

**Stage: Systems Analysis and Design**

**System Design Specification**

**[SMI012 Digital Image Submission for Staff and Visitors]**

**[PROGRAMME NAME]**

**[SMI012]**

**[ANNUAL PLAN NUMBER]**

**Document Version: [1.1]**

**Date: [08/01/2016]**

**Contents**

[1 Document Management 5](#_Toc440012202)

[1.1 Contributors 5](#_Toc440012203)

[1.2 Version Control 5](#_Toc440012204)

[2 OVERVIEW 6](#_Toc440012205)

[2.1 Development Tools 7](#_Toc440012206)

[2.2 Development Standards 7](#_Toc440012207)

[3 SYSTEM PROCESSES 8](#_Toc440012208)

[3.1 Staff Data feed 8](#_Toc440012209)

[3.1.1 Staff feed components 8](#_Toc440012210)

[3.1.2 Staff feed data flow 8](#_Toc440012211)

[3.2 Data Feed Requirements 9](#_Toc440012212)

[3.2.1 Expiry dates (mandatory) 9](#_Toc440012213)

[3.2.2 Load new staff records as form not printed and channel open (mandatory) 10](#_Toc440012214)

[3.3 Card System Requirements 10](#_Toc440012215)

[3.3.1 Bring in all staff records from HR to match those in IDM (mandatory) 10](#_Toc440012216)

[3.3.2 Add date of birth to card staff view (desirable) 11](#_Toc440012217)

[3.3.3 Not project work 11](#_Toc440012218)

[3.3.4 Add department name to VPEOPLE in CARD schema (mandatory) 11](#_Toc440012219)

[3.3.5 Student fix – restrict recipients of initial email (desirable) 11](#_Toc440012220)

[3.3.6 Periodically delete rejected images (desirable) 11](#_Toc440012221)

[3.4 MyEd channel 12](#_Toc440012222)

[3.4.1 Integrate staff into MyEd channel (mandatory) 14](#_Toc440012223)

[3.4.2 Add email field to form for staff in MyEd channel (mandatory) 15](#_Toc440012224)

[3.4.3 Email to be customised based on user type (mandatory) 16](#_Toc440012225)

[3.5 Image Handling 16](#_Toc440012226)

[3.5.1 Uploaded image file name to be in the format staffref.jpg (mandatory) 16](#_Toc440012227)

[3.5.2 Uploaded image file no larger than 500kb (mandatory) 16](#_Toc440012228)

[3.5.3 Uploaded image file to have jpeg extension only (mandatory) 16](#_Toc440012229)

[3.6 Card Applications 16](#_Toc440012230)

[3.6.1 CSU: New tab for pending images for staff (mandatory) 16](#_Toc440012231)

[3.6.2 CSU: Change status to form printed when accepting staff image (mandatory) 17](#_Toc440012232)

[3.6.3 FrontOffice: Add restriction on printed forms (mandatory) 17](#_Toc440012233)

[3.6.4 FrontOffice: Add a note to form (mandatory) 17](#_Toc440012234)

[Unit testing 18](#_Toc440012235)

[4 USER INTERFACE 19](#_Toc440012236)

[4.1 Transactional Interface 19](#_Toc440012237)

[4.1.1 MyEd Interface 21](#_Toc440012238)

[4.2 Reporting Interface 21](#_Toc440012239)

[5 APPLICATION SECURITY 22](#_Toc440012240)

[5.1 Authentication 22](#_Toc440012241)

[5.2 Authorisation 22](#_Toc440012242)

[5.3 Business Objects 22](#_Toc440012243)

[6 DATABASE DESIGN 23](#_Toc440012244)

[7 DATA 25](#_Toc440012245)

[7.1 Data Migration 25](#_Toc440012246)

[7.2 Archiving Policy 25](#_Toc440012247)

[8 IMPLEMENTATION 26](#_Toc440012248)

[9 Appendix 26](#_Toc440012249)

[9.1 Database Queries 26](#_Toc440012250)

# Document Management

When completing this document, please mark any section that is not required as ‘N/A’. A brief description of why the section is not required should also be included.

