User Requirements Specification

Group 5

**Student Names:**Nicolas Casuso

**Class:**I3-CB-CMK

**Student numbers:** 5265614

Table of Contents/Nicolas

[**Requirements 2**](#_mk5hcrbu2klo)

[**Introduction/Nicolas 3**](#_gbc17pm47lsd)

[**Product Vision 4**](#_70v2nvkosfvj)

[**User Personas 5**](#_2hjn8r91i6w8)

[**User Story Mapping: 5**](#_ntsi59ogl83i)

[1. Backbone (High-level project goals): 5](#_342j3xvhtq5v)

[2. User Activities (Main functionalities users will interact with) 5](#_9je58dvif1vb)

[3. User Tasks (Steps to complete each activity) 6](#_kkabxc9esozs)

[4. Technical Activities (Infrastructure and Development Tasks) 7](#_yshdmg198v65)

# Requirements

• Cover page/Nicolas

• Table of contents/Nicolas

• Introduction/Nicolas

• Product vision Chris

• User personas Chris

• User story mapping/Nicolas

• Product Roadmap

• Sprint’s goal

# Introduction/Nicolas

This project aims to develop a cloud-based storage solution, similar to Dropbox, to address two main problems: limited access to locally stored files and high upfront costs for additional storage space. Traditionally, files stored on a local device can only be accessed from that device, or through carrying an external hard drive, which can be inconvenient. Additionally, purchasing storage often requires paying for more space than needed, leading to wasted resources.

Our solution will allow users to store files both in the cloud using AWS S3 and on-premises at Fontys’ Infralab. The system will be accessible through a web interface, and we will use tools like Ansible and Terraform to deploy the infrastructure. The goal is to create a fully functional application where users can share files, collaborate on documents, and ensure files are scanned for malware before uploading.

By the end of this project, we aim to have a cloud storage solution that is easy to use, scalable, and fully deployable with automation tools.

# Product Vision

Our shared vision for our product is a simple-to-use application with extensive usability. With our product, users will be able to store files safely using end-to-end encryption, share and receive files from other users and download files on any device that can access the web. Our application will be easy to modify and expand due to its modular nature, and it will be able to interact with other applications making use of APIs.

A good application should be easy to use and its main functions should be performable with just a few clicks while still displaying at all moments information relevant to the user. This is what we strive to do; The user should have a seamless experience and all functions should be easy to find and use.

# User Personas

Andres:

Andres is a 17 year old finishing high school who is interested in architecture and uses programs such as Blender to make some digital building models in his spare time. He has quite a few of these models and they each occupy about ~100 MB of space. Andres only has an older Chromebook with limited space and not only is he running out of space but he is afraid his laptop is becoming too old and he might lose his models. Andres does not like carrying around USB sticks and

# User Story Mapping/Nicolas

## 1. Backbone (High-level project goals):

* File Storage & Sharing
* Data Synchronization
* User Authentication & Permissions
* Security and Compliance
* Hybrid Cloud Implementation
* Scalability and Cost Optimization

## 2. User Activities (Main functionalities users will interact with)

1. Store Files

User Story 1: As a user, I want to upload my files to the cloud so that I can access them from anywhere.

User Story 2: As a user, I want to manage my files (edit, delete, organize) from any device for better accessibility and control.

1. Share Files

User Story 3: As a user, I want to share my files with others, so I can collaborate efficiently.

User Story 4: As a user, I want to set permissions on shared files (view, edit) so that I maintain control over my content.

1. Synchronize Data

User Story 5: As a user, I want my files to be automatically synchronized across all my devices, so I always have access to the latest version.

1. Manage Versions & Backup

User Story 6: As a user, I want access to previous versions of my files to recover from unintended changes.

User Story 7: As a user, I want my files to be backed up in both cloud and local storage, ensuring they are always secure.

1. Access Files Anywhere

User Story 8: As a user, I want to access my files from any device using a web-based interface to ensure flexibility.

User Story 9: As an admin, I want to implement hybrid cloud storage (AWS S3 + on-premises) to allow for flexible data storage options.

1. Collaborate on Documents

User Story 10: As a user, I want to collaborate on documents with others in real-time, enabling seamless teamwork.

User Story 11: As a user, I want to chat with collaborators while working on shared files,

so we can communicate more easily.

1. Secure Files

User Story 12: As a user, I want to ensure all my files are scanned for malware upon upload to protect my data from threats.

User Story 13: As an admin, I want to ensure all user data is encrypted and complies with security standards for safe cloud storage.

## 3. User Tasks (Steps to complete each activity)

1. Store Files

Upload files through the web interface.

Create, delete, and organize folders.

Access files from both cloud and local storage.

1. Share Files

Send a shareable link to collaborators.

Set file permissions (view/edit).

Track changes made by collaborators.

1. Synchronize Data

Automatically sync files across devices.

Ensure version history is saved in case of changes.

1. Manage Versions & Backup

Access file history.

Set up automatic backups in AWS S3 and on-premises storage.

1. Access Files Anywhere

Use the web-based interface to access files.

