

**SPIDERBOT
WEB CRAWLER & DATA SCRAPER**

**BY
MR. NEEL V. ZADAFIYA
(15BCE178)**



**DEPARTMENT OF COMPUTER ENGINEERING
AHMEDABAD 382481
MAY 2018**

“SPIDERBOT”
(Web Crawler & Data Scraper)
Major Project Report

Submitted in Partial Fulfillment of the Requirements for Degree of
Bachelor of Technology
In
Computer Engineering

By
Mr. Neel V. Zadafiya
(15BCE178)

Guide
Prof. Tarjni K. Vyas



DEPARTMENT OF COMPUTER ENGINEERING
Ahmedabad 382481

May 2018

CERTIFICATE

Acknowledgement

I take the opportunity to thank my instructor Prof. Tarjni K. Vyas for the most effective and valuable guidelines in my Python Application. With the help of her knowledge and experience I further directed to develop any advance data scraper system.

Prof. Tarjni K. Vyas gave valuable suggestions regarding my application functionality and also motivated me for further development in python-based data scraping in another domain.

I also like to thank my classmates who helped me when I started learning web crawling and data scraping (In especially for python programs).

I also thankful to my teaching and other guideline faculty of college to accomplish and support my project and gave necessary guideline.

Mr. Neel V. Zadafiya

15BCE178

Abstract

Web crawling is a technique to crawl a links on web like internet. Internet is filled by so much data and most of the data is represented by websites. Using data scraper, it is possible to scrap those data into structured format of database. A data stored in database can be processed easily rather than a data displayed on website. Without any tool user have to copy and paste data manually from website to database. Spiderbot is a tool that makes this process automatic. Spiderbot or PyScrap is a system consist of various data scraping and storage tools. Once user make an assembly line like flow diagram on PyScrap, data scraping will be done automatically. Scraping a data may sound illegal, but if user can see data in browser, then they also can scrap data automatically. If copying data from website to anywhere manually is a legal, then PyScrap is just an automation tool that makes things easy.

Contents

Chapter	Title	Page No.
1	Introduction	1
	1.1 About the company	2
	1.1.1 Introduction of the company	2
	1.1.2 Quality Policy	2
	1.1.2.1 Vision	3
	1.1.2.2 Mission	3
	1.1.2.3 Quality Statement	3
	1.1.2.4 Quality Objectives	3
	1.1.3 Communications	3
	1.1.4 Resources	4
	1.2 The System	4
	1.2.1 Definition of the system	4
	1.2.2 Purpose and objective	4
	1.2.3 About present system	4
	1.2.4 Proposed system	5
	1.3 Project Profile	5
	1.3.1 Project title	5
	1.3.2 Scope of the project	5
	1.3.3 Project team	5
	1.3.4 Hardware/Software environment in company	6
	1.4 Literature Survey	6
	1.4.1 Existing web crawlers/bots	6
	1.4.2 Existing data scrapers	6
	1.4.3 Architecture of Spiderbot	7
	1.5 Project planning and scheduling	8
2	System Analysis	10
	2.1 Feasibility Study	11
	2.1.1 Operational feasibility	11
	2.1.2 Technical feasibility	11
	2.1.3 Financial feasibility	11

	2.2 Requirement Analysis	11
	2.2.1 Hardware requirements	11
	2.2.2 Software requirements	12
	2.2.3 Functional requirements	12
	2.2.4 Non-functional requirements	12
	2.3 Context Diagram	13
3	System Design	14
	3.1 System Flow	15
	3.2 Entity Relationship Diagram	16
	3.3 Data Dictionary	17
	3.3.1 List of tables	17
	3.3.2 mapper	17
	3.3.3 online_shopping_data	17
	3.3.4 stock_market_data	18
	3.3.5 link_grabber_data	18
	3.3.6 quotes_data	18
4	User Manuals	19
	4.1 Main Window	20
	4.2 Sub Frames	21
	4.2.1 Title frame	21
	4.2.2 Bot frame	22
	4.2.3 Tool frame	22
	4.2.4 Canvas frame	23
	4.2.5 Input frame	24
	4.2.6 Output frame	25
	4.2.7 Status frame	25
	4.3 Tools	26
5	Testing	34
	5.1 Individual tool testing	35
	5.2 Testing of relation among tools	35
	5.3 Testing of tools in execution queue	35
	5.4 Testing of deletion of tools	35

6	Future Enhancement	36
7	Appendix	38
	7.1 Tools Used	39
8	Bibliography	40
	8.1 Books	41
	8.2 Websites	41
	8.3 References	41

Figures

Fig.No.	Caption	Page No:
1.1	Architecture of Spiderbot	7
1.2	Project Plan	8
1.3	Gantt chart of project plan	9
2.1	Context diagram of PyScrap	13
3.1	System flow diagram of PyScrap	15
3.2	Entity relationship diagram of PyScrap	16
4.1	Main window of PyScrap	20
4.2	Structure of main window of PyScrap	21
4.3	Bot frame	22
4.4	Tool frame	22
4.5	Canvas frame	23
4.6	Mini-map	24
4.7	Input frame	24
4.8	Output frame	25

Tools

Sr.	Name	Page No:
1	Start	26
2	URL Provider	27
3	Web Crawler	27
4	Media	28
5	Rename	28
6	Sort	29
7	Object Detector	29
8	Keys	30
9	Mouse	30
10	Delay	31
11	Filter	31
12	DB Connection	32
13	Table	32
14	Column	33
15	Download	33

Chapter 1: Introduction

1.1 About the company – Institute of Technology Nirma University



1.1.1 Introduction of the company

Institute of Technology, Nirma University, earlier known as Nirma Institute of Technology, started in 1995 by Nirma Education and Research Foundation (NERF), was the first self-financed engineering college in Gujarat. Within 18 years of inception, Institute of Technology is a leading hub of education, offering multidisciplinary undergraduate, postgraduate and Ph.D. programs in engineering. The institute is ranked within top 25 self-financed engineering colleges of India in the survey conducted by various rating agencies. The faculty members and students of the Institute have won many prestigious awards and bring laurels to the institute.

The Institute is located in peaceful and sylvan surroundings of Ahmedabad city in the heart of Gujarat. The Institute provides disciplined, serene and conducive environ for reflection, repose and research. The Campus is overwhelmed with lush green sceneries masking the concrete beneath.

The ardent pursuit of knowledge by the young aspirants leads to knowledge generation and innovative solutions for the community and society. And, to give wings to these aspirations, the Institute presently has more than 4500 students and 180 faculty members, making relentless efforts for making a mark with their presence globally. The campus vibrates with not only world class curricular activities but also with myriad activities like international conventions, symposia, conferences, student competitions, conclaves, short-term industry relevant programs, cultural activities and many more.

1.1.2 Quality Policy

Nirma University, since its inception, has always strived for achieving highest standards of quality in all its endeavors and has taken several initiatives in its quest for excellence. It has developed and deployed robust systems and processes for continuous improvement of quality and this quality document which is aligned to the vision, mission and strategic objectives of the university, prepared for the purpose of providing guidelines for designing and implementing quality to various stakeholders, is a part of that process.

1.1.2.1 Vision

Shaping a better future for mankind by developing effective and socially responsible individuals and organizations.

1.1.2.2 Mission

Nirma University emphasizes the all-round development of its students. It aims at producing not only good professionals but also good and worthy citizens of a great country, aiding in its overall progress and development. It endeavors to treat every student as an individual, to recognize their potential and to ensure that they receive the best preparation and training for achieving their career ambitions and life goals.

1.1.2.3 Quality Statement

To develop high quality professionals who reflect and demonstrate values that the university stands for through innovation and continuous improvement in facilitation of learning, research and extension activities.

