



Project Fact Sheet

Kuali Student Accounts Software

Washington DC, 21 August 2014

The Kuali Student Accounts software is more than an accounts receivable system. It was designed to automate compliance with the rules of the U.S. Department of Education and interpretations of the internal Revenue Service enforcing the use of federal financial aid only for “educational purposes”—called payment application, accommodate billing fee-based higher education products and services by automating complex fees pricing, and to improve the productive and accuracy of staff and better communications with students, faculty, and staff through new technology for screens and new screen designs.

The fee management module used for pricing and billing can also be used in student registration systems.

The technologies

KSA was designed using service –oriented methodology balancing flexible data-driven processes with rule-driven flexibility for highly-complex tasks.

KSA can plug into Kuali, or other identity management solutions to provide role-based authorization via KSA real-time services.

Written in Java using the Spring Framework, JavaScript, XML, CSS, and HTML consistent with practices in the finance industry, Screens were developed using Kuali Rapid Application Development (KRAD). Some, for performance, will use Google’s contributed open-source AngularJS. These will be done using Bootply for editing Twitter’s Bootstrap, and Adobe’s Dreamweaver.

The FSA system has over 100 screens—more than 80 are complex screens as shown. Screen designs are patterned on research by Ben Sheiderman and Jakob Nielsen that describe design principles so staff could be most productive and most comfortable. The user interface design began with “personae”—fictional typical users with different roles— to identify the different needs and expectations of those users.

By a crude but often used measure of system size KSA has 3.1 million lines (611 megabytes) of code. The project began in November 2011 and was completed in January 2014 - 27 months from start to finish. Subsequently some additional work has been identified pending further development of other Kuali systems.

Project Financing

The project was sponsored by the University of Maryland College Park and the University of Southern California, with contributions by Sigma Systems Inc. Boston College is funding the implementation that includes some specific additions enhancing the functionality of the system.

Contributions to the Project	Project Resources	
	Costs	Person-Hours
Project Sponsors	\$775,000	
Equivalent person-hours		7,000
Sigma Systems Inc.	\$1,633,500	16,500
Project Total	\$2,408,500	23,500

For additional information:

Contact

Cliff Clevenger

Sigma Systems Inc.

clevenger@sigmasys.com

+1-303-758-4610 x541

The screenshot displays the Kuali Student Accounts interface. The main heading is "Transactions Running Balance" for the student "Griffin, Peter". A "Reverse Transaction" dialog box is open, showing details for a transaction with an effective date of 04/03/2012 and a description of "Tuition". The dialog includes a "Reverse Transaction" button and a "Cancel" button. In the background, a table shows a running balance with columns for Effective Date, Description, Charges, Payments, Allocated, Unallocated, and Balance. The table lists several transactions, including a Bank Initial Charge, Tuition, Health Center, Parking Fee, and Cash Payments.

Effective	Description	Charges	Payments	Allocated	Unallocated	Balance
24032012	Bank Initial Charge (1)	\$50.00		\$0.00	\$0.00	\$50.00
24032012	Tuition (2)	\$10,000.00		\$1,874.88	\$8,125.12	\$10,000.00
18022012	Health Center (1)	\$17.88		\$17.88	\$0.00	\$10,087.88
11072012	Parking Fee (1)	\$88.12		\$88.12	\$0.00	\$10,133.12
10072012	Technology Fee (1)	\$102.80		\$102.00	\$0.00	\$10,235.12
24032012	Cash Card (or Person Payment) (1)		\$2,500.00	\$2,500.00	\$0.00	\$7,735.12
24032012	ACA Payment (1)		\$1,200.00	\$1,200.00	\$0.00	\$6,535.12
24032012	Cash Payment (1)		\$50.00	\$50.00	\$0.00	\$6,485.12

Transaction Screen for a Student with a Light Box for Transaction Reversal