**Centennial College**

**AI ENABLED CARD ORGANIZER**

**Group - 6**

Cloud Machine Learning – COMP264 Sec004

Professor Name: Sajid Hussain

April 16, 2023

Group Members: -

Daniyal Ahmed - 301152472

Kanishka Dhir - 301220757

Utkarsh Kaushik - 301209079

Soham Rajput - 301180583

Vikas Bhargav Trivedi - 301217554

**Table of Contents:**

[Research Results: 2](#_Toc37482104)

[Available options in market 3](#_Toc527142287)

[ML and DL Algorithms 3](#_Toc1698692930)

[Technology Stack 4](#_Toc459721746)

[Design Documents: 5](#_Toc1716862734)

[Functional Requirements 5](#_Toc904776319)

[Architecture of solution 5](#_Toc1368667004)

[Communications Diagram: 6](#_Toc511087527)

[List of private and public endpoints 8](#_Toc2139806551)

[User Interface Mock up 8](#_Toc1297896365)

[Future work 11](#_Toc63180705)

[Conclusion 12](#_Toc1146295865)

[Assumptions 12](#_Toc1291838982)

[References 12](#_Toc2057762579)

[Appendix 1: Project plan 13](#_Toc594250173)

[Appendix 2: Meeting register 14](#_Toc1081312252)

# Research Results:

## Available options in market

In today's world, businesses and individuals are looking for more efficient and streamlined ways of managing their contacts and information. That's where business card scanning applications come in.

Hub.cards is a mobile application that is used for scanning and digitizing business cards. It uses OCR technology to extract contact information and allows users to organize and manage their contacts. Hub.cards is free to download, but it offers in-app purchases for additional features.

CamCards is another business card scanning app that uses OCR technology to extract contact information from business cards. It also offers the ability to add notes and reminders to contacts. The app is free to download but offers premium features for a monthly or yearly subscription.

SanSan scan to Salesforce is a tool that allows users to scan business cards and automatically create new leads in their Salesforce account. It uses OCR technology to extract contact information and can be used on both desktop and mobile devices. The pricing model for SanSan is subscription-based, with different plans available based on the number of users and features needed.

HubSpot is an all-in-one marketing, sales, and service platform that includes features such as CRM, email marketing, social media management, and analytics. It uses advanced technology to automate processes and improve efficiency. HubSpot offers a freemium model, with a free version available for small businesses, and paid plans with more features for larger businesses.

Covve scan is a business card scanning app that uses advanced OCR technology to capture and store contact information. It also includes features such as automatic updates, reminders, and data analysis. Covve scan is free to download but offers in-app purchases for additional features.

Abby is an OCR software that allows users to convert scanned documents, PDF files, and images into editable formats such as Word, Excel, and HTML. It uses advanced OCR technology to accurately recognize text and improve efficiency. Abby offers different pricing models based on the specific product and features needed.

Overall, these applications offer efficient and effective ways of managing contact information and streamlining business processes. The use of OCR technology has made it possible to extract information quickly and accurately from business cards and other documents, and the different pricing models make it accessible for businesses and individuals of all sizes.

## ML and DL Algorithms

* Support Vector Machines (SVM): SVM is a supervised ML algorithm used for classification and regression analysis. It has been used in the past for business card recognition tasks.
* Convolutional Neural Networks (CNNs): CNNs are a type of DL algorithm used for image classification and object recognition tasks. They have been used in the past for reading business cards by detecting text regions on the card and extracting text from them. An example of this can be found at Chandrasekhar et al. (2016)
* Recurrent Neural Networks (RNNs): RNNs are a type of DL algorithm used for sequential data processing. They have been used in the past for recognizing text on business cards, by predicting the next character in a sequence of characters until the entire text is recognized.
* Deep Learning-based Optical Character Recognition (OCR): This is a DL-based approach that uses a combination of CNNs and RNNs to recognize text on business cards. This approach has been used in the past to accurately recognize text on business cards, even when the text is distorted or appears in different fonts. An example of this can be found at H. Saiga et al. (1993)

Overall, while different algorithms have been used in the past for reading business cards, DL-based approaches, such as CNNs and RNNs, have shown the most promising results in recent years.

