Project Requirements Document 1

Project Title: Automated 3 Way Match (tentative)
Team Name: $-flow

Members:	**Email**:
● Millan Batra [Lead]	millanbatra@umail.ucsb.edu
● Yoon Lee [Scribe]	yoonlee@ucsb.edu
● Dennis Fong	dfong@ucsb.edu
● Alexander Kang	alexanderkang@ucsb.edu
● Alexander Yuen	atyuen@ucsb.edu

Intro

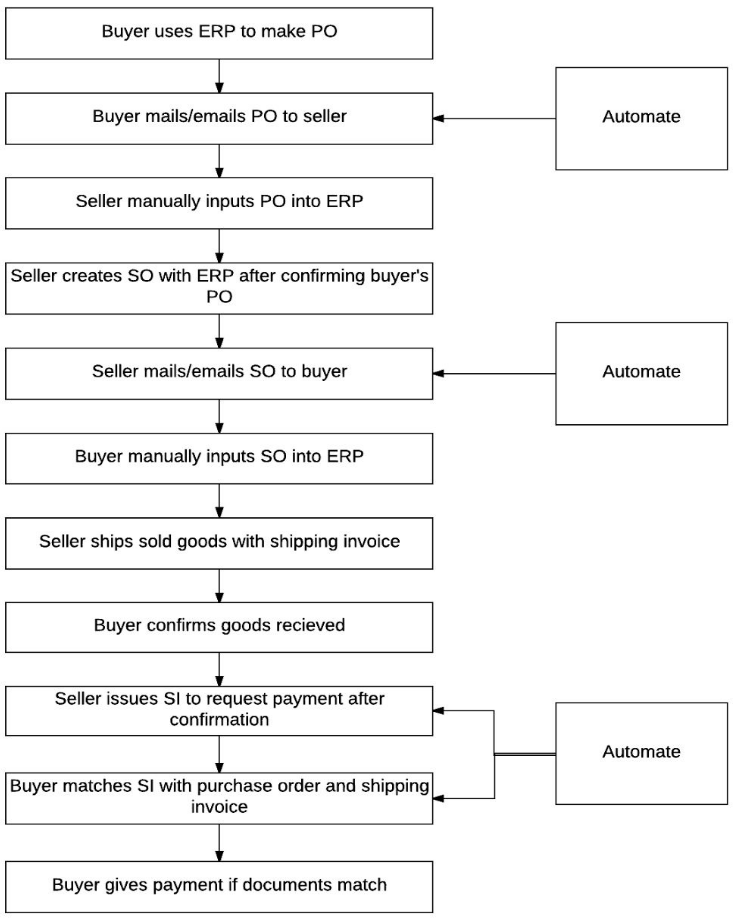
IMPORTANT TERMS

Purchase Order (PO) - Describes the price, quality, and quality of goods to be purchased.

Sales Order (SO) - confirms the PO details.

Enterprice Resource Planner (ERP) - Creates SOs and POs based on a forecast

Sales Invoice (SI) - Describes price, quality, and quantiity of goods in the shipment and may include PO and/or SO identifiers.


Buyer uses ERP to make PO
Buyer mails/emails PO to seller
Automate
Seller manually inputs PO into ERP
Seller creates SO with ERP after confirming buyer's PO
Seller mails/emails SO to buyer
Automate
Buyer manually inputs SO into ERP
Seller ships sold goods with shipping invoice
Buyer confirms goods recieved
Seller issues SI to request payment after confirmation
Automate
Buyer matches SI with purchase order and shipping invoice
Buyer gives payment if documents match

Problem

The current purchase order pipeline imposes up to a 24 hour delay before the data is available online and has the potential for human error. This is due to the large amount of manual input required when attempting to match a purchase order, shipping order, and shipping invoice. This process continues to have many steps which introduce errors and delay planners, purchasers, and accountants by up to 24 hours.

Project Specifics

We will be creating a mobile app that will help reduce the errors and delays associated with the purchasing process. Our mobile application will allow users to take pictures of purchase orders, shipping orders, and shipping invoices and upload the relevant data to our system. Once uploaded, we will match the documents to one another and map this back to the customer's ERP system.

Innovation

This mobile application aims to reduce delays in the purchase order pipeline by providing a clean and efficient platform that automates document matching. Companies are stuck in the past passing around paperwork between companies and scanning documents into their ERP systems. Our service will be the bridge that helps bring companies into the present by implementing an automated purchase order pipeline.

