

A Framework to build Information Systems using a Public Cloud

Diogo André Cardoso Serafim
diogoserafim@tecnico.ulisboa.pt

Instituto Superior Técnico, Oeiras, Portugal

October 2017

Abstract

In the last years cloud computing has revolutionized the way organizations use computers and the internet. Most of the services offered by popular cloud providers, allow organizations to migrate their information systems to the cloud based on subscription-plans regarding the resources they need. However, a few number of small organizations are still unable to support the costs associated. On the other hand, most of systems still in use reside exclusively in Microsoft Excel, with the information spread over multiple documents without any automatization leading to long and exhaustive operations. The purpose of this thesis was the development of a framework that can be adapted to different small organizations by offering a set of tools to migrate the existing systems, as also to easily create a physical database using public cloud services. Finally, considering all the research made about the capabilities and features provided by some cloud providers, extend the framework to new web applications fully integrated with the system implemented in the cloud and, most important, with no costs associated. The proposed framework was evaluated by users to obtain feedback about the framework's usability and user experience. After a positive evaluation, some improvements were made which generated the second version of the new DB4G framework.

Keywords: Cloud computing, Software as a Service, Framework, Google Docs, Extend.

1. Introduction

In the last years cloud computing has revolutionized the way organizations use the computers and the internet. Most of the existing services offered by the popular cloud providers, allow organizations to migrate their current information systems to the cloud based on subscription-plans regarding the resources they need as well as the number of users.

At the same time, organizations are able to improve their business performance, by speed up the existing processes and reducing infrastructure and computing costs. However, a few number of small organizations are still unable to support the costs associated and so they are still reluctant to adopt cloud computing resources as a solution for replacing the current systems.

On the other hand, most of systems currently in use by those organizations resides exclusively in Microsoft Excel, with parts of the information spread over multiple documents without any automatization leading to long and exhaustive operations.

The purpose of this work is to explore all the capabilities and features provided by some cloud providers. Then, taking into account all the research made around these services, build a framework that can be adapted to any type of organization by offering a set of tools to migrate the existing systems, as also tools to easily create a database based on a specific domain model, and finally extend the framework to new web applications, fully integrated with the system implemented in the cloud.

2. Related Work

Since the main goal of this project is the development of a framework using the resources and technologies available across the cloud, it is necessary to understand in a first approach the concept of cloud computing including his architecture and business models. This chapter provides an overview about the core concepts of cloud computing and some open standards related to the authentication and authorization protocols. It also provides a careful analysis of different Software as a Service (SaaS) solutions and Application Programming Interfaces (APIs) offered by some cloud providers.

2.1. Cloud Computing Overview

Cloud Computing is described by the National Institute of Standards and Technology (NIST) as a *model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*, [1].

2.2. Cloud Computing Core Concepts

Cloud provides an architecture that can be divided into 4 layers: hardware, infrastructure platform and application layers. It also employs a service business model where hardware and platform-level resources are provided as services on-demand that can be assigned to each layer and grouped into three categories [1]:

- *Infrastructure as a Service (IaaS)* - consists in the provisioning of infrastructure layer resources, i.e. users can subscribe to their favorite computing infrastructures with specified requirements in terms of hardware configuration, software installation and data access. Examples of IaaS providers include Amazon EC2 and Google Compute Engine.
- *Platform as a Service (PaaS)* - consists in the provisioning of platform layer resources, including operating system support and software development tools to develop, run and manage applications. Examples of PaaS providers include Google App Engine and Microsoft Azure.
- *Software as a Service (SaaS)* - is hosted as a service and provided to customers over the Internet. This is a very important service since it eliminates the need to install and run the application on the local computer, by providing subscriptions to the software hosted on the Internet [4]. Examples of SaaS include Google Docs, Microsoft Office 365, AppSheet.

2.3. Web Authentication & Authorization Open Technology

This section provides an overview about the protocols used for authentication and authorization through the cloud [3].

- *OpenID* - is an open standard for user authentication that allows the use of an existing account to sign into multiple websites without needing to create new passwords. With OpenID the user's password is only given to the identity provider and this one confirms the identity to the websites visited so, no website sees the password.
- *OAuth2.0* - is an open standard designed to work specifically with HTTP, providing a *secured delegated access* for internet users, authorizing websites or applications to access their private information, without sharing their credentials such as passwords or usernames [4].

3. Cloud Computing Solutions

The following SaaS and PaaS solutions were relevant for studying and gaining knowledge about the services offered by some cloud providers. The purpose was to deeply analyse these services and explore their features and capabilities, advantages, disadvantages and points that could be improved.

3.1. Software as a Service Solutions

In SaaS model, users subscribe for the software rather than purchase it, and most important, they subscribe for the exact quantity of the service they require. Applications are subscribed and used online, with files saved in the cloud rather than on individual computers.

3.1.1 Microsoft Office 365

Office 365 is a software as a service (SaaS) suite that allows users to use applications such Word, Excel and PowerPoint, and, depending of the subscription chosen, Outlook, OneNote, Access and Publisher - to save, edit and modify documents in the cloud.

