## Vivarn Backend SysteM

Based on the requirements, functional features and our experience in CMS implementations, We recommends Drupal 7 as the preferred Content Management System for implementing the backend for this application.

Why Drupal?

* Open Source
* Huge Community
* Highly Extensible
* Over 20000 community modules available for rapid development
* Excellent content management features
* Scalable
* Secure

**Drupal 7** isbased on the Linux Apache MySQL PHP (LAMP) stack and has been developed using PHP and MySQL as its default database. Drupal performs best when hosted on the Linux OS. While Red hat Enterprise Linux 6 is recommended, Drupal can also be hosted using the community supported clone CentOS 7 or Ubuntu Server edition.

**Highlighting features of Drupal that will used to achieve our requirement:**

* Users Management
* Access control
* Custom content types
* Management Data + content
* Restful web services.
* Configurable widgets/blocks

## Drupal Cms Features To Be USed

We have provided a brief description of each feature that is off-the-shelf with Drupal core or community modules. The following appended list has been included in the context of the requirements discovered by going through the requirements provided by Vivarn web application. In addition, we have recommended some features that will positively affect user experience (front-end and admin) on the site:

#### User registration and login

Out of the box (OOTB), Drupal handles user login. User registration is supported, and email notifications are

automatically sent to users when accounts are created.

Drupal also supports authentication through community modules for

* Open ID
* LDAP

#### Custom content types

Drupal provides the ability to create custom content types easily. Fields can be added to any content type.

Content creation or update forms are auto generated by Drupal. Forms can be themed as necessary.

This is a feature for developers and site builders only.

#### Content categorization or taxonomy

Content in Drupal can be categorized using taxonomy fields associated with content types:

* New vocabularies can be created through the administration interface and associated with content

types. Marketing will be able to categorize content at the time of creation or updation.

* Content categorization can be predefined and available through select fields or can be text tags with

#### Access control

Drupal provides a robust, role-based access control mechanism. Permissions can be configured by an administrator or a site builder. Access is granular, and access can be controlled at content type level and field level.

#### Content search

Drupal OOTB provides content indexing and search. Drupal also has modules to integrate with Apache Solr,

which provides advanced search capability such as full text search and faceted search and better scalability.

Apache Solr is an open source and license free enterprise search platform. Its major features include:

* Full-text search
* Hit highlighting
* Dynamic clustering
* Database integration
* Rich document handling
* Distributed search and index replication
* Plugin-based customization
* Unicode support

#### Developer-friendly features

* Drupal provides several developer-friendly features such as the following:
* Easy extensibility: Drupal 7 provides a number of hooks, which can be implemented in custom modules

to extend core functionality without touching core code.

* Drupal UI templates are separate from the module code that makes theming for a site easy. Templates

for individual pages, blocks and content types can be overridden.

* Theme switching to handle mobile devices
* Developer-friendly code generator modules
* Modules to manage all site configurations in code, which assists in better configuration management
* Form API for building secure forms
* Workflow engine to build a custom workflow
* Services API that can be used to expose functionality as services
* Features module allows developers to store CMS configuration in a configuration management tool.

#### Scalability and performance

In addition to the inbuilt Performance Management modules that Drupal provides, we can implement certain

additional modules such as the following:

* Zend Opcode Cache: This caches the interpreted PHP code thus reducing the time for interpreting every request.
* Varnish: Varnish can be used as a web accelerator to cache static content and deliver it extremely fast.
* Memcache: Memcache can be used to cache the heavy database (DB)-related queries.

#### Web analytics

Web analytics can be achieved by using a third-party tool such as Google Analytics. Alternatively, there is also a

Google Analytics module for Drupal that adds the GA web statistics tracking system to your website.

