CHAPTER – I

# 1. INTRODUCTION

# 1.1 OVERVIEW OF THE PROJECT

Project is entitled as NSE WEBSITE AUTOMATION Get Text using RPA. In this project first the browser is being opened and NSE in that search box type Required Company Name will be displayed share value of the company. Using the Get Text they are all read by the excel file. Required company Share Values are read by the excel file.

For e.g. SBINShare Value are read by the excel file then Share Values are displayed.

RPA software are the perfect choice to put them into implementation for this specific use cases as the data that needs to be collected or scraped off the websites are already known for example, stock trading websites, futures trading websites, commodities trading websites, news and media sites (based on keywords). These websites can comfortably be scraped for the specific information of interest, summarize them and later can be presented to the required stakeholders for a call on what to do next with this information.

# 1.2BACKGROUND STUDY

# 1.2.1 Study On Existing System

Robotic process automation mimics human actions associated with a variety of business processes. In taking on predictable and repeatable tasks, software “robots” have the potential to vastly reduce costs, improve process quality and consistency, and enable greater scalability.

The existing system is a manual system. Here the products needs to save the information in the form of excel sheets or Disk Drives. There is no sharing is possible if the data is in the form of paper or Disk drives. The manual system gives us very less security for saving data. Excel sheet will manage the loosed data. It will display only a limited information. Searching of Required Company is very critical it takes lot of time.

**1.2.2 Drawbacks**

* They may able to generate different products in one document file.
* There is no interface to provide various items in the existing system.
* In the existing system Required Company is searched manually for e.g.VWAP, Face Value, Trade Volume, Trade Value and etc.
* The Share Values for the user is presented with a list of alternatives and the user chooses one of alternatives

**CHAPTER - II**

**2. SYSTEM ANALYSIS**

* 1. **STUDY ON PROPOSED SYSTEM**

VWAP, Face Value, Trade Volume, Trade Value and etc.Using Get Text Entered the Excel Sheet Company Name are Automatically Entered the NSE URL and Searched for Company Share Values.Read the Excel sheet of items. Using Orchestrator robots will be created. Using this robots the Scheduled and it will be running by Scheduled Time.

**Advantages**

* In one package multiple processes can be connected.
* In the process of execution details of Required Companies Share Values will be listed in excel sheet.
* By doing this the project time will be saved.
* Scheduling time can be created.

**2.1.1 Defining the Problem**

Computing paradigm has significantly changed the dimension of risks on user’s applications, specifically because the failures that manifest in the data centers. Nevertheless, these failures impose high implications on the applications deployed in virtual machines and, as a result, there is an increasing. The traditional way of achieving reliable and highly available software is to make use of fault tolerance methods at procurement and development time. This implies that users must understand fault tolerance techniques and tailor their applications by considering environment specific parameters during the design phase. However, for the applications to be deployed in the Cloud computing environment, it is difficult to design a holistic fault tolerance solution that efficiently combines the failure behavior and system architecture of the application.

**FEASIBILITY SYSTEM**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

**TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**2.2 SYSTEM SPECIFICATION**

**2.2.1 Application Specification**

**CLIENT SERVER ARCHITECTURE**

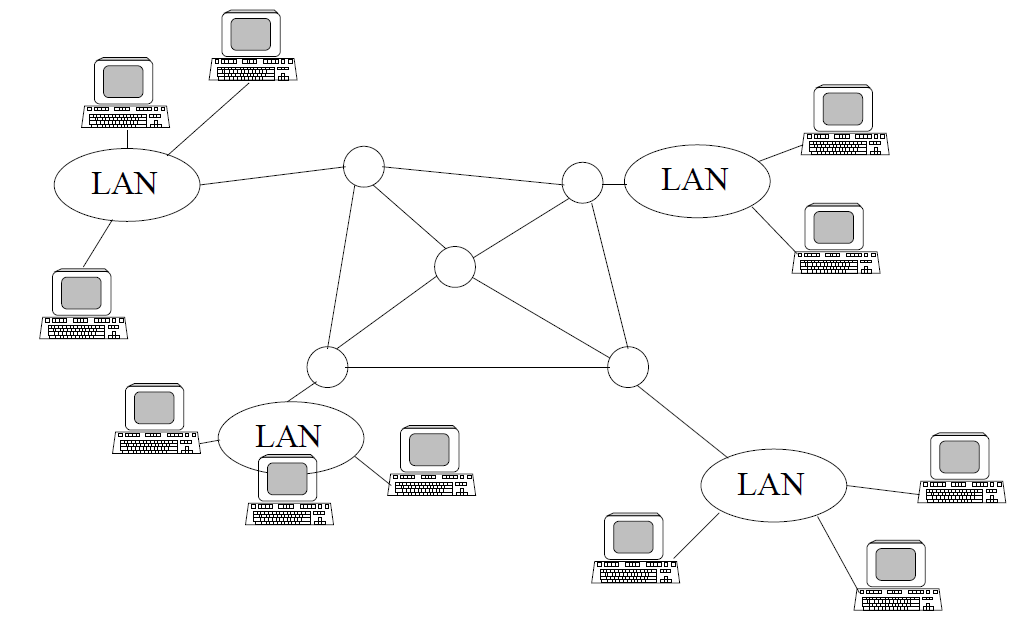
Client Server Architecture introduces a server (or an "agent") between the client and the server. The role of the agent is manifold. It can provide translation services (as in adapting a legacy application on a mainframe to a client/server environment), metering services (as in acting as a transaction monitor to limit the number of simultaneous requests to a given server), or intelligent agent services as in mapping a request to a number of different servers, collating the results, and returning a single response to the client.



**2.2.2 NETWORK SPECIFICATION**

A wide area network (WAN) is a network that covers a broad area (i.e., any telecommunications network that links across metropolitan, regional, national or international boundaries) using leased telecommunication lines. Business and government entities use WANs to relay data among employees, clients, buyers, and suppliers from various geographical locations. In essence, this mode of telecommunication allows a business to effectively carry out its daily function regardless of location.

The Internet can be considered a WAN as well, and is used by businesses, governments, organizations, and individuals for almost any purpose imaginable. Related terms for other types of networks are personal area networks (PANs), local area networks (LANs), campus area networks (CANs), or metropolitan area networks (MANs) which are usually limited to a room, building, campus or specific metropolitan area (e.g., a city) respectively.



2.2.3 Hardware Specification

The components of the computer such as electrical, electronic and mechanical units are known as the hardware of computer.

**Hardware Requirement**

* Processor : Pentium IV
* RAM : 4 GB.
* Hard Disk Drive : 10GB or More.
* Monitor : 15”CRT or LCD monitor.

# 2.2.3 Software Specification

It deals with defining software resource requirements and prerequisites that needed to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation of package and needed to be installed separately before the software is installed.

**Software Requirement**

* Tool : RPA in Uipath.
* Operating System : Windows 10.
* Application : Uipath

**2.2.4 SOFTWARE ENVIRONMENT**

**TOOL: RPA in Ui Path**

Robotic process automation mimics human actions associated with a variety of business processes. In taking on predictable and repeatable tasks, software “robots” have the potential to vastly reduce costs, improve process quality and consistency, and enable greater scalability.The term “robotic process automation” may conjure images of machines on a shop floor building cars or repairing aircraft engines, but the “robots” involved in robotic process automation aren’t robots in the physical sense. Rather, they are software that resides on a PC and interacts directly with business applications. By mimicking the way people use applications and following simple rules, software robots automate routine business processes, such as gathering and comparing data from different systems, adjusting insurance claims, or processing orders.

## FRAME WORK: NET FRAMEWORK

Visual studio .Net is a complete set of development tools for building ASP Web applications, XML Web services, desktop applications and mobile applications. Visual Basic .Net, Visual C++ .Net, and visual c# .Net all use the same integrated development environment (IDE) which allows them to share tools and facilities in the creation of mixed-language solutions.

**2.3 COST ESTIMATION AND SCHEDULING**

|  |  |
| --- | --- |
| **DESCRIPTION OF TASK** | **NO OF DAYS** |
| Abstract | 2 |
| Problem Statement | 2 |
| System Requirements | 2 |
| Design | 7 |
| Coding | 4 |
| Implementation | 8 |
| Testing | 7 |
| Reports | 11 |
| Deployment | 3 |
| Scope | 2 |
| **Total** | **48 days** |

Software cost is related to many variables such as Human, Technical, Environment and Effort applied to develop it. To estimate the effort needed for the software project, Function Point Analysis (FPA) and COCOMO model are used to predict the size and cost of developing the system. Function points are derived using an empirical relationship based on countable measures of software’s information domain and assessments of software complexity. COCOMO, Constructive Cost Model, is a good measure for estimating the number of person-months required to develop software. The COCOMO cost estimation formula is

E = c\*size k.

