

# PUMS Module Specification

Projects Idea Database

UP732011

<b>Introduction</b>	<b>2</b>
<b>Key Personnel</b>	<b>2</b>
Table 1 - Key Personnel	2
<b>Future Proofing</b>	<b>2</b>
<b>Entities</b>	<b>2</b>
Table 2 - Entities	2
<b>Requirements</b>	<b>3</b>
Functional Requirements	3
Table 3 - Functional Requirements	4
Non-Functional Requirements	4
Table 4 - Non functional requirements	5
Approach	5
<b>Architecture and Designs</b>	<b>5</b>
Architecture	5
Figure 1 - Application Architecture Overview	5
Interoperable API Design (Common API)	6
Table 5 - Initial thoughts table	6
User Journey Diagram	6
Figure 2 - User Flow Diagram	6
Initial Designs	7
Table 5 - Wireframes	8
User Search Criteria Initial Thoughts	8
<b>User Accessibility</b>	<b>9</b>
<b>Timescale</b>	<b>9</b>

# Introduction

The PUMS website has a project search interface but currently the functionality this component is limited to searching and does not permit editing of previously created projects or to delete a project.

This project aims to update the component and add new functionality.

## Key Personnel

Responsible Owner	Jim Briggs
Design and implementation	Peter De Sousa

*Table 1 - Key Personnel*

## Future Proofing

The component is to prioritise interoperability, and maintainability of code with appropriate frameworks, such as maven to support dependency management.

## Entities

Entity	Attributes	Relationships
Person	UserID, Name, Email, Contact Number	
Ideas	UserID	
Projects		
ProjectHistory		

*Table 2 - Entities*

# Requirements

Requirements in the below table are ranked using the MOSCOW technique of:

1. Must Have
2. Should Have
3. Could Have
4. Won't Have

This is to allow for quick reference and rescheduling should this be required.

## Functional Requirements

#	Requirement	Priority	Change request(s)	Scoped Release Version	Notes
1	Projects			1	
1.1	User can submit and idea	Must Have		1	
1.2	User can edit idea	Must Have		1	
1.3	User can delete project	Must Have		1	
2.1	User can search projects, including their own	Must Have		1	
2.2	User can filter projects	Should have		1	
2.3	Filter by user	Could have		2	
2.4	User can sort projects	Should have		1	
2.5	Suggested search	Should have		2	
2.6					
3	Validation				
3.1	Per page validation features	Must have		1	

3.2	Global validation features	Must have		1	
4	Auth				
4.1	Auth must interoperate with other modules	Must Have		1	
4.2	User must be redirected to auth if not logged in	Must Have		1	/auth endpoint w/dummy
4.3	Dummy Auth Module	Must Have		1	
5	Saved Projects	Could Have		2	
5.1	User Project List	Could Have		2	
5.2	Email Notification of changes	Could Have		2	
5.3	User "save to list" functionality	Could Have		2	
6	Documentation	Must Have		1	
6.1	Auth docs	Must Have		1	
6.2	Module Docs	Must Have		1	
6.3	Search docs	Must Have		1	

Table 3 - Functional Requirements

## Non-Functional Requirements

#	Requirement	Priority	Change request(s)
1			
1.1	User interface should be clean	Should have	
1.2	User interface should be easy to use	Should Have	
1.3	User interface should be	Should	

	engaging	Have	
--	----------	------	--

Table 4 - Non functional requirements

## Approach

This project will be approached with a Prototyping methodology due to the learning nature of the project.

As some aspects of the module will be new to the developer a prototype will likely be made for each top-level feature, except documentation.

Issues will be tracked with JIRA and tasks/subtasks created for top-level child tasks respectively.