In addition, it provides a cloud file-hosting service - **Microsoft OneDrive** - which allows the storage, backup and sharing of documents, photos and videos and their access using a computer or mobile device.

One of the biggest strengths of Office relies on its support for collaboration and sharing of information by offering various subscription plans to fit the needs of organizations, personal users and also non-profit organizations. It can also be extended with add-on tools from third-party developers.

3.1.2 MLab

MLab is a database as a service platform designed to host MongoDB databases, and is specially dedicated to document storage. It runs on cloud providers, Amazon, Google, and Microsoft Azure (IaaS partners), providing 500MB of free storage, typically used for the development and prototyping of web applications.

3.1.3 G Suite

G Suite is another software as a service suite developed by Google, that provides a set of productivity and collaboration tools and software such Gmail, Google Docs, Google Drive, Hangouts and Calendar.

Google Docs is a totally free web-based software office suite that allows users to create, share and edit documents through the cloud. It also includes Google Sheets and Google Slides allowing users to create spreadsheets and presentations.

It is compatible with Microsoft Office file formats, so users can upload or export word or other text-based documents into the system, and convert them to the online editing program.

The revision history tool (not available in Office 365) provides a full history of all of the changes made to the document, which is extremely useful for keeping track of changes.

It is fully integrated with Google Drive, and all files created are by default saved in Google Drive. The platform also provides tools to import and export Microsoft Office files to and from the cloud.

Google Drive is the cloud storage service created by Google. It allows users to store, synchronize and share files in the cloud. However, the 15GB initially provided are shared with other services such as Gmail and Google+.

3.1.4 AppSheet

AppSheet is another software as service platform that uses a cloud provider, SQL server or any spreadsheet file containing data to create functional mobile applications. The service automatically parses all the content inside the spreadsheet and create the initial application in seconds, where it is possible to customize the interface, security and permissions of other users enrolled. However, the service was developed only to build functional mobile applications.

3.2. Application Programming Interfaces

APIs are defined as a set of routines, functions, protocols and programming standards, used for accessing software application or just for interacting with an external system. When used in the development of third-party applications and services, they act as a gateway enabling direct or indirect access to hardware, software and platform services hosted on the cloud.

3.2.1 Google APIs

Google APIs is a set of programming libraries and protocols developed by Google, whose main goal is to provide communication with google services and their integration with third-party applications. It also gives developers, capabilities to build their own services and to extend their functionality. There are client libraries in various languages, allowing developers to use Google APIs within their code, such as Java, PHP, JavaScript, .NET, etc.

3.2.2 Microsoft Graph API

Microsoft Graph API is a unified API that exposes multiple APIs and other Microsoft cloud services through a single endpoint⁴. It can be used to access data from multiple cloud services, including Azure Active Directory, Exchange Online as part of Office 365, SharePoint, OneDrive, OneNote, etc.

4. Problem Analysis and Solution Design

After considering the different possibilities for the development of a solution taking in consideration the main objectives and available services, it was defined the desired characteristics for a final solution design.

4.1. Requirements

One of the biggest challenges of the previous research was to find a service where our solution could be deployed with no costs associated. After a careful research it was found that only Google Docs satisfies that condition. The service clearly adapts to the solution currently in place in small organizations, allowing them to continue to use a database schema based on spreadsheets, but in a more centralized and secure way using cloud services.

This section describes the analysis and discussions made to elaborate and justify the decisions taken for

the final solution of DB4G Framework, a web-based integrated framework which makes use of the Google Cloud Platform and Google Docs services to communicate, create and maintain an information system in the cloud, available anywhere and without any costs associated. .

In order to meet the solution requirements of the framework, a **domain model** and **use case diagrams** were created in order to guide the implementation of the framework.

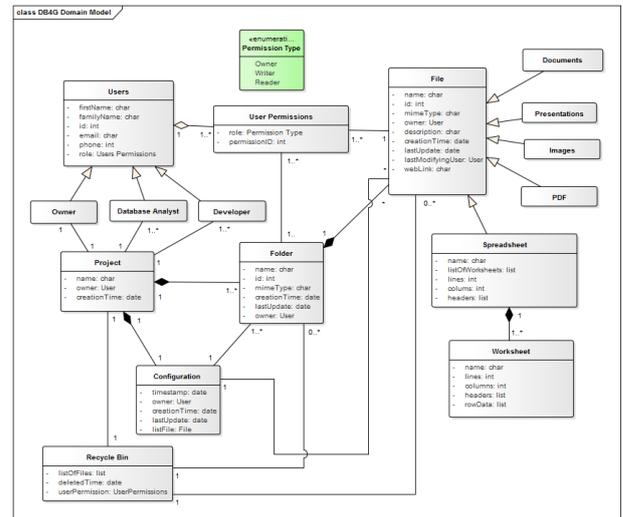


Figure 1: Domain Model

4.2. Front-end Users

The use case presented in figure 2 displays the final requirements for the front-end side of the new DB4G framework.

