulates through the network which covers the whole world. We spend our time rushing after this information cause knowledge is power and data is money. Nowadays website are the efficient way to present and spread information to the maximum number of people in the world. By using web browsers you can access the rendered information from a webpage which basically represents a structure from blocks of media or text content.

Websites, which represents a mix between hypermedia and information systems, from users point of view websites have relatively simple architecture but from developers point of view websites have a more complex dynamic architecture. Websites need to respond to an unlimited number of heterogeneous user's request, privacy and security concerns, up to date information sources.

Any organisation in the world need a website alike the government institutions need one. Using a website an organisation is promoting itself by providing information about them, such as who they are, what they do and how they do. The goal of my thesis is to develop a web platform for all governmental institutions.

In the first chapter we will discuss about domain analysis of the implemented product and we will see the difference and similarity between this product and others. Of course we will talk about why do we need this system. Here we will define notions which will be used mostly in thesis and why they are important from various perspectives.

After comes system design and how it was planned to be. Using various visual representations will help you understanding the structure and relationships between system components.

In the third chapter you will find out how the product was developed, the languages and techniques were used to create the product. Also you will see pieces of system's source code, what problems were encountered developing the product and their solutions. Here you will perceive what technologies was used and why, further you will learn the implementation of their futures.

In continue we will explain how the product can be used. We will show up installing, configuration steps. This chapter is a user guide for the product, that can be used by a non technical person. User guide includes screenshot images which describes step by step how the user can use the product.

The last chapter but not the least is about a business plan for developing the product. This plan represents an economic representation for the project which covers how much time we need to accomplish projects goals, who is involved in project and how much they will be paid. Mostly this chapter is about projects profitability for this product.

1 Description and Analysis of the Domain

The days of building websites that serve as static, digital pamphlet are long gone. Today, any information providing entities, including governments, are expected to provide constant, dynamic, customized, and shareable content. Without a tool that allows government website administrators to manage their own content, site updates can only be accomplished by a website developer with the tools and knowledge to write source code. To accomplish updates through such a vendor service model, governments must contract with a development company, submit requests, and wait days or even weeks for new site content and functionality to be online.

1.1 The purpose and objectives of the project

What can a website do for you, and for the government? First of all a website is a communication tool, it provides to users and government based in remote locations with a way to exchange and share information over the network.

Being more specifically, from the point of view of government a website can fulfill one or more functions. A website can be a publishing tool, such as a magazine, newspaper, radio, television by publishing about government various activities. Also a government website can be a marketing tool, promoting their services, activities or it may be used for building and enhancing the image of public institutions. For example, increasing the trust of the public for a new regulatory authority or promoting commercial services such as documentation services, activities of statistical office or sites that promote public libraries. Additionally governmental website can be a transactional tool, which allow the users and governmental institution to exchange all the information required to support a transaction. For example paying taxes, requesting licences or managing contracts. Also it can be a work tool, allowing a public agency to exchange information with its employees and thus facilitate professional activity. As an example websites with restricted access that allows police authorities to access the vehicle registration databases managed by transport authorities. Another example would be a site designed to make medical knowledge resources available to doctors.

Government institutions are interested in one or more of the possible use cases illustrated above and if planned and realised appropriately than a website can be a relatively inexpensive and efficient solution if compared with other methods of communication.

1.2 Analysis of this field

Like other media applications, a website is only one of variety of communication methods likely available to governmental organisations, which has its advantages and disadvantages.

Before investing in a website agencies first evaluate if communicating through a website is an appropriate and viable way to address their communication needs. Here are some examples of the advantages and disadvantages of using a website for this purpose.

Let's begin with the most loveable benefit by our government, it is cheaper. A website can be a very cost effective way to exchange information, for both sides of monitor, for website's owner and the end user. For instance, from the point of view of the government agency or institution a website

can reduce the number of employees the agency or institution has to deal with, by providing answers to the most common queries or questions, also known as FAQs (Frequently Asked Questions), it normally receives, and so reduce the amount of employees effort and cost needed to answer them. Generally speaking a website makes it possible to publish extensive information on regulations, procedures and other aspects of an institute's work, which would otherwise need to be announced through direct human intervention, this means a governmental institute or agency can dedicate less staff time to dissemination activities, allowing either for a reduction in staff numbers, or for their employment elsewhere in the agency. Websites also represent a cheaper alternative to the production and dissemination of printed materials, like brochure, letters and so on. Also, it should always be considered that building and maintaining a website also has its costs, see disadvantages below, so agencies planning to use a website to try to cut staff costs should always make sure that the savings obtained to offset the cost in terms of staff and money necessary to build and maintain a web application. A website can also represent a cheaper communication alternative from the users point of view, especially if they are located in remote regions or abroad, as a phone call to an internet provider is often cheaper than a long-distance call, or a physical talk with the agency.

We all know the phrase "Time is money" and here step in another benefit which is quicker. Web publishing is immediate, and this enables agencies to introduce changes to their public easily and relatively quicker than with traditional and more costly methods depending on print. The immediacy of web publishing also allows the fast release of news and other information items to the public, which an agency may find advantageous, respond to promptings from the media, or act in moments of crisis. In addition to this, if maintained properly a website provides information 24 hours a day, 7 days a week, and can be available for the parts of the population either that find it hard to visit the agency in person, especially people with disabilities or those who cannot contact it during working hours. This can extend to providing the possibility to conduct transactions, for example requesting a licence, outside working hours and/or without visiting government offices, which can be useful in our country where travel is slow, costly or generally not easy. Conducting transactions online is also generally faster, as data can be recorded and transferred very quickly, as opposed to what happens with other, more traditional ways of handling bureaucratic procedures, such as paper-based forms, manual input by officials and so on.

Websites can provide better ways to manage information compared to traditional means. Hypertext enables you to provide access to complex sets of information in relatively easy and user-friendly ways and combined with databases, it can also provide users with ways to interact directly with the data, for example to consult it, update it and/or send it to other destinations. In other words, a website can enable users to initiate and/or complete service transactions without recourse to human intervention. The absence of a gate keeping role on the part of humans not only saves time and money, but it is also likely to help reduce and prevent phenomena such as corruption, blackmail and bribery, a problem experienced by many governments, especially in our. By potentially cutting out "middlemen", websites represent a relatively transparent way of administering services and information, in comparison to traditional means. Having a website also provides governmental bodies with an opportunity to systematically collect information about their clients and users, this includes

running surveys, asking users to share their personal data in order to register for specific services and therefore easily create and maintain a database of users of these services, and generally monitoring the behaviour of visitors to the site, for example through statistical analysis, which can be used to establish what information users are most interested in.

Websites provides a new, additional point of contact for the growing part of the population that uses the web. It is estimated that currently just over 47% of the global population has access to the internet, an increase from just one year ago, when the same agency estimated that just over 43 percent of the global population were Internet users. That said, it should be considered that the percentage of internet users in developing countries is very often significantly lower than in developed countries, though it will increase slowly, in the next years. This means that agencies with international audiences such as donors, international institutions, and governments of other countries may be in a better position to exploit the potential of communicating on the web than those who deal directly with the local population. Implementing the use of email and/or online forms through a website also offers a fast and efficient alternative communication method for those who prefer indirect contact to communicating in person or on the phone for time or personal reasons. Basically, a website can generally introduce new ways of doing things , or new services altogether like paying your tax by credit card, which may have been too expensive via traditional channels because of the expenses connected to having card-reading machines. Finally, a website also represents an innovative tool for hiring new employees, providing agencies with a way to announce vacancies to a wide audience relatively cheaply, and therefore enlarge the potential base of recruitment, with the added bonus that those who apply via a website are likely to be in possession of some computing and web skills that could be put to good use in the organisation.

Now let's look on the other side of the coin, disadvantages. Websites cost, building and maintaining governmental websites can require quite a lot of time and money. Apart from technical costs, there are also important human resource costs. Any existing or new staff working on the web must be trained both on the technical and communicational aspects of the new technologies they are to use. Agencies should therefore always factor in recruitment, training and other organisational costs when budgeting for a website, all of which can be particularly expensive in developing country contexts, where local capacities and expertise may not be sufficient, and trainers, consultants and technologies have to be called in from abroad.

Implementing the use of e-mail through an government website potentially leads to a new flow of enquiries from the public, through a different channel. Agencies should therefore ensure their organisational structure is capable of dealing with these communications in a timely and efficient way, especially considering that web users often expect quicker reaction and communication than those who communicate with traditional methods. It is hardly acceptable to adopt e-mail as an additional, fast method of communication and then leave emails unanswered for long periods of time. Again, this points to the costs mentioned in the previous point.

Web publishing usually represents an alternative and additional source of information to print publications, press releases and media broadcasts. It is essential that the entire information output by a governmental organisation is consistent and not contradictory, and adding an extra source like

a website makes this even more complex, especially considering the immediacy of web publishing. This is particularly so when it comes to information with legal value such as laws or contracts, where discrepancies can cause serious issues.

Publishing information of legal value or granting access to information databases through a website can pose security dangers to governmental organisations, as malicious users, also known as hackers, can break into website systems through the internet and access, modify or delete information, up to and including making a website inaccessible, or deleting it completely. Defending your website from the attacks of hackers should be a priority, especially if it carries or gives access to legal or confidential information. All this leads to more costs in terms of software and expertise.

A website does not reach the entire population, but only the part of it that has access to the web. This is especially a problem in developing countries, where access to the internet is still often limited to a very small part of the population. Websites should therefore be considered only as alternative means of communication, and traditional methods of exchanging information should also be offered and maintained, at least until they become redundant as was recently the case with the telegraph.

How can government agencies reap the benefits of having a website and reduce the impact of the disadvantages outlined above? The answer lies in good planning. If web projects are planned and funded appropriately, they usually have higher chances of being a success.

1.2.1 Planning

You will agree that if a government agency decides to produce a new publication, or introduces a new system for paying tax, or an innovation in the way it works, it would be good practice to plan the operation as effectively and efficiently as possible, in order to reduce the risk of possible failures and the waste of public funds, if not worse, they generate.

The same can be said for websites. Building, developing and maintaining successful sites are all operations that require a good deal of investment in terms of time and money. Government agencies considering the adoption of a website as one of their communication tools should therefore make sure that the operation is planned appropriately, by giving particular attention to both the agency's and the user's communication needs, and producing a sustainable and realistic project, in accordance with the resources at their disposal.

Planning your website properly will increase the chances of success, and save you a lot of money and resources, which can be limited in the context you work in. So, before you even start to design and implement a website, sit down and plan how to turn the project into reality with the resources you have at your disposal.

A simple and effective approach to planning a website is to answer five basic questions:

- Why do you want to set up an Government website?
- Who is the website for?
- What information do you need to put on the website?

- Whom will you make responsible for it?
- How much can you spend on it?

Why do you want to set up an Government website? Your website will be more likely to be successful if you set a series of clear objectives for it. These objectives can be improved if they are SMART: Specific, Measurable, Agree, Realistic and Timed.

Specific - setting goals as specifically as possible will enable you to provide clear guidelines on the features and functions that need to be implemented on your website, and therefore economise in terms of time and resources. For instance, instead of defining your goal as "providing information to the public", it would help to say "provide X type of information, to Y type of public, in Z format and at XYZ conditions". Indetermination in your objectives may lead to the realisation of an unfocused, confusing and potentially unsustainable web application.

Measurable - the objectives you set should be measurable. If, say, one of your objectives is "to reduce the time it usually takes to obtain a licence by half", for example by allowing the transfer of documents in electronic format through the site, rather than relying on traditional mail, you should make sure that the organisation you work for already has reliable methods in place to measure this reduction. In this case it would mean that you need to firstly, establish the average time taken to obtain a licence with traditional methods, and secondly record the time when the request is filed through your site, and the time of release of the licence, and compare this to the time taken with traditional methods: some of this requires the implementation of specific technical features in your website, and it is essential to establish this from the start.

Agreed - it is important that the objectives you set are agreed with the key stakeholders in your organisation, and particularly with management; this will provide you with common and formally accepted criteria to review the progress of the project in the future, and with the clarification of the different lines of accountability by the people involved in the project. If you obtain the sponsorship of key people in your organisation, you will also be in a better position to solicit resources for the project's realisation. Best practice would be also to include representatives of user groups in your discussions and consult them on their expectations from the website you intend to set up.

