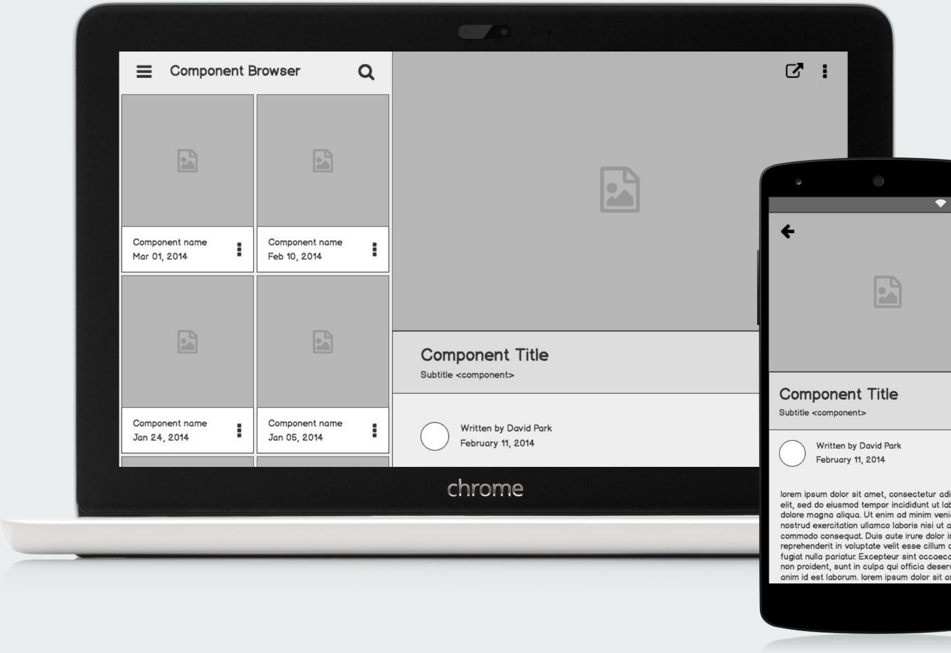
Secure Personal Cloud

Optimize Prime



Outline

Linux Client

Web Client

Server Side

Encryption Schemas

Linux Daemons

spc Command

Android SPC App

Linux Client

Each command executes its corresponding python file stored in the client, which interacts with the server.

Features

- Upload
- Remember
- Set-url
- Status
- Sync
- Download
- Logout
- Dump
- Change-schema

Web Client

We had made different php files for each of the following features.

Features

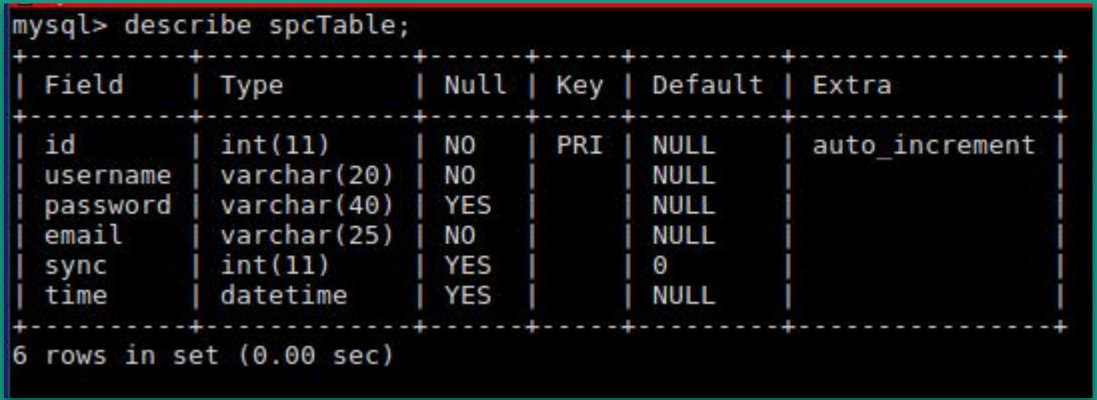
- Register
- Reset password
- Login
- Web list of files
- View File
- Store Schema
- Set reminder

