|  |  |  |  |
| --- | --- | --- | --- |
| **ILP 2.0** |  | | |
| **Date:** | 13 March 2011 | **Release:** | Draft |
| **Author:** | James Ballard | | |
| **Owner:** | Mick Kahn | | |

# Purpose

This document outlines the specification for the consolidation and rewrite of the ULCC Individual Learning Plan (ULCC). This is intended to rationalise the code and database to improve flexibility and sustainability as well as upgrade the system to Moodle 2.0.

Developments are described as essential where they include existing functionality with ‘Essential 1’ indicating this is required to release a new version and ‘Essential 2’ indicating essential items when we upgrade existing sites. ‘Desirable’ indicates new functions that should be included where possible.

# Expected Benefits

* Reduce support calls for small interface tweaks to a negligible level;
* Reduce lead time on implementing new services;
* Create new business opportunities for additional plug-in types

# Background - Personalised Learning Framework

The Individual Learning Plan (ILP) is a core component of the ULCC Personalised Learning Framework. The ILP is introduced as a formative assessment tool providing scaffolding to heighten learner awareness and essential recording of discussions for continuity and structure through subsequent activities. The dialogic structure is judgemental creating a shared vision for future development which is not predetermined and can only be judged in relation to that vision. Customisation is adaptive where learners use standard tools of development planning (target setting, progress review, etc.) to create a unique learning path.

# Current Design (Limitations)

The current design is implemented through four plug-ins. This was intended to allow each element to be used independently and required the database and functions to be self-contained. A ‘template’ file, can be configured through the code, to determine how the page is displayed and pull together data from each of the subsequent modules.

# New Design (Principles)

* Create a single plug-in that manages all elements consistently;
* Recognise that the current inputs for targets, reviews, and information are essentially all the same structurally and need not have their own tables;
* Provide the flexibility to have a potentially infinite number of reports, rather than an arbitrary limit. Additional reports will be easier to customise locally;
* Simplify the number of capabilities to view, add, edit where each plug-in defines which contexts allow this using the existing student, teacher, tutor, other hierarchy;
* Allow the same report to be presented in different ways for different purposes, which is essential for introducing more visual indicators.
* Allow categories, tagging, and templates to be implemented consistently across all reports.
* Allow other types of report (e.g. diagnostic, gradebook, etc.) to be implemented within the structure.
* Allow users the opportunity to personalise their view of the ILP more appropriate to their needs.

# 1. Feature Overview

The core functionality of the ILP can be abstracted as collecting and processing responses that are prompted by a standard set of tools. These tools may vary be sector and subject area, however the underlying process should remain unchanged.

In this new model the existing reports such as attendance, student profile, targets, subject reports, and progress reviews become plug-ins of which there will be several defaults and a the possibility to develop extra ones.

There are then three main output formats for the ILP: screen, print and data (csv/xml) and two variants of each: one for the individual user and one for a group of users. Each plug-in should define how it presents itself to each format, how it utilises capabilities and the frequency it can be taken.

## 1.1. Entries and Responses (Core/Plug-in)

The main user action within the ILP is creating/editing an entry. An 'entry' records the type of report used and relevant metadata (user, time, etc.). Responses are attached to an entry and refer to individual response items. The implication here is that responses can be plural rather than singular for any entry where a response might be text, numerical, date etc.

* The ‘category’ classifies the area the entry relates to from a pre-set list of categories;
* The ‘state’ allows an entry to progress (e.g. a target may have state ‘achieved’ or ‘withdrawn’);
* The ‘status’ records any status change resulting or recommended from this entry.

## 1.2. Reports and Prompts (Core/Plug-in)

A user makes entries against questions or prompts, which are contained within a report. A report should be user configurable, which allow prompts to be selected from a range of plug-in types. Each prompt plug-in defines the validation for its completion (e.g. must be positive integer).

Default prompt types include:

* Text entry;
* Date selector;
* Deadline (date for calendar);
* Value scale;

The report configuration will define how this is presented along with rules about how often a new entry can be made.

Frequencies include two entries: the first is a numerical value and the second a duration (e.g. 1 week, 2 days).

In principle the same prompt may be used in multiple forms. While default will be simple types there is the possibility that some projects may require complex bespoke validation rules.