1.1.2.4 Quality Objectives

- To equip students with relevant knowledge, skills, attitudes and global competencies to make them more employable and capable of contributing to the growth of organizations, communities and the nation.
- To provide students access to high quality infrastructure and learning resources and support systems to enhance their learning experience.
- To promote world class research and innovation and build high quality intellectual capital.
- To adhere to regulations and guidelines prescribed by various regulatory agencies from time to time.
- To foster a culture of excellence by following quality management frameworks developed by reputed accreditation agencies, following best practices and actively engaging and institutionalizing continuous quality improvement practices using feedback from various stakeholders.

1.1.3 Communication

Contact person: Prof. Tarjni K. Vyas

Office Address: B block – B 103, Institute of Technology, Nirma University. Sarkhej-Gandhinagar Highway, Post: Chandlodia, Via: Gota, Ahmedabad - 382 481.

Office Phone: 07930642252

1.1.4 Resources

Nirma University is at location on SG highway and provides state of the art infrastructure that supports more than 5000 students. Nirma University provides access to all the labs and libraries for development of project. It also provides hardware, software and financial support in order to complete project.

1.2 The System

1.2.1 Definition of the system

The system revolves around the need of automation in data scraping. Data scraping can be done manual but it consumes lots of time and human resources. When data is very large to be scraped, inconsistency can be occurred by human hands. The system provides automation tools to scrap data in assembly like line. It's like making a car step by step. The system provides to scrap data in different level and different types. The system also provides to store and process scrapped data by user.

1.2.2 Purpose and objectives

The system consists of new architecture introduced to scrap data while crawling websites. It helps to automate data scraping and web crawling. It saves significant amount of time and labor. Drag and drop tools introduced in the system provides very interactive GUI and gives good look and feel system. The system also gives facility store and manage scraped data in structured format supported by MySQL.

1.2.3 About present system

The present system provides its built-in web browser to scrap data. In this system user has to select data which need to be scraped and table of data will be displayed. Table of data contains structured format of data^{[1][2]}. The purpose of this system is to convert unstructured data into structured data. Here, unstructured data means data is in structure which is not suitable for user or system used by user. For website owner data is structured because it directly comes from database. But for scraper that data is unstructured^[5]. So present systems are able to scrap those data and store them into structured format like table. The limitation of the system is that they can't crawl data while scrap them. Because user can view only one page at a time and crawling demands hundreds of pages to be viewed at a time. To overcome issues with current system PyScrap is introduced^{[3][4]}.

1.2.4 Proposed system

The issue with present system is an inability of system to scrap data while crawling. The proposed system uses new architecture for crawl websites and scrap data. The proposed architecture contains different types of tools. Used need to place tools in the center of the area and connect them in order to make a flow chart. Each tool contains their unique name, symbol, input panel and output panel. The output of a toll will be passed to another tool as its input. It will be processed by that tool and passed to another tool till the end of a queue. The system is able to process multiple queues at a time. The tools which are not connected with a start toll will not take part in execution process.

1.3 Project Profile

1.3.1 Project title

Spiderbot – web crawler and data scraper are tools that provides automation to the process of web crawling and data scraping. Spiderbot provides interactive GUI and ease of access to data scraping techniques to the user. Current system is named as PyScrap. In this document, Spiderbot and PyScrap both are referring to the same system.

1.3.2 Scope of the project

- The scope of Spiderbot is firstly limited to what is publicly accessible. Some sites will require credentials to access all content, and what is technically accessible.
- If a content is contained within a "difficult" to examine container such as flash, then it might not be crawled.
- If content is presented via JavaScript, then this may not be crawled faster, as crawling engines often do not execute scripts on the sites they access.
- The Spiderbot can only be sued by users who have basic knowledge of database and inspect element from browser.

1.3.3 Project Team

- Mr. Neel V. Zadafiya – Software Development, Design Lead and Management

1.3.4 Hardware/Software environment in company

The company has innovators from various fields with various domains of knowledge. The company has each and every hardware and software needed for project development. The company also provides library and research paper access in order to boost research on project. Access to printer is also available in the company. If the software is not available for project development, then it can be arranged even if it is paid. Experts are available to support in project development and enhancement.

1.4 Literature Survey

1.4.1 Existing web crawlers/bots[6]

- GoogleBot
- Bingbot
- Slurp Bot (Yahoo)
- DuckDuckBot (DuckDuckGo)
- Baiduspider
- Yandex Bot (Russian)
- Sogou Spider (Chinese)
- Exabot (French)
- Facebook External Hit
- Alexa Crawler

1.4.2 Existing data scrapers[6]

- Import.io
- Webhose.io
- Dexi.io
- scrapinghub
- Spinn3r
- Visualscraper
- Parsehub

1.4.3 Architecture of Spiderbot

The general architecture followed by web crawlers is given below. It represents block diagram of components used by web crawlers[7][8]. A crawled link will be crawled again from URLs database. In this system content processor works as data scraper and also passes data to URL extractor.

