“TRADE MONITORING”

By

GROUP-8

Ashok Yaganti (68)

Venkata Sasidhar Kanumuri (35)

Mohamed Gharibi (23)

Yatheeswara Reddy Pulicherla (52)

Under the esteemed guidance of

Dr. LEE

UMKC

UNIVERSITY OF MISSOURI
KANSAS CITY

INDEX

1. Introduction :

Trade Monitoring is an application service for the people to exchange their products i.e. buying and selling shares through online. The main purpose of this online service is to provide easy exchange of shares between buyers and sellers.

1.1 . Requirements Specification :

- Trade Monitoring system requires the database to store the company's stock details and users account details and also past market details.
- Trade Monitoring system requires the external web services like Stock quotes, Stock Ticker and Finance charts.
- System should be user friendly and maintains all the stock information, payments and user accounts.

1.2. Software Interface

- **Client:** Web Browser.
- **Web Server:** Tomcat 7.0
- **Framework :**Microsoft Visual Studio 2010
- **Data Base Server:** Microsoft SQL Server 2008.

1.3. Non Functional Requirements:

1.3.1 Performance:

Performance is always a programming issue not related to any physical or hardware part of the system. Since we have gathered every piece of information is needed for the development and performance orientation.

1.3.2 Reliability:

Since the application is being developed through web server pages which is from the family of ASP.NET, the most famous efficient and reliable language, so it is reliable in every aspect until and unless there is an error in the programming side. Thus the application can be a compatible and reliable one.

1.3.3 Availability:

Making all the components available to the authenticated user is the major responsibility of the developer. Thus care should be taken in order to ensure the perfect available measures at times when needed.

1.3.4 Security:

Making all the components available to the authenticated user is the major responsibility of the developer. Thus care should be taken in order to ensure the perfect available measures at times when needed.

1.3.5. Maintainability:

The main aim of any application is to save the data efficiently and its perfect maintenance avoiding the mistakes. This application can be able to provide such maintenance of data through its functional modules that has been generated.

1.3.6. Portability:

This System must be flexible such that user with basic background in using computers can quickly use the system and learn to use the project. The system must have a friendly user interface.

1.3.7. Design Constraints:

For using our project they have to use the web site address and can login into the session.These constraints are related with the real world thatthe persons follow before the system has been installed. It is not often taken intoconsideration that the requirements that are not necessary in the project may be gathered butare neglected in the design.

1.4 External Interface Requirements:

1.4.1 User Interface:

The user interface requirements include the necessary Graphical User Interface that is required for the user to easily communicate with the system. An efficient and accepted system has a very good Graphical User Interface provided that makes it well accepted and efficient. The admin that work on the system should not be disgusted with the system appearance for working with the system.

1.4.2 System Interface:

The system interfaces include the interfaces that the system maintains. This mainly includes the module to module interaction and how the data is shared between differentmodules and maintain a proper maintenance of the database for the purpose of efficiencyand accuracy. This is done mainly to efficiently use the Personal Computer's resources efficiently and use the database more efficiently and quickly.

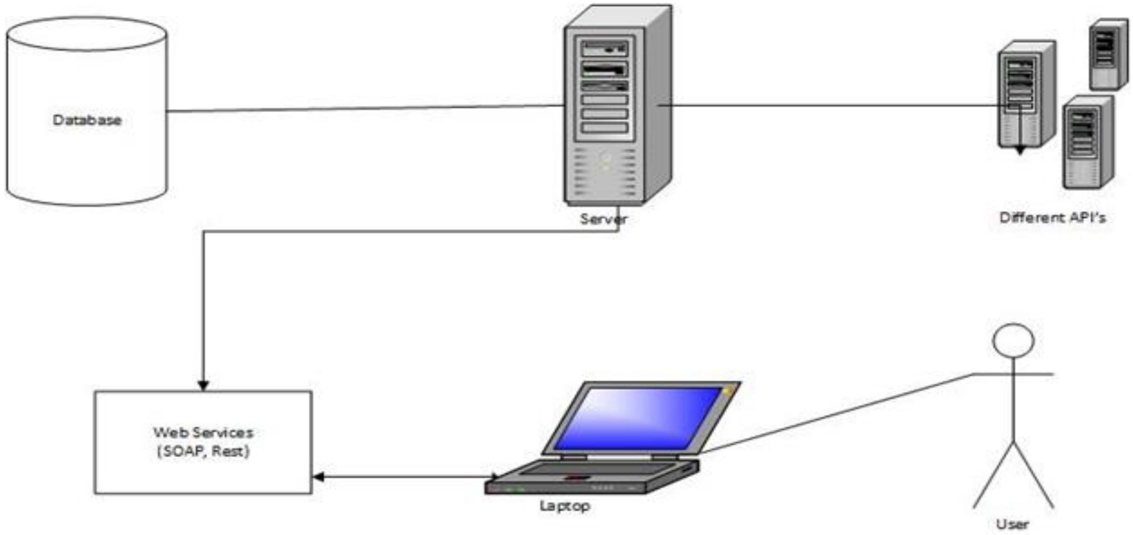
2. Design Overview

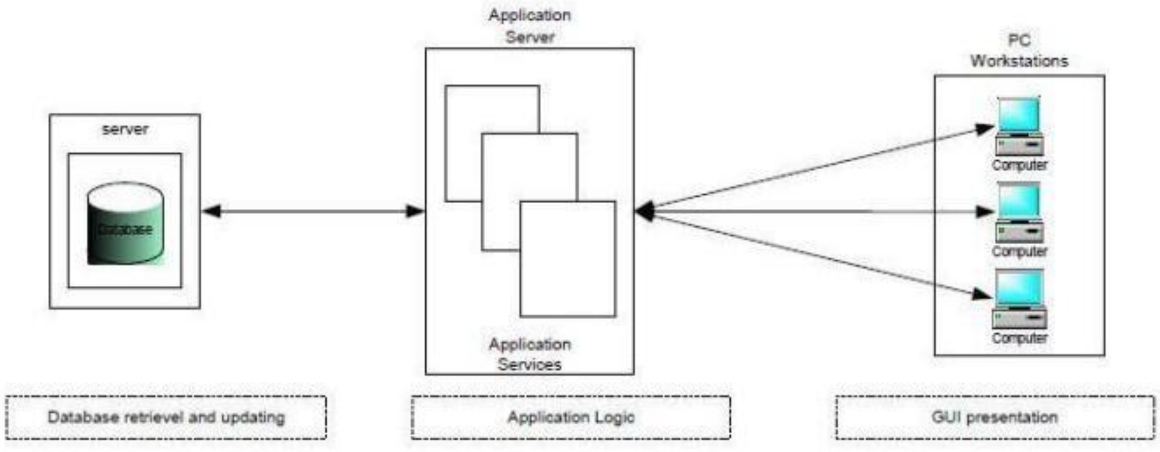
The SDD is a representation of a software system that is used as ainterfaceto communicate software design information.This document contains description of the high level architecture used to develop the system. Communicating at a high level, it will form the basis for the Software Detailed Design and implementation. However, the SDD itself is not sufficient to implement the code. It conveys the overall design of the system, the user interface design and higher level module design. Design details that are not be included in the SDD are:

Low level classes that will be used in the implementation. The full description of the implementation of each module is not needed, but the public modules that will be interfaced will be described.

Exact detailed description of interactions within each module.

2.1 System Architecture





This aims at decomposing the entire project into many modules, concurrent processes and data which will help in developing the software easily. A top level description of Project will be given, dividing it into its modules and explain their relation. The modules in the system contain public methods that run parallel processes and use data that has been modified during the system's active life period.

A module is a well-defined subsystem that is useful in various applications. Each module has a well-defined purpose. The modules can be individually compiled and can be stored individually in a library. These are easier to build. The entire project is decomposed into 3 modules.

- **Front module (GUI):**

  This module basically provides a graphical interface for those who cannot work with command line based applications.

- **Database module:**

  This module provides the main functionality for the information management system.

- **Middle module:**

  This module provides a connection between the front and the business tier. It includes a request module and is as an interface that can be used by the front- tier

- **Business module:**This module contains all the functionalities that validate, update and control the data entering into the database. Also this module is responsible to validate user.

2.2 Data Design

2.2.1 Data Tables:

Table:User_Registration

Column Name	Data Type(size)	Constraint
User ID	int(10)	Primary Key
User_name	Varchar(20)	Not Null
Password	Varchar(20)	Not Null
Email ID	Varchar(30)	Not Null
Address	Varchar(255)	Not Null
Mobile No	int(10)	Not Null
Occupation	varchar(20)	Not Null

Table: Login

Column Name	Data Type(size)	Constraint
User ID	int(10)	Primary Key
Password	varchar(10)	Not Null

Table: Shares Details

Column Name	Data Type(size)	Constraint
Company ID	Int(4)	Primary Key
Company Name	Varchar(20)	Not Null
Product_Name	varchar(20)	Not Null
Product Cost	Double(10)	Not Null

Table:Registartion_Pay

Column Name	Data Type(size)	Constraint
User ID	int(10)	Primary Key
Payment ID	Int (20)	Primary key
Payment Amount	double(255)	Not Null
Payment Date	Datetime	Not Null
Payment_Status	Varchar(20)	Not null