## 1.3. Comments (Core)

Comments may be made against any entry as the dialogue evolves over time.

## 1.4. Individual Display (Core/Plug-in)

Each individual should be able to view their ILP as well as customise the view as necessary. Entries should be grouped by academic year with easier navigation. The core system will provide the display framework, with each plug-in providing the format. Again some display rules may require complex processing to interpret responses. A configurable print format and LEAP2A XML export should be provided.

## 1.5. Group Display (Core/Plug-in)

Members of staff should be able to view any users they have responsibility for within the ILP contexts, this should include search functionality and provide summary information that allows quick identification of support needs. A configurable print format and CSV export should be provided.

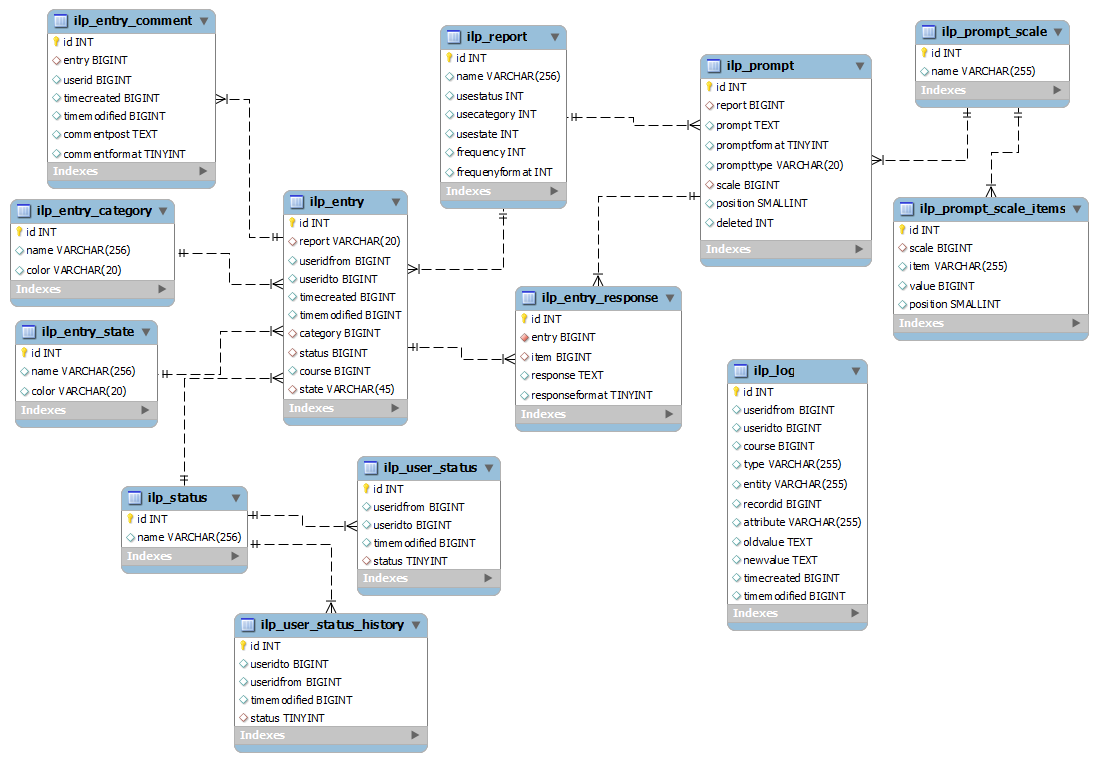
## 1.6. Block Display (Core)

The ILP block itself should provide a simple dashboard into the relevant views for the role and context.

# Specification

## 2. Database

The database entity diagram describes how the new structure might work based on early prototyping.

The structure for this has been derived from the existing ilp, message and feedback modules for consistent convention.

