SUNAPSIS LEGOS:

BUILDING CUSTOM FEATURES – MARCH 4TH, 2025

UNIVERSITY OF GEORGIA
OFFICE OF GLOBAL ENGAGEMENT

- Micah Cooper
- IT Manager
- mrcooper@uga.edu

- Marie Goodwin
- Immigration Data Analyst
- mariet@uga.edu

AGENDA

- Introduction
- Problem solving at UGA
- Planning phase
- Building phase
- Testing and Go-Live
- Demo
- Final tips, takeaways & questions

THE PROBLEM AND CONTEXT AT UGA

The UGA Immigration Services Office charges service fees.

Sponsoring department

International visitor

Previously, we used a basic custom fee table in sunapsis.

Manual process

Error-prone

Several stakeholders, each with their own goals and preferences.

Immigration

Finance

IEP

UGA FMS (Accounting + TouchNet)

Goal: Improve accuracy and reduce the overall workload.

Scope: Immigration Services fees for employees and scholars

Not in scope: Fees paid to other parties like USCIS, Student related fees

PLANNING PHASE

PLANNING

Project Requirements
Automate as much as possible to reduce human errors
Store all fee and billing information in the record
Dynamically add a fee to the sunapsis record
Easily maintain the correct fee amount
Use automated emails when fees are due
Ability to document and reconcile fees

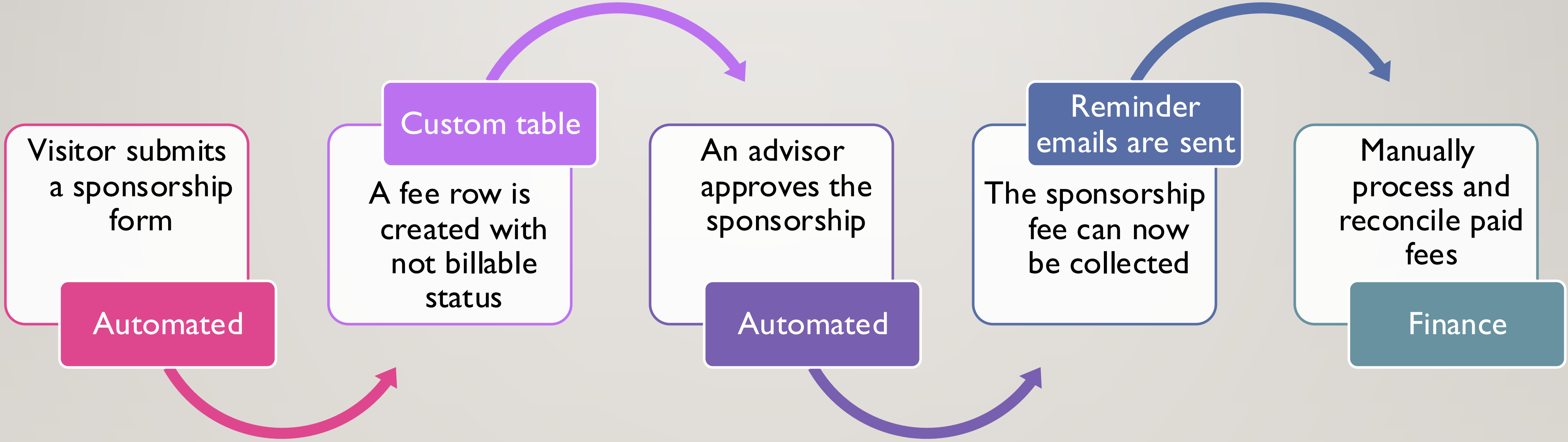
PLANNING – FEES AND INVOICE STATUS

	Sponsorship Fees	Annual Fees
Number of fees	13	2
Examples	DS-2019 application H-1B application PR application	Scholar compliance fee
Frequency	One-time	Recurring
When to collect	Dynamic invoice status	Due at time the fee is added

PLANNING

Finance Considerations
Centralize all fee information on the new custom fee table
Finance uses three different systems to process fees and requires human data entry
Audit checks will be needed
Automation will be limited
Finance will need to monitor the fee to ensure it is fully processed

