**AUTOMATED CLASSIFICATION OF FILES**

**BASED ON SAAS CLOUD**

**MUHAMMAD MUNEEB**

**AFIF SHEIKH**

**MEHREEN SULTANA**

**A project report submitted in partial fulfilment of the**

**requirements for the award of the degree of**

**Bachelor of Computer Science (Honours)**

**Department of Computer Science**

**Bahria University, Karachi Campus**

**June 2020**

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Signature : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name : Muhammad Muneeb

Reg. No. : 46009

Signature : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name : Afif Sheikh

Reg. No. : 45594

Signature : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name : Mehreen Sultana

Reg. No. : 46080

Date : June 15’ 2020

APPROVAL FOR SUBMISSION

We certify that this project report entitled **“AUTOMATED CLASSIFICATION OF FILES BASED ON SAAS CLOUD”** was prepared by **MUHAMMAD MUNEEB, AFIF SHEIKH & MEHREEN SULTANA** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Computer Science (Honours) at Bahria University.

Approved by, Dr. Ghulam Muhammad Shaikh

Signature : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Supervisor : Dr. Ghulam Muhammad Shaikh

Date : June 15’ 2020

The copyright of this report belongs to the author under the terms of the copyright Ordinance 1962 as qualified by Intellectual Property Policy of Bahria University. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2020, Muhammad Muneeb, Afif Sheikh & Mehreen Sultana. All right reserved.

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express my gratitude to my research supervisor, Dr. Ghulam Muhammad Shaikh for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, we would also like to express my gratitude to our loving parent and friends who had helped and given me encouragement.

**AUTOMATED CLASSIFICATION OF FILES**

**BASED ON SAAS CLOUD**

ABSTRACT

The issue of properly categorizing and storing files on computer, security problems and specially searching and recovering files is very paramount, especially in organizations and individuals. Therefore, we aim to resolve this problem in classifying the files in database and we also intend to aid with cloud services so that all files will be well protected and easily accessible from anyplace through internet. This project will give the ability to upload the files on cloud with a single click and by applying machine learning algorithms and searching techniques, all the files will be organised on the basis of extension and titles. And it also gives the service to download the files to local storage.

Throughout the years since cloud has been a great influence in the IT world, we decided to take these services to make file and documents categorization, simpler and more uncomplicated. The File system stores files and it is very time-consuming task to organize data manually. This application will generate automated named folders and will organize itself all files which will make easy to search and relocate the data.

We will work on SAAS model of cloud computing. It has 3 different types of service models SAAS, PAAS and IAAS. The consumer will use the application and upload the file or document it will be processed by using the files categorization techniques and then it will be converted to Dockers file then and image will be created, then it will go to a container from where it will be uploaded to the cloud.

This project will give benefits in many platforms, it will be globally available to everyone, either an organization or an individual, and all the unstructured data will be organized and recoverable quickly within just a click of a button.

TABLE OF CONTENTS

[DECLARATION ii](#_Toc377203183)

[APPROVAL FOR SUBMISSION iii](#_Toc377203184)

[ACKNOWLEDGEMENTS vi](#_Toc377203185)

[ABSTRACT vii](#_Toc377203186)

[TABLE OF CONTENTS viii](#_Toc377203187)

[LIST OF FIGURES x](#_Toc377203189)

[LIST OF SYMBOLS / ABBREVIATIONS xi](#_Toc377203190)

[LIST OF APPENDICES xii](#_Toc377203191)

**CHAPTER**

[1 INTRODUCTION 12](#_Toc377235197)

[1.1 Background 12](#_Toc377235198)

[1.2 Problem Statement 15](#_Toc377235199)

[1.3 Aims and Objectives 16](#_Toc377235200)

[1.4 Scope of Project 17](#_Toc377235201)

[2 LITERATURE REVIEW 18](#_Toc377235202)

[2.1 Introduction 18](#_Toc377235203)

[2.1.1 SAAS 18](#_Toc377235213)

[2.1.2 The Technology of Cloud Native Application 18](#_Toc377235213)

[2.1.3 Micro Services, Containers & Service Fabrics 19](#_Toc377235213)

[2.1.4 The Bad Old Days 19](#_Toc377235213)

[2.1.5 Hello VMware! 20](#_Toc377235213)

[2.1.6 VMwarts 21](#_Toc377235213)

[2.1.7 Hello Containers! 21](#_Toc377235213)

[2.1.8 The Google Way 21](#_Toc377235213)

[2.2 Cloud Computing 22](#_Toc377235204)

[2.3 Mobile Cloud Computing 23](#_Toc377235205)

[2.4 Cloud Services 24](#_Toc377235208)

[2.5 Latest Researches 25](#_Toc377235208)

[3 DESIGN AND METHODOLOGY 28](#_Toc377235209)

[3.1 Framework 28](#_Toc377235210)

[3.1.1 Flask 28](#_Toc377235213)

[3.3.1 Werkzueg 29](#_Toc377235213)

[3.3.1 Template Engine 29](#_Toc377235213)

[3.3.1 Flask over Django 29](#_Toc377235213)

[3.2 Tools & Languages 30](#_Toc377235208)

[3.3 Developing Life Cycle 30](#_Toc377235208)

[3.3.1 Product Baclog 30](#_Toc377235213)

[3.3.2 Sprint Planning 31](#_Toc377235213)

[3.3.3 Sprint Backlog 31](#_Toc377235213)

[3.3.4 Daily Scrum 31](#_Toc377235213)

[3.3.5 Sprint Review 31](#_Toc377235213)

[3.3.6 Release Planning 32](#_Toc377235213)

[3.3.7 Release Product 32](#_Toc377235213)

[3.4 Proposed Model Design 33](#_Toc377235208)

[3.4.1 Application Flowchart 34](#_Toc377235213)

[4 IMPLMENTATION 35](#_Toc377235214)

[4.1 Method to Implement 35](#_Toc377235215)

[4.1.1 Interfaces 36](#_Toc377235218)

4.1.1.1 Login Phase 36

4.1.1.2 Registration Phase 37

4.1.1.3 Display Uploaded Files 37

4.1.1.4 Account 39

[5 RESULTS AND DISCUSSIONS 40](#_Toc377235219)

[5.1 Results 40](#_Toc377235220)

[5.2 Discussion 40](#_Toc377235221)

[6 CONCLUSION 41](#_Toc377235224)

[6.1 Conclusion 41](#_Toc377235225)

[REFERENCES 42](#_Toc377235229)

[APPENDICES 44](#_Toc377235230)

LIST OF FIGURES

**FIGURE TITLE PAGE**

[Figure 1.1: Cloud Computing Services 14](#_Toc377232623)

[Figure 2.1: Cloud Models 23](#_Toc377232624)

[Figure 2.2: Mobile Cloud Computing 24](#_Toc377232625)

[Figure 2.3: Cloud Services 25](#_Toc377232626)

[Figure 3.1: Life Cycle 32](#_Toc377232623)

[Figure 3.2: Proposed Model Design 33](#_Toc377232623)

[Figure 3.3: Application Flowchart 34](#_Toc377232623)

[Figure 4.1: Login Phase 36](#_Toc377232623)

[Figure 4.2: Registration Phase 37](#_Toc377232623)

[Figure 4.3(a): Displaying Drive 38](#_Toc377232623)

[Figure 4.3(b): Displaying Drive 38](#_Toc377232623)

[Figure 4.4: Account 39](#_Toc377232623)

LIST OF SYMBOLS / ABBREVIATIONS

*SAAS* Software-as-a-Service

*PAAS* Platform-as-a-Service

*IAAS* Infrastructure-as-a-Service

*HAAS* Hardware-as-a-Service

*CIAT* Confidentiality-Integrity-Availability-Traceability

*ip* Internet Protocol

*GAE* Google App Engine

*EC* Elastic Cloud

*OS* Operating System

VM Virtual Machine

*CPU* Central Processing Unit

*RAM* Random Access Memory

*MCC* Mobile Cloud Computing

*DAS* Domain Administrating Server

*WSGI* Web Server Gateway Interface

LIST OF APPENDICES

**APPENDIX TITLE PAGE**

[A Graphs 44](#_Toc217106970)

[B Computer Programme Listing 45](#_Toc217106971)

## INTRODUCTION

### Background

Everyday more and more data is generated, from the business, individuals or communities. Data gets saved or uploaded to cloud, users move on and data is forgotten and lost. Valuable data is saved in files and document, not protected and unrecoverable because no one knows where to find it. This software will help in categorizing and classifying the unstructured data. [1]

With this software you can categorize and search your data to find the important information. This software will use searching techniques to help finding the right files more easily and quickly.

The ability to send files and get files with just a click of a button brings ultimate convenience and ease.

No doubt today most of the users are manually categorizing and ordering their files and data in their storage. Apparently, there are many business organizations where most of the data is unstructured and it’s very difficult to search for valuable information in long gone data in the storage. Previously there were no cloud storage and even today many users are using local storage to save their data, which is unprotected and unreachable in large datasets.

This software will provide all the convenience to help your data to be properly categorized and classified accordingly. The automated classification will be based on file formats and titles, all the data will be categorized properly within the folders and directories.

Also, simple user interface with a single-click classification and graphical choices for quick recognition.

#### Cloud Computing

The expression “cloud computing” refers to the access of computing resources through the Internet for purposes of data storage, aggregation, synthesis, a retrieval, together with the capacity to act on the data with computational algorithms and software packages. Cloud computing is available on-demand and provides flexible and scalable computing resources from remote locations [2].

Cloud computing is a rapidly developing and excellent promising technology. It has aroused the concern of the computer society of whole world. Cloud computing is Internet-based computing, whereby shared information, resources, and software, are provided to terminals and portable devices on-demand, like the energy grid. Cloud computing is the product of the combination of grid computing, distributed computing, parallel computing, and ubiquitous computing. It aims to build and forecast sophisticated service environment with powerful computing capabilities through an array of relatively low-cost computing entity, and using the advanced deployment models like SaaS (Software as a Service), PaaS (Platform as a Service), IaaS (Infrastructure as a Service), HaaS (Hardware as a Service) to distribute the powerful computing capacity to end-users. This paper will explore the background and service models and also presents the existing research issues and implications in cloud computing such as security, reliability, privacy. [3]

Rather than owning their own computing infrastructure or data centres, companies can rent access to anything from applications to storage from a cloud service provider.