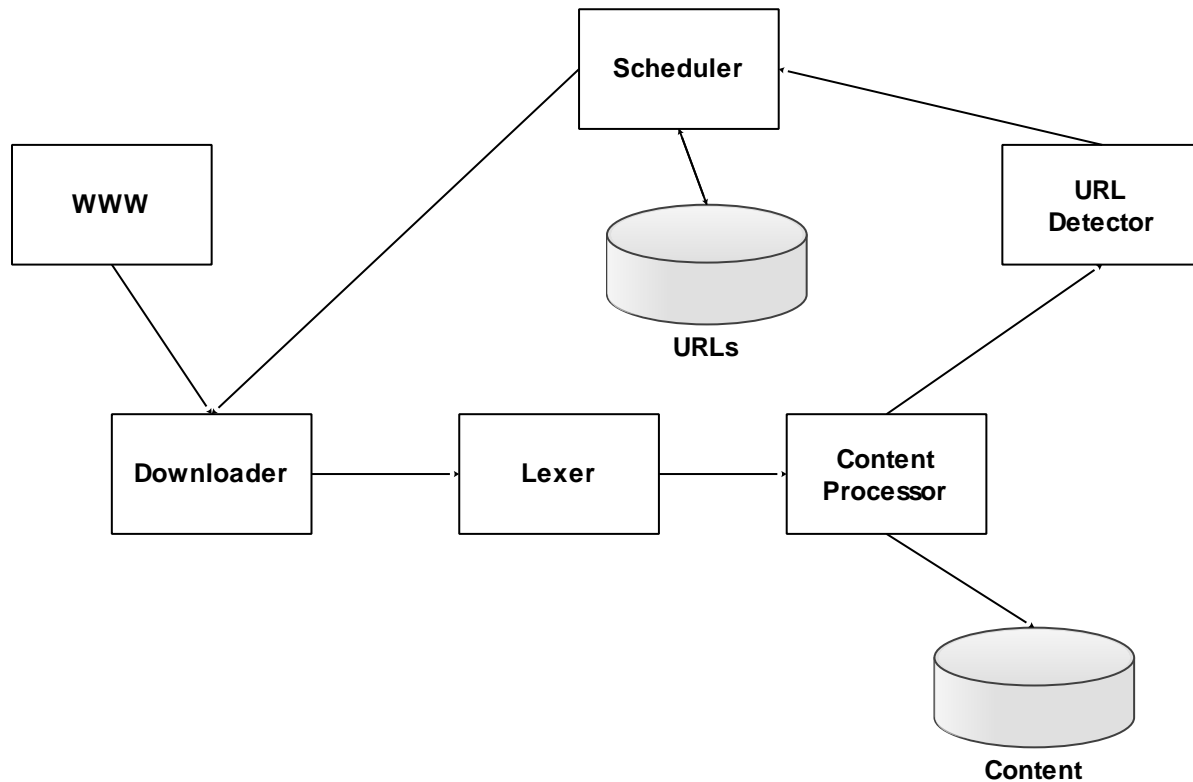


Figure 1.1: Architecture of Spiderbot

1.5 Project planning and scheduling

	Task Name	Start Date	End Date	Duration
1	<input type="checkbox"/> Literature Survey	01/01/18	01/31/18	31 Days
2	Requirement Gathering	01/01/18	01/15/18	15 Days
3	Comparing Existing Tools	01/16/18	01/31/18	16 Days
4	<input type="checkbox"/> GUI for Proposed System	02/01/18	02/15/18	15 Days
5	Learn widgets	02/01/18	02/15/18	15 Days
6	Implement system UI	02/01/18	02/15/18	15 Days
7	<input type="checkbox"/> Resource Grabber	02/16/18	02/28/18	13 Days
8	Resource Scanner	02/16/18	02/20/18	5 Days
9	Resource Downloader	02/21/18	02/25/18	5 Days
10	Rename Utility	02/26/18	02/28/18	3 Days
11	<input type="checkbox"/> Crawler Integration	03/01/18	03/15/18	15 Days
12	Basic Crawler	03/01/18	03/10/18	10 Days
13	Integration	03/11/18	03/15/18	5 Days
14	<input type="checkbox"/> Bot Creator	03/16/18	03/31/18	16 Days
15	Data Selectors	03/16/18	03/25/18	10 Days
16	Scheduled Bot	03/26/18	03/31/18	6 Days
17	Bot Manager	04/01/18	04/07/18	7 Days
18	<input type="checkbox"/> System Improvement	04/08/18	04/17/18	10 Days
19	Improved GUI	04/08/18	04/12/18	5 Days
20	Distributed Processing	04/13/18	04/17/18	5 Days
21	<input type="checkbox"/> Finalization	04/18/18	04/30/18	13 Days
22	Perform Tests	04/18/18	04/21/18	4 Days
23	DCR	04/22/18	04/25/18	4 Days
24	Publish Product	04/26/18	04/30/18	5 Days

Figure 1.2: Project Plan

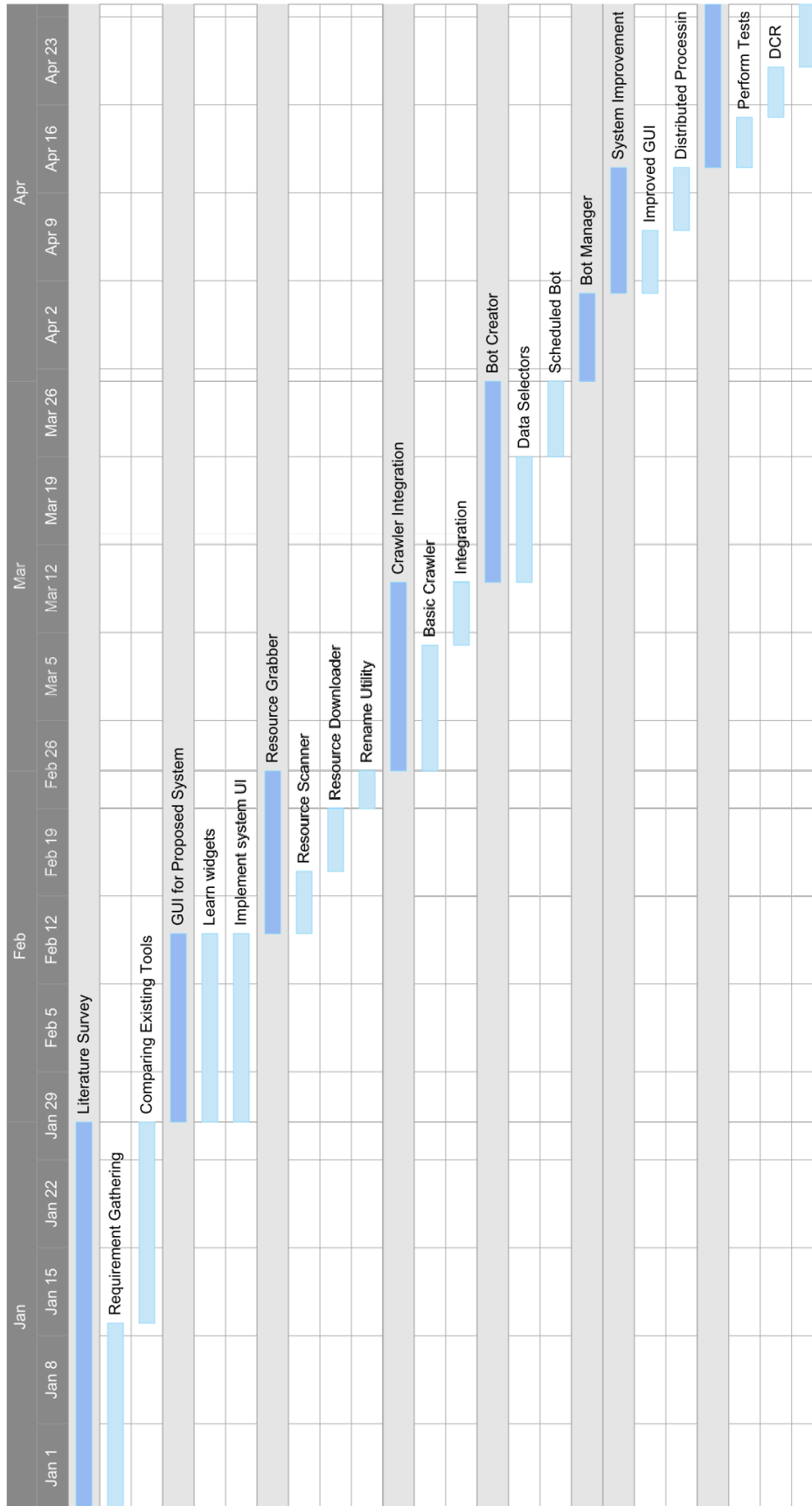


Figure 1.3: Gantt chart of project plan

Chapter 2: System Analysis

2.1 Feasibility Study

Feasibility study of a project defines its rate of success. It also defines is project going to completed in given time or not. There are number of fields in which feasibility study can be applied. Separate areas of this kind of study is examined and described below in major three categories.

2.1.1 Operational feasibility

This system can be beneficial only for user which have knowledge of handling databases and can inspect elements from web browsers. But users of this era are intelligent enough to use current system. So, it can be said that current system is operationally feasible. As the development proceed efforts are made for data retrieval and ease of access to user.

2.1.2 Technical feasibility

Spiderbot can be developed using current available hardware and software which are Windows 8.1, Python 3.6 and respective libraries. So, it can be said that current system is technically feasible.

2.1.3 Financial feasibility

Spiderbot can be developed without using any extra finance. No any extra hardware or software from outside of university campus is need to implement Spiderbot. So, it can be said that current system is financially feasible.

