**Software Development Project Plan**

**Food Order-And-Delivery Website**

*Li Chen #661693723*

*Jianhui Lu #661752675*

*Weiran Yuan #661677444*

*[Version 2.0]*

*[Version 12/14/17]*

|  |  |  |  |
| --- | --- | --- | --- |
| *Version* | *UpdateDate* | *Authors* | *Comment* |
| 1.0 | 10/28/17 | Li Chen  Jianhui Lu,  Weiran Yuan | Starter version |
| 1.0 | 12/14/17 | Li Chen  Jianhui Lu,  Weiran Yuan | Final version |

Table of Contents

[1. Overview 2](#_Toc501131656)

[1.1 The goals and scope 2](#_Toc501131657)

[1.2 Assumptions and Constrains 3](#_Toc501131658)

[1.3 The stake holders 3](#_Toc501131659)

[2. The Deliverables 4](#_Toc501131660)

[2.1 Final product 4](#_Toc501131661)

[2.2 Interfaces deliverables 4](#_Toc501131662)

[2.3 Documentation delivery 5](#_Toc501131663)

[3. The Project Organization 5](#_Toc501131664)

[3.1 External organization 5](#_Toc501131665)

[3.2 Internal organization 6](#_Toc501131666)

[3.3 Role and work dividend 7](#_Toc501131667)

[4. The Project Management 7](#_Toc501131668)

[4.1 methodology 7](#_Toc501131669)

[4.2 model 7](#_Toc501131670)

[4.3 framework 8](#_Toc501131671)

[4.4 the development environment and tools 8](#_Toc501131672)

[5. The Project Schedule 8](#_Toc501131673)

[5.1 The milestones for each delivery 8](#_Toc501131674)

[6. The Quality Assurance 9](#_Toc501131675)

[6.1 Component Quality Assurance 10](#_Toc501131676)

[6.2 Configuration Quality Assurance 11](#_Toc501131677)

[7. The Risk Management 11](#_Toc501131678)

# Overview

This document is to serve as a guidance for us to choose models and implement the methodology, so that the final product system and functions could meet or exceeds customer’s/our initial requirement. It mainly discusses topics on: the feature of product, organization and structures, project management, working schedule, quality assurance, risk management and configuration management. This management plan not only describes the purpose, scope, and objectives of our project, but also contains our ground assumptions and constraints and evaluation of product.

To implement the project, we adopt Feature Driven Development, one of agile models to develop our products. We strictly follow our feature list to analyze method, design plan, build features with development tools and conduct testing cases, this model also offers a framework for us to execute our ideas.

The project overall can be divided into 3 basic interface: ordering interface, menu management interface and ordering retrieval interface. And in further we can develop more dimensional service like racking food interface and customer rating interface.

All data that generated for the project or by the product would be well structured and stored in a SQL database. The data generated by users would be implement into relational model.

To evaluate and avoid flaws our project, we apply Quality Assurance to check whether everything on feature lists is perfectly implemented. To avoid big change risk, business risk and release risk, we also include Risk Management into our project.

## The goals and scope

The goal of our project is to develop a 3-tier web application, which includes the cooperate of browser, server and database, using what we covered in Software Development class and online resources.

The content of our project is to develop a food order-and-delivery website, allowing customers reach food and restaurants franchising. This product offers more choices and better comparison of restaurants for customers, and large advertising and consuming platform for restaurants.

## Assumptions and Constrains

There are several assumptions and constraints that are of importance for the project.

Important assumptions:

* Our team can refer, use and modify all resources we have access to, including course materials, online examples;
* Members have software development experience and are able to work efficiently together;
* Our members can play clients’ role and provide revise advices during the implementation process;

Main constrains:

* Additional financial sponsors are not available in this project.
* The individual working hours would be limited to 10 hours per week during this semester.
* Frequent group meeting may be not available for each week.
* Although group members can give feedback as clients, there’s insufficient customer involvement in our developing and maintaining process.

## The stake holders

The stakeholders include:

* Potential customer: they have easily access to relatively comprehensive information from our products so it’s convenient and time-saving to choose food;
* Franchise restaurants: they are offered a large platform to advertise and serve food, which brings them more potential customers and profits;
* Product owners (developers): they can make profit by charging restaurants initial fee and make difference profit as intermediary.

