WWU Department of Computer Science

JumpStart Scheduler

Building Where the People Are

Catherine Thomsen

Walla Walla University
SEVENTH-DAY ADVENTIST HIGHER EDUCATION

ABSTRACT

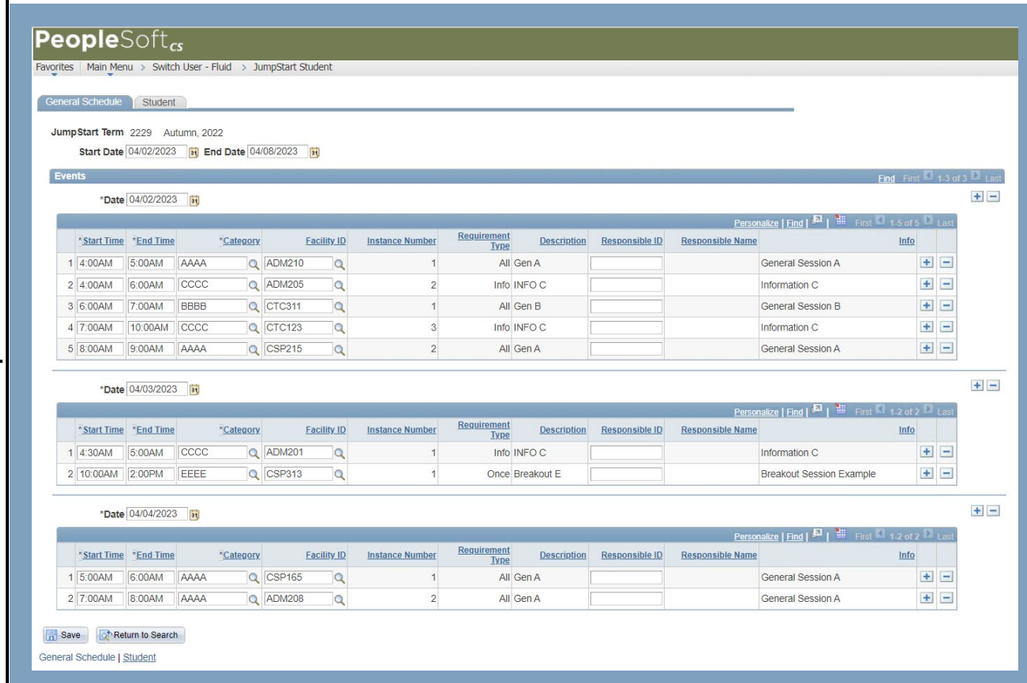
The current scheduler for Walla Walla University's freshman orientation (JumpStart) was designed using an outside system leading to redundant manual data entry and confusing formatting. The goal of this project was to combat these problems by creating a scheduler which utilizes our current database and set up a more user-friendly UI. TSQL and PeopleCode offer a different way of visualizing and interacting with data to streamline the process of schedule creation. The new JumpStart Scheduler implements the ability to edit student schedules manually, create group schedules, and automatically generate schedules for a given group of students. The hierarchy of T-SQL working together with People Code creates the structure needed to understand the complexity of the schedule as a whole and provides the user with a clean working environment to understand what is going on.

INTRODUCTION

JumpStart changed scope during the production of this project. Sessions were previously divided by who was required to go and type, focusing on electives to create a unique student schedule. The Advisement office decided to shift how this works by focusing unique setups into group sessions hoping to create easier to understand schedules for students. With this goal in mind, I divided my work into 3 main categories: general sessions, group sessions, and student schedules. Student schedules would need to be compatible with both group and general sessions but group sessions would need the additional identifier of group number. PeopleCode nicely accommodates this through the use of related records which allow hierarchies of data to be combined with further information based on a shared set of keys.

PROJECT IMPLEMENTATION

People code is designed to sit on top of a database and interact with T-SQL scripts. Because record keys influence page set up, components naturally form around shared sets of information. Much of my work was in setting up a logical flow for user entry so that set up information would logically trickle down to the student. This example of the general schedule shows how the time range dictates which days can be added, then category instances are restricted based on category, instance, and time.



These items are created by the user and subsequently scheduled during the pre and post build event, based on the category type, onto the student schedules. The user can also add items manually such as Athletics holds or adjusted breakout sessions.

The actual generation of the schedules is a collaboration between People Code and T-SQL scripts. Some requirement types such as ALL or GRP get scheduled on every student in their subset and can be inserted through T-SQL. Breakout (ONCE) sessions on the other hand are scheduled conditionally based on the student schedule and random selection.

SUMMARY

This project gave me a deeper understanding of how databases are set up and why. Throughout the process there were several times where I found myself reassessing my original concept of the scheduler based on the structure of the system I was working with. How is a leader associated with a session? Can there be multiple people in charge? How will modification to the schedule effect previous years? All of these questions and many more influenced me to seek out a more flexible design, but also to take advantage of the structure that already existed. Developing in a system I had used from the front end also deepened my appreciation for existing components and the systems which I use every day.

REFERENCES

PeopleCode Documentation:
https://docs.oracle.com/cd/F52213_01/pt859pbr3/eng/pt/index.html
Zach Parker, Enterprise Software Engineer at Walla Walla University

T-SQL

PeopleSoft

ORACLE

May 30, 2023, Computer Science Department