2.2 Requirement Analysis

2.2.1 Hardware requirements

- RAM: 2GB or more
- Processor: More of equivalent of Intel i3
- HDD space: Minimum of 700 MB
- Internet 1 Mbps (Network Requirement)

2.2.2 Software requirements

- OS: Windows 8 or above
- Python 3.6
- Google Chrome 66.0.3359 or more
- Selenium chrome driver
- WAMP 2.4
- Python libraries: PyQt5, selenium and urllib

2.2.3 Functional requirements

- Customized Web Crawling
- Distributed Processing
- Intelligent Recrawling
- Easy data extraction
- Ability to Clone (Multithreading)
- Trackable Crawling

2.2.4 Non-functional requirements

- Language Independent
- Crawl Politely
- Intercommunication among Crawlers
- Backup and Recovery
- Follow Robots Exclusion Protocol
- Efficient File Management
- User Friendly GUI

2.3 Context Diagram

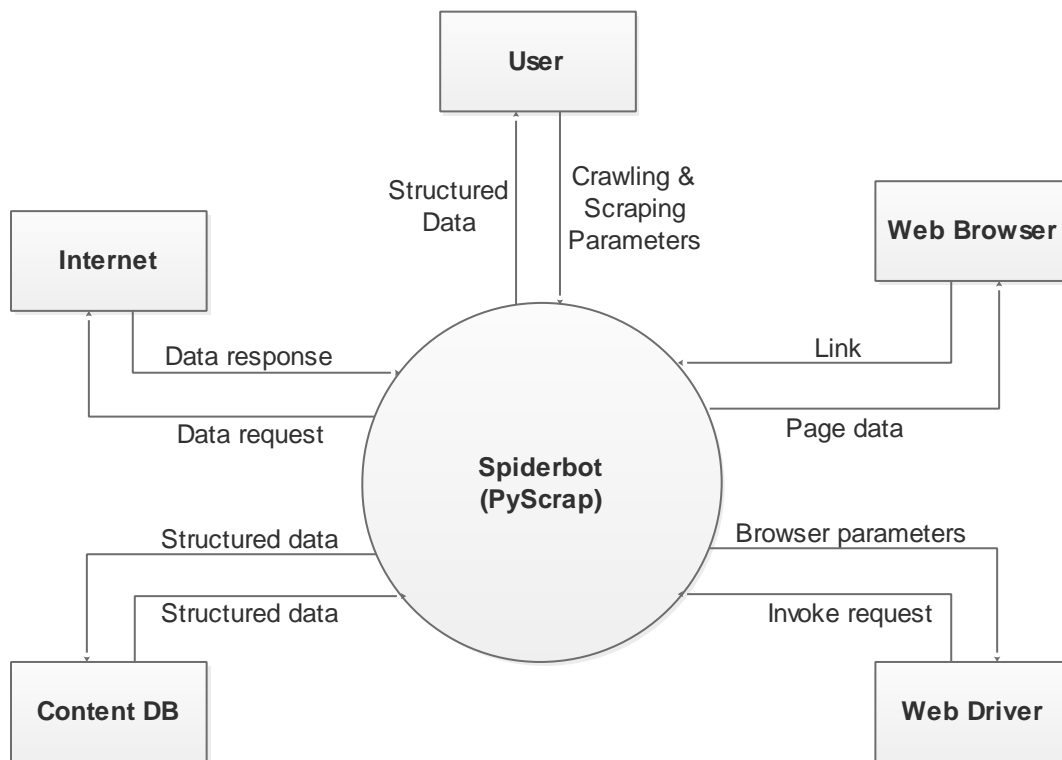


Figure 2.1: Context diagram of PyScrap

Chapter 3: System Design

3.1 System Flow

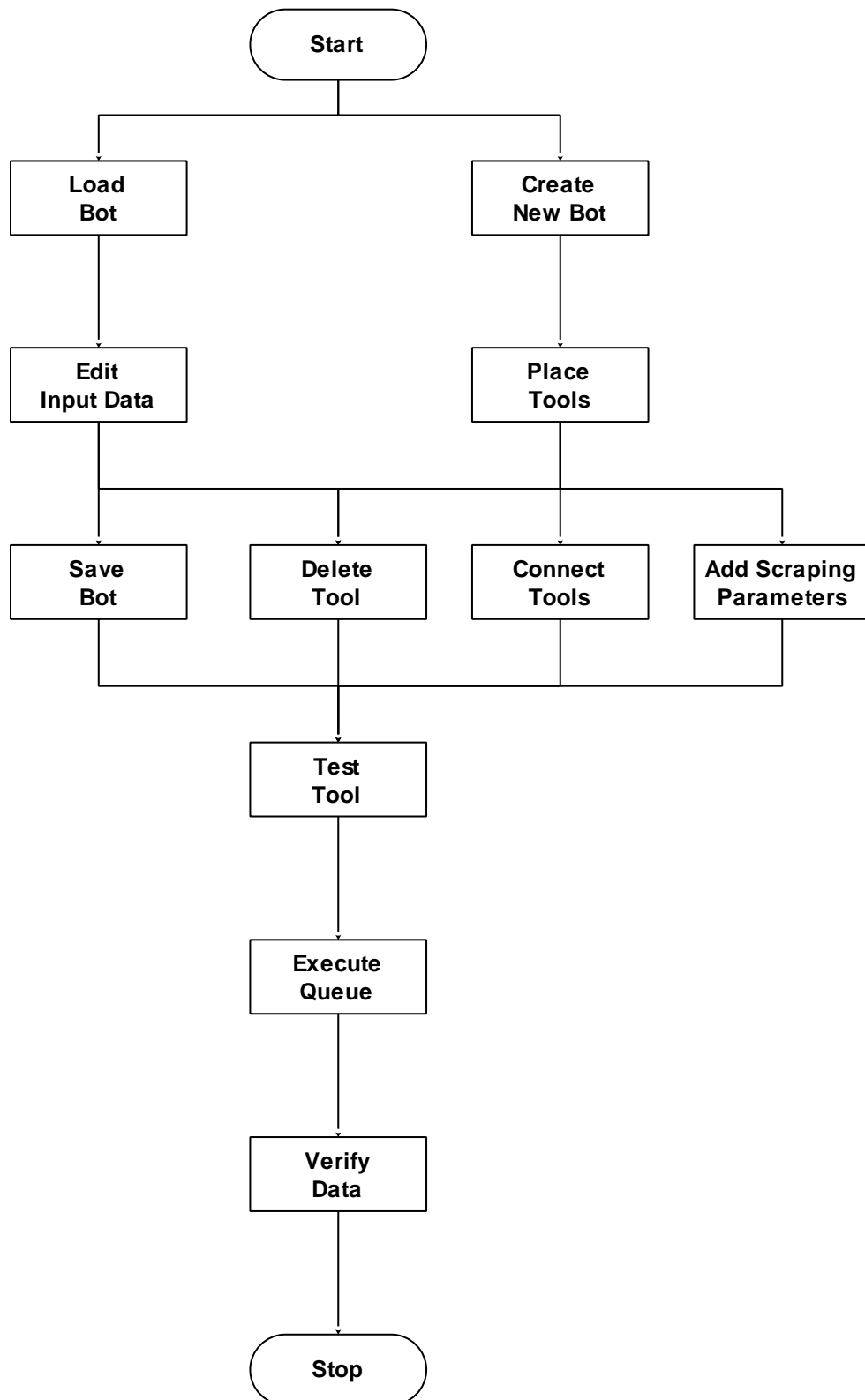


Figure 3.1: System flow diagram of PyScrap

