



Student Number: 12300216

Student Name: John Brennan

Programme: CASE4 - B.Sc. in Computer Applications (Software Engineering)

Faculty: School of Computing

Academic Year: 2015/2016

Project Title: Final Year Project Functional Specification - Invoice Processing and
Visualisations - Sonrasc

Supervisor: Dr. Alexander O'Connor

Blog URL: johnbrennan.me


Module code: CA400

Project Due Date: 27th November 2015

Declaration

I the undersigned declare that the project material, which I now submit, is my own work. Any assistance received by way of borrowing from the work of others has been cited and acknowledged within the work. I make this declaration in the knowledge that a breach of the rules pertaining to project submission may carry serious consequences.

I am aware that the project will not be accepted unless this form has been handed in along with the project.

Signed: 

0. Table of Contents

- 0. Table of Contents
- 1. Introduction
 - 1.1 Overview
 - 1.2 Business Context
 - 1.3 Glossary
- 2. General Description
 - 2.1 Product / System Functions
 - 2.1.1 Front-end System
 - 2.1.2 Back-end System
 - 2.2 User Characteristics and Objectives
 - 2.3 Operational Scenarios
 - 2.3.1 New Data for the system
 - 2.3.2 Desire to view the data
 - 2.3.3 Info from a business
 - 2.3.4 Looking for information
 - 2.4 Constraints
- 3. Functional Requirements
 - 3.1 Uploading Invoices
 - 3.2 Information Extraction
 - 3.3 Graphical Representation of Invoices
 - 3.4 Business profiles
 - 3.5 Invoice view
 - 3.6 Searching
- 4. System Architecture
- 5. High-Level Design
- 6. Preliminary Schedule
 - 6.2 Hardware and software requirements
- 7. Appendices

1. Introduction

1.1 Overview

Sonrasc enables users to capture images of invoices, receive expenditure reports and aggregated views of all invoices currently on the system that are relevant to the business.

The purpose for this system is to provide business owners ("the users") in a given organisation a means of tracking their expenditure. Despite the wide-spread of e-business, electronic payments etc, many businesses still use paper invoicing. By having an electronic means of storing paper invoices, it will allow users to review and analyse spending within the company. It provides the user with a means of assessing the relevant information in relation to the company's expenditure.

This project will use its Application Programming Interface (API) to read financial information and will only be available to the dedicated front-end client application. It is the system's intent to be integrated into the business's existing expenditure processes. The overall goal is to augment a business's current existing process in dealing with invoices, removing the need for manual input of items purchased for tracking and bookkeeping purposes. Project management will also be aided when overseeing what has been purchased.

Sonrasc will use a classifier to recognise an invoice and extract its individual pieces of information using machine learning techniques. It will store this information in a database. A client can use the front-end application to review all the invoices uploaded to the system in a number of visualised displays.

Users will be able to see the list of businesses that have invoices in the system, including their contact information and the products purchased from an individual business.

1.2 Business Context

To have a system which is capable of identifying patterns and text in a given invoice, it must be trained using a data set. This data will be coming from Pepperstown Holsteins. This is a farm run by Gerard and Mark Brennan. It is a 270 acre dairy farm in Pepperstown, Ardee, County Louth.

Invoices are received regularly from a number of different sources from the operation: livestock purchasing, veterinary charges, machinery purchasing, machine maintenance, animal feed, building renovation and upkeep charges. The current process involves manually inputting the invoiced amounts into Excel spreadsheets for comparing budgetary projections to the actual expenditure. It is a time consuming process and a lot of information is lost due to the amount of time it would take to capture all the data manually.

Pepperstown Holsteins are providing all the invoices and relevant associated financial data for this projects proof of concept and building a classifier that can handle data relevant to the farm. The aim of this project is to produce a system that can be integrated into the daily operation of the farm. Whenever an invoice will come into the flow, the idea is that this project will provide an electronic means of storing its contents.

