PROJECT REPORT ON

COMPONENT SHARING IN NETXINVESTOR

Submitted in partial fulfillment of the requirements for
The award of the degree of

**BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY
OF
SASTRA UNIVERSITY**

Submitted by

V RADHA LAKSHMI – 117015106

**Under the Guidance of
MAREESWARAN PAULRAJ MUTHURAJ
INAUTIX TECHNOLOGIES Pvt. Ltd, CHENNAI**

SCHOOL OF COMPUTING

**SHANMUGHA
ARTS, SCIENCE, TECHNOLOGY & RESEARCH ACADEMY
(SASTRA UNIVERSITY)**
(A University Established under section 3 of the UGC Act, 1956)
**TIRUMALAISAMUDRAM
THANJAVUR – 613 401
April 2017**

I

(SASTRA UNIVERSITY)
(A University Established under section 3 of the UGC Act, 1956)
TIRUMALAISAMUDRAM
THANJAVUR – 613401

SCHOOL OF COMPUTING

<u>BONAFIDE CERTIFICATE</u>

This is to certify that the Project entitled
COMPONENT SHARING IN NETXINVESTOR

is a work done by

V RADHA LAKSHMI **117015106**

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY
OF SASTRA UNIVERSITY, Thanjavur during the year 2016-17

External Guide   **Associate Dean**
                                                          Department of Information Technology

Submitted for the university examination held on:

Internal Examiner   **External Examiner**

| |

SCHOOL OF COMPUTING

SHANMUGHA
ARTS, SCIENCE, TECHNOLOGY & RESEARCH
ACADEMY
(SASTRA UNIVERSITY)
(A University Established under section 3 of the UGC Act, 1956)
TIRUMALAISAMUDRAM, THANJAVUR – 613401

<u>DECLARATION</u>

I submit this project work entitled "**COMPONENT SHARING IN NETXINVESTOR**" to the Shanmugha Arts, Science, Technology & Research Academy (SASTRA) University, Tirumalaisamudram–613 401, in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY** and declare that it is my original and independent work carried out under the guidance of **MAREESWARAN PAULRAJ MUTHURAJ**, iNautix Technologies, Chennai.

Date :   **Name :**V RADHA LAKSHMI   **Signature:**

Place :   **Reg. No:**117015106

ACKNOWLEDGEMENT

I would forever remain grateful to honourable **Prof. R. Sethuraman, Vice Chancellor** for his encouragement in my academic life at SASTRA University.

I wish to express my profound gratitude to **Dr. S. Vaidhyasubramaniam, Dean – Planning & Development, Dr. S. Swaminathan, Director CeNTAB** and **Prof. G. Bhalachandran, Registrar** for their overwhelming support provided during my course span in SASTRA University.

I am extremely thankful to **Dr. K.S. Ravichandran, Associate Dean, Department of Information Technology** for providing me the opportunity to do this project and for all the academic help extended in my project.

I sincerely express my gratitude to my manager **Mr. Padmaraj Ramachandran** and my team mentor **Mrs. Anuja Rani Sasidharan, Inautix TECHNOLOGIES INDIA PVT LTD** for their assistance and guidance for the successful implementation of project in a systematic and professional manner. I also express my gratitude to my mentor **Mr. Mareeswaran Paulraj Muthuraj, Inautix TECHNOLOGIES INDIA PVT LTD** for always guiding me in the right direction to complete my project.

I also thank all the **Teaching and Non-teaching staff,** and those who have directly or indirectly helped me by extending their moral support and encouragement for completion of this project.

I thank God Almighty and my parents for helping me procure such a challenging and interesting project, and in completing the same in due course without much difficulty.

TABLE OF CONTENTS

1.SYNOPSIS

There has been a tremendous development in the field of technology. Initially, the entire application was developed by a single party. With the advancement in technology, components are outsourced to different third parties. Large Organizations use these APIs to build their own applications. This way of outsourcing components to various different organizations is termed as Component Sharing. Component Sharing is a product-based strategy that depends on the fact that families of similar products have similar components.Companies increasingly view Component Sharing as a way to have high variety in market place and low variety in their operation. With Component Sharing, different organizations can use the APIs to build more powerful applications.

NetXInvestor is a platform that enables Component Sharing. It combines online investment solution for clients with exceptional flexibility and customization options for any firm. With NetXInvestor, one can keep investors connected with the firm, advisors and the markets. It offers anytime, anywhere access to consolidated account information, online documents, portfolio management solutions, and trading-plus market research, quotes and news from leading providers. With NetXInvestor, clients can get instant access to account information and view statements and trade confirmations, access quotes and news from leading sources-plus research on stocks, options and mutual funds.

NetXInvestor uses wide range of technologies both in client and server side. It provides end-to-end service for its clients. Server side technologies include Liferay, Grails and Groovy, Spring MVC and JSF. It uses several frameworks in the client side like Dojo for Javascript and Bootstrap for CSS.

2. SYSTEM CONFIGURATION

<u>Hardware Requirements:</u>

The hardware requirements of the current project are

- Minimum of 4GB of RAM
- Minimum memory 512MB
- Minimum disk free space 250MB
- Recommended disk free space 500MB
- 233 MHz Intel compatible CPU

<u>Software Requirements:</u>

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

> PLATFORM   : Windows 8+, Mac OS X(10.5x and 10.6x).

> TOOLS USED   : Eclipse, Java 1.7,Dojo 1.10.4.5,

                            Liferay Tomcat 6.2 ga4

3. INTRODUCTION

3.1 INTRODUCTION TO THE COMPANY – iNAUTIX

3.1.1 INTRODUCTION

iNautix Technologies India Private Limited is a group company of Bank of New York Mellon - a leading financial services provider. iNautix provide technology development, business & technology operations and remote infrastructure management services for BNY Mellon and its subsidiaries. iNautix also develops and delivers comprehensive technology solutions and software development products for customers of BNY Mellon.