One benefit of using cloud computing services is that firms can avoid the upfront cost and complexity of owning and maintaining their own IT infrastructure, and instead simply pay for what they use, when they use it.

Cloud computing services cover a vast range of options now, from the basics of storage, networking, and processing power through to natural language processing and artificial intelligence as well as standard office applications. Pretty much any service that doesn't physically close to the computer hardware that you are using can now be delivered via the cloud. [4]

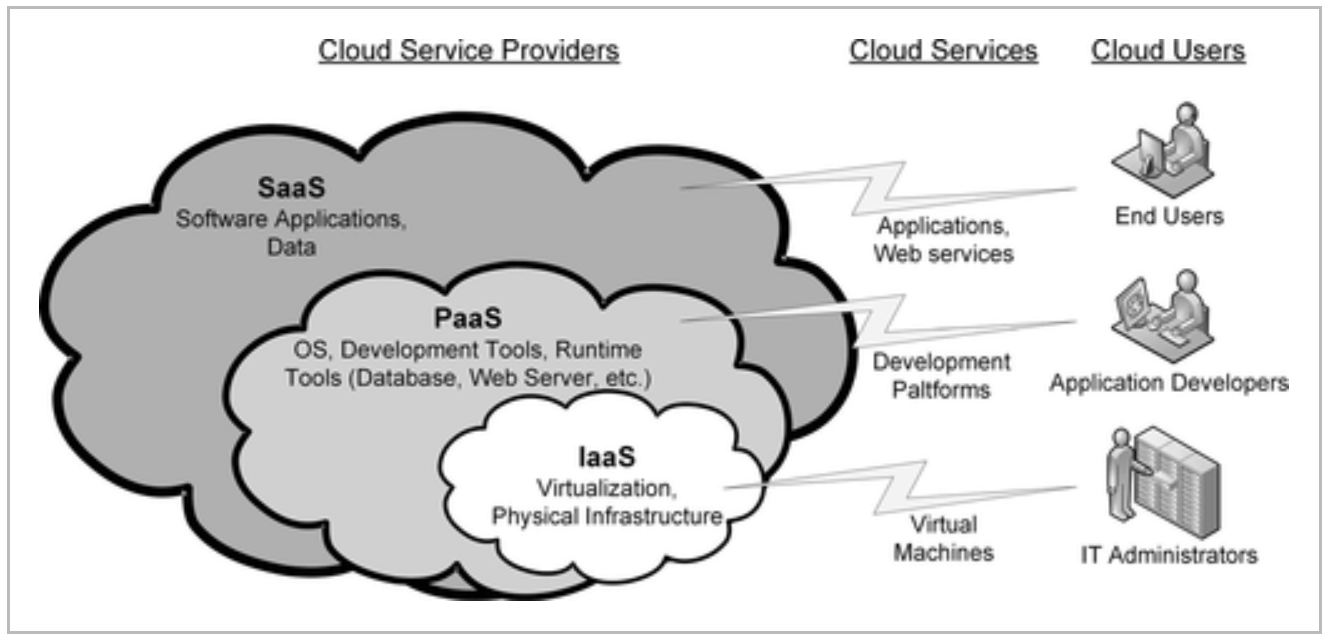


Figure 1.1: Cloud Computing Services

### Problem Statements

#### Data Security Concern

Data security in the cloud refers to data confidentiality, integrity, availability and traceability (CIAT), and these requirements pose major problems for cloud computing. Multiple serious security threats like virus attack and hacking of the client’s site are also the biggest cloud computing data security issues.

#### Reliability on new technology

It is a fact of human nature that we trust on the things present in front of our eyes. Normally entrepreneurs feel hesitation in letting out the organizational information to any unknown service provider. [5]

#### Cloud Infrastructure

Cloud infrastructure includes an [abstraction layer](https://whatis.techtarget.com/definition/hardware-abstraction-layer-HAL) that virtualizes resources and logically presents them to users through [application program interfaces](https://searchapparchitecture.techtarget.com/definition/application-program-interface-API) and API-enabled [command-line](https://searchwindowsserver.techtarget.com/definition/command-line-interface-CLI) or [graphical interfaces](https://searchwindevelopment.techtarget.com/definition/GUI). It was problem to establish a cloud environment with large setup.

#### Cost Concern

As Cloud services are paid to use for 1-year service.

#### Mapping Connection of Application

User require access to application which expose on some ip: port to be mapped for external user.

### Aims and Objectives

The aims and objectives of this proposed methodology are shown as following:

1. An application will have a User Interface so that users can upload their files and these files will be organized by using Machine Learning algorithms, on the cloud.
2. By using cloud services, the user can access their data from anywhere and can upload or download any file from anywhere.
3. Main objective of this project is to make easier for the user to classify and organize the files with titles so that any file cannot be misplaced.

### Scope of Project

Following are the scopes of this project:

1. Files organization is an issue in every small or large storage. Users must organize their files manually in their storage. For example, when you download a file in any extension or Exe File.
2. If a person downloads multiple extension files like Pdf, Word, Excel and Images after some time it goes bunch of files in your storage.
3. File will save with Title in automated created folder.
4. This project is developing an application based on cloud to self-organize the title based on their format.
5. Another side is that application can be down due to any failure in application or host to save application it will be manage by new technology with concern to cloud infrastructure.

## LITERATURE REVIEW

### Introduction

#### SAAS

Software-as-a-service (SaaS) is one of three principal components of cloud computing, with the other two being platform-as-a-service (PaaS) and infrastructure-as-a-service (IaaS). SaaS runs on top of PaaS that in turn runs on top of IaaS. SaaS has not only its business model but also its unique development processes and computing infrastructure. At the system level, unlike traditional software that runs on operation systems, SaaS is usually deployed on a PaaS system such as GAE1, EC22, orAzure3, or specialized SaaS infrastructure. To manage the software data, conventional systems often use relational databases that support concurrent processing and give readers priority over writers. Data schemas are usually normalized. For reliability, availability, and security, conventional systems use security kernels and redundancy and rollback mechanisms.

#### The Technology of Cloud Native Application

The first wave of cloud computing relied on Infrastructure as a Service that replaced on premise infrastructure with virtual machines running in cloud data centres. While this was suitable for small applications such as basic web services, it was still very difficult to engineer scalability and manage security at the same time. Additionally, the lack of cloud-based data services, event services, and debugging facilities meant that the programmer had to cobble together solutions from different open source components.

By 2010, platform services that provided abstractions for data management and event handling began to appear in the commercial cloud offerings. In the area of big data analytics, companies like Google and Microsoft were already using internally developed, highly parallel distributed file systems and map-reduce tools. After Google published a research article about their experience, Yahoo! released an open source product called Hadoop that allowed anybody to deploy a distributed file system and analytics tools that anybody could deploy to virtual machines in any cloud. Hadoop may be considered the vanguard of cloud-native applications. But managing scale reliably was still a daunting task. Integrating Hadoop as a reliable, scalable platform service was a challenge for both vendors and expert users. [6]

#### Micro Services, Containers and Service Fabrics

By 2013, the first major design pattern for cloud-native applications began to emerge. It was clear that to achieve scale and reliability, it was essential to decompose applications into very basic components, which we now refer to as micro services. [6]

#### The Bad Old Days

Applications run businesses. If applications break, businesses suffer and sometimes go away. These statements get truer every day!

Most applications run on servers. And in the past, we could only run one application per server. The open-systems world of Windows and Linux just didn’t have the technologies to safely and securely run multiple applications on the same server. So, the story usually went something like this… Every time the business needed a new application, IT would go out and buy a new server. And most of the time nobody knew the performance requirements of the new application! This meant IT had to make guesses when choosing the model and size of servers to buy. As a result, IT did the only thing it could do - it bought big fast servers with lots of resiliency. After all, the last thing anyone wanted - including the business - was under-powered servers. Under-powered servers might be unable to execute transactions, which might result in lost customers and lost revenue. So, IT usually bought bigger servers than were actually needed. This resulted in huge numbers of servers operating as low as 5-10% of their potential capacity. A tragic waste of company capital and resources! [7]

#### Hello VMware!

Amid all of this, VMware, Inc. gave the world a gift - the virtual machine (VM). And almost overnight the world changed into a much better place! We finally had a technology that would let us safely and securely run multiple business applications on a single server.

This was a game changer! IT no longer needed to procure a brand new oversized server every time the business asked for a new application. More often than not they could run new apps on existing servers that were sitting around with spare capacity. All of a sudden, we could squeeze massive amounts of value out of existing corporate assets, such as servers, resulting in a lot more bang for the company’s buck. [7]

#### VMwarts

But… and there’s always a but!As great as VMs are, they’re not perfect!

The fact that every VM requires its own dedicated OS is a major flaw. Every OS consumes CPU, RAM and storage that could otherwise be used to power more applications. Every OS needs patching and monitoring. And in some cases, every OS requires a license. All of this is a waste of op-ex and cap-ex.

The VM model has other challenges too. VMs are slow to boot and portability isn’t great - migrating and moving VM workloads between hypervisors and cloud platforms is harder than it needs to be. [8]

#### Hello Containers!

For a long time, the big web-scale players like Google have been using container technologies to address these shortcomings of the VM model.

In the container model the container is roughly analogous to the VM. The major difference through, is that every container does not require a full-blown OS. In fact, all containers on a single host share a single OS. This frees up huge amounts of system resources such as CPU, RAM, and storage. It also reduces potential licensing costs and reduces the overhead of OS patching and other maintenance. This results in savings on the cap-ex and op-ex fronts. Containers are also fast to start and ultra-portable. Moving container workloads from your laptop, to the cloud, and then to VMs or bare metal in your centre is a breeze. [7]

#### The Google Way (Official Way)

From Gmail to YouTube to Search, everything at Google runs in containers. Containerization allows our development teams to move fast, deploy software efficiently, and operate at an unprecedented scale. Each week, we start over two billion containers. We’ve learned a lot about running containerized workloads in production over the past decade, and we’ve [shared this knowledge](https://research.google.com/pubs/pub44843.html) with the community along the way: from the early days of contributing [groups to the Linux kernel](https://www.kernel.org/doc/Documentation/cgroup-v1/cgroups.txt), to taking designs from our internal tools and open sourcing them as the [Kubernetes](https://cloud.google.com/kubernetes/) project. We’ve packaged this expertise into [Google Cloud Platform](https://cloud.google.com/) so that developers and businesses of any size can easily tap the latest in container innovation. [9]

### Cloud Computing

### Cloud computing is computing in which big clusters of remote servers are networked to sanction centralized data storage and online access to computer services or resources.

