Group 1

**UI Shop**

Software Design Document

Name (s):

Chen Xi 陈曦 16126135

Li Chenyang 李晨阳 16126163

Wang Dongjin 王东晋 16126205

Lab Section:

Workstation:

Date: (12/25/2016)

**TABLE OF CONTENTS**

[**1.**](#4) [**INTRODUCTION**](#4) [**2**](#4)

[1.1](#4) [Purpose](#4) 2

[1.2](#4) [Scope](#4) 2

[1.3](#4) [Overview](#4) 2

[1.4](#4) [Reference Material](#4) 2

[1.5](#4) [Definitions and Acronyms](#4) [2](#4)

[**2.**](#4) [**SYSTEM**](#4)[**OVERVIEW**](#4) **2**

[**3.**](#4) [**SYSTEM**](#4)[**ARCHITECTURE**](#4) [**2**](#4)

[3.1](#4) [Architectural Design](#4) 2

[3.2](#5) [Decomposition Description](#5) 3

[3.3](#5) [Design Rationale](#5) [3](#5)

[**4.**](#5) **M**[**ODULE DESIGN**](#5) [**3**](#5)

[4.1](#5) [Module Description](#5) 3

[4.2](#5) [Data Dictionary](#5) [3](#5)

[**5.**](#5) [**COMPONENT DESIGN**](#5) **3**

[**6.**](#6) [**HUMAN INTERFACE DESIGN**](#6) [**4**](#6)

[6.1](#6) [Overview of User Interface](#6) [4](#6)

[6.2](#6) [Screen Images](#6) [4](#6)

[6.3](#6) [Screen Objects and Actions](#6) [4](#6)

[**7.**](#6) [**REQUIREMENTS**](#6)[**MATRIX**](#6) **4**

[**8.**](#6) **I**[**NSTALL GUIDELINE**](#6) [**4**](#6)

[**9.**](#6) **SUMMARY** [**4**](#6)**. INTRODUCTION**

**1.1  Purpose**

Identify  the  purpose  of  this  SDD  and  its  intended  audience.  (e.g.  “This  software  design

document describes the architecture and system design of XX. ….”).

**1.2  Scope**

Provide a description and scope of the software and explain the goals, objectives and benefits

of your project. This will provide the basis for the brief description of your product.

**1.3  Overview**

Provide an overview of this document and its organization.

**1.4  Reference Material**

*This section is optional.*

List any documents, if any, which were used as sources of information for the test plan.

**1.5  Definitions and Acronyms**

*This section is optional.*

Provide  definitions  of  all  terms,  acronyms,  and  abbreviations  that  might  exist  to  properly

interpret the SDD. These definitions should be items used in the SDD that are most likely not

known to the audience.

**2. SYSTEM OVERVIEW**

Give a general description of the functionality, context and design of your project. Provide any

background information if necessary.

**3. SYSTEM ARCHITECTURE**

**3.1  Architectural Design**

Our project is an online trading system for designers and buyers of design works. Our system is B / S structure. The front-end part of our system is based on the Html5 and CSS3 standards, and we also use of AngularJs, jQuery, Twitter Bootstrap frameworks to support our system. The backend part of the system we use Node.js to provide restful api, and use mongodb to save our data. To make our system more robust, we introduced unit tests based on the karma and Jasmine frameworks in the front-end.

We use these frameworks because we want to provide users with a better user experience. Using the AngularJS framework allows us to build a SPA (Single Page Application). This will allow the user's browser to load all the necessary JavaScript and css files at once, and in the course of the next step, AngularJS will dynamically load and display the data required by the user's actions. The whole process is very fast, will greatly reduce the user's waiting time.

Front part is divided into ui-shop, ui-shop-designer, ui-shop-buyer three module, respectively responsible for all front page, designer background, buyer background. Each module also has some controllers and services for displaying data and responding to user actions.

The use of the Twitter Bootstrap framework and CSS3 enables us to provide a responsive layout application interface. This will provide a good experience for mobile devices. In this mobile Internet era, it is a must choice.

Background is currently providing the main user, work, order three data related API interface. API interface conforms to restful API interface specification. Site background database is MongoDB. MongoDB is a No Sql database, has a very good query performance, can greatly enhance the user experience. Node.js is currently the best support for MongoDB in all backend development technologies, so we chose it as our backend technology solution. And the site backend is based on Express framework.