UCD Veterinary Hospital
UNIVERSITY COLLEGE DUBLIN
Belfield, Dublin 4, Ireland
Tel: (01) 716 6000 Fax: (01) 716 6005

Mr. Gerard Brennan
Pepperstown House
Ardee
Co. Louth

Invoice number: 105263
Case Number: 72839
Animal:
Owner: Gerard Brennan
Clinician: Eoin Ryan
Date submitted: 14/11/2019
Invoice Date: 14/11/2019

Invoice 105263

procedure or treatment	qty	rate	total	procedure or treatment cont.	qty	rate	total
Hospitalisation: Bovine	3	13.50	73.44				
Noroclay 100ml Per Mt	1	13.50	24.48				
Flunixin 100ml	125	21.00	40.80				
Norocaine 500ml Per Mt	80	21.00	22.85				
Charastat 5l	100	21.00	1.63				
Caspex 40% 400ml Per Mt	100	21.00	1.63				
R D A Surgery	1	21.00	3.57				
	1	13.50	81.60				
Ex VAT	Rate	VAT Amt.	Ex VAT				
155.17	13.5	21.35	216.42				
58.25	21	12.23	33.58				
			Total				250.00

A teaching discount of 56.37 has been given
Amount Outstanding 450.00

Vat Number: IE6517386K

An example of an invoice from UCD Veterinary hospital

1.3 Glossary

Invoice

An invoice details a sales transactions between two parties, indicating the products/services the seller had provided for the buyer and the cost of each item. Invoices usually include the issuer and the receiving party.

Machine Learning

The field of computing regarding the study of pattern recognition, computational learning theory and artificial intelligence. Machine Learning explores the building and usage of algorithms that can produce predictions on data.

Layout Segmentation

This is the process of identifying and separating out a document/page into its individual components. In respect to an invoice, this would consider the address, company name, item list, item, etc. as components that make up a invoice document.

Optical Character Recognition (OCR)

'OCR' is the process of taking a document, handwritten or printed and translating it into encoded machine-readable text. Will be used to extract text out of invoices.

Application Programming Interface (API)

A set of function calls which allow the access to a particular application/dataset.

Microservice

In computing-system architectures, this is a style which has large systems comprised of small, independent processes communicating via APIs. Microservices focus on a small task, for the building of a modular and scalable system.

2. General Description

2.1 Product / System Functions

Sonrasc is broken up into two main sections: front-end and back-end.

2.1.1 Front-end System

Uploading invoices

The user will interact with this part of the system. They will be able to upload pictures that they have taken on their smartphone. Once the system has processed this image of an invoice, the user will see the extract contents shown to them in a number of different views. First they will be shown it in a textview, once they confirm these items, they have the option to explore the data in a visual form.

Expenditure over time

A dashboard showing all recorded invoices over time will be shown. The user will be able to see a graphical representation of each invoice plotted over time in comparison to how much was spent. There will also be filtering options to see certain time periods, include only certain businesses and a certain threshold of money spent will be available to manipulate the graph.

Business Profiles

For every creditor who have invoices recorded for, they will have a profile in the system. This will include their information (address, contact details, relevant account numbers for this business), a list of all the invoices uploaded for that business, you will be able to interact with this invoice list to get details of a specific invoice. You will be able to see a graph of time vs money spent in respect to that business.

Searching

The user will be able to search from any screen on the application. They will be able to look for businesses, dates, invoice specific information (invoice numbers, prices, items) where they will be returned a list of relevant content to the query. I.e if you search for a business name, it will return the business profile and all the invoices that contain that name.

2.1.2 Back-end System

Layout segmentation

In order to extract information, a classifier must be used to identify what constitutes an invoice. A layout segmentation console application will be trained using invoices and used to determine the different regions in an individual invoice. It will make the image more contrasted to pick out distinct areas. It will pick out the import regions that need information extracted.

This is a multi-step process where there will be separate layout segmentation processes for addresses, item lists, contact numbers, invoice numbers etc. Work on existing image and document segmentation, such as the findings of the state-of-the-art from [Klein, Agne, and Dengel](#).

