Torrent Notifier

A Minor Project Submitted to

Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal

Towards Partial Fulfillment for the Award of

Bachelor of Engineering

(Computer Science and Engineering)

Submitted By

Siddharth Sharma (0827CS121164)

Siddharth Yadav (0827CS121165)

Under the Supervision of

Mr. Rahul Patel

Assistant Professor

Computer Science and Engineering Department

Department of Computer Science and Engineering

Acropolis Institute of Technology and Research, Indore

January-June 2015

Acropolis Institute of Technology and Research, Indore

Department of Computer Science and Engineering

RECOMMENDATION

The Project entitled "**Torrent Notifier"** submitted by **Mr. Siddharth Sharma (0827CS121164) , Mr. Siddharth Yadav (0827CS121165)** is satisfactory on account of the bonafide work done under our supervision and is recommended towards partial fulfillment for the award of **Bachelors of Engineering** (Computer Science and Engineering) degree by **Rajiv Gandhi Proudyogiki Vishwavidyalaya**, Bhopal.

Date:

Mr. Rahul Patel

Project Guide

Computer Science & Engineering Department

Acropolis Institute of Technology & Research, Indore

Ms. Kavita Namdev

Project Coordinator

Computer Science & Engineering Department

TORRENT NOTIFIER

Acropolis Institute of Technology & Research, Indore

Prof. Sanjay Bansal

Professor & Head

Computer Science & Engineering Department

Acropolis Institute of Technology & Research, Indore

Acropolis Institute of Technology and Research, Indore

Department of Computer Science and Engineering

CERTIFICATE

The project entitled "Torrent Notifier" **submitted by** Siddharth Sharma **(0827CS121164),** Siddharth Yadav **(0827CS121165) has been examined by us and is hereby approved for the award of degree** Bachelor of Engineering **in** Computer Science and Engineering **discipline, for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.**

Internal Examiner	**External Examiner**
Date: **18-May-2015**	Date: **19-May-2015**

Dr. Kamal Bharani

Principal

AITR, Indore

Acropolis Institute of Technology and Research, Indore

Department of Computer Science and Engineering

PROJECT APPROVAL SHEET

The project work entitled "Torrent Notifier" **submitted by** Siddharth Sharma **(0827CS121164),** Siddharth Yadav **(0827CS121165) is approved as partial fulfilment for the award of the** Bachelor of Engineering (Computer Science and Engineering) **degree by** Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal (M.P).

Date: 18-May-2015

Mr. Rahul Patel
Project Guide
CSE Dept., AITR Indore.

Ms. Kavita Namdeo
Project Coordinator
CSE Dept., AITR Indore

Acropolis Institute of Technology and Research, Indore

Department of Computer Science and Engineering

STUDENT DECLARATION

We the student of **Bachelors of Engineering** (Computer Science and Engineering) 2013, hereby declare that the work presented in this project synopsis entitled “Torrent Notifier” submitted towards completion of Minor Project in 6th semester of B.E. (Computer Science) at Acropolis Institute of Technology & Research, Indore, is an authentic record of our own work. Due acknowledge have been made in the text to all other material used. The project was done in full compliance with the requirement and constraints of the prescribed curriculum.

Date: 16-May-2014

Siddharth Sharma
(0827CS121164)

Siddharth Yadav
(0827CS121165)

ACKNOWLEDGEMENT

We would like to avail this opportunity to express our sincere thanks to all those who helped us in making this project. Even a most vivid collection of words, yield to express our heart fully thank towards one and all to have successfully assisted us in our expenditure of carrying out this project.

We wish to express our deep sense of gratitude to **H.O.D Prof. Sanjay Bansal**, our project coordinators **Ms. Kavita Namdeo, Mr. Rahul Patel** and the whole faculty members of the department of Computer Science for encouraging and giving moral support, not only regarding this project but also throughout our studies at this institute. Also, we express our gratitude to all my fellow classmates, friends and well wishers for their support and cooperation towards us

Siddharth Sharma

(0827CS121164)

Siddharth Yadav

(0827CS121165)

TABLE OF CONTENTS

LIST OF FIGURES

TORRENT NOTIFIER

Introduction

Rationale

The purpose of this document is to give a detailed description of the requirements for the “Torrent Notifier” software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended for approved and as a reference for developing the first version of the system for the development team.

Problem Definition

The “Torrent Notifier” is a utility desktop application which helps people find torrents of their desired content based on time uploaded and other specification like category, size, seeds and more. Its main purpose is to be able to set reminders as to when a new torrent gets uploaded, based on the user preferences given. When found, the user will be notified through a popup notification, and other means, such as e-mail. The application should be free to download from a code hosting website: Github, or a similar service.

Torrent websites provide their hosted torrent information using the website and APIs. This information will act as basis for the search results displayed to the user. There is no administrator, as the application is stand-alone, therefore for delivery of secure content, the software will be ensured to use secure and verifiable sources by developers.

Furthermore, the software needs both persistent Internet connection, and background processing to fetch and display results at regular intervals of time. All system information is maintained in a database, which is located locally. The software also interacts with a web browser, and a torrent downloader software which are required to be already installed application on the user's computer. By using browser, users can view desired torrent download links, its description and user comments on the respected torrent website. By using torrent downloader software, users can download the torrent content. The application also has the capability to directly display torrent download links, in it. The application does not have the ability to download torrent content.

TORRENT NOTIFIER

The goal of the notification feature aims to cut the time delay of when the content gets posted online and when the user discovers it. When the application is running, it automatically alerts the user, as soon as the torrent gets uploaded.

Process Flow of Existing System

There are certain online sources available in order to provide information about health related data. They focus on exhibiting details about medicines, human diseases and other useful knowledge regarding healthcare though their idea of demonstrating the entire data is a bit complicated, as the users need simple and easy access to the services. With the increase in options and facilities available on the website, it also becomes a complex task for a normal user to find the desired information.

Limitations of Existing Systems

- The major limitation in the existing systems is lack of user-friendliness. Normal users need all the basic services at one place yet the graphical user interface should be simple for them to understand and use.
- Searching torrents manually can be time consuming. Also following up & interacting with the torrent sites can be made easier with the automation based technologies.
- Finding good torrents at a particular site is an important requirement. Locating them on a software will also make the task easier.
- Simply visiting a torrent site & getting a torrent information is not enough. A torrent user may want to know more about the torrent. Also before visiting a torrent site, on the basis of any constraint (seeders, leechers or time) a user may want to know what content he/she wants.

Proposed Solution

The following features are added to overcome the problems stated above:

- A single place where user can access and view the torrent information with a user friendly interface.
- The application is made cross platform. There is no bounding of the Operating System the user is using. Any user (OS-X, Windows or linux) can use the application, the only thing he/she needs is the python runtime environment for running the application.

- The user will be able to see all the torrent names in a listed format, thus he/she doesn't have to use a browser for getting the torrent list.
- User can directly view the description of the torrent he/she wants to view by just clicking a button on the main window of the application which redirects the user to the description page of the torrent in a browser.
- User also has an option to create filters (torrents added for the notification purpose). Application has some key features like CREATE, PAUSE, RESUME and DELETE a filter. User can also view whether a filter is active or not.

Report Organization

Our project report consist of Literature Survey: In literature survey we have described different technologies for designing our desktop application and differentiated between them. And also we have described methodology of our project.

System Analysis: In system analysis we have described our projects requirement analysis, object oriented analysis, feasibility analysis and development method.

System Design: In system design we have shown our Use Case Diagram, Use Case Specification, Activity Diagram and Sequence Diagram.

Project Implementation: In this we have shown all the commands which were executed by us.

Test Case Design: This section consist of our projects Test Approach, Test Plan, Features to be tested, Features not to be tested and TEST CASES.

Output Screens: This section consist of Screenshots of outputs of executed commands.

Conclusion: This section consists of Inferences Drawn, Future Extensions, Scope and limitation.

2 Literature Survey

2.1 Python

Python is a widely used general-purpose, high-level programming language.[20][21][22] Its design philosophy emphasizes codereadability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such asC++ or Java.[23][24] The language provides constructs intended to enable clear programs on both a small and large scale.[25]

Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.[26]

Python interpreters are available for installation on many operating systems, allowing Python code execution on a wide variety of systems. Using third-party tools, such as Py2exe or Pyinstaller,[27] Python code can be packaged into stand-alone executable programs for some of the most popular operating systems, allowing for the distribution of Python-based software for use on those environments without requiring the installation of a Python interpreter.

CPython, the reference implementation of Python, is free and open-source software and has a community-based development model, as do nearly all of its alternative implementations. CPython is managed by the non-profit Python Software Foundation.

2.1.1 Indentation

Python uses whitespace indentation, rather than curly braces or keywords, to delimit blocks; this feature is also termed the off-side rule. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block.