## Contributors

Please provide details of all contributors to this document.

|  |  |  |
| --- | --- | --- |
| **Role** | **Unit** | **Name** |
| Systems Analyst Designer (Owner) |  |  |
| Business Analyst |   |  |
| Project Manager  |   |  |
| Project Sponsor  |   |  |
| Business Area Manager  |  |  |
| *Other document contributors* |  |  |

## Version Control

Please document all changes made to this document since initial distribution.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Section** | **Amendment** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# OVERVIEW

Please provide a general overview of the system design so that anyone reading this section is able to gain an understanding of what functionality is provided and how this functionality will be technically implemented.

The existing photo submission channel in MyEd allows only students to submit a passport style photograph for the production of the university card. There is a need to allow all university staff, and at a later date, visitors to use this facility. The Card systems therefore need to be extended to integrate staff first, and later visitors. Changes will be made to existing systems to separate student, staff and visitor concerns within card applications (such as photo approval, email etc.) to allow more efficient card processing.

This document will describe the staff process only (stage 1 of the project).

DEVELOPMENT TOOLS AND STANDARDS

## Development Tools

Specify which technologies will be used to develop the application, eg Oracle 9i, Cold Fusion MX, Oracle Forms, Business Objects, Java, PL/SQL etc.

Oracle 11g

PL/SQL

Java Spring

ColdFusion

## Development Standards

Tick the appropriate box to indicate the standards being followed for this application:

|  |  |
| --- | --- |
| **Standard** | **√ indicates compliance** |
| Database Design |  |
| Cold Fusion |  |
| Java |  |
| Uportal Development |  |
| Accessibility |  |
| Web Style Standards |  |
| [Supported Web Browsers](http://www.ed.ac.uk/schools-departments/information-services/services/computing/application-development/application-support/supported-web-browsers/browser-application) |  |
|  |  |
|  |  |

# SYSTEM PROCESSES

Take each process specified in the Business Requirements Document (BRD) and state how it will be implemented technically (please make reference to the corresponding paragraph number in the BRD). Where applicable, a specification of each screen which makes up a process should be provided (ie a screenshot and descriptions of every data item displayed).

A resource estimate should be provided for each process.

All paragraphs should be numbered to assist cross referencing in testing stages.

## Staff Data feed

### Staff feed components

The staff feed is executed as a cron job. The first part refreshes the organisational data from HR. No changes are required here.



The second part loads staff records from HR to Card database. Here is where changes to staff records in line with requirements are to be applied.



### Staff feed data flow



Fig1. Population of organisation data.



Fig2. Population of staff data

**Note**: recreated CARD synonyms both in DEV and TEST with the following DDL:

CREATE OR REPLACE PUBLIC SYNONYM "SMRTCARD\_STAFF\_EXTRACT" FOR "UOEPP"."UOE\_CARD\_EXTRACT"@"CARDHR.IS.ED.AC.UK";

CREATE OR REPLACE PUBLIC SYNONYM "SMRTCARD\_ORG\_VIEW" FOR "UOEPP"."UOE\_CARD\_ORG\_VIEW"@"CARDHR.IS.ED.AC.UK";

## Data Feed Requirements

### Expiry dates (mandatory)

A check for form status needs to be added to ‘*update\_status’* procedurein *staff\_pop\_dets.sql* script.

For existing staff records with **MONDEX\_STATUS = ‘P’** (form printed) **OR MONDEX\_STATUS = ‘E’** (form not printed), update the card expiry date (SMRTCARD\_EMAIL.**CARD\_EXPIRY\_DATE**, the one printed on card) to the latest expiry date (SMRTCARD\_EMAIL.**NESI\_CARD\_EXPIRY\_DATE**).

old SMRTCARD\_EMAIL.**CARD\_EXPIRY\_DATE** = new SMRTCARD\_EMAIL.**NESI\_CARD\_EXPIRY\_DATE**

Then for all existing staff records irrespective of form status, update the latest expiry date field.

old SMRTCARD\_EMAIL.**NESI CARD\_EXPIRY\_DATE** = new SMRTCARD\_EMAIL.**NESI CARD\_EXPIRY\_DATE**

### Load new staff records as form not printed and channel open (mandatory)

Set form not printed and channel open flags in ‘*insert\_new\_staff’* procedurein *staff\_pop\_dets.sql* script.

