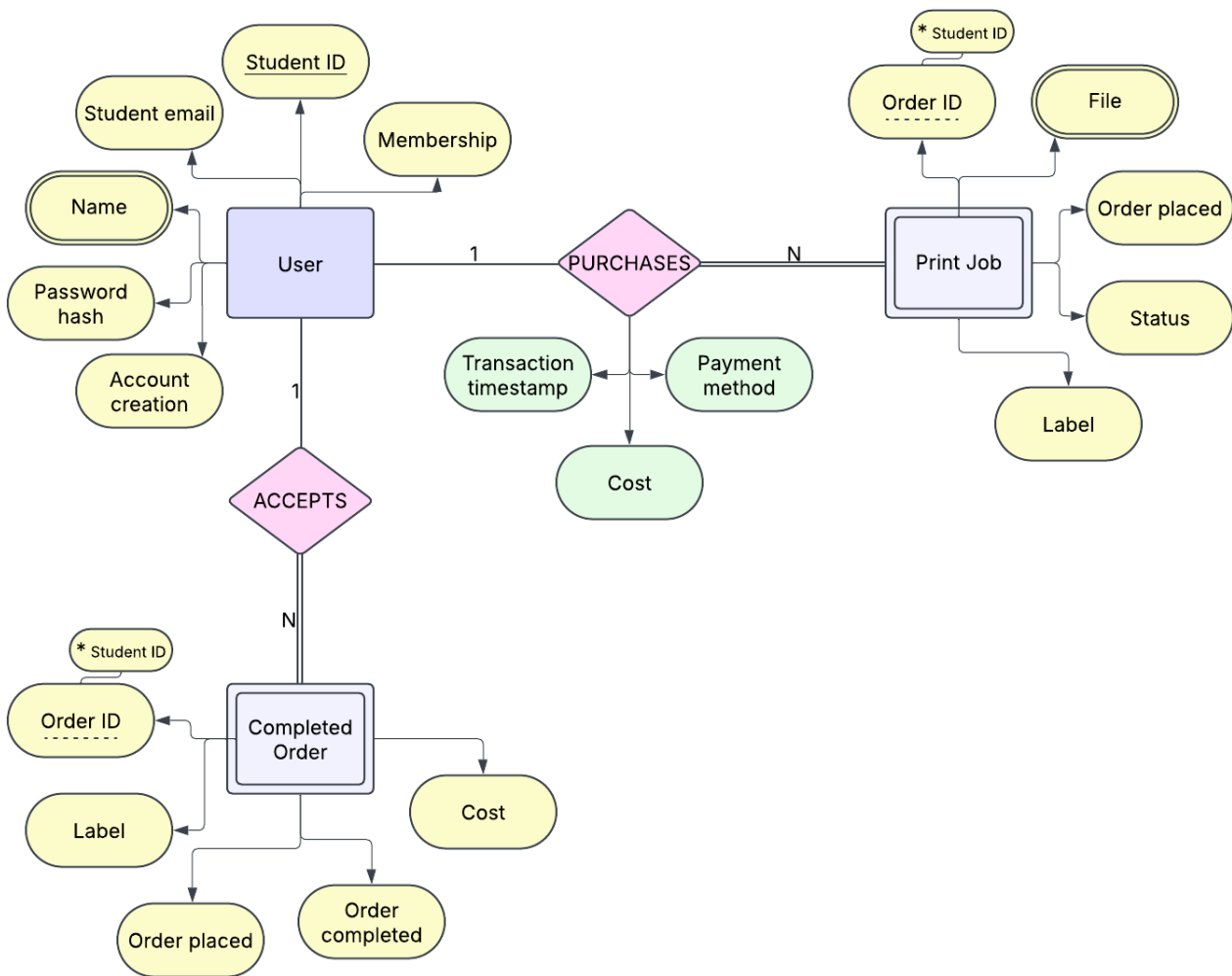# 3DW Database Documentation

1. ER Diagram

Figure 1: Entity-Relationship Diagram for 3DW Website Database
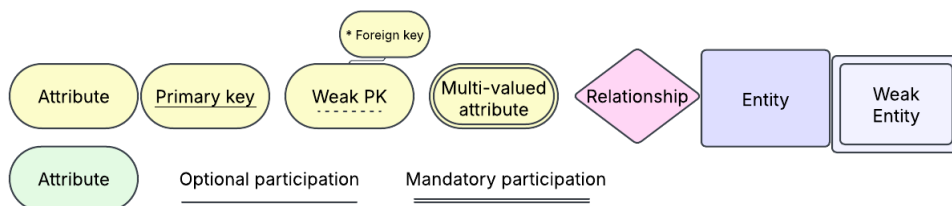

Figure 2: ER Diagram Legend

2. Entities

User - (a user account on the website)

- **Student ID**: User's UWO student number; serves as a <u>primary key</u> for the database and a <u>UUID</u> for the site.

- **Student email**: User's UWO email; used for account verification and notifications.

- **Name**: User's first and last name.

- **Password hash**: User's encrypted password.

- **Account creation**: Timestamp for when the account was created.

- **Membership**: "Member" | "Non-member"; can be used to separate orders into separate queues.

Print Job - (an order currently in queue)

- **Order ID**: User's nth order; coupled with Student ID <u>foreign key</u> of its parent entity to form the <u>primary key</u> for this entity.

- **File**: Information for the file that the user uploaded.

- **Order placed**: Timestamp marking when the order was placed into queue.

- **Status**: "In Queue" | "Printing" | "Finished"

- **Label**: Title or short description that the user can provide to label their order.

Completed Order - (a previous order)

- **Order ID**: User's nth order; coupled with Student ID <u>foreign key</u> of its parent entity to form the <u>primary key</u> for this entity.

- **Label**: Title or short description that the user can provide to label their order.

- **Order placed**: Timestamp marking when the order was placed into queue.

- **Order completed**: Timestamp marking when the order was completed; can be used together with the order placed timestamp for administrative purposes.

- **Cost**: Price of the order; can be displayed to the user or used for administrative purposes.

3. Relationships

Purchases - (a user purchases a print job, <u>one-to-many</u>)

- **Transaction timestamp**: Time of transaction.

- **Cost**: Transaction cost.

- **Payment method**: Method of payment used in the transaction.

Accepts - (a user accepts their completed order, <u>one-to-many</u>)

*In-progress and completed orders will be placed into separate tables for organizational and efficiency reasons; i.e. no need to query over past orders when updating the queue.