MINOR PROJECT

System and Network Programming

SharePy

Batch- B3

SUBMITTED BY:-

Mayank Bhola (10103468)
Varun Malhotra(10103478)
Harsh Bhatia(10103487)

Index

Abstract

SharePy is an automated network application that provides hosting platform for data and services. It logs user activities and database in offline or online mode. It provides an extensible environment also provides facility to generate trackers for specific files/data that does not exist at the moment but is expected to be available in near future.

It can used as real time collaboration application in coding environments or in other environments where resource sharing and notification of events is critical and can subtly improve efficiency.
Furthermore it facilitates resource and service sharing with optimum performance

Local file search is taken up to next level by providing transparent search method that will search for requested resource in whole network , but this activity is transparent to the client and enumerate all results from network . Later resource can be requested , and even then process of copying file from other client will be transparent.
SharePy provides infrastructure for different services on different clients throughout the network. The architecture is so designed that new functionalities/ modules can be integrated later easily and is fully customizable.

SharePyis a cross platform application that works on Mac, Windows 7 or higher, Ubuntu, Fedora and has “slick n clean” UI. Asynchronous socket programming is used in project extensively to provide non blocking sockets for real time applications and swift performance.

Built on pure python sockets , SharePy is committed to provide uniform experience on every OS and clean interface and robustness of powerful python support.Python catalysis performance and development process by providing rich API

Batteries included with initial version of SharePy. It contains file tracker which gives notifications about changes made to files , autoconnect module and file pusher.Any custom functions can be added to functionality that SharePy provides in modular and compact form with almost hassle free experience

SharePy is more of an environment for deployment of networked services and hence a unified content hosting system that provides routines for easy socket communication and make content hosting a very easy and hassle free job by wrapping lower level thread interactions and socket routines in compact and usable form.

Division among group members

Names	Work
Varun Malhotra	● Twisted Server-Client Connection Establishment ● GUI implementation in KIVY ● File listing ● File Searching ● File events Notification Broadcasting ● Animation Effects ● Maintaining file events log(creation,deletion,modify,etc.) ● Efficient ChatBot for client-to-client chatting ● File Browser ● UML Diagrams ● Further Report
Mayank Bhola	● Auto-Connect ● Twisted Server-Client Connection Establishment ● File Transferring ● File Syncing ● Maintaining client log ● File Preview ● File Searching ● Finger ● Listing ● Report
Harsh Bhatia	● GUI implementation ● File Searching ● File Browser ● Animation Effects ● File transferring ● Server log ● Preview functionality ● Gantt Chart ● Documentation ● Report

INTRODUCTION

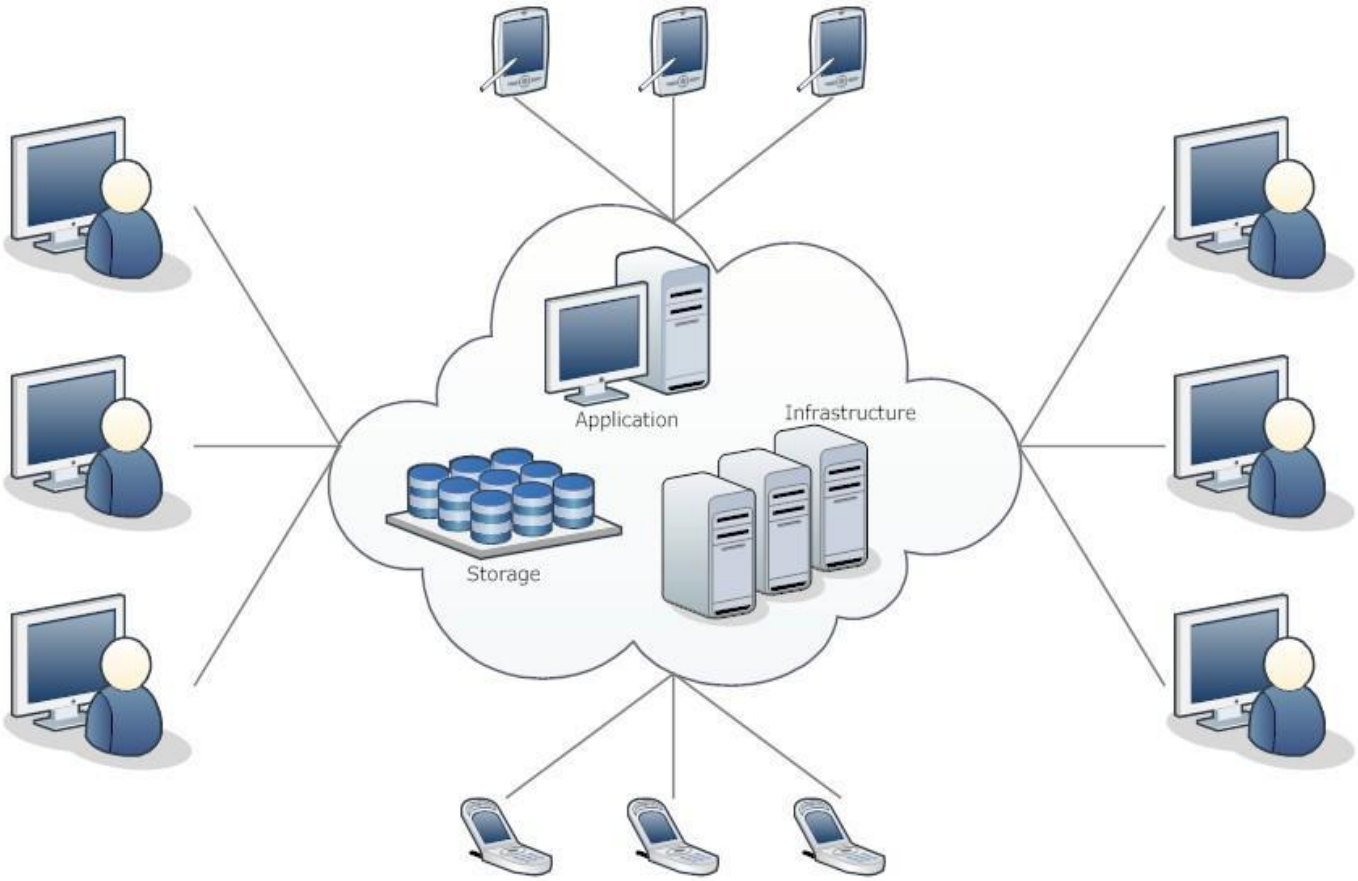
A network service is a service hosted on a computer network. Network services provide some functionality for members or users of the network. Services are usually based on a defined service protocol.