The current feed already inserts new staff records as form not printed; value ‘**E**’ for **mondex\_status**.

Add column **CHANNEL\_BLOCK** to insert statement, and set value to ‘**N**’.

## Card System Requirements

The current Card setup does not provide a way to determine whether submitted image belongs to a student, member of staff or visitor. Linking to SMRTCARD\_DETAIL table will give partial results since awaiting record individuals would be missed.

To record user type in Card database, add **USER\_TYPE** column to **SMRTCARD\_DETAIL\_UPLOADS** table. The allowed values will be an enumeration of student, staff and visitor (might use CHECK constraint).

The type value will be persisted into SMRTCARD\_DETAIL\_UPLOADS.USER\_TYPE column from PhotoPortlet while saving the image and will be required by CSU application to separate student, staff and visitor pending images in corresponding tabs.

The new column should be updated with type value for all existing records during deployment to TEST and LIVE.

### Bring in all staff records from HR to match those in IDM (mandatory)

Problem: Staff records appear in IDM but do not appear in CARD at all.

Modify card staff view **UOE\_CARD\_EXTRACT** in HR (user **UOEPP**).

Remove the following constraint from both SELECT statements in the view:

AND (PG.**NAME** IS NULL OR PG.**NAME** != 'NONE')

This will unblock the records where grade is ‘NONE’.

### Add date of birth to card staff view (desirable)

Add **DATE\_OF\_BIRTH** from **PER\_PEOPLE\_F** to select statement in UOE\_CARD\_EXTRACT view in HR (user UOEPP).

In ‘*insert\_new\_staff’* procedurein *staff\_pop\_dets.sql* script, replace default UNIX epoch date, with the date passed from HR.

Replace TO\_DATE ('01-01-1900', 'DD-MM-YYYY) with UOE\_CARD\_EXTRACT.DATE\_OF\_BIRTH.

### Not project work

### Add department name to VPEOPLE in CARD schema (mandatory)

Add SMRTCARD\_ORGANISATION.**DEPARTMENT** column to VPEOPLE view select statement. Join on

SMRTCARD\_DETAIL.**DEPT** = SMRTCARD\_ORGANISATION.**DEPT\_ID**

### Student fix – restrict recipients of initial email (desirable)

In *smrtcard\_email.sql* script, add the additional constraints to student selection query

and **CARDHOLDER\_STATUS = ‘A’** (active students)

and **CHANNEL\_BLOCK = ‘N’** (no card issued)

### Periodically delete rejected images (desirable)

In awaitingRecord application (<https://svn.ecdf.ed.ac.uk/repo/is/apps/software/card/awaitingRecord/trunk/awaitingRecord>), add the delete rejected images query handler method to *AwaitingRecordDAO.java* file.

DELETE FROM **SMRTCARD\_REJECTED\_IMAGES** WHERE HOST\_ID IN (

SELECT IMAGE.HOST\_ID FROM SMRTCARD\_REJECTED\_IMAGES IMAGE INNER JOIN

SMRTCARD\_DETAIL DETAIL ON IMAGE.HOST\_ID = DETAIL.HOST\_ID

WHERE

DETAIL.CARDHOLDER\_STATUS = 'I' -- (inactive)

OR

DETAIL.CHANNEL\_BLOCK = 'Y'-- (card issued)

);

Call the new delete method in *TransferData.java* process() method.

Log the outcome.

## MyEd channel



Fig3. PhotoPortlet UML design

The image will no longer be temporarily saved to database as pending (IMAGE\_STATUS=’P’) when the form is uploaded. This results in unnecessary records when the user walks away from the portlet without accepting an image.

Instead, the image file will be saved in local **temp** directory and its URL will be served back to the client. The image file will be deleted when the user accepts or rejects the image. Any image files left in temp directory will be deleted after some time by a cron job.

Fig.4 illustrates the database interaction.