Server Side

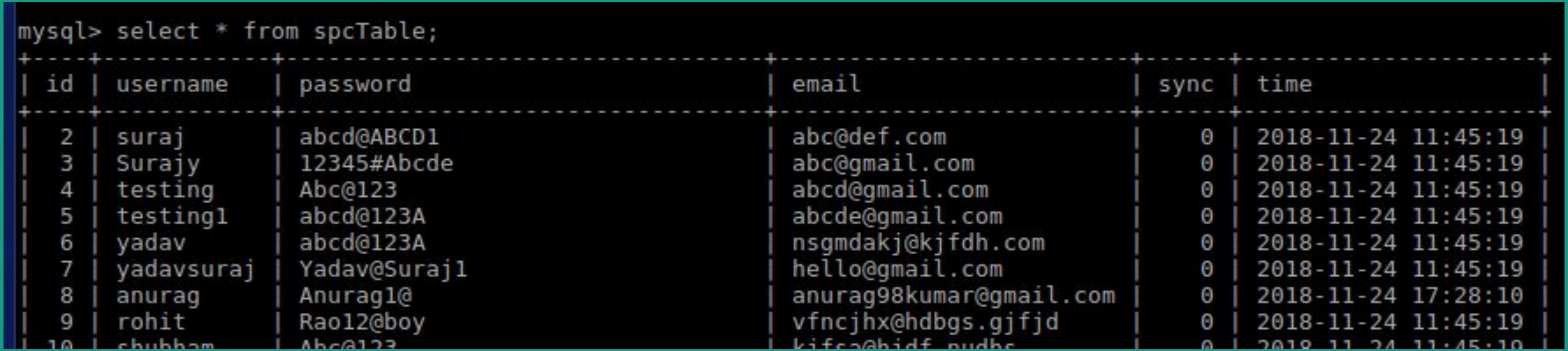
- PHP was chosen over Django because Django provides a built-in framework for Web Development and most of the things have packages. So we preferred starting everything from scratch and learning on the way. It would also improve our PHP skills. Also PHP can run bash and python.

- We used MySQL as our database storage for everything.

DataTable to store user-info

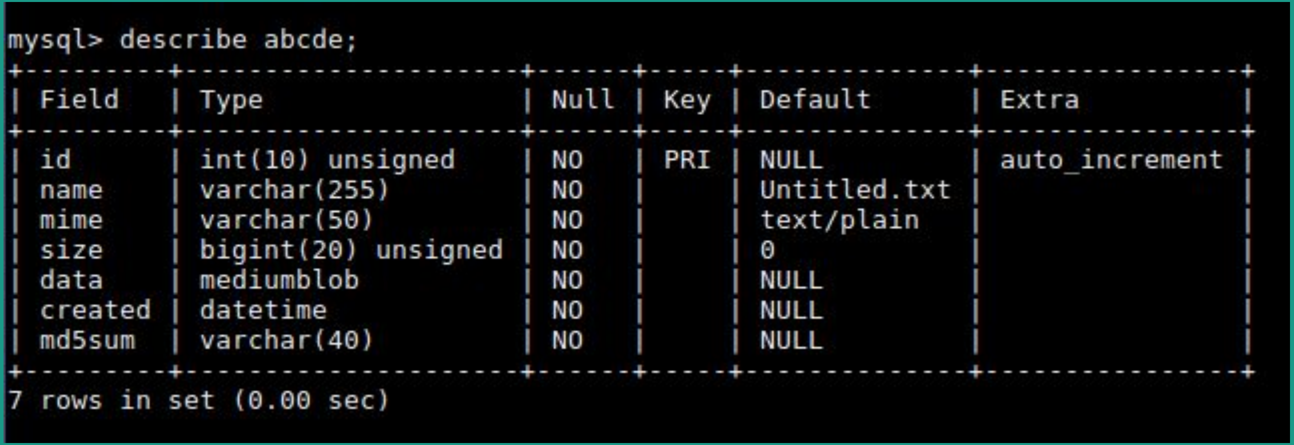


```
mysql> describe spcTable;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| id       | int(11)     | NO   | PRI | NULL    | auto_increment |
| username | varchar(20) | NO   |     | NULL    |                |
| password | varchar(40) | YES  |     | NULL    |                |
| email    | varchar(25) | NO   |     | NULL    |                |
| sync     | int(11)     | YES  |     | 0       |                |
| time     | datetime    | YES  |     | NULL    |                |
+----------+-------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

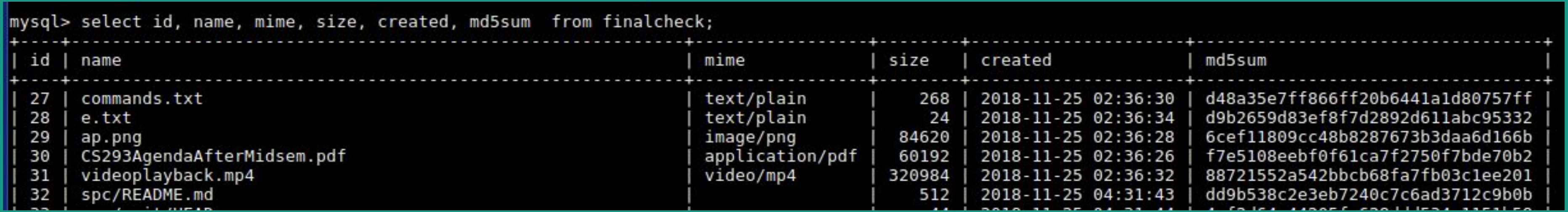