Optical Character Recognition (OCR)

An OCR process will take in an individual piece of an invoice (broken up by the layout segmentation) and will extract the text information out of the image, packaging it up in a usable format to be amalgamated with all the other pieces of that invoice to be sent for storage.

2.2 User Characteristics and Objectives

Users will be people who are part of the business environment and have a special interest in the financial matters of the business they are employed with/in ownership of. These users should be aware of what invoices are and their use to a business, with the authorization to view invoice data. Finances management should be of interest to said user.

This user should have basic knowledge of how to use a smartphone/the web and dealing with functionality of taking a picture. When taking a picture, a user should make the effort to take a steady picture of just an invoice in focus of the text. Interacting with the application, a user should understand how to read simple graphs and how to make queries for search.

Main user objectives with this is that they want be given insights on their invoices that are coming in from other businesses. A user wants to understand how much is being spent on a timely basis. Insights as to what items/services are being purchased and what kind of businesses money is being spent on are desired outputs for users. A method to remove the

need for manual input (typing) of invoices and everything was purchased, keeping everything together for filing and accountancy purposes by means of taking pictures.

As such, the requirements for a user is that this system should extract information accurately, given that the user upholds their responsibility of taking a clear, focused picture of an invoice. The system should be quick to analyse an image and extract the information and present it back to the user.

Using the application should be accessible for users, intuitive to use and understand both on mobile and desktop. Features such as search and viewing graphs should be fast and responsive to user interaction. All data stored on the application should be safe and secure since it deals with sensitive information.

2.3 Operational Scenarios

2.3.1 New Data for the system

Gerard, a co-owner, receives an invoice in the post from Kelly Grain Ltd. in Ardee, county Louth for the ordered feed which will last the animals for the next month. He reads the invoice, satisfied that the content is correct. Using his smartphone's camera, he takes a picture of the invoice, for all the supplies Kelly Grain Ltd have provided.

Invoice							Page 1	
Kelly Grain Ltd								
Peppermount								
Ardee								
VAT No: 0544474K								
Telephone: 041 6856130 / 041 6853455 Mobile: 096 2579534								
Email: kellygrain@btconnect.com Fax: 041 6856164								
VAT Reg No: 0544474K								
GERARD BRENNAN								
Peppermount								
Ardee								
Co.Louth								
VAT Reg No:								
Invoice No: 230561								
Date: 27/01/2010								
Cust Ref:								
Docket No: 424								
Account: BRENO5								
Code	Details	Quantity	Unit Price	Net Amt.	VAT%	VAT		
RBARLEY	ROLLED BARLEY	0.5000	130.00	117.00	0.00	0.00		
RWHEAT	ROLLED WHEAT	1.0000	140.00	140.00	0.00	0.00		
Title of these goods does not transfer to the customer until full payment is received by Kelly Grain Ltd.								
Full terms and conditions available on request.								
Interest 1% per month after 1 month.								
Total Net Amount							257.00	
Total Tax Amount							0.00	
Invoice Total							€257.00	

Kelly Grain Ltd's invoice segmented after proposed processing


```

{
  "Business Name": "Kelly Grain Ltd",
  "Business Address": "Kelly Grain Ltd, Pepperstown, Ardee, Co. Louth",
  ...
  "Invoice Number": "238581",
  "Date Issued": "27/01/2010",
  "Item List": [
    {
      "Product Name": "Rolled Barley",
      "Product Code": "RBARLEY",
      "Quantity": 0.9,
      "Unit Price": 130,
      "Cost": 117
      "VAT": 0
    } ...
  ],
  "Total Cost": 527
}

```

The output in JavaScript Object Notation Format (JSON) when OCR has been applied to each region of text

