



Pasture Picker Study Report



Version: 1.2

Release State: Final

Released: 30/05/2014

Confidential

we live IT

Table of Contents

1. Introduction	1
1.1. Purpose of Document	1
1.2. Background.....	1
1.3. Scope	1
1.4. Context	2
1.5. Deliverables	2
1.6. Terms and Abbreviations	2
1.7. References	3
2. Business Drivers	4
2.1. Business Problem	4
3. Technology Drivers	5
3.1. Enterprise Environments.....	5
3.2. Policy	5
3.3. Scalability	5
3.4. Availability.....	5
3.5. Integration.....	6
3.6. Usability	6
3.7. Accessibility	6
3.8. Supportability	6
3.9. Disaster Recovery.....	6
4. Issues 7	
4.1. Redesign of Pasture Picker application.....	7
4.2. Hosting of the new Pasture Picker application.	7
4.3. Commercialisation of the Pasture Picker application.....	8
5. Architectural Model	9
5.1. Application Architecture	9
6. Delivery Estimates	11
6.1. Application Development	11
6.2. Hosting	11
Appendix UI Design (Wireframes).....	12

Revision History

Issue	Date	Description	Name
0.1	29/4/2014	First draft	Jacky Zhu
0.2	22/5/2014	Updated as per review feedback	Jacky Zhu
1.0	27/5/2014	Finalised document.	Jacky Zhu
1.1	29/5/2014	Updated wireframes	Jacky Zhu
1.2	30/5/2014	Final review	Alex Pavlov

1. Introduction

1.1. Purpose of Document

This document describes the study results of the aspects of the Pasture Picker application. It provides a high level architectural view of the business strategy and requirements, and discusses the issues relating to the Architecture, UI and evaluates solution options, which include application and infrastructure components.

It is firmly in the solution space and “funnels” a broad range of inputs into a single document that leverages off current practice to focus on the major issues. This document defines how the solution will be built including the major elements that are required.

1.2. Background

The Pasture Picker was developed to provide a powerful resource for selecting pasture species suitable for local conditions. The tool targets producers, advisors and agribusiness, and enables users to identify the species most appropriate to a particular application in a particular environment, as defined by selection criteria.

Object Consulting has been requested to provide the services of an experienced research and e-extension provider to develop a future strategy for possible re-design and re-development of the online Pasture Picker Selection Tool (Pasture Picker).

1.3. Scope

The scope of the study includes:

1. the costs associated with re-designing the user interface of the tool as well as re-developing its technical framework, providing sufficient flexibility for changes and new information to be added on a regular basis;
2. feasible and practical recommendations on hosting models, including a possible integration with other existing and planned tools such as the PVTN database and tool;
3. an appropriate commercialisation strategy whereby seed companies and other interested parties will be invited to co-invest in the ongoing maintenance/updating of the tool – including license fees, systems for



uploading data etc. This is likely to include workshops with key owners and industry stakeholders to scope out the opportunities.

1.4. Context

The Pasture Picker Study and the PVTN tool development are two projects running around the same time. Although they are both focusing on giving advice on the selection of pasture species, they serve at different levels: While the PVTN tool provides access to quantitative (production) data of a number of pasture species and varieties in different trial locations, the purpose of the Pasture Picker would be to provide in-depth information about a much larger range of pasture species that grow in Australia. Used together, the tools can help producers select suitable “fit for purpose and location” pastures, and subsequently boost productive capacity.

It has been agreed that the integration of the two applications will be simple hyperlink at this stage.

The wireframes for the Pasture Picker tool have links to the PVTN built in. Linking back from the PVTN tool to the Pasture Picker tool can only be achieved once the tool has been built and published.

1.5. Deliverables

The primary deliverables of this project are:

- Pasture Picker Study Report (this document).
- Wireframes for the new UI design.

1.6. Terms and Abbreviations

Term / Abbreviation	Description
CMS	Content Management System. A software that allows business users to manage the website content without assistance from technical personnel. A more detailed description can be found at the following URL: http://en.wikipedia.org/wiki/Web_content_management_system
UI	User Interface
W3C	World Wide Web Consortium (W3C). An international community that governs the web standards.



1.7. References

List all relevant documents, including requirements definitions and those which contain general domain knowledge.

2. Business Drivers

2.1. Business Problem

The previous version of the Pasture Picker application (<http://pasturepicker.com.au>) was developed and hosted by University of Queensland.

There were a few issues with the application, primarily around the usability. All references to the tool were removed from the MLA website in mid-2013, and the actual tool was taken offline in early 2014.

The redesign and rebuild of the tool is subject to the feasibility study, recommended solution and funding.

As the domain name (pasturepicker.com.au) was not renewed, it has been taken up by an unrelated organisation. A new domain name needs to be secured for the new pasture picker application. At the time of writing, domain name pasturepicker.org.au is still available, which could be a good fit for the new application.