Where, E = effort in person-months. The effort measure helps to make estimates like the number of person months that will take for the project to execute. The size estimate is converted in to effort estimates.

c = 3.0 for semidetached mode

k = 1.12 in semidetached mode

Size = (SLOC)/1000=3.0

Thus the effort for making Android College Campus application is 10.31 pm.

D = a\*Eh

Where, d = Development time in chronological months

a = 2.5 in semi-detached mode

h = 0.38 in semi-detached mode

No of days worked = 48 days

1 day work = 8 Hours

Total no of hours = 384 Hours

Cost for 1 hour = ₹ 100

Total number of cost=Total number of hours\*cost for 1 hour = 384 \*100 =₹38,400/- Therefore total cost of the product is ₹ 38,400/-

**2.4 FINAL OUTLINE OF THE PROPOSED SYSTEM**

The project uses RPA in uipathit has the following features.

* Copy to document content.
* Open the browser in NSE
* Search for Required Company Name
* VWAP, Face Value, Trade Volume, Trade Value and etc.
* Using Get Text Already Entered in Excel File Company Name are Get and Searched by in a URL
* Get Share Values From the Excel Sheet.

Using Orchestrator robots will be created. Using this robots the Scheduled and it will be running by Scheduled Time.

**CHAPTER – III**

**3. DESIGN AND DEVELOPMENT PROCESS**

**3.1 Fundamental Design Concepts**

Although the degree of interest in each concept has varied over the year, each has stood the test of time. Each provides the RPA designer with a foundation from which more sophisticated design methods can be applied. Fundamental design concepts provide the necessary framework for “getting it right”.

A software design is a meaningful engineering representation of some RPA product that is to be built. A design can be traced to the customer’s requirements and can be assessed for quality against predefined criteria.

During the design process the software requirements model is transformed into design models that describe the details of the data structure, system architecture, interface, and components. Each design product is reviewed for quality before moving to the next phase of RPA development.

**PRIMARY KEY**

The Primary Key constraint uniquely identifies each record in a database table. Primary keys must contain unique values, and cannot contain null values. A table can have only one primary key, which may consist of single or multiple fields.

**FOREIGN KEY**

A Foreign key is a field (or collection of fields) in one table that refers to the primary key in another table. The table containing the foreign key is called the child table, and the table containing the candidate key is called the referenced or parent table.

**NOT NULL**

Not Null constraint enforces a column to NOT accept NULL values. This enforces a field to always contain a value, which means that you cannot insert a new record, or update a record without adding a value to this field.

**3.2 DESIGN NOTATIONS**

A Data Flow Diagram is a graphical technique that depicts information flow and transforms that are applied as data move from input to output. The DFD is used to represent increasing information flow and functional details. A level 0 DFD, also called a fundamental system model or a Context Model, represents the entire software elements as a bubble with input and output indicated by incoming and outgoing arrows respectively. Additional process and information flow parts are represented in the next level i.e., Level 1 DFD. Each of the processes represented at Level 1 are sub functions of overall system depicted in the Context Model. Any processes, which are complex in Level 1, will be further represented into sub functions in the next Level, i.e., in level 2.

**Basic DFD Symbols**

To Construct a Data Flow Diagram, we use

* Arrow
* Circles
* Open End Box
* Squares

**Arrow**

An arrow identifies the data flow in motion. It is a pipeline through which information is flown like the rectangle in the Flow Chart.

Data may flow a source to a processor and from a data store or process. An arrow line depicts the flow, with the arrowhead pointing in the direction of flow.

**Circle**

Circle stands for process that converts the data into information

A process represents transformation where incoming data flow is changed into outgoing flows.

**Rectangle**

A Rectangle defines a source or destination of system data. A source is a person or a part of organization, which enters or receives information from the system but is considered to be outside the context of the data flow model.

**Open End Box**

An Open End Box represents a data store, data at rest or temporary reposition of data.

A graphical picture of the logical steps and sequence involved in a procedure or a program is called a flow chart. Unlike detailed flow chart, Data Flow Diagram does not supply detailed description of the modules but graphically describes a system’s data how the data interact with the system.

Six rules for considering the Data flow Diagram

* Arrows should not cross each other
* Squares, circles and Data Store must have names
* Decomposed data flow squares and circles can have the same names.
* Choose meaningful names for data flow
* Draw all data flows around the outside of the diagram.
* Process should be named and numbered for an easy reference. Each name should be representative of the process.
* When a process is exploded into lower level details, they are numbered.
* The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each work capitalized.

A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out.

**CONTEXT FLOW DIAGRAM**

**LEVEL 0**

Check ShareVal

Share Val

Orchestrator

**Robot**

Studio

Package

Machine Key

**Fig 3.2.1 CONTEXT FLOW DIAGRAM**

**Fig 3.1 ContextFlowDiagram**

**3.2.1 DATA FLOW DIAGRAM**

**Level 1 DFD**

Copy to browser

OpenBrowser1

Studio

Excel Sheet

Move to

Excel

Select Particular Share

Value3

Orchestrator

Connect

NSE2

Robot

Get Text4

Stores in Excel Sheet

For Each Row 5

Get Text

Use

Send Mail

6

Move to Mail

Extract Value

ShareValue

Retrieves from Product Excel sheet

**Fig 3.2.1 Data Flow Diagram**

**3.2.2 STRUCTURE CHART**

A structure chart in software engineering and organizational theory is a chart, which shows the breakdown of the configuration system to the lowest manageable levels. It is used to show the hierarchical arrangement of the process in a structured program. A structure chart is a top-down modular design tool, constructed of squares representing the different process in the system and lines that connect them.

Product Search Using in RPA

Studio

Studio

Open browser

Robot

NSE

Share value

NSE Site

Get Text using Get Share Value

**Fig 3.2.2 Structure Chart**

**3.3 DESIGN PROCESS**

**3.3.1 Database Design**

**Content Table**

**Tablename** : Share\_table

**Abbreviation**: Com\_Tab

**Primary key** :Var\_Com\_Tab\_Values

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Size** | **Description** | **Sample Data** |
| Var\_CompanyName\_Share\_Details | Varchar | 20 | Select the Company Name | SBIN |
| Var\_CompanyName\_ShareValu\_  VWAP | Double | 10 | Select the Share Value for the VWAP | 294.48 |
| Var\_ComanyName\_ShareValue\_  Face Value | Double | 10 | Select the Share Value for the Face Value | 1.00 |
| Var\_CompanyNmae\_ShareValue\_  Traded Volume | Double | 10 | Select the Share Value for the Traded Volume | 2,33,77,581 |
| Var\_CompanyNmae\_ShareValu\_  Traded Value | Double | 10 | Select the Share Value the Traded Value | 68,816.59 |
| Var\_CompanyName\_ShareValue\_  Free Float Market Cap | Double | 10 | Select the Share Value for the Free Float Market Cap | 1,11,807.98 |

**3.3.2 Input Design**

The input design is the process of converting the user-oriented inputs in to the computer-based format. The goal of designing input data is to make the automation as easy and free from errors as possible.

UI Path Studio

Uipath studio in to view the drag and drop Process.

**Excel Sheet**

Excel Sheet create the n-number ofproduct items.

**Robot**

He would give the input and view the all product inExcel sheet output.

**3.3.3 Output Design**

A quality of output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other systems through outputs. In the output design it is determined how the information is to be displayed for immediate need. It is the most important and direct source information to the user. Efficient and intelligent output design improves the systems relationship the user and helps in decision-making.

**3.4 DEVELOPMENT APPROACH**

Ui path studio conversion was designed and developed based on the Waterfall Model. This model particularly expresses the interaction between subsequent phases.

Testing Uipath is not an activity, which strictly follows the implementation phase. In each phase of the Uipath development process, we have to compare the results obtained against that which is required. In all phases quality has to be assessed and controlled.