3.1.2 FOUNDERS

The Bank of New York Mellon was founded by Alexander Hamilton in 1784. Gerald Hassell is an American bank executive and is the Chairman and CEO of the Bank of New York Mellon from 2011. Suresh Kumar is the CEO of iNautix Technologies.

3.1.3 ABOUT COMPANY

iNautix Technologies, established in the year 2000,provides technology solutions to BNY Mellon. As an innovation-focused company, iNautix specializes in providing insightful and new age technology solutions to help meet clients' business needs end-to-end. Since its inception in 2000, iNautix has maintained a relentless commitment to develop products and solutions that are designed to improve operational efficiencies, make planning more dynamic and realistic, monitoring and evaluation simple, risk-free and methodical, increase productivity, enhance delivery performance and ensure success for our clients.

3.2 INTRODUCTION TO THE PROJECT

3.2.1COMPONENT SHARING

Component Sharing is a technique that allows client to take components from NXI and embed them directly in their own website.

AIM

To implement Component Sharing in NetXInvestor in order to reduce human resource and enhance re-usability.

OVERVIEW

Windows-based applications can share the following:

- Code
- Application
- Component state in the registry
- Application specific data in the file system
- Windows APIs which expose global namespaces.

Sharing allows for the efficient leverage of limited hardware resources and reduces the exposed front that Quality Assurance groups must test.

However, there are costs to sharing. Sharing makes applications interdependent upon one other, which introduces an element of fragility. In the extreme, applications that once worked will mysteriously start functioning oddly, or even fail. Typically, an application is dependent on a particular version or implementation of a shared component. If that shared component is upgraded (or downgraded) as a result of installing another application, the former application may break. This condition is often referred to as "**DLL Hell**"

4. LITERATURE SURVEY

EXISTING SYSTEM

Component sharing is the process through which technological components and platforms are transferred and shared between projects. The vast majority of evidence about component sharing is based on empirical research in mass-producing companies. Component sharing is useful as it reduces human effort and increases re-usability. Component sharing is done using iframes.

An iframe is used to display a web page within a web page. It lets users view content from another user on the same page, and can sometimes be accompanied with its own scrollbar, and navigation.

An HTML iframe is defined with the **<iframe>** tag. Iframe stands for inline frame. The syntax for <iframe> tag is:

<iframe src="*URL*"></iframe>

The **src** attribute is used to specify the URL of the document that occupies the inline frame.

DRAWBACKS OF EXISTING SYSTEM

Usage of iframes is easier but it has the following disadvantages.

1. Style applied to iframe may not comply with the existing page's styling.

2. Iframes don't scale well .When an iframe's content needs to be expanded it is forced to stay within the container, hence a scrollbar will appear. This is not good user experience.

3. Unreliability of different users. In an iframe, because the HTML can be changed without the discretion of the website's creator, malicious content can be uploaded without permission of the website's designer. A website may have some content on it, with another website's content, but the content can change anytime, to material that one may or may not approve of.

4. Some browsers don't support iframes. The most important aspect of the internet is that users can access almost every website, no matter their region or computer system. The open nature of the internet is what draws users into checking out website. Because some browsers don't support iframes, using iframes stops some websites' accessibility to users that have browsers that don't support the application. When the internet is based around the amount of hits your website gets, this could create a major problem in gaining popularity amongst users.

5. Using iframe won't help the caller domain to control, manipulate and for applying styles as needed because of cross domain restrictions.

6. Javascripts and other theme support available at the parent domain cannot be utilized by the domain running within the iframe.

PROPOSED SYSTEM

System Design

Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. It is the most crucial phase in the developments of a system. The logical system design arrived at as a result of systems analysis is converted into physical system design. Normally the design proceeds in two stages: Preliminary and Detailed design.

Preliminary or General Design

In the preliminary or general design, the features of the new system are specified. The costs of implementing these features and the benefits to be derived are estimated. If the project is still considered to be feasible, we move to the detailed design stage.

Structured Or Detailed Design

In the detailed designed stage, computer oriented work begins in earnest. At this stage, the design of the system becomes more structured. Structured design is a blue print of a computer system solution to a given problem having the same components and inter relationships among the same components as the original problem. Input, output, databases, forms, codification schemes and processing specifications are drawn up in detail.

In the design stage, the programming language and the hardware and software platform in which the new system will run are also decided. There are several tools and techniques used for describing the system design of the system. These tools and techniques are:

➢ Flowchart
➢ Data Flow Diagram (DFD)
➢ Data Dictionary

Overcoming the drawbacks of iframe-A new Framework

To overcome the above mentioned drawbacks of HTML iframes, we go for a new framework named **"OpenF2".**

F2 is an open and free web integration framework designed to help you and other financial industry participants develop custom solutions that combine the best tools and content from multiple providers into one, privately-labeled, seamlessly integrated front-end. F2 is maintained by Markit On Demand.

5. PROJECT DESCRIPTION

5.1.ABSTRACT

With the growing technology, it is tedious for an organization to develop an entire application. So large organizations started to use the already developed components by some other party. This concept is termed as "Component Sharing" .In this project, we use the same concept of component sharing in our product named NetXInvestor. NetXInvestor is an online website which facilitates trading. Component Sharing in NetXInvestor is done using the OpenF2 framework.

5.2What is Component Sharing?

Component Sharing is a new technique that allows clients to take components from NXI and embed them directly into their own site. Clients can mix and match into their site as they want or need.

Windows 2000 and Windows 98 Second Edition enable a new form of sharing called **side-by-side sharing** which minimizes this application fragility. Side-by-side sharing enables multiple versions of the same COM or Win32 component to run at the same time in memory. This means **applications can use the specific components that they were designed for and tested with**, even if another application requires a different version of the same component. This allows developers to build more reliable applications because developers can choose the version of the component they will use for their application, independent of the other applications on the system.