2.1.2 Statements and control flow

Python's statements include (among others):

- The `if` statement, which conditionally executes a block of code, along with `else` and `elif` (a contraction of else-if).
- The `for` statement, which iterates over an iterable object, capturing each element to a local variable for use by the attached block.
- The `while` statement, which executes a block of code as long as its condition is true.
- The `try` statement, which allows exceptions raised in its attached code block to be caught and handled by `except` clauses; it also ensures that clean-up code in a `finally` block will always be run regardless of how the block exits.
- The `class` statement, which executes a block of code and attaches its local namespace to a class, for use in object-oriented programming.
- The `def` statement, which defines a function or method.
- The `with` statement (from Python 2.5), which encloses a code block within a context manager (for example, acquiring a lock before the block of code is run and releasing the lock afterwards, or opening a file and then closing it), allowing RAII-like behavior.
- The `pass` statement, which serves as a NOP. It is syntactically needed to create an empty code block.
- The `assert` statement, used during debugging to check for conditions that ought to apply.
- The `yield` statement, which returns a value from a generator function. From Python 2.5, `yield` is also an operator. This form is used to implement coroutines.
- The `import` statement, which is used to import modules whose functions or variables can be used in the current program.
- `print()` was changed to a function in Python 3.

Python does not support tail-call optimization or first-class continuations, and, according to Guido van Rossum, it never will. However, better support for coroutine-like functionality is provided in 2.5, by extending Python's generators. Prior to 2.5, generators were lazy iterators; information was passed unidirectionally out of the generator. As of Python 2.5, it is possible to pass information back into a generator function, and as of Python 3.3, the information can be passed through multiple stack levels.

2.1.1 Beautiful Soup

It is a Python package for parsing HTML and XML documents (including having malformed markup, i.e. non-closed tags, so named after Tag soup). It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for web scraping.[2]

It is available for Python 2.6+ and Python 3.

Code example

```
# anchor extraction from html document
from bs4 import BeautifulSoup
import urllib2

webpage = urllib2.urlopen('http://en.wikipedia.org/wiki/Main_Page')
soup = BeautifulSoup(webpage)
for anchor in soup.find_all('a'):
    print(anchor.get('href', '/'))
```

2.2 PyQt (Gui Toolkit)

PyQt is a Python binding of the cross-platform GUI toolkit Qt. It is one of Python's options for GUI programming. Popular alternatives are PySide (the Qt binding with official support and a more liberal licence), PyGTK, wxPython, and Tkinter (which is bundled with Python). Like Qt, PyQt is free software. PyQt is implemented as a Python plug-in.

PyQt is developed by the British firm Riverbank Computing. It is available under similar terms to Qt versions older than 4.5; this means a variety of licenses including GNU General Public License (GPL) and commercial license, but not the GNU Lesser General Public License (LGPL). PyQt supports Microsoft Windows as well as various flavours of Unix, including Linux and OS X.

PyQt implements around 440 classes and over 6,000 functions and methods including:

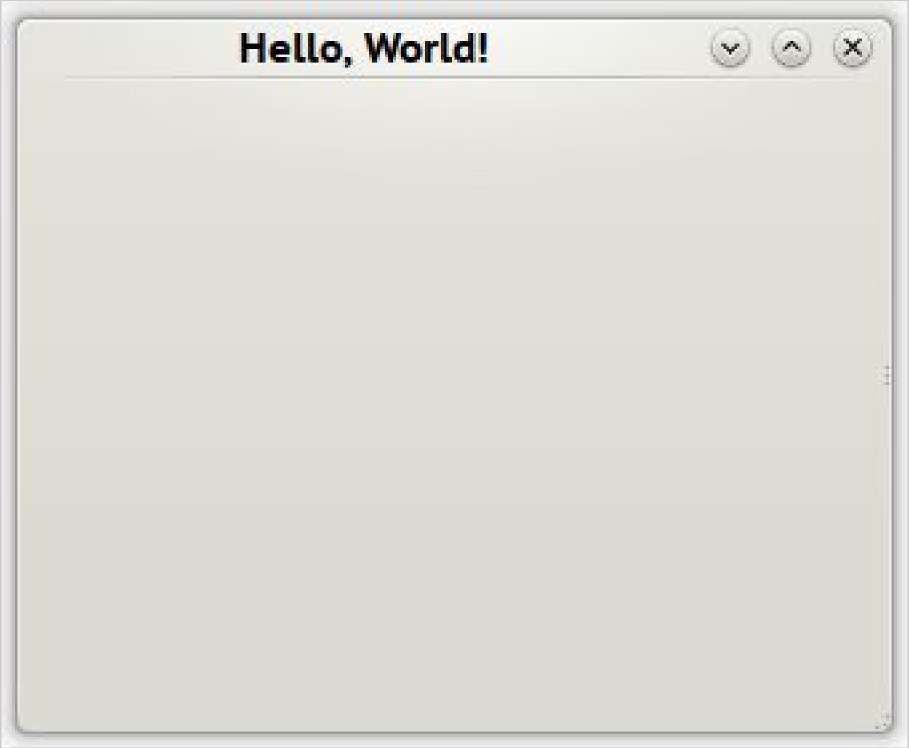
- a substantial set of GUI widgets
- classes for accessing SQL databases (ODBC, MySQL, PostgreSQL, Oracle, SQLite)
- QScintilla, Scintilla-based rich text editor widget
- data aware widgets that are automatically populated from a database
- an XML parser
- SVG support
- classes for embedding ActiveX controls on Windows (only in commercial version)

To automatically generate these bindings, Phil Thompson developed the tool SIP, which is also used in other projects.

In August 2009, Nokia, the then owners of the Qt toolkit, released PySide, providing similar functionality, but under the LGPL,[8]after failing to reach an agreement with Riverbank Computing to change its licensing terms to include LGPL as an alternative license.

2.2.1 Hello world example

The below code shows a small window on the screen.



The result in KDE

```
#! /usr/bin/env python
# -*- coding: utf-8 -*-
#
# Here we provide the necessary imports.
# The basic GUI widgets are located in QtGui module.
import sys
from PyQt4.QtGui import *

# Every PyQt4 application must create an application object.
# The application object is located in the QtGui module.
a = QApplication(sys.argv)
```

```
# The QWidget widget is the base class of all user interface objects in PyQt4.
# We provide the default constructor for QWidget. The default constructor has no parent.
# A widget with no parent is called a window.
w = QWidget()

w.resize(320, 240)  # The resize() method resizes the widget.
w.setWindowTitle("Hello, World!")  # Here we set the title for our window.
w.show()  # The show() method displays the widget on the screen.

sys.exit(a.exec_())  # Finally, we enter the mainloop of the application.
```

2.1.2 PyQt main components

PyQt4 contains the following Python modules.

- The *QtCore* module contains the core non-GUI classes, including the event loop and Qt's signal and slot mechanism. It also includes platform independent abstractions for Unicode, threads, mapped files, shared memory, regular expressions, and user and application settings.
- The *QtGui* module contains the majority of the GUI classes. These include a number of table, tree and list classes based on the model–view–controller design pattern. Also provided is a sophisticated 2D canvas widget capable of storing thousands of items including ordinary widgets.
- The *QtNetwork* module contains classes for writing UDP and TCP clients and servers. It includes classes that implement FTP and HTTP clients and support DNS lookups. Network events are integrated with the event loop making it very easy to develop networked applications.
- The *QtOpenGL* module contains classes that enable the use of OpenGL in rendering 3D graphics in PyQt applications.
- The *QtSql* module contains classes that integrate with open-source and proprietary SQL databases. It includes editable data models for database tables that can be used with GUI classes. It also includes an implementation of SQLite.
- The *QtSvg* module contains classes for displaying the contents of SVG files. It supports the static features of SVG 1.2 Tiny.
- The *QtXml* module implements SAX and DOM interfaces to Qt's XML parser.
- The *QtMultimedia* module implements low-level multimedia functionality. Application developers would normally use the phonon module.

- The *QtDesigner* module contains classes that allow Qt Designer to be extended using PyQt.
- The *Qt* module consolidates the classes contained in all of the modules described above into a single module. This has the advantage that you don't have to worry about which underlying module contains a particular class. It has the disadvantage that it loads the whole of the Qt framework, thereby increasing the memory footprint of an application. Whether you use this consolidated module, or the individual component modules is down to personal taste.
- The *uic* module implements support for handling the XML files created by Qt Designer that describe the whole or part of a graphical user interface. It includes classes that load an XML file and render it directly, and classes that generate Python code from an XML file for later execution

2.3 XML

Extensible Markup Language (**XML**) is a markup language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable. It is defined by the W3C's XML 1.0 Specification and by several other related specifications, all of which are free open standards.

The design goals of XML emphasize simplicity, generality and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures such as those used in web services.

Several schema systems exist to aid in the definition of XML-based languages, while many application programming interfaces(APIs) have been developed to aid the processing of XML data.