BASIC SPONSORSHIP FEE WORKFLOW



BUILDING PHASE

BUILDING PIECES

I) How to add a fee (Automate)

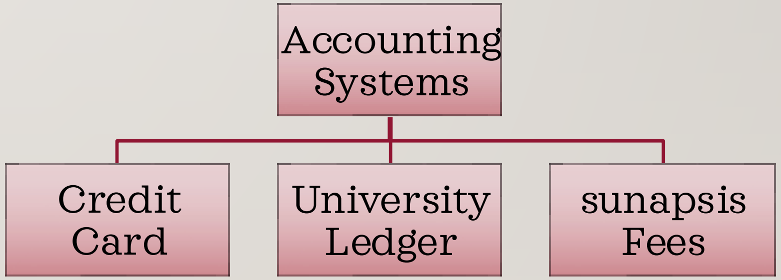
- Custom table to hold applicable fees
- E-form extensions and SQL script add fees
- A mapping of fees and amounts

II) When to collect (Automate)

- A SQL script updates the invoice status to let Finance know when it can be collected
- Annual Fees are due once they are added

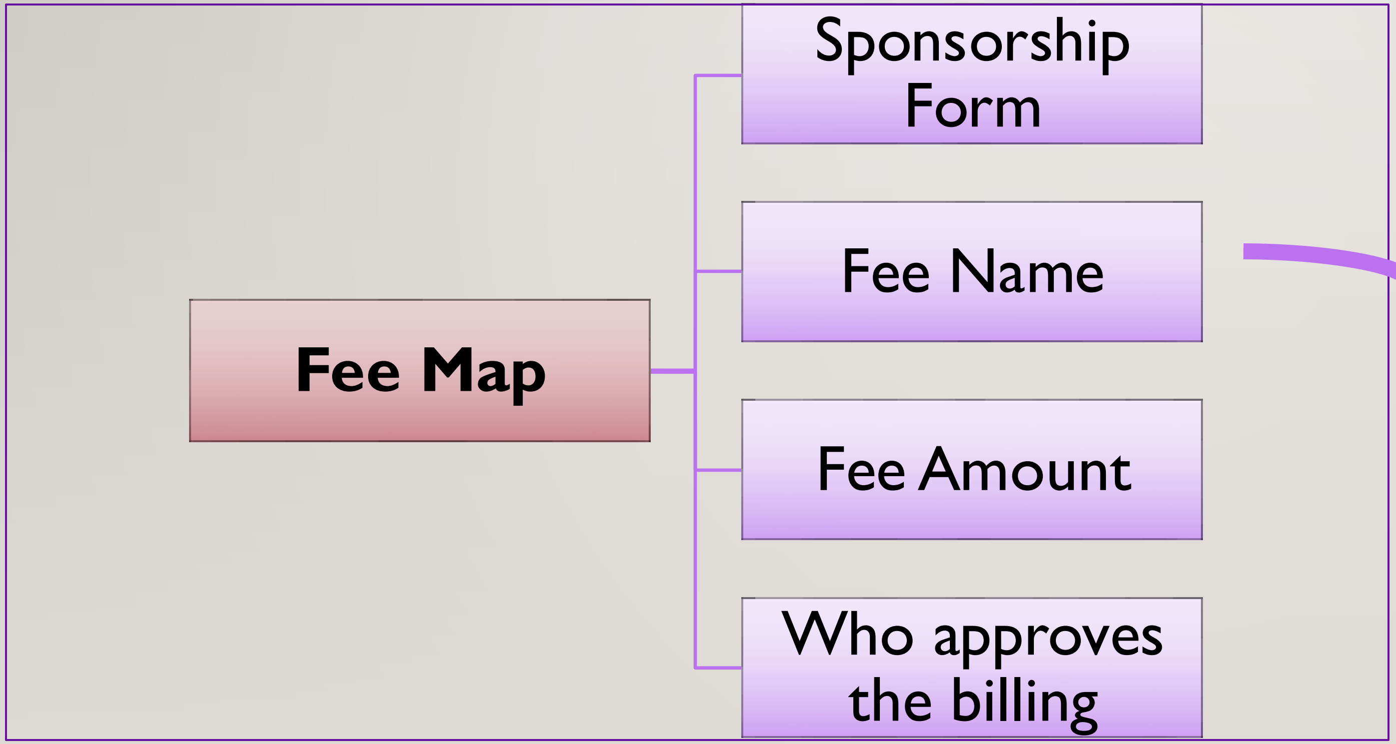
III) How to reconcile (Manual)

- Custom reports for Finance
- Finance manually updates the fee because three different systems are involved.