5.2.1Advantages:

- Not locked to our page layouts

- Free to combine components from different NXI pages into one page

- Free to order / place components wherever they want

- Can override styling with their own

5.3 What is Component ?

Before going into Component Sharing, definition for component needs to be understood. Many misinterpret component to be

- An entire page in NXI.

- The full contents of the main container DIV.

- Navbar widgets like the Messages or Settings icons.

But these are not components. A component is any existing portlet in the NXI site namely:

- Portfolio in the News

- Balances Details

- UGL or RGL Grid

- Account Picker

In some cases, a group of portlets that must be kept together to maintain the overall functionality is also termed as Component.

- Trading

- Asset Transfer.

5.4. Using OpenF2 for Component Sharing

OpenF2 is open source and free web integration framework that allows to tightly integrate the front end components. F2 allows firms to choose the data and content they want from any single provider or multiple providers, and integrate it seamlessly into their solution. With F2, Component Sharing becomes simpler and easier.

5.4.1.Features of F2:

- Streaming video sourced from a third party won't slow down page load times and frees the page owner from hosting costs. The provider can create one App and deploy it many times, either as a singular view or as part of an aggregated view.

- A heat map benefits greatly from F2's responsive design specification. Regardless of the provider, navigation and size adapt to the device.

5.4.2. Problems solved by using F2

Many problems have been solved with F2.Some of them are listed below.

- Standardization: F2 provides a standard mechanism for integration across organizations, teams, and third parties, allowing companies to work together smoothly.

- Efficiency: Using F2, developers can build once and deploy to any F2-enabled platform on any device, eliminating the need for duplicate work.

- Agility: Unlike with a closed monolithic system, F2 allows platform owners to progressively make enhancements and migrate users and tools gradually as time and resources allow.

5.4.3Benefits of F2

OpenF2 framework has been a huge success in the market due to the following benefits.

- Reduced time to market: Using F2, developers can work off of a shared standard specification to quickly build Apps. In addition, firms can immediately deploy their products to one or many F2-enabled platforms, meaning a shorter time to market.

- Quick iterations: F2 allows for multiple quick iterations based on user feedback or changing business needs. Using F2, firms can compete on features, make enhancements, or switch content providers with little risk.

- Security: In an F2-enabled platform, Apps can communicate and pass information to each other without revealing sensitive user information to a third party provider.

5.4.4F2 Container

The container is the foundation of any F2-enabled solution. By leveraging the F2.js SDK, Container Providers offer a consistent and reliable mechanism for all App Developers to load their apps on that container regardless of where it is hosted, who developed it, or what back-end stack it uses. Setup a basic container HTML template :

```html
<!DOCTYPE html>
    <head>
        <title>F2 Container</title>
        <link rel="stylesheet" href="/path/to/your/bootstrap.css">
    </head>
    <body>
        <script src="/path/to/your/F2.js"></script>
    </body>
</html>
```

In developing a more advanced container, the HTML document's body element would contain additional markup and allow for specific positioning or placement of apps. Additionally, more advanced containers could introduce features and functionality to their apps in the form of authentication APIs, streaming data feeds, federated search, etc.

```
var _appConfig = {
    appId: 'com_openf2_examples_javascript_helloworld',
    manifestUrl: 'http://docs.openf2.org/demos/apps/JavaScript/HelloWorld/manifest.js'
};


$(function(){
    F2.init();
    F2.registerApps(_appConfig);
});
```

5.4.4.1Developing F2 Containers

A container is a browser-based web application which brings F2 apps together onto a seamless user interface. It can also provide data and user context to its apps in the form of request-response web services or streaming data feeds.

5.4.4.1.1.Including the F2 SDK

For a webpage to be considered an F2 container, it must first include the F2.js JavaScript SDK. This is as simple as downloading the F2 project from GitHub and adding a script tag to the page.

```
<script src="/path/to/your/f2.js"></script>
```

You will find a basic container in the project repo on GitHub along with a number of sample apps. Once the script tag has been added, it is up to the Container Developer to configure and customize the container. The first step is getting a ContainerID.

5.4.4.1.2.F2 ContainerID

To develop a production F2 container, you need a unique identifier called a ContainerID. This ContainerID will be unique to *your container* across the entire open financial framework ecosystem. The format of the ContainerID looks like this:

com_container_companyName_containerName

where the companyName "namespace" is your company name and containerName is the name of your container.

As an example, your ContainerID could look like this:

com_container_acmecorp_watchlist

If you built more than one container while working at Acme Corporation, you could create more ContainerIDs. All of these are valid:

- com_container_acmecorp_activetrader
- com_container_acmecorp_retail
- com_container_acmecorp_mobilestreamer

To guarantee uniqueness, we will provide a ContainerID generation service that allows customization of your ContainerID in the Developer Center.

5.4.4.1.3.Setting Up Your Project

Once you have your ContainerID, start by setting up your container project. You will need at least one configuration in addition to an HTML page: the app configs. (In the GitHub repository, an example is found in /examples/container/js/sampleApps.js.) This doesn't need to be a static javascript file like sampleApps.js but the structure and format of the app configs is important.

5.4.4.1.4.Container Config

The F2.js JavaScript SDK provides an API for providers to configure their containers. Every container must be setup using ContainerConfig and the methods available, however if the F2 defaults are acceptable, the ContainerConfig is not required.

To initialize a container *using F2 defaults*, call this function:

```
F2.init();
```

To initialize a container with a ContainerConfig, use:

```
F2.init({
    UI: {},
    xhr: function(){},
    supportedViews: []
});
```

5.4.4.1.5.Debug Mode

To enable debug mode in a container, use the following property in F2.init(). Setting debugMode: true adds additional logging, resource cache busting, etc.

```
F2.init({
   debugMode: true
});
```

For obvious reasons, this property should only be used in a development environment.