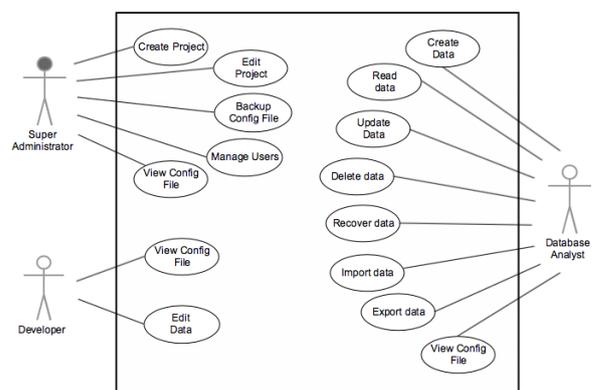


Figure 2: Use Case: Front-end

While the use case in figure 3 displays the final requirements for the back-end side of the new framework.

4.3. Back-end Users

The use case presented in figure 3, presents the current requirements already addressed for the back-end side of the new framework. The use case only contains one actor, the developer.

⁴<http://graph.microsoft.com>

The developer is responsible for the maintenance of the whole framework, its updates and the extension to new functionalities and new web applications. The mainte-

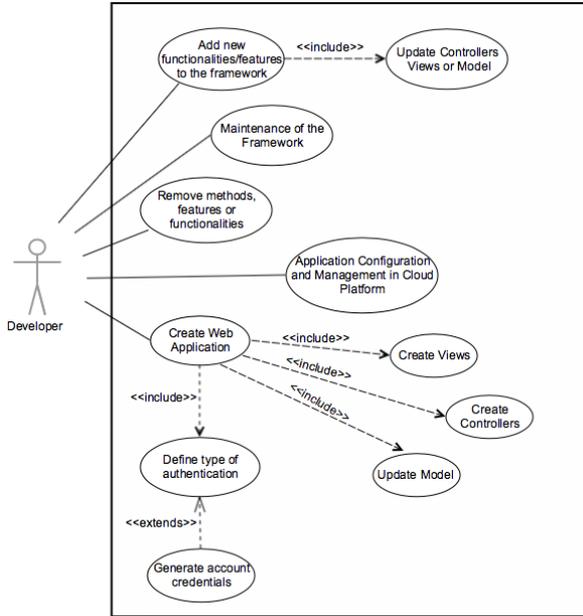


Figure 3: Use Case: Back-end

nance of the system includes not only the front-end and back-end, but also the respective database and the existing resources of cloud. Some example files were created, containing the main aspects of the user interface and the main necessary functions, thus reducing the implementation time and guiding the developer during the development.

5. Framework Architecture and Technology

The architecture used in the final solution of DB4G Framework is presented in figure 4.

The Web browser and the Application Server, represent the main infrastructure communicating with the infrastructure defined by Google Cloud Platform. In order to use cloud services, the framework needs to be registered at Google App Engine. Therefore, private information about the registration will be kept in a JSON file in the application server, containing a *clientID* and a *client secret*, used to identify the framework.

In this architecture, the user interacts with the Web Browser and performs the login in order to use the framework. Every time a user authenticates in the system, the server establishes a communication with the Google OAuth Authorization Server that will be responsible for granting access to the user's private data using a Google API.

After the user authentication, the respective user interface containing all the information about the project is displayed. The Project Area component displays information about the system structure and provides access to the System Management and Data Management areas. It

contains three main components:

- System Management - in this component the user manages all the data related to the system environment (e.g. users management, recycle bin management, etc...).
- Data List Management - in this component the user defines and manages data presented in the cloud database (e.g. file information, spreadsheet information, etc..).
- Task Execution - in this component the user executes tasks related to information provided in the data list management (e.g. edit file metadata, create file, etc...)

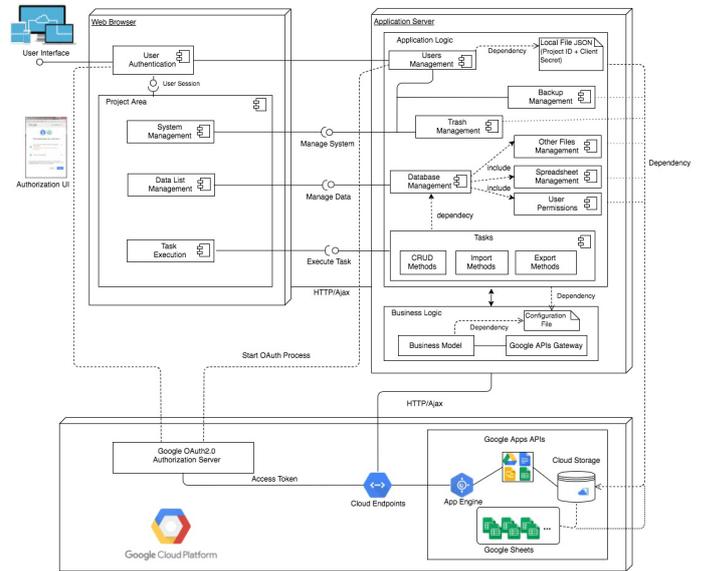


Figure 4: Architecture of DB4G Framework

All functionalities done in the Web Browser by the user will interact with the Application Server through a HTTP/Ajax communication. The Application server is divided in two layers: Application Logic and Business Logic.