Goals

The goal of $-Flow is to help alleviate the delay between the time an invoice is created and the time it is inputted into a company's ERP system. Our app will help match related invoices and allow companies to retrieve this information from our database. We will also have an OCR pipeline which will be able to identify and parse invoices when given an image.

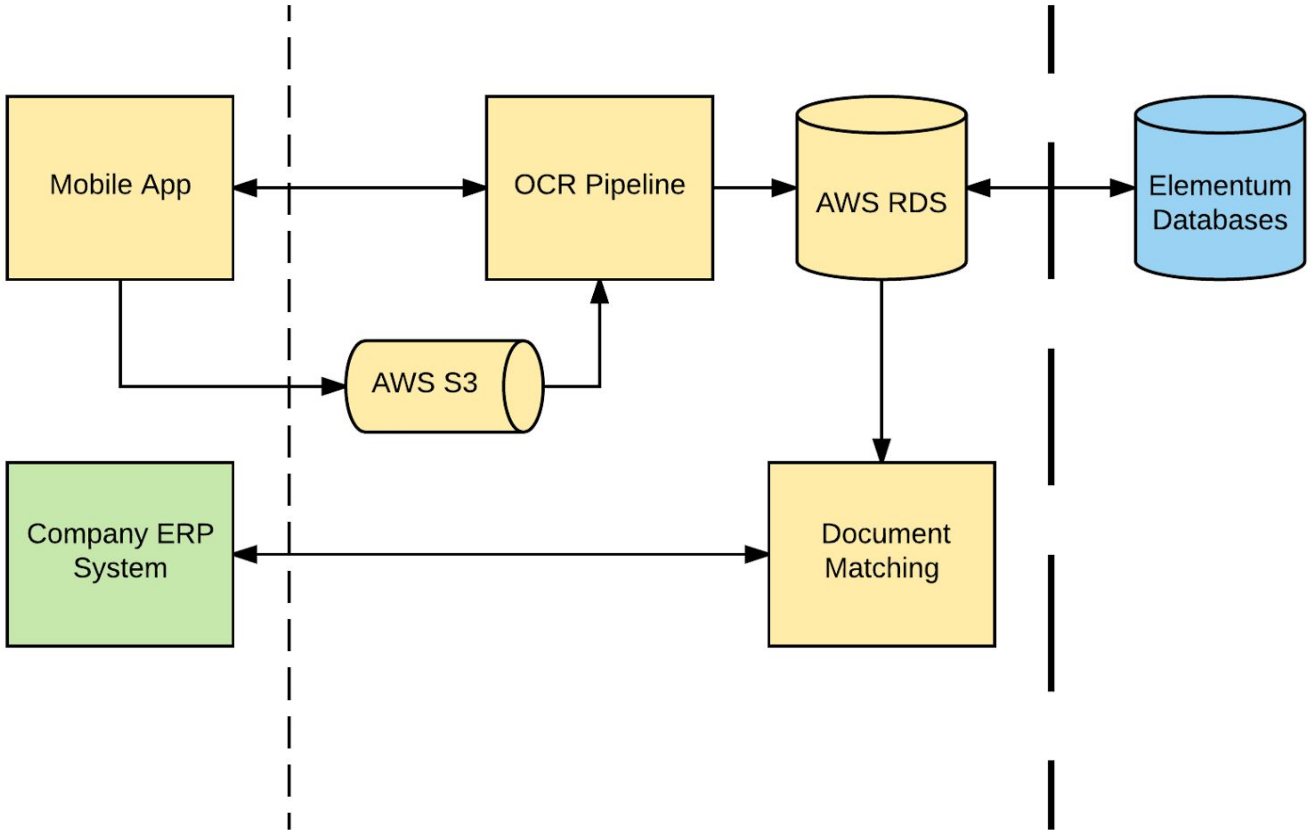
Background

There are no systems currently in place to address these inefficiencies. With this solution, we will help Elementum with their mission to transition the world from one based on excel spreadsheets and emails to one based solely on data.

Assumptions

The documents that we are parsing will contain no deviants; each company's documents will maintain consistent formatting. Documents given to the system should be printed and contain no handwriting. We will not be expected to parse handwritten text. Upon opening the app, we assume the user's device will have strong connectivity to the internet and no interference. Once document matching has completed, it is the responsibility of the company to grab the data from our system since we do not have access to their enterprise resource planners.