Fig4. Data flow

To retrieve channel\_block flag from Card database, query smrtcard\_detail first for users that exist in the Card system, then query smrtcard\_image\_uploads for those who don’t. If a user doesn’t have a record in either, then channel is open (channel\_block=‘N’).

Detailed description in [Appendix 9.1](#_Database_Queries).

### Integrate staff into MyEd channel (mandatory)

This requires changes in PhotoPortlet. N.B. The portlet will be upgraded to Spring 3.2.15.RELEASE and will be annotation based.

Configure portlet to parse database queries, email wording and other portlet messages from properties file(s) based on user type: student, staff and visitor.

### Add email field to form for staff in MyEd channel (mandatory)

#### Process description

The email capture is a separate step in the process, enabled for staff only.

What happens after user accepts the image in *confirm* form (Fig 7) depends on the user type.

If student, the process is finalised. The image and program data are persisted to the Card database. If the process completed without error, the channel gets blocked and the *success* view is shown (Fig 9).

If staff, there is an extra step. The user is shown an email form (Fig 8). The form will have two input fields to enter the email address into to ensure a match. There will be a client side and the server side validation to verify the match and to check email is in correct format. On successful validation, the process is finalised and the *success* view is shown (Fig 9).

Email address to be stored in session.

Email address to be persisted in Card database in SMRTCARD\_IMAGE\_UPLOADS.EMAIL column. It will be used by CSU application for image accepted/rejected communications. This email will only be available throughout the image approval/rejection process. It gets deleted once the process is complete.

Note: SMRTCARD\_IMAGE\_UPLOADS.EMAIL will hold student email address (uun+ @sms.ed.ac.uk) and email address submitted by staff via the upload form.

#### Required components

Create *EmailBean* form backing bean.

Create *EmailValidator* class that will check that email address has been submitted (not null and not an empty string), is in valid format and both email addresses match.

Create *email* form view.

Create controller to handle the email validation logic.

### Email to be customised based on user type (mandatory)

PhotoPortlet: in *PersonService* subclass (StaffService if user type is staff, StudentService if user type is student) parse email wording from properties file based on user type.

CSU: In *ApproveImagesController*, modify getApprovedEmailText and getRejectedEmailText methods to accept user type. Parse the email text from properties file based on user type.

## Image Handling

Create *FileUploadValidator* class that will contain validation logic from section 3.5.2 and 3.5.3. Extra validation to be added to check if file has been uploaded (not null, not empty).

Perform uploaded file validation in *Controller*. If validation fails, send error messages back to the upload form.

### Uploaded image file name to be in the format staffref.jpg (mandatory)

In *Controller*, change uploaded file name to personnel number attribute from portlet user attributes (as defined in portlet.xml). This attribute should already be stored in user session.

### Uploaded image file no larger than 500kb (mandatory)

In *FileUploadValidator* class, check uploaded file size. If the file is larger than 500kb, add an error with appropriate message to be sent back to the form.

### Uploaded image file to have jpeg extension only (mandatory)

In *FileUploadValidator* class, check the uploaded image extension. If the extension is not image/jpeg, add an error with appropriate message to be sent back to the form.

## Card Applications

### CSU: New tab for pending images for staff (mandatory)

The original ‘Approve images’ tab must be replaced with two new tabs that will separate the student and staff pending images.

1. In WEB-INF/jsp/menu\_images.jspf add ‘Approve student images’ and ‘Approve staff images’ links in place of generic ‘Approve Images’ link.

The new links should be parameterised with an identifier pointing to whether it is to fetch student or staff data. This can be done with a request parameter i.e. approveimgs.htm?type=student

2. Modify method listSubmittedNotApproved() in JdbcCSUDao.java to accept the type parameter.

3. Retrieve the type parameter from HttpServletRequest and pass it to method call listSubmittedNotApproved() in ApproveImagesController.java formBackingObject() method.

4. In images.jsp add two new instructions paragraphs in place of generic ‘Approve Images’ paragraph.

### CSU: Change status to form printed when accepting staff image (mandatory)

1. Modify the smrtcard\_detail update query in saveApprovedUploads() in *JdbcCSUDao* to change MONDEX\_STATUS = ‘P’ when USER\_TYPE is staff (case when).