The Business logic layer implements the core functionality of the system and encapsulates the business rules that determine how data can be created, stored, and changed. It is defined by a set of generic classes and can be reused across multiple applications or changed in all relevant applications at once.

The Application Logic is a layer built on top of the business logic and serves to implement all the other elements (e.g. specific use cases) that are specific to an application. It contains all the sections that are managed by the user which include Users, System and Database Management as also all the tasks performed by the user. Each component depends from the communication with the cloud database as explained by the flow of dependencies. Every time the user performs a task the Application Logic layer

send the commands to the Business layer which communicates with the Google APIs and send the request to the cloud endpoint to update or retrieve data according to the business rule implemented.

It is also important to mention that the framework was designed using a MVC (Model-View-Controller) pattern.

The technologies used for the concept of this solution are HTML for the front-end (plus Javascript, JQuery, Ajax and CSS) and for the back-end it was created a PHP framework. In addition, it is also used bootstrap for styling purposes.

For the access and management of the cloud services it is used Google Drive API and Google Sheets API, each one directly related to the Google Docs service.

6. Implementation

This section describes in detail the solution implementation and justifies the proposed solution. It was created a User Manual which contains a brief description of the framework, but also displays a detailed tutorial how every function works.

6.1. Home Page Section

The Home Page of the framework is the first page to be seen and accessed by the users. It provides some essential information and contact information as shown in figure 5. At the top right hand corner of the page a login button is also available. In order to access the framework, users

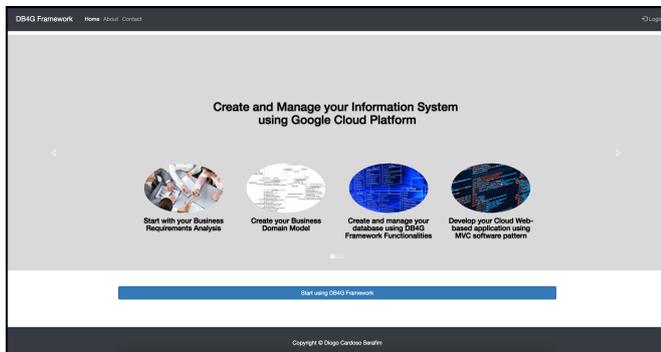


Figure 5: Main Page - Home section

must authenticate using their google account credentials. When the user clicks on the "Login" button, it is redirected to a Google Authentication Server page where his credentials are requested in a process called *user consent*.

If the user grants the access, the Google Authorization Server returns an access token that is temporarily stored in session variables (it is destroyed when the user does a logout).

Finally, the blue button "Start using DB4G Framework", located at the center of the page allows the user to enter into the project interface.

6.2. Project Management Section

After the login is done, and assuming that there is no project created, the user goes directly to the project cre-

ation page in order to create a new project and set up the initial requirements for his own system.

6.3. Create Project

The project creation page, is where the Super Administrator specify details about the project such the name and as also the possibility to add more users according to two different type of roles: Database Analyst and Developer.

All the required fields must be filled or warning messages appears and the project is not created. After validate all the required fields, the administrator can create the project by clicking on the "Create Project" button. A "Cancel" button is also available to cancel all the actions performed. When clicking on the "Create Project" button, the framework automatically creates the project structure into the user's drive account, which is composed by three sub-folders:

1. Root: is the main folder of the project created in Google Drive. This folder contains all the system information including users, database and configuration files.
2. Database: is a sub-folder of the project where all the database information is stored and updated.
3. Configuration: is a sub-folder of the project used to backup the configuration file from the application server.
4. Users: is a sub-folder of the project containing a spreadsheet file where all the information regarding the users are stored.

Note that only the owner of the project has access to the root folder which adds an additional security, since the integrity of the file system structure is not compromised by other users. Once the initial structure is created, the users permissions are added. Like mentioned in the related work, multiple users can work (i.e. can share, edit, view or delete) on the same file in a real-time collaboration.

6.4. Project Area

Once the project is created, the user is redirected to the project area, as showed in 6, which displays the system structure created by the framework. The three sub-

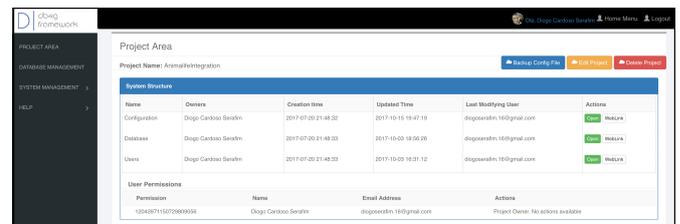


Figure 6: Main Page - Home section

folders are displayed in a table, containing some information about each folder like name, owner, the creation time

and the time of the last update. Note that each folder represents a different area in the system. So the "Database", "Users" and "Configuration" sub-folders correspond to the Database, Users and Configuration Management areas, respectively. For each folder, there are two types of functions available: Open and WebLink. The "Open" button opens the folder and goes to the respective folder section. The "WebLink" button redirects the user to the Google Drive website and displays the select folder.