2.3.1 Applications of XML

As of 2009, hundreds of document formats using XML syntax have been developed, including RSS, Atom, SOAP, and XHTML. XML-based formats have become the default for many office-productivity tools, including Microsoft Office (Office Open XML), OpenOffice.org and LibreOffice (OpenDocument), and Apple's iWork. XML has also been employed as the base language for communication protocols, such as XMPP. Applications for the Microsoft .NET Framework use XML files for configuration. Apple has an implementation of a registry based on XML.

XML has come into common use for the interchange of data over the Internet. IETF RFC 7303 gives rules for the construction of Internet Media Types for use when sending XML. It also defines the media types *application/xml* and *text/xml*, which say only that the data is in XML, and nothing about its semantics. The use of *text/xml* has been criticized as a potential source of encoding problems and it has been suggested that it should be deprecated.

RFC 7303 also recommends that XML-based languages be given media types ending in *+xml*; for example *image/svg+xml* for SVG.

Further guidelines for the use of XML in a networked context may be found in RFC 3470, also known as IETF BCP 70, a document covering many aspects of designing and deploying an XML-based language.

2.3.1 XML example :

```
<item>

        <title>Interstellar trailer 3 4k UHD Ultra HD 4096x1716</title>

        <link>http://torrentz.in/03493c32a1fd0e73e4c7d7a50142f5ab07349fb0</link>

        <guid>http://torrentz.in/03493c32a1fd0e73e4c7d7a50142f5ab07349fb0</guid>

        <pubDate>Mon, 20 Oct 2014 04:00:27 +0000</pubDate>

        <category>movies</category>

        <description>Size: 635 MB Seeds: 28 Peers: 1 Hash:
        03493c32a1fd0e73e4c7d7a50142f5ab07349fb0</description>

        </item>
```

3 System Analysis

3.1 Feasibility Analysis

The project deals with providing the user with a desktop software to better access and discover Torrent files across the Internet. It has a unified interface for searching across various torrent sharing websites, with options to filter and categorize. Users create filters to get alerts when new torrents get uploaded, therefore not needing to search manually from time to time. The application searches periodically on the Internet matching for filter. There are various options on ways to get notified. The outcome is convenient for the user as it provides a single place for creating alerts.

The project is a Torrent Search and Notification Application implemented as a standalone GUI software for desktops. It is designed to be cross-platform. It provides the user with searching and creating various filters for future automated search and notification. It uses Python programming language, and PyQt GUI toolkit for implementation. The triggers used for notification are keywords that user provides.

3.2 Development Method

In our project we have used Python (programming language), PyQt5 (Gui Toolkit), XML data for parsing purpose. Python3 is a cross platform, runtime environment for server side and networking applications. In addition, we use Github which is a repository of files and data. It is a place where you can store your program code on cloud. Multiple users can access and manipulate the data at the same time which helps in managing the code easily.

For development of our project initially we’ve created the basic layout of our web application using PyQt5 which is responsible for GUI of the Application.

In order to install PyQt5 we’ve followed the following steps:

TORRENT NOTIFIER

1. Linux (Ubuntu and Debian-based)

For Debian- and Ubuntu-based Linux distributions, installation of PySide or PyQt is simple; just do:

```
sudo apt-get install python-pyside
```

from the command line. For PyQt:

```
sudo apt-get install python-qt4
```

Alternatively, use Synaptic to install your choice of `python-pyside` or `python-qt4`.

2. Linux (CentOS and RPM-based)

Installation of PySide or PyQt is also simple for most RPM-based distros using `yum`; just do:

```
yum install python-pyside pyside-tools
```

as root from the command line to install PySide. For PyQt, do

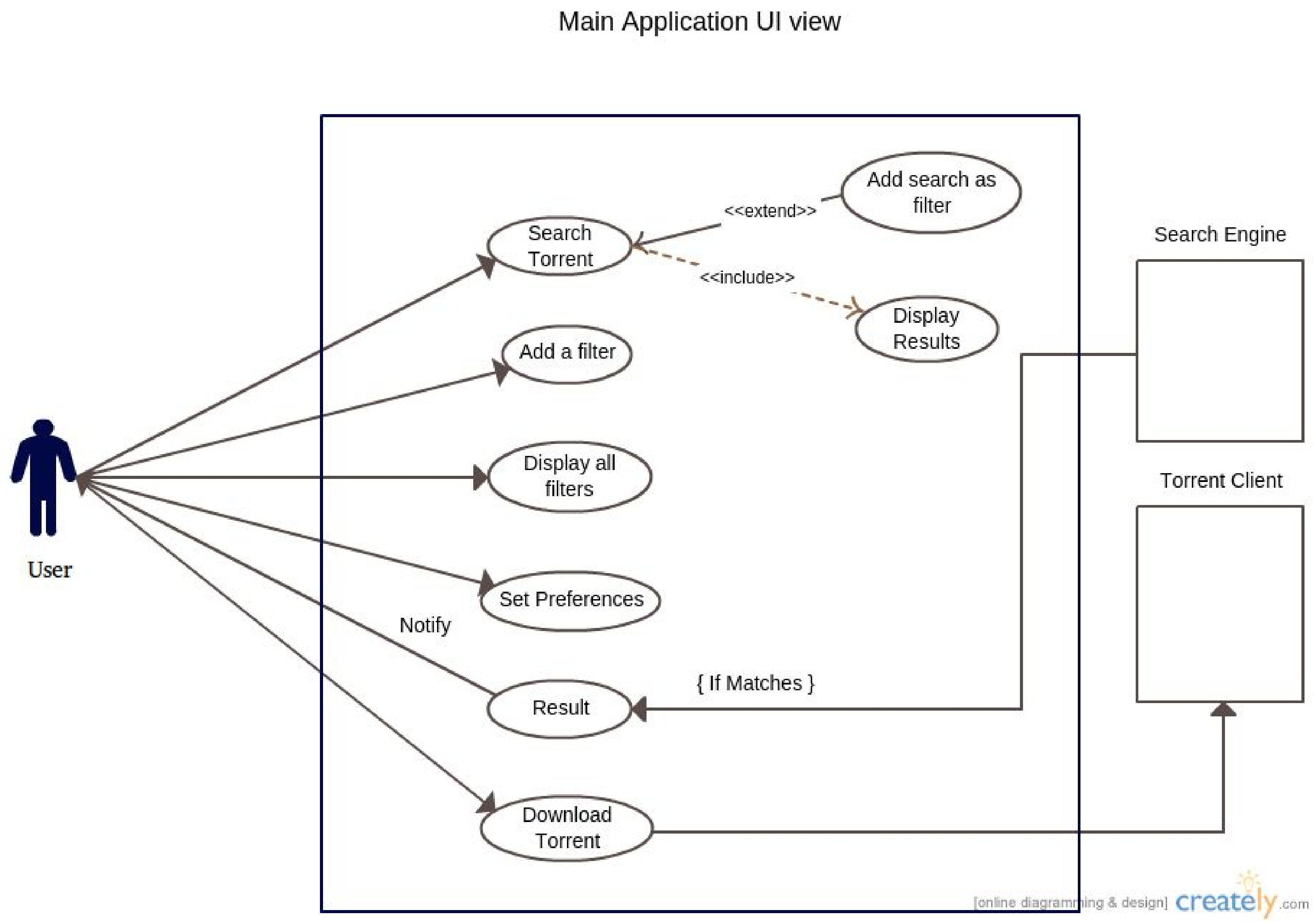
```
yum install PyQt4
```

Now that you have an installation of PySide or PyQt, we are almost ready to begin learning to use it — but first, we must discuss editors and IDEs. We'll do so in our next installment.