3.2 Entity-Relationship Diagram

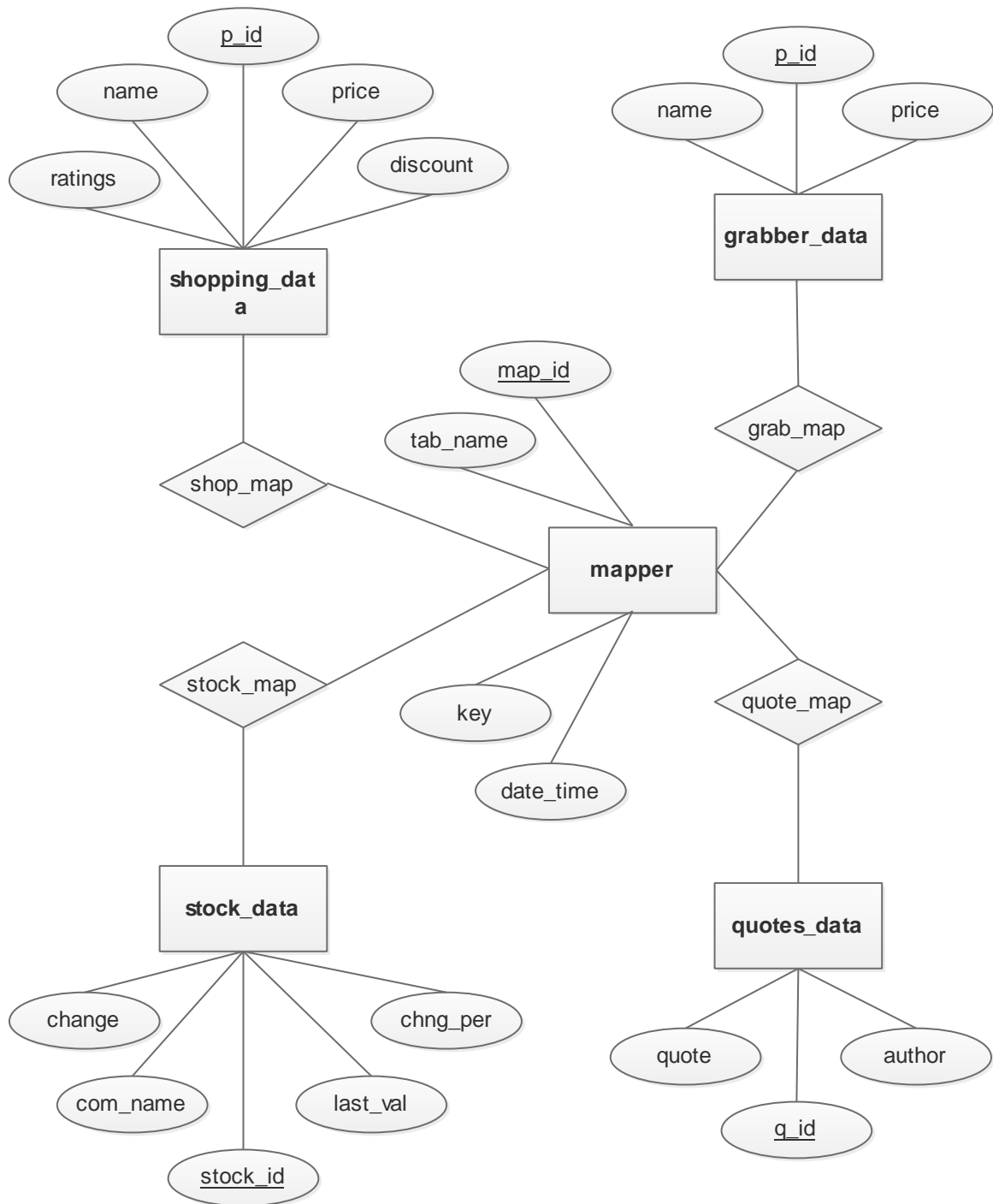


Figure 3.2: Entity relationship diagram of PyScrap

3.3 Data Dictionary

3.3.1 List of tables

Sr.	Table Name
1	mapper
2	online_shopping_data
3	stock_market_data
4	link_grabber_data
5	quotes_data

3.3.2 mapper

Sr.	Field	Type	Constraints
1	map_id	int(11)	primary key
2	table_name	text	
3	key	int(11)	
4	date_time	datetime	

3.3.3 online_shopping_data

Sr.	Field	Type	Constraints
1	p_id	int(11)	primary key
2	name	text	
3	price	decimal(10,0)	
4	ratings	decimal(10,0)	
5	discount	decimal(10,0)	

3.3.4 stock_market_data

Sr.	Field	Type	Constraints
1	stock_id	int(11)	primary key
2	company_name	text	
3	last_value	decimal(10,0)	
4	change	decimal(10,0)	
5	relative_change	decimal(10,0)	

3.3.5 link_grabber_data

Sr.	Field	Type	Constraints
1	link_id	int(11)	primary key
2	link	text	
3	parent_link	text	

3.3.6 quotes_data

Sr.	Field	Type	Constraints
1	q_id	int(11)	primary key
2	quote	text	
3	author	text	