FEE MAPPING TABLE

We are using a map data object to manage and configure fees



Visitor's Fee Table
Visitor ID
Fee Name
Fee Amount
Invoice Status
Due Date
Has the fee been reconciled?
Paid Date
Payment ID

FEE TYPES

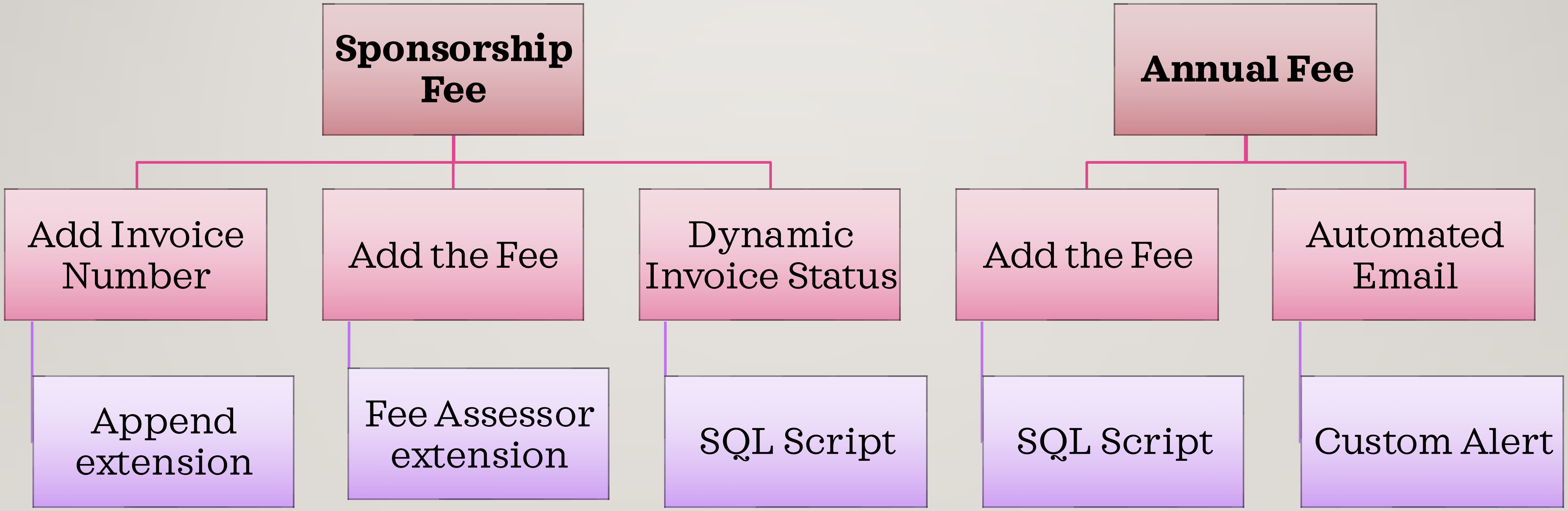
Sponsorship Fees

- Added with an e-form extension
- Append the e-form recnum to use as the unique invoice number (paper trail)
- A fee assessor creates a fee entry utilizing the information on the fee map table

Annual Fees for Scholars

- Requires an institutional funding level to be populated
- A SQL script adds the fee based on the DS-2019 start date and anniversary dates
- A custom alert is used to send reminder emails

FEE TYPES


Sponsorship Fee
Add Invoice Number
Append extension
Add the Fee
Fee Assessor extension
Dynamic Invoice Status
SQL Script
Annual Fee
Add the Fee
SQL Script
Automated Email
Custom Alert

INVOICE STATUS

Using a dynamic invoice status

Sponsorship Fees require an invoice status to indicate when it's ready for collection