Realistic - agencies should make sure that they evaluate the skills and resources they have at their disposal, and plan accordingly; setting unachievable goals for an e-government website represents a waste of time and public resources. As an example, rather than aiming to handle all applications online, it may be better to aim to handle "at least 90%" of applications this way.

Timed - it is valuable to set deadlines for the achievement of objectives, as it helps to measure the achievement of your objectives: "handle 90% of enquiries online by 2018". Deadlines can be reviewed if priorities change or unforeseen difficulties arise, including technical problems, which can be quite common with websites. It is therefore important to also build some flexibility into their definition.

By consulting with the relevant people in your organisation, it is likely that you will come up with more than a single objective for your website. In this case, it can help to prioritise objectives.

Assigning different priorities to your goals will help you make important decisions that affect

the design and implementation of your website. This will help especially if resources are poor: if you don't possess the necessary resources to achieve all goals, you can use what is at your disposal to obtain at least the most important ones.

For example, the prioritised goals for a public agency's website could be:

- to publish a specific set of information in order to reduce the number of queries from the public,
- to enhance the profile of the institution in order to attract donors to collaborate with it, and
- to enable users to carry out online transactions with the authority.

If you only possess very limited resources, you may consider dropping the third goal or deferring it to a later stage of the life of the website, especially because it is likely to put heavy technical requirements on the project . If instead you had assigned top priority to online transactions, you would need to use most of your resources towards that goal, which may risk your chances of achieving the other ones.

The bottom line is that it is usually better to achieve a limited number of goals efficiently than to pursue a larger number of goals and only achieve them partially.

Who is the website for? After establishing the primary objectives for your website, you should focus on who the audience for your website is likely to be.

In fact, this is the factor that will have the biggest influence on the way you structure and design your site: you should always bear in mind that a website is primarily a communication tool, and that as such, it should be as easy and straightforward as possible to use for your intended audience. In other words, the way you organise and structure contents, the navigation system you implement, and the layout of the site should be as user-friendly as possible. In real-life situations, and especially if you set multiple objectives for your website, it is likely that you will be looking at catering for a variety of users . In this case, it would help to prioritise users in different categories, as this will provide you with clear criteria for taking decisions in case you cannot satisfy the needs of all users with the resources you have at your disposal.

There are two main aspects it would help to focus on when you analyse who the users of the website could or should be: the type of activity they would be likely to be undertaking when visiting your website, and the user's technical profile.

The activity users intend to undertake, or the informational needs they intend to satisfy when they visit your government website, will influence the way they use the website itself, and you should be prepared to take this into account when planning the site. In the case of government websites, main user profiles can be roughly divided into two categories:

- Seekers of information: these users are looking for a specific piece of information, and they will browse the site until they succeed or fail, as such, they usually benefit from the implementation of site search features, indexes, sitemaps, and a hierarchical, category-based navigation system, which all help them to find the information they want more quickly. In order to cater for cases in which the information required cannot be found on the site, it would also help to

supply e-mail contacts or indicate other communication channels through which enquiries can be made.

- Users in need of online services: these users are looking to undertake an online transaction through a website for example requesting a permit, or paying tax. They need to be pointed directly to this service, as carrying out the transaction itself is likely to require consulting a variety of web pages and completing different steps. Implementing direct links or buttons to services from the homepage, or from every page on the site, would reduce the time it takes for the transaction to be completed. Clear instructions and explanations for every step in the transaction are also very beneficial, and can foster the user's trust of the application, and of the institution at large.

Technical skills, this includes both the technical setup your users are likely to be using when they access your website, and their skills and proficiency in using IT and alphabet-based communication tools.

Users in developing countries often have limited access to recent technologies, and access the web through relatively slow connections: it is therefore important that you design your website to be accessible also in these conditions. This means you should, for instance, avoid adopting very complex graphical designs and avoid relying on very large files to convey information.

The typical users of government websites may also be people with varying IT and general knowledge: this is especially the case if the site is designed to cater for a very wide section of the population. You should therefore make sure that the site is usable and comprehensible by most, if not all, users, including those whose skills and knowledge are limited. This means first of all using common language and clear, extensive instructions. But it also means that you may have to provide alternative routes for users, for example instructions for disabled people, or information in the different languages in use in your audience. If, instead, your typical users are specialists in a specific discipline, and your website is designed to cater for their needs, for example a website providing access to a medical knowledge resource database for doctors and hospital staff, you should make sure the terminology used in the site is specialised enough to be meaningful to them.

Failing to understand these requirements from the outset may lead to the need for an expensive and time-consuming review of the project later on, or to its ultimate failure.

What information do you need to put on the website? An government website is primarily a communication tool designed to facilitate the exchange of information between the public agency and citizens, or other parts of a society. The aim of this exchange of information is usually to support decisions, for example helping a citizen or an organisation to decide whether to apply for a public grant, or convincing a donor that they should fund a government project, or to facilitate a transaction, for example enabling a citizen to request a permit or licence. In short, government websites are there to support actions on the part of their users, by supplying all the information needed to take decisions, or finalise transactions.

You should therefore ask yourself what will the typical user be trying to do when visiting my website? What actions is she or he likely to want to undertake, to which the website could be of

support?

This will help you establishing the type of information you should supply through your site. By running through what could be the typical decision-making processes of your users, you will be able to identify the information they require in order to make their decisions, and evaluate whether this can be supplied through your website.

For example, if one of the goals of your website is to help citizens decide whether to apply for a grant, you would probably need to provide:

- a navigational system designed to direct users from the first page they consult to a page about the grant that would best fit their needs;
- general information about the aims and scope of the grant;
- information on the criteria of eligibility for the grant, both in terms of the applicant and the project to be funded;
- information on how to proceed with the request, and on the practical aspects of the application, of the awarding and of the transfer of funds, including deadlines and other time-sensitive information;
- the possibility to apply online, if this is possible;
- if relevant, information on alternative funding possibilities if the criteria for eligibility are not met, or the grant does not fit the needs of the applicant.

In the very likely case that you have set up a number of different, prioritised goals for your site, and have identified a number of prioritised groups of users, you will also have to assign priorities to the types of information you will need to supply through the site.

Reflecting on what information you should provide through your website is particularly important if you consider that sourcing, editing and managing contents is perhaps the most time-consuming and human resource-intensive aspect of the realisation of a web project: while design and technical implementation normally require the input of a relatively limited number of people, the production of relevant, up to date contents usually depends on the involvement of staff from different offices and departments in your institution, and needs to be planned efficiently, and fit into people's workloads. It is therefore essential that you focus on what the key requirements are in terms of contents for your site, and this is where prioritising types of information, and the formats in which they are made available, will be useful.

It should also be noted that in some cases you may want to provide particular pieces of information which may appear as being of secondary importance if compared to the core goals of the site, but which may help you attracting a larger number of users, which you can then try and direct towards the core functionalities of the site, for example in order to attract young people to a site that contains information about birth control or HIV prevention, it may help to also publish information that is relevant to other aspects of their life, such as entertainment or sport, which may

not be entirely relevant to the ultimate objective of the site, but which could exercise an important "pulling" power for young users.

Whatever you choose to publish on your site, you should make sure that you set some priorities for the different types of contents, and that you actually are in possession of the information needed, or of efficient ways to obtain it from your organisation. And you should always put yourself in the position to review contents and priorities in the future, if the perceived users needs change after carrying out an evaluation, or survey.

Whom will you make responsible for it? Although it may be designed to help you reduce administrative work in some areas, you should always consider that a website will create work for your agency.

Staff resources will be needed not only to produce and launch the site at the start of the project, but also to maintain and review it in the long term, and to respond to the extra flow of enquiries through different channels it may generate.

However you decide to allocate the necessary staff resources for developing and running a website, you should make sure you think in a long-term perspective : you will need to find staffing solutions that fit with the agency's priorities and budgets in the long run, to make the project viable and sustainable.

There are three main areas of activity in the development and maintenance of a website, for which you should assign responsibilities:

Design and Development: elaboration of the structure and graphical layout of the site, implementation of the navigation system, creation of sample pages, also called templates, and other technical aspects connected to the creation of the site. This is not necessarily a permanent function: most of this work will in fact be done only at the start of the project, during which the person responsible for the project will need to work in close collaboration with whoever you have made responsible for it, be it internal to the agency or hired externally. But you should consider that you may need to review the design or implement new features later on, once the website is up and running, so you should always make sure you have access to the skills and resources needed to carry out this work if necessary.

Website Administration: day-to-day running of the site, including reviewing and updating contents and making sure that links, forms and other features of the site keep working. This is an essential aspect of the delivery of a website, as it guarantees that the site is online, that it works properly and that it provides the information you want it to provide, in a timely and accurate fashion. The management and oversight of a website are long-term operations which usually require medium to advanced technical skills, and are best carried out in-house, for example using resources internal to the organisation, for better immediacy and accuracy.

Content Provision: this aspect of producing and maintaining a website is often underrated, but it's a very important part of the process. Content should be uploaded regularly and kept up to date: relevant members of staff in your organisation should therefore commit to providing content for the areas in which they operate and on which they have knowledge and expertise. Remember, too, that website contents need to be adapted for publishing: writing for the web follows different

rules than, say, writing for a magazine or writing a report, and you should make sure that someone in the organisation is trained to handle different types of contents and make them fit for publishing on the web.

To cover all these functions you will need to allocate resources and identify the people who will carry out the work, both in the short and in the long term. In order to do so, you should ask yourself the following questions:

In-house or outsourced?

For every aspect of the production of a website, you should choose whether it is more convenient and efficient to address it by using internal resources that can be made available within your organisation, in-house or by hiring another public agency or a private company to do it outsourcing. This depends very much on how important the website is for your organisation. Outsourcing can be expensive, as private companies generally charge hefty amounts of money for web work. If the website becomes a constant and essential feature of the agency's work and requires a constant input of resources, you would probably be better off using internal resources. You may also decide that you want to outsource some of the work, and keep in-house the rest. A typical example: the first stages of the design and implementation of a website are often assigned to professional web designers outside the organisation. The maintenance of servers, which can be complex and rather expensive to run, is also very often outsourced, unless the organisation is very large or needs to control their servers directly, for security or legal reasons. Content management is instead often done in-house, as it requires knowledge and expertise that can be found more easily within the organisation than outside it.

In the case of in-house solutions, you should evaluate whether it is possible to assign tasks to existing staff, or if you need to employ new staff. This depends on the workload generated by the website: if this can be sustainably integrated into present workloads, then you may choose to work with existing staff. But it is likely that you will need extra resources to run a website properly, so you should also consider employing new staff. This doesn't necessarily mean employing a new person to work on the web, perhaps one or more current employees possess the necessary skills, or are willing to develop them, the bonus in this case is that they already know the organisation very well. They could therefore be the right person for the job, so you might want to give them new web-related positions, and recruit new staff to cover the one they are leaving vacant. Training and redeploying existing staff may also be an interesting solution for organisations undergoing reform and restructuring, as it may help in finding a new role for members of staff that may otherwise become redundant.

However you decide to allocate the necessary resources for developing and running a website, you should always make sure that you do so in a long term perspective. While a website can be quite expensive and resource intensive to set up at first, if you want it to be successful you will have to take into account a constant effort, both in terms of time and money. Regular reviews and updates of contents are essential, as is the general maintenance of the site. If you opt for in-house solutions, you can try to make sure that website related tasks are integrated into the job descriptions of the staff you employ, and that sufficient time and incentives are provided in order to carry them out.

Don't leave this to improvisation: it's a recipe for disaster. If no one has the time to work on the site, this will become unusable and will detract from the overall image of the organisation. Like all other operations within your organisation, creating and running a website must be sustainable, if it is to be a success.

How much can you spend on it? Creating and maintaining a website costs money, and you should make sure you budget for these costs appropriately. Most of the expenses are human resource-related, but you should also budget for technical costs, and unforeseen circumstances, as technical problems, unexpected requirements, training due to staff turnover and so on.

Building the necessary skills to construct and maintain a website can be pretty expensive, especially if you choose to hire external companies to do part of the work.

Opting for in-house solutions may appear less expensive, but it may turn out to represent an important cost in the long term, especially if you choose to employ new staff. And even if you decide to only use existing staff, they will need to be trained to do the job, and will not be in the position to contribute to the rest of the organisation's work during training, so you should factor that in when you budget for the project.

