Funky Torrents

©The Group

Software Design Specification

Version 1.0
IT University of Göteborg
TIG029 – Software Architecture for Distributed Systems

1. Revision History

Name	Description	Date	Version
Michal Musialik	First draft	29/09/10	0.01
Michal Musialik	Added, 2,3,4	03/10/10	0.02
Aydin Halilov	Changed 2	04/10/10	0.05
Michal Musialik,Savas Aydin	Updated 3	15/10/10	0.09
Michal Musialik	Added7,8,9	18/10/10	0.1
Michal Musialik	Changed 6	25/10/10	0.2
Michal Musialik	Updated 2,3,5,6,7,8,9	06/11/10	0.3
Michal Musialik, Magnus Bergqvist	Review entire document	07/11/10	0.4
Michal Musialik	Review entire document	25/11/10	0.5
Michal Musialik	Updated 3	29/11/10	0.6
Michal Musialik	Updated 2	06/12/10	0.7
Michal Musialik, Björn Eriksson	Updated 2	20/12/10	0.8
Björn Eriksson	Updated 1,2,3,4,5,6,7	21/12/10	0.9
Michal Musialik	Review entire document	21/12/10	1

Table 1. Revision History

2. Introduction

1.1 Purpose

The purpose of this document is to present design information of Funky Torrents client, developed by The Group in the term 3 of Software Engineering and Management at IT University.

1.2 The Scope

This document covers the following:

- Purpose
- System overview
- Related design areas of the project

1.3 The audience

This document refers mainly to teachers and supervisors that may use it as one of the criteria in grading this project and also other groups that seek guidance or reference.

1.4 References

- http://www.cmcrossroads.com/bradapp/docs/sdd.html (access on 27-09-2010)
- http://en.wikipedia.org/wiki/BitTorrent_(software) (access on 30-09-2010)
- http://torrentfreak.com/bittorrent-still-king-of-p2p-traffic-090218/ (access on 30-09-2010)
- http://www.softopia.demon.co.uk/2.2/designmethodsforeveryone.html#1 (access on 15-10-2010)
- http://www.bittorrent.org/beps/bep_0003.html (access on 30-09-2010)
- http://www.erlang.org/doc/man/crypto.html (access on 22-10-2010)
- http://en.wikipedia.org/wiki/List_of_web_browsers (access on 06-11-2010)
- https://github.com/mochi/mochiweb (access on 26-12-2010)
- http://www.chicagoboss.org/ (access of 26-11-2010)
- http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf (access on 26-11-2010)

1.5 Term, Acronyms

- BitTorrent software - is a peer-to-peer program developed by Bram Cohen and BitTorrent, Inc. used for uploading and downloading files via the BitTorrent protocol. BitTorrent was the first client written for the protocol. It is often nicknamed Mainline by developers denoting its official origins.
- SDS – Software design specification
- BitTorrent protocol - is a peer-to-peer file sharing protocol used for distributing large amounts of data. BitTorrent is one of the most common protocols for transferring large files, and it has been estimated that it accounted for roughly 27–55% of all Internet traffic (depending on geographical location) as of February 2009.
- Funky Torrents – name of the torrent client created by The Group
- The Group – name of developing team of Funky Torrents
- Bencode - the encoding used by the peer-to-peer file sharing system BitTorrent for storing and transmitting loosely structured data. It supports four different types of values:
    - byte strings
    - integers
    - lists
    - dictionaries (associative arrays).
- GUI – Graphical User interface
- Erlang - general-purpose concurrent, garbage-collected programming language and runtime system.
- Mochiweb - Mochiweb is an Erlang library for building lightweight HTTP servers.
- HTML – hyper text mark-up language
- Chicago Boss - Rapid MVC web developing for Erlang
- MVC – (model, view and controller), one of the architecture models
- Crypto – Erlang decoding module for SHA1 and other cryptographic functions

3. System overview

Funky Torrents is a bit torrent client which allows library customers to download electronic publications using bit torrent protocol.

3.1.Interface

The interface of the Funky Torrents is build with HTML and Erlang using the Chicago boss web framework. Customer is provided with following functions:

- logging in with user name and password
- searching through electronic publications provided by Google
- information about project group

3.2.Data processing

- **I/O files:**

The electronic publication will be written to the customer's hard-drive for every piece of received data.

- **Connection:**

Funky Torrents use the gen_tcp module which is providing functions for communicating with sockets using the bit torrent protocol.

- **Message passing:**

Starting with sending a handshake to requested peer from the information provided by the Tracker. Receiving what pieces the peer has in a form of a bit-field. Afterwards sends an interested message to the peer which returns an un-choked, when un-choked Funky Torrents sends a request for each piece at the time. The pieces are returned in form of blocks in form of a "piece" message.