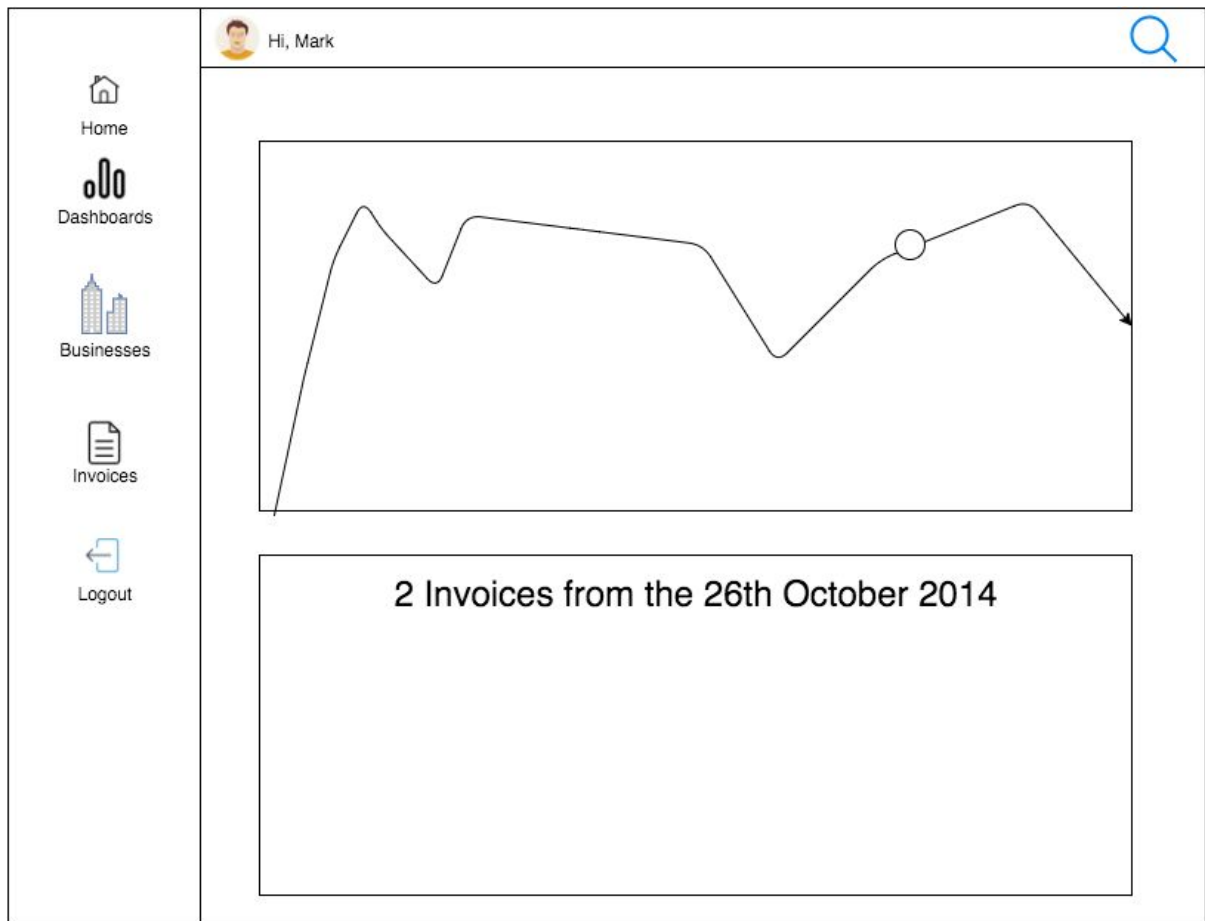
He allows the system to render the results and he is able to view all the items extracted in a confirmation screen. Gerard is content with the result, being shown the details of the business who sent the invoice, along with the items Gerard is being charged for. He confirms the items by clicking “OK” on his smartphone. He is brought back to the dashboard of the application where he can explore the newly uploaded data.

2.3.2 Desire to view the data

With a few months worth of invoices uploaded to the system Mark, a co-owner, wants to look at how much the farm has been charged in total over the last financial year from the various businesses with whom he has purchased commodities and services from. He logs onto the system where he clicks on the “overview” dashboard. Initially, a graph is displayed showing the total costs over a certain period of time. The time goes back to the earliest dated invoice in the system.