Chapter 4: User Manuals

4.1 Main Window

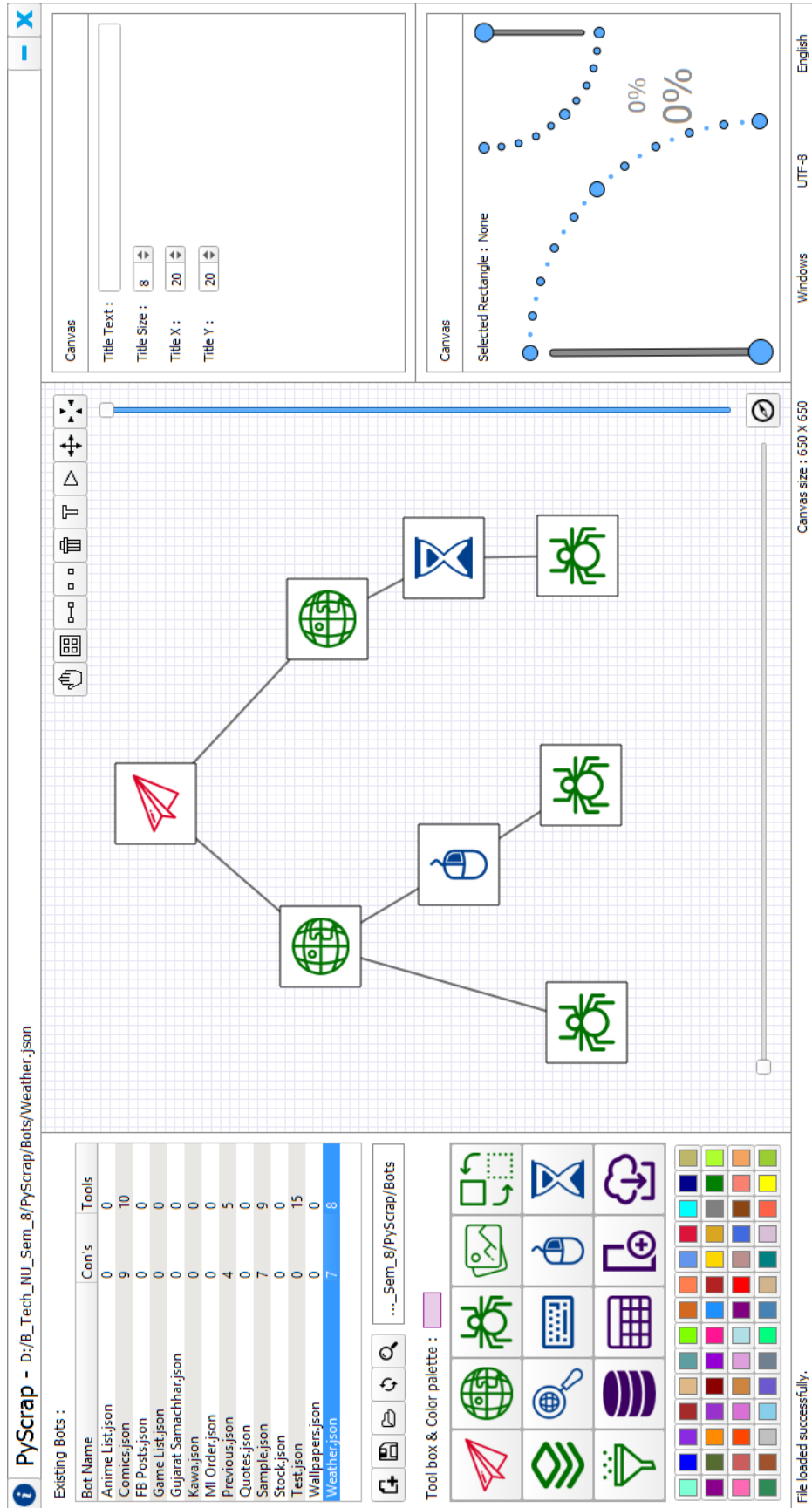


Figure 4.1: Main window of PyScrap

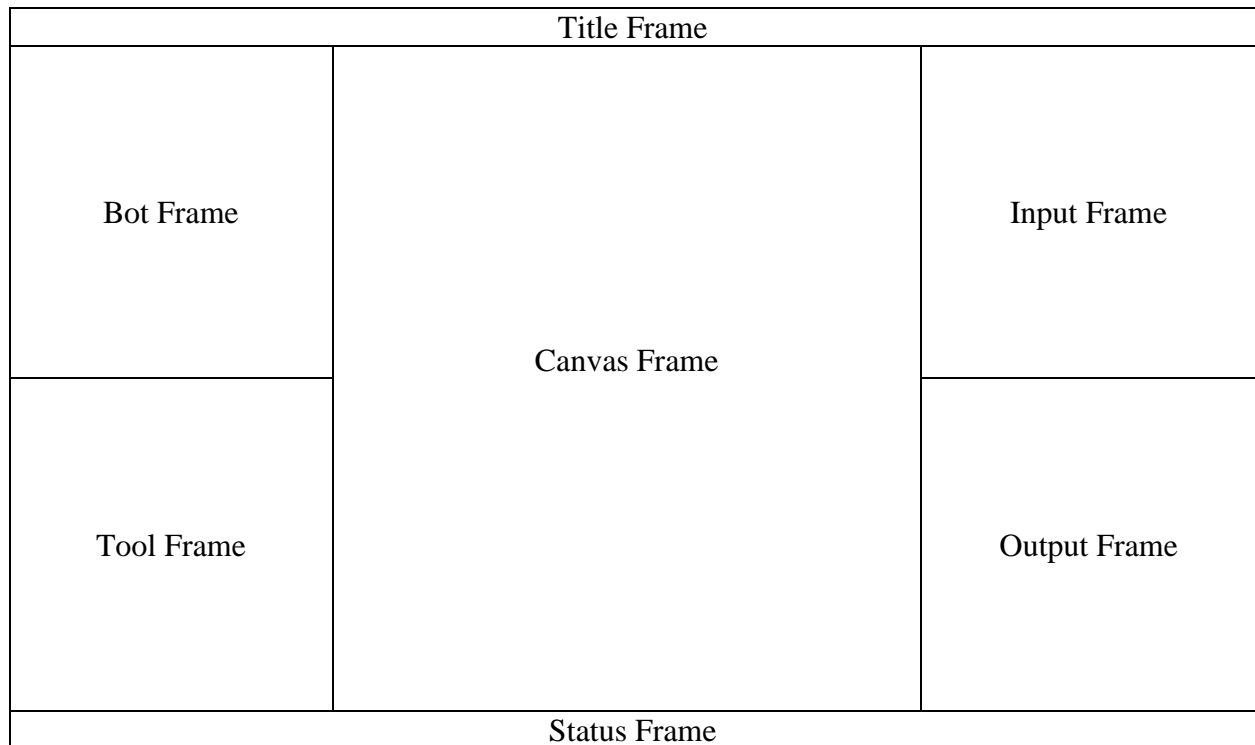


Figure 4.2: Structure of main window of PyScrap

4.2 Sub Frames

4.2.1 Title frame

Title frame is the first line of main window. It contains help button, minimize button and close button. It also indicates current loaded bot with its name and path.

4.2.2 Bot frame

Existing Bots :

Bot Name	Con's	Tools
Anime List.json	0	0
Comics.json	9	10
FB Posts.json	0	0
Game List.json	0	0
Gujarat Samachhar.json	0	0
Kawa.json	0	0
MI Order.json	0	0
Previous.json	4	5
Quotes.json	0	0
Sample.json	7	9
Stock.json	0	0
Test.json	0	15
Wallpapers.json	0	0
Weather.json	7	8












Figure 4.3: Bot frame

Bot frame displays list of available bots with their connections and total number of tools. The mini toolbar shown in diagram contains buttons for new file, save file, load/open file, refresh list and browse directory.

4.2.3 Tool frame

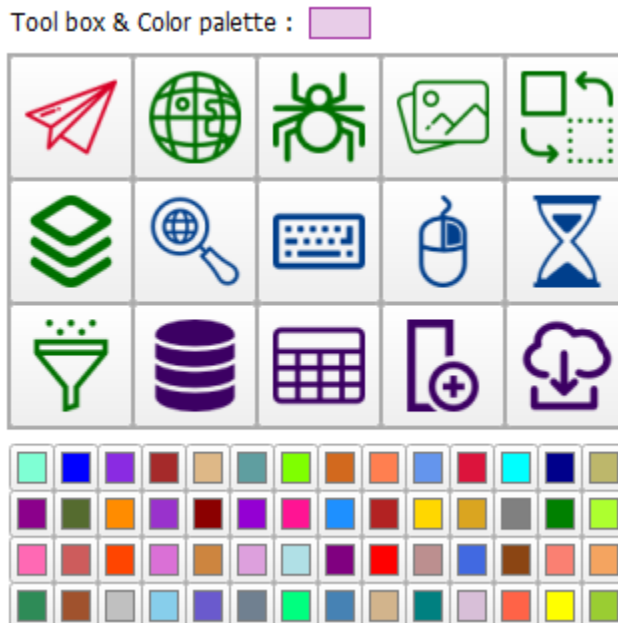


Figure 4.4: Tool frame

Tool frame contains available tools for data scraping and web crawling. It also provides color pellet to draw rectangles for visual groupings of tool.