The "Backup Config File" button allows the Super Administrator to backup the configuration file from the application server, which is automatically stored in the "Configuration" folder. It is also possible to change the name of the project using the "Edit Project" button.

On the left side, a vertical bar was also added where it is possible to perform some actions. The "Project Area" and "Database Management" buttons redirects the user to the areas mentioned above. The "System Management" button allows the user to switch between the areas that are part of the System Management section. The "Help" button contains some guides to help the users while they are using the framework.

7. System Management

The system management section provides access to the following areas: Users Management, Recycle Bin and Configuration File. Each one can be accessed by the user using the left vertical tab available.

7.1. Users Management

The Users Management section displays information about the users of the system. Each user is identified by a unique identifier, name, email address, type of permission and phone.

It is also possible to edit the information of each user by clicking in the "Edit" button or add new users by clicking on the "Add User Button". The option is only available for the Super Administrator. Finally, if the permissions are changed the respective user is notified by email.

7.2. Configuration File

The configuration file contains information regarding the project structure and each file created by the users. Basically, when a new file is created, a reference containing the name and the ID of the specific file is stored in the configuration file. So, every time a user logs into the framework, the server automatically uses the information contained in this file to perform calls to the Google API. Furthermore, when extending the framework to a web application, developers only have to use the references contained in this file to performs actions and make calls to the APIs using the functions already provided by the business model implemented.

7.3. Recycle Bin

All the files available in the recycle bin of a user's drive account, are displayed in a table containing some information regarding each file and options to recover the file

or delete the file permanently. When the user wants to recover a file, the framework automatically moves the file back to the his previous location. Finally, when a file is removed, his reference in the configuration file is removed.

8. Database Management

The Database Management displays all the information about the files and folders that are stored inside the "Database" folder. At the moment, the framework supports the following file types: folders, documents, spreadsheets, pdfs, powerpoints and images. The page contains some functions (on the right) to create and upload files, as also to add users and refresh the page. The page also displays the current location where it is possible to navigate between folders.

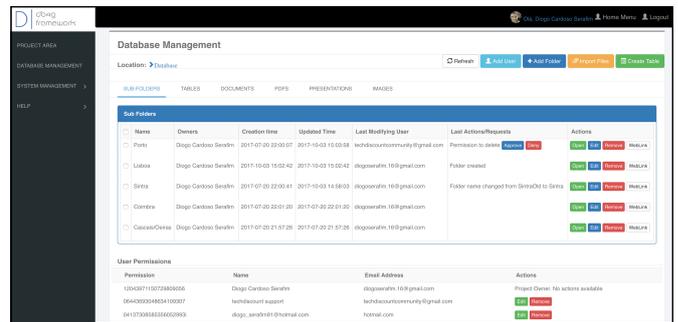


Figure 7: Database Management area

8.1. Create File

In figure 7, the files are displayed in tables containing information of each file which includes the name, the owner, the creation time, the last time the file was modified, the last user that made changes in the file and small information about the last actions or requests. Each file type is identified by a color in the respective table.

The framework provides functions to create new folders or tables. The "Add Folder" button allows the user to create a folder. The user must fill the name, which is validated after the users answers the confirmation message.

The "Create Table" button allows the users to create a new spreadsheet. In order to create a new table the user must provide a name for the new spreadsheet and for the first worksheet, and define the column headers of the spreadsheet. The columns field available is dynamically incremented as more headers are added to the form. All the required fields must be filled or an alert message appears and the spreadsheet is not created.

When the user clicks on the "Create Spreadsheet" button the framework creates the new spreadsheet. It is also possible to cancel all the actions by clicking in the "Cancel" button.

8.2. Preview, Edit and Delete File

It is also possible to preview, edit and delete the files. The "Preview" button allows to preview a specific file.

The "Edit" button allows to change the folder name. Finally, the "Remove" button allows the user to remove the file from the project (a Super Administrator authorization is required).

Note that, for the spreadsheet files, there are five functions available: manage content, edit, remove and download. The "Manage Content" and "Download" buttons will be explained later. The functions "Edit" and "Remove" have the same functionality used for folders. For the rest of the files available (i.e. documents, pdfs and images) the five functions mentioned in the last two sections are also available and have the same purpose.

8.3. Import and Export Files

The "Import Files" button, allows users to import multiple files according to specific file extensions. At the moment the following extensions are supported: .doc, .docx, .xls, .xlsx, .ppt, .pdf, .jpg and .png. The files can be imported by choosing a specific location in the computer or using drag-and-drop features. Each file object contains 3 buttons where is also possible to upload, preview or delete a specific file from the set of files being imported. When the user clicks on the "Upload" button the framework checks, for every file, if an existing file is already available in the database (returning an error message if a file with the same name already exists). Therefore, for each Microsoft Office file being imported, the framework automatically convert the files to a Google Docs format which facilitates the migration of Office files from older systems to the cloud.