### Function Hierarchy

|  |  |  |
| --- | --- | --- |
| Function | | Priority |
| 2.1 | Database install (/db/install.xml) | Essential 1 |
| 2.2 | Database upgrade (/db/upgrade.php) | Essential 2 |

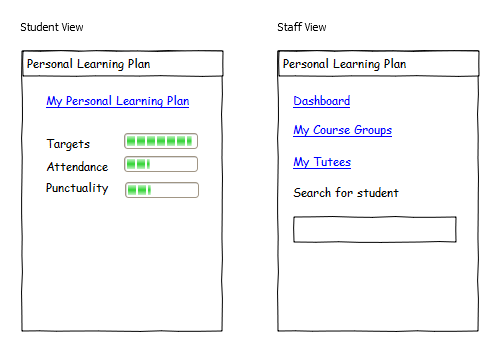
## 3. Interfaces

### 3.1 Block

### Function Hierarchy

|  |  |  |
| --- | --- | --- |
| Function | | Priority |
| 3.1.1 | Block library (block\_ilp.php) | Essential 1 |
| 3.1.2 | Summary library (ILP) | Essential 1 |
| 3.1.3 | Summary library (MIS) | Essential 1 |
| 3.1.4 | Role definitions | Essential 1 |
| 3.1.5 | User search library | Desirable |

|  |  |
| --- | --- |
| User Intention | System Responsibilities |
|  | 1. Identify user |
|  | 1. Display summary and links |
| 1. Click Link | 1. Process link |
|  | 1. Return Page |



|  |  |
| --- | --- |
| User Intention | System Responsibilities |
|  | 1. Identify user template |
|  | 1. Display page |
| 1. Click Link | 1. Process link |
|  | 1. Return Page |