2.3. Class Diagram

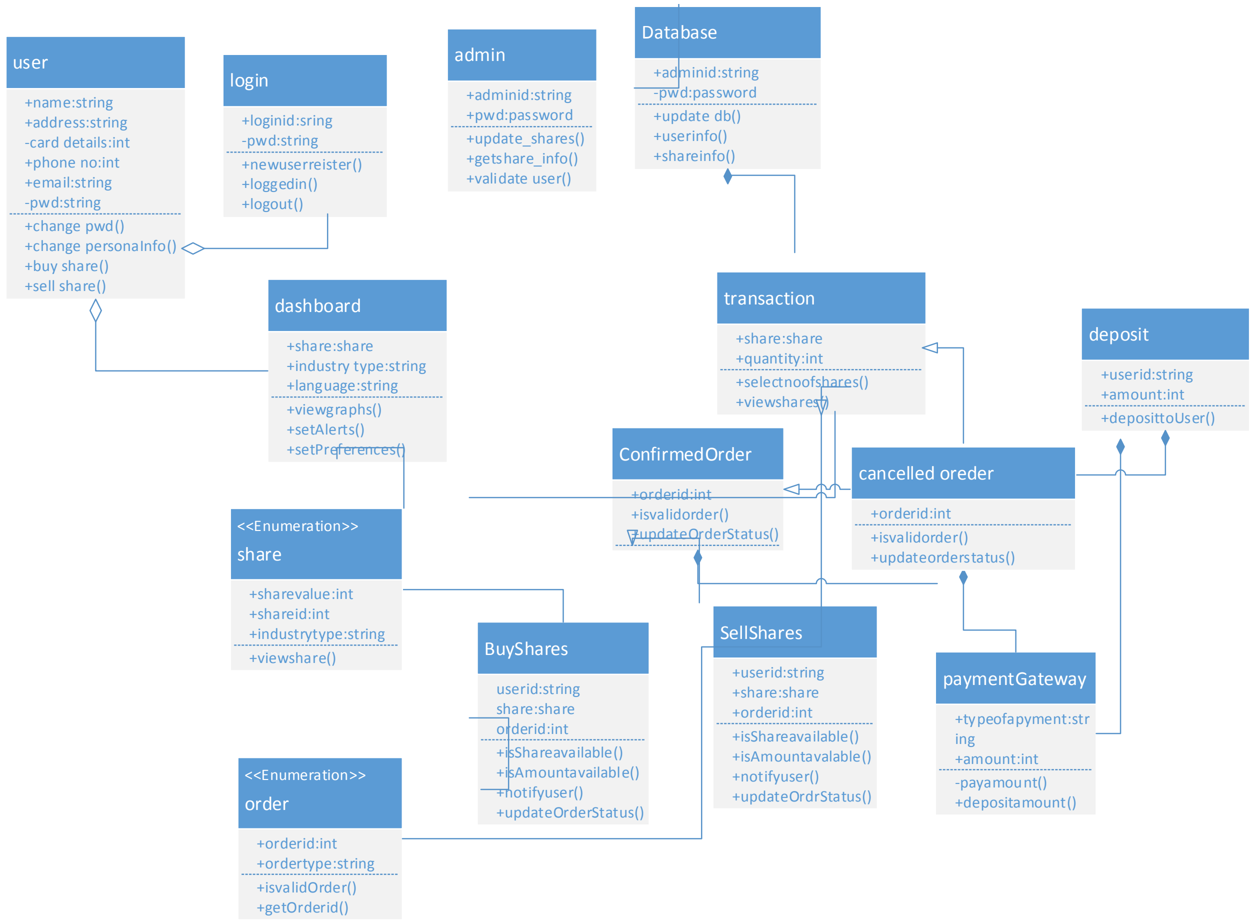
Used to represent various classes present in the system and relationships between them. Gives clear picture of the system and communications between classes present in the system.

- In this system User or the investor enters the system, if the user enters the site for the first time he/she has to Register with the system
- Then the Admin provides login details for the user after successful registration.
- Now the user can login with the system with his login credentials.
- Here he can view the shares of companies and view share values, analyze the shares.
- Then the user can buy or sell shares and make payments through Payment gateway.
- He can check if quantity of shares is available or not. If the user wishes to cancel the order he can or else continue to checkout.
- After buying or selling shares the user can continue to view shares or logout of the system.

Here is a list of classes involved and the diagram below represents the relationships involved between them.

- User
- Login
- Share
- Order
- Transaction
- Admin
- Database
- Buy shares
- Sell shares

- Payment gateway
- Confirm order
- Cancelled order
- Deposit



2.4. Use Case Diagram

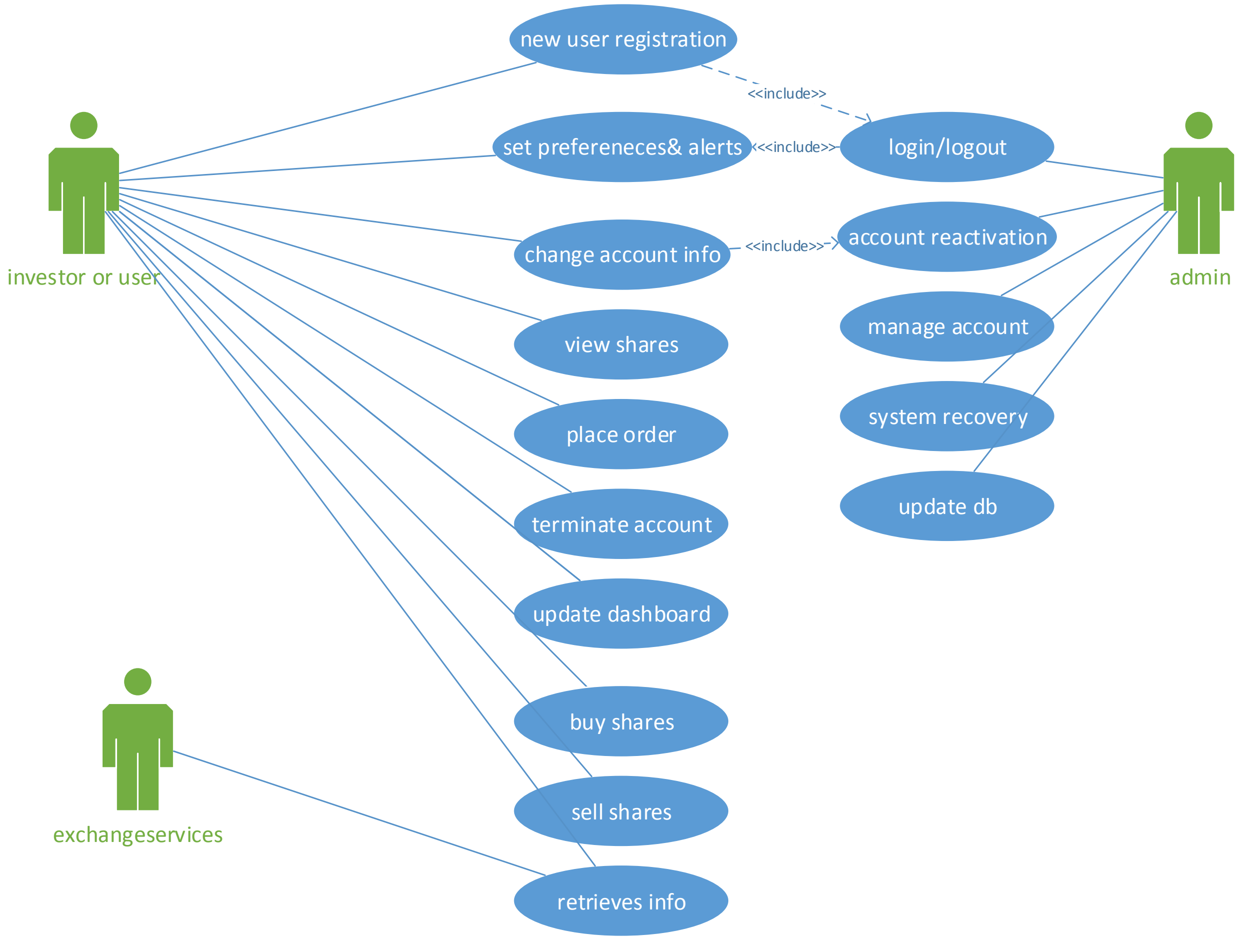
Use case diagram is the structure diagram which shows the communications among the use cases, actors in the system. Use cases are the elements present in the system

Actors involved are

- User
- Admin
- Exchange service

Usecases involved are

- New user registration
- Set preferences alerts
- Change account info
- View shares
- Login/logout
- Account reactivation
- Manage account
- System recovery
- Terminate account
- Retrieve amount
- Update database
- Place order
- View shares
- Sell shares
- Buy shares

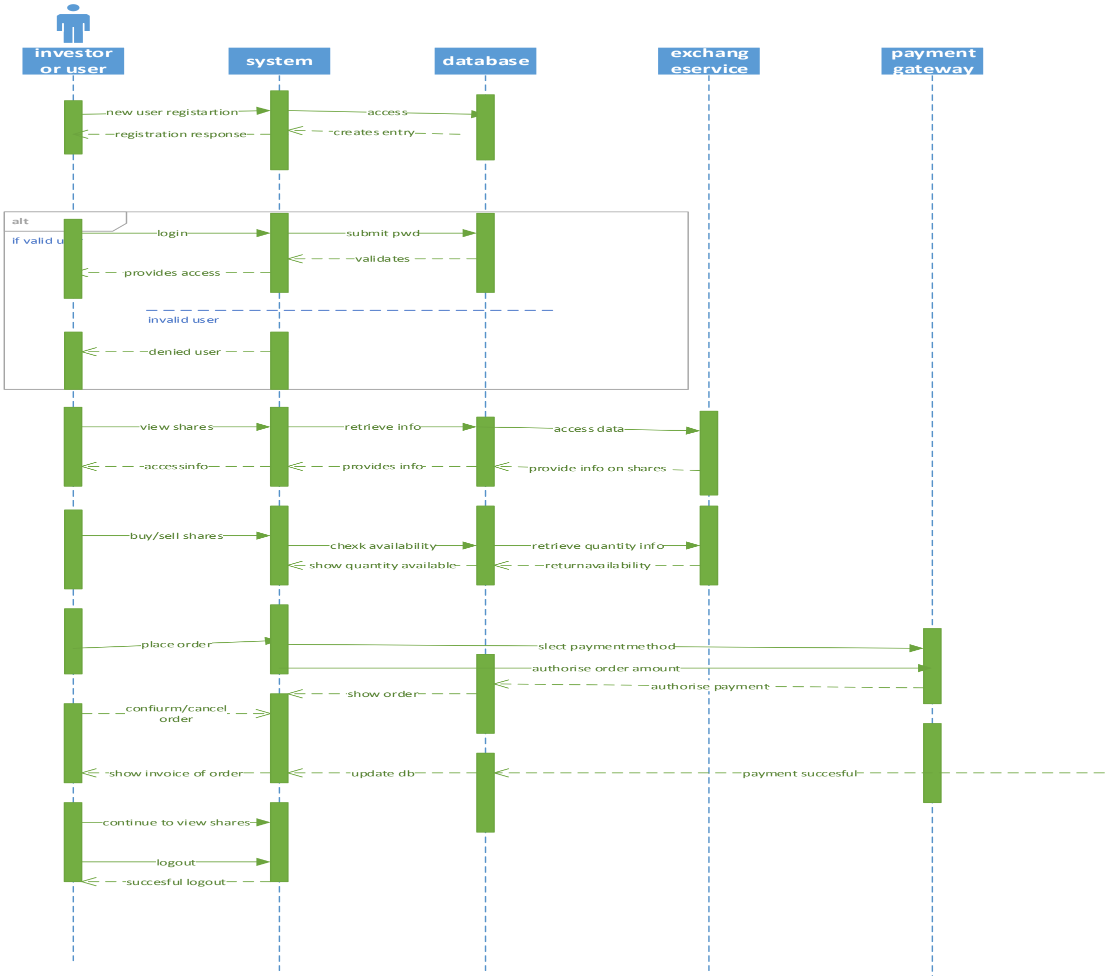


2.5. Sequence Diagram

This diagram show the interactions between various objects present in the system. The entire set of communications are represented by using messages with respect time.