Opting for outsourced solutions does not entirely eliminate internal costs, as staff internal to the organisation will have to set up tenders and manage relationships with the company or companies chosen to do the work. Liaising and collaborating with external companies in order to explain what your organisation needs, and managing this relationship, can also be rather demanding, especially in the initial phases of the project. Often the amount of time needed for planning and setting up the site is underestimated: make sure you don't make this mistake.

There are obviously some technical costs you have to take into account when it comes to producing and maintaining a website:

- hardware costs (computers for internal staff, printers, scanners, connections etc.).
- software costs (web design software, content editing software, picture editing software): avoiding such costs by trying to use pirated software can be particularly inappropriate for a government organisation.
- hosting costs: unless you decide to implement your own servers, which has itself its own costs in terms of hardware, software and maintenance or a central public agency can provide hosting for you, you will have to pay for commercial web hosting services, pay a company to provide file storage and to connect the site to the internet.
- technical support costs (technical help in case of software or hardware failures, general maintenance).
- domain name registration costs: the address of a site must be registered with the relevant authorities. Depending on national arrangements, this may be free of charge for governmental organisations, if not, you will have to pay a company to register and handle the web address or domain name for you.

- promotion costs: the cost of publicising and promoting the website, especially when it is first launched. It is no good creating a website and letting no one know about it: you will need to spend money in order to make the public aware it exists.

Nowadays governmental institutions usually face important financial constraints, and have limited resources at their disposal. This is especially true in developing countries. It is therefore likely that you will not be able to command princely sums to create and develop your website, and that you will have to work with a tight budget.

When budgeting for the project, you should therefore try to establish a balance between, what you ideally want to do with your website and so, ideally, how much you would need to spend on it. How much you can justify as a basic and fundamental need, without which the basic features of the project cannot be implemented.

This balance depends on a variety of factors, such as the goals you set for the website, and the level of sponsorship and support you manage to obtain for the project inside and outside your organisation. This is also where setting priorities for your goals will prove most useful. If the budget only covers a part of the goals, you should concentrate on the most important ones, and drop other, secondary goals, or defer them to a later stage, when the business benefits of having a website can be proven and you are in a better position to solicit more resources. If the budget available does not even allow you to achieve the most basic and primary goals of your website, the project might have to wait, or you might consider modifying its goals, or catering for a more restricted category of users.

In order to make the case for specific budgets to be put aside for the realisation of a website, you will probably need to demonstrate how this investment can be beneficial for the agency, and how it may lead to savings, or increase the agency's income by attracting donors, or combating tax evasion by providing easier ways to pay tax. When you do this, a full calculation would make sure you offset any benefits with the real expenses of creating and maintaining a website, especially in the long term.

1.3 Project Analysis

Thanks to advance in content management systems, governments no longer need to depend on IT companies to support with updates. Using a content management system, government website administrators without any knowledge in code editing, they can successfully update content, events, images, documents, videos, and other dynamic media by themselves.

Aside from ease of use, timeliness and data storage, a CMS offers governments the next benefits.

“True genius lies in simplicity”, in a society where constant access to the latest information is requested and expected, governments must be able to easily and immediately to provide information to citizens, especially when it comes to emergencies, weather, traffic, election results, and other time sensitive information. Having a CMS that is easy to use and easy to update can save government's time and money.

In an intuitive content management system, the content and design lives in isolated functionalities. When is set with specific permissions, each feature would allow website administrators to work with content without fear of accidentally changing the overall website design.

Governments require to comply with government resolution number 188 of the 3 april 2012. It's vital for websites to provide the capabilities needed to ensure content is easily accessible by everyone. Not meeting website guidelines for accessibility could raise legal issues between citizens and government authorities.

On the rise of cyberattacks, it's becoming more important to protect government websites from data breaches and risk exposure. Without a secure CMS municipalities are exposed to any loss and theft of information taken from the website. Using a CMS with multiple levels of security to reduce and cover a cyberattack can save government institutions from the loss of money, time and reputation.

Partnering with a government website design company, such as Center of Special Telecommunications, that offers access to a CMS and hosting will help to avoid much of the technical work needed for housing your site. The most valuable partners will be able to support your hosting needs with advanced data center technology supported by redundant networks, firewalls, and network providers, to help keep your information online 24/7.

The most important functions of your government website is its ability to equip citizens with online services that can save them time from coming into the office. Services such as bill payment, employment submissions, program registrations and facility reservations can easily be done online from the convenience of their home or on any mobile device. An engaging CMS solution that provides online services prevents the need for multiple third-party software integrations.

A CMS functions as a repository for every type of content made available to citizens. Partnering with multiple vendors to plug-in tools for document access, video players, photo galleries, calendars, etc. can lead to integration incompatibilities, and vendor management headaches. Working with one CMS to handle all digital requirements allows your website administrator to self-manage all content types using one intuitive solution with one contact for support services.

There are several providers out there that specialize in developing websites. However, the needs of businesses and bloggers is far different than that of a city or county. Having a CMS and website provider specifically focused on local government's needs can save you a lot of time, money and headaches in the end.

1.3.1 Drupal

Developing a website from zero is difficult job that requires a lot of time. Using a CMS simplifies your job as web developer. There are lots of content management systems that provides many features. Of course all of them have advantages and disadvantages. Few examples of CMS that you might heard about is Wordpress, Joomla and Drupal. Those are the most popular CMS at the moment, firstly because they are open source. This project is based on Drupal, naturally you might want to know why, the answer is simple:

"Simplicity is the ultimate sophistication." - Clare Boothe Luce

The biggest disadvantage of Drupal is hard to learn, but when you get used to work with Drupal, you will be able to do amazing things that cannot be implemented in other content management systems.

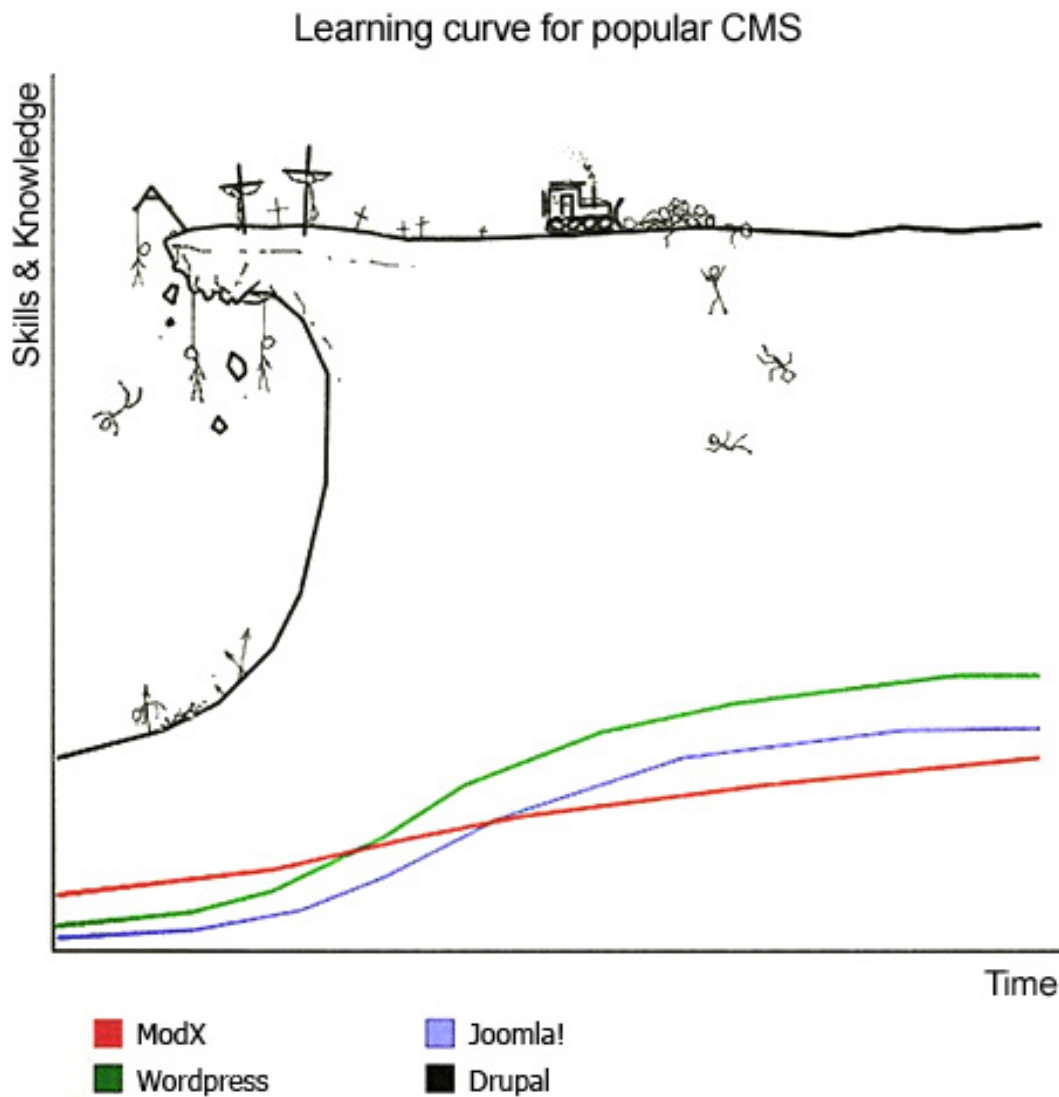


Figure 1.1 – CMS learning curve

From content manager's perspective it will be simple and intuitive to use features provided by developers. There are some standard requests from government and developing those requests takes time. The main goal of this project is to automate and simplify the process of creating new website for Moldavian Government.

1.3.2 mGov

Introducing the result of this project, mGov is a Drupal distribution developed specifically to meet Moldavian Government website guidelines. mGov uses Drupal 8, the latest and most advanced version of the Drupal CMS platform. Give your users a cutting edge digital experience with the power of open source. This latest version of Drupal gives you the option of custom website CMS functionality and integrates with third party tools when you need them.

mGov meets all Moldavian Government website requirements, so when you install mGov you can be confident that your Moldavian Government CMS is equipped to grow with future developments in Drupal 8 and beyond.

Here are some of the mGov's CMS platform specifications:

- it is complete open source and follows Drupal security and coding standards;
- pre-packed installer with Drupal functional modules and front-end themes;
- completely extensible with any Drupal 8 modules or custom modules developed for specific features;
- it is designed to give you full control over installation and management;

Saves your time and effort mGov saves your time and effort by allowing to move quickly to create a seamless website experience with mGov’s ready to go features. From downloading to hosting, from transferring content to customising and organising your Government website, you’ll quickly speed ahead.

mGov can be installed on the hosting platform of your choice in under 10 minutes, and provides a fully featured Drupal 8 website out of the box, while meeting all Moldavian Government website standards. The advanced content management features will help you to collaboratively keep track of documents, page changes and other day to day activity with ease.

Focus your time and effort on creating engaging content and customised features. Because your Government Drupal site comes fully loaded with a core set of user interface elements, functionality and features - these can be reused as the basis for any new Moldavian Government website.

For small Government websites, the majority of requirements have been met in the out of the box version of mGov. Once it’s installed, you’re ready to go. For larger, more complex websites, you will have all the tools you need for customising and enhancing the features on your website.

Improved Performance Improve the performance of your website as you go. Get detailed insights into what works and what doesn’t for your users, then easily update and change anything and everything you need to.

To analyse how your website performs, simply synchronise with Google Analytics out of the box, or easily extend it with modules for most major analytics systems.

It makes sense to have full control of your website, and mGov is a Government website CMS that is entirely yours with no strings attached. You can configure all content without compromise, including terms and conditions, accessibility, contact details and copyright notices. With just a few clicks, your navigation can be easily restructured and new sections added.

1.4 Comparing with other systems

mGov was refined to meet the specific requirements of the Moldavian Government, from perspectives of features this project is unique, yes there are similarities with other projects that are developed for other governments, but each government have their own list of requirements. Drupal is pretty widely used for building government websites and almost all of them are not open source.

An example of analogous system is govCMS, which is a distribution that combines Drupal Core and Drupal modules to enable the quick creation of Australian government websites. Content

management and website hosting for government. Making it easier for agencies to create modern, affordable, responsive websites to better connect government with people.

1.4.1 Similarities and differences

govCMS and mGov are built on Drupal open source software, so any new functionality developed for one agency can be shared across all governmental CMS websites. From security perspective, mGov govCMS proactively tested and maintained to meet government standards and fend off threats to agency websites.