Network services are hosted by servers to provide Network services are configured on corporate local area networks to ensure security and user friendly operation. They help the LAN run smoothly and efficiently. Corporate LANs use network services such as DNS (Domain Name System) to give names to IP and MAC addresses (people remember names like “nm.lan” better than they remember numbers like “210.121.67.18”), and DHCP to ensure that everyone on the network has a valid IP address.

DHCP eases administrative burden by automating the IP assignment of nodes on the network. Adding or removing nodes from the network doesn’t create problems with IP address retrieval; the DHCP service handles this automatically. Authentication servers are another network service, they allow every user to have their own account, and everything they do on that account is logged under their user name. This means that not only are users accountable for anything they do while on the network, but also it increases security as anyone wanting to access the LAN must have a registered user name and password.

Doing network administration without having user accounts to track user activity (illegal or otherwise) or DHCP to automate IP assignment to nodes, or DNS to simplify IP address access would be troublesome indeed. Enabling these few network services automates complex and time consuming administration to the network, and thus eases downtime for network administrators.

E-mail, printing and network file sharing services are also network services. They are seldom not used in a LAN environment, as they allow users to access any printer connected to the network, files on the server or other nodes connected, and streamline data transfer within the network. They require users to have permissions to access the resources shared, and are simple to configure security and access rights for, with the directory service- also a network service.



These network services provides ease and convenience of accessing resources and managing them.Hosting services on network have numerous advantages. Here are some of them :

Low Maintenance :

With hosted applications the hardware and software is the responsibility of the vendor with the client only having to maintain access to the Internet. As there is no software to download a hosted application can be made instantly available across an entire organization even where the organization has multiple regional and international offices.
As there is no software the considerable time and resources required for periodic updates is also completely eliminated.

Reduced Cost :

The need to license software through a ‘per use,’ ‘per concurrent user,’ ‘per server’ or ‘site’ license has been eliminated. Although some hosted applications still do charge ‘per user’ many have taken the opportunity to charge on a ‘usage’ and/or ‘period’ basis meaning that the customer does not have to estimate the number of licenses that will be required, nor install and pay for full licenses that may only be used by some employees occasionally.
This hosted application model also allows the smaller organization and sole trader access to applications written using advanced software development tools and that utilize sophisticated database technology and high-end servers. The capital investment of the hardware and software as well as the security, backup and server maintenance cost are all the responsibility of the vendor.

Minimal Risk :

Many hosted applications require no long term commitment from their customers. New customers! can often test hosted applications either through a free introductory offer or with minimal risk using a Pay-As-You-Go model.
Access

Because the application software and the data is stored on the Internet server the application and data is available to the authorized client once they have logged on from anywhere there is Internet access. The customer can now work anywhere in the office environment, from remote offices or from their home or hotel without the need to download their work or synchronize with portable devices.

Integration :

For organizations that have legacy systems from different suppliers running different operating systems the hosted application will provide an integrated solution to any system that supports a standard Internet browser including Windows, Unix and Macs.

Support :

The hosted application vendors benefit by not having to maintain multiple version numbers or individual maintenance and licensing agreements. New updates are installed centrally negating the need for expensive roll outs. For many applications the need for version numbers is eliminated as updates are made incrementally bringing the benefits to the customer without having to bundle new features up into a new periodic version release. In some cases a user can

request a new feature in the morning that can be developed, tested and deployed globally by the afternoon.

