

Willowbrook School Case Study

L09: System Design

Philip Squires

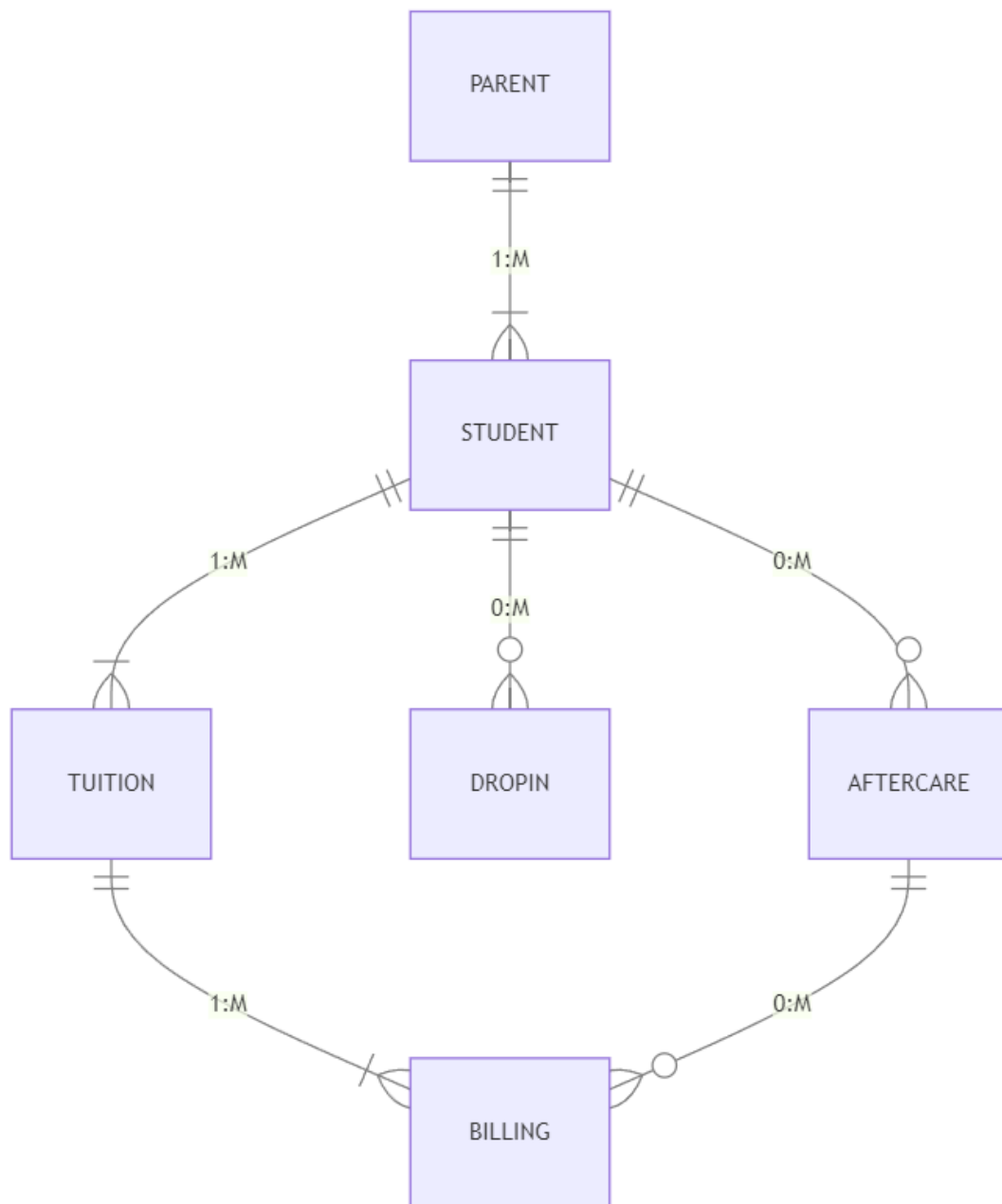
Pts5167

IST260

Professor Long

3/28/23

Willowbrook ERD



The ERD consists of six main entities that make up the student information and billing system: PARENT, STUDENT, TUITION, AFTERCARE, DROPIN, and BILLING.