Finally, the export function which is represented by the "Download" button available for each file, allows the user to download the files. Additionally it also provides the capability to download exported versions of Google Docs in a format compatible with other office productivity programs such as those offered in Microsoft Office.

8.4. Table Management

When the user clicks in the "Manage Content" button of a specified spreadsheet file it is redirected to the table management area. The page maintains the same aspect and consistency of the pages previously described. It contains four main functions: Back, Refresh, Add User and Create Worksheet. The worksheets are displayed in a table containing some information about each worksheet regarding the name, last time modified by the user, the headers, the number of lines and the number of columns. For each worksheet the three buttons (edit, remove and weblink) described in the previous sections are also presented and have the same purpose.

The "Create Worksheet" button allows the user to add a new worksheet to the current project. The user must choose a name for the new worksheet and define the headers, using the same process used in spreadsheets. After fill the necessary fields, the user answers the confirmation message and creates the new worksheet (recall that the user has the possibility to cancel all the actions before going further).

8.5. User Permissions

The User Permissions section is located in the end of the page of every area of the framework. It displays information about each user assigned to files and folders in the project and provides methods to manage the user permissions, which in this case consists in functions to edit user permissions or just remove user access to a specific file.

8.6. Framework vs API Limitations

Several challenges were found during the implementation of the framework and his integration with Google APIs. This section describes the limitations and challenges encountered during the development and the set of solutions adopted to overcome those limitations.

8.6.1 File Ownership

Since that the Super Administrator is sharing a project in Drive with other collaborators, it was extremely important to prevent other collaborators from delete permanently the files from the project without the permission of the Super Administrator (i.e the project's owner).

Fortunately, Google provides a feature that allows a certain user to transfer the ownership of files and folders to other user. However, due to limitations of Drive API, when a user transfers the ownership of a shared folder to other user, the ownership of files inside the folder remains the same, which will cause security problems.

After a careful research about the possible solutions for the problem, it was found that it could be possible to transfer the ownership of the file at the moment of his creation. At the same time it was found that, the other users could request permissions to delete a certain file to the Super Administrator. The solutions adopted are described bellow:

- Every time a new file is created by a Database Analyst, the framework automatically transfers the ownership of the file to the Super Administrator.
- Every time a Database Analyst wants to delete a file from the database, the framework automatically uses file description tag's to request the permission to the Super Administrator, who decide to approve and remove the file (note that the file will be moved to the trash), or reject the request.

8.6.2 Quota Limits

Google Cloud Platform is used by millions of sites around the world. In order to protect the system from receiving more data that it can handle, and to ensure an equitable distribution of the system resources, Google imposes limits and quotas on the API requests. Taking into account all these aspects, the following 3 tables describe respectively the Google Drive API and Google Sheets API limits and quotas assigned by the Google Cloud Platform to the framework.

Table 1: Google Drive API - quota limits

Quota Name	Limit
Queries per day	1.000.000.000
Queries per 100 seconds per user	1.000
Queries per 100 seconds	10.000

Table 2: Google Sheets API - read quota limits

Quota Name	Limit
Read requests per 100 seconds	500
Read requests per 100 seconds per user	100
Read requests per day	Unlimited

Table 3: Google Sheets API - write quota limits

Quota Name	Limit
Write requests per 100 seconds	500
Write requests per 100 seconds per user	100
Write requests per day	Unlimited

8.7. Performance

Another challenge found during the development of the framework is related with the API performance and the necessary time to retrieve data from the cloud. To overcome this problem, both Drive and Sheets APIs supports batching operations, by pipelining several API calls into a single HTTP request. In short, the implementation of these techniques, resulted in a better utilization of the network and speed up the framework by increasing the network throughput and reducing latency.

The batching operations were applied in the following situations:

- Retrieving metadata for a large number of files.
- Updating file metadata.
- Changing permissions for a large number of files, such as adding a new user or transfer ownership.
- Read and write requests over spreadsheets.

9. Framework Extension

This section describes how the extension of framework to new functions and web applications is done, while maintaining the same interface, consistency and main functionalities for the new additions (which is done by the developer). An Administrative Manual was created which describes all the necessary steps in detail.

9.1. Requirements and Planning

First of all, the developer should define a list of requirements before start to do any coding or design. The list should clarify all the necessary functions, capabilities and characteristics related to the application logic and the plans for creating it. If the developer decides to use a different API that is not yet available in the framework, a new class containing new functions should be added to the business logic layer. To help and guide developers regarding the front and back-end aspects, 5 example files

were created which explain how to add new functions and new APIs to the framework, as also how to add the new functions to the main interface. Finally the developer should plan all the development cycle using an appropriate software development method.