SharePy thus addresses the queries by providing these advantages in hosting.
SharePy is unique in manner the architecture is designed to provide zero-configuration connection and auto discovery of clients and services , thereby keeping user from troubles of manually entering server connection details and discovering automatically.

It uses ping to determine online nodes and then queries each node to determine if instance of SharePy is running on them .After that authentication process starts where server handshakes with nodes and authenticates using private key hashed using md5.The exchange of hash key is fool proof because is already encrypted/hashed.

Use of keys make it possible to run several instances of SharePy in local environment for different services and complex setups

Background Study and Findings

Twisted Client-Server

What is Twisted Client-Server

- Twisted is designed for complete separation between logical protocols (usually relying on stream-based connection semantics, such as HTTP orPOP3) and physical transport layers supporting such stream-based semantics (such as files, sockets or SSL libraries).
- Connection between a logical protocol and a transport layer happens at the last possible moment—just before information is passed into the logical protocol instance. The logical protocol is informed of the transport layer instance, and can use it to send messages back and to check for the peer's identity. Note that it is still possible, in protocol code, to deeply query the transport layer on transport issues (such as checking a client-side SSL certificate).
- Naturally, such protocol code will fail if the transport layer does not support such semantics.

Deferreds

- Central to the Twisted application model is the concept of a *deferred* (elsewhere called a *future*). A deferred is an instance of a class designed to receive and process a result which has not been computed yet, for example because it is based on data from a remote peer. Deferreds can be passed around, just like regular objects, but cannot be asked for their value.
- Each deferred supports a callback chain. When the deferred gets the value, it is passed to the functions on the callback chain, with the result of each callback becoming the input for the next. Deferreds make it possible to arrange to operate on the result of a function call before its value has become available. For example, if a deferred returns a string from a remote peer containing an IP address in quad format, a callback can be attached to translate it into a 32-bit number. Any user of the deferred can now treat it as a deferred returning a 32-bit number.
- This, and the related ability to define "errbacks" (callbacks which are called as error handlers), allows code to specify in advance what to do when an asynchronous event occurs, without stopping to wait for the event. In non-event-driven systems, for example using threads, the operating system incurs premature and additional overhead organizing threads each time a blocking call is made.

Thread support

- Twisted supports an abstraction over raw threads—using a thread as a deferred source. Thus, a deferred is returned immediately, which will receive a value when the thread finishes. Callbacks can be attached which will run in the main thread, thus alleviating the need for complex locking solutions.
- A prime example of such usage, which comes from Twisted's support libraries, is using this model to call into databases. The database call itself happens on a foreign thread, but the analysis of the result happens in the main thread.

Zero Configuration Protocol

- **Zero configuration networking** (**zeroconf**) is a set of techniques that automatically creates a usable Internet Protocol (IP) network without manual operator intervention or special configuration servers.
- Zero configuration networking allows devices such as computers and printers to connect to a network automatically. Without zeroconf, a network administrator must set up services, such as Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS), or configure each computer's network settings manually, which may be difficult and time-consuming.
- Zeroconf is built on three core technologies:

i Assignment of numeric network addresses for networked devices
ii Automatic resolution and distribution of computer hostnames
iii Automatic location of network services, such as printing devices

Software Specification

The various Operating Systems that supports SharePy are listed below
Windows 7, 8
Mac OS ,
Fedora 15 +,
Ubuntu 12.04 onwards
Python 2.7

Hardware Specification

Hardware specification for SharePy are quite simple, computer System running it will require following configuration :
Processor 2.1 GHz+
Ram 500MB+
Graphics Card

User Specification

SharePy works on Windows, liux and Mac Os platform, but still needs many dependencies to be installed to run the developer version. These are as follows :

1. Python 2.7
2. Kivy Gui Platform
3. Python's Twisted Module
4. Watchdog Module Python
5. Python's Pywin32 module
6. Sphnix module
7. OpenGL 2.0 +

Requirements

Functional :

- Ability to view online clients
- Ability to finger(information retrieval) individual clients
- File search platform
- Login based authentication
- File transfer
- File syncing capabilities
- Network file search capability
- Notification Broadcasting of local events
- Tracking changes in local folder
- Chat Functionality
- Pinging individual clients(by their nicknames)
- GUI based server client control
- Maintaining log files
- GUI based file browser
- Auto-connection of clients

Non-Functional :

There are requirements that are not functional in nature. Specifically, these are the constraints the system must follow. They are often called qualities of a system. Other terms for non-functional requirements are “constraints”,” quality attributes”,” quality goals”, ” quality of service requirements” and “non-behavioural requirements”.
Qualities, that is, non-functional requirements, can be divided into two main categories:
1. Execution qualities, such as security and usability, which are observable at run time.
14
2. Evolution qualities, such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the software system.