3. Technology Drivers

The technical drivers are anything that determines the nature of the technical solution. Think of them as requirements that are being placed on the solution because they are needed to meet the requirements. This section is providing essential guidance to later sections and to other documents. Often the answer is not simple and a discussion of options, attributes and approaches needs to be included.

3.1. Enterprise Environments

Although the Pasture Picker application will be managed by MLA, it is understood that it is desirable to have the application running outside of MLA's enterprise environment.

The Pasture Picker application is to be designed as an independent application, without reliance on resources from MLA's infrastructure or applications.

3.2. Policy

There is no stated policy in the governance of the Pasture Picker application. However, best practice in the architectural and UI design will be followed in the solution design.

3.3. Scalability

It is understood that the load on the previous Pasture Picker application was low to medium. The web site load is usually defined by the peak pageviews, a typical low to medium load is between 1000 to 5000 pageviews per hour during the peak usage.

However, scalability must be part of the consideration of the solution architecture and the hosting options.

3.4. Availability

Availability is the percentage of time that a system is able to provide service. It is about the systems readiness to perform its function on demand. While there are no specific requirements on the availability, the solution aims to



deliver high availability through the hosting on a capable public cloud platform, which commits to a high standard of availability.

3.5. Integration

The integration with the external applications/websites will be via hyperlinks. The external websites include:

- PVTN
- MLA's public website
- Partners' public websites

3.6. Usability

Usability is a key issue for the UI design, with specific consideration of the following:

- The web based application needs to be easy to use by non-technical users.
- It must support tablet device.

3.7. Accessibility

Accessibility functionality makes an application or resource usable by persons with disabilities. The UI design should take into consideration of the accessibility best practices.

The W3C Web Content Accessibility Guidelines can be found in the following URL:

<http://www.w3.org/TR/WCAG20/>

3.8. Supportability

As the solution is likely to be running outside of MLA's enterprise environment, and the Pasture Picker will have limited technical resources, the solution must keep the technical support required to a minimal level.

3.9. Disaster Recovery

The architecture must consider the need for DR support, preferably delivered by the hosting solution.



4. Issues

4.1. Redesign of Pasture Picker application.

Issue: The new Pasture Picker application must be a user friendly web-based application that supports mobile devices (tablets), and should be easy to maintain by non-technical users.

The options for the application development are:

- Complete custom development of a web application. This option provides higher flexibility, however, it is usually at a high cost, and more difficult (and costly) to maintain.
- Customisation of a CMS driven web application. A Content Management System is software that allows business users to manage the website content without assistance from technical personnel. A more detailed description can be found at the following URL: http://en.wikipedia.org/wiki/Web_content_management_system
This option leverages the out-of-box CMS functionality, and provides an easy to maintain web application, with good mobile device support. However, the application may need to be designed with the constraints set by the CMS product that is to be chosen.

Conclusion: Given the limited resources that can be committed to the Pasture Picker development and maintenance, customisation of a CMS driven web application is the best option for Pasture Picker.

4.2. Hosting of the new Pasture Picker application.

Issue: A hosting environment needs to be selected for the new Pasture Picker application. The hosting environment needs to be scalable, and support the CMS technology that is to be selected to support the application.

The options are:

- Use MLA's web environment to host the Pasture Picker application.
- Use a public cloud hosting environment.

Conclusion: Given the desire to have the Pasture Picker application running outside of MLA's infrastructure, it is obvious that a public cloud hosting environment is preferred. The actual provider (eg. Microsoft Azure or Amazon AWS EC2) is to be decided once the CMS technology is selected. The rule of thumb is that Azure is preferred for Microsoft .net technology, while AWS is preferred for non-Microsoft technology.



4.3. Commercialisation of the Pasture Picker application.

Issue: There have been some discussions around commercialisation of the tool.

The obvious options are:

- Seed companies and/or other interested parties to sponsor the development and the on-going maintenance of the application. This option is more palatable, as it enables the free access for the producers and other intermediate users (eg. advisers). However, it may cause the false impression that Pasture Picker endorses some of the business entities if their advertising (or even their names) appears on the Pasture Picker web site. This may be against the policy of Pasture Picker and/or the partners.
- A fee for using the application. This option will discourage the take up and the regular use of the tool by the targeted audience.

