OAMK
OULU UNIVERSITY OF
APPLIED SCIENCES

Heng Zhang

WEB APPLICATION DEVELOPMENT OF A JOB BROKER PLATFORM

WEB APPLICATION DEVELOPMENT OF A JOB BROKER PLATFORM

Heng Zhang
Bachelor’s Thesis
Autumn 2017
Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology

Author: Heng Zhang
Title of the bachelor’s thesis: Web Application Development of A Job Broker Platform
Supervisor: Pertti Heikkila
Term and year of completion: Autumn 2017 Number of pages: 35 + 4 appendices

The purpose of the thesis was to study the Job Broker platform project and supply platforms for the Job Broker.

The work focused on the feature design and development of Job Broker web application. CodeIgniter was used as the base framework. The thesis introduced distinctive user roles, the integration of social media and the Paypal Gateway, the user-friendly bidding module with the related full stack web development.

The project idea came from a small company, the initial plan was to build a convenient, practical middle platform. Although it has not released yet, the market and business value of Job Broker is still existing. This thesis can also be used as a technical reference or to provide a kind of idea in future for someone who considers implementing a similar application.

Keywords: Job broker, Codeigniter, Paypal, Social Media Login, Bidding

TABLE OF CONTENTS

VOCABULARY

UI/UX Designer

UX Design refers to the term **User Experience Design**, while UI Design stands for **User Interface Design**. Both elements are crucial to a product and work closely together. (CareerFoundry, Cited 18.12.2017)

LAMP

Software components, Linux, Apache, MySQL and alternatives, PHP and alternatives, used in the web development field.

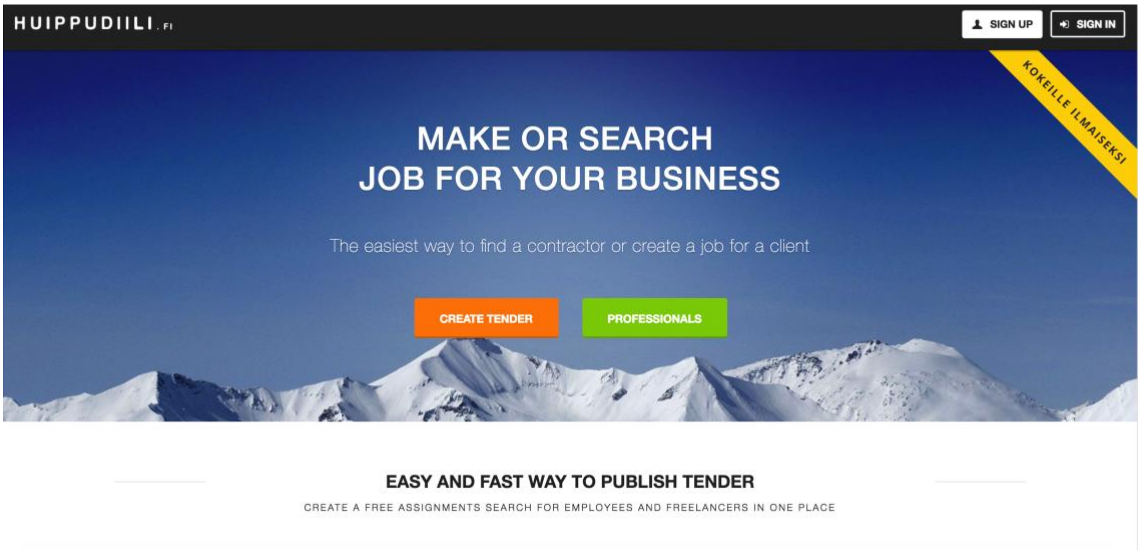
1 INTRODUCTION

By investigating the Finnish free occupation market, Finnish freelancers often encounter problems such as less project sources and low efficiency, Employers face the poor distribution and fewer platforms of publishing or the Information release area is too narrow. To alleviate these problems, the Job Broker platform project was created. The full site of the UI design is designed by the senior UX designers, and it provided an opportunity to develop the project. This paper will introduce the design process, development flow, main technical point and unique features of development in the project.

2 INTRODUCTION OF MAIN FEATURES

2.1 Frontend UI

Obviously, to be a nice web application, its Frontend User Interface is a component which is indispensable and cannot be ignored. A senior UI/UX designer is needed in the team.



MAKE OR SEARCH JOB FOR YOUR BUSINESS

The easiest way to find a contractor or create a job for a client

EASY AND FAST WAY TO PUBLISH TENDER

CREATE A FREE ASSIGNMENTS SEARCH FOR EMPLOYEES AND FREELANCERS IN ONE PLACE

FIGURE 1. The UI of the project home page

2.2 User Account Module

The user system is the foundation of all the other business logic. The Job broker platform has a flexible User account System. The user can choose “Freelance” and “Employee” as a user’s role. First each role has common basic settings, for example a user email, a username and basic personal info. But both of them have their unique functions. Below Figure 2 shows their difference in detail.

“Freelancer” has its own reference gallery to showcase the past project results. Naturally, the core feature of the Freelancer user can participate in the project bidding. The Bidding system will be discussed more later.

Normally, in a web application the user roles are clearly distinguished. There is not only there is less common features between different roles, but also there is no converting between different roles. But in the platform, once the Freelancer has published a project, they will automatically convert to the 'Third' role which has the all the functions and settings from the Freelancer account and Employee account. Similarly, when the Employee participates in bidding a project, they will have the Freelancer's features also.