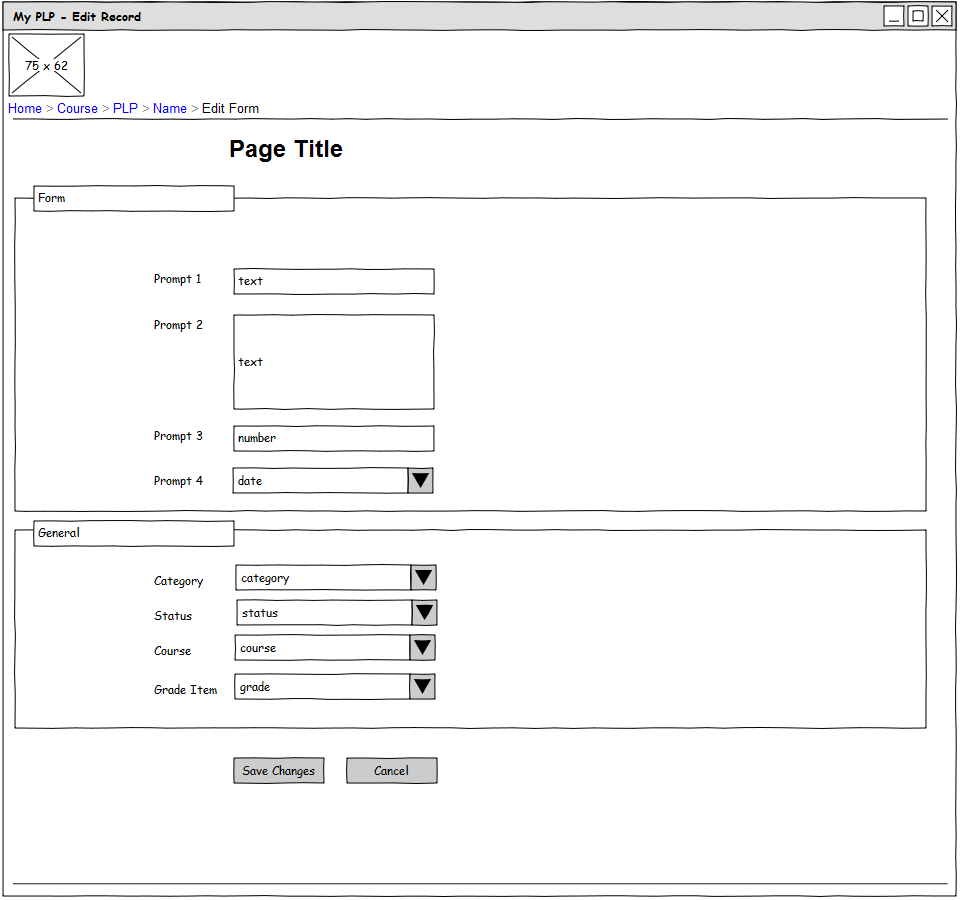
### view.png3.2 Individual View

### Function Hierarchy

|  |  |  |
| --- | --- | --- |
| Function | | Priority |
| 3.2.1 | View library (overview) | Essential 1 |
| 3.2.2 | View library (detailed) | Essential 1 |
| 3.2.3 | Display library (MIS plug-in) | Essential 1 |
| 3.2.4 | Display library (default plug-ins) | Essential 1 |
| 3.2.5 | View template (screen) | Essential 1 |
| 3.2.6 | View template (print) | Essential 1 |
| 3.2.7 | Display library (bespoke plug-in) | Essential 2 |
| 3.2.8 | Academic year filter | Desirable |
| 3.2.9 | View library (custom) | Desirable |
| 3.2.10 | View library (narrative) | Desirable |
| 3.2.11 | View template (RSS) | Desirable |
| 3.2.12 | View template (PDF) | Desirable |
| 3.2.13 | View configuration GUI | Desirable |

### 3.3 Add/Edit Entry

|  |  |
| --- | --- |
| User Interaction | System Responsibilities |
|  | 1. Create and display form |
| 1. Complete form |  |
| 1. Submit form | 1. Validate data |
|  | 1. Process and save data |
|  | 1. Return confirmation |

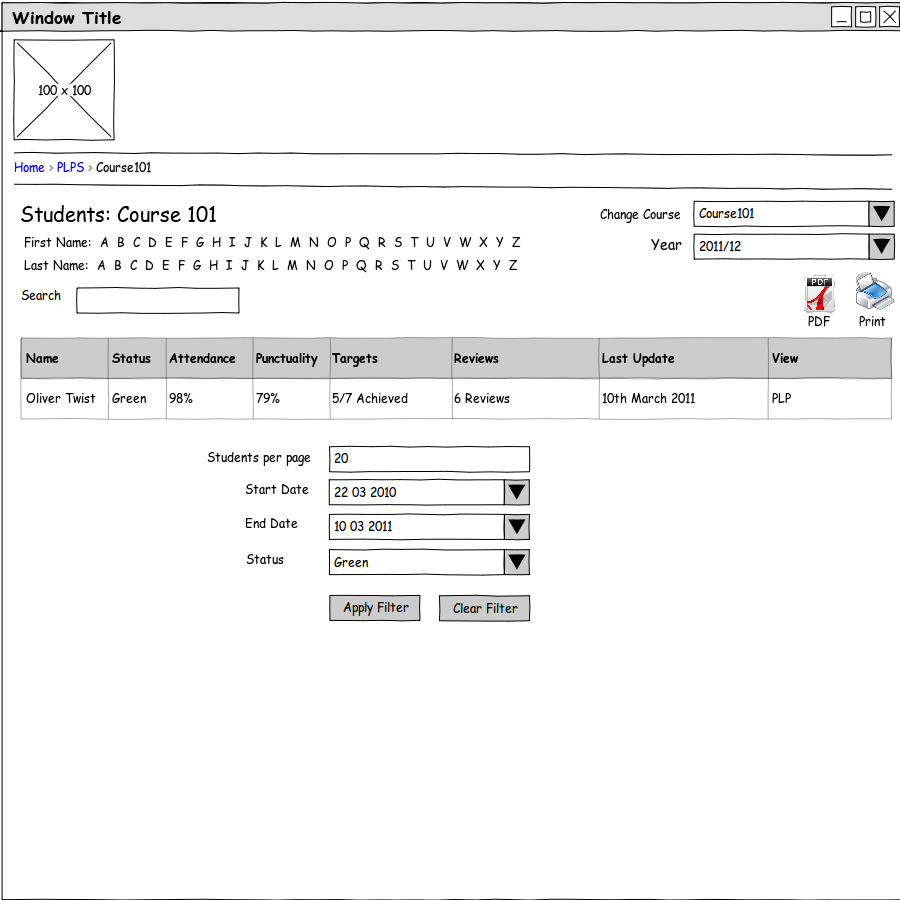


### Function Hierarchy

|  |  |  |
| --- | --- | --- |
| Function | | Priority |
| 3.3.1 | Entry library | Essential 1 |
| 3.3.2 | Form library (Default plug-ins) | Essential 1 |
| 3.3.3 | Form library (General) | Essential 1 |
| 3.3.4 | Form library (Bespoke plug-ins) | Essential 2 |