Set up infrastructure with Terraform and Ansible for scalable, accessible storage.

1. Collaborate on Documents

Edit documents in real-time with multiple users.

Use built-in chat for collaboration.

1. Secure Files

Scan files for malware before upload.

Encrypt user data and ensure compliance with security policies.

## 

## 4. Technical Activities (Infrastructure and Development Tasks)

1. Set Up Cloud Infrastructure (AWS S3)

User Story: As a developer, I want to deploy the cloud infrastructure using Terraform to ensure scalability and automation.

1. Hybrid Cloud Implementation

User Story: As a developer, I want to integrate both cloud (AWS S3) and on-premises (Fontys Infralab) storage to provide users with flexible storage options.

1. Deploy Web-Based Interface

User Story: As a developer, I want to build and deploy a user-friendly web interface so that users can interact with the system.

1. Automation & Scalability

User Story: As a developer, I want to automate infrastructure deployment using Ansible to reduce manual setup and improve efficiency.

1. Containerization

User Story: As a developer, I want to containerize the application using Docker so that it is portable and easily deployable.

### 

### 

### 

### **Product Roadmap**

Our product roadmap provides a high-level timeline for the development and deployment of the cloud-based storage solution. This roadmap is divided into four major phases called sprints, each focusing on specific development goals to ensure a smooth, progressive implementation of features and functionality. There arent many specifications yet because my group and I need to still discussed the infrastructure of our service. After we create one we will divide the tasks to get to our goal into the sprints to try and be always up do day.

#### **Phase 1: Project Initiation and Planning (Weeks 1-3)**

* **Objective**: Establish the foundation of the project and prepare the team for development.

-Finalize project requirements and scope.

-Identify key tools (Terraform, Ansible, AWS S3) and infrastructure.

-Create user personas and user stories.

-Define project milestones and timelines.

-Set up development environment and version control system.

**Deliverables**:

* User requirements specification v1
* Development tools and resources set up.
* Project Plan v1

#### **Phase 2: Core Infrastructure Development (Weeks 4-5)**

* **Objective**: Build the infrastructure necessary for cloud storage and implement basic functionality.

-Set up AWS S3 storage and on-premises infrastructure (Fontys’ Infralab).

-Use Terraform to automate cloud infrastructure provisioning.

-Implement file storage functionality in the cloud.

-Ensure data synchronization between cloud and database storage.

**Deliverables**:

* AWS S3 cloud storage setup.
* Initial deployment scripts using Terraform, prototype release
* Basic file upload and management features.

#### **Phase 3: Web Interface Development and Core Features (Weeks 6-7)**

* **Objective**: Develop the user-facing web interface and core file management functionality.

-Design and deploy a web-based user interface.

-Implement user authentication and permissions management.

-Build file sharing capabilities and version control.

-Develop automatic data synchronization across devices.

**Deliverables**:

* User-friendly web interface.
* Authentication system with file permissions.
* File sharing and version history features.
* Design Document
* Security Management analysis v1.
* Peer review

#### **Phase 4: Advanced Features, Testing, and Deployment (Weeks 7-8)**

* **Objective**: Finalize advanced features, conduct thorough testing, and deploy the product.

-Implement end-to-end encryption and security compliance (file scanning for malware).

-Build real-time collaboration features (editing documents, in-app chat).

-Optimize for scalability and cost efficiency.

-Perform user testing, load testing, and final bug fixing.

-Containerize the application using Docker and finalize Ansible automation scripts for deployment.

**Deliverables**:

* Real-time collaboration and file scanning features.
* Fully encrypted and secure platform.
* Dockerized application with automated deployment.
* Product ready for production deployment.

### **Sprint Goals**

We will break down the roadmap into bi-weekly sprints, each focused on specific goals and deliverables. Below is the outline of sprint goals:

#### **Sprint 1 (Weeks 1-3): Requirement Gathering & Setup**

* **Goals**:

-Finalize the project requirements.

-Complete user personas and user stories.

-Set up the development environment.

**Deliverables**: User Requirements Specification Document, development tools configured, and project timeline established. Project Plan v1 submission.

#### **Sprint 2 (Weeks 4-5): Infrastructure Setup**

* **Goals**:

-Set up AWS S3 and Fontys’ on-premises storage.

-Automate infrastructure provisioning with Terraform.

**Deliverables**: AWS S3 environment, Prototype release presentation in class

#### **Sprint 3 (Weeks 6-7): Basic File Storage and Management**

* **Goals**:

-Implement basic file storage and management functionality.

-Begin data synchronization between cloud and local storage.

**Deliverables**: Cloud storage system with file upload and management. Design Document. Peer review. Security Management analysis v1.

#### **Sprint 4 (Weeks 7-8): Web Interface and Authentication**

* **Goals**:

-Develop the web interface.

-Implement user authentication and permission systems.

**Deliverables**: Functional web interface with login and permissions. Test report. Working in Infrastructure. Design Document Final. User and Technical Manuals v1.

This product roadmap and sprint structure ensures that the project progresses smoothly with clear goals and deliverables for each phase of development.