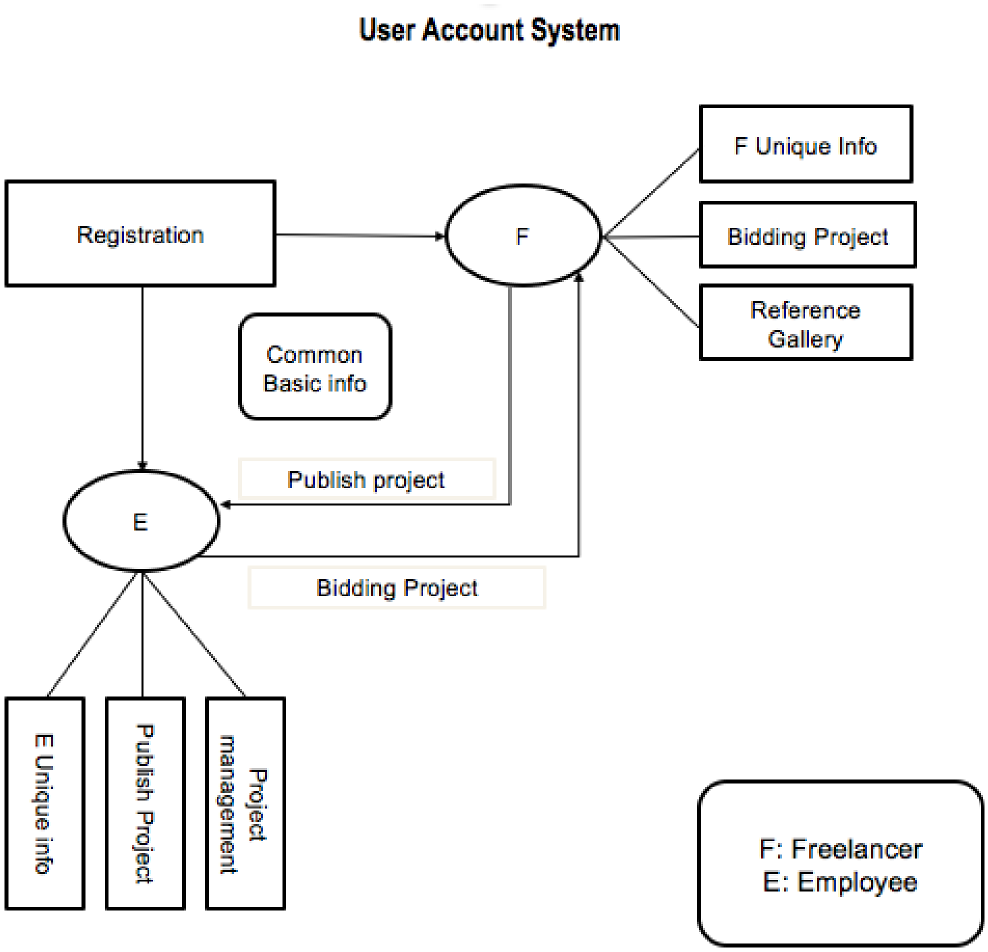


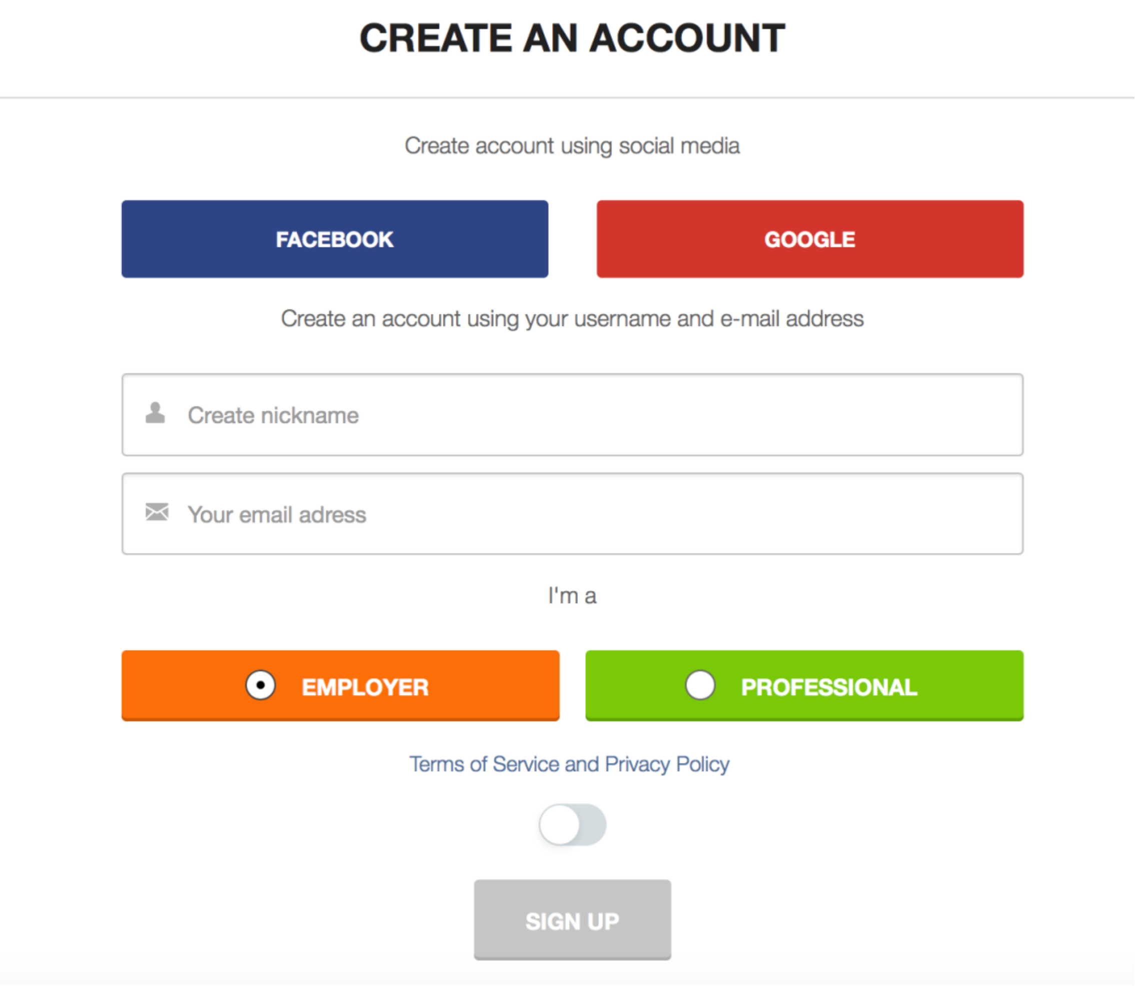
FIGURE 2. User Account System Schematic Diagram

2.2.1 Account Registration

Registration is a door to browse all features of the application. It usually contents a registration form, a request register, an active account and some affiliated functions.

Nowadays, several applications have really long registration process and a complex form, so that many users stop the registration in the beginning of it. A complex registration may obstruct the user from creating a user account for the application. The user can even completely stop using the application.