System Architecture



Mobile App

The mobile app will be the platform for taking pictures and matching shipping orders, purchase orders, and shipping invoices. Users can register under their respective companies and will be restricted to viewing those company's documents. The app will upload images to the AWS S3 bucket and make API calls to the RESTful web service to notify users whether the document has been matched or not.

Document Matching

After the documents have been parsed, the parsed data will be matched against a pre-existing document's data stored in our backend. If the data in the documents is exactly the same, the backend will send an HTTP request to the mobile app to notify the user that the documents match. It will then proceed to send the documents to the appropriate parties.

OCR Pipeline

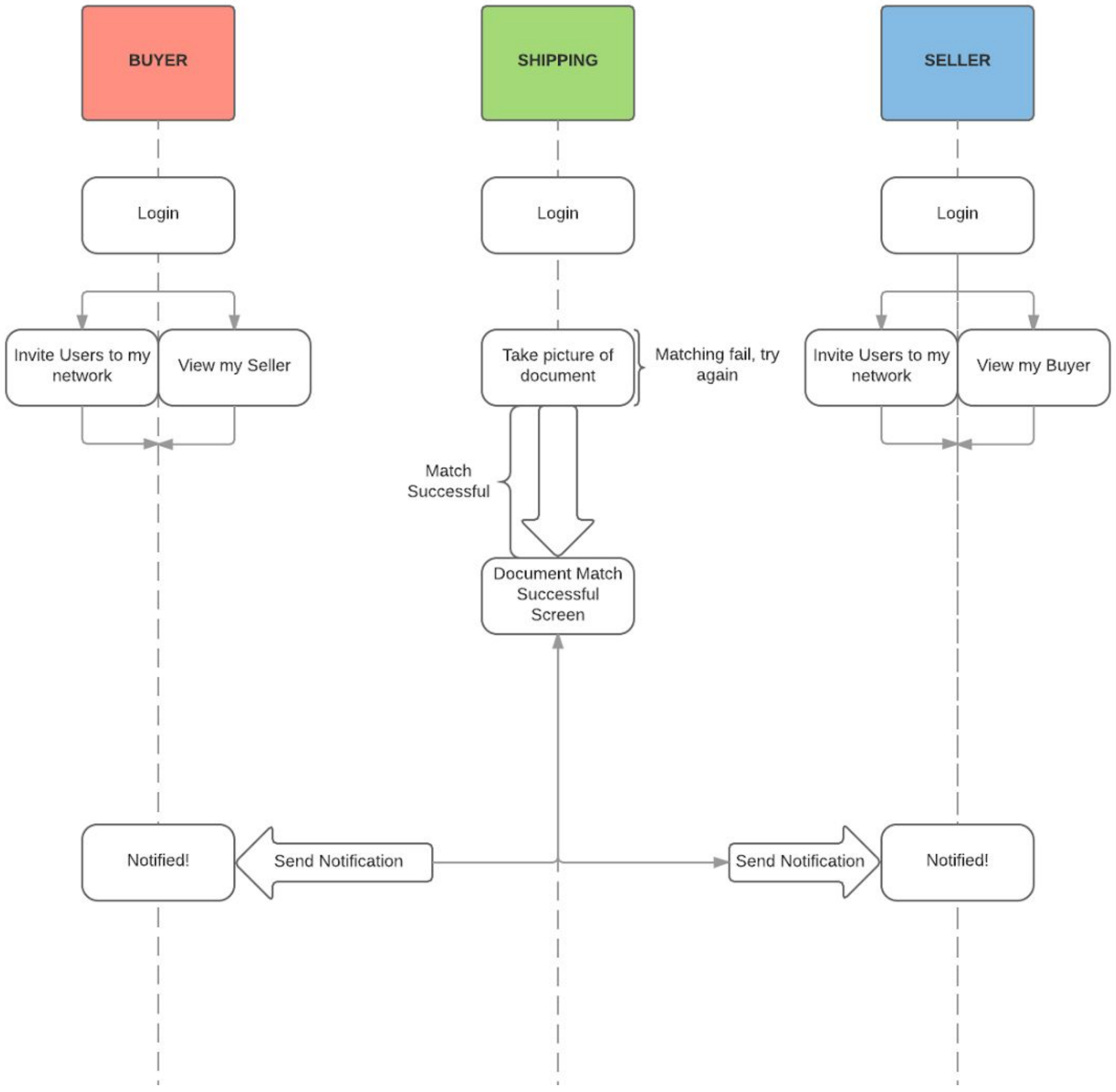
The OCR pipeline will receive requests from the mobile app to parse invoices that are stored in AWS S3. The pipeline will take care of preprocessing, text identification, and text parsing. Once the text is parsed, we will grab the relevant data from the parsed data and add this into our PostgreSQL database in AWS RDS.