This system contains the following objects

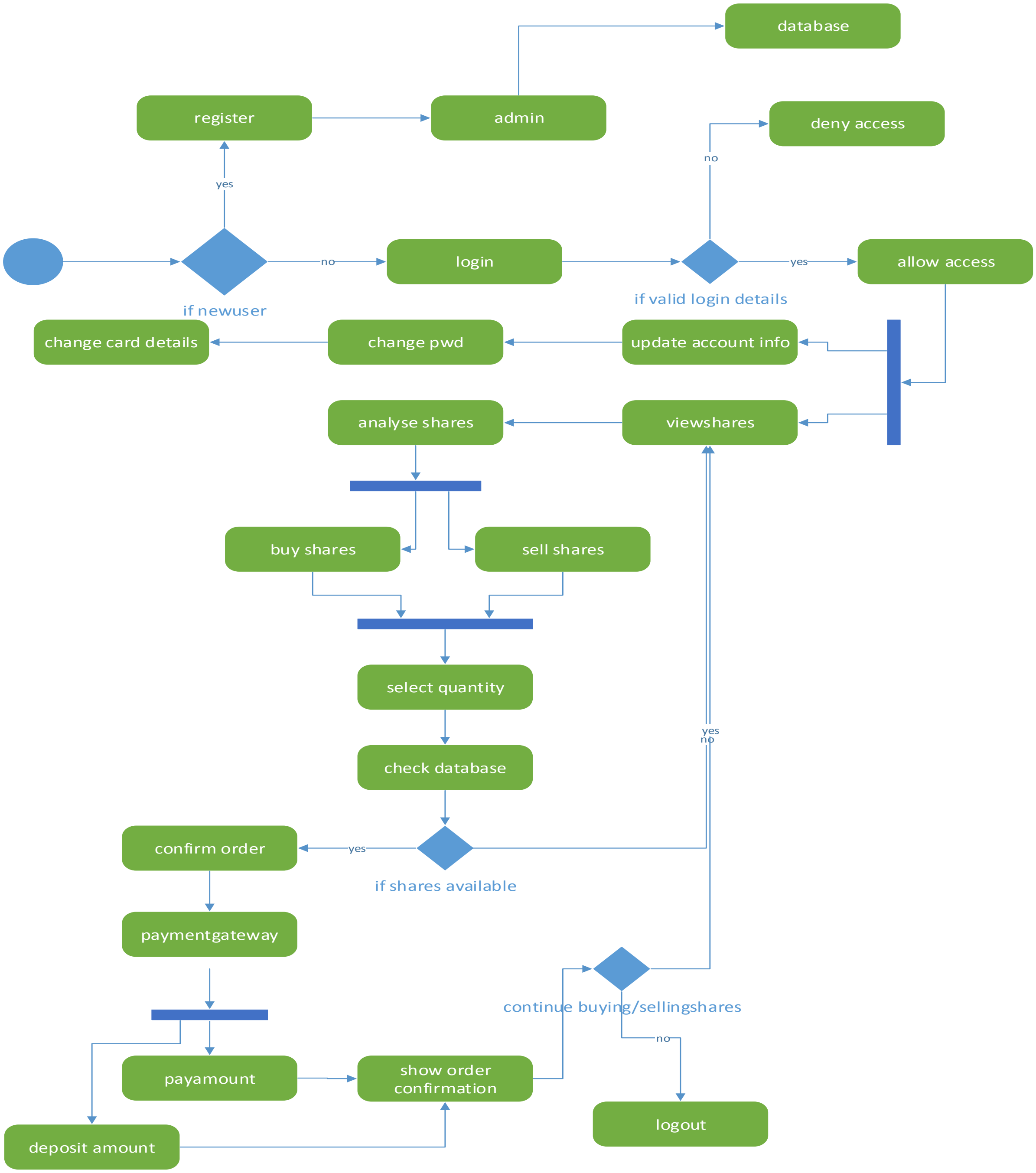
- User or the investor
- System
- Database
- Exchange service
- Payment gateway
- The user retrieves the share information from the exchange service provided in the system in the form of graphs, the messages between them is represented by the diagram below



2.6. Activity Diagram

This diagram shows the flow of activities involved in the system i.e.; entire workflow of the activities can be determined in this activity diagram.

The following diagram gives the entire representation of the workflow of our trade monitoring system

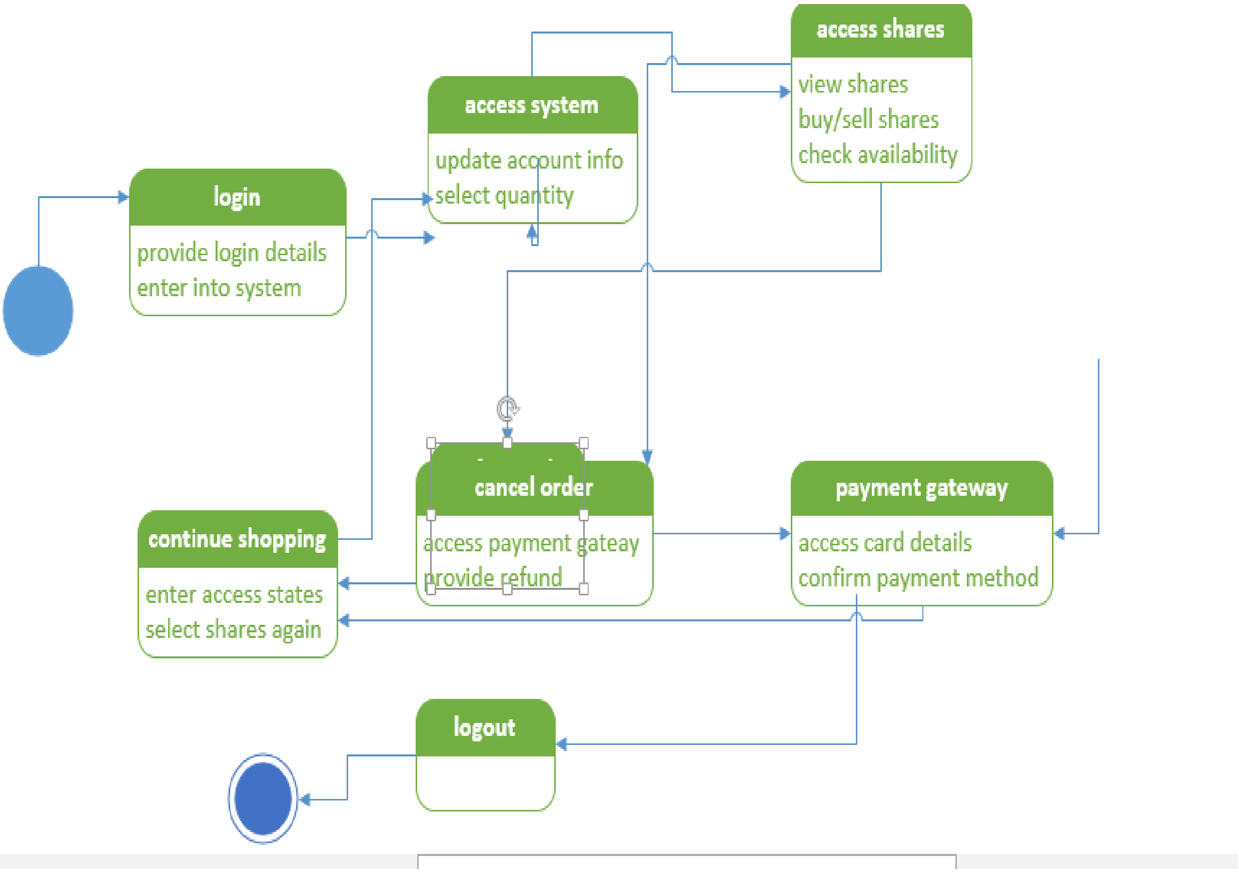


2.7. State Chart Diagram

This diagram represents the behavior of the system and describes various states present in the system.

The states involved in this system are represented in the following diagram

- Login
- Access system
- Access shares
- Continue shopping
- Cancel order
- Payment gateway
- Logout



3. Implementation Overview

3.1 Implementation of Web services :

We have successfully implemented the web services and also we have used some external services which are useful for our project.

External services :

1. http://finance.yahoo.com/d/quotes.csv?s=SAN.MC&f=snl1d1t1cv&e=.csv

2. http://ichart.finance.yahoo.com/b?s=" + symbols[i].Trim().ToUpper() + "& " + random.Next()

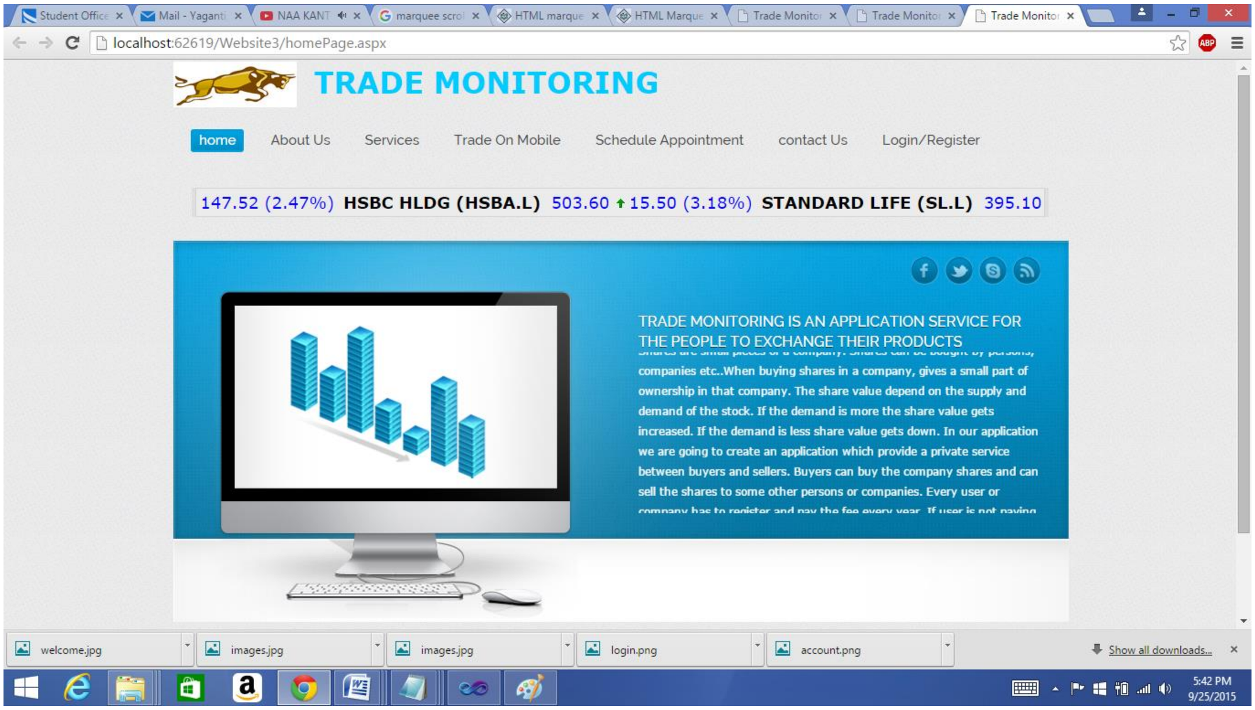
3.http://query.yahooapis.com/v1/public/yql?q=select%20*%20from%20yahoo.finance.quotes%20where%20symbol%20in%20(%22" + CNames + "%22)&env=store://datatables.org/

3.2. Implementation of User interface :

The user interface provided is used to perform all the actions specified above in the Interaction, Sequence, Activity diagrams. The user will be provided with a customizable interface .And the page design will be same for all the web pages.