- **Torrent File:**

Torrent Files are built of bencode and info hash. Extracted bencode data will be saved in a Key-Value tuple that will be matched in the next step with the appropriate Key for further computation.

- **Bencode:**

Bencode supports four different kinds of data:

  - String – any kind of text

- Integer – any number
- Dictionary – Keys and Values, every key returns a value
- Lists – lists of strings, integers , dictionaries or other lists
- Info Hash - Bit torrent file's info hash using the SHA1 hash coding, which will be handled by an Erlang build in module crypto.

4. Design assumptions and constrains

4.1.Assumptions

Funky Torrents is designed for downloading electronic publications provided by The Group. Library customers will be prompted with various selections of available publications.

4.2.Constrains

4.2.1. Hardware

Disk space: Electronic publications file size on hard-drive space is required by library customer. Network communication: Working internet connection.

4.2.2. Software:

Working web browser – Firefox, Google Chrome, Internet Explorer or Chromium
Erlang platform – Funky Torrents VM

4.2.3. System:

User log in and password is stored in Chicago-boss' database, that is a part of Chicago-boss framework.

4.2.4. Performance:

1 peer shall always be available with uploading the complete file.
Chicago- boss server is always up and running.

4.2.5. Security:

The Funky Torrents' publications are only available for registered users. Users login are handed out by the administrator.

5. Verification and validation

5.1.Log in:

Customer will be prompted with an option to log in to be able to download publications.

5.2.Torrent files:

Funky Torrents' torrent files publications which are to be produced by The Group and are adjusted to Funky Torrents software.

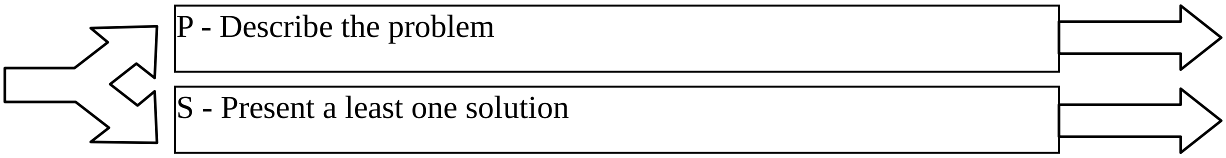
6. Design methodology

The Groups design process contains of smaller parallel running tasks.

Another of the techniques used in the design process is brainstorming. By using group meetings we present the problems to unaware group members and then ask them to come up with solutions, afterwards together The Group go through a more applicable solution.

The Group focused most on tasks that where easier to accomplish during the development process, but The Group will not neglect more complex tasks.

We starting with what final design will look like and discuss what the problems might be.



P – Problem

S – Solution

Both process evolve during the design process. When all necessary data is gathered, The Group discovers what they actually have to work with and adjust the design to what is done. The design and the final product or a part of the system need to look like.

7. Risks and Volatile Areas

7.1.Corrupted BitTorrent file:

Due to various factors that may occur in creating a bit torrent file, Funky Torrents' developing team decided that software will result in an error and crash.

7.2.Lacking of disc storage space:

Since Funky Torrents publications are containing with small amount file size the disk storage availability is set at library customer part.

7.3.No available internet connection:

No internet connection will result error message and software error and crash.

7.4.Not suitable torrent file

Funky Torrent can handle with specific data inside of torrent file, usage of not supported data will result in error message and software crash.

7.5.No available seeders

With no available seeders Funky Torrent will result in error message and software crash.

8. Goals and Priorities

8.1.Goals

The goal of Funky Torrents is to deliver a functional bit torrent client that is easy for the library customers to operate. Main functionality of Funky Torrents is to be able to receive electronic publications and store them on the hard disk drive.

8.2.Priories:

- Robustness, that software does not crash under normal usage.
- Availability, since BitTorrent is a continuous process any breakdown will not occur until the library customer cancels the process itself.
- Modifiability, due to software is build in modules there is an option for extend it for further implementation.