### The cloud setup is deployed in four different deployment models as follows:

1. Public Cloud: Services in public clouds are available to the all customer without pay.
2. Private cloud: Service in private clouds are available to only some customer which are sitting behind the firewall. These services are paid services.
3. Community cloud: A community cloud in computing is a collaborative effort in which infrastructure is shared between several organizations from a specific community with common concerns (security, compliance, jurisdiction, etc.), whether managed internally or by a third-party and hosted internally or externally.
4. Hybrid Cloud: Services in hybrid cloud are the combination of public and private in which some services are freely available, and some are paid.

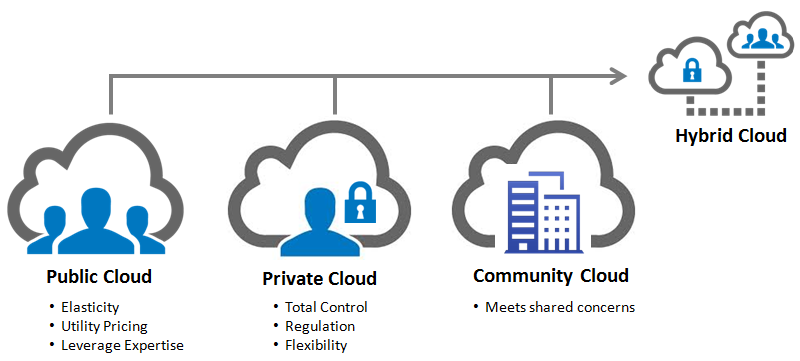


Figure 2.1: Cloud Models

### Mobile Cloud Computing

Another latest technology is cloud computing which allows access to the stored data from anywhere at any time, and can be utilized in different organizations or by individuals to improve efficiency and increase performance and decrease the cost and complexity. Moreover, integrating the mobile devices with cloud computing to utilize the unlimited service provided by the cloud through the mobile device results in what is known as Mobile Cloud Computing. The Cloud Computing relies on a set of network-connected resources shared to maximize their utilization resulting in reduced management and capital costs. Mobile Cloud Computing (MCC) is set to advantage many sectors. As an example, Mobile commerce is a business model for commercial activities using a mobile phone. Now a day, maximum financial transactions are done through the internet. Mobile commerce can enhance the finance, advertising, shopping because of its mobility and availability. Different secure applications (e.g., mobile payment, transaction, ticketing) can be designed for mobile devices using the power of cloud computing. [10]

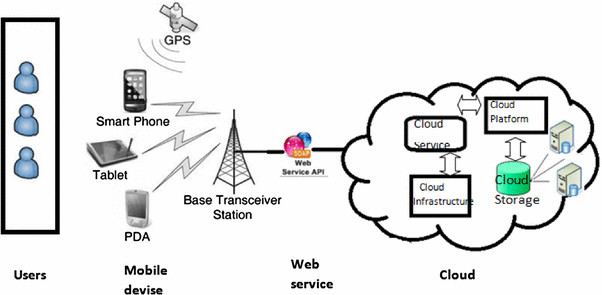


Figure 2.2: Mobile Cloud Computing

### Cloud Services

Cloud applications are provided as one of the three main service models as:

1. Software as a Service (SaaS): SaaS provides software to the user in the form of services which are hosted in a cloud.
2. Platform as a Service (PaaS): PaaS enables programming environments to access and utilize additional application building blocks. Such programming environments have a visible impact on the application architecture, such as constraints on which services the application can request from an OS.
3. Infrastructure as a Service (IaaS): IaaS enables customer to access the storage, other resources which are important for the user applications.

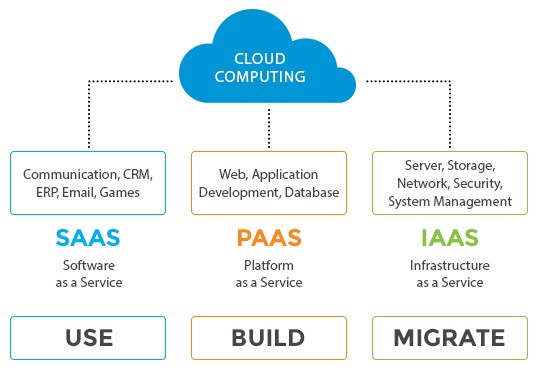


Figure 2.3: Cloud Services

### Latest Researches

#### Storage management system with file aggregation and space reclamation within aggregated files.

By

*David Maxwell Cannon, Howard Newton Martin*

A data storage subsystem employs managed files, each comprising one or an aggregation of multiple constituent user files, to reduce file management overhead costs. After receiving user files from a client station, the subsystem creates a contiguous managed file by aggregating selected ones of the received user files according to certain predetermined criteria. Managed file creation and use are transparent to the client stations. Eventually, unused or "deleted-file space" arises as individual user files are deleted from their respective managed files. "Reclamation" is triggered when the amount of deleted-file space in a prescribed storage area (e.g. device, volume, managed file) satisfies certain criteria, e.g. poor data storage efficiency. Reclamation is performed to regain wasted space between managed files. [11]

#### File Assignment Control for a Web System of Contents Categorization

By

*Masaki Kohana, Hiroki Sakaji, Akio Kobayashi, Shusuke Okamoto*

In this Web system, the file transfer time becomes a problem when a user sends larger files. In this paper, we propose a way to resolve the issue of longer file transfer time by controlling the file assignment. We assign the large files to the Web browser process, and we assign the smaller files to the calculation machines over the network. [12]

#### System and method of monitoring and controlling application files

By

*Kester, Harold M., Hegli, Ronald B., Dimm, John Ross, Anderson, Mark Richard.*

A system and method for updating a system that controls applications requested for execution on a workstation. A workstation management module is configured to detect requested execution of an application. A workstation application server receives data associated with the application from the workstation. The application server module can determine one or more categories to associate with the application by referencing an application inventory database or requesting the category from an application database factory. The application database factory can receive applications from multiple application server modules. The application database factory determines whether the application was previously categorized and provides the category to the application server module; which forwards a hash/policy table to the workstation management module. Upon receipt of the hash/policy table, the workstation management module applies the policy to control access to the requested application on the workstation. [13]

#### Automated file acquisition, identification, extraction and transformation

By

*Bruhn, David M. (Moorpark, CA, US), Capitano, Douglas L. (Berthoud, CO, US)*

Managing large amounts of third party client data may require sorting through files for patterns and extracting data to create a customized user interface for the third party client. One example method of operation may include examining file names for data files stored in a database, parsing specified names and specified dates from the file names, categorizing the data files according to the specified names and specified dates, tagging the data files, and transforming content of the data files into a customized data table format associated with known client requirements. [14]

#### Network management system having virtual catalog overview of files distributive stored across network domain

By

*Thomas Pisello, David Crossmier, Paul Ashton*

A network management system includes a domain administrating server (DAS) that stores a virtual catalog representing an overview of all files distributively stored across a network domain currently or in the past. The current and historical file information is used for assisting in auditing or locating files located anywhere in the domain. The current file information is used for assisting in transferring files across the domain. The domain administrating server (DAS) also includes a rule-base driven artificial administrator for monitoring and reacting to domain-wide alert reports and for detecting problematic trends in domain-wide performance based on information collected from the network domain. [15]

## DESIGN AND METHODOLOGY

### Framework

#### Flask

Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

Flask is part of the categories of the micro-framework. Micro-framework are normally framework with little to no dependencies to external libraries. This has pros and cons. Pros would be that the framework is light, there are little dependency to update and watch for security bugs, cons is that some time you will have to do more work by yourself or increase yourself the list of dependencies by adding plugins. In the case of Flask, its dependencies are:

1. Werkzeug a WSGI utility library
2. jinja2 which is its template engine

WSGI is basically a protocol defined so that Python application can communicate with a web-server and thus be used as web-application outside of CGI. [16]

#### Werkzeug

Werkzeug is a comprehensive WSGI web application library. It began as a simple collection of various utilities for WSGI applications and has become one of the most advanced WSGI utility libraries.

Flask wraps Werkzeug, using it to handle the details of WSGI while providing more structure and patterns for defining powerful applications. [17]

#### Template Engines

Using templates, you are able to set a basic layout for your pages and mention which element will change. This way you can define your header once and keep it consistent over all the pages of your website, and if you need to change your header, you will only have to update it in one place.

Using a template engine will save you a lot of time when creating your application but also when updating and maintaining it. [16]

#### Flask over Django

Flask is considered more Pythonic than the Django web framework because in common situations the equivalent Flask web application is more explicit. Flask is also easy to get started with as a beginner because there is little boilerplate code for getting a simple app up and running.

Flask was also written several years after Django and therefore learned from the Python community's reactions as the framework evolved. Jökull Sólberg wrote a great piece articulating to this effect in his experience switching between Flask and Django. [18]

### Tools and Languages

* Python 3.7.1
* Flask 1.1.1
* Werkzeug 0.15.5
* Jinja2
* Conda 4.5.12
* SQLite DB
* Sublime Text
* PyCharm
* Windows Command Prompt
* FireFox Web Browser, Google Chrome
* Laptops
* Mobile Phones

### Developing Life Cycle

#### Product Backlog

Product Backlog is simply a list of all things that needs to be done within the project. It replaces the traditional requirements specification artefacts. These items can have a technical nature or can be user-centric e.g. in the form of user stories.

#### Sprint Planning

Sprint planning is an event in the [Scrum](https://www.agilealliance.org/glossary/scrum/) framework where the team determines the product backlog items they will work on during that [sprint](https://www.agilealliance.org/glossary/iteration/) and discusses their initial plan for completing those product backlog items.

#### Sprint Backlog

The sprint backlog is a list of tasks identified by the Scrum team to be completed during the [Scrum](https://www.mountaingoatsoftware.com/agile/scrum) sprint. The sprint backlog is commonly maintained as a spreadsheet.

#### Daily Scrum

In [Scrum](https://www.mountaingoatsoftware.com/agile/scrum), on each day of a sprint, the team holds a daily scrum meeting called the daily scrum. Meetings are typically held in the same location and at the same time each day.