Mark scans over this graph, hovering over individual data points which tells him how many invoices are recorded for a given date/data point. He clicks on a data point where it informs him two invoices and a total of six items were recorded for the 6th October 2014.

Underneath the graph on click, individual ‘invoice cards’ appear which detail, what business (with a link to go into the business profile if Mark wished) each invoice came from, the items that were purchased and the total cost from that invoice.



A wireframe of the interface described

Wishing to see just 2014 data, hiding all of 2015 and the previous years before this on the graph, Mark does so with the filtering options underneath the graph.

2.3.3 Info from a business

Wanting to work remotely whilst traveling, Elisa needs to follow up on order status for a feed delivery from South Teffia Co-op, Elisa is out of the office running an errand and needs to ring the business. Being on the move, she has no old paper information with her. She logs onto the system where she goes into the 'Business Profiles' section. Here she is presented with a search bar where she can enter the business she is looking for, there is also a list of 'business cards', sorted by how much dealings Pepperstown Holsteins would have with that particular business. Elisa types in South Teffia in the search bar, where the business cards filter to make that the only business to show.

Clicking on the card, she is brought to a business profile that gives various details of the business, a history of the invoices in a list form and a small graph showing the expenditure over time, its address and what she was looking for - the contact number. She clicks on the most recently recorded invoice to double check a previous orders details (the contents of the invoice) before making the phone call.

2.3.4 Looking for information

Andrew is following up on a query from the accountant, who claims to have mixed up some figures while preparing the accounts for the current financial year. He asks Andrew to confirm the total spent from Progressive Genetics to see if they match what he has. Andrew, who is a rush to fulfill this information need, quickly logs onto the system.

Making use of the search functionality available in the top right hand corner, he types in "Progressive Genetics", where the results appear on screen. The results consist of the business card and the matching invoices which have "Progressive Genetics" in the contents. He clicks on the first result, bringing up the business profile, where he makes a note of the total spent as detailed by the system, going back to the accountant with the information.

2.4 Constraints

There are a number of general constraints with this project:

Reliable picture uploads and information extraction

The time that will be taken from picture upload to bring back the extracted results to the user: this must be marginal. Users should not be kept waiting for the back-end of the system to break up and invoice and perform OCR on each part.

Users also need to be shown how to take a picture correctly. Many users will try to break this by taking blurry photos, photos of things that are not invoices, and missing information by crumpled/ripped pages. Graceful and resilient error handling for these sort of errors are mandatory. There needs to be a fallback mechanism, such as correcting wrong information extracted when we hit an incorrect result.

Speed

General speed and performance of the system is pivotal. Rendering graphs and searching shouldn't be a time consuming process, the user will expect results to be delivered in a timely manner. This holds true for all front-end functionality of the application.

Security

This application deals with sensitive, business critical information. Privacy of the user and their dealings needs to be upheld and shown only to that user and whoever they wish to do so. Data must be encrypted and the user must have a way of proving that it is them trying to access this sensitive information.

Lists general constraints placed upon the design team, including speed requirements, industry protocols, hardware platforms, and so forth.

3. Functional Requirements

3.1 Uploading Invoices

Sonrasc will have functionality that will allow business users to upload a picture of an invoice. Multiple means of uploading a file can be used. A picture might be taken on the spot using a smartphone camera if it was the mobile application, or the user could upload a file via the desktop application. The file could be a PDF or JPEG file format. If the user is taking pictures on the spot, they can upload multiple images that constitute as one invoice (that may have multiple pages.)

This function is critical for the system to work. It is the direct route of inputting data into the system so it can populate itself. You must have a means of taking pictures or uploading files to the system to run as required.

Handling should be catered for the detection of no camera/a very poor image. Response time for the extraction should be maximum of five seconds. Security and resilience is essential in this part of the application which gives a means for malicious users to upload arbitrary data. Measures need to be in place to ensure only the correct type of file is uploaded.