4. System Design

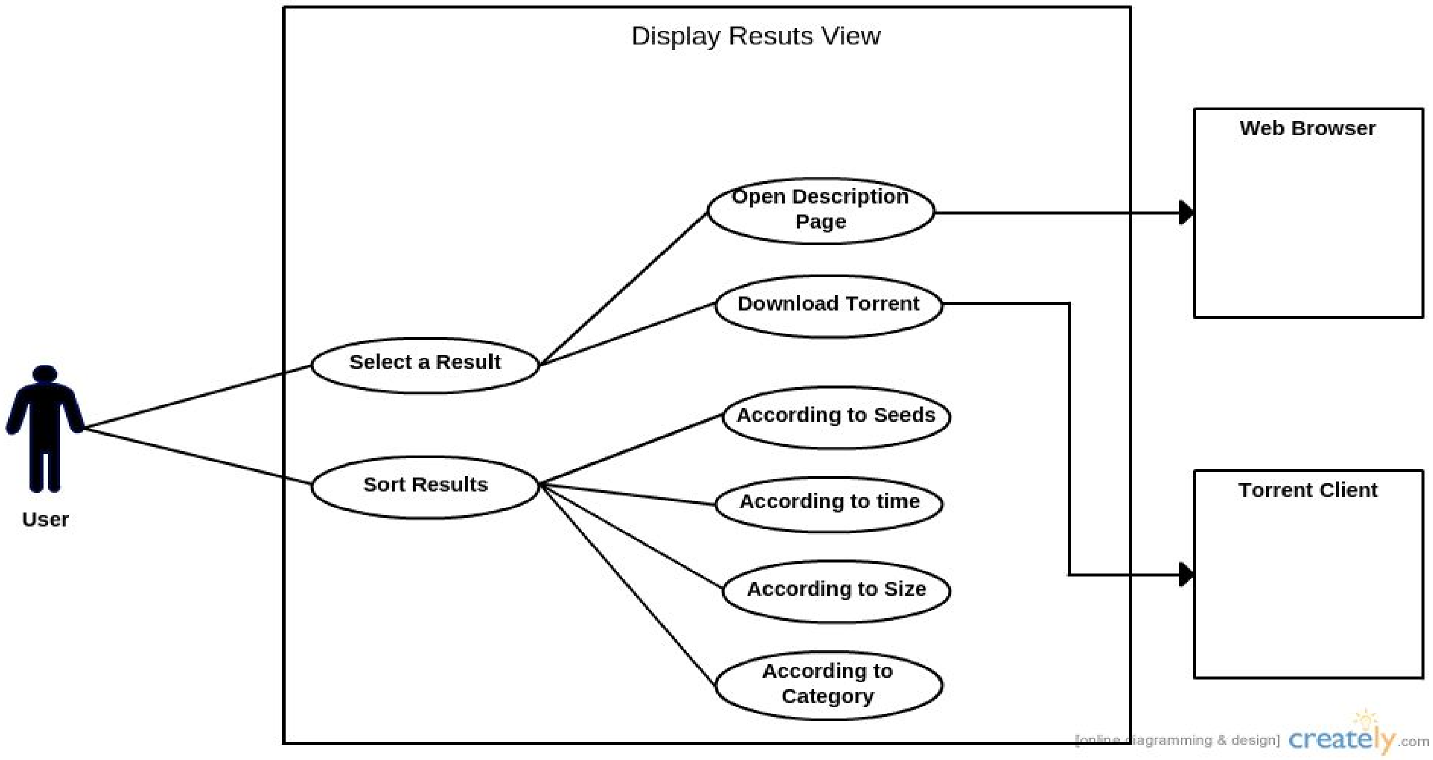
4.1 Use-case Diagram

4[1].1.1 Use-case 1:

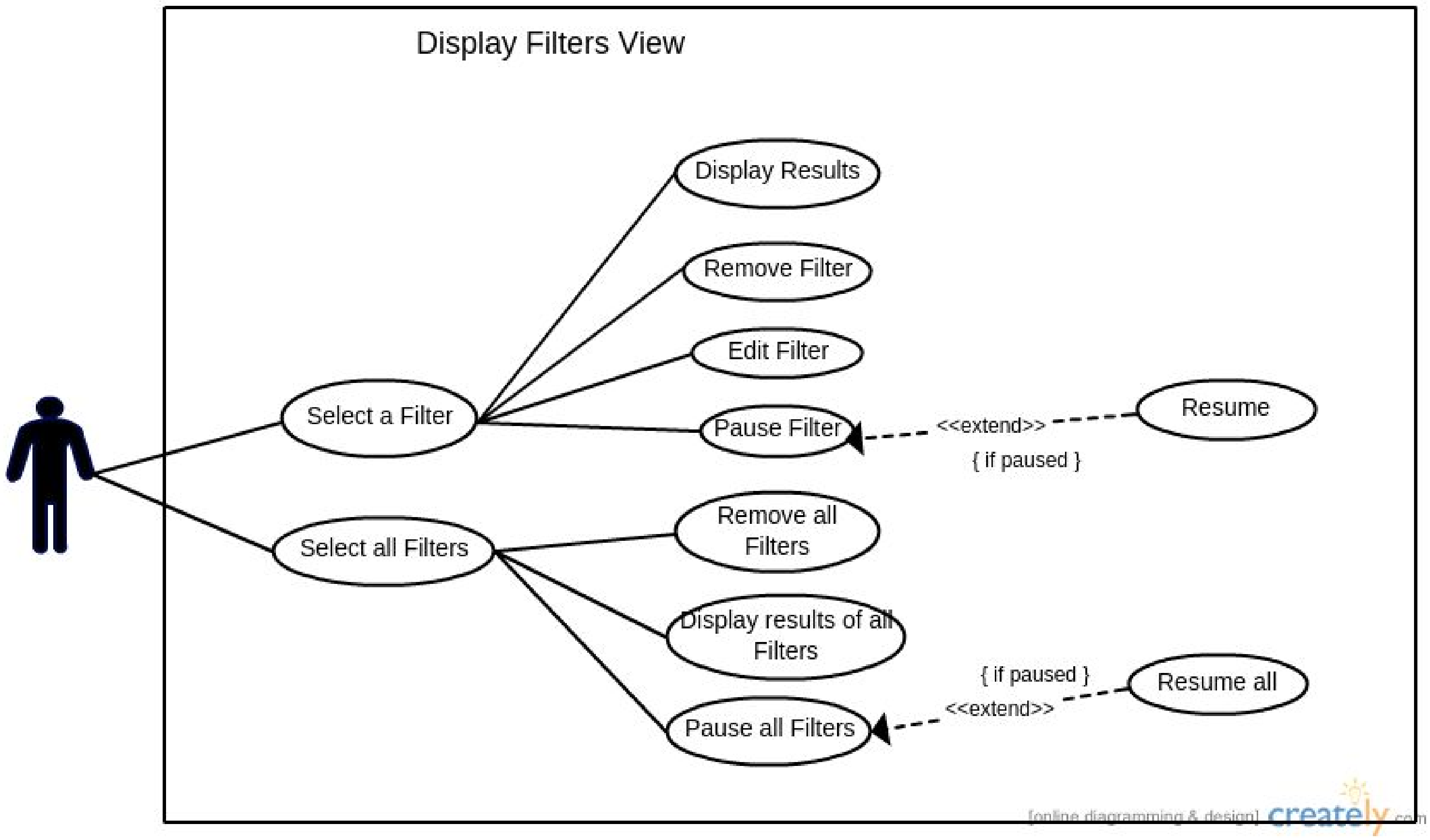


[1]