- **Error handling**

  This shall handle expected and non-expected errors in ways that prevent loss in information and long downtime period.
- **Performance Requirements**

  The system shall accommodate high number of books and users without any fault. Responses to view information shall take no longer than 5 seconds to appear on the screen.
- **Safety Requirements**

  System use shall not cause any harm to human users.
- **Security Requirements**

  System will use secured database

Normal users can just read information but they cannot edit or modify anything except their personal and some other information.
System will have different types of users and every user has access constraints.

Implementation

SharePy is completely written in python with its graphical user interface designed in KIVY framework. Twisted module of python is used in establishing connection between server and client.

The various functions used in Server App are as follows:

- connectionmade
  Function displaying whether a connection with client is established, input address of client. List of available/online clients is then displayed.

- connectionlost
  Detects whether client connection has been lost and removes the respective client's state from the availllable clients' list.

- addclient
  adds client's state and its related information to the list of available clients.

- deleteclient
  deletes client's state and its related info from the list, if the connection happens to be lost between client and server.

- sendAll
  Sends a particular message to all available clients at the same time as in case of Notification Broadcasting on file events on local repository.

- sendTo
  Sends message to a specific client from a client.

The various and main functions used in Client's App are:

- Setup_GUI
  This function sets up/ initialize all the gui elements in kivy language to be used in app of server and client.

- setup_Load
  Main window for starting client. Inputting hostname and port no

- connect_to_server
  this function is defined in client to connect server at specific address and at specific port

.

- send_message
  This function sends a message from client to server for connection.

- start_utilities
  Here elements for functionalities like file listing, watching recent file events log, chatbot are implemented (GUI based)

- print_message
  It decides what type of functionality has server sends. Based on the server's message, this function decides what client will provide. Like, whether a file event has been invoked, request for listing files has been received.

- open_textinput
  this function opens the text window for the client to open input the file names to be listed in App console

- destroy
  destroy all the existing windows opened at current time

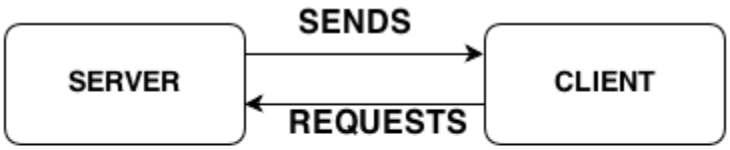
- Backgroubd_stuff
  This function is the core foundation for all the file events notification. A thread is maintained which keeps a watch for any file related events in any client's local repository and notifies all other clients for it so that others can sync.

- chathistory
  this function opens the chat a new window only for the purpose of client-to-client communication for efficient workflow.

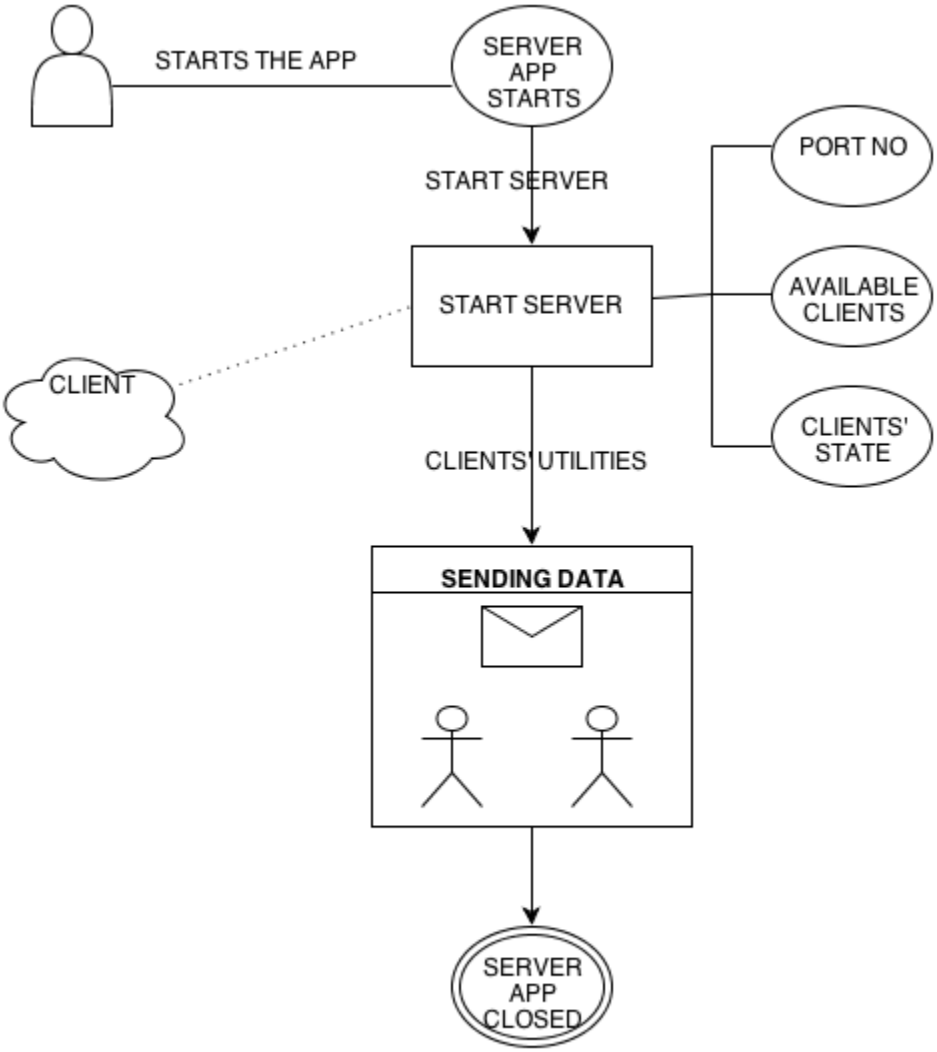
- recent_activities
  Here the log is maintained for all file related events which once came as notifications pop-up