1. In the **Requirements Analysis** phase

* The problem is specified along with the desired service objectives (goals)
* The constraints are identified

2.     In the **specification** phase the system specification is produced from the detailed definitions of (a) and (b) above. This document should clearly define the product function.

3.      In the system and software **design phase,** the system specifications are translated into a software representation.

The software engineer at this stage is concerned with:

* Data structure
* Software architecture
* Algorithmic detail
* Interface representations

The hardware requirements are also determined at this stage along with a picture of the overall system architecture. By the end of this stage the software engineer should be able to identify the relationship between the hardware, software and the associated interfaces. Any faults in the specification should ideally not be passed ‘downstream’

4. In the **implementation and testing** phase stage the designs are translated into the software domain

* Detailed documentation from the design phase can significantly reduce the coding effort.
* Testing at this stage focuses on making sure that any errors are identified and that the software meets its required specification.

**CHAPTER-IV**

**4. TESTING AND IMPLEMENTATION**

**4.1 SYSTEM TESTING**

System testing is a type of testing to confirm that all code modules work as specified, and that the system as a whole performs adequately on the platform on which it will be deployed. System testing should be performed by testers who are trained to plan, execute, and report on application and system code. They should be aware of scenarios that might not occur to the end user, like testing for null, negative, and format inconsistent values.

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. A tester should be able to repeat the steps that caused an error. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

**4.1.1 Testing And Methodologies**

We begin the testing process by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free.

At the end of each testing day, we prepare a summary of completed and failed test. And the application is redeveloped and retested until every item is resolved. All changes and retesting are tracked through spreadsheets. Applications are not allowed to launch until all identification problem are fixed. Finally a report is prepared at the end of testing to show exactly what was tested and to list the final outcomes.

Our software testing methodology is applied in four distinct phases:

* Unit Testing

##### Integration Testing

##### System Testing

##### Usability Testing

##### Acceptance Testing

**INTEGRATION TESTING**

Integration testing, also known as integration and testing (I&T), is a software development process which program units are combined and tested as groups in multiple ways. In this context, a unit is defined as the smallest testable part of an application.

**UNIT TESTING**

Unit testing is designed to test small pieces of functionality rather than the system as a whole. This allows us to conduct the first round of testing to eliminate bugs before the other major tests (i.e.) each module is tested individually. Unit testing is testing changes made in an existing or new program to find out each module is found to be working satisfactorily. In this project based on the modules it will check one by one.

##### **SYSTEM TESTING**

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing. This project tested in various operating systems and their versions. And also its works fine with browsers. After windows 7 os this application is supports and working properly with their browsers. And also working in all MAC OS versions.

##### **USABILITY TESTING**

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

This project mainly concentrated with GUI process. Each module and functionalities are integrated and work together. All the modules are controlled in backend dash board with help of advanced concepts of programming language. It helps to improve the gui and processing the all functionalities.

##### **ACCEPTANCE TESTING**

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

This project is fully achieving the customer requirement without any problem. As the coding is completed according to the requirement, we have to test the quality of the software. Software testing is an empirical technical investigation conducted to provide stakeholders with information about the quality of the product. Testing is the process of executing a program or application with the intent of finding errors. Problem with software testing is that number of defects in a software product can be very large and defects that occur infrequently are difficult to find in testing. It is commonly believed that the earlier a defect is found the cheaper it is to fix it.

**TESTING OF OBJECTIVES**

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say,

* Testing is a process of executing a program with the intent of finding an error.
* A successful test is one that uncovers an as yet undiscovered error.
* A good test case is one that has a high probability of finding error, if it exists.

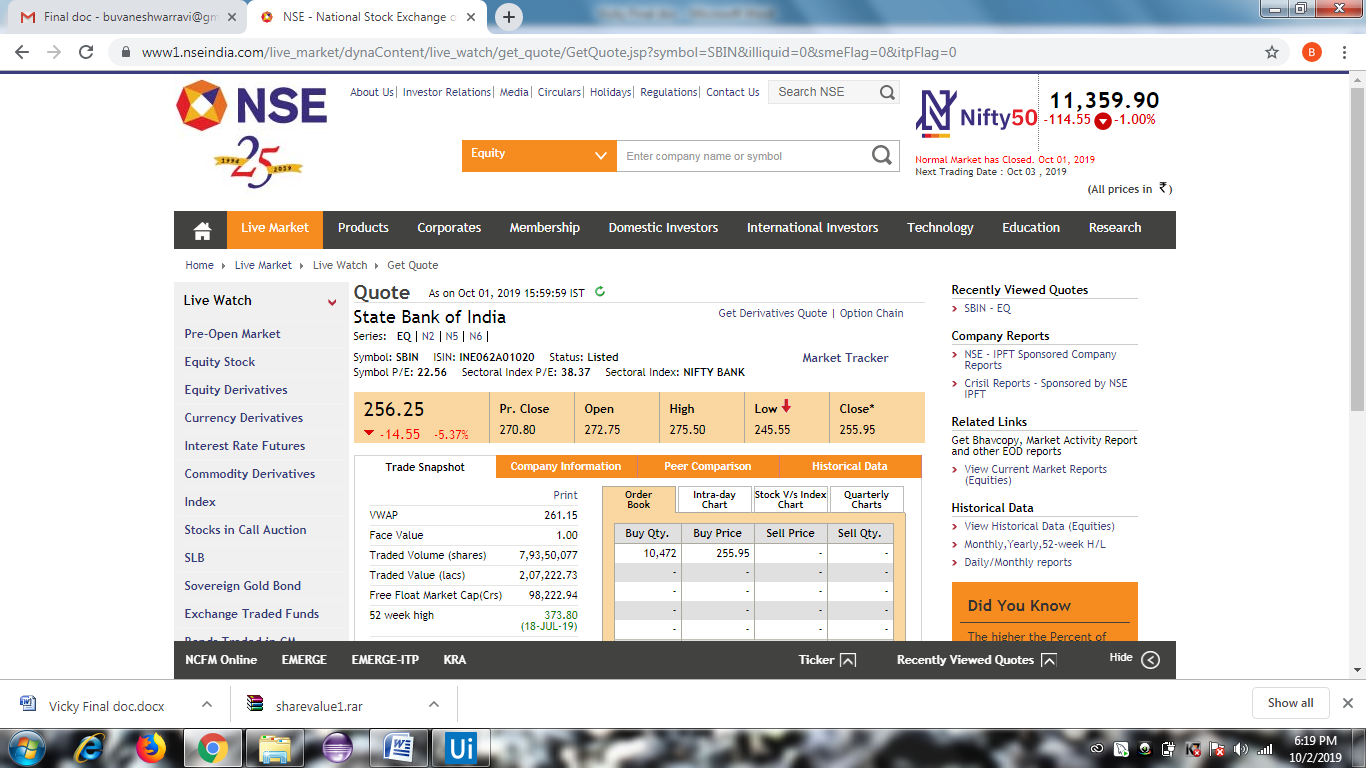
But there is one thing that testing cannot do (just quote a very famous sentences)”Testing cannot show the absence of defects, it can only show that software defects are presents”.As the test results are gathered and evaluated they begin to give a qualitative indication of the reliability of the software .If sever errors are detected, the overall quality of the software is a natural suspect. If, on the other hand, all the errors, which are encountered, are easily modifiable, then one of the two conclusions can be made:

* The tests are inadequate to detect possibly present errors.
* The software more or less confirms to the quality and reliable standards.