The relationship between parent and student is one-to-many (1:M), which implies that while each student in this scenario only has one parent designated as accountable, a parent may have one or more students enrolled at Willowbrook.

A one-to-many (1:M) relationship exists between student and tuition, meaning that any student may have one or more records for their tuition, each of which may reflect a different payment schedule or set of fees. There is just one student listed for each tuition document.

The relationship between student and dropin is zero-to-many (0:M), meaning a student may or may not have one or more drop-in records for using aftercare services on a drop-in basis and each drop-in record belongs to a specific student. I'll admit I am not 100% confident in the design and lack of connection to other tables that dropin has.

Student and aftercare have a zero-to-many relationship (0:M). This was done on the assumption that not every student uses the aftercare services and those that do may have one or more entries in the aftercare table. This is the same reason aftercare and billing have a zero-to-many relationship.

A one-to-many (1:M) relationship exists between tuition and billing, meaning that each tuition record may be connected to one or more billing records that represent various billing cycles or payment statuses. A single tuition record is linked to each billing record.

Table Design

For the table design below and for use in the information system, I would likely include far more informational (non-key) fields in each table for the sake of greater functionality in the final information system. I tried to think of everything I could but realized when making the prototype that there were a lot more fields I could have added, thus you may notice there are fields in the prototype that don't exist in the tables. The table below shows each Primary Key and Foreign Key marked by (PK) and (FK) and the most important fields.

Parent	
int	ParentID(PK)
string	FirstName
string	LastName
string	Email
int	Phone

has

Student	
int	StudentID(PK)
string	FirstName
string	LastName
string	Teacher
int	ParentID(FK)

has

uses

has

has

Tuition	
int	TuitionID(PK)
int	StudentID(FK)
double	Amount
double	Discount
string	PaymentOption

DropIn	
int	DropInID(PK)
int	StudentID(FK)
int	AfterCareID(FK)
date	Date
time	StartTime
time	EndTime

AfterCare	
int	AfterCareID(PK)
int	StudentID(FK)
string	TimeBlock
double	FixedFee
double	DropInFee

part of

part of

Billing	
int	BillingID(PK)
int	StudentID(FK)
int	TuitionID(FK)
int	AfterCareID(FK)
double	TotalMonthlyFee
double	PaymentReceived
double	AmountPaidToDate
double	AmountOutstanding

Willowbrook School

Student Management

Student Management

Billing Management

Search

Student Name

▼

Search

Add New Student

Student ID	First Name	Last Name	Parent Name	Teacher	Actions
1001	Billy	McDonaldsonberg	Laura McDonaldsonberg	Ms. Mathenson	<div>Hide Details</div> <div>Edit</div> <div>Delete</div>
<div><div>Student Information</div><div><div>First Name:</div>Billy</div><div><div>Last Name:</div>McDonaldsonberg</div><div><div>Date of Birth:</div>January 15, 2011</div><div><div>Address:</div>123 Willowbrook Lane, Fakeville, PA</div><div><div>Allergies:</div>None</div><div><div>Medical Conditions:</div>None</div><div><div>Grade:</div>5th</div><div><div>Enrollment Date:</div>September 1, 2016</div><div><div>Teacher:</div>Sandra Mathenson</div><div><div>Parent Information</div><div><div>First Name:</div>Laura</div><div><div>Last Name:</div>McDonaldsonberg</div><div><div>Address:</div>123 Willowbrook Lane, Fakeville, PA</div><div><div>Phone:</div>(123) 696-4602</div><div><div>Email:</div>lauramcd@gmail.com</div></div></div>					
1002	Samuel	Jackson	Samuella L. Jackson	Ms. Englishberger	<div>Detailed Record</div> <div>Edit</div> <div>Delete</div>

In the first prototype, there are many control features all in the form of buttons excluding the search box. You can press the “Detailed Record” button to expand and show significantly more detailed student information. It changes to “Hide Details” on press. You can edit student records via the “edit” button which makes all the fields editable. In practice, I think editing would be done on a separate page, like how the “add new student” button would function. There is a search box with drop-down options to search by such as student name, parent name, and teacher. Every entry has a delete button to delete student records. This button would theoretically have a confirmation pop-up on press to prevent accidental deletions. The “Add new student” button would create a new entry into the “Student” and “Parent” tables and would take you to a new page. The buttons up top would be for navigation.

