SUPINFO Academic Dept.

SUPINFO International University

INSTITUTE OF INFORMATION TECHNOLOGY

Group Projects
M1 - Cubbyhole

Project presentation

2013-2014

Version 1.0
Last update: 20/11/2013
Use: Students
Author: Samuel CUELLA

Conditions d'utilisations : SUPINFO International University vous permet de partager ce document. Vous êtes libre de :

- Partager — reproduire, distribuer et communiquer ce document
- Remixer — modifier ce document

A condition de respecter les règles suivantes :

Indication obligatoire de la paternité — Vous devez obligatoirement préciser l'origine « SUPINFO » du document au début de celui-ci de la même manière qu'indiqué par SUPINFO International University – Notamment en laissant obligatoirement la première et la dernière page du document, mais pas d'une manière qui suggérerait que SUPINFO International University vous soutiennent ou approuvent votre utilisation du document, surtout si vous le modifiez. Dans ce dernier cas, il vous faudra obligatoirement supprimer le texte « SUPINFO Official Document » en tête de page et préciser notamment la page indiquant votre identité et les modifications principales apportées.

En dehors de ces dispositions, aucune autre modification de la première et de la dernière page du document n'est autorisée.

NOTE IMPORTANTE : Ce document est mis à disposition selon le contrat CC-BY-NC-SA Creative Commons disponible en ligne http://creativecommons.org/licenses ou par courrier postal à Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA modifié en ce sens que la première et la dernière page du document ne peuvent être supprimées en cas de reproduction, distribution, communication ou modification. Vous pouvez donc reproduire, remixer, arranger et adapter ce document à des fins non commerciales tant que vous respectez les règles de paternité et que les nouveaux documents sont protégés selon des termes identiques. Les autorisations au-delà du champ de cette licence peuvent être obtenues à support@supinfo.com.

© SUPINFO International University – EDUCINVEST - Rue Ducale, 29 - 1000 Brussels Belgium . www.supinfo.com

TABLE OF CONTENTS

© SUPINFO International University – http://www.supinfo.com

1 PROJECT OVERVIEW

Cumorah interactive is creating a new storage service that allows clients to store and synchronize their data online, from various devices. They also want to have a business dashboard to monitor the service exploitation.

You've been selected as a subcontractor to carry the design and development of this product.

2 FUNCTIONAL EXPRESSION

Cubbyhole provides online storage, synchronization and sharing to users tied to a set of "plans". The "Plan" type associated with a certain user determines the rights (Space usage, bandwidth, etc) of that user. Users can share directories with other users, either in read-only or read/write. Users can also share links (over HTTP) to their own resources to non-users: The link is sufficient to access the resource, even though the person using the link doesn't have an account.

The resources can be browsed through two kinds of clients:

- A web client
- A mobile client

It should also provide at least a Win32 binary client that synchronizes the remote folders locally, but doesn't offer a "browsing" functionality.

2.1 SOFTWARE DEVELOPMENT

2.1.1 User accounts

Users can create accounts from the website and select a plan. Users should have at least a mail address and a password. It's not necessary to ask for more mandatory information. They should be able to buy and renew plans using Paypal.

2.1.1.1 Plans

Plans represent access levels and matching rights. A plan has:

- A price
- A duration after which the plan will "expire" and the user will have to pay again the fee to have the plan continued
- A usable storage space

© SUPINFO International University – http://www.supinfo.com

- A max bandwidth usage (up and down) for the account: As an example, if the user is connected through a link that is 1Mbps capable but only have a 500kbps allowed download bandwidth, is downloads from the service will be throttled to 500kbps max.
- A shared link daily transfer quota: If the user has a per-day quota of 1GB and shares a resource using a link to a 500MB file, it will be downloadable twice a day, regardless of the number of persons who have received the link.

It should be possible to create and manage plans. Your demo solution should include a “free” option with low settings.

2.1.2 Application architecture

The application itself run on so-called application servers and provides a webservice API to clients (SOAP, REST, XML-RPC, etc.). These “workers” will in turn execute the actual action on the storage backend on which there is no other requirements than availability, performance and scalability.

Various application clients can use this API. The most important of them is the Web client. The Web client runs on different servers and uses the API to perform the client requested actions.

Other clients, such as mobile client and sync client also use the API from the application, without interfering with the web client.

2.1.3 Web client

The web client provide two main features:

- A commercial website that allows users to register, pay and manage their subscriptions/plans
- A file explorer

The web client should allow users to browse and organize (create directory, move, copy, delete, rename, etc.) their files and folders. They also should be able to download and upload files from/to the machine they are browsing from.

The web client should take advantage of drag and drop features in these management tasks.

The web client should also allow users to share resources with other Cubbyhole users and manage the share permissions (read-only, read-write). For each shared resource it should be possible to see with whom it’s shared, modify the share permissions and revoke the share for that user.

It should also be possible to generate a share link to give access to that resource to non-users.

© SUPINFO International University – http://www.supinfo.com

2.1.4 Mobile client

The mobile client should be a **native application** that replicates all features of the Web client. You're free to choose whichever platform (iOS, Android, Windows Phone, etc.) you want.

The application must be written directly in the native language of the platform. It's allowed to use a library for all platforms to share code, but it's strictly forbidden to use a code-generation tool that automatically generates applications from a common template such as Unity.

2.1.5 Synchronization client

The synchronization client only blindly synchronizes (back and forth) the user Cubbyhole on the local machine. However, the user should be able to select which folders he (doesn't) wants to synchronize.