**3.2  Decomposition Description**

Provide a decomposition of the subsystems in the architectural design.  Supplement with text

as needed. You may choose to give a functional description or an object­oriented description.

For  a  functional  description,  put  top­level  data  flow  diagram  (DFD)  and  structural

decomposition  diagrams.  For  an  OO  description,  put  subsystem  model,  object  diagrams,

generalization  hierarchy  diagram(s)  (if  any),  aggregation  hierarchy  diagram(s)  (if  any),

interface specifications, and sequence diagrams here.

**3.3  Design Rationale**

Discuss the rationale for selecting the architecture described in 3.1 including critical  issues

and  trade/offs  that  were  considered.  You  may  discuss  other  architectures  that  were

considered, provided that you explain why you didn’t choose them.

**4. MODULE DESIGN**

**4.1  Module Description**

In the font-end part, we have 3 modules. They are ui-shop, ui-shop-designerbacke, ui-shop-buyerbackend.

The ui-shop module is response for the foreground part. Includes display data, response to the user’s action, and send requests to the server then deal with the retrivaled data. In the foreground, buyer can search the work, view the detail of the work, and add work to the shopping cart or buy the work.

In the buyer background, buyer can see works they already paid, download the work and confirm the work.

In the designer background, designers can see their works, and the state or orders.

All these module are based on AngularJS. AngularJS is a MVC framework for the backend web application. It has view, controller and other conponents. And it can support bidirectional data binding, so it is very convinient to implement the mvc architecture.

**4.2  Data Dictionary**

Alphabetically list the system entities or major data along with their types and descriptions. If

you  provided  a  functional  description  in  Section  3.2,  list  all  the  functions  and  function

parameters. If you provided an OO description, list the objects and its attributes, methods and

method parameters.

**5. COMPONENT DESIGN**

In this section, we take a closer look at what each component does in a more systematic way. If

you gave a functional description in section 3.2, provide a summary of your algorithm for each

function  listed  in  3.2  in  procedural  description  language  (PDL)  or  pseudocode. If  you  gave  an

OO description, summarize each object member function for all the objects listed in 3.2 in PDL

or pseudocode.  Describe any local data when necessary.

**6. HUMAN INTERFACE DESIGN**

**6.1  Overview of User Interface**

Describe  the  functionality  of  the  system  from  the  user’s  perspective.  Explain  how  the  user

will  be  able  to  use  your  system  to  complete  all  the  expected  features  and  the  feedback

information that will be displayed for the user.

**6.2  Screen Images**

Display  screenshots  showing  the  interface  from  the  user’s  perspective.  These  can  be  hand­

drawn or  you can use an automated drawing tool.  Just  make them as  accurate as possible.

(Graph paper works well.)

**6.3  Screen Objects and Actions**

A discussion of screen objects and actions associated with those objects.

**7. REQUIREMENTS MATRIX**

Provide a cross­reference that traces components and data structures to the requirements in your

SRS document.

Use  a  tabular  format  to  show  which  system  components  satisfy  each  of  the  functional

requirements from the SRS.  Refer to the functional requirements by the numbers/codes that you

gave them in the SRS.

**8. Install Gidelines**

**8.1  Install Node.js**

You can reference [This Document](http://www.cnblogs.com/pigtail/archive/2013/01/08/2850486.html)

**8.2  Install MogoDB**

You can reference [This Document](http://jingyan.baidu.com/article/d5c4b52bef7268da560dc5f8.html)

**8.3  Run MongoDB**

1. copy data folder to C:/

2. run C:/data/startup.bat

**8.4  Build project**

1.get into UIShop folder

2. run 构建项目.bat

**8.5 Run project**

1.get into UIShop folder

2. run 启动项目命令.bat

3.browse http://localhost:3000

**9. Summary**

In this project, we used the AngularJS framework to build the Single Page Application. This is a very unique experience. Using this technology makes our product performance is very good, the user experience is great, it is very rare harvest.

At the same time, we also use the unit testing technology to make our front-end code more robust. Karma and Jasmine are great test frameworks.

Finally, thank the teacher for a semester of hard teaching, gives us a chance to experience new technologies.