## Architecture and Designs

### Architecture

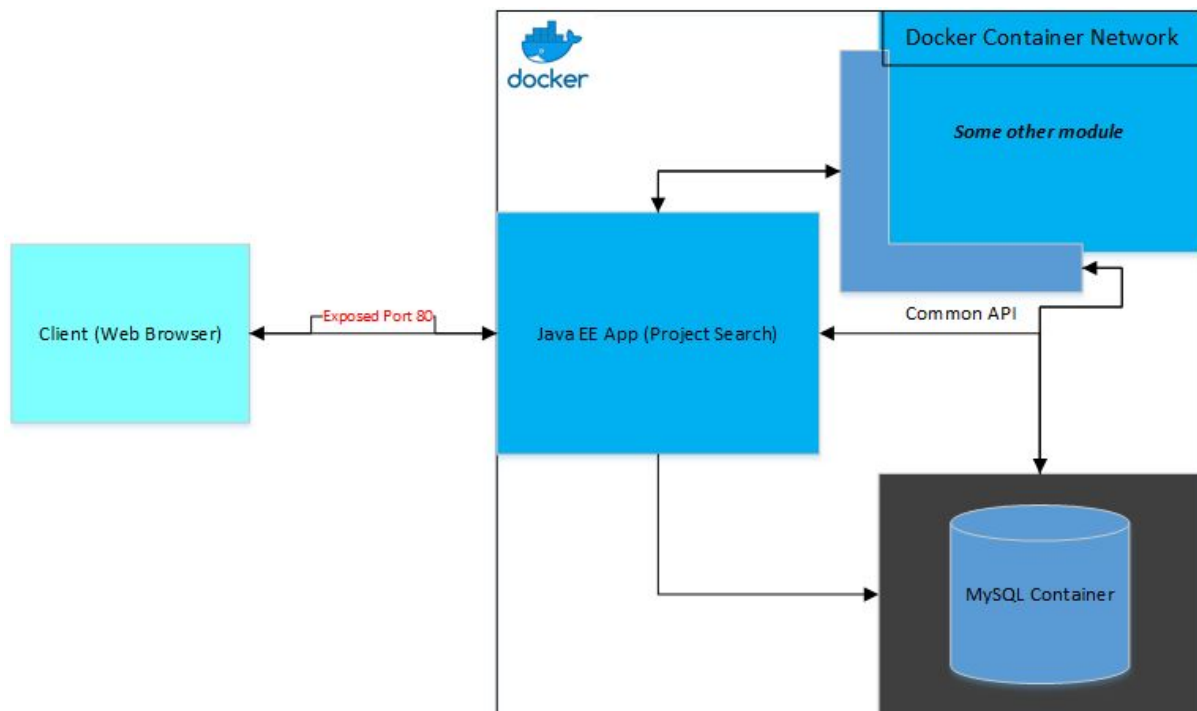


Figure 1 - Application Architecture Overview

## Interoperable API Design (Common API)

The modules are to have a push design, where jobs and tasks are pushed onto the module, for example, a user has now navigated to the "project search" module, a push trigger is set and details passed.

Function Name	Args	Notes
TriggerHandOver	PUMSSession Object	
RegisterPUMSEvent	Function - void	Register the module for notifications, for example a user signout.