- Not billable yet
- Fee is ready to be processed

Fee approval to is based on three scenarios:

1. Single e-form approval
2. E-form group approval
3. Second Approver

SAFEGUARDS

Financial Concerns
Certain fields can’t be changed by finance
Avoid duplicate fee charges
Avoid incorrect fee amounts
E-forms are often used for invoice documentation (invoice number)
The invoice status is a gate keeper for when a fee can be collected but it cannot reverse the status

PROJECT FILES AND CUSTOMIZATIONS

- **istart/eform-extensions**
  - UGA Append Fields
  - UGA Assess Fee
- **ioffice/xml**
  - dataobjects_code_institution.xml
  - dataobjects_jb_institution.xml
  - dataobjects_map_institution.xml
  - display_institution.xml
- **ioffice/alerts**
  - J1Scholar UGA Annual Fee
- **IOM**
  - Custom table
  - Funding template
  - Emails
  - Reports
- **SQL Scripts Tasks**
  - Add the annual fee
  - Update the invoice status

TESTING, DEMO, WRAP-UP

A NOTE ON LOGGING

- You can cut down on development time with helpful logging.
- There are several methods to use logging in ColdFusion and sunapsis. In this project, we relied on sunapsis for custom logging to the iOfficeLog in the database.
- Code sample for informational logging to iOfficeLog. This method takes four arguments for each column in the table.

```
new istart.core.ErrorLogger().logMessage("Component", "Func", "Usr" ,"Msg");
```

```
component extends="AbstractSimpleAlert" {

    public AlertType function getAlertType() {
        new istart.core.ErrorLogger().logMessage("ISCF Reminder", "getQueryString", "" ,"");

        var alertType = new AlertType();

        alertType.setServiceID(getImplementedServiceID());
        alertType.setAlertName(getServiceLabelType() & "UGA Recurring ISCF Fee Alert");
```

IOFFICE LOG

View System Logs

The following provides access to view sunapsis system logs. Log entries with message detail in HTML have links to open in a new window. By default this will show the 100 most recent error entries.

You can use the filters at the bottom of the page to add the informational entries. You can also set a starting error id if you're coming here because of a specific error message or to go back further than 100 entries. Finally you can choose a date to start listing on a particular day. In all cases, the most recent entries appear first, so filtering to a day will bring you to log entries from the end of that day, and subsequent pages will bring you earlier in the day.

Show 10 entries

Error ID	Application Area/Level	Component	Function	User	Message	Datestamp
32407	info	ISCF Due - query2	getQueryString	micah	View Full Message	03/04/2025 01:00:10.487
32406	info	ISCF Due - query2	getQueryString	micah	View Full Message	03/04/2025 01:00:10.453
32405	info	ISCF Due - query2	getQueryString	micah	View Full Message	03/04/2025 01:00:10.377

TESTING

Before launching

- Use a testing to-do list to cover basic interactions with sunapsis (ala unit testing)
- Use a development server for the proof of concept before making changes in production

After launching

- Maintain code and map tables
- Use a versioning system to keep up with changes over time

WRAP UP

sunapsis can be tricky and helpful

1. New alert groups and custom alerts need a correct setup to work; this can take time (campus, alert level)
2. On the helpful side, you can move slowly to automated emails (email template)

Understanding the different unit needs requires continued collaboration to ensure requirements are being met

Communication is vital – Utilize tools like Microsoft Teams and Microsoft Loop

SHAREABLES

- Sunapsis Community on GitHub

  Demo video, PowerPoint slides, project code files

  github.com/sunapsis-community/custom-projects/tree/main/uga.edu/Fee-Tracker-Project

Previous Tech Session

- 2024 - Beginner IT Skill for sunapsis – Password is conference
- 2019 – Sunapsis Workshp Materials
- 2017 - Reporting Gone Rogue
- 2014 - Sunapsis Extensions

DEMO

1. Add e-form extension
2. Add a fee configuration entry
3. Submit an e-form
4. Check reports
5. Approve the e-form
6. Run SQL script
7. Check reports
8. Reconcile
9. Check reports

QUESTIONS