9.2. Implementation

After plan and establish the software requirements, the implementation can now go forward. First the back-end, the developer needs to decide the type of authentication used to authenticate users into the web application. Currently, the framework uses an authentication process based on OAuth2.0 client credentials. However, it is possible to use a **service account**, which is an account that belongs to the web application instead of to an individual user. This type of account is typically used, when a web application uses Google APIs to work with its own data rather than a user's data, so user's aren't directly involved. Moreover, the application call the APIs on behalf of the service account instead of the user.

9.3. Adding a new Google API

If the developer decides to use a different API library that is not yet available in the framework it must add a new class to the business logic layer (i.e. to the model).

9.4. Adding new controllers and views

Since the framework have some controllers already implemented, in order to avoid mistakes and make the structure much more clean and organized, the framework provides a folder named as "system". So, in order to add new controllers to create the new web application, the developer must define them inside that folder.

9.5. Working with data

After the example files have been added to the framework, the developer now adapts and makes the necessary changes according to the list of requirements. It implements the functions needed to call the application server functions (like get data, update data, delete, etc...). Regarding the interface of each section, the developer has to follow the file's structure and change the parts that need to be changed according to the steps described in the Administrator Manual.

10. Evaluation

Since the involvement of real users is a crucial step in the development of a user-centered solution, a series of evaluations was conducted with a group of volunteer users in order to evaluate the framework solution.

10.1. Planning

Before proceeding with the evaluation process, a careful planning was made in order to define all the tasks that will be performed by the participants, as also the amount of time that will be spend on each participant. So, in order to avoid mistakes and have an effective time-management, a timeline containing the sequence of events of the evaluation was prepared and was divided as follows:

- **Prepare the evaluation and provide all the necessary documents to the participant:** 3 minutes;
- **Explain the evaluation and the main goals to the participant:** 5 to 10 minutes;
- **Start executing the tasks:** 25 to 35 minutes in total;
- **Short interview with the participant about the initial impressions:** 5 minutes;
- **Ask the participant to fill a survey regarding the usability, user experience and personal opinion about the framework:** 5 minutes;

The estimated time to complete the tasks was between 25 and 35 minutes, and the total estimated time for the evaluation was approximately 1 hour.

10.2. Setup

Regarding the setup for the evaluation, the participants only needed a computer with internet access and one of the following recommended browsers: Chrome or Safari. A flash drive was also provided containing a folder with some files to be used on specific tasks, a Guidelines document and a User Manual to be read before starting the tasks.

10.3. Tasks

The participants followed a specific order to execute the tasks which started with simple tasks like creating the project, with the level of difficulty increasing when going to the next task. The idea is to obtain a feedback about the functionalities of the framework as also to show the power of Google App engine and Google Docs.

10.4. Participants

In total 7 volunteers participated in the evaluation with all of them having some knowledge about Google Drive and database management. The evaluations were made in person in IST and INESC-ID (Taguspark). These choices were useful to obtain a closer observation of the user's movement and register the mistakes and the time taken to complete the tasks and finally obtain a valid feedback.

10.5. Survey

After completing all the tasks, each user answered the respective survey questions which were divided in three parts: **Usability, User Experience and Personal Opinion.**

In order to evaluate both the usability and user experience two different approaches were used. To conduct the usability evaluation it was used the **System Usability Scale (SUS)**, which is a questionnaire provided to the users for measuring the perceived ease of use of software, hardware, cell phones and websites [5].

Regarding the user experience, it was used the **User Experience Questionnaire (UEQ)**, which is a data

analytical approach that evaluates the interaction of a product[6].

To conclude, the survey had a total of 15 questions divided in: 4 questions regarding the usability, 7 questions regarding user experience and the last 4 regarding the personal opinion) and can be found in Appendix D.

10.6. Results and Discussion

As mentioned, as a measure of usability, it was registered the time that users took to accomplish some tasks, as also the errors and mistakes during specific tasks and their ability to recover from those mistakes.

In addition, it was measured the usefulness of specific functions and aspects provided by the framework. By using the scale provided by the SUS, the participants classified their understanding and satisfaction regarding the whole experience. Finally, for statistical purposes, measures of central tendency were applied, as well as the average of the time that participants spent to perform the tasks. In general the participants considered most of the available functions very easy to accomplish. However, a few number with less knowledge about Google Drive, had some difficult to perform some actions. The results are presented in table 4.

The average time that participants took to execute all the tasks was 28 minutes which indicates that most users performed the tasks inside the expected duration. Regarding the users expectations and satisfaction, the participants had almost a good understanding of what was happening in the framework, as shown in tables 5.

Overall, users found the framework to be easy to use with a pleasing and friendly interface. They considered that the majority of the functions were well integrated and the framework's efficiency was good, but also felt very confident when using the framework (even the users who did not understand the Google Docs service). In general, the 6 scales (i.e. Attractiveness, Perspicuity, Efficiency, Dependability, Stimulation and Novelty used by the UEQ) to be evaluated were archived with a positive note, but the Attractiveness (i.e. users would like to use it) and Perspicuity (i.e. it was easy to use the framework) scales were the least positive. These two depend mostly in the users objectives and knowledge about Google Docs. The positive feedback received regarding the whole framework shows the framework's potential and future growth.