5.4.5.F2 Apps

Apps are the building blocks of any F2-enabled solution. F2 apps are web-based, built with HTML5, CSS3, and JavaScript, and contain entitled data. F2 enables App Developers to build once and deploy to any container with a simple and modern API.

F2 apps are synonymous with modules, widgets and portlets. Think charts, portfolios, trade tickets, and screeners.

F2 apps only need to be programmed once, no matter where they will be used. To start, F2 Apps are either:

- **Display App**: A display app presents information to users in the form of a visible widget (using HTML, CSS, and JavaScript).

- **Data App**: A data app is a content feed available in industry-standard formats including JSON, JSONP, RSS or app developer-designed XML.

5.4.5.1Developing F2 Apps

Let's take a close look at how to build an F2 app. We'll explain how to get an F2 AppID, what the AppManifest is all about, what output format your app needs to support, how the contents of the AppContent.html property work, and the two hooks for adding form and function to your app: scripts and styles.

5.4.5.1.1F2 AppID

To develop an F2 app, you need a unique identifier called an **AppID**. This AppID will be unique to *your app* across the entire open financial framework ecosystem. The format of the AppID looks like this:

<p align="center">com_companyName_appName,</p>

where the companyName"namespace" is your company name and appName is the name of your app.As an example, your AppID could look like this:

<p align="center">com_acmecorp_watchlist.</p>

If you built more than one app while working at Acme Corporation, you could create more AppIDs. All of these are valid:

- com_acmecorp_watchlist2
- com_acmecorp_watchlist_big_and_tall
- com_acmecorp_static_charts
- com_acmecorp_interactive_charts

5.4.5.1.2.Setting Up Your Project

Once you have your AppID, start by setting up your project. You will need at least one file: the **App Manifest**.

- ➤ Create a new file called manifest.js.

- ➤ Also, chances are you'll want custom styling and functionality, so go ahead and create appclass.js (for your app logic)

- ➤ Create app.css for your CSS.

Your project folder should look like this:
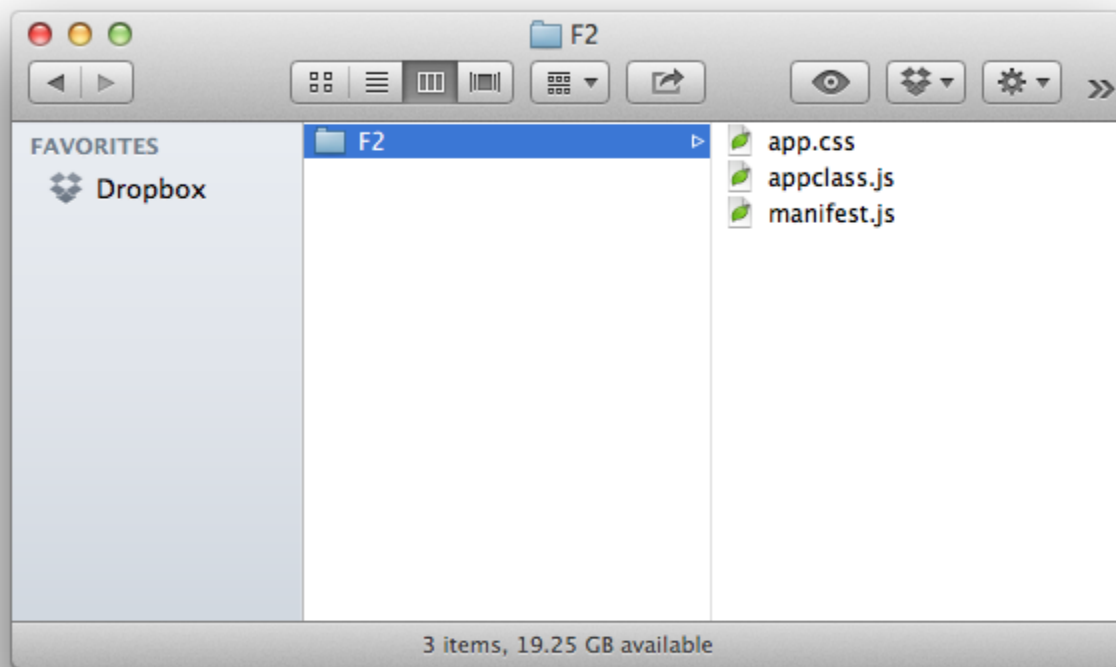
5.4.5.1.3.App Manifest

For an app to be considered F2-capable, it must first have this basic structure—called the **App Manifest**—represented in JSON.

The following is the basic structure of App Manifest.

```
{
    "inlineScripts":[],
    "scripts":[],
    "styles":[],
    "apps":[{
            "data":{},
            "html":"",
            "status":""
    }]
}
```

The App Manifest can be generated by the server-side code of your choice or be written-by-hand in your favorite text editor.

5.4.5.1.3.1.Apps

The apps property is an array of AppContent objects. Each AppContent object contains three properties:

1. html
2. data
3. status

HTML

The html property contains the view of your app represented in (optionally encoded) HTML. While you can modify the way your app appears or functions within the container, the html property is what the container will show when it registers your app and displays its contents for the first time.

Example:

```
"html": "<div class=\"sunrise\">Hello world.</div>"
```

Data

The data property exists to support the placement of arbitrary data needing to be passed along with the app. *This field is optional.*

Example:

```
"data": {
    foo: "bar",
    value: 12345
}
```

Status

The status property allows app developers to communicate a server-side arbitrary status code to itself or to the container. *This field is optional.*

Example:

```
"status": "good"
```

5.4.5.1.3.2.Styles

The styles property is an array of URLs. The styles array refers to any CSS files needed by the app so it will be displayed properly on the container. The externally-referenced CSS files should be fully-qualified, including a protocol.