Provides agencies with a starting point and tools to create accessible websites. Out of the box, mGov and govCMS core functions and features are WCAG AA compliant and follow best practice service design based on the Digital Services Standard external resource. With government website audiences increasingly mobile and new devices appearing all the time, mGov and govCMS provides a responsive and adaptable solution without any effort on your part. Sign a Memorandum of Understanding with the Department of Finance to commence work on your agency website.

Both systems provide publications, media and event management functionality that can be used to deliver announcements and information to the website audience. Both systems lets agencies create customised workflows with built-in permissions. Workflows retain content from all previous versions of each page and can be retrieved if required. Pages can be set to automatically publish and unpublish on specific dates. Each system has ministerial sites which agencies can use as a basis for your site. Both system lets content authors and approvers preview the content as it would appear on the website, prior to publishing to the live site. Also both of them provide the ability to develop custom forms, such as ‘contact us’ and ‘feedback’ forms to gather basic information from website visitors. These systems allows users to gather website analytics by embedding the Google analytics code. Both of them includes a ‘What You See Is What You Get’ (WYSIWYG) content editor, giving content authors the ability to develop and edit content in a way that is familiar, such as the format used by Microsoft Word. Agencies can implement their own standards compliant themes and customise their branding. govCMS provides a list of features which agencies can choose to use or not, simply by configuring those systems. Optimise your website for search engines: create and set custom URL paths, create custom page titles, update sitemap.xml and include specific key-worded site content.

2 Designing the System

Using diagrams and charts that will help you to understand the structure and relationships in the system and how it was planned to be from the beginning.

The Core is the heart of Drupal, it represents a standard release of Drupal which encloses fundamental features to build a basic site. As you can see Drupal Core encircle Themes, User Permissions, Blocks, Modules and Nodes.

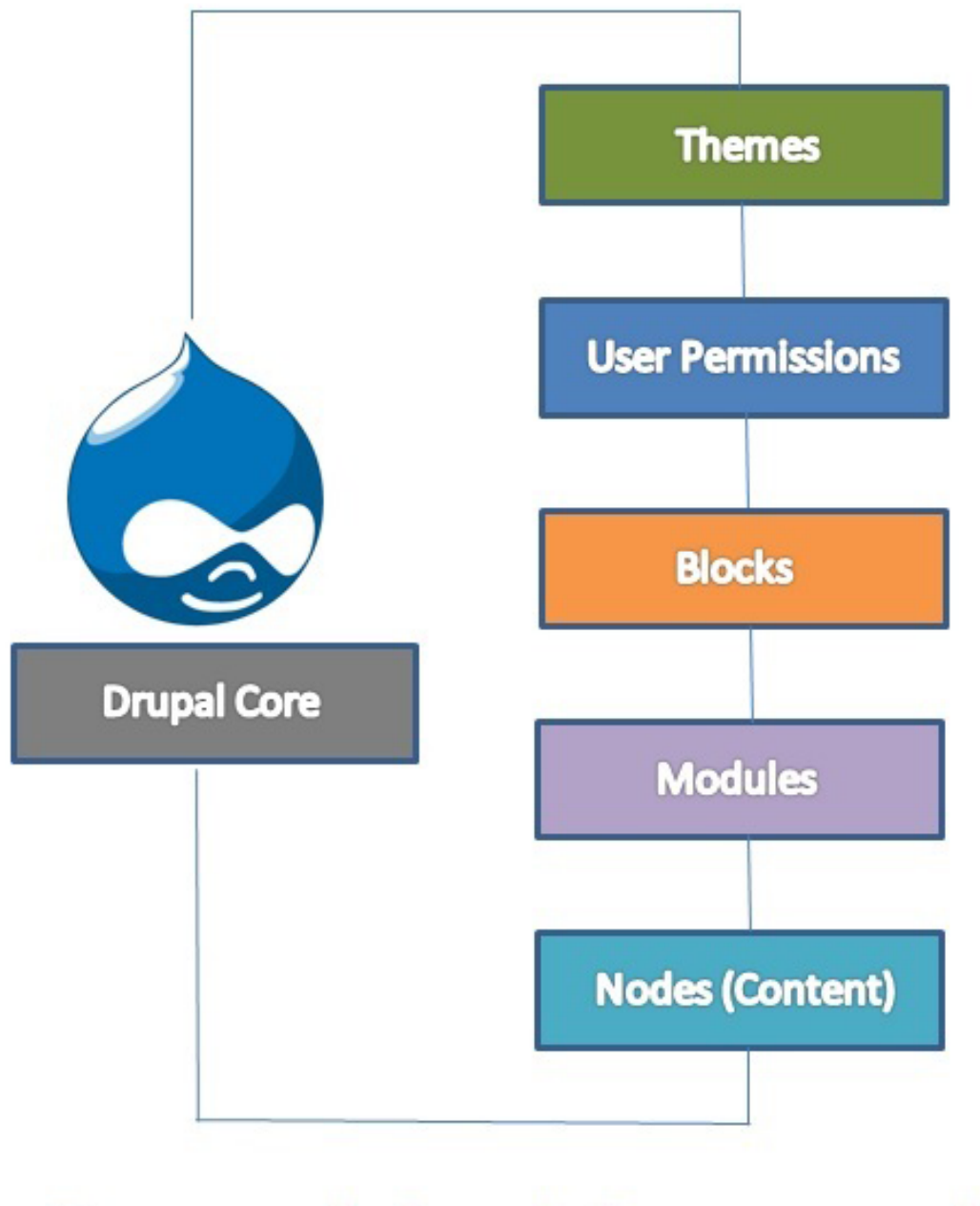


Figure 2.1 – Drupal Architecture Layers

2.1 Themes

A theme defines your site's user interface (UI). It is a collection of files that are interpreted by a Theme Engine to create the desired visual output. Here is used Javascript and jQuery to code

visual elements. To understand how Drupal works, you need to understand Drupal's page serving mechanism.

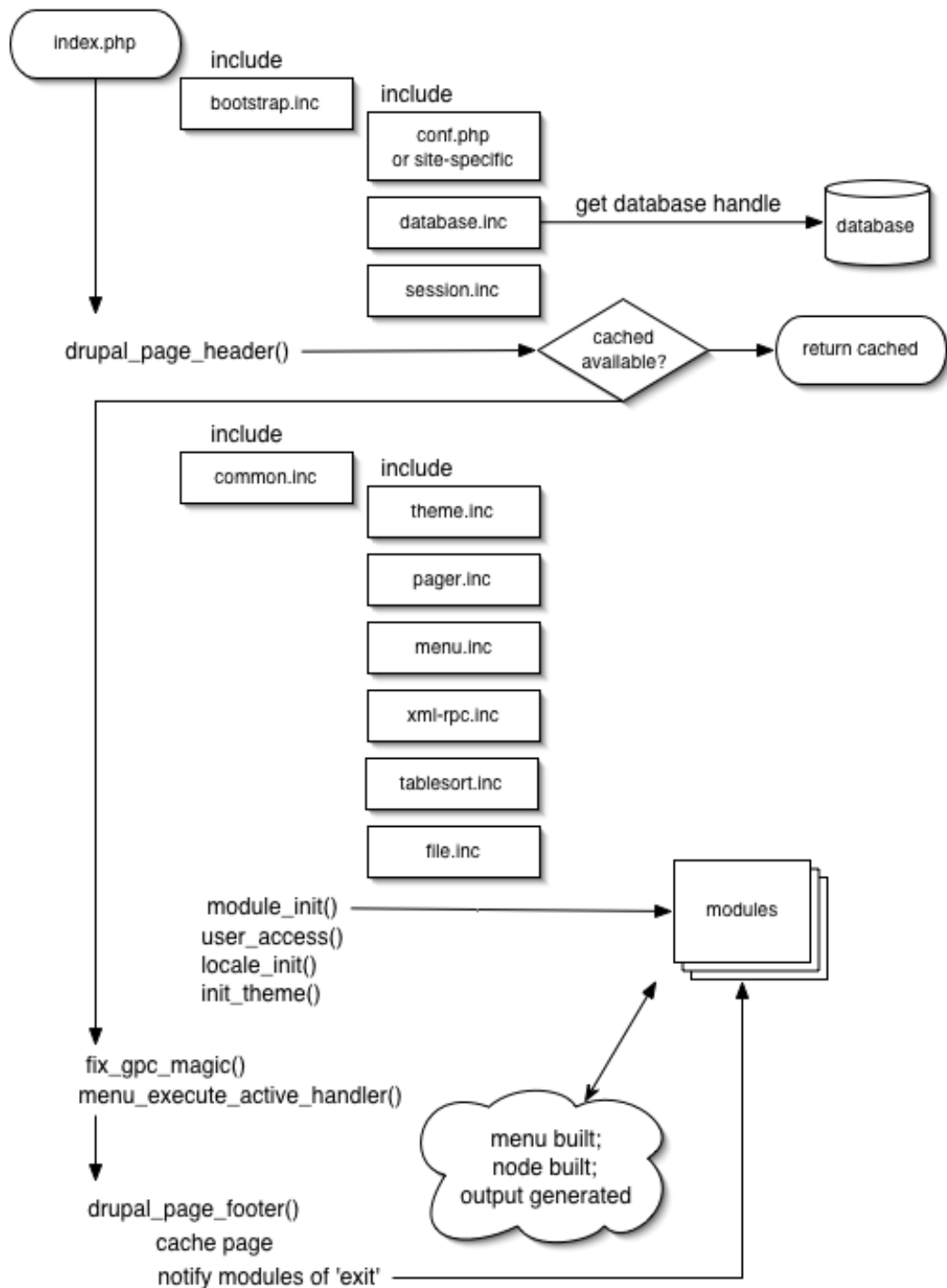


Figure 2.2– Drupal Page Serving Mechanism

In short, all the calls/urls/requests are served by `index.php` which loads Drupal by including various include files/modules and then calling the appropriate function, defined in module, to serve the request/url.

2.1.1 The Bootstrap Process

Drupal bootstraps itself on every request by going through a series of bootstrap phases. These phases are defined in `bootstrap.inc` and proceed as described in the following sections.

2.1.2 Initialize Configuration

This phase populates Drupal's internal configuration array and establishes the base URL (`$base_url`) of the site. The `settings.php` file is parsed via `include_once()`, and any variable or string overrides established there are applied.

2.1.3 Early Page Cache

In situations requiring a high level of scalability, a caching system may need to be invoked before a database connection is even attempted. The early page cache phase lets you include (with `include()`) a PHP file containing a function called `page_cache_fastpath()`, which takes over and returns content to the browser. The early page cache is enabled by setting the `page_cache_fastpath` variable to `TRUE`, and the file to be included is defined by setting the `cache_inc` variable to the file's path. See the chapter on caching for an example.

2.1.4 Initialize Database

During the database phase, the type of database is determined, and an initial connection is made that will be used for database queries.

2.1.5 Hostname/IP-Based Access Control

Drupal allows the banning of hosts on a per-hostname/IP address basis. In the access control phase, a quick check is made to see if the request is coming from a banned host; if so, access is denied.

2.1.6 Initialize Session Handling

Drupal takes advantage of PHP's built-in session handling but overrides some of the handlers with its own to implement database-backed session handling. Sessions are initialized or reestablished in the session phase. The global `$user` object representing the current user is also initialized here, though for efficiency not all properties are available (they are added by an explicit call to the `user_load()` function when needed).

2.1.7 Late Page Cache

In the late page cache phase, Drupal loads enough supporting code to determine whether or not to serve a page from the page cache. This includes merging settings from the database into the array that was created during the initialize configuration phase and loading or parsing module code. If the session indicates that the request was issued by an anonymous user and page caching is enabled, the page is returned from the cache and execution stops.

2.1.8 Language Determination

At the language determination phase, Drupal's multilingual support is initialized and a decision is made as to which language will be used to serve the current page based on site and user settings. Drupal supports several alternatives for determining language support, such as path prefix and domain-level language negotiation.

2.1.9 Path

At the path phase, code that handles paths and path aliasing is loaded. This phase enables human-readable URLs to be resolved and handles internal Drupal path caching and lookups.

2.1.10 Full

This phase completes the bootstrap process by loading a library of common functions, theme support, and support for callback mapping, file handling, Unicode, PHP image toolkits, form creation and processing, mail handling, automatically sortable tables, and result set paging. Drupal's custom error handler is set, and all enabled modules are loaded. Finally, Drupal fires the init hook, so that modules have an opportunity to be notified before official processing of the request begins.