Interface screens :

1. Home Page.



TRADE MONITORING

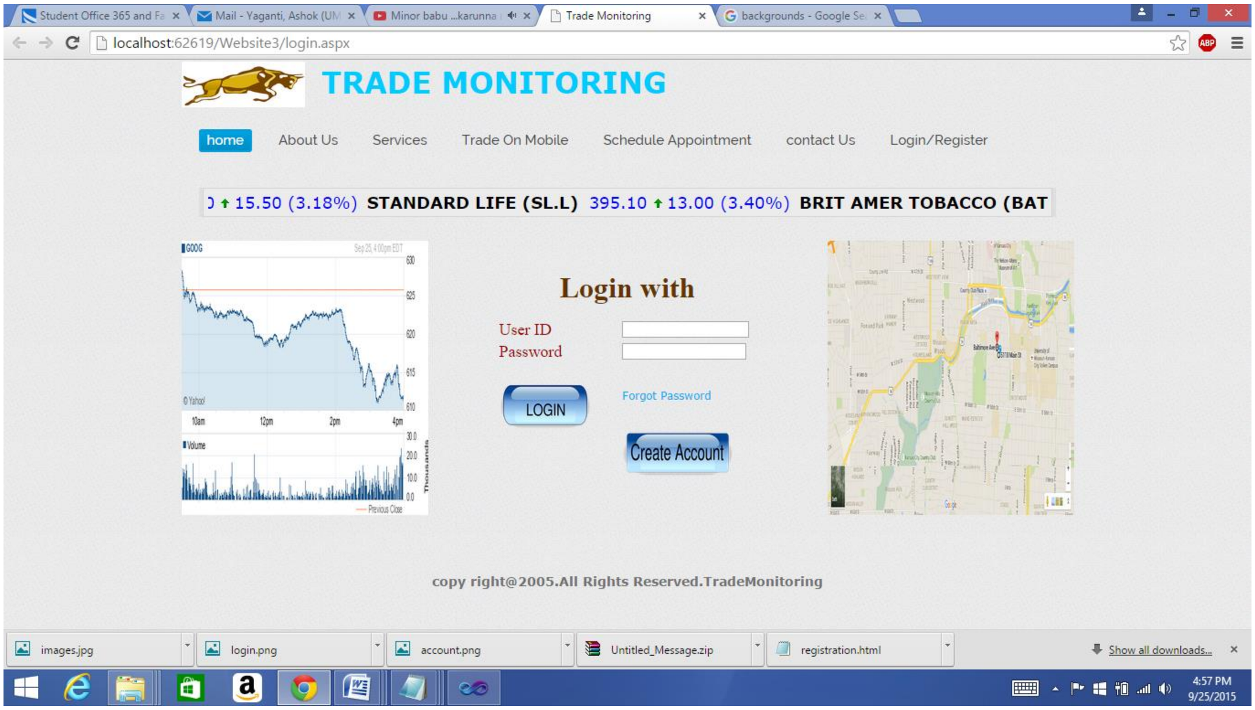
home About Us Services Trade On Mobile Schedule Appointment contact Us Login/Register

147.52 (2.47%) HSBC HLDG (HSBA.L) 503.60 ↑ 15.50 (3.18%) STANDARD LIFE (SL.L) 395.10

TRADE MONITORING IS AN APPLICATION SERVICE FOR THE PEOPLE TO EXCHANGE THEIR PRODUCTS

Shares are small pieces of a company. Shares can be bought by persons, companies etc..When buying shares in a company, gives a small part of ownership in that company. The share value depend on the supply and demand of the stock. If the demand is more the share value gets increased. If the demand is less share value gets down. In our application we are going to create an application which provide a private service between buyers and sellers. Buyers can buy the company shares and can sell the shares to some other persons or companies. Every user or company has to register and pay the fee every year. If user is not paying