Table 5 - Initial thoughts table

## User Journey Diagram

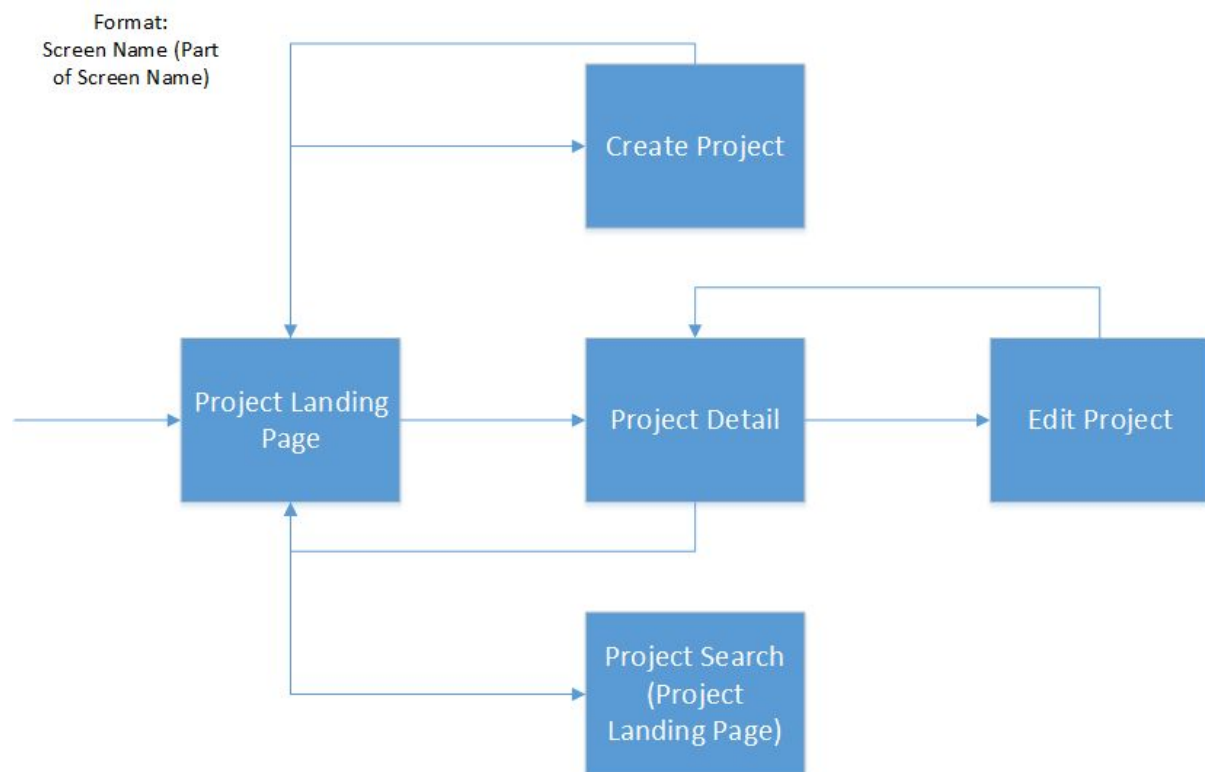


Figure 2 - User Flow Diagram

# Initial Designs

No.	Name	Wireframe
1	Landing Page	<div><div>Project Search</div><div><div>&gt;&gt;</div><div>Enter Text</div><div><div>Filter Criteria Group A</div><div>Filter Criteria Group B</div><div>Filter Criteria Group C</div></div></div><div><div>Projects</div><div>Project A</div><div>Project B</div><div>Project C</div><div>Project D</div><div>Project E</div></div></div>



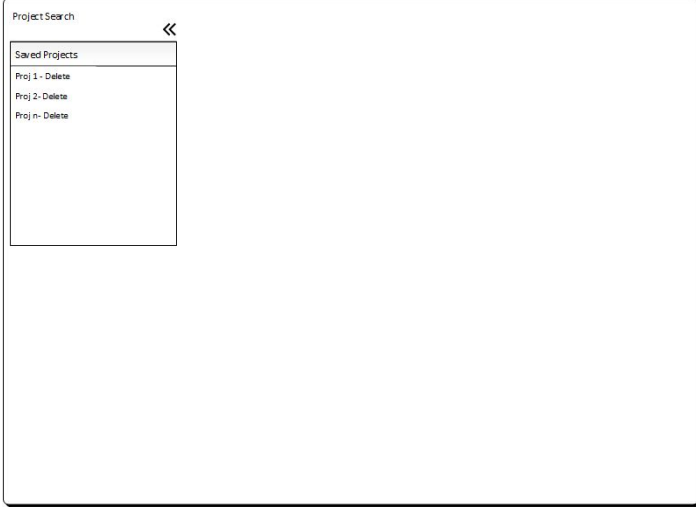
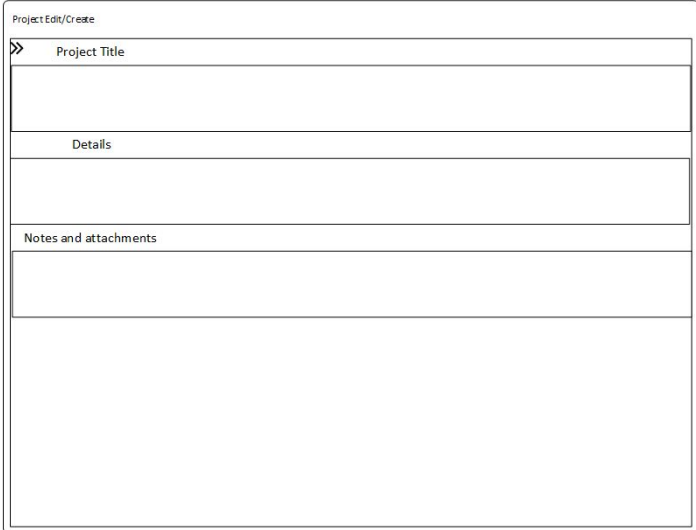
3	Projects save list - sidebar - appears on user is able to show on all pages	
4	Project Edit/Create page	

Table 5 - Wireframes

## User Search Criteria Initial Thoughts

Criterion	Details	Notes
Keyword	Keyword search that can be used together with criterion below	Optional
Project Creator	User which created this project	
Supervisor	Supervisor interested in this project	

## User Accessibility

An approach was taken for designs in this project and will be followed for this project, meaning an emphasis on:

- Minimalist and clear interface
- Clear wording
- Time permitting - High contrast themes
- Clear colouring - without high contrast themes

Where possible.

## Timescale

This project is to be completed by 17 May 2019, using bi-weekly sprints to implement features.

Sprint one is due to start 19 February, with planning done on 18 of February, tracking and planning is to be done via JIRA.