### 3.4 Student list

|  |  |
| --- | --- |
| User Interaction | System Responsibilities |
| 1. Select filters | 1. Create user list |
|  | 1. Process and display list |
| 1. Click link | 1. Process link |
|  | 1. Return page |

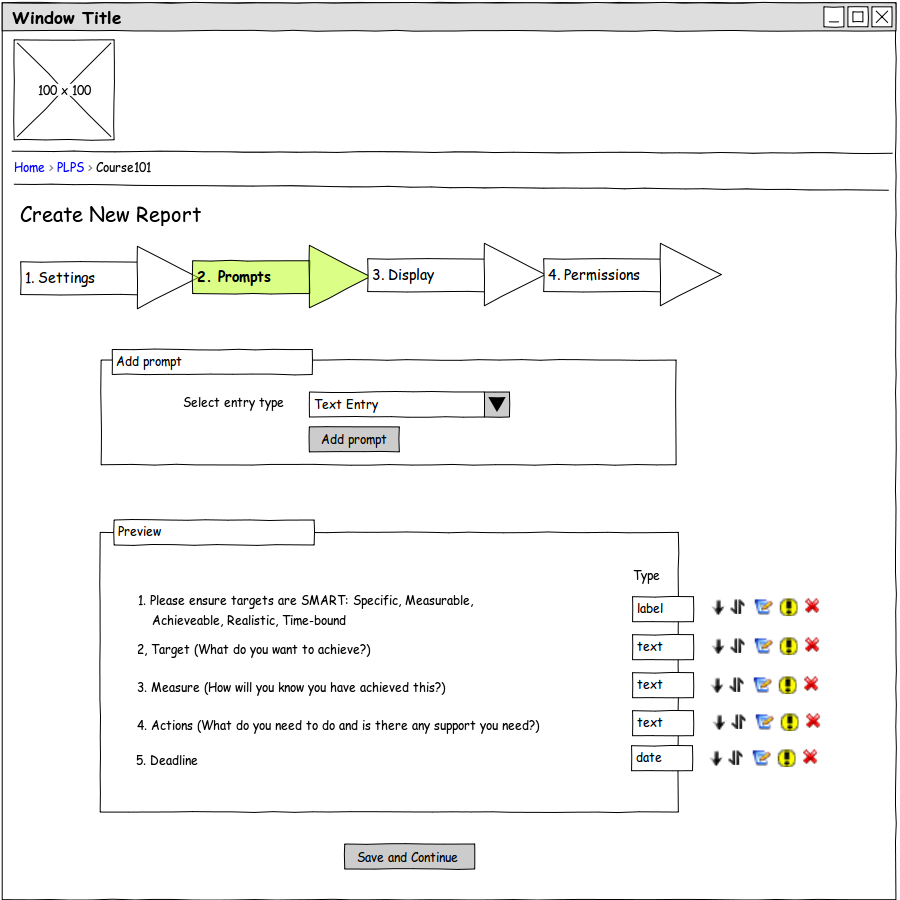


### Function Hierarchy

|  |  |  |
| --- | --- | --- |
| Function | | Priority |
| 3.4.1 | Display library (group) | Essential 1 |
| 3.4.2 | Group view template (screen) | Essential 1 |
| 3.4.3 | Group view template (print) | Essential 1 |
| 3.4.4 | User search library | Desirable |
| 3.4.5 | Academic year filter | Desirable |
| 3.4.6 | Group view template (print) | Desirable |

### 3.5 Add/Edit Report

|  |  |
| --- | --- |
| User Interaction | System Responsibilities |
|  | 1. Create and display form |
| 1. Complete form |  |
| 1. Submit form | 1. Validate data |
|  | 1. Process and save data |
|  | 1. Return confirmation |



### Function Hierarchy

|  |  |  |
| --- | --- | --- |
| Function | | Priority |
| 3.5.1 | Report library | Essential 1 |
| 3.5.2 | Prompt library | Essential 1 |
| 3.5.3 | Prompt types (default plug-ins) | Essential 1 |
| 3.5.4 | Role definitions | Essential 1 |
| 3.5.5 | Prompt types (Bespoke plug-ins) | Essential 2 |