# The Deliverables

## Final product

Our final product would be a food order-and-delivery website. There would be dozens of restaurants on the web and customers scan and order food online, then wait for the food delivery. The website could serve the following functions:

Support customers to:

* Create an account and log in, or purchase as guest to order their food;
* Have options to pick-up at store or delivery;
* Input and save their delivery address and payment information in their account;
* Have access to their history orders;
* Star-mark their favorite restaurants, so that they don’t have to search;
* Have options to pay their orders online (Paypal, Venom) or offline;
* Check their food status online;
* Telephone to restaurants to modify their orders;
* Give ratings and comments to restaurants.
* Get loyalty refund or bonus;
* Send suggestions to developer team;

Support franchise restaurants to:

* Create an account and log in, post and update their food menu, pictures, prices and special dishes etc, on our website.
* Receive notifications of new orders.
* “close” their windows in closing hours
* Send suggestions to developer team;

## Interfaces deliverables

* Ordering Interface

Users of the web application interact with it via a series of simple forms. Each category of food will have its own form associated with it, like drop-down menus for users to select category they want, simple click to edit food order. This part also includes payment transactions.

* Menu Management Interface

Interactions with the menu management interface is quite similar to those with the web ordering interface. An function to support upload picture would be included in this interface, and additional things like google map may be contained.

* Ordering Retrieval Interface

The application will automatically ass new orders from customer interface to database, fetch new orders at regular intervals and display the information to the restaurant end, such as the order numbers, order detail, delivery time and any comments, in a panel on the left hand side of the application.

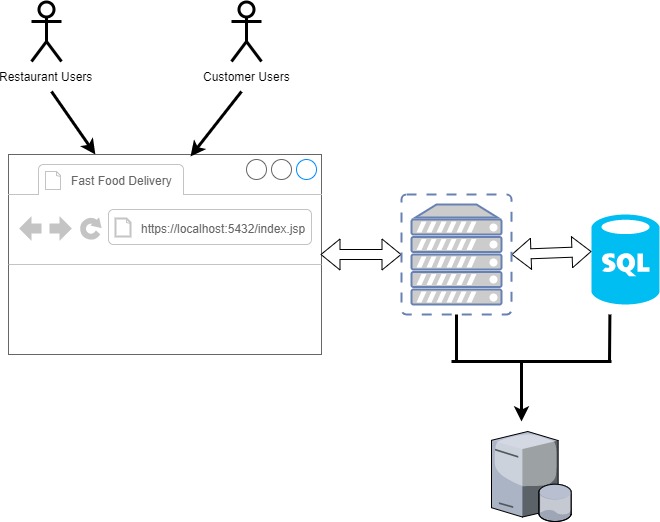
## Documentation delivery

Development documentation would be generated by developers and save in local repository and GitHub project repository, for the main reference of planning and developing outlines. The additional data generated by the usage of final product, such as customers’ information, restaurants’ information and order details would be transfer into tables of relational model, and stored in database server.

# The Project Organization

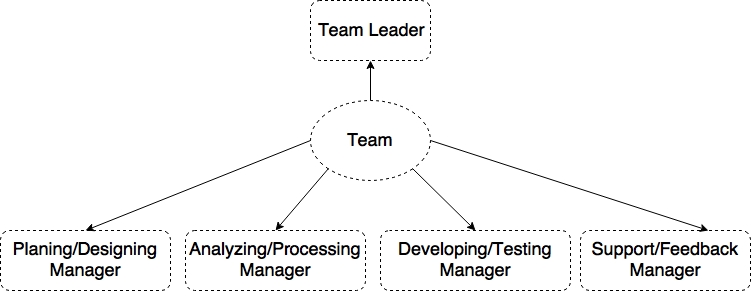
## External organization

Our final product connects customers and franchise restaurants through our server and database. Ideally, consumers and restaurants are able to get to our website though their browsers, restaurants can get order notifications and we are responsible to control and manage the database and server.



## Internal organization

The team structure includes a team leader and several section managers. We divide the sections into plan and design, analysis and process, develop and test, support and collect feedback. This work dividend is scientific since it exactly meets the life circle of our model, each step is managed rigorously and iteratively.



## Role and work dividend

|  |  |
| --- | --- |
| Roles | Responsibilities |
| Team Leader | * To organize and manage the team, including divide work, control schedule and check progress |
| Planning/Designing Manager | * To do starter work, including producing task plan, manage time schedule, track the progress and revise the plan if needed |
| Analyzing/Processing Manager | * To grade the plans, measure potential challenge and risk, develop a detailed tool, overall follow on the coding thing |
| Developing/Testing Manager | * To realize the plans using development tools, including handling codes, fixing bugs, designing testing case, improve algorithm and efficiency |
| Support/Feedback Manager | * To cooperate developing team to maintain the product, collect feedback from users and data, get involved in re-planning and analyzing. |

# The Project Management

## methodology

There’re two methodology that we can choose from, the waterfall and agile.

Considering that feedback from user is very important in our product, we need a more flexible and easy-to- change methodology. Therefore, we choose to use agile methodology in implement our model.

## model

From our final product description, the goal of our product is to implement each feature, therefore Feature Driven Model would be used. The feature of our project is small, client-valued function expression, and the final product is the combination of many feature functions. Therefore, FDD could be the most efficient and easy-to-expand model for us.