4.1.2 Use-case 2:

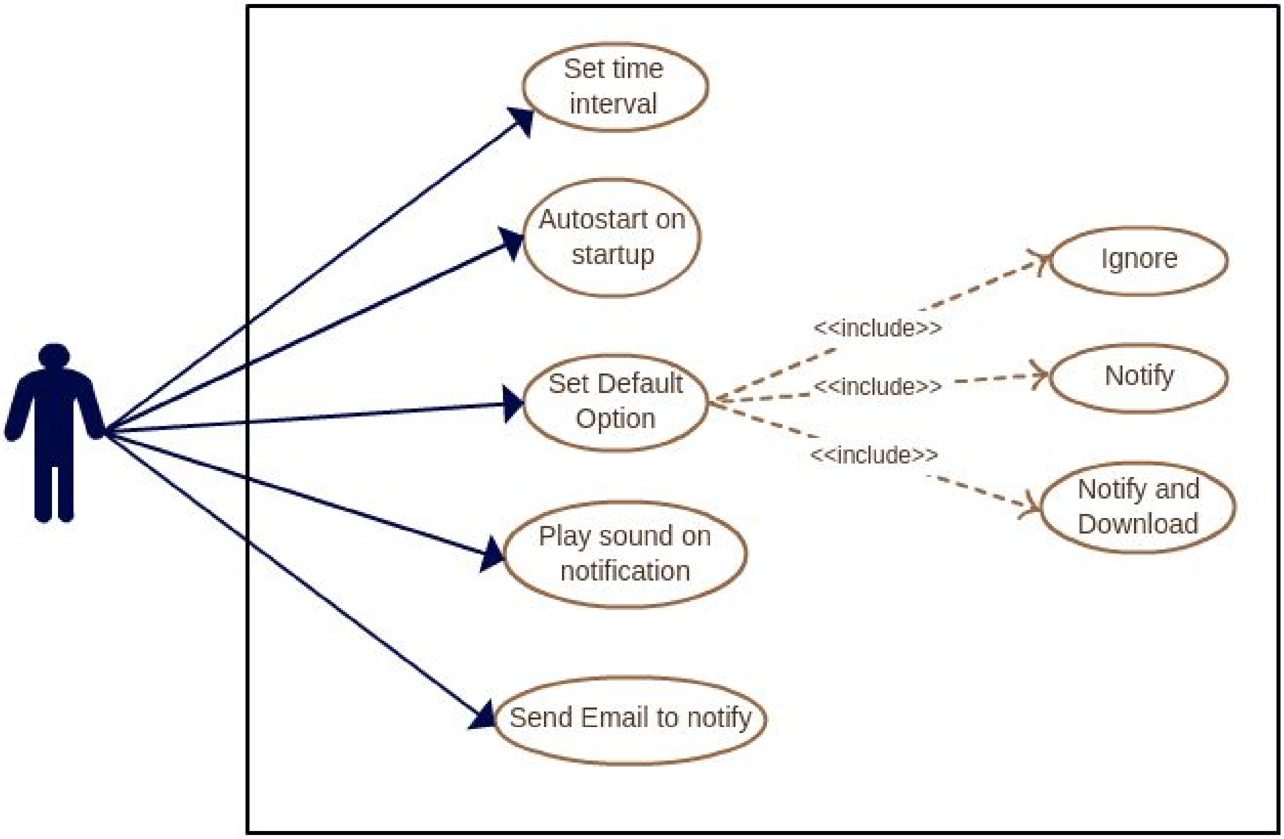


4.1.3 Use-case 3:

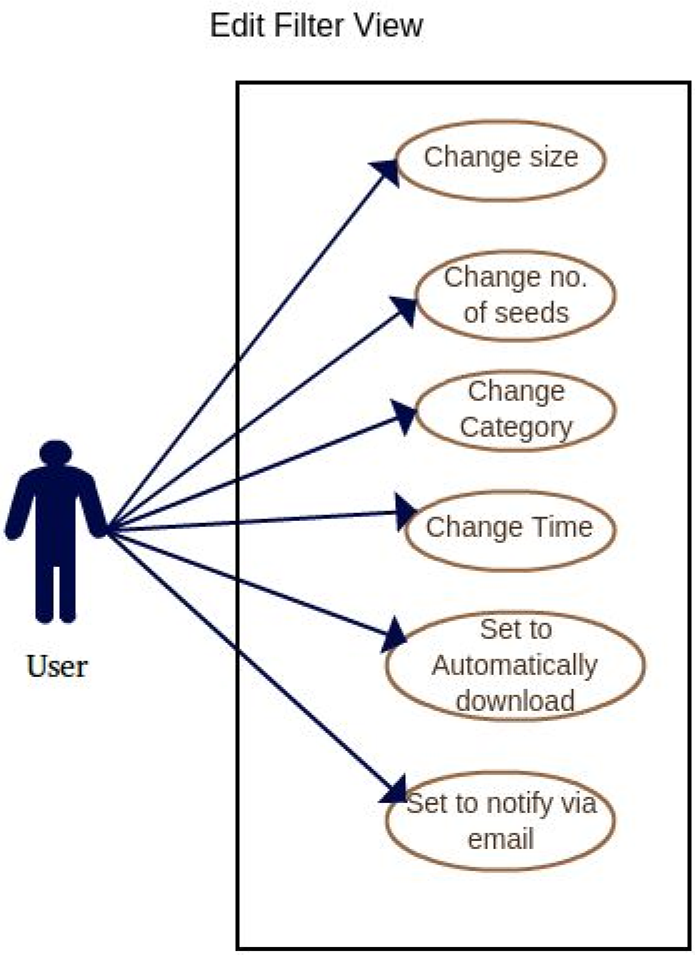


4.1.4 Use-case 4:

Set Preferences View



4.1.5 Use-case 5:



4.2 Use-case Specification

Actors:

- Torrent User
- Torrent Client
- Web Browser
- Search Engine

Use-cases:

1. Torrent User

    i. Search Torrent

        Add search as filter

        Display results

    ii. Add a filter

    iii. Display all filters

    iv. Set Preferences

        Set time interval

        Auto start on start-up

        Set default option

            Ignore

            Notify

            Notify and Download

        Play sound on notification

        Send email to notify

v. Notify

email notification only

pop-up notification only

pop-up and email notification both

vi. Download Torrent

2. Torrent Client

i. Add Torrent

3. Web Browser

i. Description of the Torrent

ii. Download Torrent

4. Search Engine

i. Gives the result in a list format.

4.3 Activity Diagram

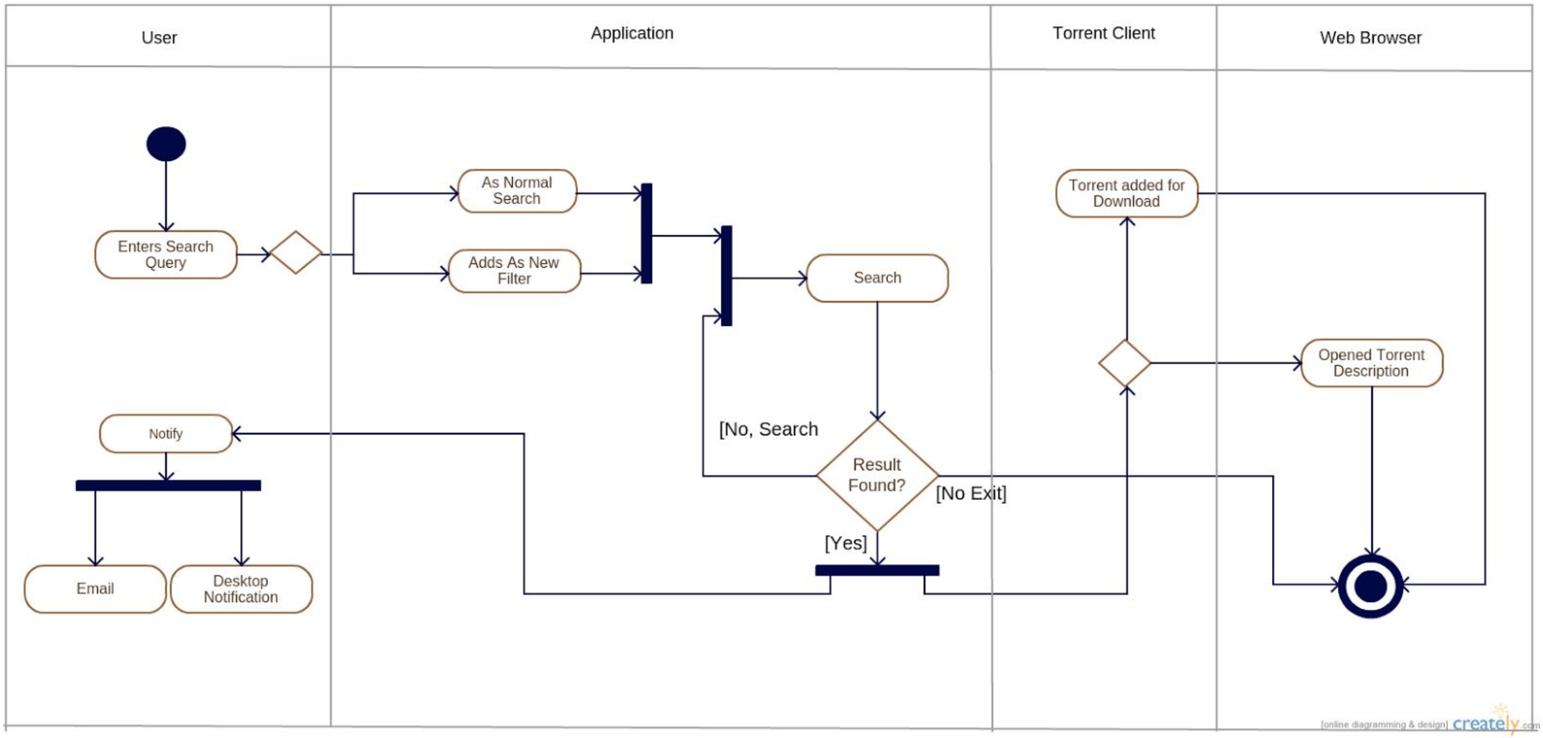
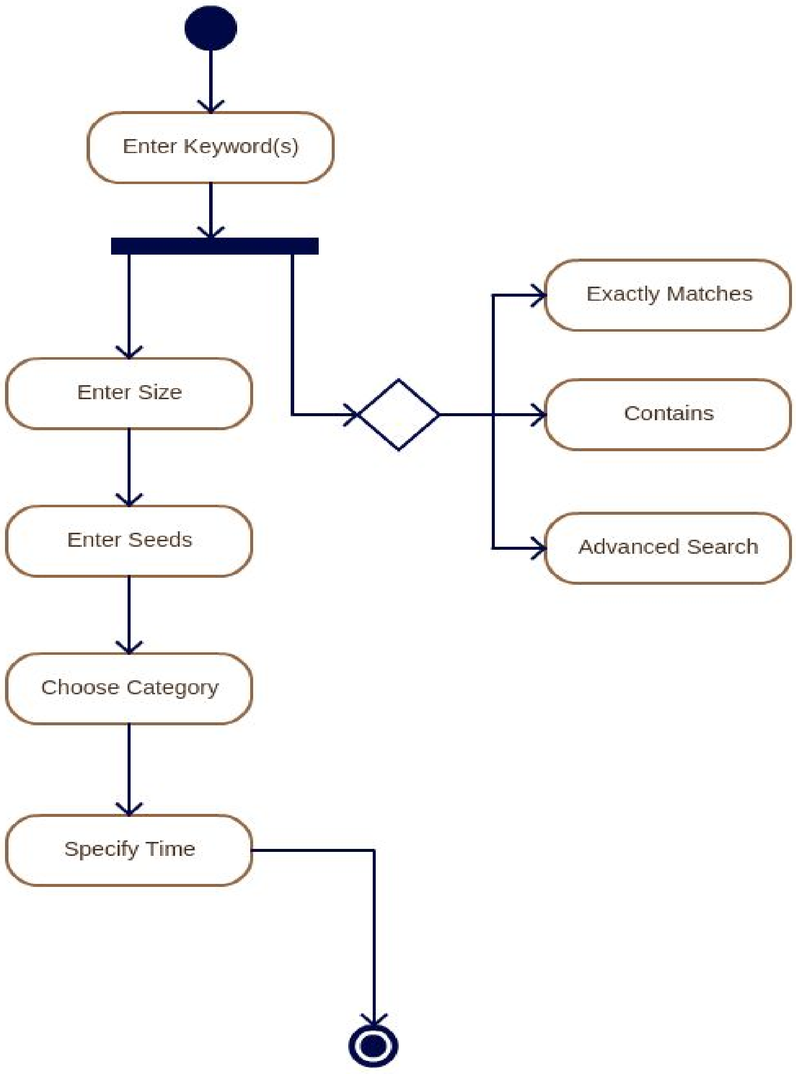