Example:

```
"styles": [
    "http://www.domain.com/css/app.css"
]
```

In the case when multiple stylesheetes are needed, simply add to the array as shown in this example:

```
"styles": [
    "http://www.domain.com/css/app.css",
    "http://www.domain.com/css/app-responsive.css"
]
```

URLs referenced in the Scripts and Styles arrays are loaded synchronously by F2.js, so be sure to order your scripts properly.

5.4.5.1.3.3.Scripts

The scripts property is an array of URLs. The scripts array refers to any JavaScript files needed by the app so that it will function correctly on the container. The externally-referenced JS files should be fully-qualified.

Example:

```
"scripts": [
    "http://www.domain.com/js/appclass.js"
]
```

In the case when multiple scripts are needed, simply add to the array as shown in this example:

```
"scripts": [
    "http://www.domain.com/js/lib.js",
    "http://www.domain.com/js/appclass.js"
]
```

URLs referenced in the Scripts and Styles arrays are loaded *synchronously* by F2.js, so be sure to order your scripts properly.

5.4.5.1.3.4.Inline Scripts

The inlineScripts property is an array of strings. The inlineScripts array can include any JavaScript code needed by the app that *cannot be included in your App Class*. The contents of the inlineScripts array will be evaluated as JavaScript (using eval()) when all scripts have finished loading.

Example:

```
"inlineScripts": [
    "(function(){ var foo = bar; doSomething(); })()"
]
```

While the use of inlineScripts is supported by F2's App Manifest, it is not recommended for use. There are many reasons for this, the main one is to avoid cluttering the global namespace. Developers should make every attempt to put their JavaScript code inside their App Class.

URLs referenced in the Scripts and Styles arrays are loaded *synchronously* by F2.js, so be sure to order your scripts properly.

5.4.5.1.4.Sample App Manifest

If we use the examples above, our AppManifest would look like this:

```
F2_jsonpCallback_com_companyname_appname({
    "inlineScripts":["(function(){ var foo = bar; doSomething(); })()"],
    "scripts":[
        "http://www.domain.com/js/appclass.js"
    ],
    "styles":[
        "http://www.domain.com/css/app.css"
    ],
    "apps":[{
            "data":{
                foo: "bar",
                value: 12345
            },
            "html":"<div class=\"sunrise\">Hello world.</div>",
            "status":"good"
    }]
})
```

You may have noticed the presence of the function name on the first line of the example above. That function name F2_jsonpCallback_com_companyname_appname is the callback.

5.5.How Are Components Enabled for Clients?

- Very small set of steps for most components:
    1. Identify the component's "App ID"
    2. Create a sharing-specific LESS file
    3. Publish the app config for the component.

- In rare cases, we need to define a "grouped component"

    1.Create a grouping template that combines the individual components as needed

    2.Define an app ID for the group

    3.Add the ID to the list of group templates in RenderManifestServletAction

    4.Create the sharing-specific LESS files for the sub-components as needed

    5.Publish the app config for the component.

5.5.How does it work?

- Based off of MOD's F2 components used in NXI
- Clients are given a "catalog" of components they can use
- Clients add a single JS file from NXI to their web page
- Clients create an "app config" JS object for the page
    - Specifies which components from the catalog are being used
- Clients add <div> tags to the page for the components
    - Tags are scanned at page load and matched against the "app config"
    - For each matched tag, we make a REST call to NXI
    - Call returns the actual component HTML code, which is injected in place of the <div> elements
    - Also returns CSS and any additional JS needed, which is loaded dynamically

5.6.Dojo Toolkit

Dojo Toolkit is an open source modular Javascript library or more specifically Javscript toolkit designed to ease the rapid development of cross-platform, JavaScript/Ajax-based applications and web sites. It was started by Alex Russell, Dylan Schiemann, David Schontzler, and others in 2004 and is dual-licensed under the modified BSD license or the Academic Free License ($\geq$ 2.1) . The **Dojo Foundation** is a non-profit organization created with the goal to promote the adoption of the toolkit.

Dojo is a Javascript framework targeting the many needs of large-scale client-side web development. For example, Dojo abstracts the differences among diverse browsers to provide APIs that will work on all of them (it can even run on the server under Node.js); it establishes a framework for defining modules of code and managing their interdependencies; it provides build tools for optimizing JavaScript and CSS, generating documentation, and unit testing; it supports internalization, localization, and accessibility ; and it provides a rich suite of commonly needed utility classes and user-interface widgets.

Dojo is completely open-source. The entire toolkit can be downloaded as a ZIP and is also hosted on the Google CDN. The toolkit includes about three thousand JavaScript modules, in addition to images and other resources.

The Dojo Toolkit is organized in several parts:

- **dojo** contains the core and most non-visual modules.
- **dijit** is a library of user-interface modules for widgets and layout.
- **dojox** holds assorted modules not yet considered stable enough to include in *dojo* or *dijit*.
- **util** includes build tools such as optimization, documentation, style-checking, and testing.

5.7.Testing Components

Testing requires a Liferay NXI server with Apache (171 or other dev VM). Apache will have the Cross-Origin config from portal-partition applied. Before testing the component, must log in via SSO using

 http://testlab-ww.pllcfiles.inautix.com/pershing.netxinvestor.componentsharing/sso2_IDP.html.

SSO response page will provide a link to the component sharing general demo page (being built). From this page, can test individual components ad-hoc.