```
mysql> select * from spcTable;
+----+------------+------------------------------------+---------------------------+------+---------------------+
| id | username   | password                           | email                     | sync | time                |
+----+------------+------------------------------------+---------------------------+------+---------------------+
|  2 | suraj      | abcd@ABCD1                         | abc@def.com               |    0 | 2018-11-24 11:45:19 |
|  3 | Surajy     | 12345#Abcde                        | abc@gmail.com             |    0 | 2018-11-24 11:45:19 |
|  4 | testing    | Abc@123                            | abcd@gmail.com            |    0 | 2018-11-24 11:45:19 |
|  5 | testing1   | abcd@123A                          | abcde@gmail.com           |    0 | 2018-11-24 11:45:19 |
|  6 | yadav      | abcd@123A                          | nsgmdakj@kjfdh.com        |    0 | 2018-11-24 11:45:19 |
|  7 | yadavsuraj | Yadav@Suraj1                       | hello@gmail.com           |    0 | 2018-11-24 11:45:19 |
|  8 | anurag     | Anurag1@                           | anurag98kumar@gmail.com   |    0 | 2018-11-24 17:28:10 |
|  9 | rohit      | Rao12@boy                          | vfncjhx@hdbgs.gjfjd       |    0 | 2018-11-24 11:45:19 |
| 10 | shubham    | Abc@123                            | kifsa@hidf.nudhs          |    0 | 2018-11-24 11:45:19 |
```

DataTable to store file-info for users

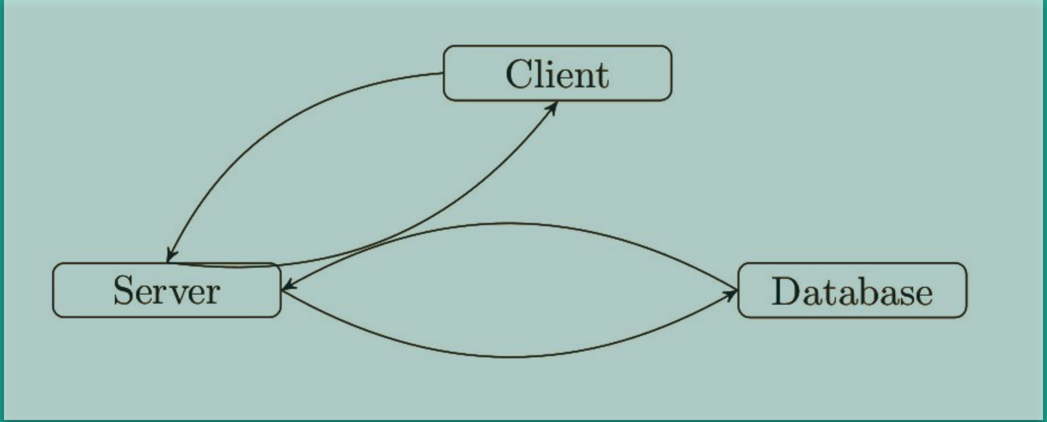


```
mysql> describe abcde;
+---------+---------------------+------+-----+--------------+----------------+
| Field   | Type                | Null | Key | Default      | Extra          |
+---------+---------------------+------+-----+--------------+----------------+
| id      | int(10) unsigned    | NO   | PRI | NULL         | auto_increment |
| name    | varchar(255)        | NO   |     | Untitled.txt |                |
| mime    | varchar(50)         | NO   |     | text/plain   |                |
| size    | bigint(20) unsigned | NO   |     | 0            |                |
| data    | mediumblob          | NO   |     | NULL         |                |
| created | datetime            | NO   |     | NULL         |                |
| md5sum  | varchar(40)         | NO   |     | NULL         |                |
+---------+---------------------+------+-----+--------------+----------------+
7 rows in set (0.00 sec)
```