Figure 4.3.1 Activity Diagram



Create Filter Activity

Figure 4.3.2 Activity Diagram

4.4 Sequence Diagram

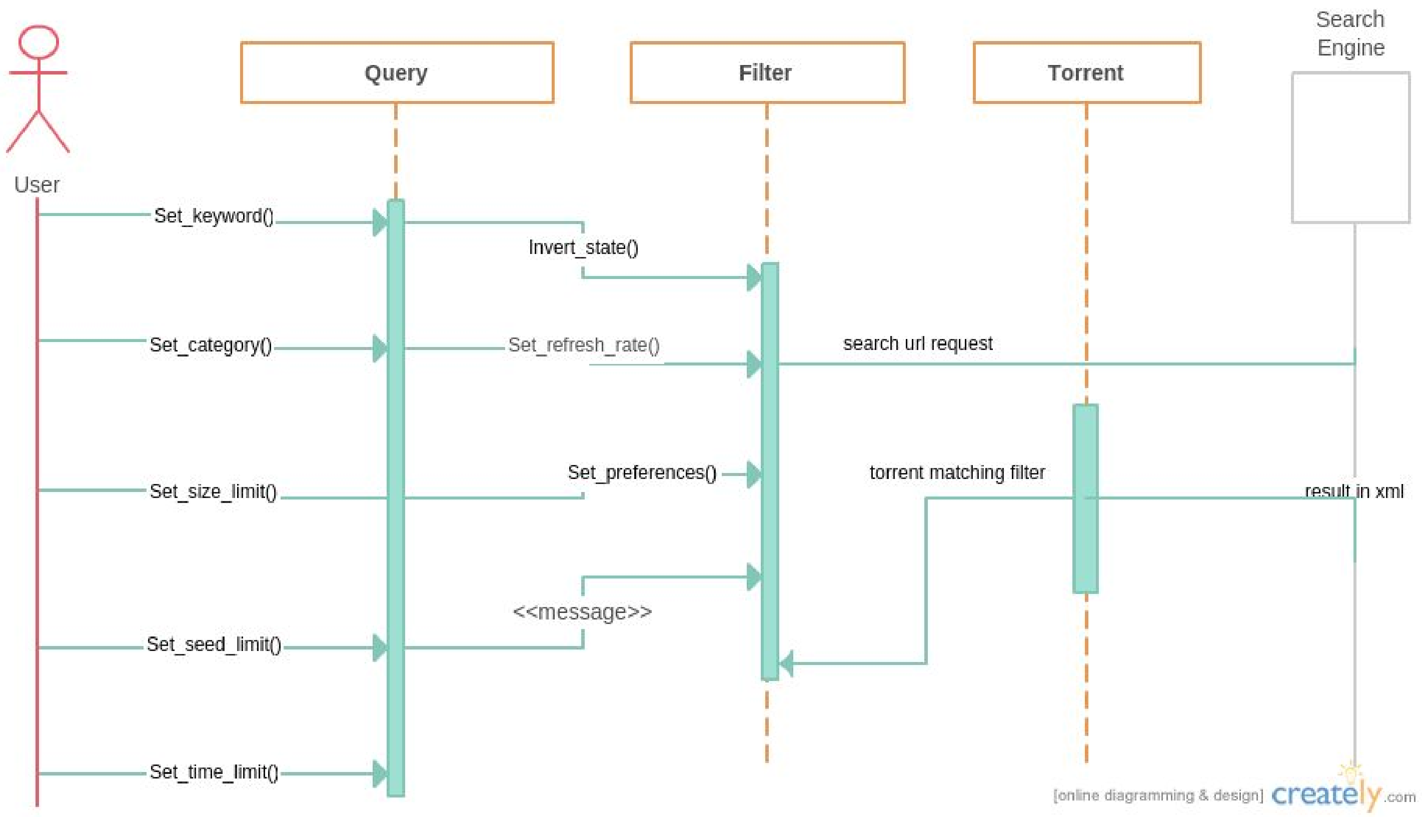


Figure 4.4.1: Sequence Diagram

5. Project Implementation

6.Test Case Design

6.1 Test Case Approach:-

In test case approach we test all the testable part of an application. In procedural programming a unit may be used an individual program, function, procedure etc., while in object oriented programming the smallest unit is a method, which may belong to a base/super class, abstract class or derived/child class. Initially test focus on each component individually, ensuring that it functions properly as a unit, hence the name unit testing. It makes heavier use of white box testing.

In the project module interface was tested to ensure that information properly flows in and out of the program unit i.e. each function under test for each individual unit. The local data structures were examined to ensure that the data stored temporarily maintains its integrity during all steps in an algorithm execution. Boundary conditions were also tested to ensure that module operates properly at boundaries established to restrict that all statements in the modules execute at least once. And last but not the least all error handling paths were tested.

6.2 Test Plan:-

In Test plan we test all the functionality of the website. So it includes following:

1. Search Engine API connectivity
2. Torrent client connectivity
3. Webpage connectivity

6.3 Features to be tested

We test the following features of our Application:

1. Create Filter
2. Advanced Search
3. Normal Search
4. Delete, Pause and Resume
5. Pop-up Notification
6. Email Notification
7. View Torrent description

6.4 TEST CASES

1. Create Filter

S.N	Test Case	Excepted Result	Test Result
1	Enter Title,size, category,seeds,age and click on “Create Filter” button.	App should display the created filter.	Successful
2	Enter invalid input (not fill all the input fields)	App should not display any result.	Successful

2. Advanced Search

S.N	Test Case	Excepted Result	Test Result
1	Enter set minimum seeds, added before, exact words, exact phrase, any of these, excluding & click on “OK” button	App should display the result satisfying the input constraints.	Successful
2	Enter invalid	App should not display any result.	Successful

3. Normal Search

S.N	Test Case	Excepted Result	Test Result
1	Enter filter or keywords & click on “Search” Button	App should display the related Torrent results by extracting data from the Torrent engine API.	Successful
2	Enter invalid Medicines	App should not display any result.	Successful

4. Delete, Pause and Resume

S.N	Test Case	Excepted Result	Test Result
1	Select the filter and click on “Delete” button.	App should delete the respected data from the filter list.	Successful
2	Select the filter and click on “Pause” button.	App should stop the searching of the torrent running in the background.	Successful
3	Select the filter (paused) and click on “Resume” button.	App should start searching for the torrent data.	Successful

5. Pop-up Notification

S.N	Test Case	Excepted Result	Test Result
1	Pop-up Window.	A pop-up should get displayed on the user's home screen notifying about the searched torrent.	Successful