- notify
  notifies through a notification to all the connected clients if there is a file event in any client's local repository.

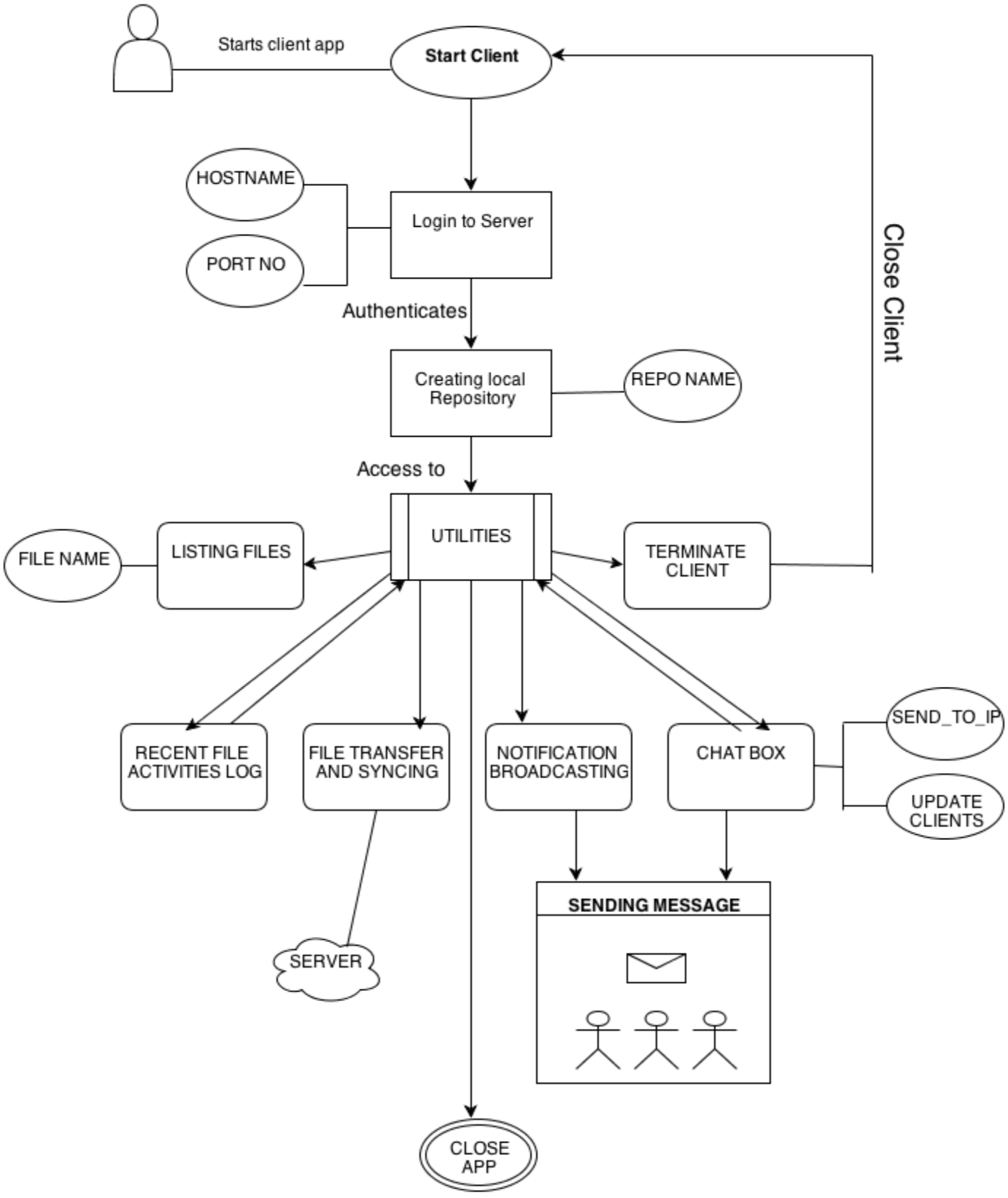
UML Design



TWISTED SERVER-CLIENT SYSTEM



SERVER APP UML DIAGRAM



CLIENT APP UML DIAGRAM

TEST REPORTS

Using ***PYLINT ON CLIENT1.PY***

Report

======

219 statements analysed.