5.8. Liferay

Liferay Portal is a free and open source enterprise portal software product. Written in Java, Liferay Portal is a web platform with features commonly required for the development of websites and portals. Liferay includes a built-in web content management system allowing users to build websites and portals as an assembly of themes, pages, portlets/gadgets and a common navigation. Liferay is sometimes described as a content management framework or a web application framework. Liferay's support for plugins extends into multiple programming languages, including support for PHP and Ruby portlets.

Liferay's programming interface is accessible to users with no programming skills for the basic website installation and administration.

Liferay Portal is Java-based and runs on any computing platform capable of running the Java Runtime Environment and an application server. Liferay is available bundled with a servlet container such as Apache Tomcat.

5.8. 1.Liferay History

Liferay, Inc., is a professional open-source company that provides free documentation and paid professional service to users of its software. Mainly focused on enterprise portal technology, the company has its headquarters in Diamond Bar, California, United States.

Liferay Portal was created in 2000 by chief software architect Brian Chan to provide an enterprise portal solution for non-profit organizations. In 2004, the company was incorporated under the name Liferay, Inc. and formalized its Germany subsidiary Liferay GmbH. In 2007, the company opened new Asian headquarters in Dalian, China, and the Spanish subsidiary Liferay SL. In March 2009, the company opened a new office in Bangalore, India. To date there are 18 offices in 15 countries worldwide with a over 110 partners in 40 countries.

The company's enterprise portal product has been acknowledged by several notable organizations. It was recognized by EContent magazine in its "EContent 100" list of industry leadersand in 2007, Info World named it a "Technology of the Year". In July 2007, it announced a partnership with ICEsoft Technologies, provider of the ICEfaces library, for developing Ajax technology for its enterprise portal software. In January 2008, the company hired the lead engineer for jQuery UI, to exclusively work full-time on the JavaScript. Gartner recognized Liferay as visionary leader in the Magic Quadrant for Horizontal Portal Products in 2008, 2010, 2011, 2012 and 2013. Most recently in the Gartner 2014 report, Liferay out performed SAP, Oracle and Microsoft Sharepoint to become the leading open source product in its class. Liferay offers different levels of partnership across various geographies for its global SI partners which includes Platinum, Gold and Silver Liferay Service Partners.

Sun Microsystems and Liferay signed a technology-sharing agreement during May 2008. Sun Microsystems rebranded the offering GlassFish Web Space Server. ZDNet further describes the relationship in the May 2008 article *Sun and Liferay launch web-presentation platform*. In 2010 Sun was acquired by Oracle and the GlassFish Web Space Server was not included in their portal roadmap, with all prospects turned over to Liferay, Inc.

Liferay 6.1 was released in January 2012 and saw several improvements and new functionality including an improved document library, dynamic data lists and an app store.

In April 2013, Liferay partnered with TIBCO Software to offer a series of Liferay enterprise Connectivity Adapters (eC Adapters) that use TIBCO ActiveMatrix BusinessWorks with the intention of easing integration of Liferay Portal with multiple systems.

5.8. 2.Liferay Core Portlets

Liferay comes with certain portlets preinstalled. These comprise the core functionality of the portal system. They include:

- Alerts and Announcements
- Alfresco, Documentum, and other document library integration
- Asset Publishing
- Blogs and blog aggregation
- Breadcrumbs
- Calendar
- Chat
- Document and Image management
- Document Library Manager, Recent Documents
- Knowledge Base
- LDAP Integration
- Mail
- Message Boards
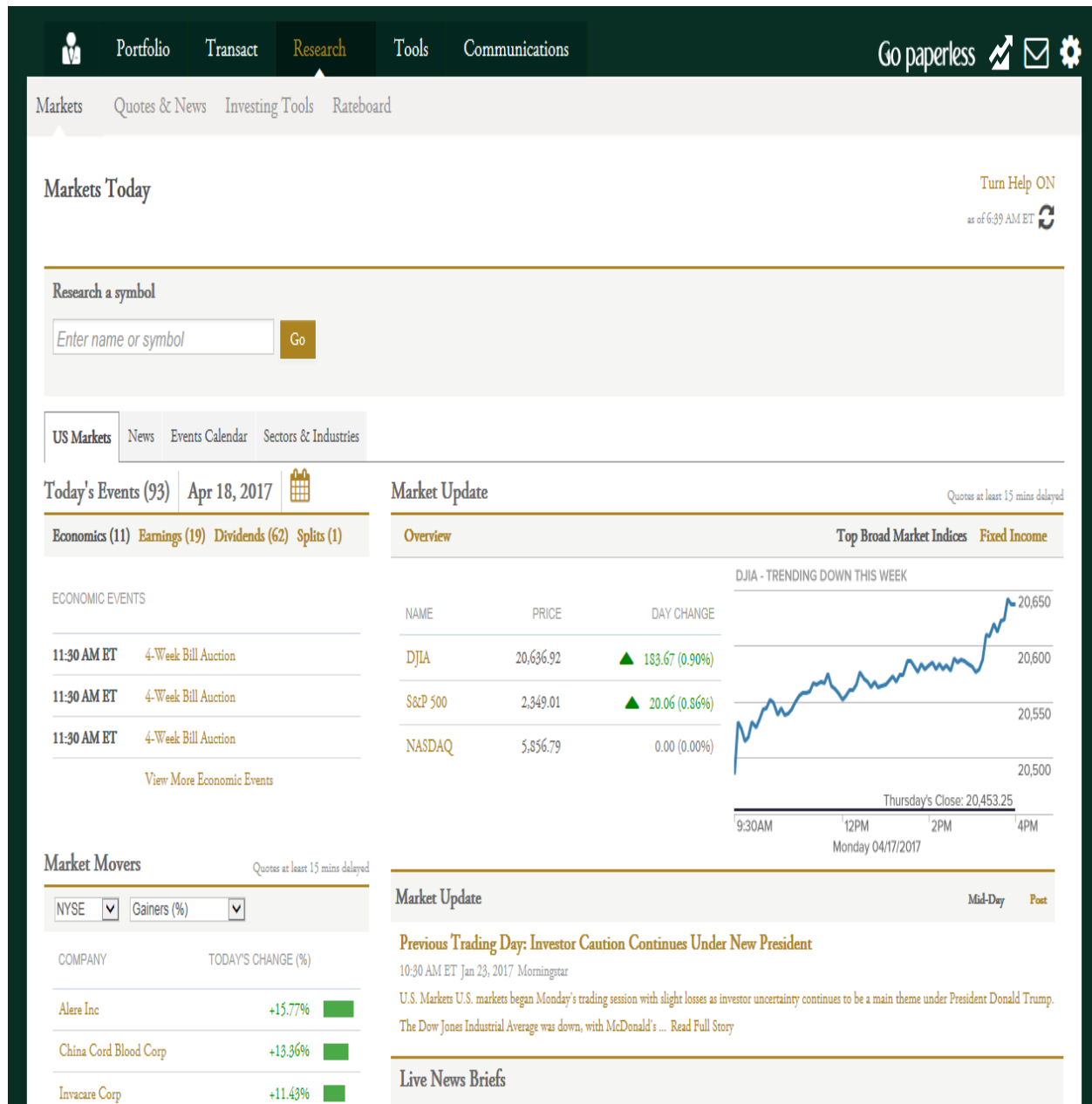- Nested Portlets
- Page Ratings & Flags
- Polls
- Site Map