```
mysql> select id, name, mime, size, created, md5sum  from finalcheck;
+----+-----------------------------------------------------------+-----------------+--------+---------------------+----------------------------------+
| id | name                                                      | mime            | size   | created             | md5sum                           |
+----+-----------------------------------------------------------+-----------------+--------+---------------------+----------------------------------+
| 27 | commands.txt                                              | text/plain      |    268 | 2018-11-25 02:36:30 | d48a35e7ff866ff20b6441a1d80757ff |
| 28 | e.txt                                                     | text/plain      |     24 | 2018-11-25 02:36:34 | d9b2659d83ef8f7d2892d611abc95332 |
| 29 | ap.png                                                    | image/png       |  84620 | 2018-11-25 02:36:28 | 6cef11809cc48b8287673b3daa6d166b |
| 30 | CS293AgendaAfterMidsem.pdf                                | application/pdf |  60192 | 2018-11-25 02:36:26 | f7e5108eebf0f61ca7f2750f7bde70b2 |
| 31 | videoplayback.mp4                                         | video/mp4       | 320984 | 2018-11-25 02:36:32 | 88721552a542bbcb68fa7fb03c1ee201 |
| 32 | spc/README.md                                             |                 |    512 | 2018-11-25 04:31:43 | dd9b538c2e3eb7240c7c6ad3712c9b0b |
```

Server and Client



Encryption Schemas

- AES (Advanced Encryption Standard) 128 bit
- DES (Data Encryption Standard) (>= 64bit)
- BLOWFISH (>= 64bit)

We used Java **Cipher** (javax.crypto.Cipher) class do implement the encryption and decryption.

Linux Daemons

The clients have been provided a feature to get **weekly reminders** to **sync** their Client data with the server data.

1. Reminder from server :-