Once Drupal has completed bootstrapping, all components of the framework are available. It is time to take the browser's request and hand it off to the PHP function that will handle it. The mapping between URLs and functions that handle them is accomplished using a callback registry that takes care of both URL mapping and access control. Modules register their callbacks using the menu hook (for more details, see Chapter 4).

When Drupal has determined that there exists a callback to which the URL of the browser request successfully maps and that the user has permission to access that callback, control is handed to the callback function.

2.1.11 Processing a Request

The callback function does whatever work is required to process and accumulate data needed to fulfill the request. For example, if a request for content such as `http://example.com?q=node3` is received, the URL is mapped to the function `node_page_view()` in `node.module`. Further processing will retrieve the data for that node from the database and put it into a data structure. Then, it's time for theming.

2.1.12 Theming the Data

Theming involves transforming the data that has been retrieved, manipulated, or created into HTML (or XML or other output format). Drupal will use the theme the administrator has selected to give the web page the correct look and feel. The resulting output is then sent to the web browser.

2.2 User Permissions - Use Case Model

Giving rights to right people is important because usually non technical people such as content managers are messing in the site trying to do what they do not know. Control what users do on your website by assigning them User Permissions to access features on your site. User Permissions provides an interface for giving additional permissions to individual users without the need to assign them to a special role. When this module is enabled, users with the 'administer permissions' permission can access the 'User Permissions' tab on each user's account.

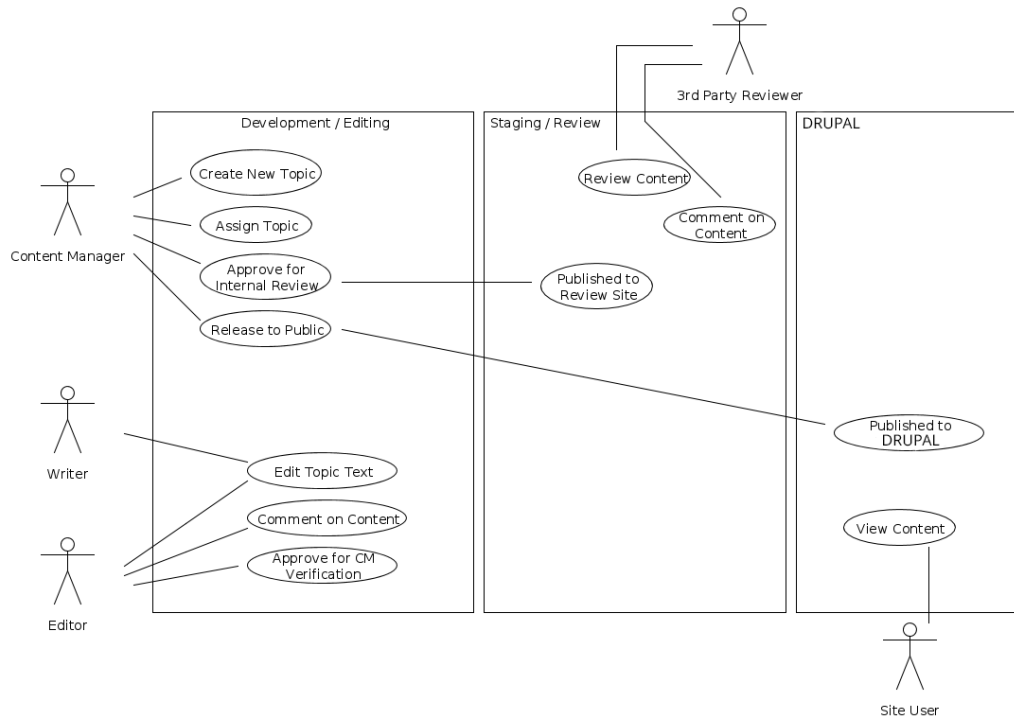


Figure 2.3– Use Case

One of the great features of Drupal is the ability to control how and what people can access on your site. You can set permissions for these users to define who can do what for Drupal core features and contributed modules. For example, you probably won't want casual visitors to edit your homepage. However, the site owner or trusted user should be able to do so.

Drupal allows you to setup any number of different kinds of users or Roles. Many websites have editor and site administrator roles; editors to make content updates and site admins to install new modules and make larger configuration changes.

2.3 Blocks

A block represents the boxes that appear in a Drupal site. Configure blocks to customize menus, footers and other widgets type sections. Blocks in Drupal 8 are actually made up of two separate API structures to create a user experience similar to what Drupal has maintained in past iterations. These two APIs are the Block Plugin API, which is a stand-alone reusable API and the Block Entity API which is a Drupal 8 specific use case of block placement and visibility control.

Blocks are made available to your site most commonly by enabling core or contributed modules. Once created, a Block can be modified to adjust its appearance, shape, size and position - or which Website pages it appears on. For example, enabling the core Poll module makes the "Most Recent Polls" block available for you to place in a region. Also note that some modules provide multiple blocks when enabled, others may not define new blocks.

2.4 Modules - Sequence Model

A Module adds functionality to Drupal's core. Hooks are used to ensure modules interact smoothly with the core.

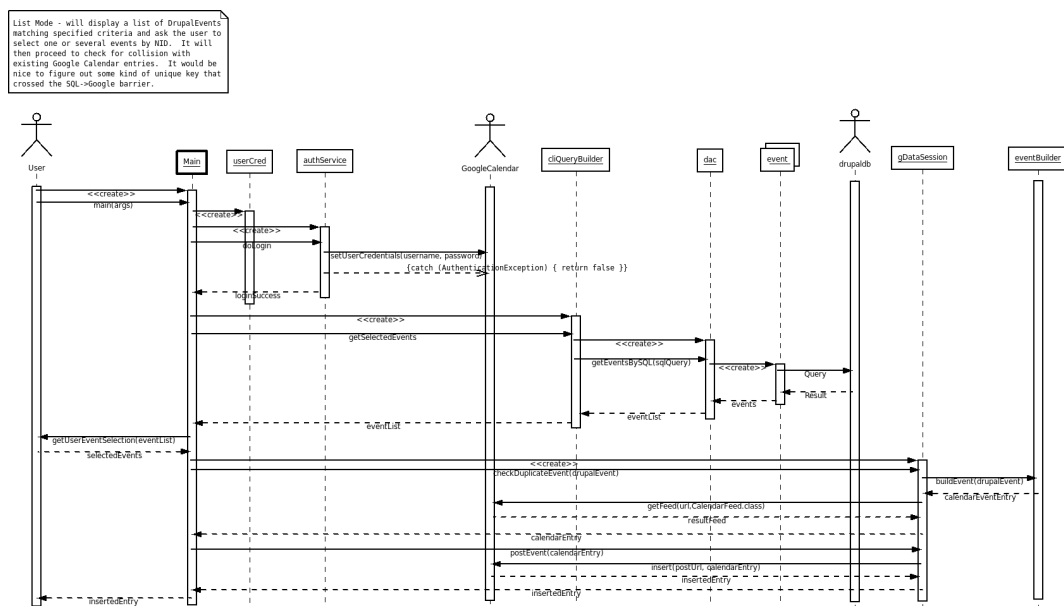


Figure 2.4 – Sequence Diagram

In the previous sequence diagram represents a list of Drupal Events matching specified criteria and ask the user to select one or several events by NID. It will then proceed to check for collision with existing Google Calendar entries. It would be nice to figure out some kind of unique key that crossed the SQL and Google barrier.

2.5 Nodes

Content on Drupal sites are represented by Nodes. A node can take the avatar of any content type a web page, poll, blog entry, form or newsletter. In reality this would probably have several other states but for the sake of this example.

2.5.1 State Machine Model

A state machine is defined by the list of possible states and the event/condition that triggers each transition. In this simple ticket example the states are in progress, approval, and finished. The transitions are completed, rejected, accepted.

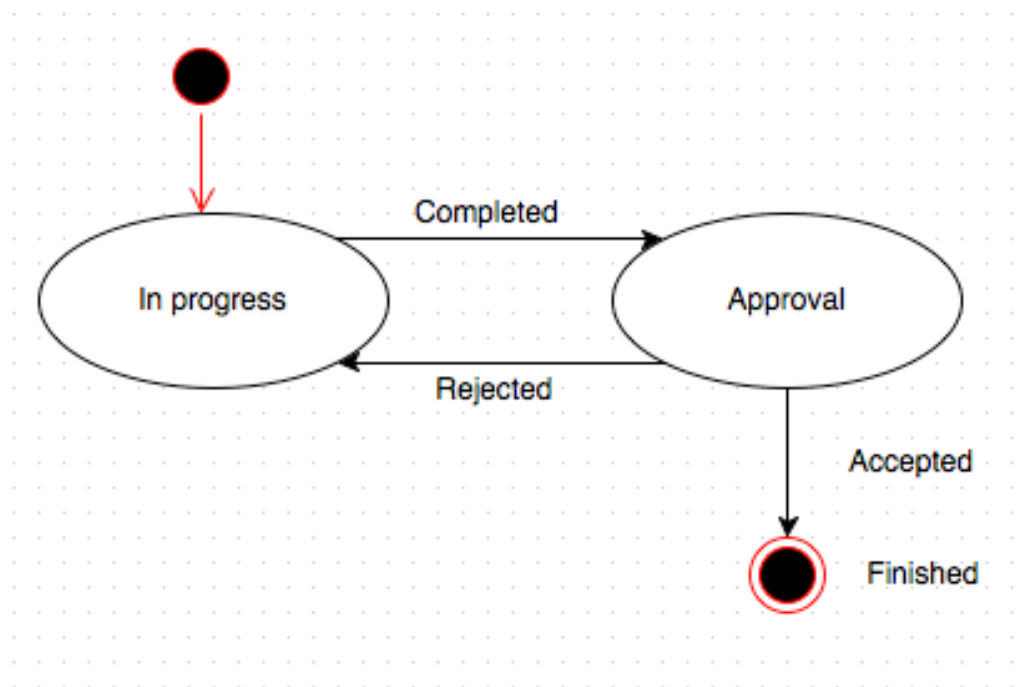


Figure 2.5 – State

Drawing out a state machine diagram to model this kinds of problems can be really useful to help identify any edge case scenarios you may not have thought of, and capture them early in the design process. It also shows you exactly what you need to test further along in the site build.

As with anything in Drupal there are several ways to achieve this functionality, in fact there's even a State Machine module, but that relies on creating custom plugins. Workbench Moderation and various other workflow modules include a state machine implementation for a specific purpose. In Drupal we'll be using a simple list field to store the list of possible states for the node.

In Drupal 7 we need a module to help us do this. In this case we are adding a field that will never be directly edited by the user so we just deny access to edit that field using the Field Permissions module.

For the simple ticket example, we have 3 states. So use an integer list field with the following allowed values:

- In progress
- Awaiting approval
- Finished

State machine was defined by the set of possible states implemented by list field, and a set of transitions. These transitions can be implemented using the Rules Link module. Using the Rules Link module you can add a button to the ticket node which manipulates the state value preventing the user from actually editing the value in the state field directly, and thus enforcing the workflow defined in our state machine. Each 'Rules link' is configured in two parts. First you define the conditions for when the link should be visible using standard Rules conditions. Secondly, you use the rules reaction to set the value of the state field to the new value and perform any other actions that you want as a side effect of the transition.

2.5.2 Class Model

The taxonomy mechanism is the heart of what makes Drupal so different from most other content management systems. But experience in the drupal support channel shows it is not always well understood.