3.2 Information Extraction

This is the process in the back-end of the system which takes a single invoice as input, and the output will be the extracted text in an appropriate data structure to be sent for storage in the database via the API. It will comprise of elements used to extract the regions on an invoice, which will pipe to a text extraction process.

In order for the system to work correctly and as intended, taking out the majority of manual inputting data for users convenience, they is also critical to the system working properly. Issues include the implementation, where a large set of invoices will be needed to be collected in order to train a classifier which can extract the information.

3.3 Graphical Representation of Invoices

Dashboards giving summarised, aggregated views of all invoices uploaded to the system. Graphs will be available for users to examine the data. This graph will plot the total costs from an individual invoice against time. Filtering options will be available to view the data according to a certain time period, money thresholds spent and the business who sent the

invoice. Users will be able to interact with the graph and click on individual data points to see what are the contents for the returned invoices for that day.

This functionality is critical to the system for it to work as intended, for the user to receive a meaningful representation of the data they fed into the system. The potential implementation issues are dealing with responsiveness, ensuring that the graph is viewable/interactable on a mobile screen as well as a desktop experience.

Graphs depend on the data being fed into the system and as such depend on uploading and information extraction.

3.4 Business profiles

For at least one invoice uploaded, a profile will be created for the respective business. This will show the user details of that business. Name, phone numbers, addresses, relevant account numbers and a list of all the invoices that have been uploaded will be shown to the user. The graphical representation of invoices will be reused here, filtered to just show details relevant to the business in question.

While not critical to the functioning of the system, provides another layer of depth with the use of the data extracted from invoices.

3.5 Invoice view

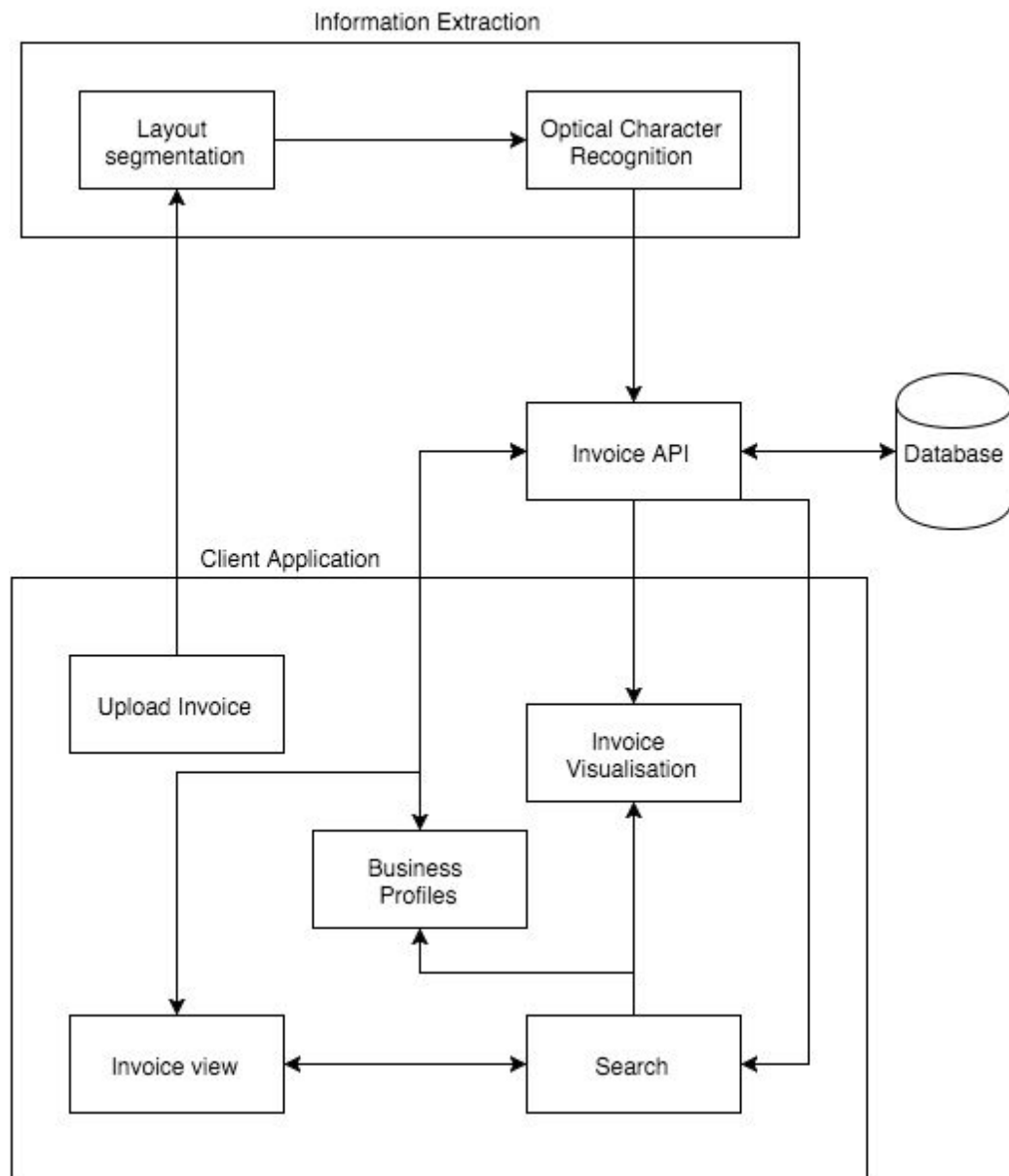
There will be an option for a user to view individual invoices previously uploaded to the system. When they select an invoice, they can view a list of the items that were purchased and the costs for each and in total. There will be links provided for the business that the invoice was issued from. Reassures the user that their invoice has been uploaded to the system.