In this application, there is a key point for registration to be considered and discussed few times during the development. That is the “Convenient and Fast”. In order to make the registration easy and fast to use, the Social Media Login is integrated into the board. The registering user only needs one click to connect their Social Media account. Then user can complete the registration in few Clicks without creating an active account. At this moment, only Facebook and Google+ are integrated. More detail will be described in chapter 3.4 Social Medial Integration.



CREATE AN ACCOUNT

Create account using social media

Create an account using your username and e-mail address

Create nickname

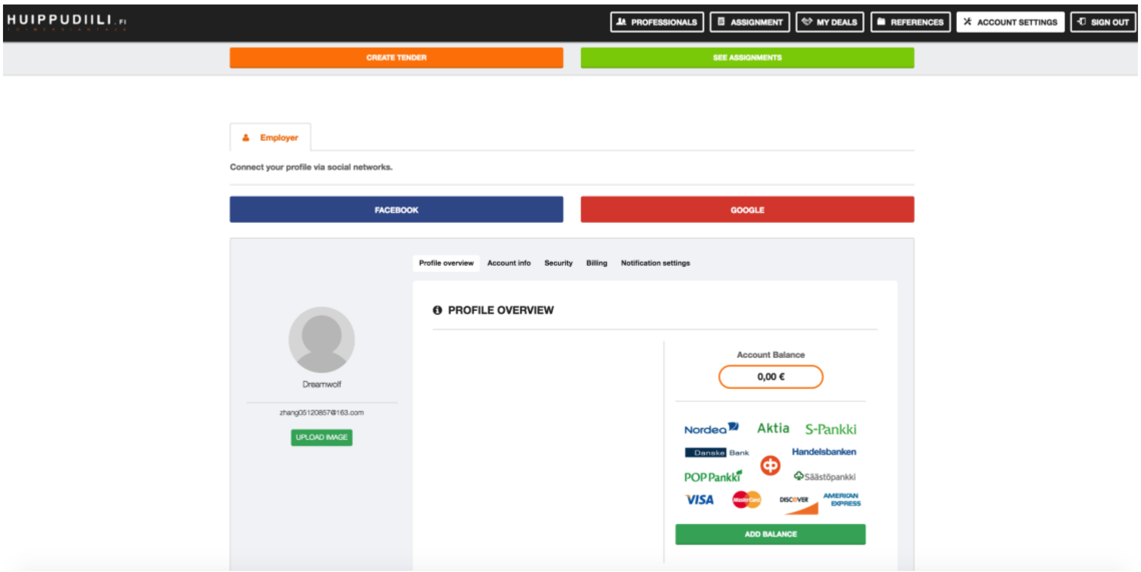
Your email adress

I'm a

● EMPLOYER ○ PROFESSIONAL

FIGURE 3. Registration form

The Social Media Login inspired me because no other program makes the normal registration as easy as the Social Media login. When the register is thought the key info needed is an email address. The other information can be filled in later to the user profile. The password is first generated from the system, and sent to the user. The user has to do the ‘Account active’ in the email. Then they can change the password as their wish. The benefit of this process is that it can be confirmed that the user’s account is a real account. Naturally there is a risk for the spam or robot register. Thus the active account requirement is a protected method. The user has to do the account active manually from the email notification. Once the user’s account has been activated, they can go to the user profile (See FIGURE 4 below) complete the user’s information at any time they want.



Connect your profile via social networks.

Dreamwolf

zhang05120857@163.com

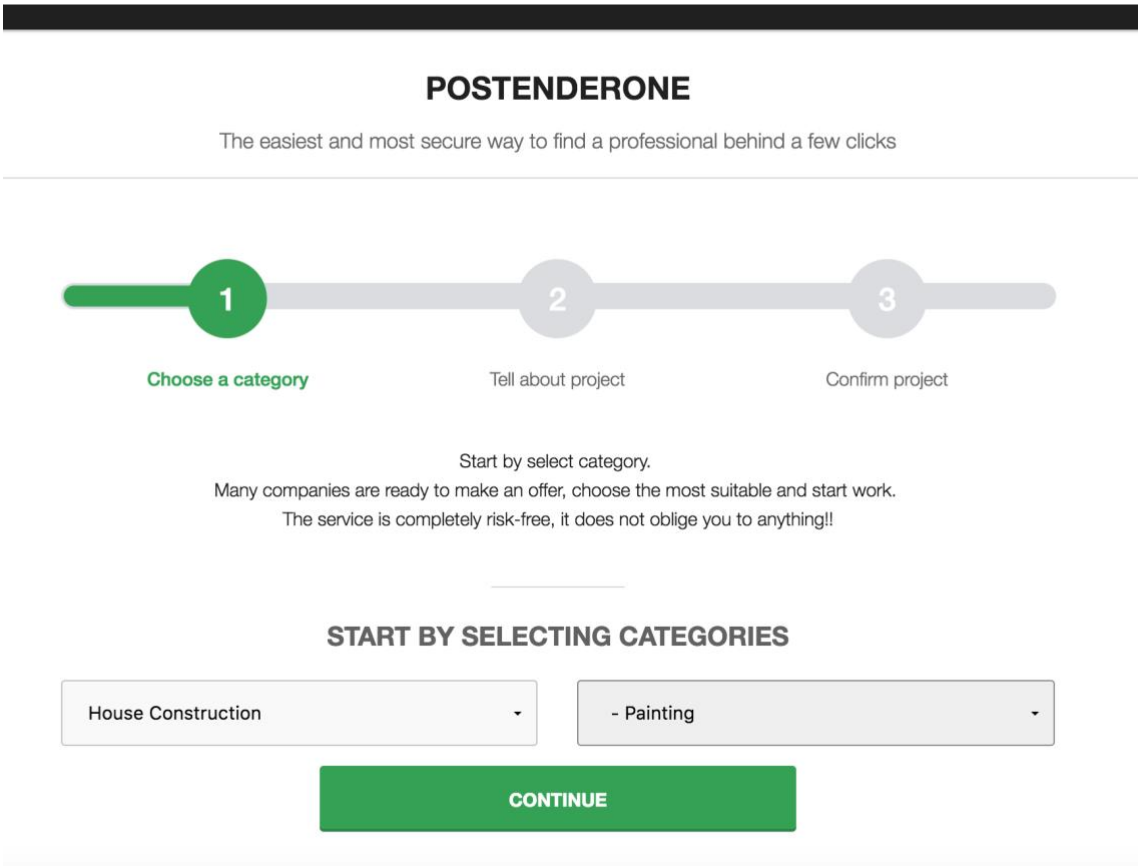
Account Balance

0,00 €

FIGURE 4. User Profile

2.3 Project Publishing and Management Module

The project publishing module is a typical source in the platform projects chain. From the business logic, the usability to process a performance, the steps of the publishing have been designed, tested and improved few times. The target was to follow some keys: Streamlining steps, getting core information only in forms, improving the backend business logic and database access performance. During the publishing processing Employee can choose the fixed Project commissions or set an allowed bidding range. (See Figures 5 and 6 below)