6. Email Notification

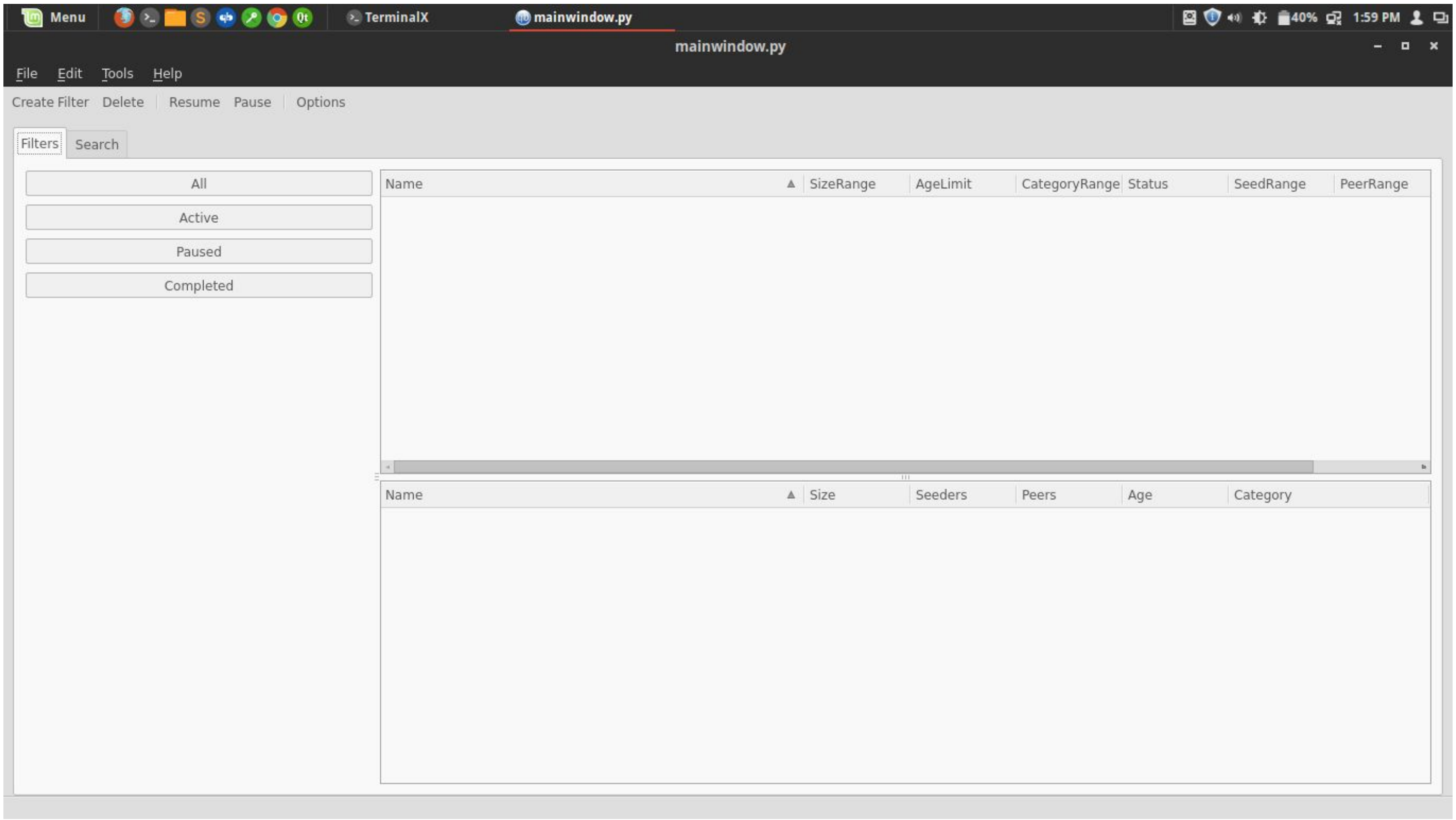
S.N	Test Case	Excepted Result	Test Result
1	Email Notification.	App will send an email to the user immediately after it finds the result.	Successful

7. View Torrent Description

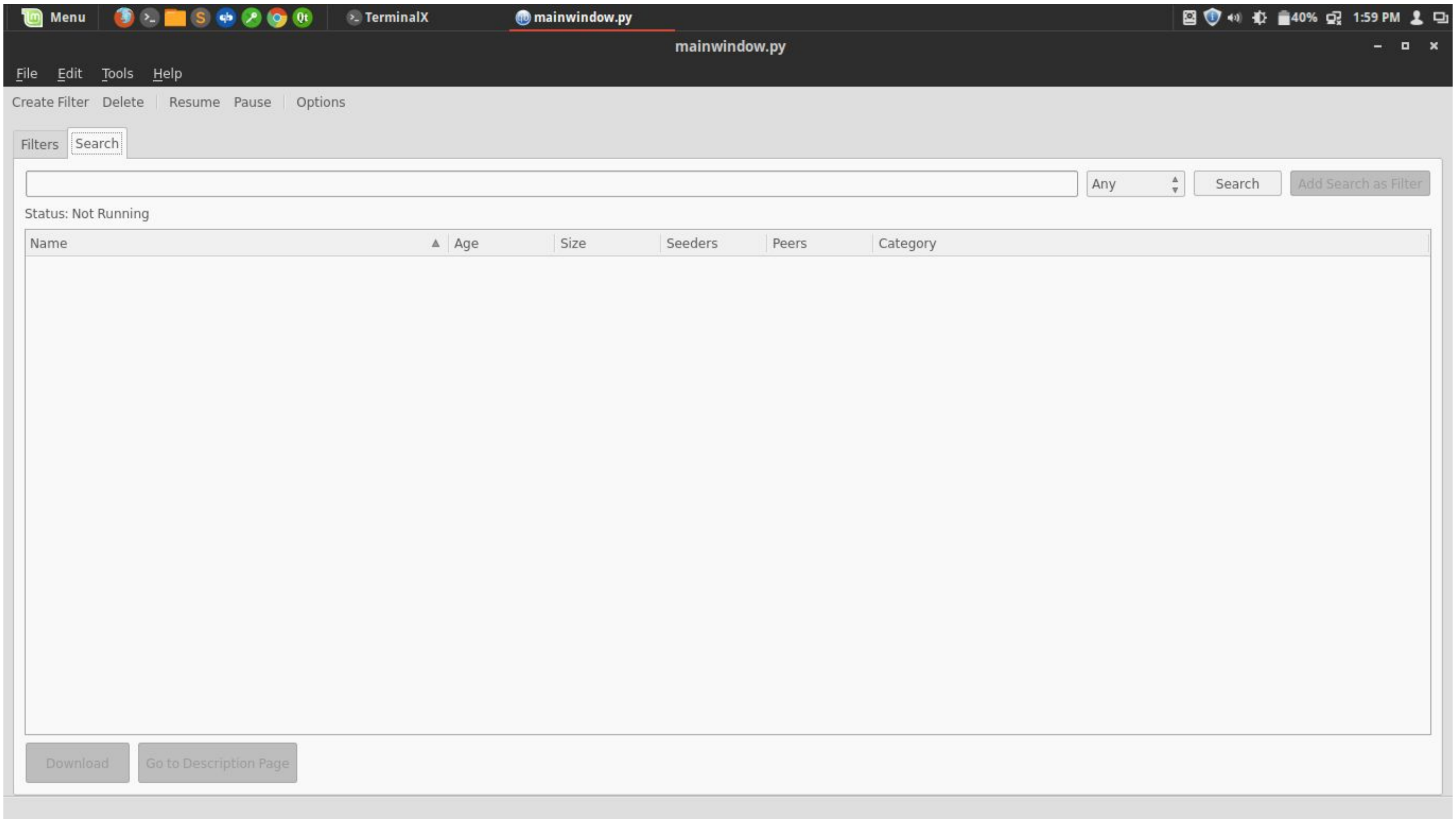
S.N	Test Case	Excepted Result	Test Result
1	Select the filter and click on "Go to Description page".	The torrent description will get opened in the browser.	Successful