3.6 Searching

Users will have the ability to search for queries regarding dates, items, business names and prices. The results will be a list of relevant content, relative to the search query entered. Results will be links to individual invoices which match the query contents, or businesses if the query string matches one. This piece of functionality critical to the overall system but shall help boost the human interaction experience. Issues such as choosing and

implementing appropriate models for search that understand different queries so it can return the correct answer to the user's information need.

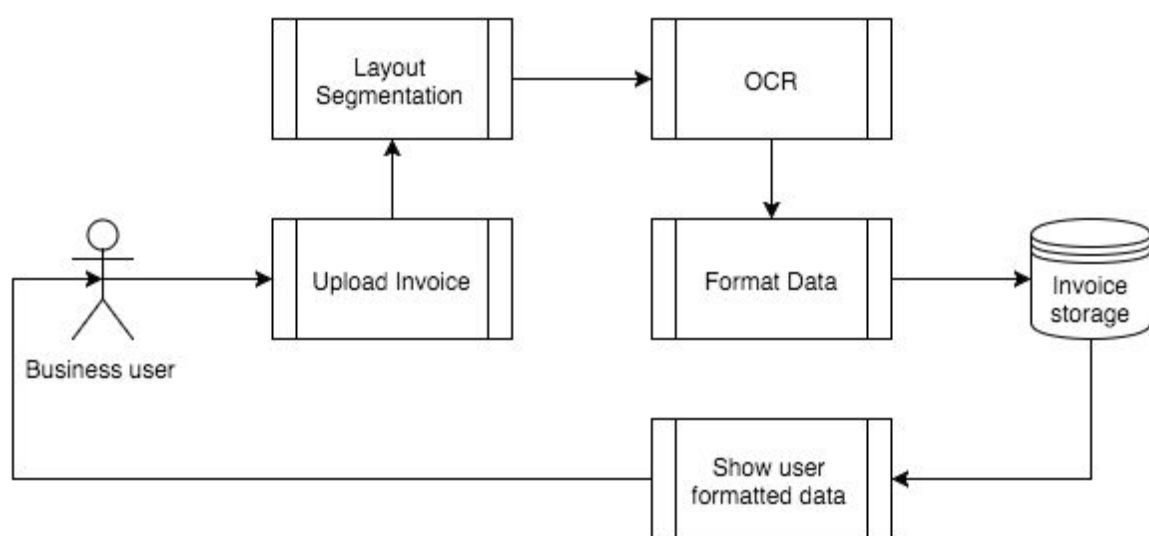
4. System Architecture



System Architecture for the frontend and backend components

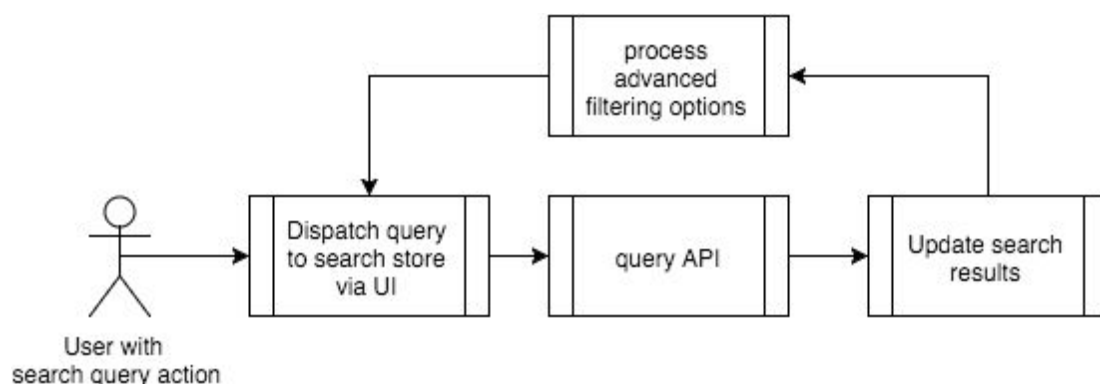
As described in the diagram, the high level overview of the proposed system architecture for the project. The functional requirements as described in section 3 communicate each other, with the main point of contact being the Application Programming Interface which will be the communication tool between the backend and the database, along with the client application and the database. It is worth noting that this will be implemented using a Microservices style architecture, where a component of the system will live standalone and communicate to the other parts using an API.

5. High-Level Design



A user uploading a new invoice to the system, showing a high level view of the backend process

One of the main design goals of this system is to provide a unidirectional data flow to keep in line with modern design patterns such as Facebook's Flux architecture.



User interacting with the searching functionality

Client-side functionality will take this sort of design pattern to make data flow easy to reason about.