## framework

we are going to implement the model as following process:

|  |  |
| --- | --- |
| Develop overall model | * Choose the model * Study document * Divide workload * Write model notes |
| Build feature list | * Layout feature list * Determine sequence |
| Plan and design | * Interactive features * Study documentation * Refine models * Layout method procedure |
| Realize the feature | * Implement class and methods * Implement code * Conduct tests * Remote control |

## the development environment and tools

|  |  |
| --- | --- |
| Documentation and Communications | * [Project wiki](http://wiki.collectionspace.org/) * Github * Wechat group |
| Source Code Version Control | * [Git](http://git-scm.com/) * [GitHub](https://github.com/collectionspace/) - public Git repositories, Project source code repository |
| Build, Packaging, and Deployment tools | * Eclipse * PostgreSQL database * Atom, SublimeText |
| [Bug and Issue Tracking](https://wiki.collectionspace.org/display/collectionspace/Issue+Tracking) | * [Project issue tracking system](https://wiki.collectionspace.org/display/collectionspace/Issue+Tracking) |

# The Project Schedule

## The milestones for each delivery

|  |  |  |
| --- | --- | --- |
| **Project milestone** | **Tools** | **Due date** |
| Project start: environment settings, IDE installations | Tomcat/ JSP/ Eclipse | 10-30 |
| Website overall design: web style | Tomcat/ JSP/ Eclipse | 11-06 |
| Customer account page: create account, login, account management | Tomcat/ JSP/ Eclipse | 11-10 |
| Customer view page: view lists of food | Tomcat/ JSP/ JavaScript/Eclipse | 11-10 |
| Place order page: shopping cart, order information, calculate prices and tips, payment, shipping method, | Tomcat/ JSP/JavaScript/ Eclipse | 11-14 |
| Customer order history page: search history | Tomcat/ JSP/ JavaScript/Eclipse | 11-17 |
| Restaurant update menu page | Tomcat/ JSP/ Eclipse | 11-17 |
| Restaurant receive order notification page\*\* | Postgres Database | 11-20 |
| Create customer table in database | Postgres Database | 12-09 |
| Create restaurant table in database | Postgres Database | 12-09 |
| Create order table in database | Postgres Database | 12-09 |
| Test Code | Tomcat/JSP/Eclipse/PostgreSQL | 12/13 |

# The Quality Assurance

Quality assurance ensures the project processes are used effectively to produce quality project deliverables. It involves following and meeting standards, continuously improving project work, and correcting project defects.

* The project processes subject to quality assurance.
* The quality standards and customer/restaurants expectations for that process.

## 6.1 Component Quality Assurance

Since our model is Feature-Driven-Development, here we plan to apply Process Analysis tool, which to include a Plan-Do-Check-Act cycle to monitor that project processes are properly followed.

Features we implemented successfully:

* Allow client-users to create an account and log in;
* Allow client-users to view different restaurant;
* Allow client-users to select food from a restaurant;
* Allow client-users to have options to choose delivery;
* Allow client-users to input and save their delivery address and payment information in their account;
* Allow client-users to pay their orders offline;
* Allow client-users to check their food status online;
* Allow client-users to have access to their history orders;
* Allow client-users to telephone to restaurants to modify their orders;
* Allow restaurant-users to create an account and log in;
* Allow restaurant-users to post and update their food menu, prices and special dishes etc, on our website.
* Allow restaurant-users to receive notifications of new orders.

Which wasn’t implemented properly:

* Allow client-users to star-mark their favorite restaurants, so that they don’t have to search;
* Allow client-users to have options to pay their orders online (Paypal, Venom)
* Allow client-users to give ratings and comments to restaurants.
* Allow client-users to get loyalty refund or bonus;
* Allow all users to send suggestions to developer team;
* Allow restaurant-users to “close” their windows in closing hours

## 6.2 Configuration Quality Assurance

Our web application needs server to request and response data from our database frequently, and server crushes occurs occasionally when interacting with PostgreSQL database, which makes our web product less stable and portable. One explained reason could be that pgAdmin doesn’t allow user to modify objects simultaneously, which could be violates when our application needs to retrieve and write data in the same time.

This problem might be solved by switch to other database which support higher speed interaction like MongoDB. MongoDB is one of non-relational database, store the data in BSON format. This database performs very good when handling with big data.

# The Risk Management

This project would be implemented under a series risk, which includes:

* Insufficient QA time to validate on all browsers and OS types
* Following end-user testing, more effort on the user guide may be necessary.
* Backup and restore may require third-party solutions
* Insufficient time for external stakeholders to submit feedback on layout and composition of reports.
* Procedures for contingency planning and the methods that will be used for tracking certain risk factors, changes in levels of the factors and responses to those changes.