4.2.4 Canvas frame

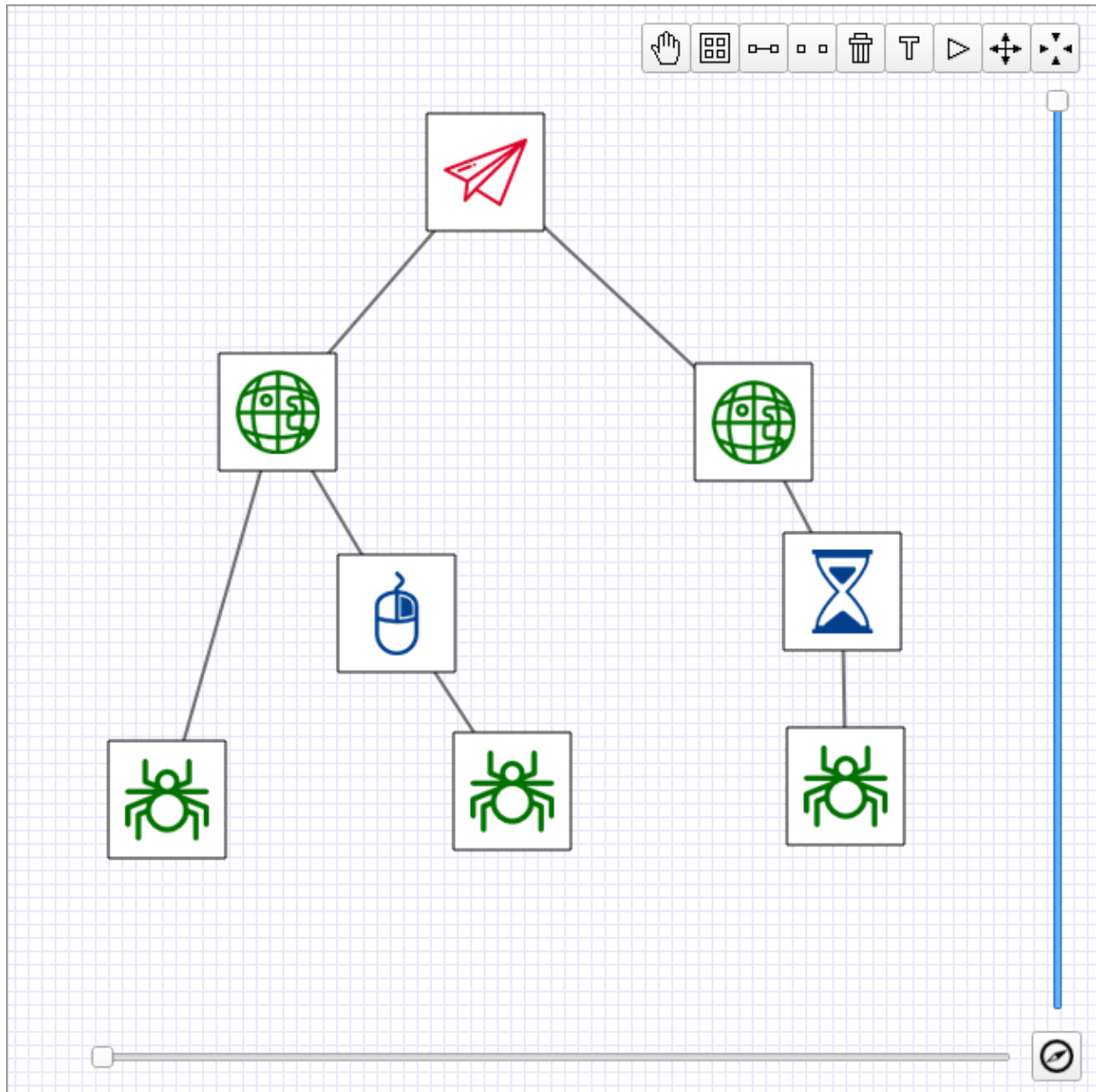


Figure 4.5: Canvas frame

Canvas frame contains tools and connections placed by user. The toolbar placed on top-right corner contains hand tool, rectangle tool, connection, disconnection, delete tool, test tool, play queue, expand canvas and collapse canvas buttons. It also contains button for toggle mini-map which displays relative location of tools in canvas. The example of mini-map is given below.

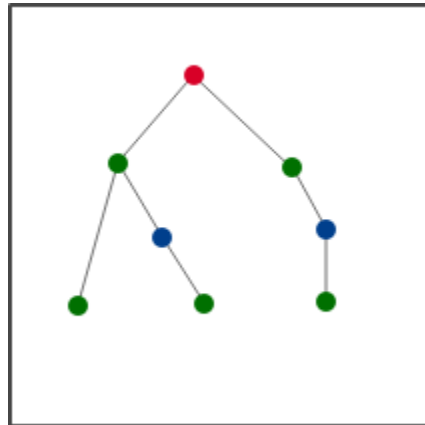


Figure 4.6: Mini-map

4.2.5 Input frame

Canvas

Title Text :

Title Size :

Title X :

Title Y :

Figure 4.7: Input frame

Input frame enables user to set web crawling and data scraping parameters. It also facilitates user to give custom text to tool in order to display that name on canvas.

4.2.6 Output frame

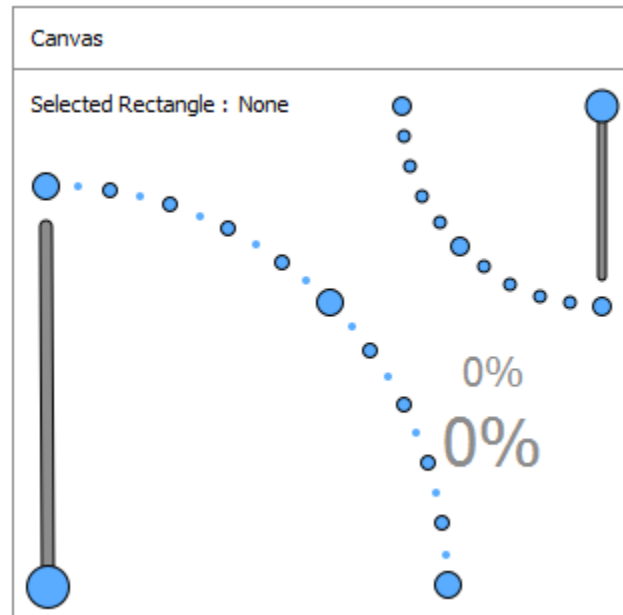


Figure 4.8: Output frame


Output frame displays the output generated by respective tool. Output can be in form of links, filenames or status data.

4.2.7 Status frame

Status frame is the last line of main window. It displays status messages along with canvas size, operating system, character encoding scheme and user language.

4.3 Tools

PyScrap contains 15 core tools for web crawling and data scraping. Each and every tool is described below with their name, purpose, input frame, instruction, output type and possible sources. User has to follow the rules provided in user manual to get desired output. The tables given below known as tool cards. Each card contains major information about tools. Sample card with start tool is given below. Other tool cards are available followed after start tool.

	Start	<div>Starting Point of Execution Flow</div> <div>Name : <input type="text"/></div>
	Provides starting point of execution queue.	
Input frame given in right box contains input widgets to name a tool. The icons given in last row of this card indicates possible sources of input for current tool.		
Output: Tool names in execution queue.		
None		

**URL Provider**

Provides URL to be crawled.

http:// or https:// is necessary to prefix to URL.

Output:
URL status with status code.

Url ProviderName : Url : *

* Use http:// or https://

**Web Crawler**

Extracts links from URL provider and repeats process in recursive way.



0 level indicates links from current page only. External link means links outside from domain of current page.



Output:
List of links with parent links.







Web CrawlerName : No. of Levels : ☐ Explore External Links







	Media extractor Extracts media from links.	<div>Media Extractor</div> <div>Name : <input type="text"/></div> <div> <input type="checkbox"/> Images <input type="checkbox"/> Documents <input type="checkbox"/> Other </div>
Documents are those files which contains simple text data inside it.		
Output: List of media file links with parent links.		







	Rename Renames media.	<div>Rename Media</div> <div>Name : <input type="text"/></div> <div> <input checked="" type="radio"/> Extract 0 0 Sub 0 to 0 <input type="radio"/> Existing <input type="radio"/> Custom <input type="text"/> </div> <div> <input type="text"/> <div> <input type="button" value="◀"/> <input type="button" value="▶"/> <input type="button" value="△"/> <input type="button" value="▽"/> </div> </div> <div>Select Separator Char: <input type="text" value="Null"/></div> <div>Not Found Character : <input type="text" value="Null"/></div>
First two boxes after extract are level of links and location of word after '/' respectively. They are used to get base word. First two boxes after sub are used to extract substring from base word.		
Output: List of media file links with parent links.		






	Sort Sorts media.	<div>Sort</div> <div>Name : <input type="text"/></div> <div> <div>Inc Dec</div> <div>Sort by Name : <input type="radio"/> <input type="radio"/></div> <div>Sort by Size : <input type="radio"/> <input checked="" type="radio"/></div> <div><input type="checkbox"/> Group by level</div> </div>
If group by level is checked then it sorts links within domain of parent link only.		
Output: List of media file links with parent links.		
		





	Object Detector Detects object from DOM.	<div>Object Detector</div> <div>Name : <input type="text"/></div> <div>XPath : <input type="text"/></div>
XPath can be copied from inspect element of page using chrome-based browsers.		
Output: Status of object.		
		



	Keys Send keys to the web page.	Keyboard Events
Output: Status of action.		
Keys will be sent in sequential manner to the object at once.		
    		