As discussed in section 4, the system will utilise microservices to accomplish its goals. The main services will be as follows:

1. Backend, information extraction.
2. Storage of invoices
3. Invoice API
4. Client frontend

6. Preliminary Schedule

	N				D				J				F				M				A				M			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Backend first verison																												
<i>Collect data</i>																												
<i>Transcribe & tag data</i>																												
<i>Build initial layout segmentation</i>																												
<i>Build initial OCR</i>																												
Frontend first verison																												
<i>Graph visualisation</i>																												
Backend																												
<i>Algorithm tuning & processing</i>																												
Frontend																												
<i>Business profiles</i>																												
<i>Invoice view</i>																												

Initial schedule for project, November 2015 - May 2016

This project is an iterative process, it will be completed in stages, where basic functionality to get the minimum requirements (uploading invoices, extracting information and basic graph viewing) will be completed first to have a working version. It will then be extended for more functionality in the front-end with the aim of developing a more sophisticated back-end as the iterations continue over the course of the project.

6.1 Hardware and software requirements

As Sonrasc is being developed using microservices (using Docker), it will be deployed onto a cloud-based system. Microservices give the benefit of not being bound to a particular operating system. It can be developed and ran locally on a laptop/desktop computer and then deployed onto the cloud.

7. Appendices



A copy of the document giving permission of use of the training data for the project