   This allows you to set a particular day of the week to receive a reminder **via email** to your registered email account by the **official spc email.**
   The user gives his preference of which day of the week is suitable for him.

2. Reminder from client :-
   This allows to notify user using desktop notification every Sunday.

Choices Made

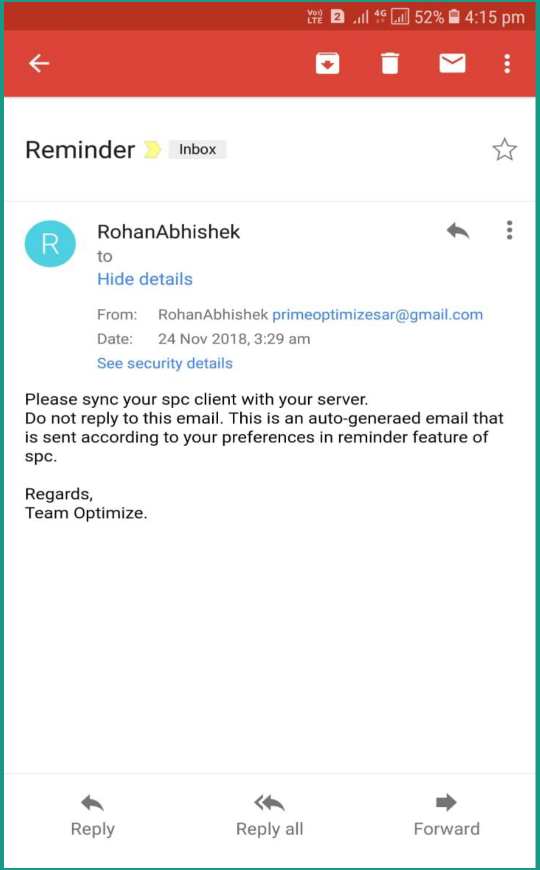
- Adding cron job to crontab of the USER (Client Machine).
- Adding anacron job to the designated USER (Client Machine).
- Adding cron job directly to the SERVER (email from spc).

- First choice is of less use since if the client system is switched off then the cron will not execute resulting in loss of reminder.
- The second choice improves upon the first disadvantage but it doesn’t seem to work if the client changes the current machine in use.
- Hence it was unanimously decided to add Cron Jobs directly in server machine which is functional all the time and is universal to all the users.

We have used 1st and 3rd methods so that user remains up-to-date.

Reminder

From Server



Reminder Inbox

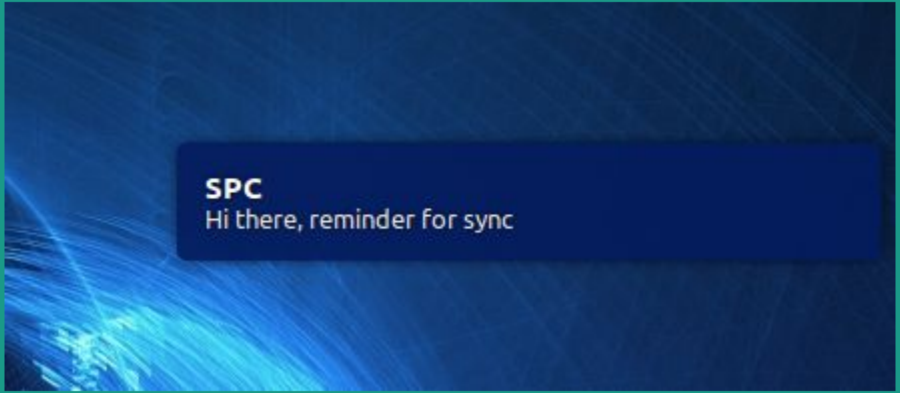
RohanAbhishek
to
Hide details

From: RohanAbhishek primeoptimizesar@gmail.com
Date: 24 Nov 2018, 3:29 am
See security details

Please sync your spc client with your server.
Do not reply to this email. This is an auto-generaed email that is sent according to your preferences in reminder feature of spc.

Regards,
Team Optimize.

From Client

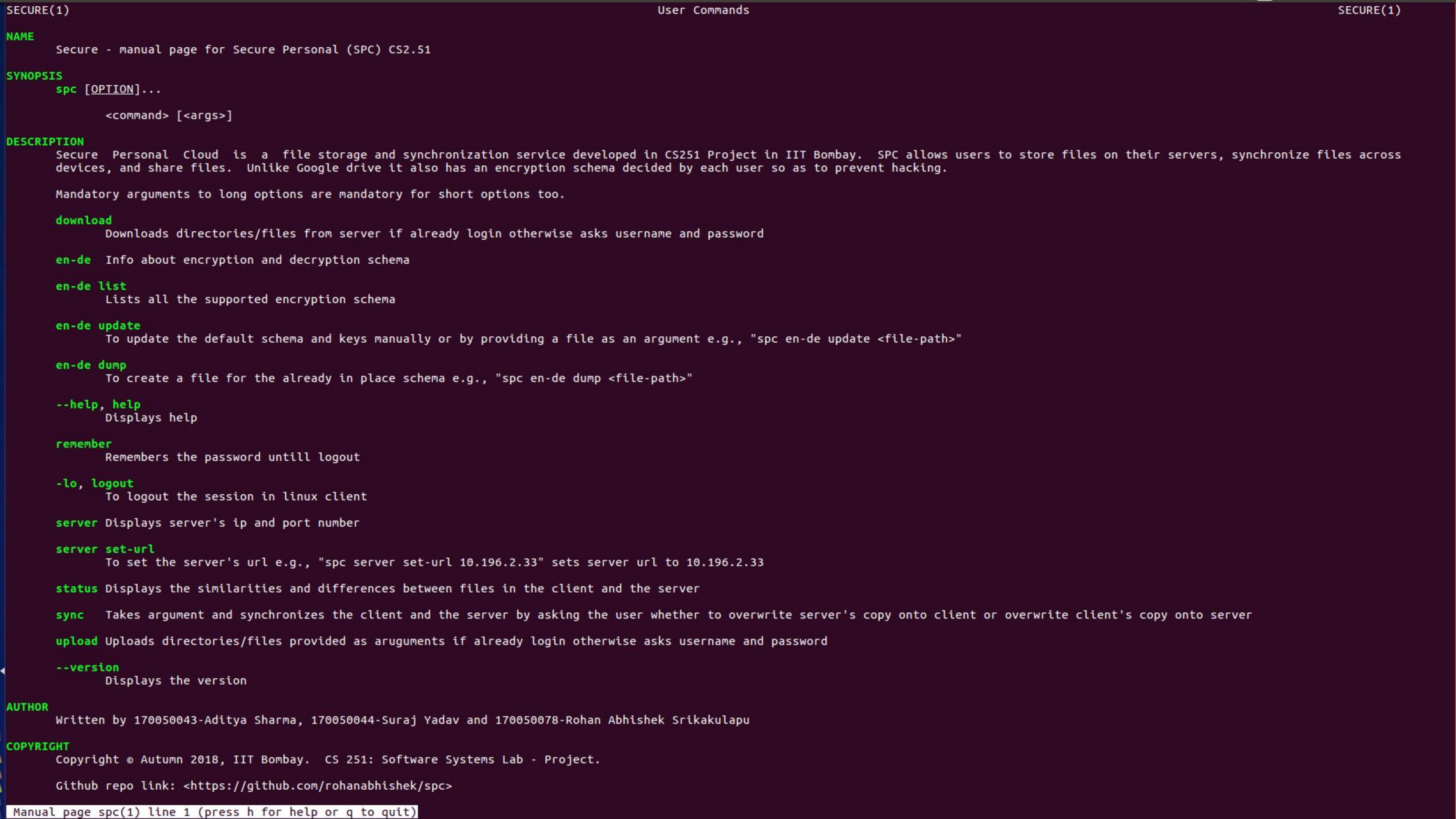


SPC
Hi there, reminder for sync

Linux Tool

A bash script to execute some of the commands and for the remaining a python file is run which indeed takes the arguments from the bash script and responds according to the arguments.

A man page made using “help2man” command



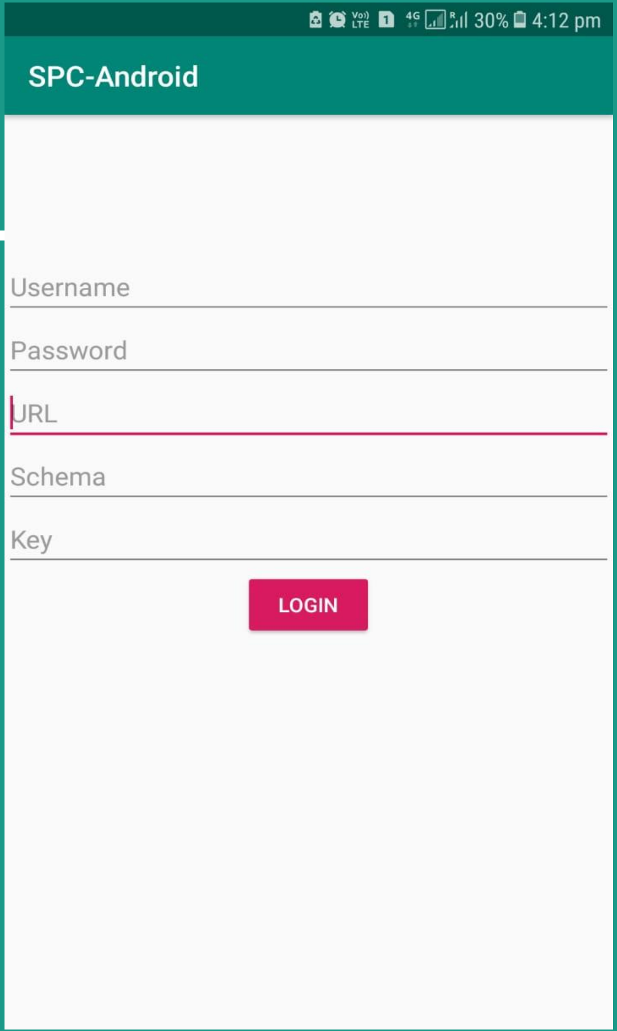
```
SECURE(1)                                     User Commands                                     SECURE(1)