#### Sprint Review

In [Scrum](https://www.mountaingoatsoftware.com/agile/scrum), each sprint is required to deliver a potentially shippable product increment. This means that at the end of each sprint, the team has produced a coded, tested and usable piece of software. So at the end of each sprint, a sprint review meeting is held. During this meeting, the Scrum team shows what they accomplished during the sprint.

#### Release Planning

A very high-level plan for multiple Sprints is created during the Release planning. It is a guideline that reflects expectations about which features will be implemented and when they are completed.

#### Release Product

A product release is the process of launching a new product for a specific market or user base.

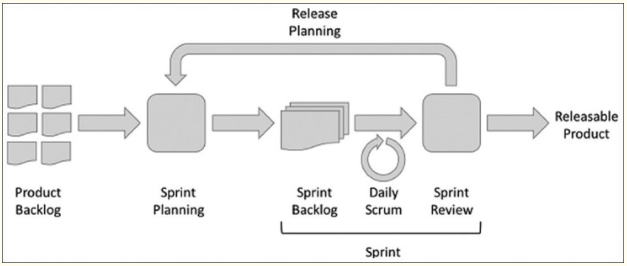


Figure 3.1: Life Cycle

### Proposed Model Design

![A screenshot of a cell phone

Description generated with high confidence](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4RD0RXhpZgAATU0AKgAAAAgABAE7AAIAAAAOAAAISodpAAQAAAABAAAIWJydAAEAAAAcAAAQ0OocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAE11bmVlYiBTaGVpa2gAAAWQAwACAAAAFAAAEKaQBAACAAAAFAAAELqSkQACAAAAAzE5AACSkgACAAAAAzE5AADqHAAHAAAIDAAACJoAAAAAHOoAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAyMDE5OjA3OjExIDIzOjA4OjQyADIwMTk6MDc6MTEgMjM6MDg6NDIAAABNAHUAbgBlAGUAYgAgAFMAaABlAGkAawBoAAAA/+ELIGh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8APD94cGFja2V0IGJlZ2luPSfvu78nIGlkPSdXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQnPz4NCjx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iPjxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iLz48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOnhtcD0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLyI+PHhtcDpDcmVhdGVEYXRlPjIwMTktMDctMTFUMjM6MDg6NDIuMTkwPC94bXA6Q3JlYXRlRGF0ZT48L3JkZjpEZXNjcmlwdGlvbj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyI+PGRjOmNyZWF0b3I+PHJkZjpTZXEgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOmxpPk11bmVlYiBTaGVpa2g8L3JkZjpsaT48L3JkZjpTZXE+DQoJCQk8L2RjOmNyZWF0b3I+PC9yZGY6RGVzY3JpcHRpb24+PC9yZGY6UkRGPjwveDp4bXBtZXRhPg0KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICA8P3hwYWNrZXQgZW5kPSd3Jz8+/9sAQwAHBQUGBQQHBgUGCAcHCAoRCwoJCQoVDxAMERgVGhkYFRgXGx4nIRsdJR0XGCIuIiUoKSssKxogLzMvKjInKisq/9sAQwEHCAgKCQoUCwsUKhwYHCoqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioq/8AAEQgBOgHYAwEiAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A+kaKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACmHr/AJ4p9MPX/PFACf5+lH9P0/8Ar0fpj9P/AK9H9P0/+vQAfp/T/wCvR+n9P/r0f5+n/wBesqbxT4ftrk29xrumQzqcGJ7yMMp9ME5zQBq/5+lFY3iXxZo/hLw7LrWtXiQ2cY+VlO4yE9FQD7xNePWt/wDE74zyG50q5bwZ4UkJEUy58+dPUEYLZ9iq+57gHtl/ruk6VIE1PVLKzcjhZ7hIyR7AkU6w1fTdVDHTNQtbwLw32adZNvtweK8r079mbwRbRH+0ZNS1KZjlpZbjZk/RQP1pmofs0+E3ZJ9Av9U0a8iO6KeGfftfseeeD6EfWgD2P/8AVx/KivCj4p+IXwdvI18ds3ijwuziL+1Yv9dACcAtnn8Gzns3r7TpGrWOvaRbanpNwl1Z3Sb4pYzww9PbB4I7EYoAuf5+tPXpTP1z+v8A9anr09fegBaKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAph6/wCeKfTD1/zxQAn6f0/+vXLeP/H+k/Dzw8dS1UmSaRvLtbSM/PPJjO0eg9WPA+pAPU/p/T/69eFeG4B8Vvj1quu6kDJovhVvs2n2+co8oJAc+vIZ/rs7A5AG2vgT4gfFgLqPj/Wrjw9os/zxaNZDbJsPQMD0OD/FuPqB0HQRfs2/DuO3Eb2d9K4GDK9424/lgZ/CvV/88Uf54/lQB4lb/s0aNa+JbGYa1fXOgW8pmk0q5O7c+OAGGBjPX5c44zzmva4oo4IUihRY40UKiIMBQOgAp3+eP5Uf54/lQAf/AKuP5Uf54o/zxR/n60AQ3lnb6hZTWd7Ck9vOhjlicZV1IwRj0rxDwiZvhB8ZH8F3ErP4b8Qkz6UzHiGU/wABJ9xt9/kPc17r/n614n+0zEtn4Z8Pa/F8t1YasixsOOCrN/OMUAe2dff+v/1qevT196gtpTPaQynrIisfxGfyqdelAC0UUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABTD1/wA8U+uG+LnjSXwL8O77VLMr9vci3tN3RZH43Y74GT+FAGV8QfjJpXhO8Og6TbT654hlGxbGz58okfxsM898AE+uOtUf2fvDWs+HfCOpt4i06Swur6/NwFlxvZSo7dRznrWh8Hvhza+EfDkWrX6G58Q6pGJ7y8m+aRS/zeWCeR2z6nr7ekf/AKv/AK1AB/8Aq4o//Vx/IUf/AKuP5Uf/AKuP5UAH/wCrij/PFH+eKP8AP1oAP1/rR+v9f/rUdffP6/8A1qOvvn9f/rUAH6/1/wDrV4l+1HKT8P8AS7fypDG2prLJMqErGqo68kdMlxj1r23r7/1/+tUV1a299aS2t7DHcW8yFJIpVDK6nggg9qAM7wxrWmeIPDdnf6JeR3to8ahZIz3AAII6gj0PNbK14FpVq3wZ+OVtodjI/wDwi/igZihlbi3m6DB9jgeu1hnoK99WgBaKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACise61PU/7cm0/TdPtJ1htop3luLxojl2kXaAIm6eXnOe/Til+0+I/wDoFaX/AODOT/4xQBr0VkfafEf/AECtL/8ABnJ/8Yo+0+I/+gVpf/gzk/8AjFAGvRWR9p8R/wDQK0v/AMGcn/xij7T4j/6BWl/+DOT/AOMUAa9FZH2nxH/0CtL/APBnJ/8AGKPtPiP/AKBWl/8Agzk/+MUAa9FZH2nxH/0CtL/8Gcn/AMYo+0+I/wDoFaX/AODOT/4xQBr0VkfafEf/AECtL/8ABnJ/8Yo+0+I/+gVpf/gzk/8AjFAGvRWR9p8R/wDQK0v/AMGcn/xitCze7e3zqEEME2T8kExlXH+8VX+VAE9eJftQ2k03w9066RN0NpqaPNz0BRgP1Ne21i+KvDdj4u8M32h6qpNveRGMsv3oz1Vl9wcH8KANCyu4b+wt7y1ffBcRrJGwHVWGR+hqb/8AVx/KvB/B/j+++El0ngf4nRTRWUDlNO1pIy0bxdQDjqPcZIzgjjNeyWHifQdUtRc6drNhcwtwJIrlGB9utAGp/wDq4/lR/niqn9rad/0ELX/v+v8AjR/a2m/9BC1/7/r/AI0AW6P1/r/9aqn9rab/ANBC1/7/AK/40f2tpv8A0ELX/v8Arz+tAFvr75/X/wCtR198/r/9aqn9rad/0ELU/wDbZef1o/tbTf8AoIWv/f5ef1oAt/rn9aKqf2vpv/QQtf8Av+v+Ncf4x+Mfg/wfayedqcWoX4O2OwsnEsjt2Bxwo57/AIZ6UAcT8dP+Jl8SPh1pNn896L9pyvom+Pn/AMcb8q90Xp/jXjXwz8K634l8Z3HxM8eWzW13Mhj0nT5Bj7LF0DEdQccDPXczY5GPZV6f40ALRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRXJz3mrS+PZrSB9VNjbrbMUtFtPJXcW3eYZf3hBAGdnOOnNAGrbf8jpqX/YPtP/AEZcVr15TY+PtThlSWOxhuJLqKGTzJ5YoyVFrauId7un8dxI275iC33TnjsvDfiWXWNX1OyuTbB7RsotuwcBN7qCXVmBJ2ZwdjDkFe9AHSUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUw9afTD1oAoavomma/YPZa1YW9/bOMGKeMOPwz0Pv1rzO+/Zr8AXVyZbaLUbFT/wAsre7JX8N4Y/rXrX+eKP8AP1oA8c/4Zi8D9rrWP/Apef8Axyj/AIZi8D/8/Wsf+BS//EV7H198/r/9ajr75/X/AOtQB45/wzF4H/5+dY9v9KX/AOIo/wCGYvA/a61j2/0pf/iK9j6++f1/+tR/n60AeOf8MxeB/wDn61j/AMCl/wDiKP8AhmLwP/z9ax/4FL/8RXsdH+ef50AeOf8ADMXgf/n51j/wKX/4ius8K/B7wT4PnW50rRklvFHF1eMZnHuN3Cn/AHQK7f8A/Xz/ADo//Xz/ADoAP8809aZ/nmnrQAtFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAFJLsT65cWDwqRbQQ3Cuecl2lXp2x5fX3q2qIjMVVVLHLEDqfesq2/5HTUv+wfaf8Aoy4rXoAKKKKACiiigAooooAKKKKACiiigAooooAKaQc06igBm00YNPooAZg/X+tGD3Gf60+igBm0+mf60gIbO0hsHBwe/vUlcF8LLTxHZr4lHiea2mM2tTTQfZ5GYR7sbk5UYAOMfU0Ad1g0YP8A+un0UAMwf8/zowfSn0UAMwacowKWigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAyLb/AJHTUv8AsH2n/oy4rXrz/QNY8R3fxu8S6ZfWlmmlWdjb+XcRo+9wS7RDJbGfnlzx/AMY7+gUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUVzviHWrvTNUsooJIVhlxuQqGlcl1HyqWXcME52ksODtPQ5Wn+Pbt9JgutT0uGIyW27dHcHDTiBJTGFKcHLlcZJ+RuOKAO3rE8M/8xf/ALCc39KxX8ezx3Fkz2NstrMMT5ujvtmMtum1/kAVlE7sy5P+rPI5plv4sltHhNjoA8vUJllJa9CFneOF2x5gA3fvcBAcny24HYA7iiuMTxbqF94d1DUbO3jSSM24hiQ+aVMmzcpyFG4biCOgI5PWql94z1bR9fFpcWM08aWfmOsrRx/O08SKSwJVQFdj94546c0Ad9RVfT7l7zTLW6mh8iSaFJGi3h/LJAJXcODjOMirFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUViavB9t8QabZyzXMcDwzyMtvcyQlmUxgZKMCcbjx70AbdFZH/CNWP/AD31T/wbXX/xyj/hGrH/AJ76p/4Nrr/45QBr0Vkf8I1Y/wDPfVP/AAbXX/xyj/hGrH/nvqn/AINrr/45QBr0Vkf8I1Y/899U/wDBtdf/AByj/hGrH/nvqn/g2uv/AI5QBr0Vkf8ACNWP/PfVP/Btdf8Axyj/AIRqx/576p/4Nrr/AOOUAa9FZH/CNWP/AD31T/wbXX/xyj/hGrH/AJ76p/4Nrr/45QBr0xZo2meJZFMiAFkDDKg5wSO2cH8jWX/wjVj/AM99U/8ABtdf/HK5nUfg/wCHtU8aQeJbu71g3VvCkUaLqcoA2szZ37vMH3ugYDjpzQBxN9rGrPrWqf8AE31FQmo3Uaql7KqqqzuqgANgAAAVD/a2q/8AQY1T/wAD5v8A4qq08Yi1TVI1LFU1O8UFmLHAuJOpPJPuaSvqaNCk6cW4rZdEfF18TXVWSU3u+r7lr+1tV/6DGqf+B83/AMVR/a2q/wDQY1T/AMD5v/iqq0Vr9Xo/yL7kY/WsR/O/vZa/tbVf+gxqn/gfN/8AFUf2tqv/AEGNU/8AA+b/AOKqrRR9Xo/yL7kH1rEfzv72WBqOpCVpRqupCRlCs/26XJAzgZ3dBk/mad/a2q/9BjVP/A+b/wCKqrRR9Xo/yL7kH1rEfzv72Wv7W1X/AKDGqf8AgfN/8VR/a2q/9BjVP/A+b/4qqtFH1ej/ACL7kH1rEfzv72Wv7W1X/oMap/4Hzf8AxVH9rar/ANBjVP8AwPm/+KqrRR9Xo/yL7kH1rEfzv72dL4N1XUn8aaZDNqd9NFM8ivHNdSSKwETsOGJHUA169Xi3gv8A5HvR/wDrpL/6Ikr2mvn8fGMK1oqx9Rlk5Tw95u7uwooorhPSCiiigAooooAKKbJIkMTyysERFLMx6ADqa5sfEbwiygrrtsQeQRu5/Smk3sJyS3Omormv+Fi+Ev8AoOW3/j3+FH/CxfCX/Qctv/Hv8KfLLsTzx7nS0VzX/CxfCX/Qctv/AB7/AAo/4WL4S/6Dlt/49/hRyy7Bzx7nS1m6NfzX/wBv8/b/AKPeyQJtGPlXGM+/NZn/AAsXwl/0HLb/AMe/wrI0Dx54Yt/7T8/WII/M1CWRNwb5lOMEcdKOWXYOePc7yiua/wCFi+Ev+g5bf+Pf4Uf8LF8Jf9By2/8AHv8ACjll2Dnj3Olormv+Fi+Ev+g5bf8Aj3+FH/CxfCX/AEHLb/x7/Cjll2Dnj3Olormv+Fi+Ev8AoOW3/j3+FX9I8UaLr08sOj6jDdyRKHdEJyoJwDzScWt0NSi9EzWooopFBRRRQAUUUUAFFFFABRRRQAUUUUAFZF5/yN2lf9etz/OKtesi8/5G7Sv+vW5/nFQBr0UUUAFFFFABRRRQAUUUUAFFFFABRRRQB4Bd/wDIa1f/ALCl5/6USVHUl3/yGtX/AOwpef8ApRJUdfX0P4UfRHweI/jT9X+YUVham80WrB4zcyj92FgQSqD83JDL8p68hh25IBqkur6q0kqPHcokhUo/kHMXzMGX7hxwF7PyevPA6qTsxqjKSujqqK5eO6vZZLS5mW6aeRIPMttk0QVsjeQR8pGSchh264Iq9qEmof2sk1tFMbezKh1UkCXd9/5f4tqkEY75FCqpq9hOi07XNqiuVi1i/aOBnmlEczKJJBbf6sktwnHzcAHvjr0OKYl3rS3Mcz283zoDI3lMQg/dAsFx1wXO3HUYpe2j0Rf1eXVo62iuZuNT1JbqFbX7RJEdqMxtiu4MWG/G08j5e4Htg8RJf6rFZFkFyW8pmjZrclpZBFFtVuOAWMgPT7vUc0e2j2EsPLudXRWVYy6g2of6VvMUhuODFtEeyULHg/7SknnrjIrVrWMuZGMo8rsbXgv/AJHvR/8ArpL/AOiJK9prxbwX/wAj3o//AF0l/wDREle0185mP8f5H1eVf7svVhRRRXnnqhRRRQAUUUUAUdb/AOQBqH/XrJ/6Ca8Csf8AkHW3/XJf5Cvfdb/5AGof9esn/oJrwKx/5B1t/wBcl/kK9nK95/I+fzraHz/QsUUUV7h86FFNd1jjZ5GCIoJZmOAB6mmpcRSsBHIjlkDjac5U9D9KVwsySiionuYYw5klVdn3snGOn+I/OjYNyWiiimAUUUUAFdZ8Lv8AkcNQ/wCvBP8A0Ya5Ous+F3/I4ah/14J/6MNcGYf7u/l+Z6WV/wC9R+f5Hq9FFFfMn2AUUUUAFFFFABRRRQAUUUUAFFFFABWRef8AI3aV/wBetz/OKtesi8/5G7Sv+vW5/nFQBr0UUUAFFFFABRRRQAUUUUAFFFFABRRRQB4Bd/8AIa1f/sKXn/pRJUdSXf8AyGtX/wCwpef+lElR19fQ/hR9EfB4j+NP1f5hRRRWxiFUL3UXt7jyYY4WKxGaR55vKRFzgc4Pv27Vfqtc2EF1IkknmLJGCFeKRkODjIypGRwKmV7aFQ5U/eMs6zbrp13NHaQvHYRJNGI5AVJZc/KQMDHIyM1p6fe/b7V5vL8vbPNDjdnPlyMmfx25/GoTolidmI3VVVUKLKwVwpyoYZ+bk9+verdvbRWkRjt02I0jyEZJ+Z2LMefUsTURU09djScqbj7q1JaKKK1MQooooA0vDMNzP4x0lLK6+ySmWTEvlh8fuJOxr1b+y9e/6GP/AMkU/wAa8x8F/wDI96P/ANdJf/REle0181mP8f5H1uVf7svVmJ/Zevf9DH/5Ip/jR/Zevf8AQx/+SKf41t0V556pif2Xr3/Qx/8Akin+NH9l69/0Mf8A5Ip/jW3RQBif2Xr3/Qx/+SKf40f2Xr3/AEMf/kin+NbdFAHmq+EfG1hqOvajqXjhrjSplkkj04WiuCu08bm5j+icc159Y/8AIOtv+uS/yFe+63/yANQ/69ZP/QTXgVj/AMg62/65L/IV7OV7z+R8/nW0Pn+hYooor3D50p6nay3toLeJ1RXdfNY/3AckAd84A+hNYk+kXVoEeXy7q3hYKIgrnzI8sQpVVPC7hjAP3BXT0VlKmpO5rCrKCt0OTh0LUpdMRSUSZ4flmklbzIsw7fL6E43fMefXuBVi48O3Eksm0QMrpJGuWIMe6OIZHH96Nun97PrXSUVPsY2sX9Yne5z83h55J3x5IhV90MYYgR/OjcADjhW6evuaW30a7ttRt5V8pkjd/mL5CRmR2VVXbkEKyjg44wQcCt+in7KN7k+3nawUUUVsYhXWfC7/AJHDUP8ArwT/ANGGuTrrPhd/yOGof9eCf+jDXBmH+7v5fmellf8AvUfn+R6vRRRXzJ9gFFFFABRRRQAUUUUAFFFFABRRRQAVkXn/ACN2lf8AXrc/zirXrIvP+Ru0r/r1uf5xUAa9FFFABRRRQAUUUUAVr7UrHTIVl1K8t7ON22K9xKsas2CcAk9cAnHsao/8JZ4c/wCg/pf/AIGx/wCNGs/8hbw//wBhB/8A0lnrXoAyP+Es8Of9B/S//A2P/Gj/AISzw5/0H9L/APA2P/GteigDI/4Szw5/0H9L/wDA2P8Axo/4Szw5/wBB/S//AANj/wAa16KAPn2aaO41PU5oJFlik1K8dHRgVZTcSEEEdRSVJd/8hrV/+wpef+lElR19fQ/hR9EfB4j+NP1f5hRRRWxiFFFFABRRRQAUUUUAFFFFAG14L/5HvR/+ukv/AKIkr2mvFvBf/I96P/10l/8AREle0181mP8AH+R9blX+7L1YUUUV556oUUUUAFFFFAFPWFZ9Cv1RSzNbSAADJJ2mvn2zlKWMCvBdBljUEG2k4OP92vo6iurD4mWHb5Ve5xYvBwxSSk7WPnfz/wDpjc/+A0n/AMTR5/8A0xuf/AaT/wCJr6Iors/tSp/Kjh/sal/Mz538/wD6Y3P/AIDSf/E0ef8A9Mbn/wABpP8A4mvoiij+1Kn8qD+xqX8zPnfz/wDpjc/+A0n/AMTR5/8A0xuf/AaT/wCJr6Ioo/tSp/Kg/sal/Mz538//AKY3P/gNJ/8AE0ef/wBMbn/wGk/+Jr6Ioo/tSp/Kg/sal/Mz538//pjc/wDgNJ/8TR5//TG5/wDAaT/4mvoiij+1Kn8qD+xqX8zPnfz/APpjc/8AgNJ/8TXY/CsO/ivUZfJmWMWSLueJkGd545Ar1eisa2OnWg4NI3w+W08PUVSMm7BRRRXnnqBRRRQAUUUUAFFFFABUBvIBqC2Rf/SGiMwTaeUBAJz06kVPWLq2kahcapFqGkX8FpOttJbN59uZVwxUhhhl5BXoeDntQBatdf0m8hikt9RtyJrcXSK0gVvKKht5U4IGGB5HGaaPEWjF9o1Wz/1H2nPnrt8rcV35zjGQRn2rnz8O7aPTTZwXEZi84SBZ4C25RYCzCMVZSeAHyCD2GOtRR+Ar157WS81eOURTRTODAzH93cecqhmck9SpZiT0PtQB2Et9aQwtLNdQxxqpZneQAAA4JJ9M8VmfbNOv/EUc1vqdq76fFJFNEsgLAyLG4PXptGc+9Y48EXq3MVwup27NZu72iNanbzN5o8z5/m9OMcgN2xVe48DXLXR2arYwyTbmVEsyqgNapby7UEnA+RGXnjJBzkGgDrpdX02HPnahax7WVDvnUYZs7R16nBwO+DSHWLD+110tbqN71lZjCjAsgAB+YD7vBGM9a5G8+GwubOOIX6l1nuJHBSWNJVmVVYMI5VYkBcfewRkEc5GnpPg+XS9btroXsT2tobowxeQfNPnurtukLnOCuBxnGMk4zQBo2fifTb+R1tftbBX8syPYzRxlvMEeBIyBW+Y9ie56CrOn61p2qRwNZXcUjXFulykW4CTy3AKsU6gcjqKxrfwRBbWAhjvrppftkdyzSXErx/LdLPtWJnKLnbtyBxkn1FUvDXw9Xw9qdtcG9+1JbqhXeZlPmC3WAsEEvljKr1KE4OM8A0AdpRRRQBkaz/yFvD//AGEH/wDSWetesjWf+Qt4f/7CD/8ApLPWvQAUUUUAFFFFAHld38Ntek1O/mgl05ori8nuE3zyKwWSVnAIEZ5G7HWo/wDhWniP+/pf/gTJ/wDGq9YorsjjcRFJKX4I4JZdhZScnHV+b/zPJ/8AhWniP+/pf/gTJ/8AGqP+FaeI/wC/pf8A4Eyf/Gq9Yop/X8T/ADfgv8if7Mwn8n4v/M8n/wCFaeI/7+l/+BMn/wAao/4Vp4j/AL+l/wDgTJ/8ar1iij6/if5vwX+Qf2ZhP5Pxf+Z5P/wrTxH/AH9L/wDAmT/41R/wrTxH/f0v/wACZP8A41XrFFH1/E/zfgv8g/szCfyfi/8AM8n/AOFaeI/7+l/+BMn/AMao/wCFaeI/7+l/+BMn/wAar1iij6/if5vwX+Qf2ZhP5Pxf+Z5P/wAK08R/39L/APAmT/41R/wrTxH/AH9L/wDAmT/41XrFFH1/E/zfgv8AIP7Mwn8n4v8AzPO/DXgPWNL8TWWoX8tj5NsXYiGV3ZiY2QDBQD+LPXtXolFFc1SrOrLmm7s7KVGFGPJTVkFFFFZmoUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVkXn/I3aV/163P84q16yLz/AJG7Sv8Ar1uf5xUAa9FFFABRRRQAUUVzvivW73SJdMisHSM3czo7tp814QBGzcRxMG6jr0FAFvWf+Qt4f/7CD/8ApLPWvXnup+Mza6vai7tpL1rVz5ax20kTNO0doqny/mdVxeyZUhmHTBI53LHxhJd6lY21xpNzYpdrgSXaPFl8uNqhkAP+rzhirYYHbQB01FFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABWRef8AI3aV/wBetz/OKtesi8/5G7Sv+vW5/nFQBr0UUUAFFFFABTHgikljkkiRpIiTGzKCUJGDg9uOKfRQBlalZ6QbmCK+023nk1GYw7jApJby95LE89LdOeuUT0GHw+HdGt7qC5t9Ls4prddsMiQqCg56HHufzPqai1n/AJC3h/8A7CD/APpLPWvQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUV57dfFZbfULu2j0K5lFrcy2/mCdBvMblCcehKmrhTnUdoK5nUqwpK83Y9CrIvP+Ru0r/r1uf5xVyH/C2v+pduf/AmOqU/xNeXWrO9GgThbeKWNlNymTvKYP8A45+tbfVa/wDKzD67hv50ep0V5z/wtr/qXbn/AMCY6P8AhbX/AFLtz/4Ex0fVa/8AKw+u4b+dHo1Fec/8La/6l25/8CY6P+Ftf9S7c/8AgTHR9Vr/AMrD67hv50ejUV5z/wALa/6l25/8CY6P+Ftf9S7c/wDgTHR9Vr/ysPruG/nRu+KPEWk6X4m8MWGoX0cF3c35MMLZ3SAwSxjHr8zqPxFdTXjOveLLDxB4i8PaxeeG52n0O4kmhDXEZB3xlcexDCNgf9iug/4W1/1Ltz/4Ex0fVa/8rD67hv50ejUV5z/wtr/qXbn/AMCY60NA+Iy63rttpr6PPam4LBZWmRgCqFug9lNTLD1oq8ouxUcVQnLljNNnbUUUVgdIUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVla/LcGG0sbOdraXULkW/noAWjUI0jlc9GKxsAecEg44rVqhq+nvqFogtphb3dvKs1vMV3BHHqMjKlSykZHDHBB5oApp4N8OxhGXSLYzxj5LtlzcA/3vOP7zdz97dn3qfQppzDdWd5N9omsLgwGY9ZF2q6E8D5tjqDjgkE8ZxVA+ItVN4NMHh+QakYjLk3cZtgu4qG8z7+On/LPPPStTSNOOnWsnnSLNdXEpnuZlTaJHOBnHYBQqj2UdetAF+iiigAooooAK8Au/+Q1q/wD2FLz/ANKJK9/rwC7/AOQ1q/8A2FLz/wBKJK9XK/4r9Dxc5/gx9f0ZHRRRX0B8uFFFFABRRRQAUUUUAFFFFABW14L/AOR70f8A66S/+iJKxa2vBf8AyPej/wDXSX/0RJXLi/4EvQ7MD/vMPU9pooor5Q+2CiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAI5/O+zv9l8szY+TzCduffFZm7xH/AM89L/7+Sf4Vr0UAcfu1/wD4Tn/V6b5n9m/89JMY836da2d3iP8A556X/wB/JP8ACqF9qUGn+PVM6XL7tM4+z2ks3/LXvsU4/Gr/APwktj/zw1T/AMFN1/8AG6ANODzvs6favLE2Pn8snbn2zUlcx/wlksepGKe1C28k3lwu4eJsbYjlg4GD+9HBxwrHqADNYeL7e8jeSSyvIUGNqrbSTSclh8yRq237vf39KAOhorI/4SWx/wCeGqf+Cm6/+N0f8JLY/wDPDVP/AAU3X/xugDXrwC7/AOQ1q/8A2FLz/wBKJK9o/wCElsf+eGqf+Cm6/wDjdeKTSrNqepyoGCvqV4wDoUYA3Eh5UgEH2PIr1cr/AIr9Dxc5/gx9f0YlFFFfQHy4UVipes+pSpPqX2eRLkRpaBUO9eMcEbjuyTkHA/A1l23iW9SC3TylmkaNGZ5JEUOfKhbALMoGTIemcenPGLrRW5vGhOWx11Fc1Nr96RCV8iNJZ1XdgnYonEbZ5759sdPepNV1Ga2vL3y9QMckMSNb2mExO53fLyNxyQBwaPbRtcPYSvY6Giuci8SXBuLgzW0cdtA7eYxkXdGiyBCxAYnodxJUAAEc5BpsHiG4a6kjnWG143g3DEKv7uNhH7Md59fung0e2gH1eZ0tFY2j6zcaleTxy26xom7H7xNy4crggMTzjOSBjp71s1cZKSujKcHB2YVteC/+R70f/rpL/wCiJKxa2vBf/I96P/10l/8ARElYYv8AgS9DqwP+8w9T2miiivlD7YKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKhvLuGwsZ7u6fZDBG0kjHsoGTU1YUxOueIRbL/AMeGlSLJOwP+tuMbkj+iAhz7lPRhQBSXS77b/wAJKYm/tgneLZsZFtgZtfrgbuv+s7leK6Szu4L+yhu7OVZYJkDxuvRgelTVhwf8SPxA9u7t9h1SRpIMjIhuOrpnsH5cf7Qk5+ZRQBuUUUUAFFFFABXgF3/yGtX/AOwpef8ApRJXv9eAXf8AyGtX/wCwpef+lElerlf8V+h4uc/wY+v6Mjooor6A+XEwN2ccjjNBAOMgHByM9qWikAhAYYYZHoaWiimAmACSByep9aCAeoz35paKQCAAEkDk9feloopgFbXgv/ke9H/66S/+iJKxa2vBf/I96P8A9dJf/RElcuL/AIEvQ7MD/vMPU9pooor5Q+2CiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAIL37UbKYaf5QuSpERmzsDdicckDrjv0461S3WPhXw6vnyOLe2ADyEbndmbljjqzMxJ9ya1KgurK2vRGLuFZhE+9VcZAbBGcd+GPWgCrJr+mRXBha6BkAY4VGb7rBDggcncwXHUngcg0ky2HiXR5oYbgyQOdq3EB5R1IZXRiCNysAQeQCPao4/C+kQxJHFbMiR/cCzyDZ84fj5uPmUN7kZq9Y2FtpllHaWMKw28edka9Bkkn9SaACwF2unwLqTRNdqgEzQ52M3cgHoD1x26ZPWrFFFABRRRQAV4Bd/8hrV/wDsKXn/AKUSV7/XztqGqWEOvaxHNfW0brqt4CrzKCP9Ik7Zr1MsaVV37HjZwm6Mbd/0ZNRVL+2NM/6CNp/3/X/Gj+2NM/6CNp/3/X/Gvf5o9z5jkl2LtFUv7Y0z/oI2n/f9f8aP7Y0z/oI2n/f9f8aOaPcOSXYu0VS/tjTP+gjaf9/1/wAaP7Y0z/oI2n/f9f8AGjmj3Dkl2LtFUv7Y0z/oI2n/AH/X/Gj+2NM/6CNp/wB/1/xo5o9w5Jdi7RVL+2NM/wCgjaf9/wBf8aP7Y0z/AKCNp/3/AF/xo5o9w5Jdi7W14L/5HvR/+ukv/oiSuY/tjTP+gjaf9/1/xrf8CX9ndeP9IS1u4JnDykrHIGIHkSc8Vy4uUXQlr0OzAxksTC66nuNFFFfLH2gUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB//9k=)