**TABLE OF TESTING**

**Image content**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Description** | **Test Data** | **Actual Result** | **Excepted Result** | **Final Result** |
| Img\_01 | Check for company name content of web page | Check the laptop excel sheet page | The laptop get text should be an text content | The laptop text should be an text content | Pass |
| Img\_02 | To check whether the text content of the web page | Check the laptop excel sheet page | The laptop get text should be an text content | The laptop text should be an not text content | Fail |
| Img\_03 | To check whether the text content of the web page | Check the mobile text page | The mobile Text should be an Text content | The mobile Text should be an Text content | Pass |
| Img\_04 | To check whether the text content of the web page | Check the text PDF page | The mobile Text should be an Text content | The mobile Text should be an not Text content | Fail |
| Img\_05 | To check whether the text content of the web page | Fan Text | The Fan Text should be an Text content | The Fan text should be an Text content | Pass |

****

**4.2 QUALITY ASSURANCE**

Greater emphasis on quality in organization requires quality assurance. To be an integral part of the information system development .The development process must include checks throughout the process to ensure that the final product meets the original user requirements. Quality assurance thus becomes an important component of the development process

It’s included in the industry standard (IEEE 1993) On the development process quality assurance process is integrated into a linear development cycle through validation and verification performed at crucial system development steps .The goals of the management is to institute and monitor a quality assurance program with in the development process. Quality assurance induces

* Validation of the system against requirements
* Checks for errors in design documents and in the system itself
* Quality assurance for usability

**4.2.1 Generic risks**

Risk identification is the systematic attempt to specify threats to the project plan (estimates the schedule resource overloading etc.). By identifying know and predictable risk the first step is to avoiding them. When possible and controlling them when necessary there are two types of risks

* Generic Risk
* Product specific risk

Generic risks are potential threats to every software project. Only those with a clear understanding of technology can identify product specific risk. The people and the environment that is specific to the project at a hand and to identify the product specific risk and the project the plan and the software statement of scope are examined and answer to the following question is developed.What special characteristics of this product may threaten the project plan one method for identifying risk is to create a risk item and checklists. The checklist can be used for risk identification and focus on some subset to know and predictable risk in the following sub categories.

* Product risk
* Risk associated with overall size of software to build or modified
* Business imparts
* Risk associated with constraints imposed with management
* Customer characteristics

Risk associated with sophisticated of the customer and developers ability to communicate with the customer in a timely manner.

**4.2.2 Security technologies and polices**

Any system developed should be secured & protected against possible hazards. Security measures are provided to prevent unauthorized access to database at various levels. Password protection & simple procedures to change the unauthorized access are provided to the users.

The user will have to enter the user name and password and if it is validated he can participate in auction. Otherwise if he/she is a new user he should get registered and then he can place an order

When he/she registered they should provide authentication through jpg files (like ration card Xerox, voter identity card Xerox). A multi layer security architecture comprising firewalls filtering routers encryption & digital certification must be assured in this project in real time that order details are protected from unauthorized access.

**4.3 SYSTEM IMPLEMENTATION**

Implementation is the stage of a project during which theory is turned into practice. Implementation phase begins after having the user acceptance of the new system developed. During this phase, all the programs of the system are loaded onto the user’s computer .After loading the system; training is given to the users. All the users are trained and the system is implemented successfully.

The project undergoes a versioning and release management before it is delivered to the client. It is a process of identifying and keeping track of different versions and release of the system. And the released product usually includes configured files defining how the release should be configured for particular installation.

The following factors are considered before implementation. Check all the component of the system is include. The appropriate version of required component and data are included. An installation program is created and the entire kit is delivered to the client.

The implementation stage involves following tasks

* Careful planning
* Investigation of system and constraints
* Design of methods to achieve the changeover
* Training of the staff in the changeover phase

**4.3.1 Implementation Procedures**

The implementation phase is less creative than system design. A system design may be dropped at any time prior to implementation, although it becomes more difficult when it goes to the design phase. The final report of the implementation phase includes procedural flowcharts, record layouts, and a workable plan for implementing the candidate system design into a operational design.

**4.3.2 User Training**

It is designed to prepare the users for testing & converting the system. There is several ways to trail the users they are:

* User manual
* Help screens
* Training demonstrations.

**USER MANUAL**

The summary of important functions about the system & software can be provided as a document to the user. User training is designed to prepare the user for testing and convening a system.

* Open http page
* Type the file name with URL home in the address bar
* home is opened existing user the type the username and password
* Click the submit button

**HELP SCREENS**

This features now available in every software package, especially when it is used with a menu. The user selects the “Help” option from the menu. The System success the necessary description or information for user reference.

**TRAINING DEMONSTRATION**

Another user training element is a training demonstration. Live demonstration with personal contact is extremely effective for training users.

**4.3.3 Operational documentation**

Documentation means of communication; it establishes the design and performance criteria of the project. Documentation is descriptive information that portrays the use and /or operation of the system. The user will have to enter the user name and password if it is valid he participate in auction. Otherwise if it is new user he needs to register

**DOCUMENTATION TOOLS**

Document production & desktop publishing tool support nearly every aspect of software developers. Most software development organizations spend a substantial amount of time developing documents, and in many cases the documentation process itself is quite inefficient. It is not use unusual for a software development effort on documentation. For this reason, Documentation tools provide an important opportunity to improve productivity.

**DOCUMENT RESTRUCTURING**

Creating document is far too time consuming. If the system works, we’ll live with what we have. In some cases, this is the correct approach. It is not possible to recreate document for hundreds of computer programs.

Documentation must be updated, but we have limited resources. It may not be necessary to fully re-document an application. Rather, those portions of the system that are currently undergoing change are fully documented.

The system is business critical and must be fully re-documented. Even in this case, an intelligent approach is to pare documentation to an essential mini

**CHAPTER V**

**5. CONCLUSION**

RPA software are the perfect choice to put them into implementation for this specific use cases as the data that needs to be collected or scraped off the websites are already known for example, stock trading websites, futures trading websites, commodities trading websites, news and media sites. These websites can comfortably be scraped for t RPA software are the perfect choice to put them into implementation for this specific use cases as the data that needs to be collected or scraped off the websites are already known for example, stock trading websites, futures trading websites, commodities trading websites, news and media sites (based on keywords). These websites can comfortably be scraped for the specific information of interest, summarize them and later can be presented to the required stakeholders for a call on what to do next with this information.

When an image is searched in GOOGLE there will be some contents below that are images In that contents there will be some PDF files. These for files are then taken into excel sheets. The world of computer fields is not static; it is always subject to be dynamic. Delivers a comprehensive fault tolerance solution to user’s applications by combining selected fault tolerance mechanisms ascertains the properties of a fault tolerance solution by means of runtime monitoring.

**SCOPE FOR FUTURE ENHANCEMENT**

There is scope for future development of this project. The world of computer fields is not static; it is always subject to be dynamic. To keep abstract of technical improvements, the system may be further refined. So, it is not concluded. Citizens may soon be able to validate the credentials of drivers and domestic helps before hiring them using infuture days identity documents such as the UID number can be checked online in real time. The service can also be used by companies that want to authenticate the records of their consumers and staff.

**BIBLIOGRAPHY**

**Textual Reference**

1. Elias M. Award, “System Analysis and Design”, Galgotia Publications, Second Edition.
2. Daniel, “**RPA IN UIPATH”,**Apress Publisher.
3. Paul Kimmel, “.Net-Unleased”, Sam’s TechMedia, Third Edition.
4. Roger S. Pressman,”Software Engineering”, Fourth Edition, 2005.