6.DIAGRAMS

6.1.Usecase Diagram

6.2.Page Rendering Robustness Diagram
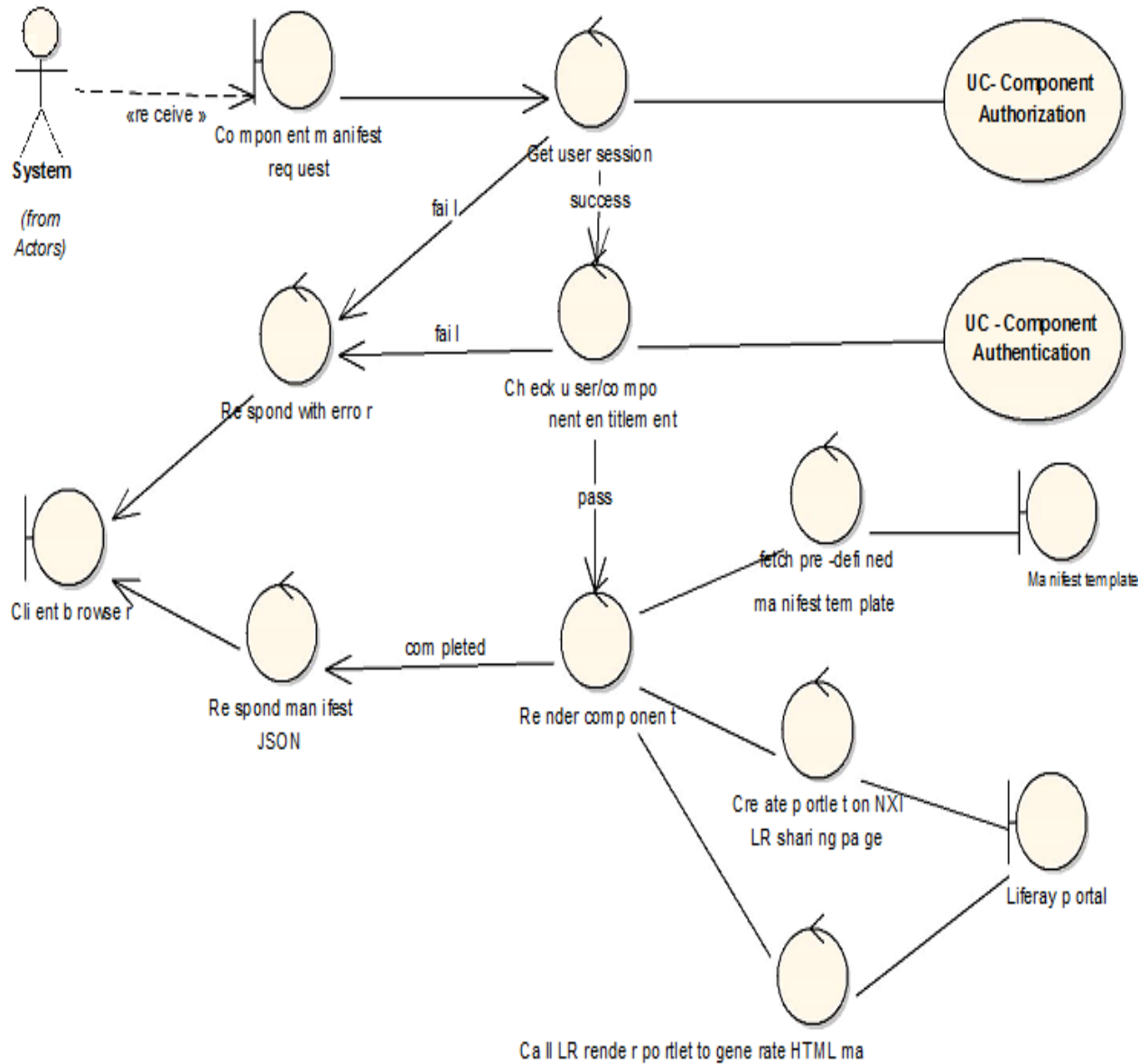
7. SAMPLE OUTPUT (SNAPSHOTS)

In the below mentioned diagram, the research tab is completely built by using the concept of Component Sharing. It is developed using F2 Framework.