POSTENDERONE

The easiest and most secure way to find a professional behind a few clicks

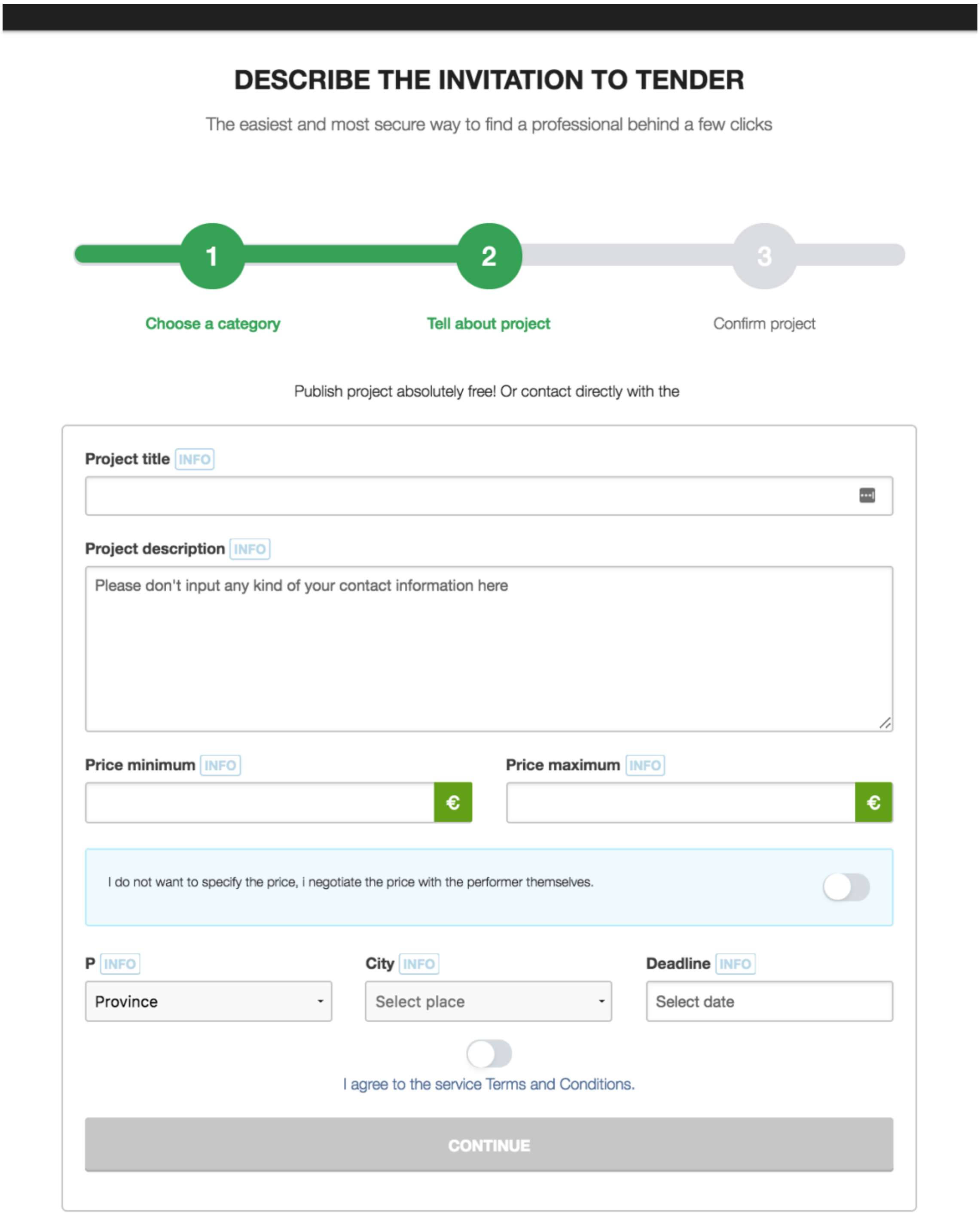
Start by select category.
Many companies are ready to make an offer, choose the most suitable and start work.
The service is completely risk-free, it does not oblige you to anything!!

START BY SELECTING CATEGORIES

House Construction

- Painting

FIGURE 5. The Project Publishing feature step 1



DESCRIBE THE INVITATION TO TENDER

The easiest and most secure way to find a professional behind a few clicks

Publish project absolutely free! Or contact directly with the

Project title INFO

Project description INFO

Please don't input any kind of your contact information here

Price minimum INFO €

Price maximum INFO €

I do not want to specify the price, i negotiate the price with the performer themselves.

P INFO: Province

City INFO: Select place

Deadline INFO: Select date

I agree to the service Terms and Conditions.

FIGURE 6. The Project Publishing step 2

2.4 Project Bidding Module

While there are projects pending in the platform, the bidding function is activated. Every account can participate in the project bidding.

According to the publish type of bidding system, there are two ways for bidding. The employee set the range of the price that must be paid. When the user participates the bidding, they can give their own bid price in the bidding. If there is no set price or if there is an unlimited price, then the user can follow their own wishes to participate in the bidding and to show their reference. This is very convenient for the Employee, who is not limited for bidding, but they can screen among of bidders by browsing their references. However, the bidder can enable the reference displaying or not.

2.5 Mail Notification

This should also belong to the basic functions as a web application. In this application, the Mail notification includes Registration Confirmation, Account Active, Password Change, Project Publish Confirmation and Project Bidding Notification (when the project status has changed.)