Messages by category

+-----------+-------+---------+-----------+

|type |number |previous |difference |

+===========+=======+=========+===========+

|convention |98 |63 |+35.00 |

+-----------+-------+---------+-----------+

|refactor |3 |3 |= |

+-----------+-------+---------+-----------+

|warning |233 |148 |+85.00 |

+-----------+-------+---------+-----------+

|error |5 |5 |= |

+-----------+-------+---------+-----------+

Messages

+-----------+------------+

|message id |occurrences |

+===========+============+

|W0312 |180 |

+-----------+------------+

|C0324 |33 |

+-----------+------------+

|W0201 |27 |

+-----------+------------+

|C0301 |24 |

+-----------+------------+

|C0111 |22 |

+-----------+------------+

|C0103 |17 |

+-----------+------------+

|F0401 |10 |

+-----------+------------+

|W0613 |9 |

+-----------+------------+

|W0611 |5 |

E1101	4
W0621	3
W0612	2
W0603	2
W0601	2
W0232	2
R0201	2
W0404	1
R0902	1
E0213	1
C0323	1
C0322	1

Global evaluation
Your code has been rated at +6.39/10 (previous run: +5.83/10)
External dependencies
getpass (client)
logging (client)
os (client)
platform (client)
subprocess (client)
threading
\-Thread (client)
twisted
\-internet
\-protocol (client)
\-reactor (client)
watchdog
\-events

```
          | \-FileSystemEventHandler (client)
          | \-LoggingEventHandler (client)
          \-observers       \-Observer (client)
```

Raw metrics

type	number	%	previous	difference
code	228	86.04	151	+77.00
docstring	0	0.00	1	-1.00
comment	14	5.28	7	+7.00
empty	23	8.68	13	+10.00

Statistics by type

type	number	old number	difference	%documented	%badname
module	1	1	=	0.00	0.00
class	4	4	=	0.00	0.00
method	24	20	+4.00	29.17	16.67
function	0	0	=	0	0

Duplication

	now	previous	difference
nb duplicated lines	0	0	=
percent duplicated lines	0.000	0.000	=

==========

PYLINT ON SERVER1.PY

119 statements analysed.
Messages by category

type	number	previous	difference
convention	46	46	=
refactor	1	0	+1.00
warning	47	47	=
error	12	15	-3.00

Messages

message id	occurrences
W0312	27
C0324	17
C0111	13
E1101	12
W0201	10
F0401	9
C0301	8
C0103	8
W0611	3
W0613	2
W0232	2

```
+-----------+------------+
|W0612      |1           |
+-----------+------------+
|W0311      |1           |
+-----------+------------+
|W0222      |1           |
+-----------+------------+
|R0902      |1           |
+-----------+------------+
```

Global evaluation
Your code has been rated at +2.94/10
External dependencies

os (server)
subprocess (server)
twisted
\-internet
\-protocol (server)
\-reactor (server)

Raw metrics

```
+----------+-------+------+---------+-----------+
|type      |number |%     |previous |difference |
+==========+=======+======+=========+===========+
|code      |119    |83.80 |110      |+9.00      |
+----------+-------+------+---------+-----------+
|docstring |1      |0.70  |1        |=          |
+----------+-------+------+---------+-----------+
|comment   |7      |4.93  |10       |-3.00      |
+----------+-------+------+---------+-----------+
|empty     |15     |10.56 |15       |=          |
+----------+-------+------+---------+-----------+
```

Statistics by type

```
+---------+-------+-----------+-----------+------------+---------+
|type     |number |old number |difference |%documented |%badname |
+=========+=======+===========+===========+============+=========+
|module   |1      |1          |=          |0.00        |0.00     |
+---------+-------+-----------+-----------+------------+---------+
|class    |3      |3          |=          |0.00        |0.00     |
+---------+-------+-----------+-----------+------------+---------+
```

|method |13 |13 |= |30.77 |46.15 |
+---------+-------+-----------+-----------+------------+---------+
|function |0 |0 |= |0 |0 |
+---------+-------+-----------+-----------+------------+---------+
Duplication

+-------------------------+------+---------+-----------+
| |now |previous |difference |
+=========================+======+=========+===========+
|nb duplicated lines |0 |0 |= |
+-------------------------+------+---------+-----------+
|percent duplicated lines |0.000 |0.000 |= |
+-------------------------+------+---------+-----------+
==========

Pylint on client2.py

124 statements analysed.