You must support at least Win32. Each supplementary platform will award you more points, even if your solution is multi-platform by essence (Java, ...).

2.1.6 Business dashboard

You must integrate a dashboard solution that can analyze and report solution-specific metrics such as:

- User statistics: Number of free users, paid user, registration date, plans, etc.
- Usage statistics: Plan specs usage (used space/paid space, used bandwidth/paid bandwidth, etc.)
- IP Geolocation (Europe/Asia/US/South America/...): Where are users located?

To answer questions like and provide indicators such as:

- How long it takes for a free user to become a paying user
- How much free users decide to buy a plan
- Are people using 100% of their plans
- Will a more expensive unlimited plan be more profitable than a less expensive but more limited plan?

Your solution must integrate a dashboard with relevant indicators that will help the management to monitor the activity. It should also provide a way to generate a report by project one metric against two others, such as visualizing the number of new users against 2011 Q3 in Europe. It should also be possible to compare these reports against each other to draw conclusions.

Generate a random set of data to feed your database to perform tests.

© SUPINFO International University – http://www.supinfo.com

2.2 SUPPORTING ARCHITECTURE

Cubbyhole is a large-scale application: It should not depend of the life state of a single machine. The solution has three main components: The application itself that runs on the application servers and is available via web services only, the web client, and the storage. All should be highly available.

Application servers run the core application that provides API services to other clients. There must be more than one of these machines, and it should be simple to add one to the pool. These machines must share the load from various clients and failover if one of them goes down. They in turn use the storage backend to fulfill the client requests.

The storage backend is responsible for storing the actual data. It also can't be a single machine's task: If that machine were to go down, the whole system would also go down. Feel free to use any solution you find fit: iSCSI cluster with replication with the nodes, etc. The only requirement is that actions from the application servers should not fail.

The web client is the main client: It provides both a commercial website and a file manager. It should run on separate servers and can be written in any language, not necessarily the same as the application itself. The web client cannot run on a single machine: The load should be distributed across several machines that also provide failover.

The internal network paths between the servers, the backend and all the machines that belong to the supporting architecture is very important. You must design a robust and scalable "internal" network (router, switches, failover links, etc.) for the solution.

This structure should be monitored and administrators should get alerts on their mail and mobiles whether something in the architecture (network, servers, storage, ...) should go down. The monitoring solution should also provide a health monitor that shows instant status of the whole service.

3 DELIVERABLES

Students should include the following elements in their final delivery:

- A zip archive with the project source code.
- Architecture network design
- Project documentation, based on the template.
    - Technical documentation explaining your choices and/or implementation choices/details on the following items (at least):

© SUPINFO International University – http://www.supinfo.com

  - Language/framework choices
  - Dashboard indicator choices
  - Indicator projection engine
  - Storage choices
  - High availability setup, for both servers and storage
  - Monitoring choices.
- Any additional document you find relevant

These documents are meant to explain why the chosen solutions are the best to fit the client needs. Address the reader as a client, not a teacher. These documents can be in French or in English, at your option.

4 GRADED ITEMS

The project will be graded as follows, on a 320+/310 scale:

Software development (180+ points)

User accounts (10 points)

- User can create accounts and choose a plan (3 points)
- Users can pay non-free accounts using Paypal (7 points)

Plans (30 points)

- There are plans with specified metrics (2 points)
- It's possible to manage (CRUD) plans (3 points)
- Plans disk quotas are enforced (5 points)
- Plans bandwidth quotas are enforced (20 points)

Web client (40 points)

- It's possible to manage (create, move, copy, delete) files and folders (15 points)
- It's possible to download files (2 points)
- It's possible to upload files (3 points)
- It's possible to share files with other users (10 points)
- It's possible to set per-user read-only or read-write sharing permissions (2.5 points)
- It's possible to modify share permissions and revoke sharing (per-user) (2.5 points)
- It's possible to generate share links for non-users (5 points)

Mobile client (40 points)

- It's possible to manage (create, move, copy, delete) files and folders (10 points)
- It's possible to download files (2 points)
- It's possible to upload files (3 points)

© SUPINFO International University – http://www.supinfo.com

- It's possible to share files with other users (10 points)
- It's possible to set per-user read-only or read-write sharing permissions (2.5 points)
- It's possible to modify share permissions and revoke sharing (per-user) (2.5 points)
- It's possible to generate share links for non-users (5 points)
- Per supported platform coefficient on that part: 1 s- 1.X, X = (number of platform – 1)/2

Synchronization client (20 points)

- Synchronize the Cubbyhole locally: download/upload (15 points)
- Users can select which remote folders should be sync locally (5 points)
- Per supported platform coefficient on that part: 1 - 1.X, X = (number of platform other than Win32)/2

Business dashboard (60 points)

- Dashboard Indicators relevancy (20 points)
- It’s possible to generate reports of one metric against two others (30 points)
- It’s possible to compare to report against each other (10 points).

Supporting architecture (130 points)

- Network design: Link redundancy, bandwidth management, etc. (10 points)
- The application servers are highly available and load-balanced (30 points)
- The web client servers are highly available and load-balanced (30 points)
- The storage is highly available and load-balanced. (30 points)
- Monitoring: Administrators are notified if something goes down (20 points)
- There is a health monitor (10 points)

Bonus points (10 points)

- Additional features done by the students (10 points max)

© SUPINFO International University – http://www.supinfo.com