Output Screens

Screenshot 1

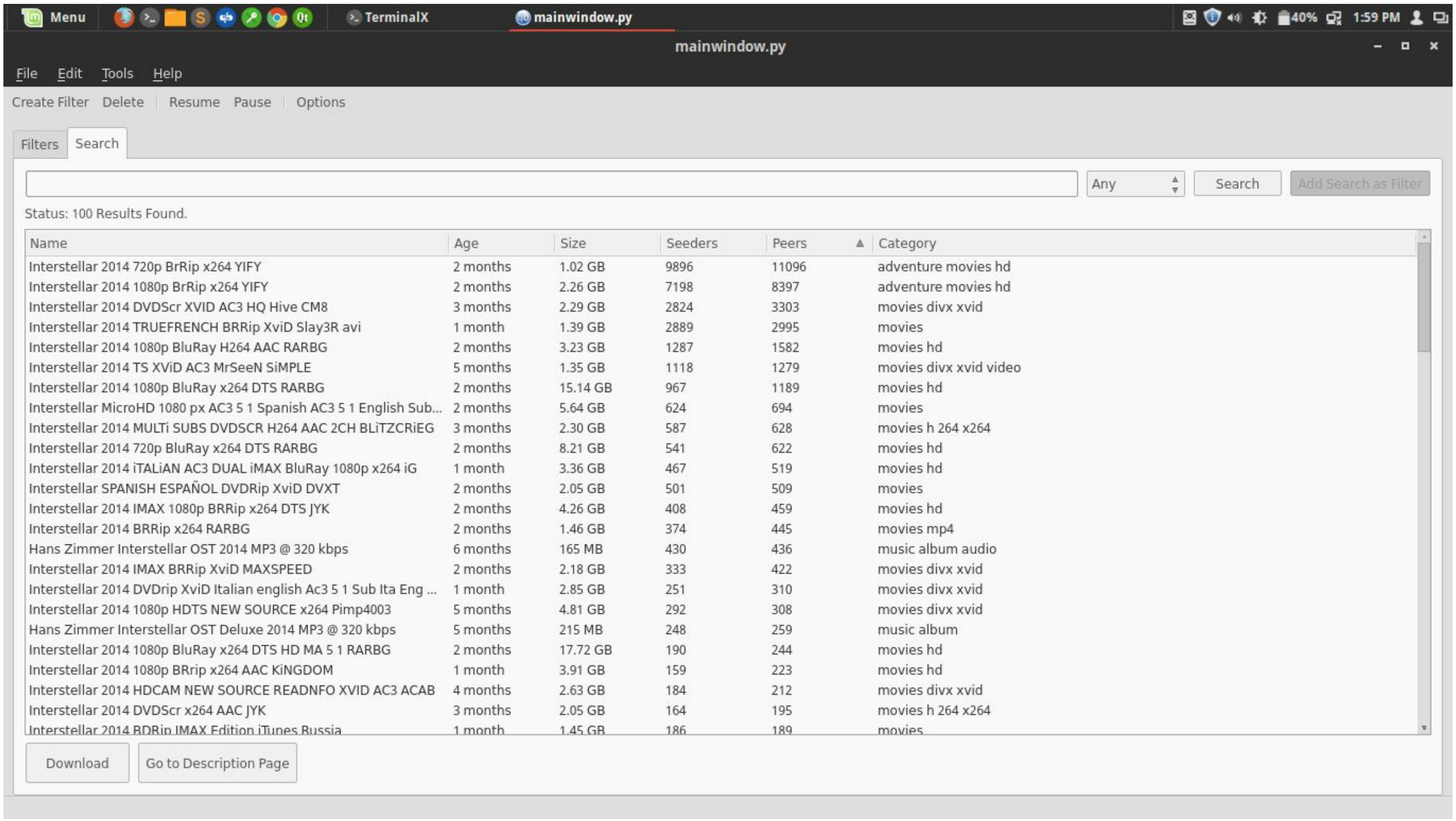


Screenshot 2



Status: Not Running

Screenshot 3



Status: 100 Results Found.

Name	Age	Size	Seeders	Peers	Category
Interstellar 2014 720p BrRip x264 YIFY	2 months	1.02 GB	9896	11096	adventure movies hd
Interstellar 2014 1080p BrRip x264 YIFY	2 months	2.26 GB	7198	8397	adventure movies hd
Interstellar 2014 DVDScr XVID AC3 HQ Hive CM8	3 months	2.29 GB	2824	3303	movies divx xvid
Interstellar 2014 TRUEFRENCH BRRip XviD Slay3R avi	1 month	1.39 GB	2889	2995	movies
Interstellar 2014 1080p BluRay H264 AAC RARBG	2 months	3.23 GB	1287	1582	movies hd
Interstellar 2014 TS XViD AC3 MrSeeN SiMPLE	5 months	1.35 GB	1118	1279	movies divx xvid video
Interstellar 2014 1080p BluRay x264 DTS RARBG	2 months	15.14 GB	967	1189	movies hd
Interstellar MicroHD 1080 px AC3 5 1 Spanish AC3 5 1 English Sub...	2 months	5.64 GB	624	694	movies
Interstellar 2014 MULTi SUBS DVDSCR H264 AAC 2CH BLiTZCRiEG	3 months	2.30 GB	587	628	movies h 264 x264
Interstellar 2014 720p BluRay x264 DTS RARBG	2 months	8.21 GB	541	622	movies hd
Interstellar 2014 iTALiAN AC3 DUAL iMAX BluRay 1080p x264 iG	1 month	3.36 GB	467	519	movies hd
Interstellar SPANISH ESPAÑOL DVDRip XviD DVXT	2 months	2.05 GB	501	509	movies
Interstellar 2014 IMAX 1080p BRRip x264 DTS JYK	2 months	4.26 GB	408	459	movies hd
Interstellar 2014 BRRip x264 RARBG	2 months	1.46 GB	374	445	movies mp4
Hans Zimmer Interstellar OST 2014 MP3 @ 320 kbps	6 months	165 MB	430	436	music album audio
Interstellar 2014 IMAX BRRip XviD MAXSPEED	2 months	2.18 GB	333	422	movies divx xvid
Interstellar 2014 DVDrip XviD Italian english Ac3 5 1 Sub Ita Eng ...	1 month	2.85 GB	251	310	movies divx xvid
Interstellar 2014 1080p HDTS NEW SOURCE x264 Pimp4003	5 months	4.81 GB	292	308	movies divx xvid
Hans Zimmer Interstellar OST Deluxe 2014 MP3 @ 320 kbps	5 months	215 MB	248	259	music album
Interstellar 2014 1080p BluRay x264 DTS HD MA 5 1 RARBG	2 months	17.72 GB	190	244	movies hd
Interstellar 2014 1080p BRrip x264 AAC KiNGDOM	1 month	3.91 GB	159	223	movies hd
Interstellar 2014 HDCAM NEW SOURCE READNFO XVID AC3 ACAB	4 months	2.63 GB	184	212	movies divx xvid
Interstellar 2014 DVDScr x264 AAC JYK	3 months	2.05 GB	164	195	movies h 264 x264
Interstellar 2014 BDRip IMAX Edition iTunes Russia	1 month	1.45 GB	186	189	movies

9. CONCLUSION

9.1 INFERENCES DRAWN

This system consists of a main desktop application, and no web or mobile counterpart. The desktop application will interface with the online torrent meta search engine, through XML requests. The desktop application will be used to search torrents, and add filters to notify, when torrents get uploaded in future, by interacting with the OS. The search engine will provide all the necessary details via XML feed, and by serving custom made queries, through application UI.

The application will need access to the Networking module of the OS, which in turn connects it to the Internet. The Internet connection will provide the application with the necessary listing of torrent information, with their details. The functionality provided by the search engine will be embedded into the application in order for the user to be able to use the functions in application in a seamless manner.

Since this is a utility application, it needs to facilitate user into extending its uses and interacting with other applications. The application will communicate with local desktop applications: a BitTorrent Client and a Web Browser. The web browser is used to open the Torrent description page on the respective Torrent hosting site from which the Search engine found the result. It provides facility to view user comments, uploader profile, and other useful information. The torrent client will be used to download the torrent data, and it will be opened automatically, directly from inside the application, providing further user convenience.

The desktop application has some restrictions about the resource allocation. To avoid problems with overloading the operating system the application is only allowed to have 50 active filters while running the application.

9.2FUTURE EXTENSIONS

Some of the future extensions may be like a server side based torrent notifier application rather than a desktop application for a 24*7 background persistent searching of the data.

The application can be a mobile based as well since there were more number of people using phones than PC(s). You'll get instant notification on your mobile device anywhere you are. You cannot access your PC from anywhere whereas you can operate your phone easily anywhere.

More features can be added to the Application other than the currently having ones to make it more customizable and functional. Searching can be made more efficient.

9.3 SCOPE AND LIMITATIONS

The software needs both persistent Internet connection, and background processing to fetch and display results at regular intervals of time. All system information is maintained in a database, which is located locally. The software also interacts with a web browser, and a torrent downloader software which are required to be already installed application on the user's computer. By using browser, users can view desired torrent download links, its description and user comments on the respected torrent website. By using torrent downloader software, users can download the torrent content. The application also has the capability to directly display torrent download links, in it. The application does not have the ability to download torrent content.

The goal of the notification feature aims to cut the time delay of when the content gets posted online and when the user discovers it. When the application is running, it automatically alerts the user, as soon as the torrent gets uploaded.

One dependency is that Torrent hosting sites, and Torrent search engine site are IP blocked on the user computer. They can be IP blocked on the host computer or the ISP itself or blocked on a network. The application will not work in that case, because it needs to communicate with these websites in order to function.

We assume that the user allows the application to run in the background and take processing power, and fetch data over a period of time. If the user shuts down the application frequently, then the functioning may be hampered. For optimum results, the application should be allowed to run as long as possible.

Another assumption is that all the programming language runtime the application uses, is installed on the user's computer.

Appendix

References

[1]IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended

[2] www.howstuffworks.com/image

[3] en.wikipedia.org

[4] unix.stackexchange.com

[5] www.askubuntu.com

[6] https://wiki.archlinux.org

[7] images.google.com

[8]Introduction to Python Programming and Developing GUI Applications with PyQT-(2011)-B. M. Harwani

[9] A Byte of Python - Swaroop C H- http://www.swaroopch.com/notes/python/

[10] Learning Python, 5th Edition – Mark Lutz 2013

[11] www.google.com

[12] docs.python.org

[13] www.stackoverflow.com

[14] www.riverbankcomputing.com/software/pyqt/

[15] doc.qt.io

[16] www.torrentz.com/help

[17] github.com/qbittorrent/qBittorrent

[18] www.zetcode.com/gui/pyqt4/

[19] Youtube Channel - "Simon Carr" – PyQt4 Tutorial Video Series