Abstract
This project constructed a friendly user interface, enabling people without Software Engineering skills to construct mainframe transactions from the contents of source files. We utilized common open-source tools such as AngularJS and SQLite to construct the solution. The solution enables users to map source data to mainframe transactions while defining cross-reference and data quality rules. As the user constructs the map, they have the ability to “preview” the results and manage versions of rules, mainframe transaction formats, and data mappings. This project reduces usage of expensive Software Engineering resources, shortens deployment time to the mainframe by preventing user input errors, and saves money through an easily deployable endpoint.

Architecture

Impact
Our project’s impact is mainly witnessed in the shorter time needed to process loans and other banking documents. Additionally, our project saves valuable time from software engineers by shifting tedious data entry work to business developers. The project allows for greater error checking abilities as data quality checks are automatically performed via the interactive UI, and users can revert to previous versions of their work by accessing the revisions and versioning system.

Metric
Current AS-IS production takes 4 hours to complete each data element (includes mapping, testing, etc.). Potentially, this project will reduce that to 1 hour, greatly saving Software Engineering resources.

<table>
<thead>
<tr>
<th>Goals</th>
<th>Testing Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>95% of API calls complete in under 1000ms</td>
<td>99% of API calls complete in under 1000ms</td>
</tr>
<tr>
<td>Web pages load in under 3 seconds</td>
<td>100% of web pages load in under 3 seconds</td>
</tr>
</tbody>
</table>

Results
The Transaction Mapper provides multiple core functionalities with room for future expansion. The back-end server was written to integrate with existing code provided by the company. The key features implemented include:

- RESTful API
  - Load, store, retrieve, and delete user-created mapping files as well as metadata from the database
  - Apply user created mapping files to transform vendor source data into 102-byte packets accepted by the MSP mainframe.
- Web Interface
  - Extract XML data from vendor and internal sources
  - Display complex relationship mappings in a simple, object-oriented system
  - Manage multiple projects via basic versioning and revision system
  - Edit, clone, delete
- Future Features to Implement
  - Expand the number of file formats accepted as input
  - Implement sign-in system to protect sensitive information
  - Allow administrators to edit metadata

Summary
Throughout the semester, our team worked with Capital One to create a web application which allows business developers to define translations between two formats. The primary advantage of this system is in alleviating workload from software engineers who previously completed this task. Some pitfalls our team experienced over the course of this project include learning new technologies, and working with an existing code base. All team members entered the project with no web development experience. Additionally, parts of the existing code base did not work as intended and required revision before progress could be made on the transaction mapper.

Acknowledgements
Our team would like to thank our faculty advisor, Dr. Neeraj Mittal, and our company sponsor, Joe Vaughan for their guidance and expertise during this project.