Messages by category

+-----------+-------+---------+-----------+
|type |number |previous |difference |
+===========+=======+=========+===========+
|convention |26 |NC |NC |
+-----------+-------+---------+-----------+
|refactor |0 |NC |NC |
+-----------+-------+---------+-----------+
|warning |11 |NC |NC |
+-----------+-------+---------+-----------+
|error |5 |NC |NC |
+-----------+-------+---------+-----------+

Messages
+-----------+------------+
|message id |occurrences |
+===========+============+
|C0301 |12 |
+-----------+------------+
|C0103 |10 |
+-----------+------------+
|W0201 |5 |

```
+-----------+------------+
|E1101      |5           |
+-----------+------------+
|C0111      |4           |
+-----------+------------+
|W0611      |2           |
+-----------+------------+
|W0622      |1           |
+-----------+------------+
|W0232      |1           |
+-----------+------------+
|W0223      |1           |
+-----------+------------+
|W0222      |1           |
+-----------+------------+
```

Global evaluation

Your code has been rated at 5.00/10

External dependencies

```
optparse (client1)
os (client1)
twisted
\-internet
| \-defer (client1)
| \-protocol (client1)
| | \-ClientFactory (client1)
| \-reactor (client1)
| \-stdio (client1)
\-protocols
\-basic (client1)
```

Raw metrics

```
+----------+-------+------+---------+-----------+
|type      |number |%     |previous |difference |
+==========+=======+======+=========+===========+
|code      |122    |77.71 |NC       |NC         |
+----------+-------+------+---------+-----------+
|docstring |6      |3.82  |NC       |NC         |
+----------+-------+------+---------+-----------+
|comment   |15     |9.55  |NC       |NC         |
+----------+-------+------+---------+-----------+
|empty     |14     |8.92  |NC       |NC         |
+----------+-------+------+---------+-----------+
```

Statistics by type

type	number	old number	difference	%documented	%badname
module	1	NC	NC	0.00	0.00
class	3	NC	NC	0.00	0.00
method	12	NC	NC	100.00	58.33
function	0	NC	NC	0	0

Duplication

	now	previous	difference
nb duplicated lines	0	NC	NC
percent duplicated lines	0.000	NC	NC

==========

PYLINT ON SERVER2.py

Report

127 statements analysed.

Messages by category

type	number	previous	difference
convention	25	NC	NC
refactor	2	NC	NC
warning	129	NC	NC
error	16	NC	NC

Messages

message id	occurrences
W0312	121
E1101	16
C0301	12
C0103	8
C0111	5
W0622	2
W0201	2
W0612	1
W0611	1
W0232	1
W0222	1
R0912	1
R0201	1

Global evaluation

Your code has been rated at +4.58/10

External dependencies

```
        optparse (server1)
        os (server1)
  twisted
        \-internet
        | \-protocol (server1)
        | \-reactor (server1)
```

\-protocols
\-basic (server1)

Raw metrics

type	number	%	previous	difference
code	125	80.13	NC	NC
docstring	4	2.56	NC	NC
comment	15	9.62	NC	NC
empty	12	7.69	NC	NC

Statistics by type

type	number	old number	difference	%documented	%badname
module	1	NC	NC	0.00	0.00
class	2	NC	NC	0.00	0.00
method	8	NC	NC	75.00	62.50
function	0	NC	NC	0	0

Duplication

	now	previous	difference
nb duplicated lines	0	NC	NC
percent duplicated lines	0.000	NC	NC

==========

PYLINT ON COMMON.PY

28 statements analysed.
Messages by category

type	number	previous	difference
convention	5	NC	NC
refactor	0	NC	NC
warning	6	NC	NC
error	1	NC	NC

Messages

message id	occurrences
W0622	5
C0301	4
W0311	1
E1101	1
C0111	1

Global evaluation
Your code has been rated at 4.29/10
External dependencies
hashlib (common)
Raw metrics

type	number	%	previous	difference
code	29	70.73	NC	NC

|docstring |12 |29.27 |NC |NC |
+----------+-------+------+---------+-----------+
|comment |0 |0.00 |NC |NC |
+----------+-------+------+---------+-----------+
|empty |0 |0.00 |NC |NC |
+----------+-------+------+---------+-----------+

Statistics by type

+---------+-------+-----------+-----------+------------+---------+
|type |number |old number |difference |%documented |%badname |
+=========+=======+===========+===========+============+=========+
|module |1 |NC |NC |0.00 |0.00 |
+---------+-------+-----------+-----------+------------+---------+
|class |0 |NC |NC |0 |0 |
+---------+-------+-----------+-----------+------------+---------+
|method |0 |NC |NC |0 |0 |
+---------+-------+-----------+-----------+------------+---------+
|function |6 |NC |NC |100.00 |0.00 |
+---------+-------+-----------+-----------+------------+---------+

Duplication

+------------------------+------+---------+-----------+
| |now |previous |difference |
+========================+======+=========+===========+
|nb duplicated lines |0 |NC |NC |
+------------------------+------+---------+-----------+
|percent duplicated lines |0.000 |NC |NC |
+------------------------+------+---------+-----------+

FUTURE SCOPE:

The project is very useful for numerous organizations as well as for the developers. This twisted Server-Client App provides *team collaboration management system*. Managing a big project is very difficult and at times, tends to be a mess for a group to integrate each other's work. Thus, the project aims at providing REAL-TIME COLLABORATION MANAGEMENT SYSTEM through its advanced features and functionalities using robust algorithms.