## Technology Stack

1. Python: Python is a powerful and widely used programming language. It will be used for the backend development of the application. Python provides various libraries and frameworks that can be used to develop efficient and scalable serverless applications.
2. JavaScript is a popular frontend framework that will be used to create the user interface of the application. JavaScript provides various features and tools that make it easy to develop responsive and interactive web applications.
3. AWS S3 and DynamoDB: AWS S3 is a highly scalable object storage service that can be used to store and retrieve data from anywhere on the web. DynamoDB is a fully managed NoSQL database that is highly scalable and can handle large amounts of data. Both S3 and DynamoDB will be used to store and manage the leads contact datastore of the application.
4. Serverless Framework: AWS Chalice is a serverless framework that allows developers to create and deploy serverless applications easily. It supports Python and integrates with AWS services such as AWS Lambda, API Gateway, and DynamoDB. It will be used to develop and deploy the backend of the application.
5. Operating System: Linux is a free and open-source operating system that is widely used in server environments. It is highly reliable, scalable, and secure. Linux will be used as the operating system for the server hosting the application.
6. AI Services: AWS Recognition and AWS Comprehend are AI services provided by AWS that can be used to extract information from images and text respectively. These services will be used to extract contact information from images and text.

RESTful API Architecture: A RESTful API is an architectural style for creating web services that are scalable, reliable, and easy to maintain. It will be used to provide a web-based interface for accessing the application's data and functionalities.

# Design Documents:

## Functional Requirements

The following are some of the functional requirements for the project:

1. **User Registration and Authentication**: The application should allow users to create accounts and log in to access the features and functionalities of the application.
2. **Lead Data Management**: The application should allow users to create, read, update, and delete lead data. Users should be able to search and filter the leads by various criteria such as name, email, phone number, location, etc.
3. **Contact Information Extraction**: The application should be able to extract contact information such as name, email, phone number, address, etc., from images and text using AI services such as AWS Recognition and AWS Comprehend.
4. **Data Storage**: The application should store the lead data securely in AWS S3 and DynamoDB.
5. **Translating the Information**: The application should be translating the information on card if it is not in English language. The translated text should be displayed, and the user should be allowed to edit the information.
6. **API Access**: The application should provide a RESTful API to allow other applications to access the lead data and functionalities of the application.

## Architecture of Solution

**Diagram, timeline

Description automatically generated**

## Communications Diagram:

AWS Services Communications

Chart, waterfall chart

Description automatically generated

DynamoDB CommunicationsChart, box and whisker chart

Description automatically generated

## List of private and public endpoints

List of Public Endpoints:

1. /home
2. /signup
3. /signin
4. /save
5. /delete
6. /search

List of Private Endpoints:

1. /images
2. /images/ translate-text
3. /detect-text

## User Interface Mock up

**Home**

Graphical user interface

Description automatically generated

**Sign Up:**

Graphical user interface

Description automatically generated

**Create Lead:**

**Graphical user interface

Description automatically generated**

**Search Lead:**

**A picture containing table

Description automatically generated**

**Update Lead:**

**Graphical user interface

Description automatically generated**

# Future work

**Data Backup and Recovery**: The application should automatically back up the lead data regularly and allow users to restore the data in case of data loss or corruption.

**Email Campaigns**: The application should allow users to create and send email campaigns to the leads. Users should be able to track the status of the email campaigns and analyze the results.

**Reporting and Analytics**: The application should provide various reports and analytics to help users analyze and understand the lead data. Users should be able to generate reports such as lead conversion rate, lead source analysis, lead activity history, etc.

**Integration with Third-party Services**: The application should be able to integrate with third-party services such as email providers, CRM systems, marketing automation tools, etc., to provide a seamless experience for the users.

**User Roles and Permissions**: The application should allow the administrator to define user roles and permissions to control the access and actions of the users.

**API Access**: The application should provide a RESTful API to allow other applications to access the lead data and functionalities of the application.

Conclusion

Conclusion: In conclusion, our team has successfully designed and developed a full stack serverless intelligent enabled application using cloud Artificial Intelligence (AI) services to address the real-world business problem of populating the “Leads contact store” on the cloud. We have utilized OCR technology (AWS Comprehend), cloud computing services, and AI web capabilities in the fields of NLP and image recognition to intelligently detect potential contact information from business cards. Our application allows users to upload business card images and automatically detect and extract name, telephone number(s), email address, company website, and company address. Users can also update these elements before they are stored in the DynamoDB table acting as the “Leads contact” datastore. Users can only update and delete the leads that they have created and will be able to only read the other leads. We have also ensured that the application takes into consideration software design principles such as “Separation of concerns” and “single responsibility” to eliminate dependency on the cloud vendor for future portability.