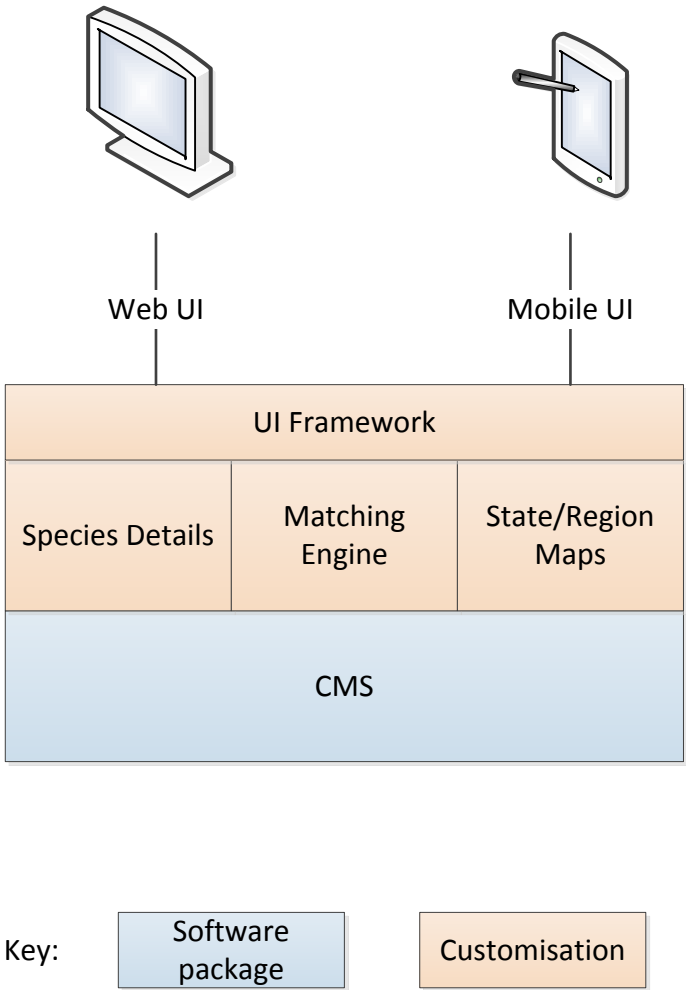
Conclusion: Commercialisation options of the Pasture Picker have to be further discussed and scoped out if this is a desired model.

5. Architectural Model

Application Architecture is produced as part of the study of the Pasture Picker application. Other architecture views, such as physical architecture, will be produced during the implementation phase of the Pasture Picker, after the selection of the hosting provider.

5.1. Application Architecture

The following diagram shows the architecture of the Pasture Picker application.



The solution is realised by customising the CMS implementation to deliver the storage and displaying of the species details, matching the species to the selection criteria, and the maintenance of the state and region maps.

The following components are involved in the solution:

1. CMS. The CMS provides the functionality of managing the content, displaying and APIs for the customisation. Given the requirements of Pasture Picker, a capable open source CMS like Umbraco should be a good fit. The key required features of the CMS are:
 - a. Customisable content type. In order to model the species details, a custom content type with strongly typed attributes is required. For example, there will be attributes like suitable regions (multiple selections) and pH range (numeric) that need to be included to support the selection of species.
 - b. Mobile devices support. Adaptive and/or responsive design must be supported in order to render the content on mobile devices properly. Adaptive design is preferred, as it can help reduce the data load on the devices.
2. Matching engine. This component is responsible of matching the species to the selection criteria.
3. State/region maps. This component is to support a two level maps (state and region), and provides the input to the selection criteria, based on user's selection of state and region.
4. UI. The user interface supports both the web browsers and mobile devices. The aim of the UI design is to have a clean, easy to use application, which is driven by the CMS. The wireframes of the proposed UI design are included in the Appendix.

6. Delivery Estimates

6.1. Application Development

The delivery project involves clarifying the requirements, defining a detailed solution, followed by configuration and development. The project would require the following specialists:

- Solution Designer – extensive CMS and software development experience
- Business Analyst – use case specialist, UI expertise
- Developer – extensive CMS template and UI development experience
- Project Manager

The following is an indicative project plan, which requires further review and adjustment.

Stage	Timeframe	Note
Requirement clarification	1 week	Document use cases, user stories and business processes. Detailed scope will be defined at the end of this exercise.
Design	1 week	Part of the design can overlap with requirement clarification.
CMS Implementation / Development	2 weeks	It is assumed only 2 templates are required.
User Acceptance Test	2 weeks	Dependent on the test strategy and planning.

6.2. Hosting

The on-going hosting of the application is estimated based on the pricing of the Microsoft Azure Cloud hosting, at the time of writing. The hosting costs of other cloud-based platform, such as Amazon AWS are comparable to the Azure cloud.

It is assumed that the traffic on the web application is reasonably low. The estimates are indicative only.



Node	Cost (per month)	Note
Windows VM – Small (1.6GHz CPU, 1.75GB RAM)	\$67.57	Basic VMs, no load balancing required.
SQL VM – Small (1.6GHz CPU, 1.75GB RAM)	\$101.36	SQL Web Edition
Bandwidth - 50GB	\$5.45	Assumed 50GB traffic/month
Total:	\$174.38	

Appendix UI Design (Wireframes)

The following is the wireframes of the UI Design, in PDF format.