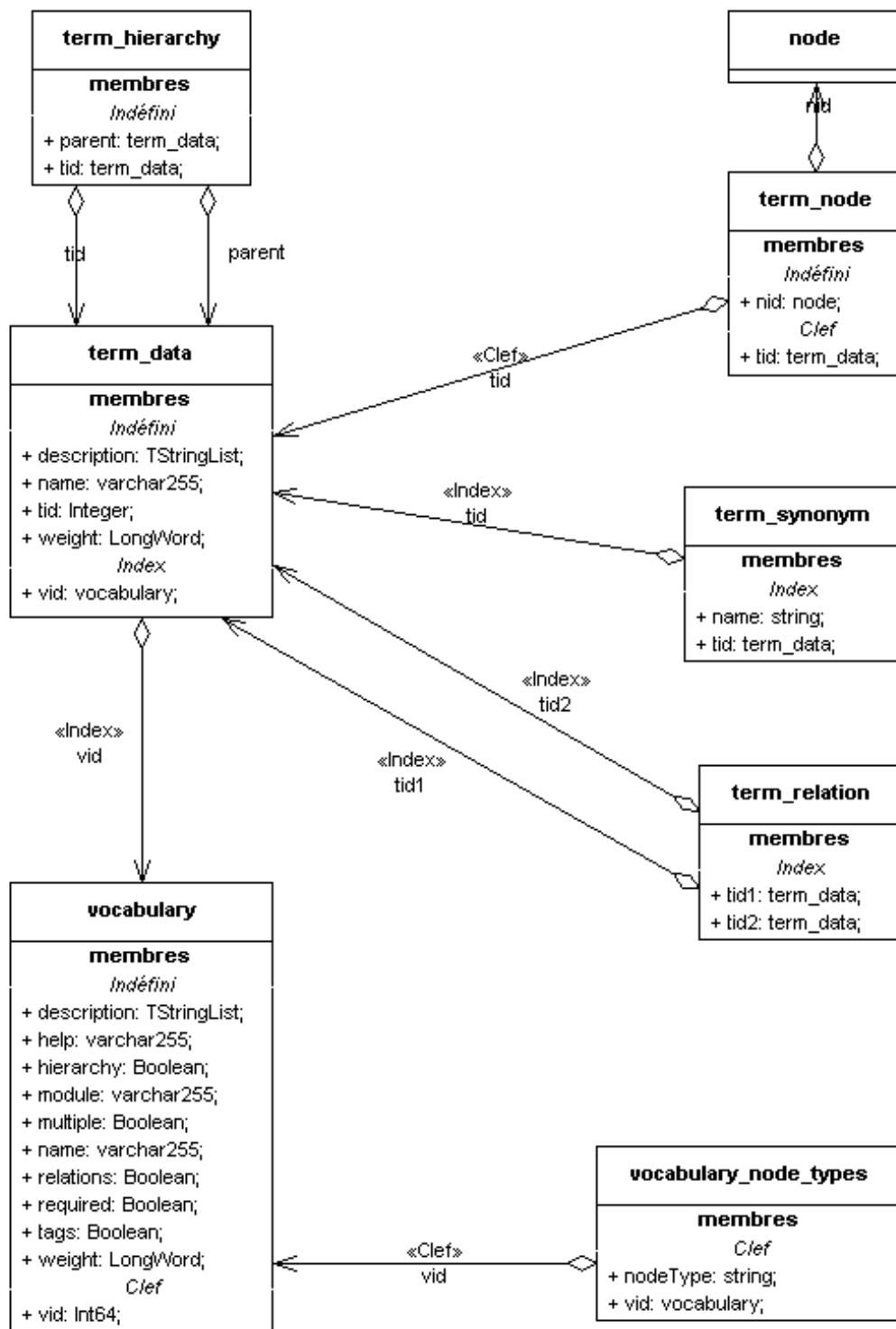


Figure 2.6– Class Diagram

First things first, although the service provided by the taxonomy mechanism is simple, the implementation requires no less than six tables for basic features:

The main table is `term_data`. This is where the terms used for classification are defined. Every term is given a unique term identifier, or `tid`. The second most important table is `vocabulary`. Each of the terms in `term_data` belongs in exactly one vocabulary, to which it is linked by the `vid` column. For vocabularies allowing it, term hierarchy is defined, obviously enough, in `term_hierarchy`, in which each `tid` has one row for each of its parent `tids`, or one row with parent `tid` 0. Terms are mostly used to classify the basic unit of content in Drupal, the node. This is the purpose of the `term_nodetable`, which implements the `termnode` relationship. Note that they can be used for other purposes like user classification. Synonyms are handled through the use of the `term_synonym` table, in which each row links to an existing `tid` and defines a new name for it. For vocabularies in which this option has been enabled, the `term_relation` allows for the inter-linking of terms: each row defines a pair of `tid` values as describing related terms. Drupal allows vocabularies to be limited to some node types. This is implemented by the `vocabulary_node_types` lookup table

There is an implied integrity relationship : `nodetype` must match `vocabulary_node_typesnodeType` for every instance of `term_nodetid`. This is currently implemented in code by drupal modules. In the current implementation each `tid` will also have at least one row in this table, with `parent = 0`, to show it is a "root"node. It may have more than one parent, which prevents replacing the `term_hierarchy` table by just a `parent` column in the `term_data`table. Since this is essentially an implementation artefact that costs significant data space, it does not seem poised to remain in place for very long.

As questions on the support channel suggest, the use of drupal categories, as implemented by the taxonomy module may not be guided enough: in case where the user wants additional code to prevent terms in one category to be used on a node along with other terms from another category on the same node. In most cases, this points to an information architecture problem at a higher level: if terms are mutually dependent, like these terms that had to be exclusive, then they belong in the same classification axis, meaning the same vocabulary.

This is where the hierarchical nature of Drupal classifications comes in handy: instead of defining a set of specialized vocabularies with dependence on other vocabularies, all it takes is for one to define a hierarchical vocabulary, within which specialized subtrees will be implemented as children of higher level terms, thus ensuring mutual exclusion. In short, if there is one only word to remember when designing an information taxonomy, or in layman's terms when configuring categories on Drupal, this word is orthogonality. Proper orthogonal category design will often save a lot of time implementing case-specific rules in code. Although the taxonomy system in Drupal is geared towards use in nodes, it can be put to other uses. As a proof of concept, Karoly Negyesi has created the user's tag module enabling the use of taxonomies on users. Use your favorite search engine to query for drupal user's tag for the current URL. This module uses a `term_datauser`, similar in purpose to the `term_node` table in regular taxonomy use. Note that this is NOT supported code, or even contributed code, and as such should not be used on a production system unless you are ready to maintain it or have it maintained.

3 Implementation, development methodologies

mGov is meant to be used as a starter kit for building a custom governmental site. But before starting working with Drupal, you need to understand it. Therefore will continue with a guide that will give you an overview of Drupal, helping you to determine if drupal is a good fit for this project.

3.1 Meeting Drupal

To clarify the difference between Drupal and other CMSs, consider the example of a news site. You want to be able to post news articles on the site, and you want the homepage to have a section featuring the five most recent ones. Next, you decide that you want to add a blog section, and put a list of links to the five most recent blog entries on the homepage as well.

If you were using an ordinary CMS, first you would install a plugin that handled news articles and could put short blurbs on the homepage. Next, you would install a plugin that would track the latest blog posts and put a list of those on the homepage. Each plugin would only be responsible for tracking and managing a particular kind of content, and each would remain relatively isolated from the others.

But, what happens when you have that brilliant, middle-of-the-night idea to blend these two functions by showing a list of blog posts about the latest news items, ordered according to contributor activity? If you're using a "toy truck" CMS, you may be out of luck. Or, you may need to hire a developer to write a custom plugin from scratch. But through the power of the Drupal way, the way of manageable abstraction, you can accomplish this task quickly and easily. Since Drupal's modules do things in a standard way and interface with a common underlying system, building all sorts of clever, customized features is just a matter of snapping parts together. In this example, you could just use Views.

Of course, this flexibility comes at a certain cost. While a toy truck is instantly understandable and ready to use without much thought, a modular vehicle construction kit will, by nature, require you to read the instruction manual first. The building blocks are available, but you'll need to learn how they fit together before you can take a paper prototype and turn it into a full-featured website.

Drupal core, and the thousands of contributed modules that build on it, requires an initial investment to learn, but mastering the Drupal way is immensely rewarding; the passionate community is a testament to Drupal's power to liberate site builders from the simplicity/flexibility dilemma. Once you've tried Drupal, you'll likely leave your toy truck and boat in the closet to gather dust.

3.1.1 How Drupal works?

People often think of a website as a collection of static pages, perhaps with some functions like a blog or a news engine thrown in to round it out. When they go to manage their site, they are thinking in terms of a tree-like hierarchy of pages that they will edit.

Drupal, however, treats most content types as variations on the same concept: a node. Static pages, blog posts, and news items are all stored in the same way, and the site's navigation structure is designed separately by editing menus, views lists of content, and blocks side content which often

have links to different site sections.

It's a lot like the separation you find in standards-compliant page coding—XHTML provides the meaningful structure of the information, while CSS arranges it for presentation. In Drupal, nodes hold the structured information pertaining to a blog post such as title, content, author, date or a news item, while the menu system, as well as taxonomy tagging of content and views, creates an information architecture. Finally, the theme system, along with display modules like Panels, controls how all this looks to site visitors.

Since these layers are kept separate, you can provide a completely different navigation and presentation of your content to different users based on their specific needs and roles. Pages can be grouped differently, prioritized in a different order, and various functions and content can be shown or hidden as needed.

3.1.2 Nodes

We don't talk about nodes every day, but since they are at the heart of Drupal's design, they deserve further investigation. Essentially, a node is a set of related bits of information. When you create a new blog post, you are not only defining its body text, but also its title, content, author link, creation date, taxonomy, etc. Some of these elements will be shown by the theme layer when the node is displayed. Others are meta-data that control when the node will show up at all such as taxonomy or publishing status.

As suggested before, you aren't limited to a single way of presenting your site's content. You can define many navigation schemes, custom themes, or designs for the site. You can look at some contributed themes [here](#).

Comments also illustrate the Drupal way. Comments are typically thought of as part of a blogging system, but there isn't a separate "blogging system" in Drupal. Drupal simply manipulates nodes to function in a manner that most people think of as a blog. But comments can be enabled on any content type you choose be it blog posts, news items, book pages, or any other type you may create. Drupal's modular system is limited only to the imagination of the site builder.

3.1.3 The Core

Creating an informational website that broadcasts from 'one to many' is something that most CMSs do right out of the box. Drupal shines, however, by empowering site users to create content and to interact with each other moving from 'one to many' to 'many to many'.

With some CMSs, you can set up a blog, and you can install plugins to handle having a community of users. But what happens when you want to give individual blogs to each of your users, sorting their contents so that they can be displayed individually with their own skins, while also generating cross-blog topical digests, top five lists, and links to elaborate, customized user profiles? What if you also want to integrate those blogs with forums, a wiki-like environment, and user-owned galleries of tagged photos? A typical CMS's approach to information makes such a scenario very difficult to implement. In contrast, the Drupal way makes such a scenario not only easy to establish, but also incredibly manageable over time.

Drupal is designed from the ground up so site builders can delegate content creation, and even site administration, to users. All a site builder has to do is define user permissions for which users get to do what, and then everyone can start collaborating.

3.1.4 Fast and Flexible

Drupal's flexibility is incredible, but installing it is surprisingly easy. With a simple FTP upload and a few short web-based configuration questions, you can connect with your database and have your first Drupal site up and running within an hour.

Pick one of the included themes, and just start adding content. Do you want to have visitors log in? Then you should switch "authentication" on or off. Want to switch on some of the included tools? Then you should turn on 'forums'. Enable commenting on node types. Turn on the book module for wiki-like collaboration, create forms and polls, use taxonomy to give site content structured, hierarchical categorization or free-form tagging.

Do you want your own skin applied to the site? Drupal's theme system uses the Twig templating system allows you to insert dynamic content without needing any raw PHP. Drupal's generated markup is clean, standards-compliant XHTML. No old-school tables.