3 TECHNICAL MAINLY USED

3.1 Full-stack Web Developer the major foundations – LAMP

Basically, a full-stack developer is simply someone who is familiar with all layers of computer software development. These developers are not experts on everything. They simply have a functional knowledge and ability to take a concept and turn it into a finished product. (2)

3.2 Codeigniter Framework

CodeIgniter is a powerful PHP MVC framework with a very small footprint. It is built for developers who need a simple and elegant toolkit to create full-featured web applications. (1)

As a light framework, CI (CodeIgniter will be called as CI in the whole thesis later) has its disadvantages: the community is not active enough, and the update is not timely enough, so that a lot of system bugs are not solved. But for small projects or multilingual programming projects, its advantages are quite obviously.

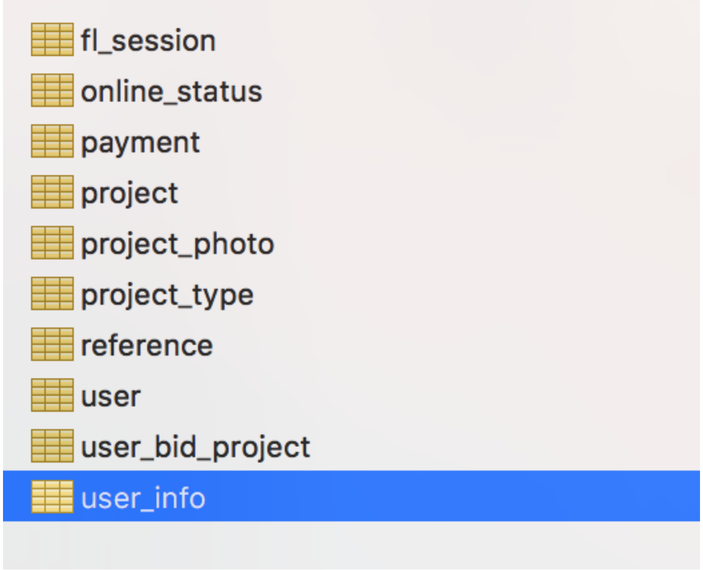
CI is easy to study and use because in the PHP world, it is often the data table as a model. Therefore, CI even needs to write View and Controller files, and then to use the Model where necessary, which means that in the process of learning CI, happy and simple, in just twenty minutes you can easily start writing CI program. (7)

3.3 MySQL database relational model

A Relational database organizes data in relations. The relationships that can be created among the tables enable a relational database to efficiently store a huge amount of data, and effectively retrieve the selected data. (3)

The project database is using a typical relational model.

To keep the scalability of ’user’ information, the ‘user_info’ table use a ‘key’ => ‘value’ to respond to the future demand, such as a new field of user info. And the primary key ‘user_id’ is cascade with the ‘user’ table.



- fl_session
- online_status
- payment
- project
- project_photo
- project_type
- reference
- user
- user_bid_project
- user_info

FIGURE 7. Tables used in the project

3.4 Social Media integration

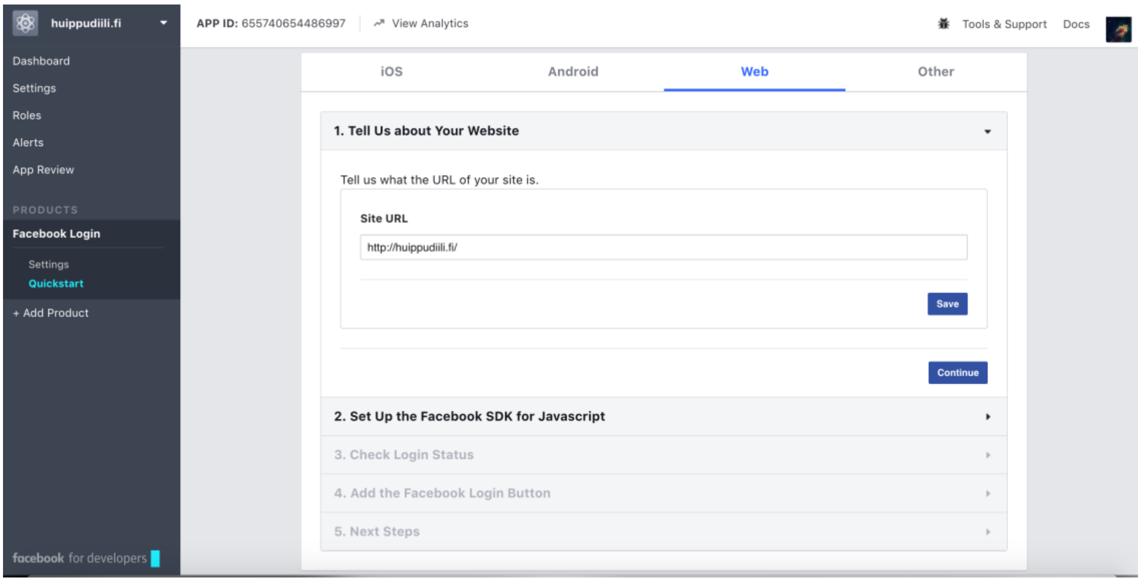
The purpose of utilising the Social Media integration is to simplify the registration and login process. There are two pieces of Social Media integrated into this platform, Facebook and Google+.

The Social Media integration requests few steps to make it work. In general, it can be distributed to four steps.

1. Registering and doing configuration in the developer panel.
2. Installing social medial lib/SDK into the project.
3. Implementing the custom feature related to the social media.
4. Deploying and testing the integration.

3.4.1 Facebook & Google+ Integrating

Facebook Login for Apps is a fast and convenient way for people to create accounts and to log into their app across multiple platforms. It is available on iOS, Android, Web, Windows Phone, desktop apps and devices, such as Smart TVs and Internet of Things objects. (4)



APP ID: 655740654486997

1. Tell Us about Your Website

Tell us what the URL of your site is.

Site URL: http://huippudiili.fi/

2. Set Up the Facebook SDK for Javascript

3. Check Login Status

4. Add the Facebook Login Button

5. Next Steps

FIGURE 8. Quick start with Facebook Login (4)

Facebook has a quick guide for assigning the Facebook Login to this project. This panel has been updated during the project developing time. When the Facebook Login was integrated, there was no such smart and convenient feature yet. The Developer documentation had to be followed step by step R&D. It was difficult at the beginning, but when the feature finally worked as expected, this feature had to be highlight. If a Retrospective study is done, the code is not that perfect, even when the PSRs (6) are not followed, but it is still possible to learn from the logic and idea.