Assumptions

Assumptions: We assume that all team members have access to the required technology stack and are familiar with Python and the frontend scripting language of JavaScript that supports RESTful architecture. We also assume that all team members are familiar with AWS services such as S3, DynamoDB, Boto3, Chalice, and the AI services (AWS Recognition, AWS Translate and AWS Comprehend) used in the project. Furthermore, we assume that the Stanford business cards dataset will be used for testing the application and that our team selected 10-20 business card images for testing. Finally, we assume that the project plan will be updated on a weekly basis and a simple log of all team meetings will be maintained.

References

Chandrasekhar, V., Lin, J., Morère, O., Goh, H., & Veillard, A. (2016). A practical guide to CNNs and Fisher Vectors for image instance retrieval. *Signal Processing*, *128*, 426–439. <https://doi.org/10.1016/j.sigpro.2016.05.021>

H. Saiga, Y. Nakamura, Y. Kitamura, & T. Morita. (1993). *An OCR system for business cards*. IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/abstract/document/395616>

*Stanford Mobile Visual Search Data Set: Business Cards*. (n.d.). <https://web.cs.wpi.edu/~claypool/mmsys-dataset/2011/stanford/mvs_images/business_cards.html>

# Appendix 1: Project plan

Week 1: Research and Requirements Gathering

* Research and document available AI-enabled business card organizers in the market
* Research and document the ML/DL algorithms used to understand the information in the business card
* Research and document each element of the technology stack used for the project and how it will contribute to the project
* Discuss and finalize project requirements with team members

Week 2: Design

* Developed a list of functional requirements
* Create design graphs illustrating the architecture of the solution, including all the endpoints and the AI stack layers
* Create interaction diagrams illustrating the interactions between the various components of the solution
* Create a list of private and public endpoints
* Create a user interface mockup
* Ensure the design adheres to the software design principles of Separation of Concerns and Single Responsibility

Week 3: Setup

* Set up the AWS infrastructure, including S3, DynamoDB, Chalice, and the chosen AWS AI services
* Develop the RESTful API
* Set up local host testing
* Ensure that all team members have access to the AWS infrastructure and necessary software development toolkit

Week 4-5: Development

* Develop the backend using Python
* Develop the frontend using Html and JavaScript
* Develop functionality to allow users to load an image of the business card to the system and the solution should detect the following: name, telephone number(s), email address, company website, and company address
* Develop functionality to allow users to update the elements before they are stored into a DynamoDB table which will act as the “Leads contact” datastore (“Human in the loop”)

Week 6: Testing and Refinement

* Test the application on local host
* Test the AI services and refine the ML/DL algorithms as necessary
* Refine the application as necessary to ensure all requirements are met
* Test the application on the AWS infrastructure

# Appendix 2: Meeting register

|  |  |  |  |
| --- | --- | --- | --- |
| **Date - Time** | **Who attended** | **Subjects Discussed** | **Assignments** |
| 13 March 2023 - 10:30am | All | Divided the work for Project Checkpoint -1 | Daniyal-ML/DL Algo Research and Communication Diagram  Kanishka-User Interface and List of endpoints  Vikas- Architecture Diagram and Market options Research  Soham – Technology Stack Research  Utkarsh -Functional requirements |
| 20 March 2023 – 10:30am | All | Divided the work for Project Checkpoint -2 | Daniyal and Vikas – Aws Services and Frontend  Kanishka, Soham, Utkarsh – DynamoDB and backend |
| 04 April 2023 – 11:00am | All | Shared initial work | Daniyal – Created the html layout  Vikas – Created the recognition, upload, translate api  Kanishka and Soham – Save, read items in DyanomoDB |
| 08 April 2023 – 12:00 pm | All | Frontend and backend working separately | Daniyal and Vikas – Comprehend Service and frontend.  Kanishka and Soham – Backend Api’s working |
| 10 April 2023 – 11:00 am | All except Utkarsh | Project working as a whole | Vikas, Kanishka, Soham – Working on Authentication and integrating the backend and frontend |
| 15 April 2023 – 12:00pm | All except Utkarsh | Checking and fixing issues | Vikas, Soham, Daniyal – Integrating backend and frontend Soham, Utkarsh – Debugged translation API integration with frontend  Kanishka – Working on report |