Figure 3.2: Proposed Model Design

This application will have a User Interface so that users can upload their files and these files will be organized by using Machine Learning algorithms, on the cloud.

#### Application Flowchart

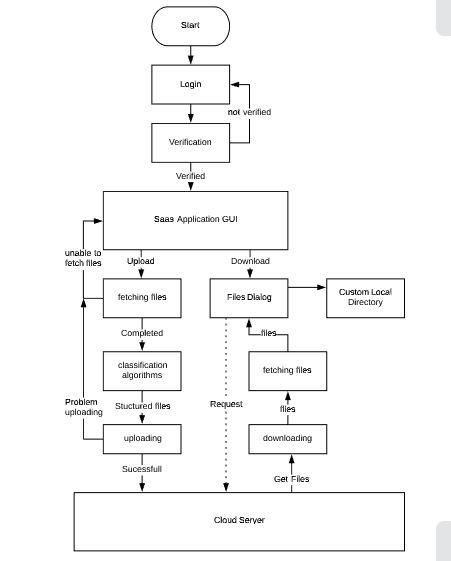


Figure 3.3: Application Flowchart

# IMPLMENTATION

### Method to Implement

A cloud-based application will be designed to provide great user experience and give user to access from anywhere. The development of the App required teamwork among pathologists and information technology professionals. The application will be developed over the course of four sprints. It was critical to allow web application to interface with the AI algorithm designed in python. Flask web framework was used to allow the application interface with the AI algorithm. The user credentials were encrypted using hash codes to eliminate the risk of electronic protected health information (ePHI) exposure. The intent was not to store any data on client site to mitigate privacy risk. The User data is stored encrypted in the database and the files are stored on the server memory. We will complete the development of the algorithm and the basic UI to test implementation of the algorithm. The application will be launched on a virtual cloud as soon as it is completed with all the user privileges and the final algorithm is completed. Currently algorithm is fully functional with classifying the files based on the file type and an additional feature is added which is user can also upload multiple files at one time.

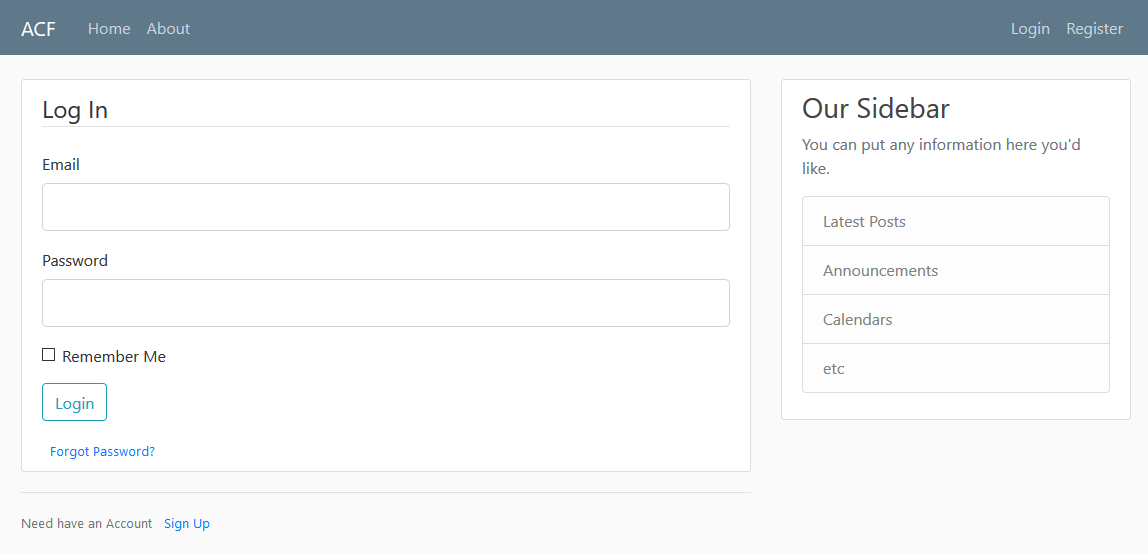
#### Interfaces

##### Login Phase

Login phase has following steps

1. User will provide his/her credentials in this phase.

2. If user is verified, then the home screen will open.



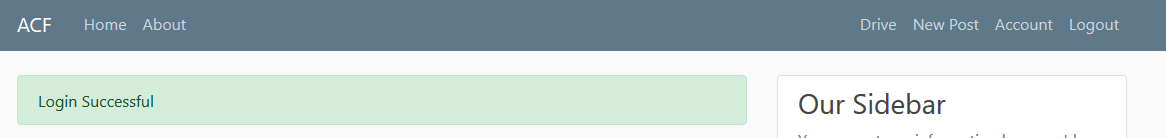


Figure 4.1: Login Phase

##### Registration Phase

Registration Phase has following steps.

1. User must provide his credentials to register an account.
2. The details will be stored on the cloud.

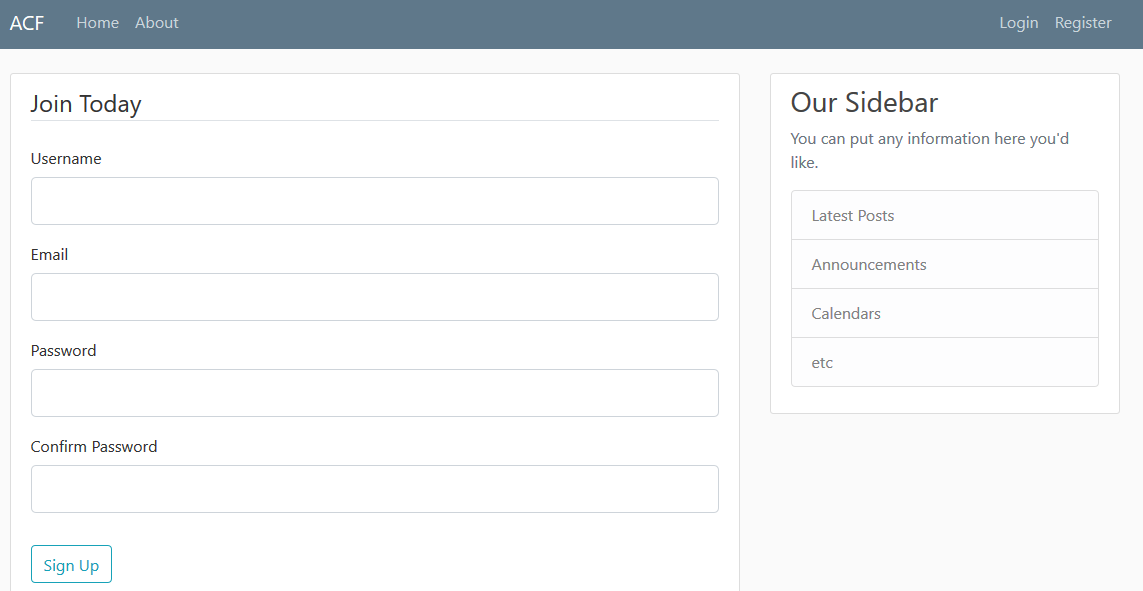


Figure 4.2: Registration Phase

##### Display Files Uploaded

This is called Drive location where all the files are uploaded shown.   
All the files are separated according to the type of files.