3.1.5 How it flows

If you want to go deeper with Drupal, you should understand how information flows between the system's layers. There are five main layers to consider:

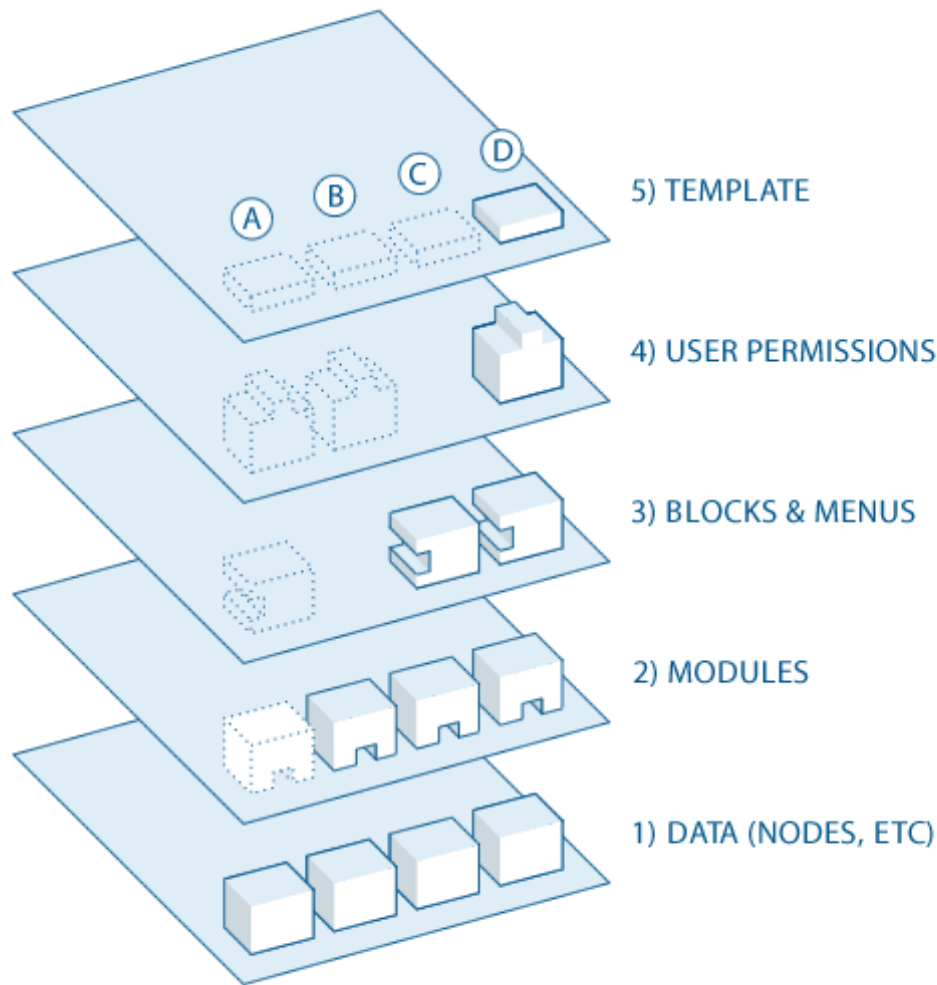


Figure 3.1– Drupal Flow

- At the base of the system is the collection of nodes—the data pool. Before anything can be displayed on the site, it must be an input as data.
- The next layer up is where modules live. Modules are functional plugins that are either part of the Drupal core (they ship with Drupal) or they are contributed items that have been created by members of the Drupal community. Modules build on Drupal’s core functionality, allowing you to customize the data items (fields) on your node types; set up e-commerce; programmatically sort and display content (custom output controlled by filters you define); and more. There are thousands of different options within the fast-growing repository of contributed Drupal modules. They represent the innovation and collaborative effort of everyone from individuals to large corporations.
- At the next layer, we find blocks and menus. Blocks often provide the output from a module or can be created to display whatever you want, and then can be placed in various spots (Regions) in your template (theme) layout. Blocks can be configured to output in various ways, as well as only showing on certain defined pages, or only for certain defined users. Menus are navigators

in Drupal, which define the content coming in each defined menu path. Menus are a core element of Drupal which provide links to all the pages created in Drupal.

- Next are user permissions. This is where settings are configured to determine what different kinds of users are allowed to do and see. Permissions are defined for various roles, and in turn, users are assigned to these roles in order to grant them the defined permissions.
- On the top layer is the site theme. This is made up predominantly of XHTML and CSS, with some Twig variables intermixed, so Drupal-generated content can go in the appropriate spots. Also included with each theme is a set of functions that can be used to override standard functions in the modules in order to provide complete control over how the modules generate their markup at output time. Templates can also be assigned on-the-fly based on user permissions.

This directional flow from bottom to top controls how Drupal works. Is some new functionality you want not showing up? Perhaps you uploaded the module into the system but have not activated it yet, and this is making everything downstream non-functional.

Maybe the module is installed and activated, but you still don't see what you want on your site. Did you forget to place the block, as in "B"? Or are your user permission settings conflicting with what you want and your users are not set to see the output as in "C"?

Additionally as mentioned earlier getting the kind of granular control you want over the details of the XHTML module outputs requires understanding this flow. Are you using a module that does exactly what you want, only you wish the markup was just a little bit different? Maybe you'd like it to use different tags, or you'd like to assign a CSS class to something? You accomplish this by copying the output function from the module and pushing it up to the functions document in your theme. Modify the code there, and when the system goes to output, it will see your customized function and use that instead.

3.2 Distribution

mGov represents a Drupal distribution, which is a full copy of Drupal that include Drupal Core, along with additional software such as themes, modules, libraries, and installation profiles. There are two main types of Drupal distributions:

- full-featured distributions: complete solutions for specialized use cases.
- other distributions: quick-start tools, starting points for developers and site builders.

3.2.1 What is a distribution?

Setting up a Drupal site typically involves downloading and configuring Drupal core, then downloading and configuring various contributed modules, as needed. To make this process easier, there are a variety of pre-configured versions of Drupal you can download and use for specific types of sites, for example a blogging site, a conference site, a corporate Intranet site, and so on. These 'pre-configured' versions of Drupal are called 'distributions'.

With a 'full-featured' distribution, you can quickly and easily set up a site for a specialized purpose, such as academic, business, government, nonprofit, publishing, social, in just a few steps:

- choose your Drupal distribution.
- install it on your web server.
- configure it and enable the desired features from your website's Administer section.

3.2.2 Distributions and installation profiles

Installation profiles are what a developer creates as the basis of distributions. They define installation steps such as enabling modules, defining content types, etc.) that run after Drupal's base installation when you first install Drupal. One or more standard installation profiles are included in the Drupal Core download; developers can create custom profiles that set up Drupal for specific purposes, and optionally release them for community use on Drupal.org. It is not always easy to attempt to use an installation profile directly, if it requires non-core modules, themes, or libraries – you would have to locate and download all the required components yourself before you could install Drupal. Instead, it's a lot easier to download a full distribution (if available).

Distributions are full copies of Drupal that include Drupal Core, along with additional software such as themes, modules, libraries, and installation profiles. The automatic packaging scripts on Drupal.org turn installation profiles into distributions, by gathering all the modules, themes, and libraries they require into a single zip archive, so that all you need to do is download the full archive and run the install script.

3.2.3 When we need distributions?

There are no hard and fast rules about when to use distributions, but here are a few guidelines:

- evaluating Drupal: If you're just getting started with Drupal it makes sense to try a distribution, since they are easier to set up and you can see real-life examples of what Drupal can do. Not all distributions are equal though, so start with a popular well-maintained distribution.
- demoing Drupal: Before building a site for someone it can be useful to show them examples of how Drupal can be configured.
- quickly building a site: If you're building a site similar to one provided by a distribution it makes sense to start with a distribution. After installing you can continue to configure the site, add modules, create themes, etc. You can even undo things that the installation profile script may have done. If you're changing too much, though, it may make more sense to just start with stock Drupal and build from there, rather than try to undo and change what was set up for you.
- learning Drupal: In the administration section of a distribution, you can study the inner workings of a real example Drupal site, learn all the details about how it is built and configured

to obtain specific functionality, and experiment with any changes and additional modules and themes.

3.2.4 Creating Drupal Installation Profile

Installation profiles in Drupal 8 have all the functionality of modules, including access to hooks and plugins and, critically, the ability to provide configuration for your site in the form of .yml files. First, you need a machine name for your profile. This is a name consisting of lowercase letters and underscores only, in our case is mgov_kit.

Your installation profile will reside in its own profilename directory in the /profiles directory of a Drupal 8 site. All installation profiles must have a profilename.info.yml file. They may also have:

- profilename.profile
- profilename.install file
- config folder
- translations folder

When packaged, your installation profile will also have modules, src, and themes directories as needed.

The profilename.info.yml file should look similar to this:

```
1 name: Profile Name
2 type: profile
3 description: 'Description of your profile.'
4 core: 8.x
5
6 # Optional: Declare your installation profile as a distribution
7 # This will make the installer auto-select this installation profile.
8 # The distribution_name property is used in the installer and other places as
9 # a label for the software being installed.
10 distribution:
11   name: Distribution Name
12   # If you want your distribution to pre-select a language you can specify
13   # the language code, as well. This will prevent the user from selecting
14   # a language code in the user interface. While not recommended for generic
15   # community distributions, this can be very useful for distributions that
16   # are tailored to a language-specific audience (for example government
17   # institutions in a certain country) but also site-specific installation
18   # profiles. Note that this language code will also be used when installing
19   # the site via Drush.
20   langcode: de
21
22 # Required modules
23 dependencies:
24   - node
```

```

25 - history
26 - block
27 - block_content
28 - breakpoint
29 - color
30 - config
31 - comment
32 - contextual
33 - contact
34 - quickedit
35 - help
36 - image
37 - options
38 - path
39 - taxonomy
40 - dblog
41 - search
42 - shortcut
43 - toolbar
44 - field_ui
45 - file
46 - rdf
47 - views
48 - views_ui
49 - editor
50 - ckeditor

```

Listing 1 – .info.yml file

```

1 <?php
2 /**
3  * @file
4  * Install, update and uninstall functions for the profilename install profile.
5  */
6
7 /**
8  * Implements hook_install().
9  *
10 * Perform actions to set up the site for this profile.
11 *
12 * @see system_install()
13 */
14 function profilename_install() {
15     // First, do everything in standard profile.
16     include_once DRUPAL_ROOT . '/core/profiles/standard/standard.install';
17     standard_install();
18
19     // Can add code in here to make nodes, terms, etc.

```

```
20 }
```

Listing 2– .install.yml file

The `profilename.profile` file has access to almost everything a normal `Drupal modulename.module` file does because Drupal is fully bootstrapped before almost anything in the profile runs.

```
1 <?php
2 /**
3  * @file
4  * Enables modules and site configuration for a standard site installation.
5  */
6
7 // Add any custom code here like hook implementations.
```

Listing 3– .profile.yml file

Drupal 8 installation profiles can contain configuration files. You can start by taking the configuration directory (config folder) of an installed, configured site and copying it into the `config/install` folder in your profile.

Once that's in place, there are some other required tasks:

- Copy all of the modules and themes listed within `core.extension.yml` into your profile's info file (using the new info file's format).
- Delete `core.extension.yml` (and possibly some other config files).
- Remove all of the UUIDs from your config files so that they don't conflict with those of new sites. This can be done quite easily on the command line like so all on one line:

```
1 find /path/to/PROFILE_NAME/config/install/ -type f -exec sed -i '' -e '/^uuid: /d' {} \;
```

Listing 4– .info.yml file

If you just want to grab an existing site's configuration, and don't need to end up with a formal installation profile (for sharing on Drupal.org, for example), you can use the Configuration Installer installation profile to install a new site from the configuration of another site.

3.3 Custom Modules

Drupal 8 leverages a number of advanced PHP language features and sophisticated 3rd party libraries in order to present 3rd party developers with the most advanced API of any CMS available. While experienced Drupal 7 developers may see some significant changes, much of the basic structure will remain familiar.

If any of the material presented while going through the D8 module developer's walk-through is new to you, then the material below may help. However, a comprehensive knowledge is not necessary to proceed to the D8 module walkthrough.

3.3.1 Object Oriented Programming

OOP, while initially daunting, is now well established as a best practice. For a general overview of PHP best practices, read through phptherightway.com. Drupal doesn't make use of all the different tools and techniques documented there but it does serve as a great introduction to PHP and the language's many features.

3.3.2 Symfony 2

Symfony 2 is a PHP framework that Drupal borrows from in order to reduce code duplication across various PHP projects. Much of the code that Drupal 8 uses to handle routing, sessions and the services container, amongst other things, is borrowed from Symfony 2.

A framework is not absolutely necessary: it is 'just' one of the tools that is available to help you develop better and faster! Better, because a framework provides you with the certainty that you are developing an application that is in full compliance with the business rules, that is structured, and that is both maintainable and upgradable. Faster, because it allows developers to save time by re-using generic modules in order to focus on other areas. Without, however, ever being tied to the framework itself. A framework is not an absolute necessity, but nonetheless it is very useful. A framework is a pledge of quality, upgradability and maintainability of applications at lower cost. Applications developed using a framework are interoperable with market standards.

Besides a framework, there are other solutions that are available to develop both Web sites and Web applications: Content Management System and their supplemental modules, as well as packaged professional solutions. If their native features line up perfectly with your immediate and future needs, no question about it. There is no point in reinventing the wheel.