2. Login Page :



) ↑ 15.50 (3.18%) STANDARD LIFE (SL.L) 395.10 ↑ 13.00 (3.40%) BRIT AMER TOBACCO (BAT

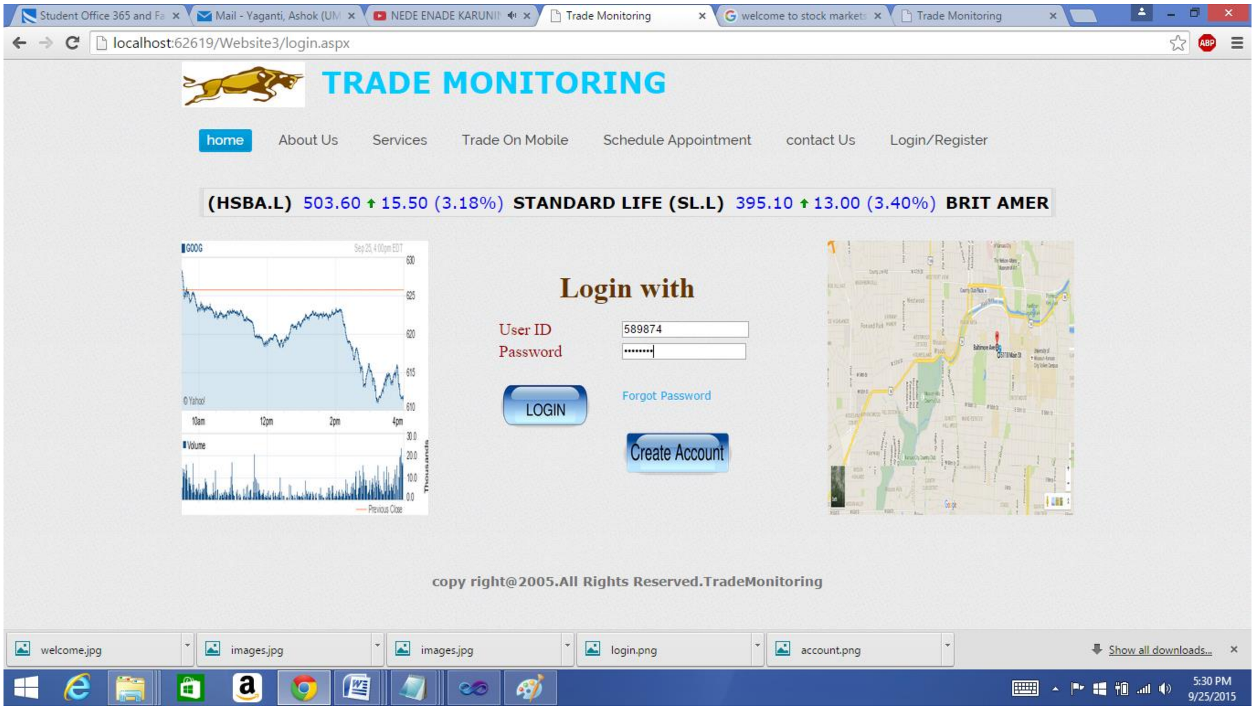
Login with

User ID

Password

copy right@2005.All Rights Reserved.TradeMonitoring

3. Input the User ID and Password.



(HSBA.L) 503.60 ↑ 15.50 (3.18%) STANDARD LIFE (SL.L) 395.10 ↑ 13.00 (3.40%) BRIT AMER

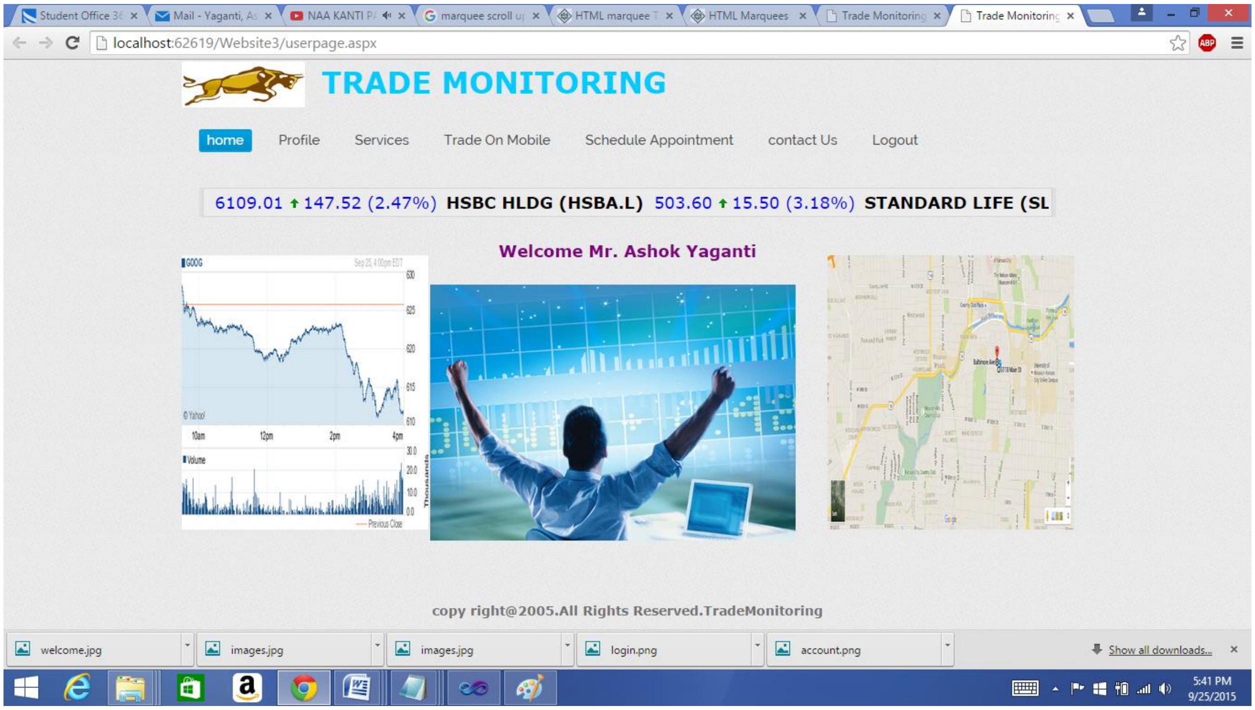
Login with

User ID 589874

Password ••••••••

copy right@2005.All Rights Reserved.TradeMonitoring

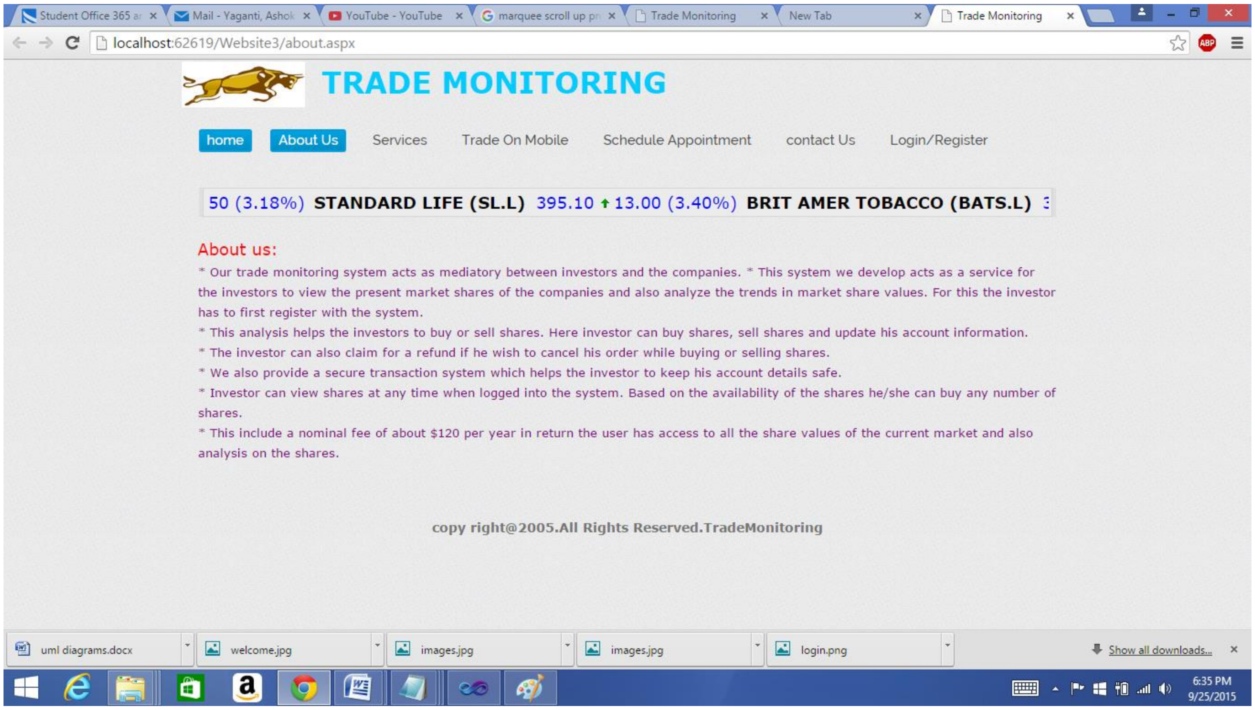
4. User home Page.



Welcome Mr. Ashok Yaganti

copy right@2005.All Rights Reserved.TradeMonitoring

5. About us Page.



About us:

* Our trade monitoring system acts as mediatory between investors and the companies. * This system we develop acts as a service for the investors to view the present market shares of the companies and also analyze the trends in market share values. For this the investor has to first register with the system.

* This analysis helps the investors to buy or sell shares. Here investor can buy shares, sell shares and update his account information.

* The investor can also claim for a refund if he wish to cancel his order while buying or selling shares.

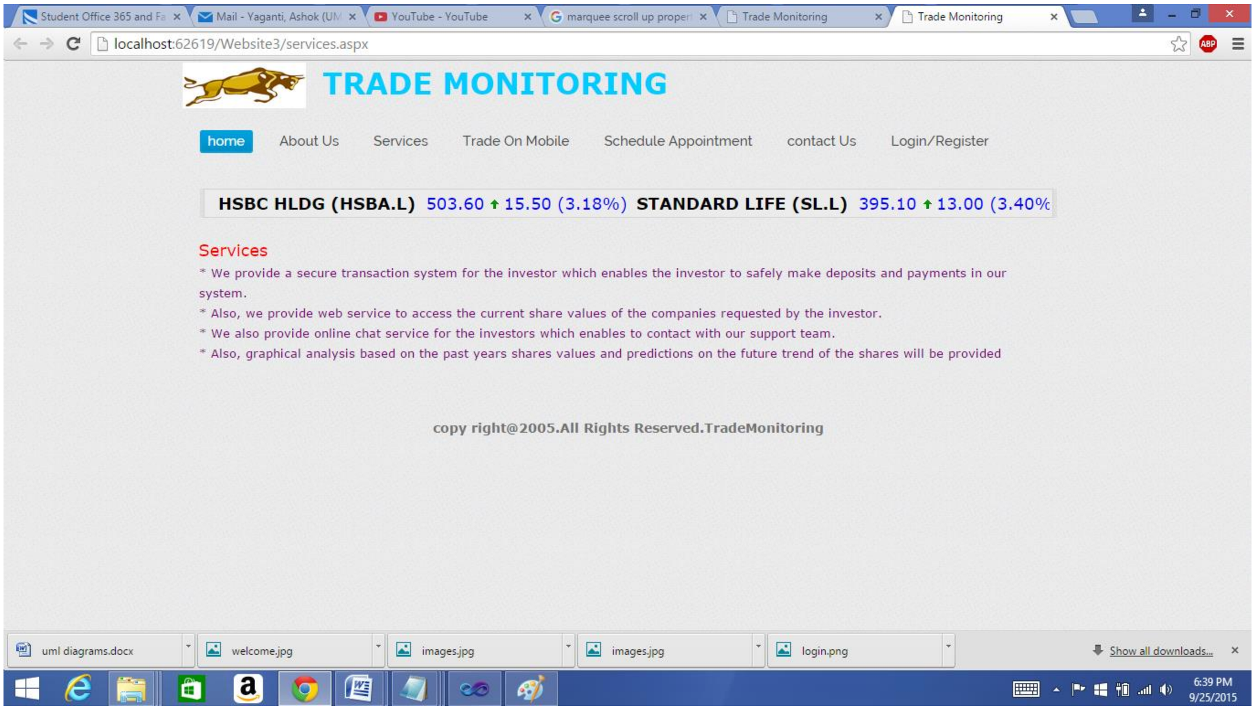
* We also provide a secure transaction system which helps the investor to keep his account details safe.

* Investor can view shares at any time when logged into the system. Based on the availability of the shares he/she can buy any number of shares.

* This include a nominal fee of about $120 per year in return the user has access to all the share values of the current market and also analysis on the shares.

copy right@2005.All Rights Reserved.TradeMonitoring

6. Services Page.

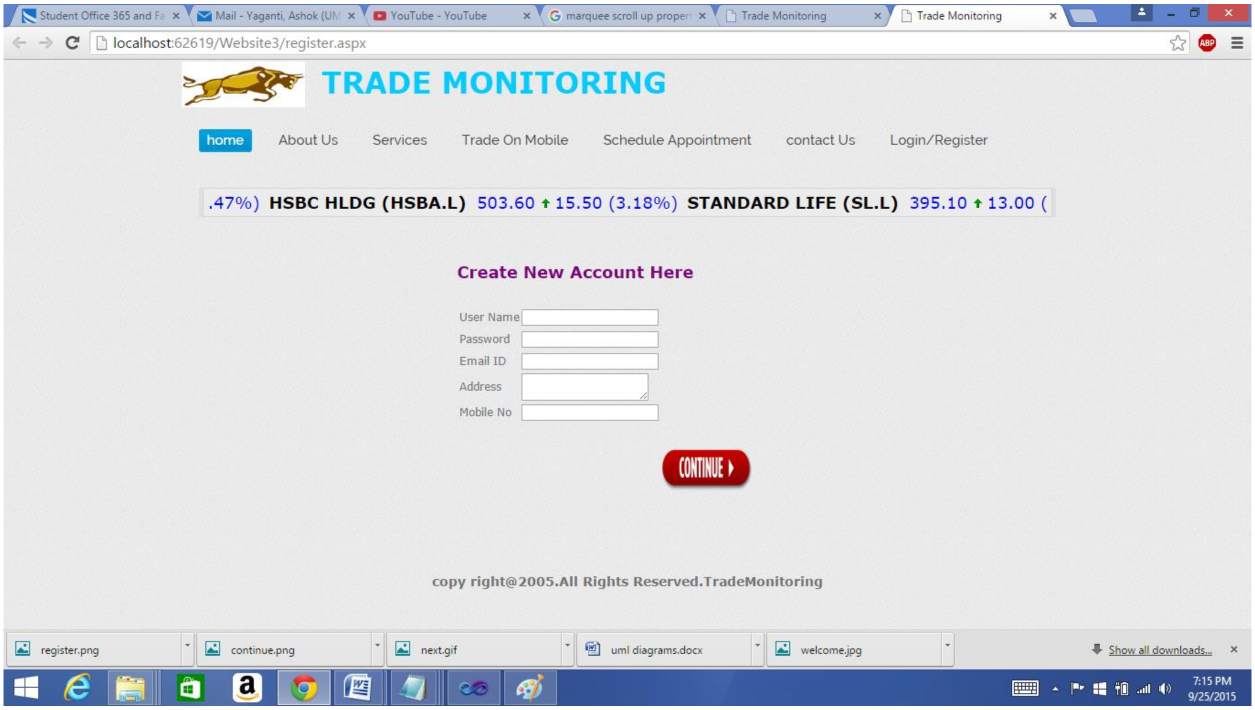


Services

* We provide a secure transaction system for the investor which enables the investor to safely make deposits and payments in our system.
* Also, we provide web service to access the current share values of the companies requested by the investor.
* We also provide online chat service for the investors which enables to contact with our support team.
* Also, graphical analysis based on the past years shares values and predictions on the future trend of the shares will be provided

copy right@2005.All Rights Reserved.TradeMonitoring

7. Create New Account.



Create New Account Here

User Name
Password
Email ID
Address
Mobile No

copy right@2005.All Rights Reserved.TradeMonitoring

4. Testing Overview

Test case:is a set of conditions or it is a document contain data and conditions using it the tester can determine whether its software or features are working as them originally supposed to. Test oracle is the point where we can say that this application or this software is passed or failed. Usually that could be only one use case and in other cases there might be many use cases for one application or one software and that known as heuristic. Often, test cases refereed as test scripts.