NAME
       Secure - manual page for Secure Personal (SPC) CS2.51

SYNOPSIS
       spc [OPTION]...

              <command> [<args>]

DESCRIPTION
       Secure  Personal  Cloud  is  a  file storage and synchronization service developed in CS251 Project in IIT Bombay.  SPC allows users to store files on their servers, synchronize files across
       devices, and share files.  Unlike Google drive it also has an encryption schema decided by each user so as to prevent hacking.

       Mandatory arguments to long options are mandatory for short options too.

       download
              Downloads directories/files from server if already login otherwise asks username and password

       en-de  Info about encryption and decryption schema

       en-de list
              Lists all the supported encryption schema

       en-de update
              To update the default schema and keys manually or by providing a file as an argument e.g., "spc en-de update <file-path>"

       en-de dump
              To create a file for the already in place schema e.g., "spc en-de dump <file-path>"

       --help, help
              Displays help

       remember
              Remembers the password untill logout

       -lo, logout
              To logout the session in linux client

       server Displays server's ip and port number

       server set-url
              To set the server's url e.g., "spc server set-url 10.196.2.33" sets server url to 10.196.2.33

       status Displays the similarities and differences between files in the client and the server

       sync   Takes argument and synchronizes the client and the server by asking the user whether to overwrite server's copy onto client or overwrite client's copy onto server

       upload Uploads directories/files provided as aruguments if already login otherwise asks username and password

       --version
              Displays the version

AUTHOR
       Written by 170050043-Aditya Sharma, 170050044-Suraj Yadav and 170050078-Rohan Abhishek Srikakulapu

COPYRIGHT
       Copyright © Autumn 2018, IIT Bombay.  CS 251: Software Systems Lab - Project.