#### Unicode support

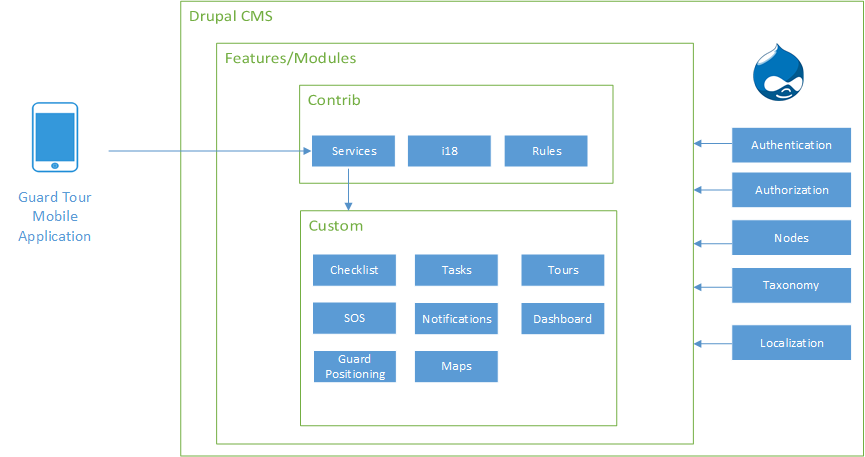
Drupal uses UCS Transformation Format (UTF)-8 for encoding all the data, and this is a Unicode encoding, so it

can contain data in any language. Also, when Drupal imports external XML data (such as RSS or XML-RPC), it is

automatically converted to UTF-8.

## logical Architecture and custom functionality

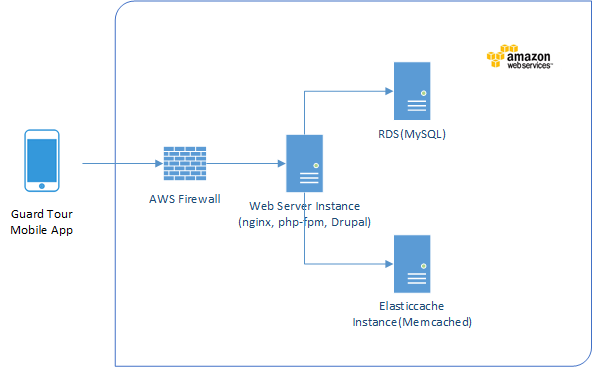
Based on our understanding of the requirements and the envisioned solution, here is a high-level logical representation of the solution.



* The backend system will be built using Drupal 7
* The following custom modules/functionality will be built
  + Checklist
  + Tasks
  + Tours
  + SOS
  + Notifications
  + Admin Dashboard
  + Maps
  + Guard Position

## Deployment Architecture

For scalable websites and portals we usually recommend AWS as the preferred medium for deployment. Based on the scalability requirements, instance sizes will be decided during the design phase.



The following instances will be required

* Application Server/Web Server instance running nginx, PHP and Drupal
* Database or RDS instance running MySQL
* Cache Server instance running Memcached server will be used for caching and improving site performance

The above diagram represents the production environment. In addition to the above there will be development and staging environments which will be replicas of the production environment. These will be smaller instances than the production instance

* Code and Configuration will be promoted from Development->Staging->Production
* Developer integration will be done on the development environment
* QA/UAT testing will be done on the staging environment.
* Code will be stored in a configuration management repository (Git) on the development environment.
* The development environment will also run Jenkins to run continuous integration builds and run checks on code quality.

We are also open to on premise deployments and similar setup can be created in an on-premises environment.

## Tools & Technologies

The following tools and technologies will be used in this project. The versions are valid for on premise deployment. Amazon cloud maintains/decides its own versions

|  |  |  |
| --- | --- | --- |
| Type | Name or vendor | Version |
| Operating system | Linux | CentOS 7 |
| Web Server | NGINX | 1.9.1 |
| Development language | PHP | 7 |
| CMS | Drupal | 7.43 |
| Database | MySQL | 5.7 |
| User Based Caching | Memcache | 1.4 |
| Configuration Management | Git | Latest |
| CI | Jenkins | Latest |
| Code Quality | Drupal Coder module |  |