Main Component

Portfolio Transact Research Tools Communications Go paperless

Markets Quotes & News Investing Tools Rateboard

Markets Today

Turn Help ON

as of 6:39 AM ET

Research a symbol

Enter name or symbol [Go]

US Markets News Events Calendar Sectors & Industries

Today's Events (93) | Apr 18, 2017 |

Economics (11) Earnings (19) Dividends (62) Splits (1)

ECONOMIC EVENTS

11:30 AM ET	4-Week Bill Auction
11:30 AM ET	4-Week Bill Auction
11:30 AM ET	4-Week Bill Auction

View More Economic Events

Market Update

| Overview | | | Top Broad Market Indices | Fixed Income |

DJIA - TRENDING DOWN THIS WEEK

NAME	PRICE	DAY CHANGE
DJIA	20,636.92	▲ 183.67 (0.90%)
S&P 500	2,349.01	▲ 20.06 (0.86%)
NASDAQ	5,856.79	0.00 (0.00%)

20,650
20,600
20,550
20,500

Thursday's Close: 20,453.25

9:30AM   12PM   2PM   4PM

Monday 04/17/2017

Market Update

Mid-Day   Post

Previous Trading Day: Investor Caution Continues Under New President

10:30 AM ET Jan 23, 2017 Morningstar

U.S. Markets U.S. markets began Monday's trading session with slight losses as investor uncertainty continues to be a main theme under President Donald Trump.

The Dow Jones Industrial Average was down, with McDonald's ... Read Full Story

Live News Briefs

MT Newswires' Analyst Notebook: KLAC

4:00 PM Apr 17, 2017 - MT Newswires

Market Movers

NYSE ⌄   Gainers (%) ⌄

COMPANY	TODAY'S CHANGE (%)	
Alere Inc	+15.77%	████
China Cord Blood Corp	+13.36%	███
Invacare Corp	+11.43%	███
CIRCOR International Inc	+9.89%	██
Enzo Biochem Inc	+8.87%	██
Axovant Sciences Ltd	+8.59%	██
Genie Energy Ltd	+7.91%	██
Systemax Inc	+7.54%	██
Mesabi Trust	+7.29%	██
Pzena Investment Management Inc	+5.73%	█

Sector & Industry Outlook

Sectors   Industries

SECTOR	3 MONTH CHANGE (%)	
Electronic Technology	+$9.34	
Consumer Non-Durables	+$8.87	
Consumer Services	+$7.19	
Utilities	+$6.87	
Consumer Durables	+$6.72	
Retail Trade	+$6.06	
Technology Services	+$6.04	
Health Technology	+$5.70	
Commercial Services	+$4.70	
Health Services	+$3.84	

TOP PERFORMING SECTOR OVER 1 YEAR

Electronic Technology   ▲ 35.73%

734 companies

BOTTOM PERFORMING SECTOR OVER 1 YEAR

Distribution Services   ▲ 3.17%

285 companies

News & Commentary

Today's Headlines

6:37 AM ET
MT Newswires

58.com Used Goods Platform Zhuan Zhuan Gets $200 Mln Investment Commitment From Tencent
06:37 AM EDT, 04/18/2017 (MT Newswires) -- 58.com (WUBA), the operator of an online marketplace in China, said Hong Kong Stock Exchange-listed Tencent Holdings has agreed to invest $200 million in its used goods trading platform Zhuan Zhuan. Under t...
Read Full Story

6:32 AM ET
Thomson Reuters

MOVES-Markel International Singapore names new marine liability underwriter
April 18 (Reuters) - Markel International Singapore, subsidiary of U.S.-based Markel Corp, appointed Colin Fordham as a senior marine liability underw...
Read Full Story

6:29 AM ET
MT Newswires

--UK Prime Minister Theresa May Calls For Early General Election on June 8
06:29 AM EDT, 04/18/2017 (MT Newswires) -- ...
Read Full Story

6:27 AM ET
Thomson Reuters

Sterling edges up, FTSE falls after UK PM calls early election
LONDON, April 18 (Reuters) - Sterling inched higher while UK stocks hit a session low on Tuesday after British Prime Minister Theresa May shocked mark...
Read Full Story

6:26 AM ET
MT Newswires

Aurinia Pharma Signs Licensing Deal with Merck Animal Health for VOS as Treatment of Canine Dry Eye Syndrome
06:26 AM EDT, 04/18/2017 (MT Newswires) -- Aurinia Pharmaceuticals (AUPH) said Tuesday that it has signed a definitive agreement granting Merck Animal Health, the global animal health business unit of Merck (MRK), worldwide rights to develop and co...
Read Full Story

6:25 AM ET
Thomson Reuters

UPDATE 1-Insurer UnitedHealth's quarterly profit, revenue beats
(Adds details)
April 18 (Reuters) - UnitedHealth Group Inc, the largest U.S. health insurer, reported a better-than-expected quarterly revenue and prof...
Read Full Story

8. CONCLUSION

The system developed has been tested and with various metrics. The main aim of this project is to ease out the bottleneck of creating the entire application. It instead makes use of already developed component. Usage of Open F2 framework in turn provides tighter integration thereby providing a good user experience. The user interface self-explanatory there by increasing the comfort level of the user. The application is very secure since end to end encryption and decryption algorithms are used. It is highly responsive and responds quickly to user commands.

The internship project was a great learning experience. I was exposed to various technologies like Dojo, Open F2 and Liferay. Working with the latest technology in the market gave me immense pleasure. I was taken through various phases of project development. I'm thankful to my project team members; they were extremely supportive and helpful. I thank SASTRA University and my IT department professors for providing such a wonderful opportunity.

9.REFERENCES

https://inautixonline.inautix.com/

https://dojotoolkit.org

http://openf2.org

https://www.w3schools.com

https://www.tutorialspoint.com

https://en.wikipedia.org

https://dev.liferay.com/develop/tutorials