       Github repo link: <https://github.com/rohanabhishek/spc>
 Manual page spc(1) line 1 (press h for help or q to quit)
```

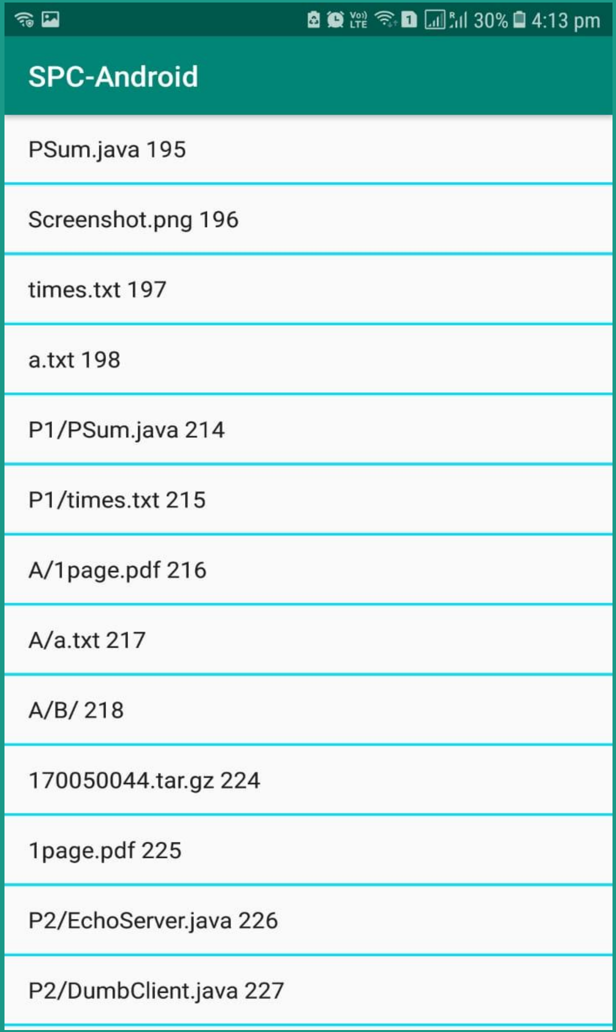
Android App

The first asks for the credentials like username, password, server url, encryption schema and key. It verifies the details and will allow to go to second page only if the credentials are valid

After login, it will show a list of all the files of the user present on the server.

On clicking on the files, it will open that file in the web browser and allow to view the contents of the file.





PSum.java 195

Screenshot.png 196

times.txt 197

a.txt 198

P1/PSum.java 214

P1/times.txt 215

A/1page.pdf 216

A/a.txt 217

A/B/ 218

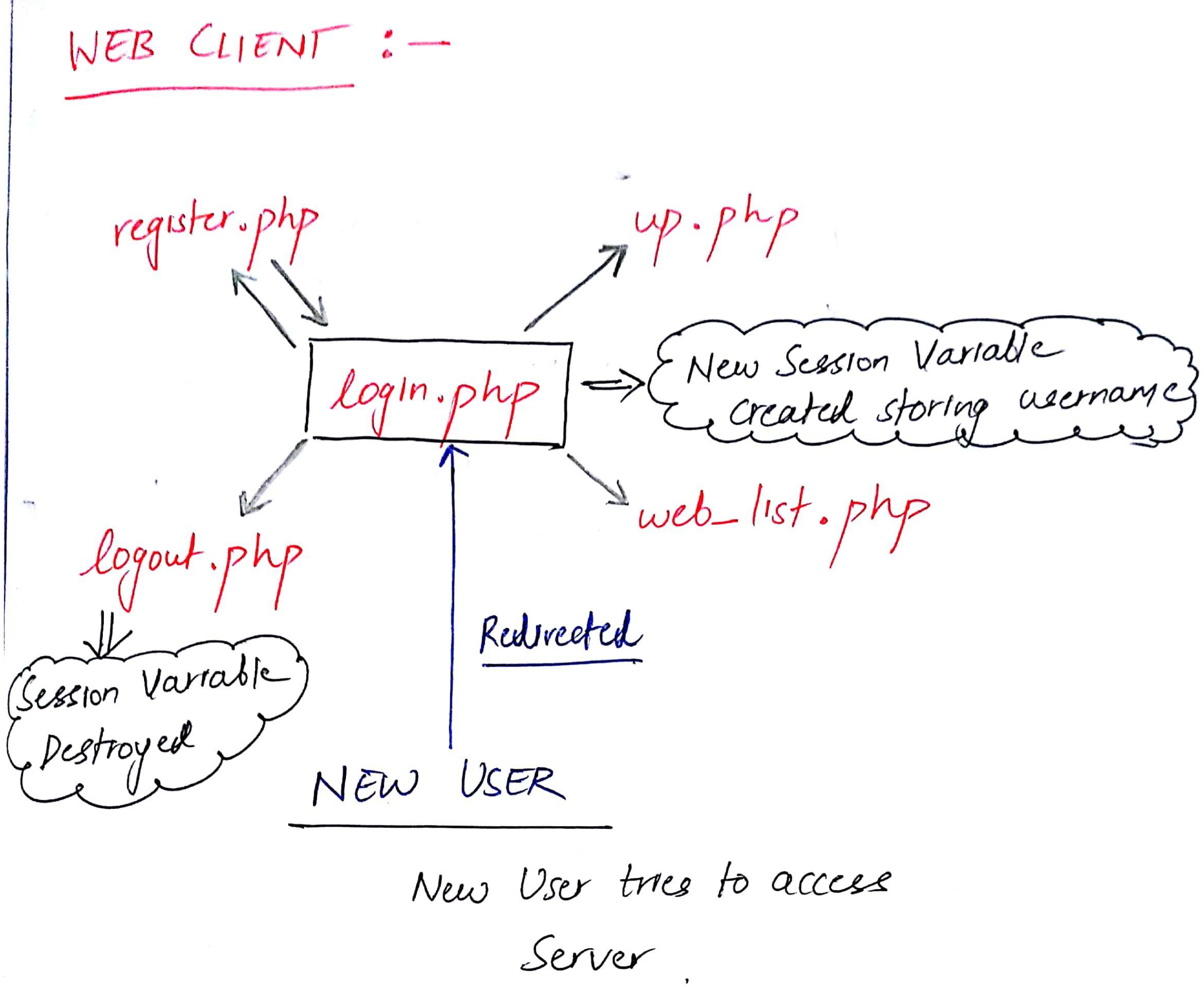
170050044.tar.gz 224

1page.pdf 225

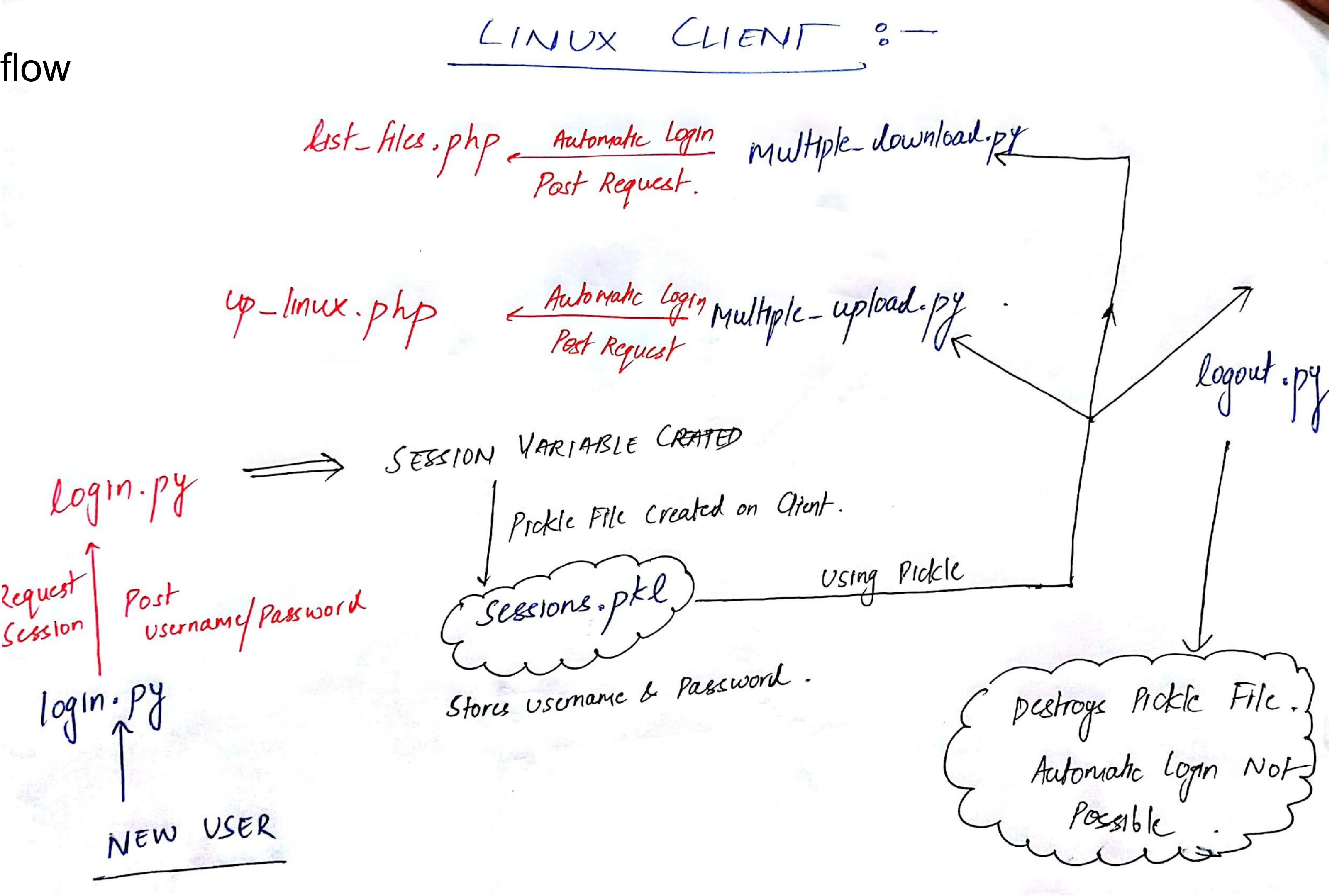
P2/EchoServer.java 226

P2/DumbClient.java 227

Workflow


WEB CLIENT :-
register.php
up.php
login.php
New Session Variable created storing username
web_list.php
logout.php
Session Varrable Destroyed
Redirected
NEW USER
New User tries to access Server

Workflow


LINUX CLIENT :-
list_files.php
Automatic Login
Post Request.
multiple_download.py
up_linux.php
Automatic Login
Post Request
multiple_upload.py
logout.py
SESSION VARIABLE CREATED
login.py
Pickle File Created on Client.
Request Session
Post username/Password
sessions.pkl
Using Pickle
Stores username & Password.
login.py
NEW USER
Destroys Pickle File.
Automatic Login Not Possible.

Reasons for the Choices:

1. PHP was chosen over Django. This was because Django provided a built-in framework for Web Development and most of the things have packages. So we preferred starting everything from scratch and learning on the way.It would also improve our PHP skills. Also PHP can run bash and python.
2. We used separate tables for multiple users.
   a. This is because there wouldn't be any security issues like one user accessing other user's files.
   b. This would increase efficiency of search and query.
   c. This will allow 2 different users to sync simultaneously and independently without any problem.

Assumptions

1. No two Users have same username (restriction imposed by us).
2. Upper limit on Uploaded file size.
3. Two files of same name of a user cannot be present on server (restriction imposed by us).