11. Conclusions and Future Work

In the last years cloud computing has revolutionized the way organizations use the computers and the internet. Most of the software as services offered by popular cloud providers, allow organizations to migrate their information systems to the cloud based on subscription-plans regarding the resources they need. However, a few number of organizations are still unable to support the costs associated. On the other hand, most of systems still in use reside exclusively in Microsoft Excel, with the information spread over multiple documents without any automatization leading to long and exhaustive operations.

Quest.	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	1.10	1.11	1.12	1.13	1.14	1.15	1.16	1.17
Med.	A	A	A	A	B	B	A	B	A	A	A	A	A	A	A	A	A
Mod.	A	A	A	A	B	B	A	B, A	A	A	A	A	A	A	A	A	A

* Quest.= Question, Med.= Median, Mod.= Mode.

* From question 1.1 to 1.17: D) Very Difficult, C) Difficult, B) Easy, A) Very Easy.

Table 4: Results regarding usability: medians and modes.

Quest.	2.1	2.2	2.3	2.4	2.5	3	4
Aver.	4.28	4.71	4.71	4.71	4.42	4.2	4.4
Med.	4	5	5	5	4	4	4
Mod.	4	5	5	5	4	4	4

* Aver.= Average, Med.= Median, Mod.= Mode; Quest.2.1 to 2.4: 1) Should not be available and 5) Extremely useful; Quest.3: On a scale of 1 to 5, 1) Strongly disagree and 5) Strongly agree; Quest.4: On a scale of 1 to 5, 1) Very unsatisfied and 5) Very satisfied.

Table 5: Results regarding usability and usefulness: average, medians and modes

Quest.	5	6	7	8	9	10	11
Avg.	4	4.1	4.6	4.3	4.6	4.3	4.7
Med.	4	4	5	5	5	4	5
Mod.	4	4	5	5	5	4, 5	5

* Quest.= Question, Aver.= Average, Med.= Median, Mod.= Mode. From questions 5 to 11: on a scale of 1 to 5, 1) Strongly disagree and 5) Strongly agree.

Table 6: Results regarding user experience

By considering all those aspects, the designed and developed solution was a new framework that explores most of the capabilities and functions offered by Google Docs, developed in a way that it can be adapted to any type of organization by offering a set of generic tools to migrate the actual systems, as also to easily create a database based on the domain model regarding a specific system. Furthermore, it provides diverse and useful functionalities (like importing and exporting files, create tables, user management, etc...).

Moreover, it is possible to extend the framework to a new web application fully integrated with the information system implemented in the cloud.

In conclusion, it can be argued that Cloud Computing can help organizations to increase their business value by developing robust and reliable applications using resources that can be dynamically adjusted. Moreover, Cloud Computing provides a lot of benefits which include:

- Significant Cost Reduction
- Quality of Service
- Scalability
- Rapid Deployment
- File Hosting and Accessibility
- Security of Data and Information

Finally an evaluation was made in order to classify the framework regarding the usability, usefulness and user experience. After measuring the results of the evaluations and analyzing all the received suggestions, it was concluded that some suggestions are considered to be interesting and are good ideas to implement into the framework. These ideas might improve the framework in general and increase the variety of available functions. Like the possibility to create multiple tables from a domain model, importing zip files or export the project folder (currently those actions are not possible to perform due to API limitations).

Currently the framework manipulates the spreadsheet data using Google Sheets API version 3 (at the moment of the development it was the most stable version). However, Google states that the Google Sheets API v4 is now stable. So, as a personal note it is recommended to migrate the framework to Google Sheets API.

Acknowledgements

Firstly, I would like to thank Professor José Borbinha for his guidance during this year and all the support given. I would like to express a special thank to my family and friends for all the help, support and effort. Finally, I would like to dedicate this thesis to my grandparents.

References

- [1] Zhang, Q., Cheng, L., Boutaba, R. (2010). Cloud computing: State-of-the-art and research challenges. *Journal of Internet Services and Applications*.
- [2] Wang, L., Von Laszewski, G., Younge, A., He, X., Kunze, M., Tao, J., & Fu, C. (2010). Cloud computing: A perspective study. In *New Generation Computing*.
- [3] Khalid, U., Ghafoor, A., Irum, M., & Shibli, M. A. (2013). Cloud based secure and privacy enhanced authentication & authorization protocol. *Procedia Computer Science*, 22, 680–688.
- [4] Hardt, D. (2012). The OAuth2.0 Authorization Framework. *Internet Engineering Task Force (IETF)*, 1–76.
- [5] Brooke, J.: SUS: A “quick and dirty” usability scale. In: Jordan, P. W., Thomas, B., Weerdmeester, B. A., McClelland (eds.) *Usability Evaluation in Industry* pp. 189–194. Taylor & Francis, London, UK (1996)
- [6] Laugwitz, B.; Held, T., Schrepp, M. “Construction and evaluation of a user experience questionnaire”. In: Holzinger, A. (Ed.): *USAB 2008, LNCS 5298*, pp. 63-76, 2008.