4 Business Plan

This project is intended for Moldavian Government and it's citizens, but the main goal is to facilitate and speed up development process of Governmental sites.

4.1 Schedule

In this section we will describe the schedule for the project to get finished in time. Schedule will be composed of following steps: determination of objectives, determination of the amount of work and tasks division, implementations of task.

4.1.1 Objectives

The main goal of every company is to earn a profit by selling products or services that it provides. Similarly, the developers aim to have a substantial profit from their invested efforts in creating and launching the system. The main economic goal was to create a system that would collect fees from its users, and to satisfy all needs and expectations from the project.

4.1.2 SWOT Analysis

Performing the SWOT analysis over a system gives a brief overview about expectations or possible problems that can appear during the lifetime of the system. It is mandatory to identify the SWOTs because they can inform later steps in planning to achieve the expected objectives. In Table 4.1 it is represented the strategic planning method, used to evaluate Strength, Weaknesses, Opportunities and Threads that can involve the given system.

Table 4.1 – SWOT analysis

Strengths	Weaknesses
<ul style="list-style-type: none"> – No competitors on local market; – Accessible and portable; – It is expected to be highly profitable; – Easy maintainable; 	<ul style="list-style-type: none"> – Systematic database updates needed; – Require internet connection in order to access; – Not fully responsive on smartphones;
Opportunities	Threats
<ul style="list-style-type: none"> – Ensures with access 2 types of users (teachers, students); – Nice solution for personal time management; 	<ul style="list-style-type: none"> – Heavy loads at each database update; – Account registration for teachers not fully secured;

Referring to the SWOT analysis as the basis of the promotion/development of any product, each company has the opportunity to foresee the possible profit as well as possible wastage. One of the most dangerous factors that can be prevented with SWOT is the risk of concurrency, which also has an important role in market development and the increase of product's quality. To predict other negative effects, any company must have a well-designed plan, like having monthly statistics, which would contribute at diminishing economical threats.

4.1.3 The time meant for the accomplishment of the project and the schedule

To schedule the working plan, first of all it is necessary study all information necessary for its implementation, realization and then to divide the whole process of implementation in small steps, called activities, which will be evaluated individually. Only in such a way can be scheduled the more correct plan, which would lead to the in-time accomplishment of the project.

The duration of the project's execution can be represented in the Formula 4.1 below:

$$Duration = startdate - finishdate + reservetime(days) \quad (4.1)$$

where: start date, finish date– indicate the days in which activities take place; time reserve – the number of extra-schedule days the developer has.

According to the schedule presented in the Table 4.2 expected time of development is 93 days.

Table 4.2 – Projects Schedule

Nr.	Activity Name	Activity Duration (days)	Worker
1	Define the concept	7	Software Developer Project Manager
2	Research and descriptive design preparing		Software Developer System Architect
3	Schematic representation of the future software system		Software Developer System Architect
4	Informational system design. Use Case diagrams design		Software Developer System Architect
5	Sequence and collaboration diagrams' design based on the Use Cases		System Architect
6	Interface design Class Diagrams design		Software Developer System Architect
7	Adding necessary components and documenting the requirements for the programmer	5	Software Developer Project Manager
8	Graphical components' design, related to system's functionality		Software Developer System Architect UI Designer
9	Programmers documentation on the project	6	Software Developer
10	Systems' execution (writing the code) Writing tests for the code	35	Software Developer Tester
11	Database design	5	Software Developer
12	Technologies' implementation	8	Software Developer Tester
13	Preventive testing. Inserting modifications in case of errors	4	Software Developer Tester
14	Writing the report about project's performance	7	Project Manager UI Designer
15	Documenting the final system	3	Project Manager UI Designer
16	Adding the last features Making the calculation for the new product	1	Software Developer Project Manager UI Designer
	Totally days to develop the system	93	

Table 4.3 – The expenses on the materials used in the project development on long term

Nr.	Name	Unit's price (MDL)	Qty	Totally (MDL)
1	PC	9000	1	9000
2	Printer	1500	1	1500
	Total			10500

Table 4.4 – Non-material expenses

Nr.	Name	Unit's price (MDL)	Qty	Totally (MDL)
1	Windows 10	2500	1	2500
2	Enterprise Architect	1500	1	1500
	Total			4000

4.2 Economical Proofs

In economy in actual to bring economic proofs for the IT projects, based on the specific of the concurrencies in economic relationships, which suppose a wide research space. In the conditions of a low degree of determination of the marketing environment, of high prices' volatility, decreased degree of prognoses depth, a common business-plan doesn't allow the exact foreseeing of the final results of the business. In this context, one of the basic instruments is choosing the methods, the right positions and indices for the economical proofs.

4.2.1 Material and non-material expenses

For any specific project it is necessary to do the calculations that need the prognosis of the necessary amount of money for the expenses. Direct materials cost is the cost of direct materials which can be easily identified with the unit of production. The manufacture of products or goods required material as the prime element. In general, these materials are divided into two categories. These categories are direct materials and indirect materials. In table 4.3 and 4.4 below are presented the material and non-material expenses that arise during its development.

Direct expenses stated in table 4.5 will include logistics products that will be used for development cycle of the project.

Table 4.5 – Direct materials costs

Nr.	Name	Unit's price (MDL)	Qty	Totally (MDL)
1	Flash Drive	250	1	250
2	Office paper (500 sheets)	50	2	100
3	Paper clips	0.5	4	2
4	Pen	3	3	9
5	Cover	35	1	35
	Total			957

Table 4.6 – Salary expenses

Nr.	Position	Amount of work (days)	Individual price (MDL/day)	SUM (MDL)
1	Web Developer	83	120	9960
2	Project Manager	23	110	2530
3	UI Designer	16	120	1920
4	Tester	47	115	5405
Total pay off for all the workers				19815
Social Fund (23 %)				4557.45
Medical Assurance (4 %)				792.6
Totally work remuneration				25165.05

4.2.2 Salary expenses

For the job done, the workers are paid off according to the tariff plan presented. In Table 4.6 where we have presented the pay of for a working day. In the table there is also included the working period of each specific worker.

Social and medical insurance funds express the amount of money added to the total pay off. Social fund represents additional 23% . Medical insurance rate is approved every year by “Compulsory medical insurance fund” law. Therefore, rate for 2014 is 4%. The amount of expenses on work remuneration was calculated using Formula 5.2:

$$TR = TP + SF + MA = 19815 + 4557.45 + 792 = 25165.05 \quad (4.2)$$

where: TR is total remuneration, TP is total pay off, SF is social fund expenses and MA is medical assurance. After calculations we obtain TR equal with 25165.05 (MDL).

4.2.3 Indirect expenses

The fixed means pay of is the partial loss of the consumable properties and value of the means during their usage, influenced by different factors and the increase of the work productivity. The importance of indirect costs is the calculation of depreciation and amortization. For accounting purposes, depreciation should be calculated evenly throughout the project activity. Besides the expenses for the printer, PC, etc, there must be included the expenses for electricity, internet. These expenses are also related to the project, because they are extremely necessary in its development. We take into account the working hours at PC 8 hours per day. The energy expenses are due to the PC usage and the illumination of the workplace during the project development. The number of hours the PC worked raises to

$$744h(93days * 8h)$$

We'll take into account that PC use

$$400W/h$$

$$Totalpowerusage = (400 * 744)/1000 = 297kW + 200kW(usedelectricity) = 497kW$$

Table 4.7– Indirect expenses

Name	Measure	Tariff, MDL	Quantity	SUM, MDL
Energy, kW	kW	1.99	497kWh	989.03
Internet, subscription	month	150	4	600
Rent office	month	400	4	1600
Mobile services	min	1.5	150	225
Totally				3414.03

. The calculation of the indirect consumption for the utilities is presented in the Table 4.7.

4.2.4 Wear calculation

4.3 Project Cost

4.4 Financial results

4.5 Conclusion

Conclusions

Elaborating the inClass project, several steps were accomplished first. Before the actual development was started, the main goal was to understand entirely the problem that needed to be solved, by providing an optimal solution to it. Nowadays, we are living in the era of technological progress, where machines get to do the most part of the jobs people once used to do entirely. That's why, it is vital to sustain the every day life improvements human kind gets while creating software, instead of using the ancient approach of doing everything individually. This is a thing that our country understood and during the last few years, more and more state structures are continuously developing their foundations, by providing reliable, modern technologies for the every day use.

The current application refers to a specific type of state structures, that are local universities. These entities are providing a high rate of technological progress during the last years, thing that encourages the stuff and also the students and teachers. Given the fact that it has been seized a continuously growing interest in determining people to work with software, now becomes much more realistic to reorganize the academic structures. This means that people already reached the moment when are ready to bring software products in universities in order to optimize the routine processes which they are responsible for accomplishing. This is a big step that has to be adjusted properly to the university's needs.

inClass is the software product that comes with a project in development and a prototype to fix an existing problem in universities. At the same time, it represents a small change for the academic infrastructure, given the fact that is easy to implement, use and maintain, while the results are consistently improved. The application was implemented as a web platform, assuming that universities don't have the problem of free connection to the internet anymore. So, in order to access the application, it is enough to own a smartphone and Wi-Fi. It represents the perfect solution for avoiding the general timetable, which is so uncomfortable to reach and use.

For a better representation of the application, a full analysis over the system was done first. It consists of UML diagrams, which have the purpose of visualizing the application from different points of view. After this schematic overview of the system is done, there remain no pending questions about the functionality provided by the application. Once things get clear, it is time to start implementing those functionalities. The technology used for developing the application is a powerful one. Ruby on Rails is one of the most high rated frameworks at the current moment, which allows the developer to implement various functionality with a minimum of written code. Also, it allows the use of gems, which simplify the task even more.

Finally, assuming that the project is finished, it is necessary to provide an user guide, for easing the tasks of the actual users. There are several moments that need to be paid attention to, moments that were clearly specified in the content of the current document. The main thing to be mentioned is that the application allows 3 types of users: administrator, teachers and student. In order to use the application it is necessary to register in the system. Registration is specific for each type of user.

At the moment, the provided solution is unique on the local market, thing that is convenient from economical point of view. There are necessary less expenses to take into account in order to finish the project successfully. The less expenses are, the less the market price will be, which makes

the software available for use for local universities. In terms of economical analysis, the numbers were pretty impressive. If computing all the possible expenses that could occur while developing the project with some more improvements, the team would also need to find sufficient clients, that would guarantee a low price of the product and more than 18 % as profit.

Also, is a good idea of coming with some future possible improvements for the application. The most recent improvement would definitely be the adjustments of the application to support sorting timetables by current weak, meaning that the even and odd weeks would be take in account. Of course, this is a requirement that is not valid for all universities, but it is still very important for the ones that have different schedules for even and odd weeks. Another improvement could be done for the student/teacher side of the application. At the current moment, the teachers and students are allowed only to visualize data. Assuming that the current project it is a prototype, this is acceptable. Still, the idea of having several tools available for checking the current status of a certain day or semester, by using progress bars and other visual components, would enlarge the boundaries of activity for these types of users. Another feature to be implemented could be the possibility of interaction between student and teachers, while discussing some updates of the courses via a special page for announcements. Many other improvements can be done once the universities get to the point when they implement the system as part of their infrastructures.

References

- 1 MISR, *Measuring the Information Society Report*, <http://www.itu.int/en/ITU-D/Statistics/Documents/publications/misr2016/MISR2016-w4.pdf>
- 2 govCMS, *SaaS Developer Guide*, 2013, <https://github.com/govCMS/govCMS/wiki/SaaS-Developer-Guide>
- 3 govCMS, *govCMS Product and Services Guide*, 2014, https://www.govcms.gov.au/sites/g/files/net2111/f/govcms_product_and_services_guide_2.pdf
- 4 Drupal, *Users, roles and permissions*, <https://www.drupal.org/node/120614>
- 5 Drupal, *Hiding Form Fields in Drupal 8*, <https://drupalize.me/blog/201404/hiding-form-fields-drupal-8>
- 6 Drupal, *Project Modules*, https://www.drupal.org/project/project_module
- 7 Symfony, *Documentation*, http://guides.rubyonrails.org/active_record_basics.html/
- 8 UML diagrams, *UML Tutorial*, <http://www.tutorialspoint.com/uml/index.htm/>