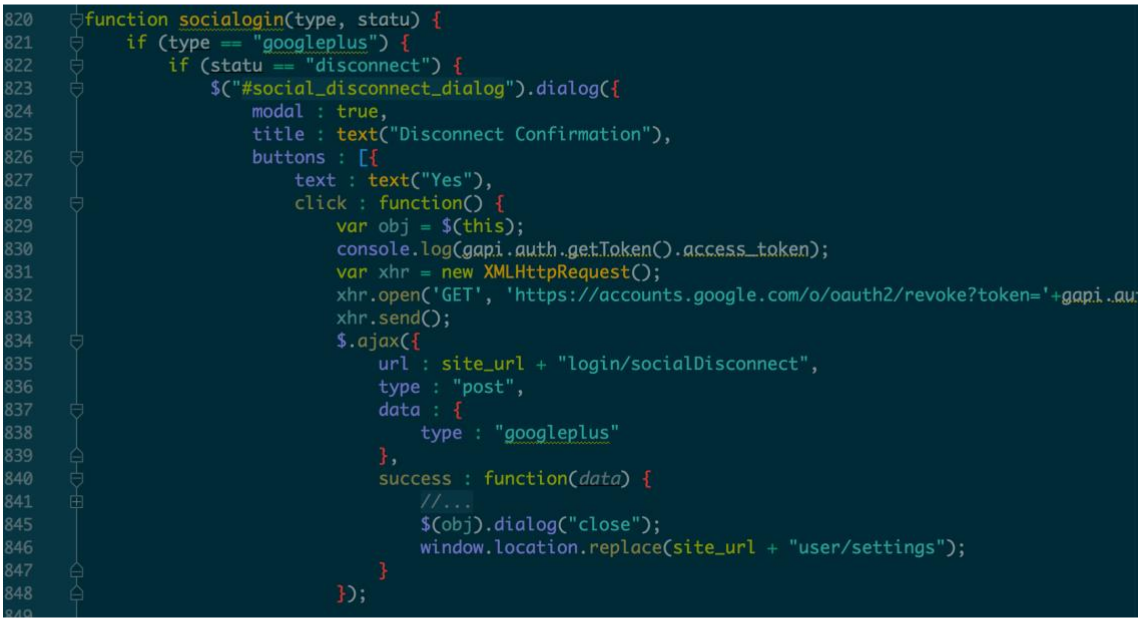
The highlight is a construction of functions, the whole social login divided into four levels and three status of functions.

1. Frontend display and style.
2. Javascript functions handle trigger, connection, redirect and error handling.
3. PHP Controller undertakes the Internal Ajax calls and responses.
4. Model is in charge of social media connection status.

The status of functions are Disconnect, Connect, Login.

* The Frontend will not be described too much in this thesis.

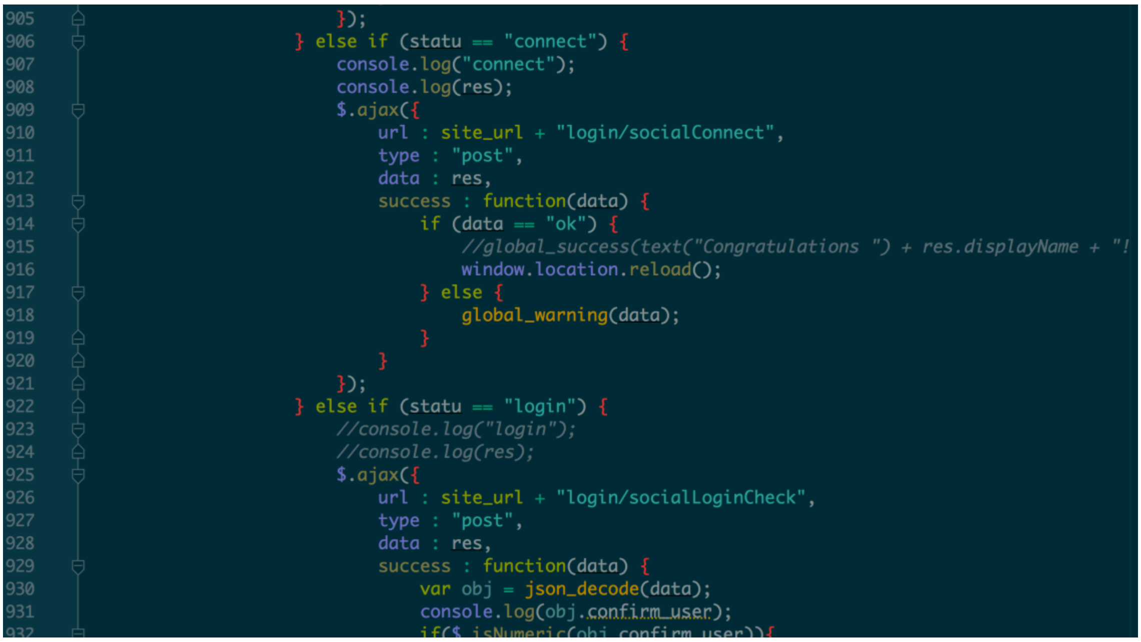
The whole features were triggered from Javascript, when the Facebook or Google+ login button is clicked, it will trigger one function called the ‘sociaLogin’ which shows in Figure 9 with its arguments. First the type is checked, then it has to separately call its own SDK or Lib. Then the second argument will decide which action will be assigned. At this stage, if all the requests are put into a function, it is going to be a huge bloated function, hardly readable and difficult to maintain. The best way is to piece the function to several reusable functions.



```
function socialogin(type, statu) {
    if (type == "googleplus") {
        if (statu == "disconnect") {
            $("#social_disconnect_dialog").dialog({
                modal : true,
                title : text("Disconnect Confirmation"),
                buttons : [{
                    text : text("Yes"),
                    click : function() {
                        var obj = $(this);
                        console.log(gapi.auth.getToken().access_token);
                        var xhr = new XMLHttpRequest();
                        xhr.open('GET', 'https://accounts.google.com/o/oauth2/revoke?token='+gapi.au
                        xhr.send();
                        $.ajax({
                            url : site_url + "login/socialDisconnect",
                            type : "post",
                            data : {
                                type : "googleplus"
                            },
                            success : function(data) {
                                //...
                                $(obj).dialog("close");
                                window.location.replace(site_url + "user/settings");
                            }
                        });
```