Usually, most of software has at least two test cases, one is positive test case and the other one is negative pose case. If one of the requirements has sub requirement or additional conations in this case also we need at least two test cases for that requirement one is positive and the other one is negative.

Any formal test case must characterized by a known input and a known output. IF there are no formal test cases then the test case should be written according to the perspectives and how do they expect the software or the application to work and execute. Hypothetical stories are being used in test cases to help the tester to think about complex problem about this application or software.

Typical test cases format and parameters:

- The ID of test case
- Description of test case
- The order of execution number
- Requirements that are related to the test case
- Depth
- Categories of the test
- Author
- Check box whether the test can run or it is already automated
- Pass or Fail of the test

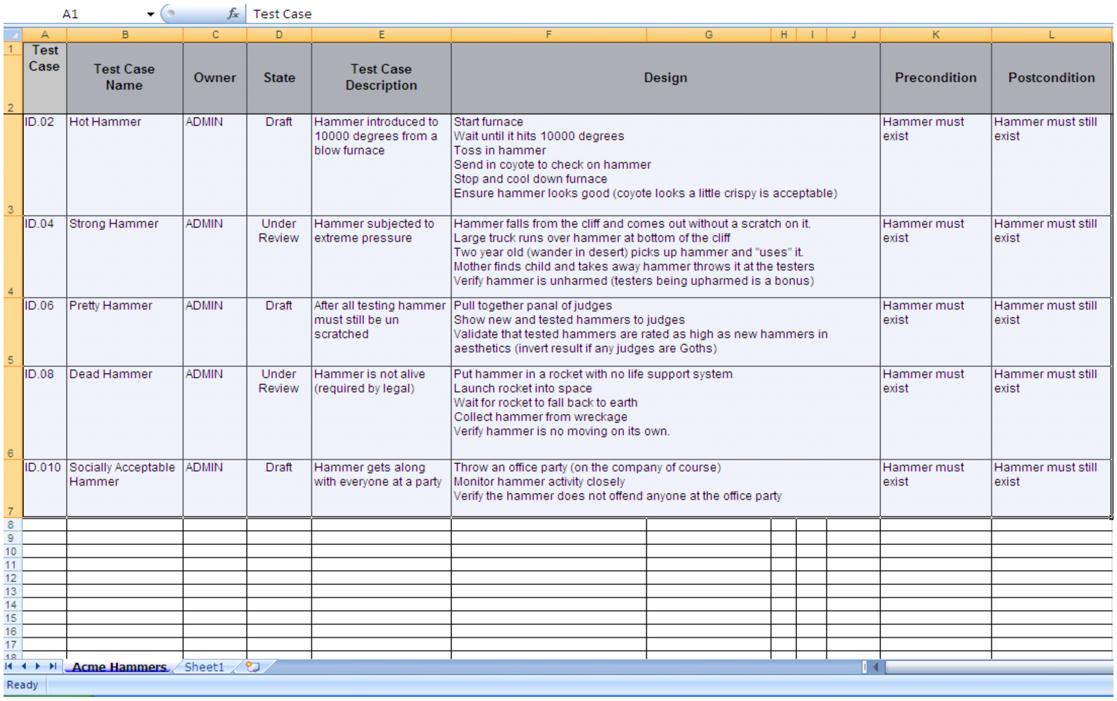
Example:

For example we will test an input field that accepts 20 characters

The test case will be documented as the following table while we run the test on our example. In our example the first test case will pass while the second one will fail.

Scenario	Test Step	Expected Result	Actual Outcome
Verify if the input field accepts 20 characters	first login to application and the key in 20 characters	Our application should accept 20 characters	Pass. The application accepted all 20 characters
Verify if the input field accepts 25 characters	first login to application and the key in 25 characters	Our application should not accept 25 characters	Fail. The application did not accept all 25 characters

If the actual result is not equal the expected result then log defect. The defect will be running in the defect life cycle and the tester have to fix it.



Test Case	Test Case Name	Owner	State	Test Case Description	Design	Precondition	Postcondition
ID.02	Hot Hammer	ADMIN	Draft	Hammer introduced to 10000 degrees from a blow furnace	Start furnace Wait until it hits 10000 degrees Toss in hammer Send in coyote to check on hammer Stop and cool down furnace Ensure hammer looks good (coyote looks a little crispy is acceptable)	Hammer must exist	Hammer must still exist
ID.04	Strong Hammer	ADMIN	Under Review	Hammer subjected to extreme pressure	Hammer falls from the cliff and comes out without a scratch on it. Large truck runs over hammer at bottom of the cliff Two year old (wander in desert) picks up hammer and "uses" it. Mother finds child and takes away hammer throws it at the testers Verify hammer is unharmed (testers being upharmed is a bonus)	Hammer must exist	Hammer must still exist
ID.06	Pretty Hammer	ADMIN	Draft	After all testing hammer must still be un scratched	Pull together panal of judges Show new and tested hammers to judges Validate that tested hammers are rated as high as new hammers in aesthetics (invert result if any judges are Goths)	Hammer must exist	Hammer must still exist
ID.08	Dead Hammer	ADMIN	Under Review	Hammer is not alive (required by legal)	Put hammer in a rocket with no life support system Launch rocket into space Wait for rocket to fall back to earth Collect hammer from wreckage Verify hammer is no moving on its own.	Hammer must exist	Hammer must still exist
ID.010	Socially Acceptable Hammer	ADMIN	Draft	Hammer gets along with everyone at a party	Throw an office party (on the company of course) Monitor hammer activity closely Verify the hammer does not offend anyone at the office party	Hammer must exist	Hammer must still exist

How to writing effective test case:

Writing an effective test case is a skill and need some experience to be written in a good way that it tests the application and software and gives good results.

- Write a simple test case from available documentation and specification.
- The test case should be written according to the application or software that is being tested.
- You have to write more than one test case and make a test procedure (which is many test cases).
- Automation of the project. In this level you have to minimize the human interaction with the system

The Goals of Testing:

To make sure that your test is effective and accurate you have set the goals and purposes:

First, you need to know what are you testing and what is the reason of this test and what the expected results.

Second, you need to set the conditions for the success and failure.

Third, you should have a good idea and good methodology for your test.

Fourth, the test cases must be designed according to the plan.

Fifth, Testing schedule must be there, and you have to run the test for a specific time according to the conditions.

if you want to have a really accurate and good test results you must have a good plan (good scenario, attribute, input, time and results) with good ideas before you start testing.

Strategies of the test:

Of course the testing will reduce the risk of the software and application, and the overall cost too. As long as you test your application and your software earlier as it gives you more opportunity to handle the risk and control it. Moreover, The number of bugs will be reduced and the quality with the functionality will be improved and work in a better manner.

There are two different approaches for the tester to describe his point of view about the test:

Black Box Testing:

In this approach the software will be treated as a "black box" with knowing about the internal implementation. This approach include: analysis of the boundary value, equivalence partitioning, fuzz testing, testing all the pairs, matrix traceability, specification-based testing and model-based testing.

White Box Testing:

In this approach the tester has an access to all the internal structures of the data and algorithms including the code.

Examples where does the white box testing exist:

1. Making the test to improve the criteria of the coverage.

2. Injection Fault methods.

3. Methods of mutation methods.

4. Statistical testing.

5. Sometimes it is being used to make sure of the quality for the black box testing.

6. Testing a specific application by using public and private APIs.

Testing Levels:

As we know that software testing is very important in each software life cycle. During the test there are several stages of development and this is a vital step for developing the life cycle. Any test can be run on the following levels:

Environment of object-oriented, under the class level and indicates for the minimum constructors and destructors.

Integration Testing: this level is about exposing the defects in the interface of the application or the software and the interaction between several components.

System Testing: This level will test the whole system to reach the expected requirements.

System Integration Testing: This level will verify if the system is integrated with any other systems or external third party systems.

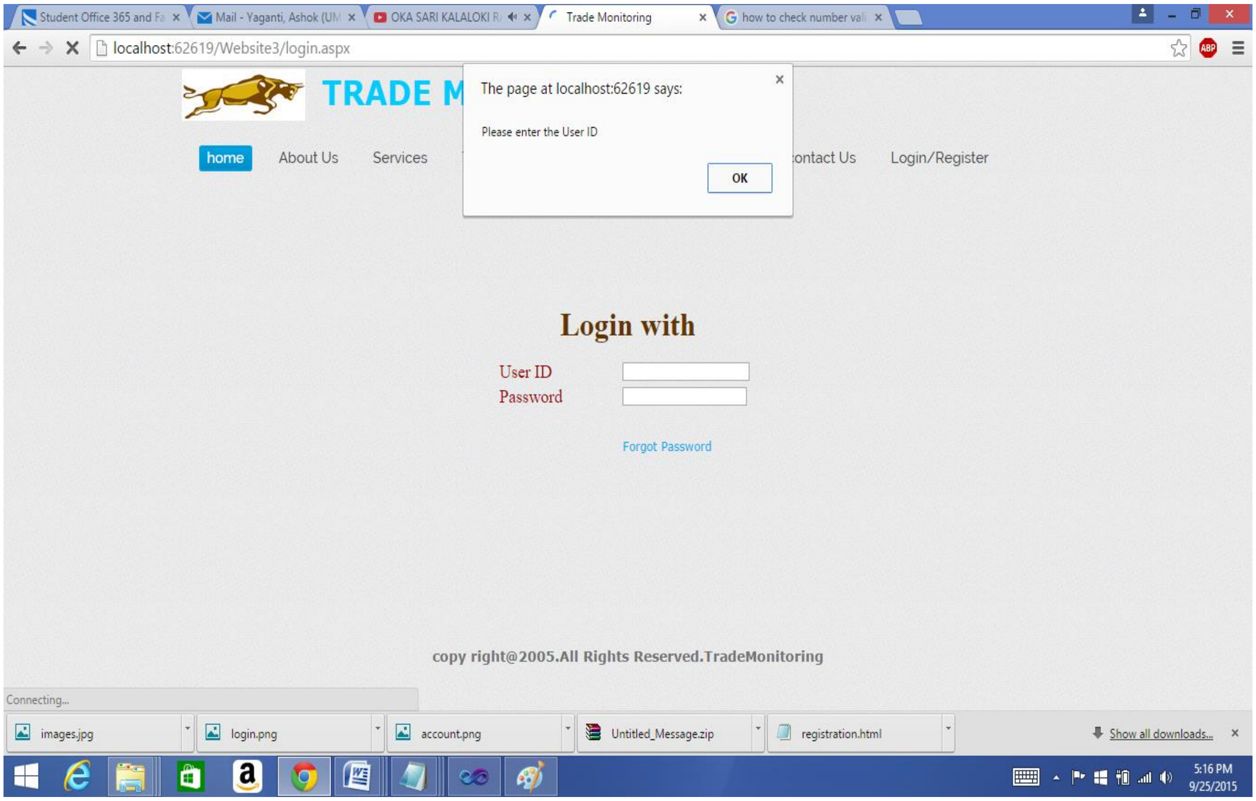
Also another two tests are important before release the software:

Alpha Test: In this test the software or the application is being used by several customers and users or another test team. Alpha test is important because it is for off-the-shelf and we required alpha testing before the software goes to Beta testing.