**Web Site Reference:**

<http://www.uipath.com>

<http://www.orchestrator.com>

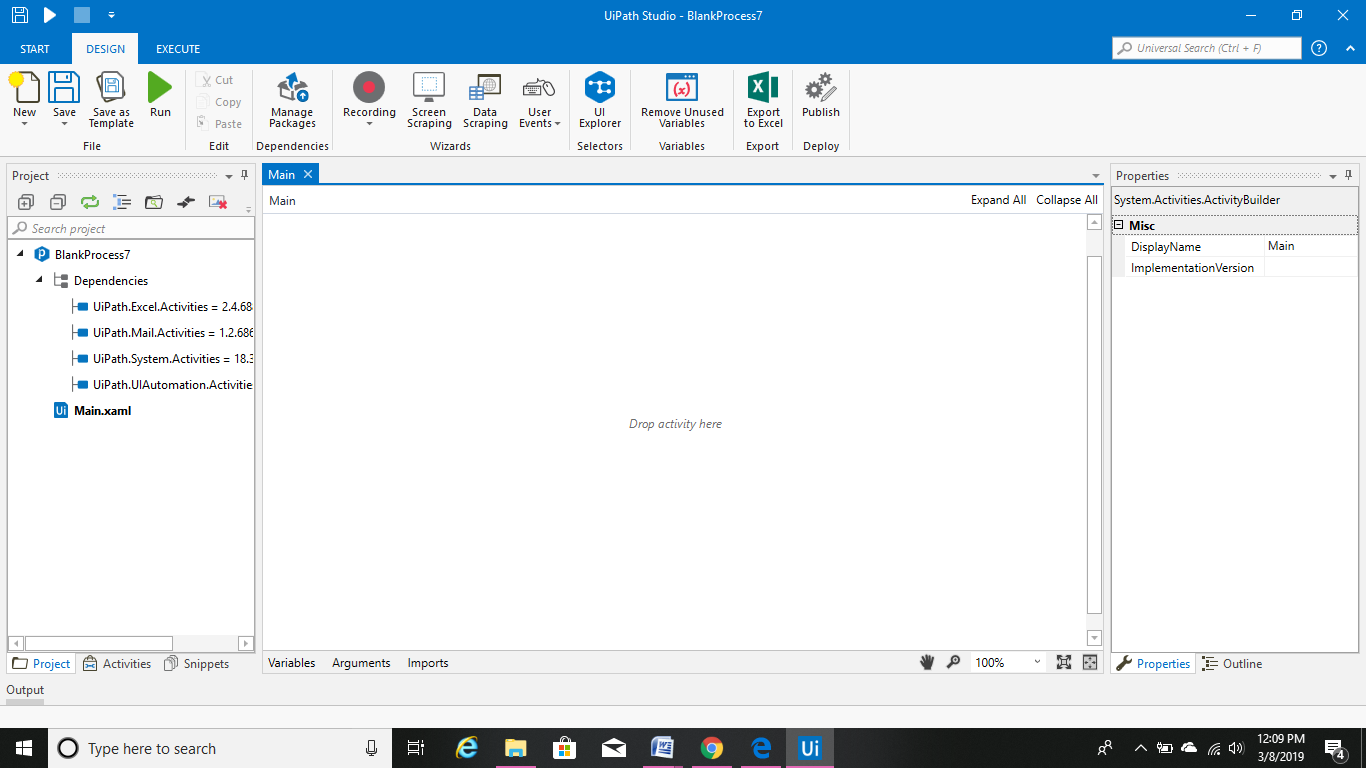
<http://www.forum.uipath.com>

https://www.academy.uipath.com

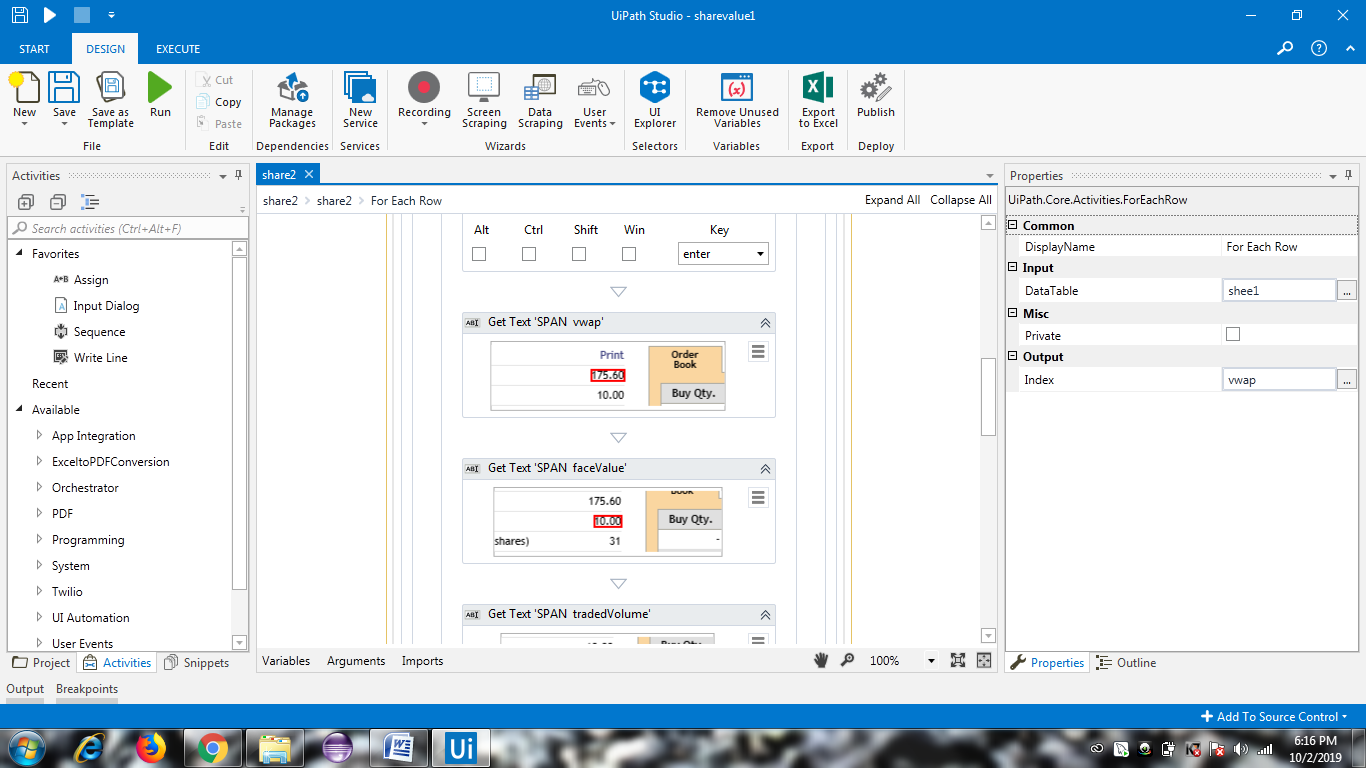
**ANNEXURES**

1. **INPUT & OUTPUT DESIGN**

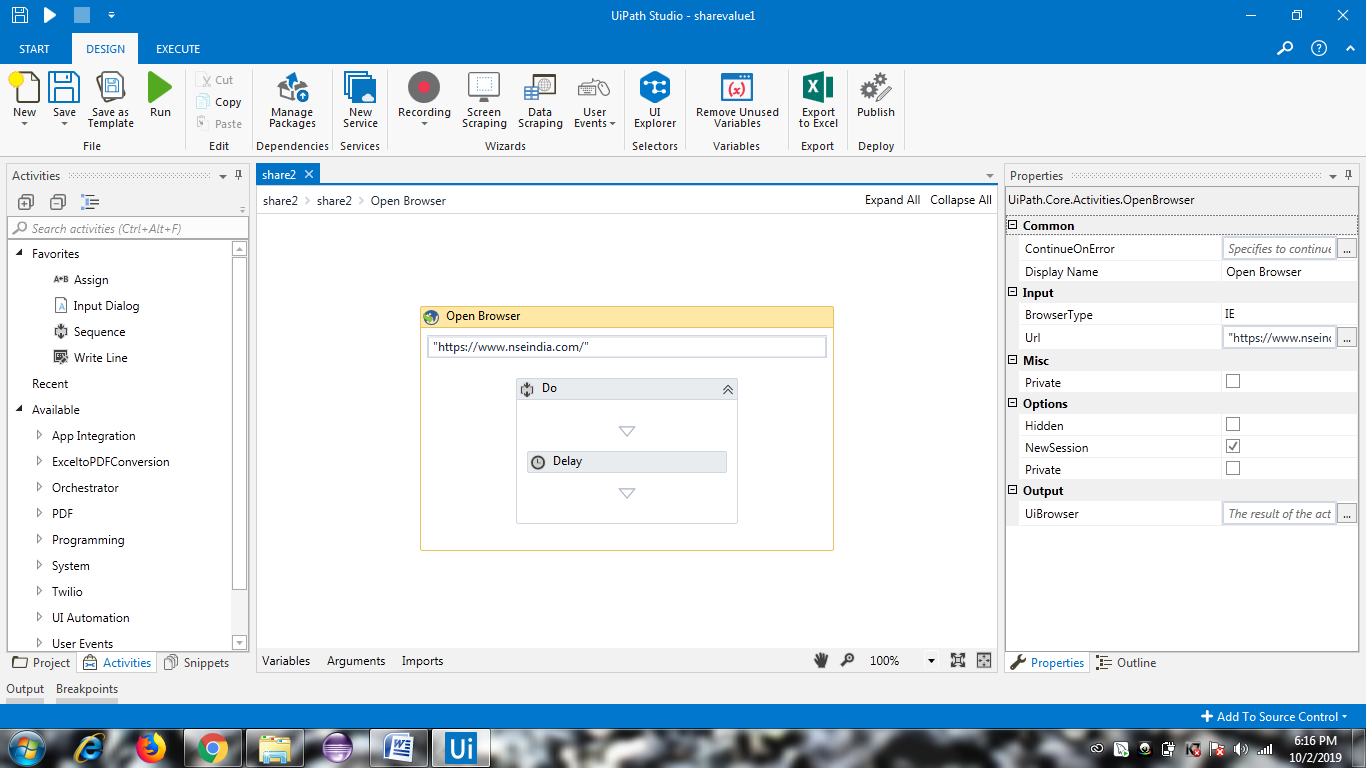
**SCREEN NAME: STUDIO RUN PAGE**

****

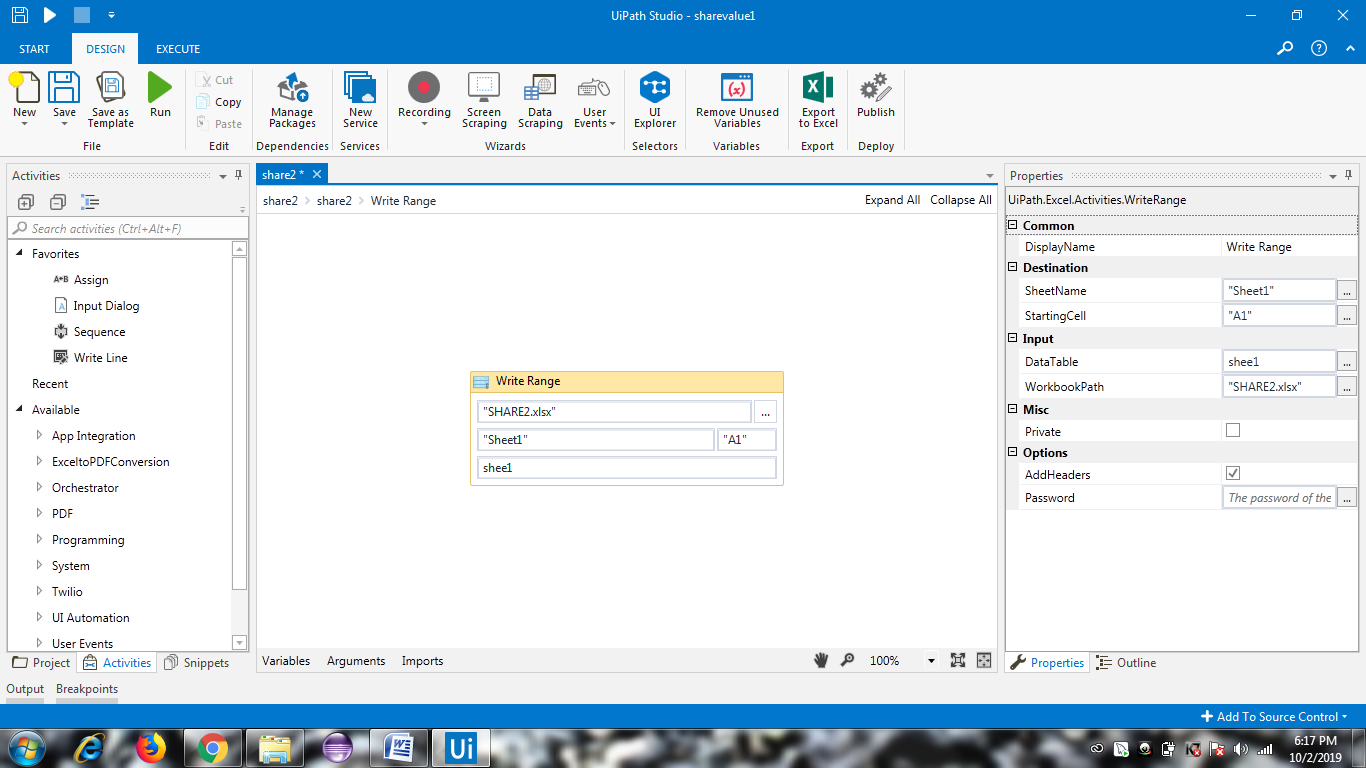
**SCREEN NAME:ACTIVITIES PAGE**

****

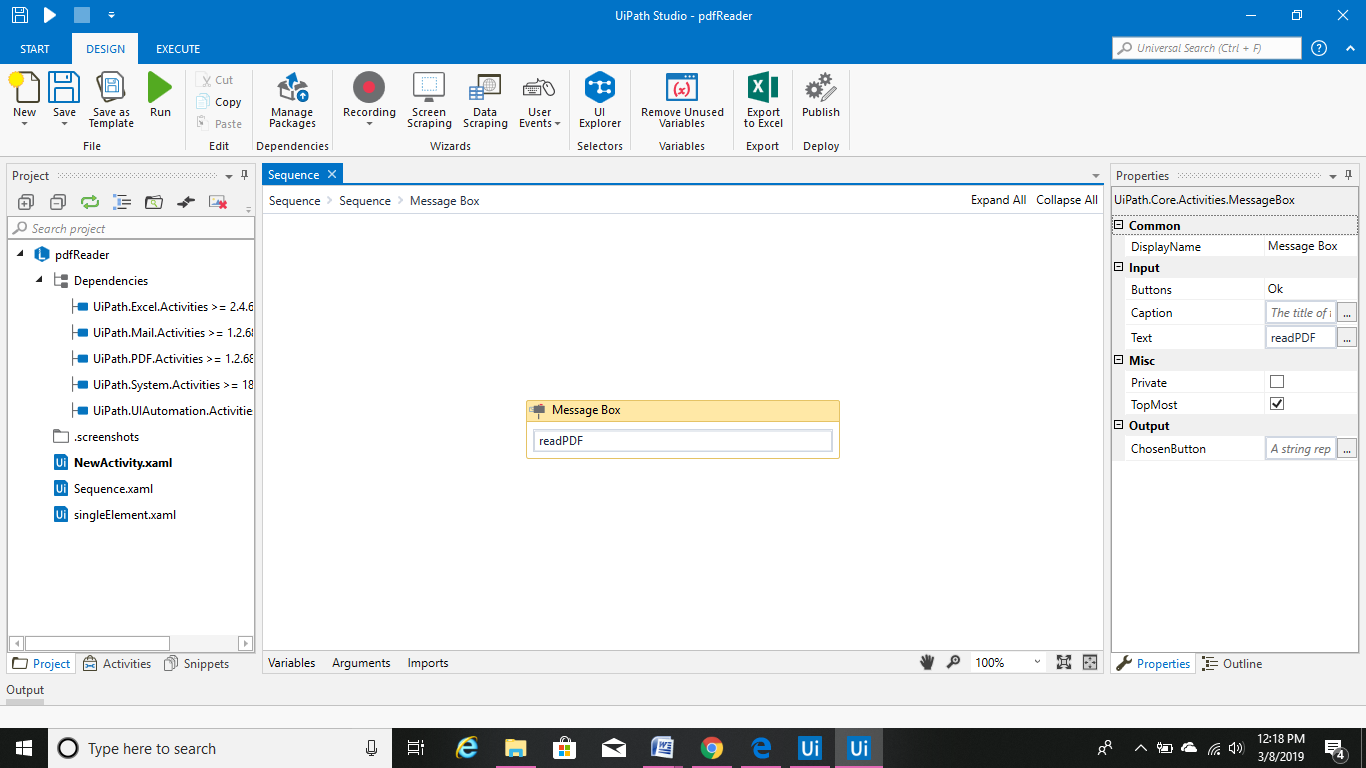
**SCREEN NAME: OPEN BROWSER ACTIVITY**

****

**SCREEN NAME: WRITE RANGE ACTIVITY**

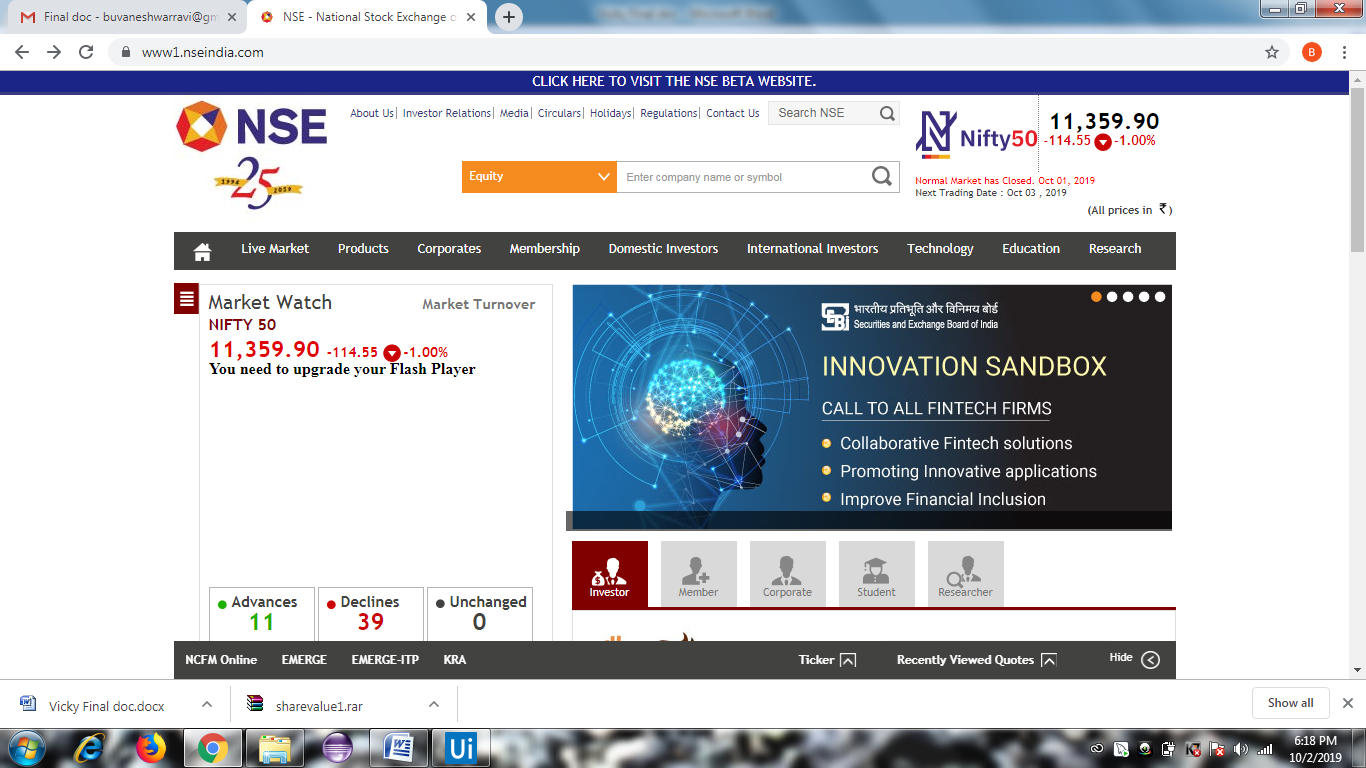
****

**SCREEN NAME: MESSAGE ACTIVITY**

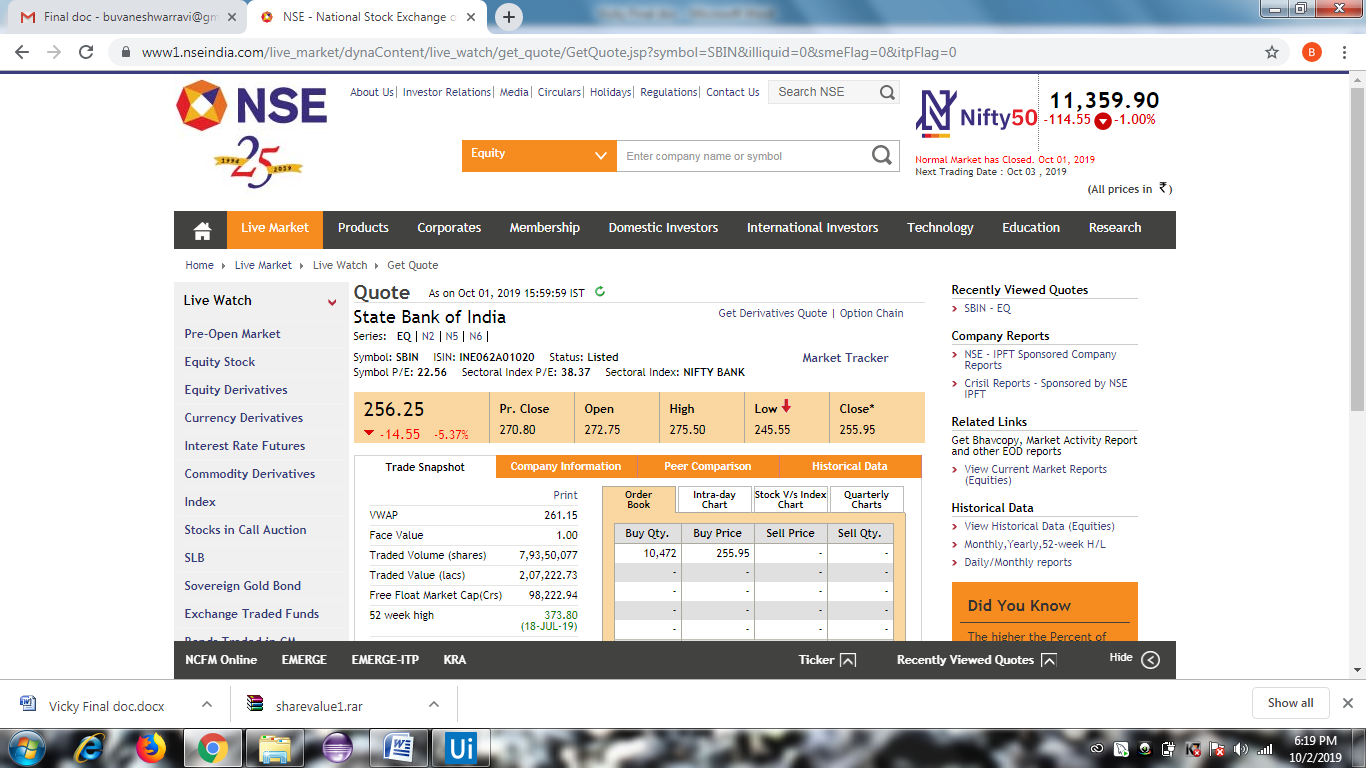
****

**OUTPUT FORM**

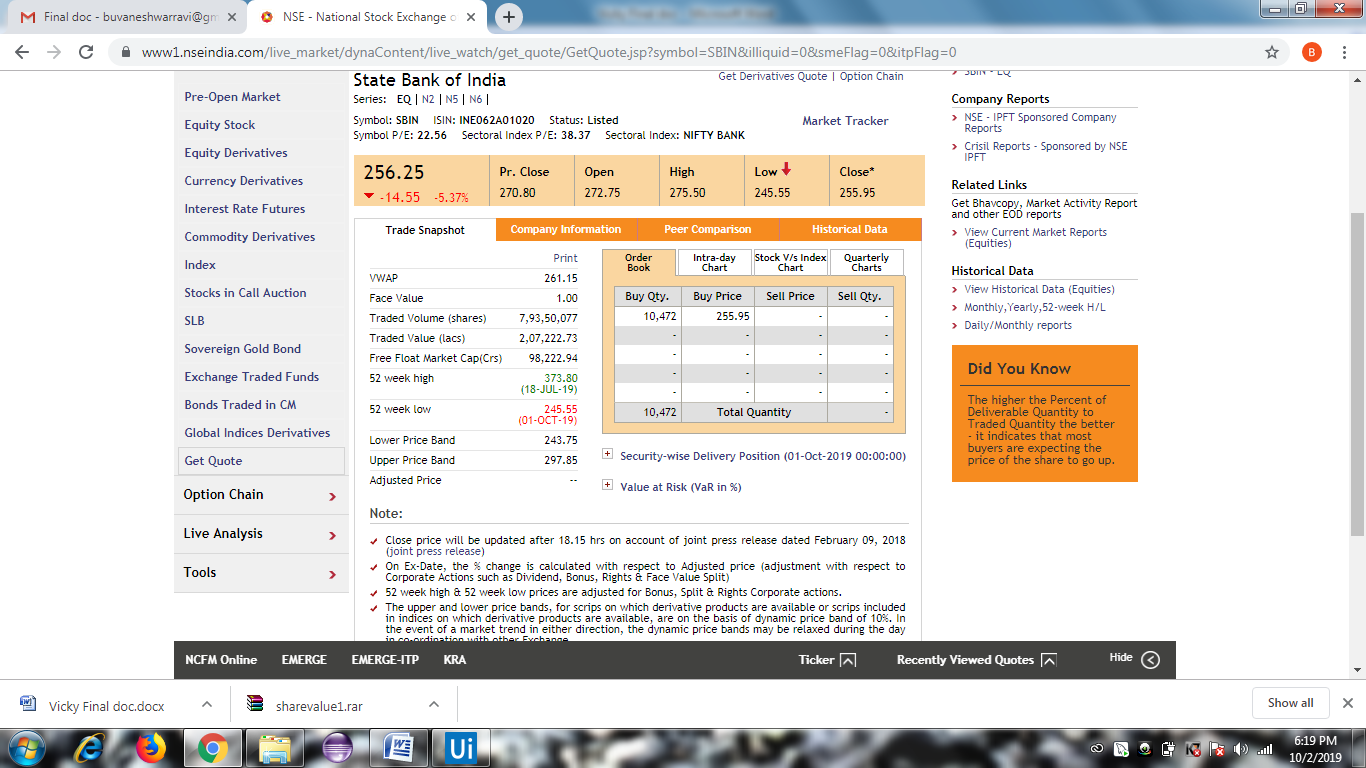
**SCREEN NAME: OPEN BROWSER**

****

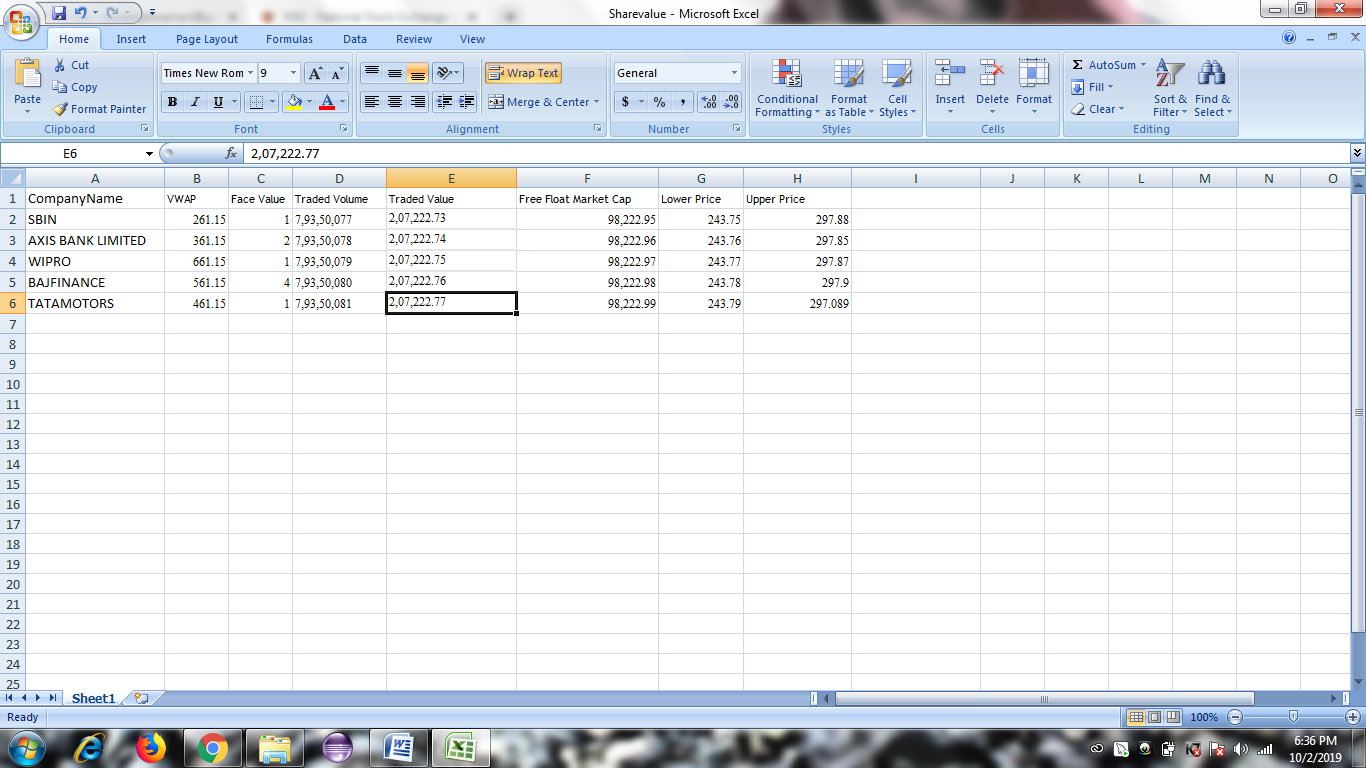
**SCREEN NAME: NSE**

****

**SCREEN NAME: MESSAGE BOX**

****

**Report**

****

**Sample Coding**

using System;

using System.Data;