FIGURE 9. Social Media Login Feature Code 1

The detail working flow starts once the social login button is clicked. The ‘Dialog’ popup displays dynamically. Then the corresponding functions are called. The flow goes to the second stage, Ajax requests and responses. If the user is connecting for the first time, the basic account info, email and nickname, will be created and saved into the database via a user model. Thus, the typical sign is that the processing calls the models, which means that it is the third stage. The “connect” and “disconnect” action will change the social medial login status, The “login” action will do a search/compare query in the database. After that, the flow arrival the final stage, responses and displays in the frontend, the ”Dialog” update or error message is displayed or success is redirected. (See figure 10)



```
    });
} else if (statu == "connect") {
    console.log("connect");
    console.log(res);
    $.ajax({
        url : site_url + "login/socialConnect",
        type : "post",
        data : res,
        success : function(data) {
            if (data == "ok") {
                //global_success(text("Congratulations ") + res.displayName + "!
                window.location.reload();
            } else {
                global_warning(data);
            }
        }
    });
} else if (statu == "login") {
    //console.log("login");
    //console.log(res);
    $.ajax({
        url : site_url + "login/socialLoginCheck",
        type : "post",
        data : res,
        success : function(data) {
            var obj = json_decode(data);
            console.log(obj.confirm_user);
            if($.isNumeric(obj.confirm_user)){
```

FIGURE 10. Social Media Login Feature Code 2

Obviously, the Javascript functions have the most important role in the whole Social Login logic. It is responsible for transferring the request, handling error message, and responding to the frontend.

3.5 Payment gateway Paypal integration

PayPal, a division of eBay, is dedicated to enabling individuals or businesses to securely, simply and easily make online payments and receipts via e-mail. The PayPal account is PayPal's most secure online electronic account, which can effectively reduce the incidence of online fraud. (8)

Paypal provides few different ways to implement Paypal payment services. It is possible to choose the solution that bests suits the user’s needs. It provides a flexible way to implement the most suitable payment service.

To make a platform becomes a value service provider, the payment feature is an indispensable part. There are many payment gateways. There are some reasons for choosing Paypal as the payment gateway at the first stage.

1. Paypal has a wide range of applications. The customer can easily accept to use it.
2. Paypal has a high visibility, detailed development documents and a mature, rich community guidance
3. Paypal has a Safe, powerful sandbox test toolkit. It can help do the payment testing before the project goes to production.

When comparing the several payment gateway expenses, Paypal is the economic solution.

PayPal mainly has the Website Payment Standard and Express checkout. (Express Checkout, EC for short) is an enterprise-class solution designed to meet the growing business needs through a scalable platform. However, the Website Payment Standard (WPS) has problems such as a complicated integration time and complex settings. On the other hand, The WPS initial implementation is relatively quick and easy. Although it is not easy to scale up and there is a lot of work to be done to get and manage reconciliation information, there are still obvious advantages for a start-up company and a small corporate project.

The main differences are as follows:

Express Checkout (EC)

- Communicating with PayPal via API calls
- Powerful business solutions
- Deep integration in web applications
- Instant response
- Business account setup is slightly more complicated
- Integration takes a long time

Website Payment Standard

- Using Form Post to communicate with PayPal
- Integration is very simple and fast
- Business account settings easier
- Non-immediate response
- Products for small businesses

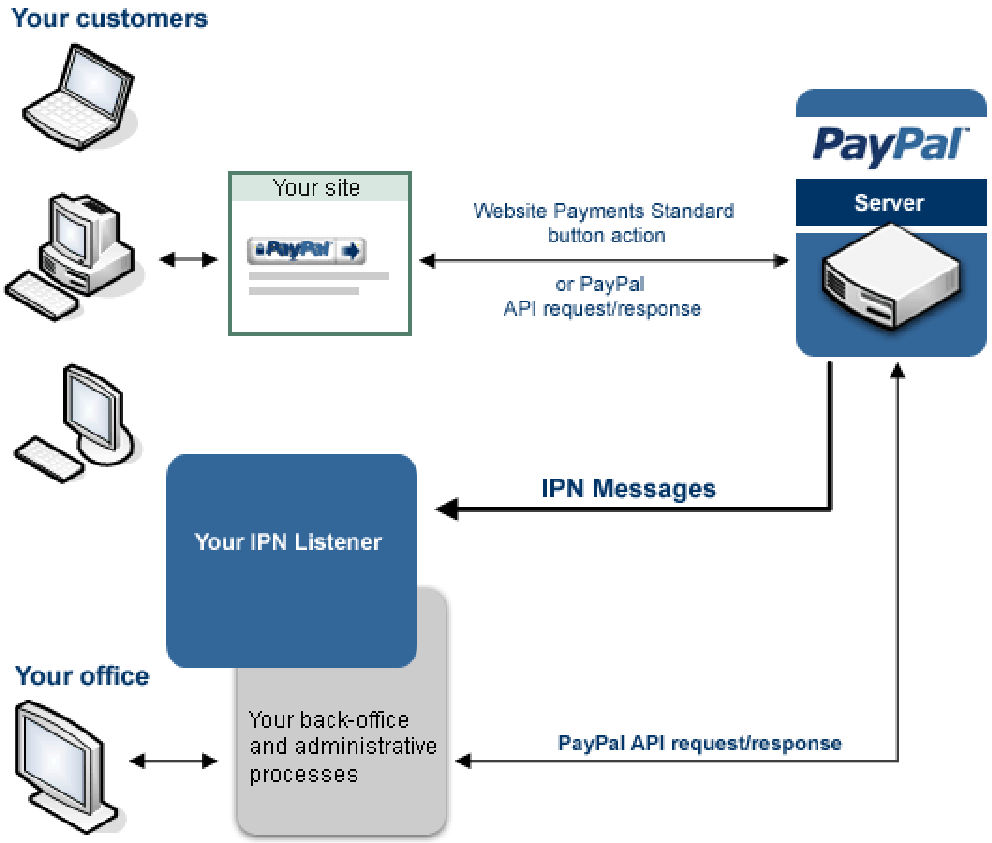
The PayPal API has two interfaces, a name-value pair (NVP) interface and a SOAP interface; EC fully supports both interfaces and is one of the interfaces where the user can choose their own development style and environment.

The Name / value pair (NVP) interface, NVP is a way to specify a name and value in a string. Request and Response HTTPS are passed in a URL-encoded format.