	Mouse Send mouse events to the web page.	Mouse Events
Two boxes after click at indicates x axis and y axis respectively.		
Output: Status of action.		
Name : <input type="text"/> <input checked="" type="radio"/> Click at current position <input type="radio"/> Click at <input type="text" value="0"/> <input type="text" value="0"/> <input type="radio"/> Scroll up <input type="text" value="0"/> <input type="radio"/> Scroll down <input type="text" value="0"/>		
    		





	Delay Delays process in seconds.	<div> Delay </div> <div> Name : <input type="text"/> </div> <div> Wait : <input type="text" value="0.00"/> <input type="button" value="Second"/> </div>
Delay should be in seconds.		
Output: Status of action.		
    		

	Filter Filters links and media.	<div> Filter </div> <div> Name : <input type="text"/> </div> <div> <input checked="" type="radio"/> Contains : <input type="text"/> </div> <div> <input type="radio"/> RegEx : <input type="text"/> </div> <div> First n : <input type="text" value="0"/> * </div> <div> Last n : <input type="text" value="0"/> * </div> <div> <input type="checkbox"/> Group by level </div> <div> * 0 means all from first/last </div>
If group by level is checked then it filters links within domain of parent link only.		
Output: List of media links with parent links,		
   		

	DB Connection Provides database connection.	<div>DB Connection</div> <div>Name : <input type="text"/></div> <div>Host Path : <input type="text"/></div> <div>DB Name : <input type="text"/></div> <div>Login ID : <input type="text"/></div> <div>Password : <input type="password"/></div>
MySQL is needed.		
Output: Status of database.		
  		

	Table Selects table from database.	<div>Table</div> <div>Name : <input type="text"/></div> <div>Table Name : <input type="text"/></div> <div>Row Code : <input type="text"/></div>
Row code can be copied from page source using chrome-based browsers. Row code contains data to be filled in entire row of the table.		
Output: Sample data of row.		
		

	<p>Column</p> <p>Selects column from table.</p>	<div> <p>Column</p> <hr/> <p>Name : <input type="text"/></p> <p>Col'n Name : <input type="text"/></p> <p>Cell Code : <input type="text"/></p> </div>
<p>Cell code should be copied from row code. It indicates data to be filled in cell of the table.</p>		
<p>Output: Sample data of cell.</p>		
		

	<p>Download</p> <p>Downloads extracted media from web page.</p>	<div> <p>Download</p> <hr/> <p>Name : <input type="text"/></p> <p>Path : <input type="text" value="D:/B_Tech_NU_Sem_8/PyScrap"/></p> <p style="text-align: right;"><input type="button" value="Browse"/></p> <p>Select action if files are existed :</p> <p><input checked="" type="radio"/> Rename files in inceasing order</p> <p><input type="radio"/> Replace files</p> <p><input type="radio"/> Skip files</p> </div>
<p>Rename files in increasing order indicates that if file x exists then new name of the file will be x1. If x1 exists then new name of the file will be x2 and so on.</p>		
<p>Output: Status of files.</p>		
  		

Chapter 5: Testing

The testing was done on various stages of project development. Tools are the components which required major part of testing. Each and every tool is developed on different stages of life cycle and they are tested through different phases described below.

5.1 Individual tool testing

Each tool is tested individually when its developed. Testing of one tool won't affect result of another tool at this stage. Sample input and parameters are given to tool and generated output is tested using static methods.

5.2 Testing of relation among tools

Each tool generates output and that is given to another tool as input. When a new tool is introducing, it needs to be tested with all the existing tools. Ability to take input from another tool is tested in this stage.

5.3 Testing of tools in execution queue

There is a separate module for queue generation and execution. Each tool connected with start will take part in execution queue. The membership of tool in execution queue is tested in this phase.

5.4 Testing of deletion of tools

If any tool is deleted by user, then it leaves an empty hole in the queue. That hole must be filled by another remaining tools. A behavior of system after tool deleted is tested in this last phase of testing.

Chapter 6: Future Enhancement

The system is aimed for the complete automation of data scraping and web crawling. It already contains core tools but still some of them are not enough to achieve the complete automation. The following tools will be implemented in future.

- Caterpillar tool that provides huge amount of data to web crawler to be crawled
- Scraping from flash media
- Store text in any available database
- Automation of all the possible user actions
- Identification of controls on webpage and send actions
- Captcha detectors
- Captcha trading platform
- Multiselecting of tool
- Groupings of tool
- Interactive and animated chat bot to help user
- Multi-language support
- Operating system independence

Chapter 7: Appendix

7.1 Tools Used

- Microsoft Windows 8.1 Home (An operating system)
- Python Interpreter 3.6 (Open source OOP language)
- PyCharm 2017.3.3 (Python IDE)
- Selenium 3.9.1 (Library for automated web testing)
- PyQt5 (Library for GUI programming in python)
- WAMP module 2.1 Collection of
 - Apache 2.2.17 server
 - PHP 5.3.4 interpreter
 - MySQL 5.1.53 database
- Microsoft Word 2016 (Used for purpose of documentation)
- Microsoft Power Point 2016 (Used for purpose of presentation)
- smartsheet.com (Used to create Project Development Gantt Chart)

Chapter 8: Bibliography

8.1 Books

- Web Scraping with Python: Collecting Data from the Modern Web by Ryan Mitchell, 2015
- Python Web Scraping by Katharine Jarmul and Richard Lawson, 2017

8.2 Websites

- <https://www.riverbankcomputing.com/software/pyqt>
- <https://www.seleniumhq.org/docs/>
- <https://www.tutorialspoint.com/pyqt>
- <https://data-lessons.github.io/library-webscraping/05-conclusion>
- http://www.cis.uni-muenchen.de/~yeong/Kurse/ss09/WebDataMining/kap8_rev.pdf
- <https://www.promptcloud.com/blog/best-software-tools-acquire-data>
- <https://www.keycdn.com/blog/web-crawlers>

8.3 References

- [1]. McAfee, Andrew, and Erik Brynjolfsson. "Big Data: The Management Revolution." Harvard Business Review. Hank Boye, 01 Oct. 2012. Web. 08 Apr. 2016.
- [2]. Vargiu, Eloisa, and Mirko Urru. "Exploiting Web Scraping in a Collaborative Filtering-Based Approach to Web Advertising." Artificial Intelligence Research AIR 2.1 (2012): 44-54. Web. 13 Apr. 2016.
- [3]. Data Toolbar. Computer software. Data Toolbar. Vers. 3.1. DataTool Services Inc, 2013. Web. 8 Apr. 2016.
- [4]. Diebold, Francis X., On the Origin(s) and Development of the Term 'Big Data'(September 21, 2012). PIER Working Paper No. 12-037. Web. 13 Apr. 2016.
- [5]. Huynh, David, Stefano Mazzocchi, and David Karger. "Piggy Bank: Experience the Semantic Web Inside Your Web Browser." The Semantic Web – ISWC 2005 Lecture Notes in Computer Science (2005): 413-30. Web. 13 Apr. 2016.
- [6]. Laender, Alberto H. F., Berthier A. Ribeiro-Neto, Altigran S. Da Silva, and Juliana S. Teixeira. "A Brief Survey of Web Data Extraction Tools." ACM SIGMOD Record SIGMOD Rec. 31.2 (2002): 84. Web. 11 Apr. 2016.
- [7]. Banerjee, Ritu. "Website Scraping." Happiest Minds. N.p., Apr. 2014. Web. 11 Apr. 2016.
- [8]. Chen, Hsinchun, Roger H. L. Chiang, and Veda C. Storey. "Business Intelligence and Analytics: From Big Data to Big Impact." MIS Quarterly 36.4 (2012): 1165-188. Web. 8 Apr. 2016.