Benefits to a team

Suppose, there are four (4) members in a team. They can utilize this App efficiently in carrying out their project as this App serves dual purpose. Firstly, it provides an editor where one can load, edit and save file. Secondly, it provide features like:

- Uploading clients' file on Server
- Easy SYNC functionality to sync with files available on Server
- Efficient and well secured File Transferring and Searching
- Notification Broadcasting whenever a local repository of any client received a file related events like creation, deletion, modification of file, etc.
- Chatting functionality is very reliable and provides a medium to communicate with each other regarding project and their doubts

All the above points help in efficient management of a project besides providing common and advanced feature of file storage, syncing, transmission, notifying, chatting, etc.

In future, it can be extended further, since it has the capability of modularizing and extending the framework it provides, which in turn, will make it more useful to be a standalone App providing a handful of File related services. It's a good source of Content Management System (CMS).

REFERENCES

[1] Richard Stevens, *UNIX Network Programming*, Volume 1(2nd Edition): Networking APIs - Sockets and XTI : Addison- Wesley,1993.

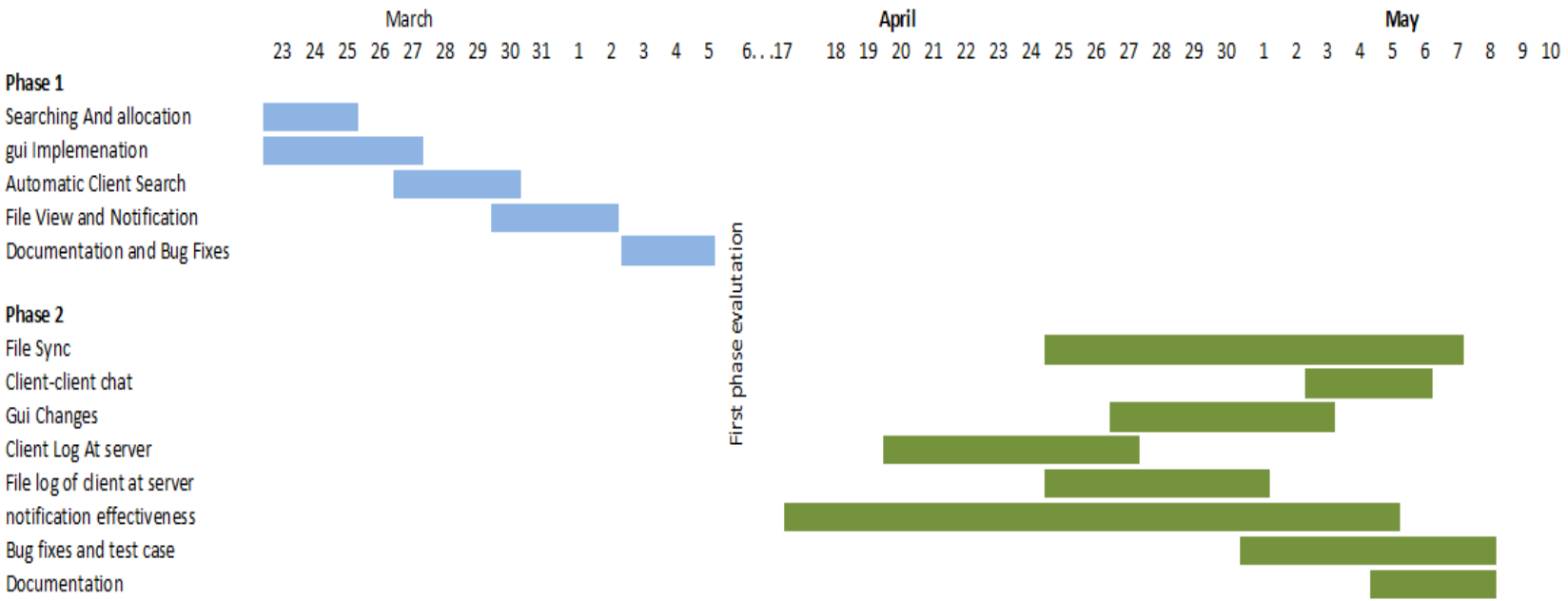
[2] Twisted Server on Multiple Clients, “http://stackoverflow.com/questions/4604580/twisted-server-for-multiple-clients”.

[3] Kivy Online Documentation, “http://kivy.org/docs/guide-index.html”.

[4] Online Gannt Chart Maker, “https://teamgantt.com/”.

[5] Official site of Twisted Module python, “http://twistedmatrix.com/trac/”.

[6] wikipedia reference for twisted server and client, “http://en.wikipedia.org/wiki/Twisted_(software)”


March
April
May
23 24 25 26 27 28 29 30 31 1 2 3 4 5 6. . .17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4 5 6 7 8 9 10
Phase 1
Searching And allocation
gui Implemenation
Automatic Client Search
File View and Notification
Documentation and Bug Fixes
First phase evalutation
Phase 2
File Sync
Client-client chat
Gui Changes
Client Log At server
File log of client at server
notification effectiveness
Bug fixes and test case
Documentation