Beta Testing: This test is required after Alpha testing. in this test versions of the software is being released for limited audience from outside the team. In this case the software will be released to limited audience so that they could investigate about any further bugs or failures.

Testing Results

1. Null Value Check



The page at localhost:62619 says:

Please enter the User ID

OK

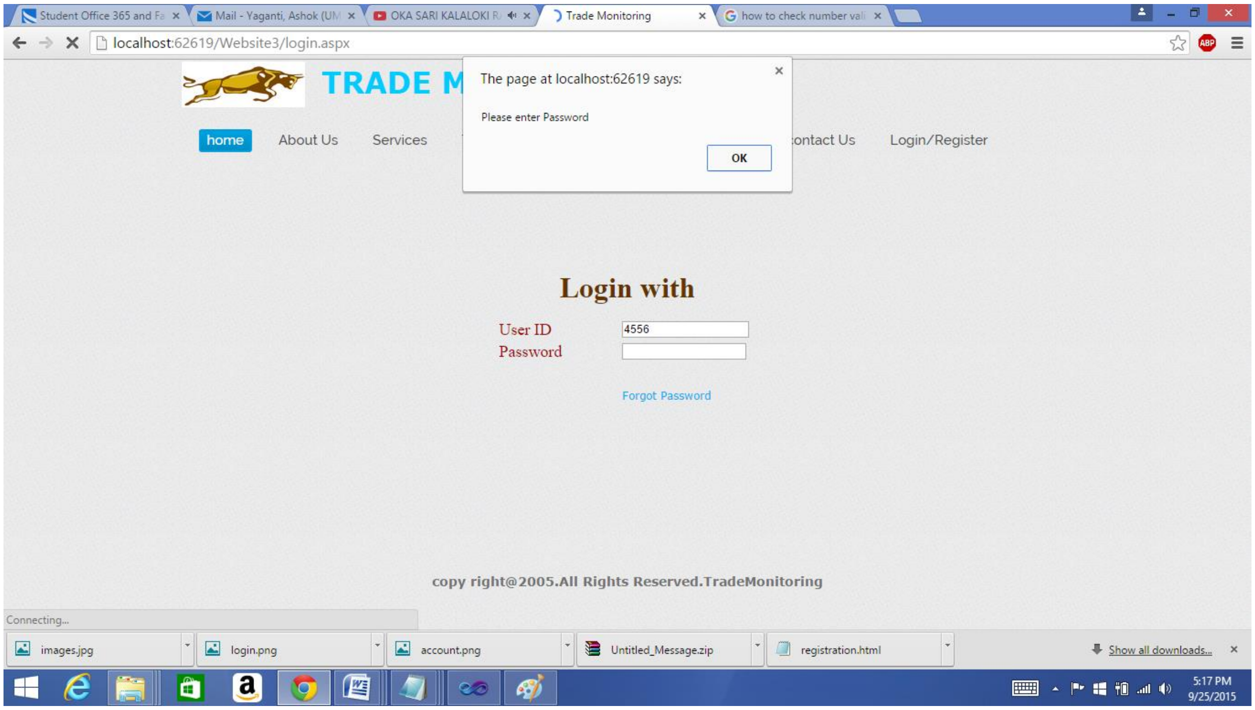
Login with

User ID

Password

Forgot Password

copy right@2005.All Rights Reserved.TradeMonitoring



The page at localhost:62619 says:

Please enter Password

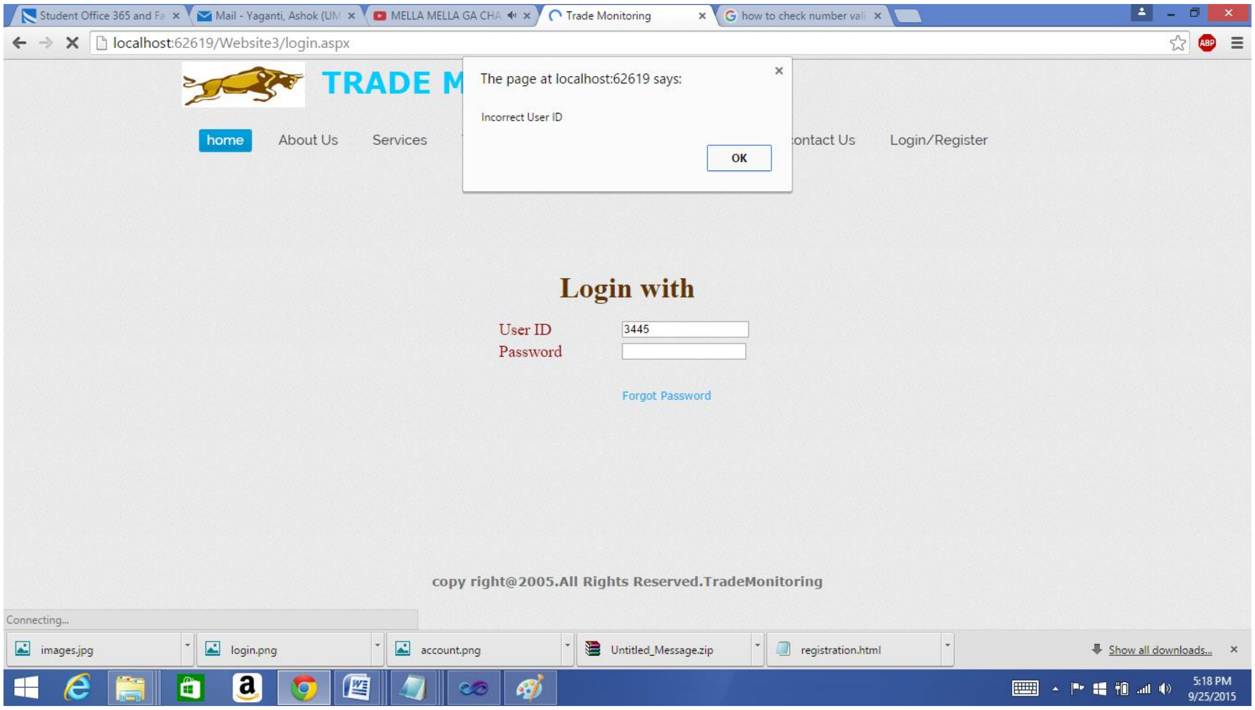
Login with

User ID 4556

Password

copy right@2005.All Rights Reserved.TradeMonitoring

2. Validation for Authentication.



The page at localhost:62619 says:

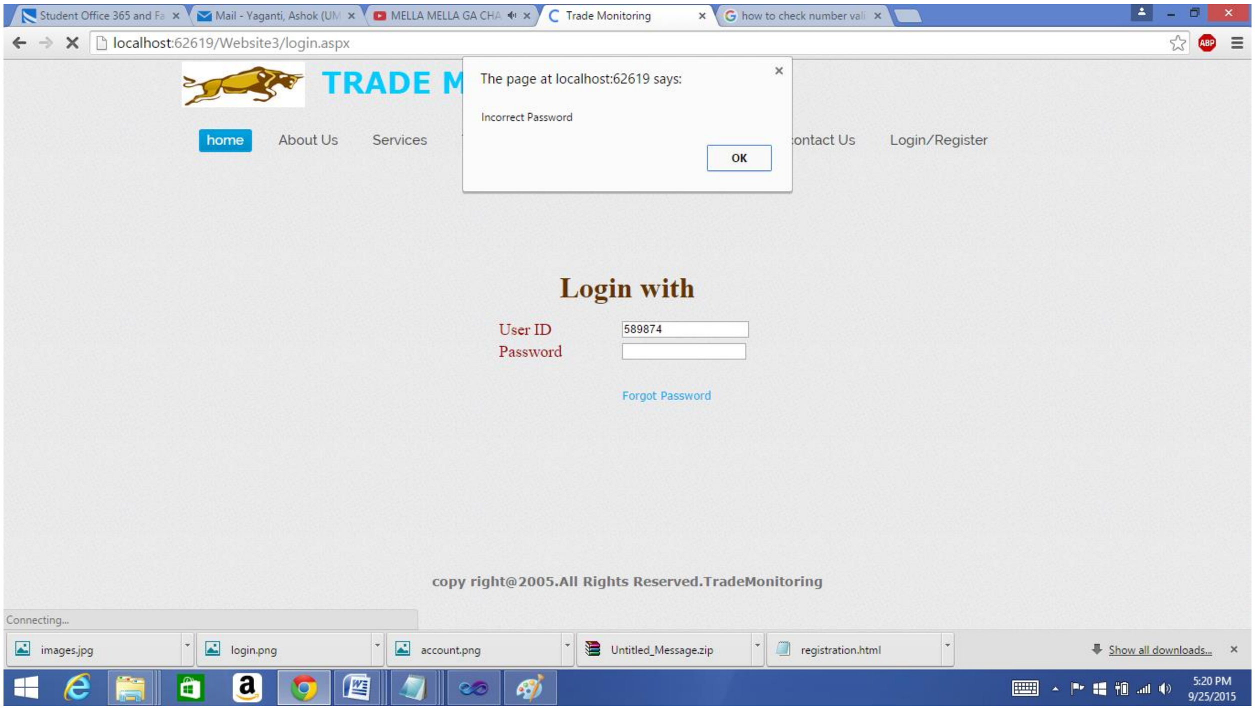
Incorrect User ID

Login with

User ID 3445

Password

copy right@2005.All Rights Reserved.TradeMonitoring



The page at localhost:62619 says:

Incorrect Password

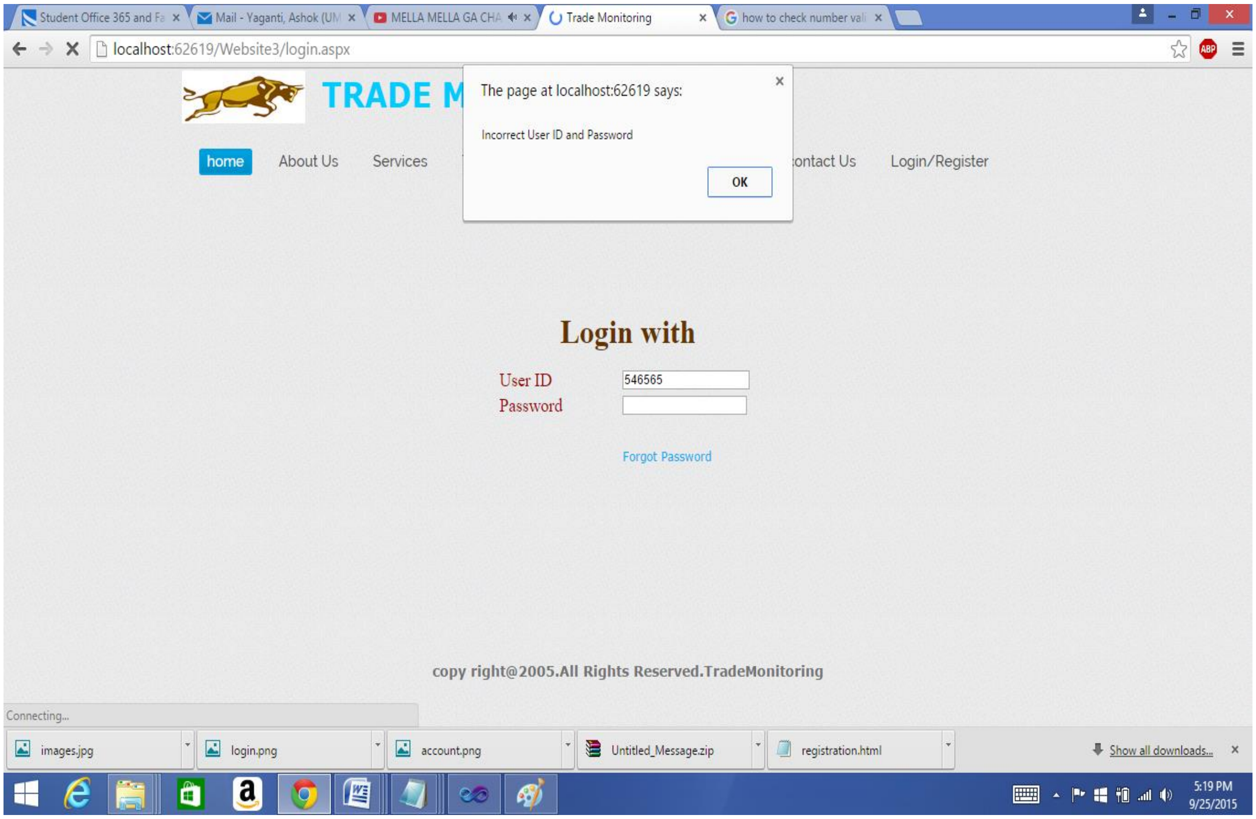
Login with

User ID 589874

Password

Forgot Password

copy right@2005.All Rights Reserved.TradeMonitoring



The page at localhost:62619 says:

Incorrect User ID and Password

Login with

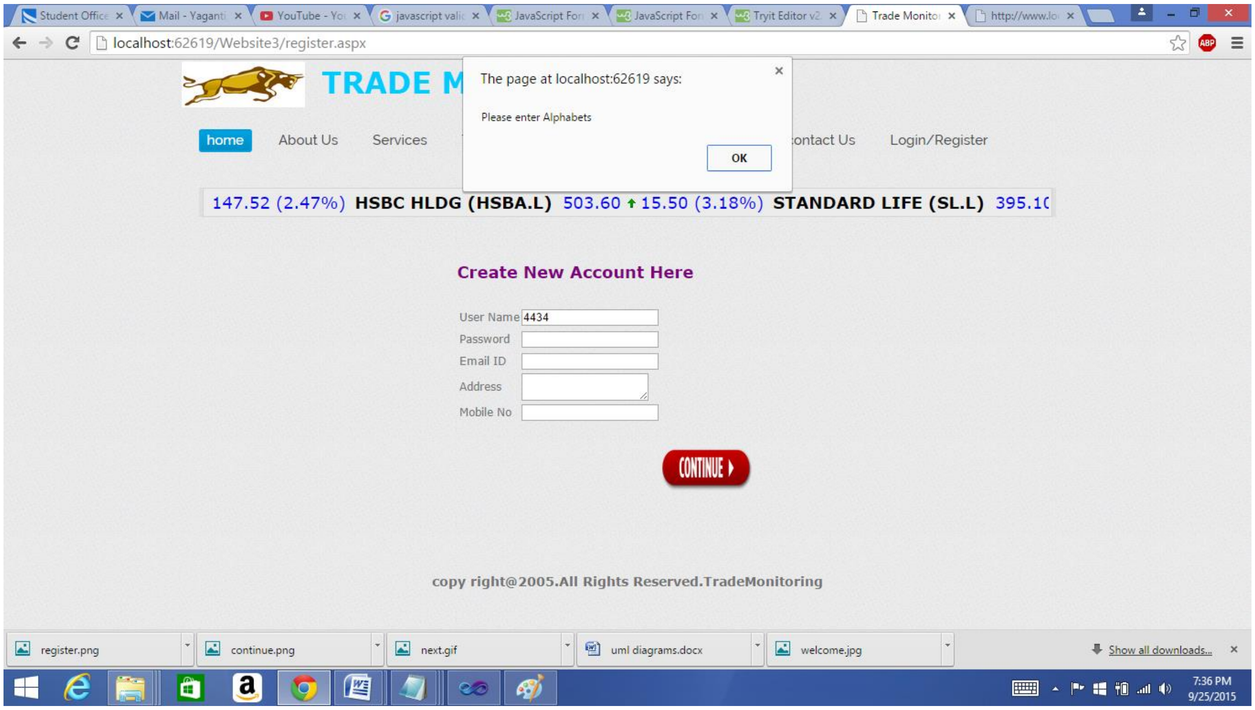
User ID 546565

Password

Forgot Password

copy right@2005.All Rights Reserved.TradeMonitoring

3. Alphabet Validation.



localhost:62619/Website3/register.aspx

The page at localhost:62619 says:

Please enter Alphabets

OK

147.52 (2.47%) HSBC HLDG (HSBA.L) 503.60 ↑15.50 (3.18%) STANDARD LIFE (SL.L) 395.1(

Create New Account Here

User Name 4434

Password

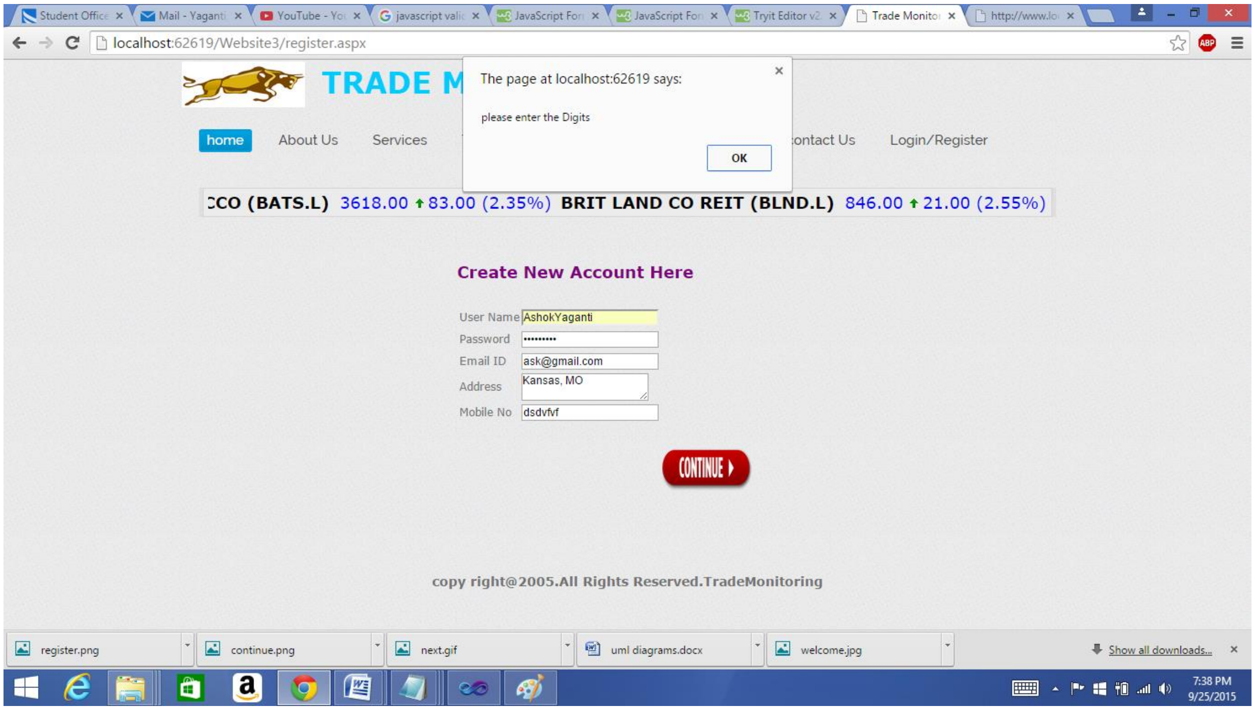
Email ID

Address

Mobile No

copy right@2005.All Rights Reserved.TradeMonitoring

4. Number Validation.



localhost:62619/Website3/register.aspx

The page at localhost:62619 says:

please enter the Digits

OK

:CO (BATS.L) 3618.00 ↑83.00 (2.35%) BRIT LAND CO REIT (BLND.L) 846.00 ↑21.00 (2.55%)

Create New Account Here

User Name AshokYaganti

Password ••••••••

Email ID ask@gmail.com

Address Kansas, MO

Mobile No dsdvfvf

copy right@2005.All Rights Reserved.TradeMonitoring

5. Deployment Summary

We have deployed our project in IIS server and Github. Please find the URL for the deployed project.

GITHUB URL :https://github.com/AshokYaganti/ASE-Project/tree/master/ASE_Project

6. Iteration Overview

We are following the Agile process model. So every Iteration we will receive the additional requirements and changes will be made from all the phases starting from requirements phase to deployment phase. So for the 1st Iteration we have successfully completed during the period of 10th SEP 2015 to 25th SEP 2015.

In this Iteration we have completed the 20% of the project including all the phases in SDLC. During this phase we have designed all the UML diagrams, few data tables and user interface diagrams. We have done all the validations, unit test cases and functionality.

We four have shared the work equally and involved in all phases of SDLC.

7. References

- “http://www.tutorialspoint.com//asp.net/index.html”
- “http://www.tutorialspoint.com//wcf/index.html”
- “https://www.iis.net/”
- “https://msdn.microsoft.com/en-us/library/ms733766(v=vs.110).aspx”
- “https://www.youtube.com/watch?v=PBAs3zw6rq0”
- “https://jquery.com/”
- “http://api.jquery.com/jquery.ajax/”