Evaluation of Usability:

- Visibility of system status: Doesn’t contain any notifications or feedback about system status.
- Match between system and the real world: Everything is easily understood by anyone that works at Willowbrook because I used familiar terminology.
- User control and freedom: There is definitely very little customization and no undo buttons or anything.
- Consistency and standards: I believe it is consistent.
- Error prevention: No error prevention is shown, but there would be a confirmation pop-up when trying to delete an entry and the default selection (in case of accidental button pressing) would be cancel.
- Recognition rather than recall: No memorization required to use the system since all the options are right there on the page in front of you.
- Flexibility and efficiency of use: I don’t think it contains any accelerators, and I’m not sure what that could be for the student management system.
- Aesthetic and minimalist design: It’s definitely minimalistic, but aesthetics aren’t its strong point.
- Help users recognize, diagnose, and recover from errors: There is no example of this shown, but presumably if an invalid form or search input is received it would highlight the text field and provide clear language indicating the problem.
- Help and documentation: There is no indication that help or documentation is available.

Willowbrook School Billing Management

Logout

Student Management Billing Management

Search

New Manual Entry Reports Outstanding Balances Print All Monthly Bills

Billing ID	Student Name	Parent Name	Total Balance	Paid to Date	Outstanding Balance	Actions
1	Billy McDonaldsonberg	Laura McDonaldsonberg	\$500.00	\$100.00	\$400.00	<div>Detailed ViewEditDeletePrint Statement</div>
2	Samuel L. Jackson	Samuella L. Jackson	\$350.00	\$150.00	\$200.00	<div>Detailed ViewEditDeletePrint Statement</div>

Previous123Next

Help

For the second prototype, there are significantly more options. The reports button would be the place to view and print all of the automatically generated reports, as well as custom reports. The outstanding balances button would take you to the list of students with outstanding balances. The “Print All Monthly Bills” button would allow you to print all students bills (that have a balance). The detailed view button would be like the student management page but give a detailed breakdown of all expenses and fees, including aftercare and drop-in related fees. There is a logout button to account for the fact that this system would require a login because it contains sensitive information, though so would the student management system. I added a navigation bar at the bottom as well as a help button. There would be tooltips on hover over the buttons that explain what they do. I made the delete button red, to signify its importance, but there would be a confirmation pop-up when pressed.

Evaluation of Usability:

- Visibility of system status: The system would display a loading bar if required when moving from screen to screen.
- Match between system and the real world: Everything seems easy to understand, especially for the administrator responsible for billing.
- User control and freedom: There would be an undo and redo button when editing a record. It would keep a history of changes up to a certain extent, likely 3 changes per record.

- Consistency and standards: I believe it is consistent.
- Error prevention: The only error prevention shown is the red “delete” button, but there would be a confirmation pop-up when trying to delete an entry as well.
- Recognition rather than recall: There would be minimal memorization required. Some of the features will take you to a new page with many customization options, which could be confusing, but the system would allow you to save settings so that you can easily apply the style you want after the first time, for example, printing the bills.
- Flexibility and efficiency of use: All of the individual billing records contain a “print statement” button, but at the end of every month you would want to just print them all. The “print all monthly bills” button would automatically print all statements with outstanding balances. If parents signed up electronically, perhaps they could automatically get their bills via email and pay online.
- Aesthetic and minimalist design: The design is definitely minimalistic and more aesthetic than the first prototype.
- Help users recognize, diagnose, and recover from errors: The prototype doesn’t show any example of what happens in the case of errors. In the event of an error, the problem would be plainly stated and highlighted (if it’s a problem with a specific field).
- Help and documentation: There is a “help” button at the bottom which would contain a tour as well as help using the system and its features to the fullest. You would be able to, for example, learn how to design a monthly statement template to your liking and save it for use later.