9. Architecture

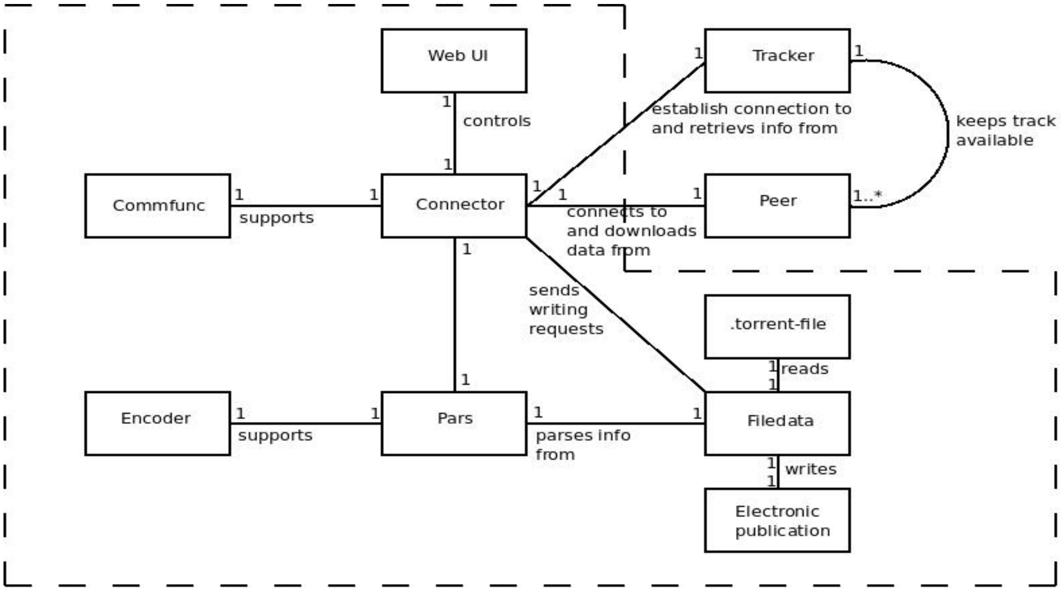


Figure 1. Funky Torrents software architecture with external interfaces

10. User Interfaces

User interface is web base interface which is using ChicagoBoss web framework for all of the functionalities

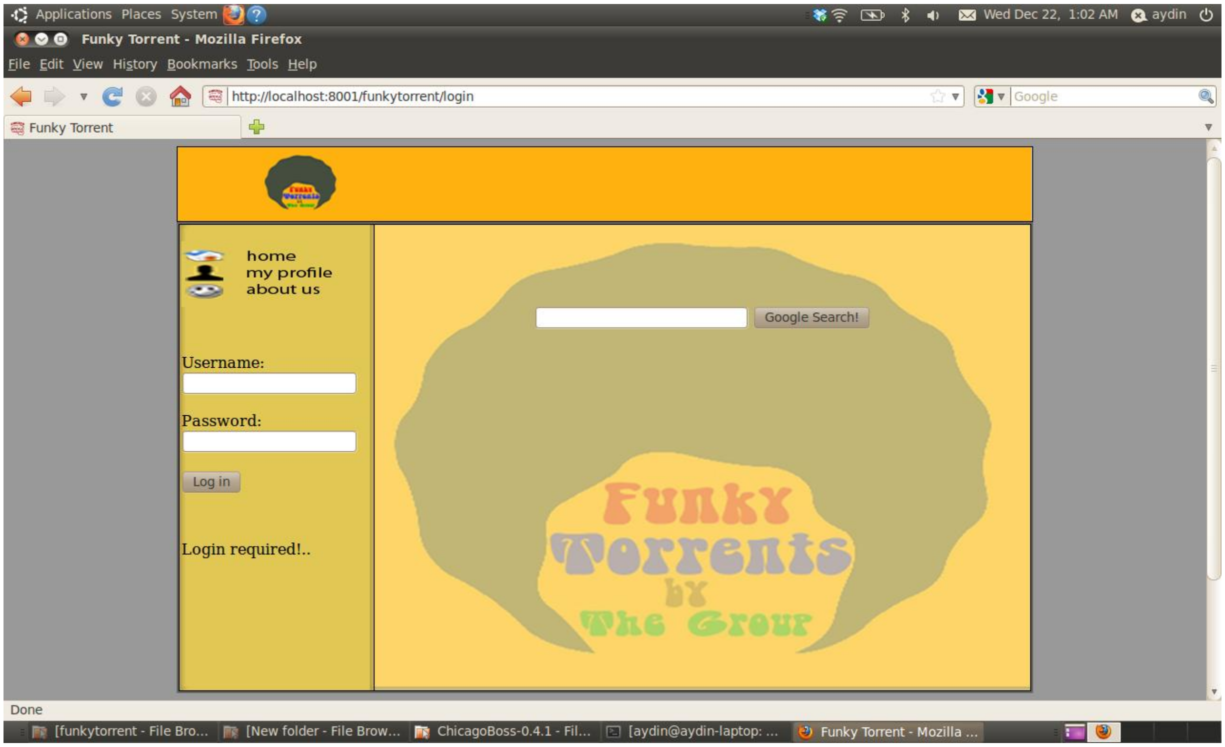

Applications Places System
Wed Dec 22, 1:02 AM
aydin
Funky Torrent - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8001/funkytorrent/login
Google
Funky Torrent
home
my profile
about us
Google Search!
Username:
Password:
Log in
Login required!..
Funky Torrents by The Group
Done
[funkytorrent - File Bro...
[New folder - File Brow...
ChicagoBoss-0.4.1 - Fil...
[aydin@aydin-laptop: ...
Funky Torrent - Mozilla ...

Figure 2. Screenshot of main page