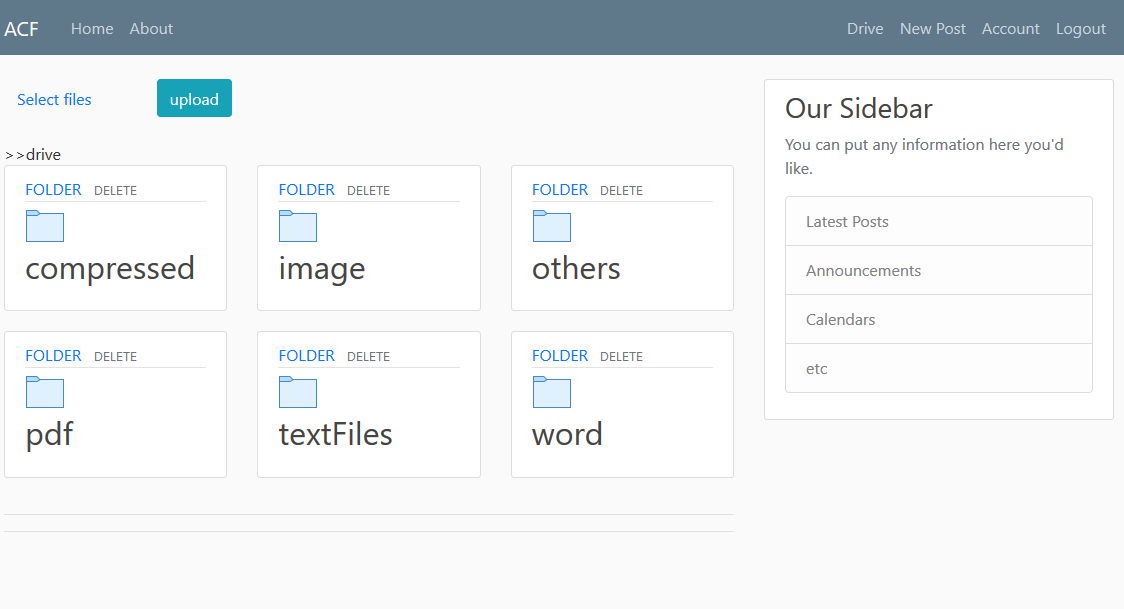


Figure 4.3(a): Displaying Drive

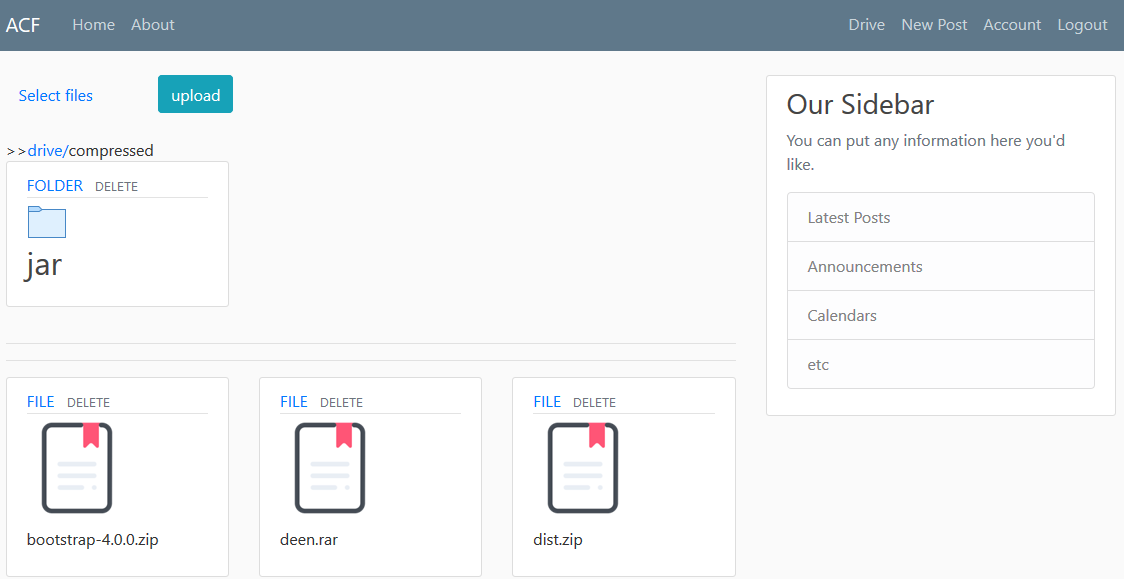


Figure 4.3(b): Displaying Drive

##### Account

In Account section user can view his/her information and can update the information.

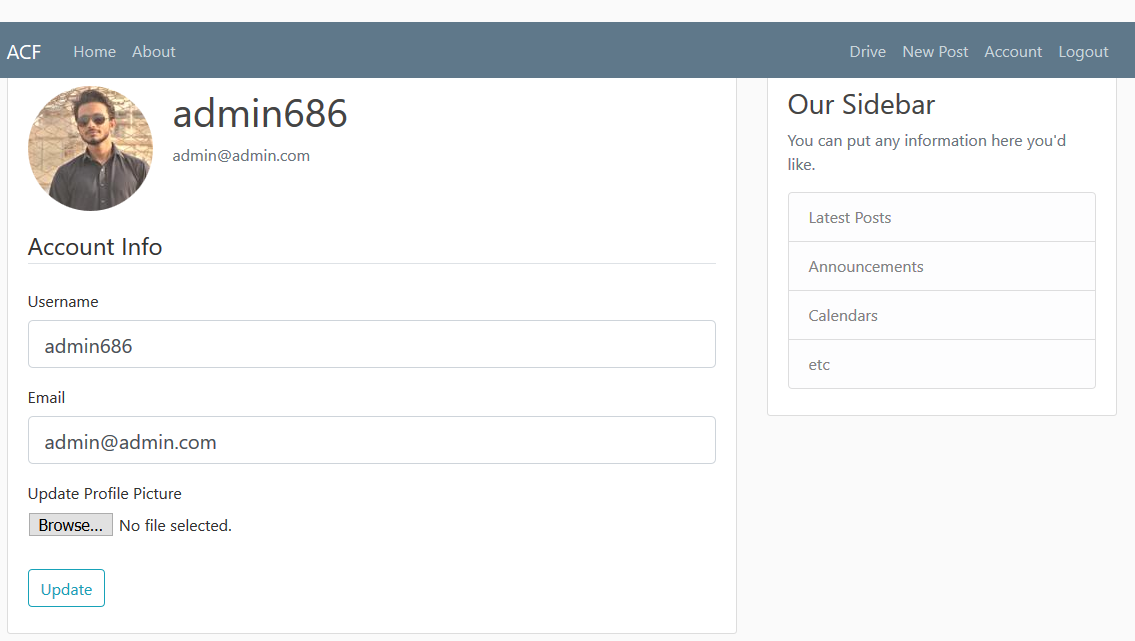


Figure 4.4: Account

## RESULTS AND DISCUSSIONS

### Results

### Discussion

## CONCLUSION

### Conclusion

Main objective of this project is to make easier for the user to classify and organize the files with titles so that any file cannot be misplaced.

Application will be stimulated on local host and will give environment of real server with help Kubernetes ease to manage application with horizontal scalability, portability, load balancing, Intelligent Scheduling and health check.

## REFERENCES

1. Athens Cyber Park Paggaious & 1 Aggistis st. Athens, “Document Classification” (2018).
2. P. Mell and T. Grence, “The NIST definition of cloud computing”, Special Publication 800–145 (2011).
3. M. Rajendra Parsad, “International Journal of Cloud Computing and Services

Science (IJ-CLOSER)”, Vol.2, No.2, April 2013, pp. 134~140.

1. [Steve Ranger](https://www.zdnet.com/meet-the-team/uk/steve-ranger/), “Cloud Computing”, vol. 34, pp.8427-445, December 13, 2018.
2. [Kgs Venkatesan](https://www.researchgate.net/profile/Kgs_Venkatesan), “Issue in Cloud Computing”, May 29, 2019.
3. [Dennis Gannon](https://www.computer.org/csdl/search/default?type=author&givenName=Dennis&surname=Gannon), “Cloud Native Applications”, vol. 4, pp. 16-21, October 2017.
4. Abhay Goswami, “Why we need Docker?” Feb 21, 2019. Available online at: <https://dev.to/abhay676/why-we-need-docker-4gdh>.
5. Adari Girish Kumar, “The emergence of Containers: Diving into Docker”, May 1, 2017. Available online at: <https://medium.com/@adari.girishkumar/the-emergence-of-containers-diving-into-docker-part-1>.
6. Michelle Moore, “Guide to Container Security”, Oct 22, 2019. Available online at: <https://www.tripwire.com/state-of-security/devops/guide-container-security/>.
7. Yang X, Pan T, Shen J (2010) “On 3G mobile e-commerce platform based on cloud computing”, 3rd IEEE International Conference on Ubi-Media Computing.
8. Kohana M., Sakaji H., Kobayashi A., Okamoto S. (2019) “File Assignment Control for a Web System of Contents Categorization”, vol 11610, June 21, 2019.
9. Kester, Harold M., Hegli, Ronald B., Dimm, John Ross, Anderson, Mark Richard “System and method of monitoring and controlling application files”, Sept 14, 2010.

[13] Bruhn, David M., Capitano, Douglas L., “Automated file acquisition, identification, extraction and transformation”, Oct 26, 2016.

[14] Thomas Pisello, David Crossmier, Paul Ashton, “Network Management System having Virtual Catalog Overview of files distributive stored across Network Domain ”, Nov 11, 2016.

[15] David Maxwell Cannon, Howard Newton Martin, “Storage Management System with File Aggregation and Space reclamation within aggregated files”, Oct 29, 2003.

[16] Parwiz, “Flask Web Development”, May 21, 2020. Available online at: <https://codeloop.org/flask-web-development-full-course-for-beginners/>.

[17] Armin Ronacher, “Werkzueg”, 2015. Available online at: <https://palletsprojects.com/p/werkzeug/>.

[18] Matt Makai, “Flask Web Development”, 2020. Available online at: <https://www.fullstackpython.com/flask.html>.

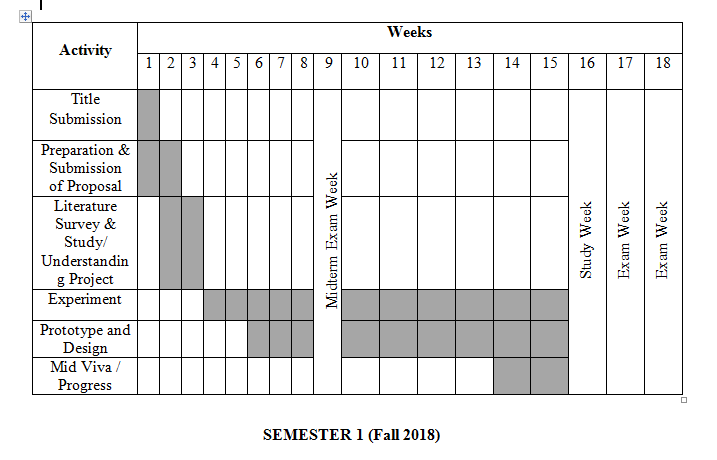
[19]

[20]

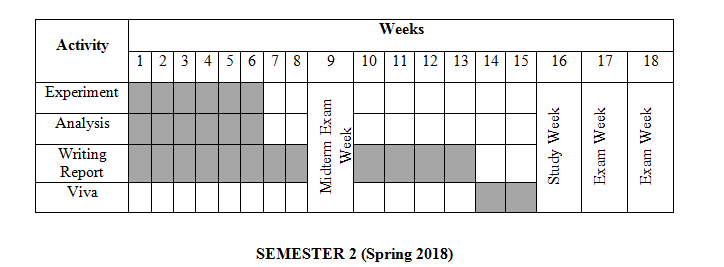
[21]

## APPENDICES

APPENDIX A: Graphs



**FALL 2019**



**SPRING 2020**

APPENDIX B: Computer Programme Listing (CODE)