using System.Configuration;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

using EmployeeFunction.DAL;

///<summary>

/// Summary description for ClsAreaMaster

///</summary>

publicclassClsAreaMaster

{

public ClsAreaMaster()

{

//

// TODO: Add constructor logic here

//

}

publicstaticDataSet ds;

publicint AreaId { get; set; }

publicstring AreaName { get; set; }

publicdecimal AreaSpan{ get; set; }

publicstring AreaDescription { get; set; }

publicstring AreaMap { get; set; }

publicint AreaTypeId{ get; set; }

publicstring AreaAbbervation { get; set; }

publicDateTime AreaIdentificationDate{ get; set; }

publicvoid InsertAreaMaster()

{

SqlParameter[] p=newSqlParameter[7];

try

{

p[0]=newSqlParameter("@AreaName",this.AreaName);

p[1]=newSqlParameter("@AreaSpan",this.AreaSpan);

p[2]=newSqlParameter("@Areamap",this.AreaMap);

p[3]=newSqlParameter("@AreaTypeId",this.AreaTypeId);

p[4] = newSqlParameter("@AreaDescription", this.AreaDescription);

p[5] = newSqlParameter("@AreaAbbervation", this.AreaAbbervation);

p[6] = newSqlParameter("@AreaIdentifiedDate", this.AreaIdentificationDate);

SqlHelper.ExecuteNonQuery(ConnectionString.Conn, CommandType.StoredProcedure, "Sp\_Insert\_AreaMaster", p);

}

catch (Exception ex)

{

thrownewArgumentException(ex.Message);

publicDataSet SelectAreaTYpeId()

{

try

{

string s = "sp\_selectAreaTypeMaster";

returnSqlHelper.ExecuteDataset(ConnectionString.Conn, CommandType.StoredProcedure, s);

}

catch (Exception)

{

throw;

}

}

publicDataSet SelectAreaId()

{

try

{

string s = "sp\_SelctAreaMaster";

returnSqlHelper.ExecuteDataset(ConnectionString.Conn, CommandType.StoredProcedure, s);

}

catch (Exception)

{

throw;

}

}

publicvoid UpdateAreaMaster()

{

SqlParameter[] p = newSqlParameter[5];

try

{

p[0] = newSqlParameter("@AreaName", this.AreaName);

p[1] = newSqlParameter("@AreaSpan", this.AreaSpan);

p[2] = newSqlParameter("@Areamap", this.AreaMap);

p[3] = newSqlParameter("@AreaTypeId", this.AreaTypeId);

p[4] = newSqlParameter("@AreaIdentificationDate", this.AreaIdentificationDate);

SqlHelper.ExecuteNonQuery(ConnectionString.Conn, CommandType.StoredProcedure, "tbl\_AreaMaster\_Update", p);

}

catch (Exception ex)

{

thrownewArgumentException(ex.Message);

}

}

publicDataSet GetAreaIdSelect(int AreaId)

{

try

{

SqlParameter p = newSqlParameter("@AreaId", AreaId);

ds = newDataSet();

ds = SqlHelper.ExecuteDataset(ConnectionString .Conn , CommandType.StoredProcedure, "sp\_SelectAreaMaster", p);

return ds;

}

}

**ABBREVIATIONS**

|  |  |
| --- | --- |
| COCOMO | Constructive Cost Model |
| CPU | Central Processing Unit |
| CSS | Cascading Style Sheet |
| DFD | Data Flow Diagram |
| FK | Foreign Key |
| GUI | Graphical User Interface |
| HTML | Hyper Text Markup Language |
| PK | Primary Key |
| RAM | Random Access Memory |
| SRS | System Requirement Specification |
| SQA | Software Quality Assurance |
| MVC | Model View Controller |