System Requirements

1.1. Users can take pictures of invoices and decide whether or not to upload them
1.2. a. Once a picture is uploaded, we store it in file storage system (AWS S3). Each image will have a corresponding entry, with its metadata, in our database (AWS RDS) to keep track of it.
b. A job will be queued for our OCR pipeline to parse the text from the image.
1.3. Our OCR pipeline will take jobs off our job queue and execute the OCR pipeline.
1.4 Once the text is parsed from the image, we store the parsed data in our database. (AWS RDS)
1.5. After parsing is done, we match the recently parsed image with other referenceable entries.
1.6 We store the matched entries in our database.
2.1. Users can request document matches for documents give a unique id.

Requirements

User Interaction and Designs



User Stories

High Priority

1. As a user, I want to be able to take a picture inside the app and view it
    a. Acceptance Criteria
        i. Take a picture of a document given a screen where a user can take it. When the picture is taken, then the user can view it and delete it if the user doesn't like it.
2. As a user, I want to be able to upload pictures taken.
    a. Acceptance Criteria
        i. Upload a picture given a picture and a user clicking upload. When a user wants to upload a picture, then we want to store it in a filesystem and store its metadata in a database
3. As a company, I want to be able to match referenceable invoices so I can save time.
    a. Acceptance Criteria
        i. Many documents in the database. Given many invoices. When the picture is uploaded, the backend will handle matching the data within the picture to other data in the database.
4. As a user, I want to be notified whether the picture is matched or not.
    a. Prompt me to retake picture with tips on better picture.
        i. Better lighting
        ii. Angle it perpendicular to the invoice
    b. Picture is of good quality but there was no match - "contact supervisor"?
    c. Otherwise, display "Match approved" screen.

Medium Priority

1. As a buyer, I want to be able to select my sellers
    a. Acceptance Criteria
        i. Given a screen that prompts the user to enter in the seller, the app will make a GET request for authenticating permission
2. As a seller, I want to be able to view my buyers (multiple)
    a. Acceptance Criteria
        i. A page in the app will show a list of buyers that the user can navigate through easily.
3. As a user, I want all participating parties to be notified of the document match.
    a. Acceptance Criteria

            i. After a ‘document match’ event in the app, participating parties associated with the user will receive an e-mail notification with a timestamp and location.

Low Priority

1. As a user, I want to be able to log in to the app
    a. Acceptance Criteria
        i. Log in to the app. Given a login screen and a user giving username and password. On button click, user is logged in and routed to another page.
2. As a company, I want to be able to invite users to join my network. (unsure about this one) Is this MVP or added on later?
    a. Acceptance Criteria
        i. Company can login to the app and on the ‘invite’ page, can fill out a form (e-mail address?) to grant authentication to a user signed up under that e-mail address.

Appendix

Purchase Order (PO) - Document issued by the seller to the buyer that indicates types, quantities, and prices for purchased products/services
Sales Order (SO) - Document generated when a buyer purchases a product/service to confirm the purchase order details
Sales Invoice - A bill sent to the customer after a sale requesting for payment
Shipping invoice - Document that accompanies the sold goods when shipped. Lists price, quantity, and quality of goods and can also include PO and/or SO identifiers
Enterprise Resource Planner (ERP) - An ERP management information system integrates areas such as planning, purchasing, inventory, sales, marketing, finance and human resources.
Tesseract - Library used to perform OCR. Comes with packages for many languages but we will only be using english
OCR - Object Character Recognition
React Native - Framework that allows users to create universal mobile apps using Javascript

Technologies Employed

JIRA, BitBucket, Slack
Tesseract, Tensorflow
AWS S3, RDS, EC2, Lambda
React Native
Docker