### FrontOffice: Add restriction on printed forms (mandatory)

Add the following condition

AND UOE\_START\_DATE <= TRUNC(SYSDATE)+1

to *getPrintedFormsNeeded* and *getPrintedFormsLargeBatch* and *setFormPrintedBatch* and *setFormPrintedBatchSubset* in system/model/DAO/commonDAO.cfc.

Both queries already are restricted to records with MONDEX\_STATUS = ‘E’ (form not printed).

### FrontOffice: Add a note to form (mandatory)

Add html formatted note to *system/views/printedforms.cfm*

### Unit testing

#

# USER INTERFACE

## Transactional Interface

Use this section to link to an application prototype built by a Designer. The screens should be built in conjunction with the business partner.

If this section is not applicable then please state why.



Fig5. Channel blocked screen



Fig6. First screen – upload form



Fig7. Second screen – confirm form



Fig8. Final screen – success view



Fig9. Final screen – success view



Fig10. Error view

### MyEd Interface

*Where a MyEd channel is being used to deliver some or all of the transactional interface, complete the following template (one per channel) and link each completed template to this document. The information in this template will form the basis of the channel publishing information used in MyEd when building channels.*

* *Channel Publishing Template*

## Reporting Interface

Where applicable, please state how reporting is to be delivered, eg Webi, HTML.

The design of any pre-written reports should be specified, and if Webi is to be used, the Universe should be defined (ie classes/measures/dimensions and source data).

# APPLICATION SECURITY

This section relates to application rather than physical security which is covered in the Technical Architecture Document (TAD).

## Authentication

## Authorisation

Include application, menu and page authorisation and any application roles.

## Business Objects

Include any specific object/row authorisation.

# DATABASE DESIGN

Insert an Entity-Relationship Diagram of the system. If new tables have been added to an existing system then highlight these in your ER Diagram (full diagram of existing system not needed).

 Full specifications should be provided for all new tables/views/columns in the system (ie name, datatype, mandatory, default values, constraints).

Table **SMRTCARD\_IMAGE\_UPLOADS** will be expended by **USER\_TYPE** column of type **NVARCHAR2(**10 byte) with check constraint (user\_type in (‘student’,’staff’,’visitor’).

|  |  |  |
| --- | --- | --- |
| Name | Null | Type |
| HOST\_ID | NOT NULL | VARCHAR(8) |
| IMAGE\_STATUS |  | VARCHAR2(1) |
| IMAGE |  | BLOB |
| IMAGE\_STATUS\_DATE |  | DATE |
| AWAITING\_RECORD |  | VARCHAR2(1) |
| CHANNEL\_BLOCK |  | VARCHAR2(1) |
| EMAIL |  | VARCHAR2(255) |
| **USER\_TYPE** | NOT NULL | VARCHAR2(10) |

HOST\_ID is uun without trailing ‘s’ for students; personnel number for staff.APPLICATION INTERFACES

Provide technical details of any inputs to the system from other applications and any outputs from the system to other applications. Full data specifications should be included. Identify possible impact on existing processes.

# DATA

## Data Migration

Include any mappings between existing systems and the new system

## Archiving Policy

# IMPLEMENTATION

Specify any implementation issues that should be considered when creating an implementation plan at the Build stage.

# Appendix

## Database Queries

Query1. Get channel block

* + - smrtcard\_detail

(also retrieve batch\_type, home\_email, preferred\_forename)

* + - smrtcard\_image\_uploads

Query 2. Upload image

If user does not have a smrtcard\_detail record, set awaiting record flag to ‘Y’

If user in smrtcard\_image\_uploads, delete the record

* + - Insert image into smrtcard\_image\_uploads
			* IMAGE\_STATUS = ‘S’
			* IMAGE\_STATUS\_DATE = SYSDATE
			* AWAITING\_RECORD (‘Y’ if user not in smrtcard\_detail and inDetails is false, ‘N’ otherwise)
			* CHANNEL\_BLOCK = ‘Y’
			* USER\_TYPE (student, staff, visitor)
		- Update smrtcard\_detail if record exists
			* IMAGE\_APPROVED='N'
			* CHANNEL\_BLOCK='Y'