

# Technical Drawing Data Extraction System

Northeast Precast SWE I Sponsor Project Proposal

## About the Sponsor

[Northeast Precast](#) is a manufacturing plant and construction firm located in Vineland. We design, produce, and build large structures like warehouses, parking garages, and highway components from precast concrete. Many of our engineers and software developers are Rowan alumni. Unlike many similar companies, Northeast Precast places a great deal of emphasis on technological innovation – especially in terms of software. Our ERP system is developed in-house, and our software development team is growing quickly.

## Project Background

Precast construction is unique in that each piece of the finished result is produced in our manufacturing plant. A “piece” is typically a huge rectangular wall section that weighs many tons and must be moved with large cranes and shipped via trailer to the construction site, where all the pieces are finally assembled by a skilled erection team.

A single warehouse may be built from hundreds of wall sections, and each wall section may have many components (e.g., doors, windows, fasteners, decorative elements) that must be placed with extreme accuracy so everything lines up when they’re all put together. Many of the components also contribute to the structural integrity of the end result.



Our fabrication shop is responsible for producing each piece to the exact specifications required for the project, and these specifications are all laid out in technical drawings produced by a team of engineers. Each and every piece is given its own technical drawing document (called a “Shop Ticket”) produced by a team of engineers. It includes a great deal of information about the piece that must be built, including its overall dimensions, the

[illegible]

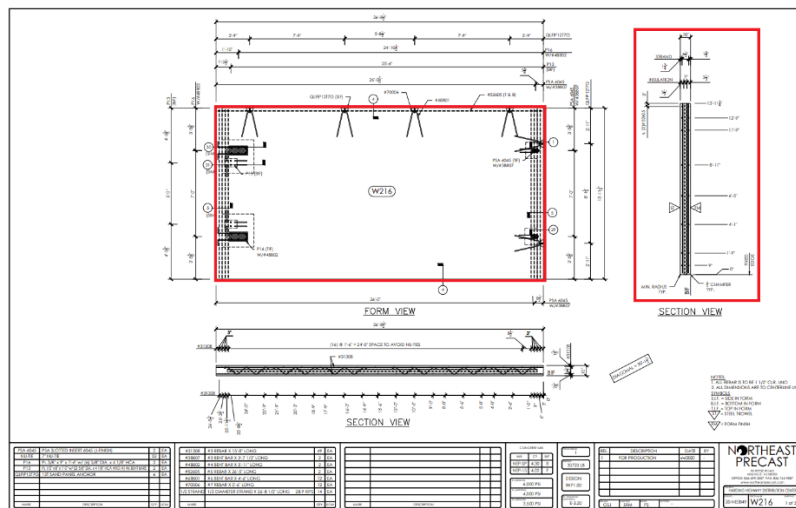
- A. Saves the engineering team lots of time and effort
- B. Stores data in an organized format to enable reporting and data analytics

Given a shop ticket document (PDF format), the data extraction system should be able to return an object with the properties displayed below. (We have designed a proof-of-concept to confirm this is a feasible task for a small team in 10 weeks).

Property Name	Type	Description
NumberOfPages	int	Number of pages in the PDF file.
PageNames	string?[]	Page names extracted from view labels.
FileName	string?	File name of the PDF file.
FileNamePieceMark	string?	Piece Mark extracted from the file name.
ProjectNumber	string?	Project Number from the title block labelled "JOB NO.".
ProjectName	string?	Project Name from the title block labelled "PROJECT:".
FileContentPieceMark	string?	Piece Mark from the title block labelled "PIECE MARK".
ControlNumbers	string[]	Control numbers from the square above the title block labelled "CONTROL NO.:".
PiecesRequired	int?	Pieces required from the title block labelled "PIECES REQ'D:".
Weight	decimal?	Weight from the title block labelled "WEIGHT:".
DesignNumber	string?	Design number from the title block labelled "DESIGN:".
RectanglePage	int	0-based index of the page containing form and section view rectangles.
FormViewRectangleX	double?	Distance from left edge of PDF to left edge of the form view rectangle (inches).
FormViewRectangleY	double?	Distance from top edge of PDF to top edge of the form view rectangle (inches).
FormViewRectangleWidth	double?	Width of the form view rectangle (inches).
FormViewRectangleHeight	double?	Height of the form view rectangle (inches).

Property Name	Type	Description
SectionViewRectangleX	double?	Distance from left edge of PDF to left edge of the section view rectangle (inches).
SectionViewRectangleY	double?	Distance from top edge of PDF to top edge of the section view rectangle (inches).
SectionViewRectangleWidth	double?	Width of the section view rectangle (inches).
SectionViewRectangleHeight	double?	Height of the section view rectangle (inches).

There are many viable ways the student teams can choose to tackle this project. For instance, our proof of concept uses libraries like PDF.js to extract shape vectors and text



content from the PDF to ascertain the data; but student teams interested in ML/AI may be more inclined to rasterize the PDF and use an object detection model to identify the outlines of important images like the Section View. We have tens of thousands of similar drawings that can be used as training data.

If the student team needs more visual appeal for their end-of-semester presentations, we can also develop specifications for a user interface / reporting dashboard that can be used to showcase their work. Our main objective for the sponsorship is to build relationships with the Rowan student body by offering a project that is fun and rewarding to work on.