In the SOAP interface, Requests and responses to APIs are passed through the Simple Object Access Protocol (SOAP), which uses the Web Services Description Language (WSDL) and the XML Schema Description Language (XSD).

To use the PayPal API, a set of credentials are needed to show the user's PayPal identity. Palpay provides both the certificate-based and signature-based Authentication method.

Figure 11 shows the basic business logic flow for Paypal.



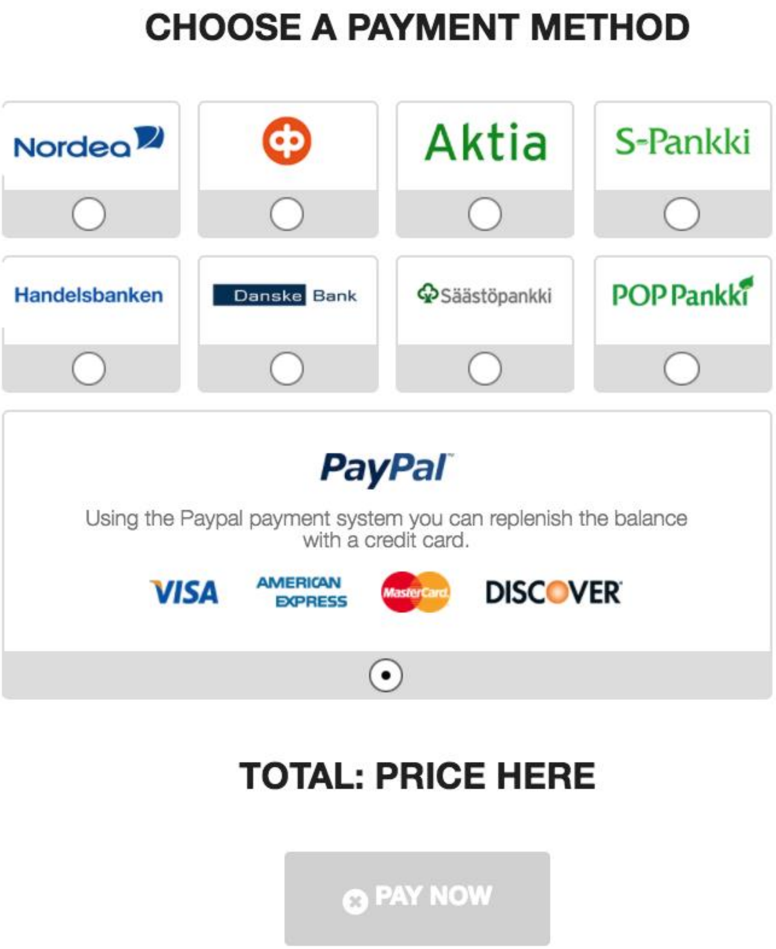
Your customers
Your site
PayPal
Website Payments Standard button action
or PayPal API request/response
PayPal
Server
IPN Messages
Your IPN Listener
Your office
Your back-office and administrative processes
PayPal API request/response

FIGURE 11. Paypal business logic flow diagram (9)

During the integration, with Instant Payment Notifications (IPNs), it is possible to automate certain aspects of the business: Whenever a PayPal payment is received or a transaction occurs.

When changed, IPN sends the transaction details to the server. IPN cannot be used for the payment when it is used in conjunction with fast checkout transactions State, but only to get an unsynchronized notification, such as electronic checks cleared or refund.

Figure 12 below shows the customize Paypal checkout button this project.



Using the Paypal payment system you can replenish the balance with a credit card.

FIGURE 12. Paypal service in the front-end.

3.5.1 Paypal Sandbox - Testing environment

Integrating the Paypal to the site, a build-in powerful testing tool has to be highlighted. It is called the Paypal Sandbox.

PayPal offers a variety of solution-specific development and test environments via the PayPal Sandbox.

The PayPal Sandbox is exact the same test environment as the instant PayPal site, with the difference that there is no real money transaction at all.

With Sandbox, you can test your entire integration before submitting the transaction to the immediate PayPal environment.

Several notable features of the Paypal Sandbox

- All test accounts, e-mail addresses, funding sources (e.g. bank accounts, credit cards, balance) are all fictitious
- Trades are analogous and do not have real money transactions
- The emails sent by PayPal are analogous and appear on the "Test Email" tab in PayPal Sandbox

To access the Paypal Sandbox you need to sign up with a real email address and create a test account to validate your PayPal integration.

In general, you need to create an enterprise test account to represent yourself and create one or more personal test accounts to represent buyers.

3.6 Project management and bidding Module

The next is the most original system, the core functions of the project bidding module. This part mainly includes the project publishing component, bidding component and process management component. in addition, from the big view, the module has paralleled with several other functional modules, such as Account module, user Id is one of other indexer to locate the projects. And Mail notification module, when the project status updated the corresponding mail will send to customer, moreover, payment module is required connected obviously.

Before we actual expand the project module, there is a pre-set need to mention again. User account has two roles we called “Employee” and “Freelancer”, as it explained at early of this thesis, one individual account can have one of them but also can be both of. Why we do repeat this, just want the logic don’t lost.

Obviously, one project has and only has one “Employee” as project publisher. And at the same time whatever his/her project in which status, they can participate bidding other projects as a “Freelancer”.

Base on understanding the concept, we will not get confusing in the project bidding later logic. The key point is only “Employee” can draft and publish project, approve bidding or close, delete own projects in any status. Opposite side, only “Freelancer” can bid projects, accept deal and abandon successful bidding. It will explain more in later detail processing.

3.6.1 Projects publishing

In this chapter, there is a brief introduction for project publishing will be described in more detail feature design combined with an account module.

When the user first time comes to the platform, they might want to publish a project without having an account on the site. It is actually possible. At this moment, the view is pinned on the publish button.

On the platform, the user can edit the project detail without logging in or registering before. Once the user has completed giving the required field, then they just click the “Publish” button. If the user is a guest or a log out user, the project information will be saved as a draft. This draft has a temporary id for identify. Another index field is a publisher Id. At the this stage, it is called the “Recording” project.

The guest will be redirected to a quick login page. Once they have logged in, they will be redirected to publish steps to complete the project publishing. If they are new users, they need one more step, to do the registration. Then the new user will get a confirmation mail with one special link. This link handles activating the account and publishing the draft project. It is called the “One Key” publish projects. This is a session and encryption utilized rationally.

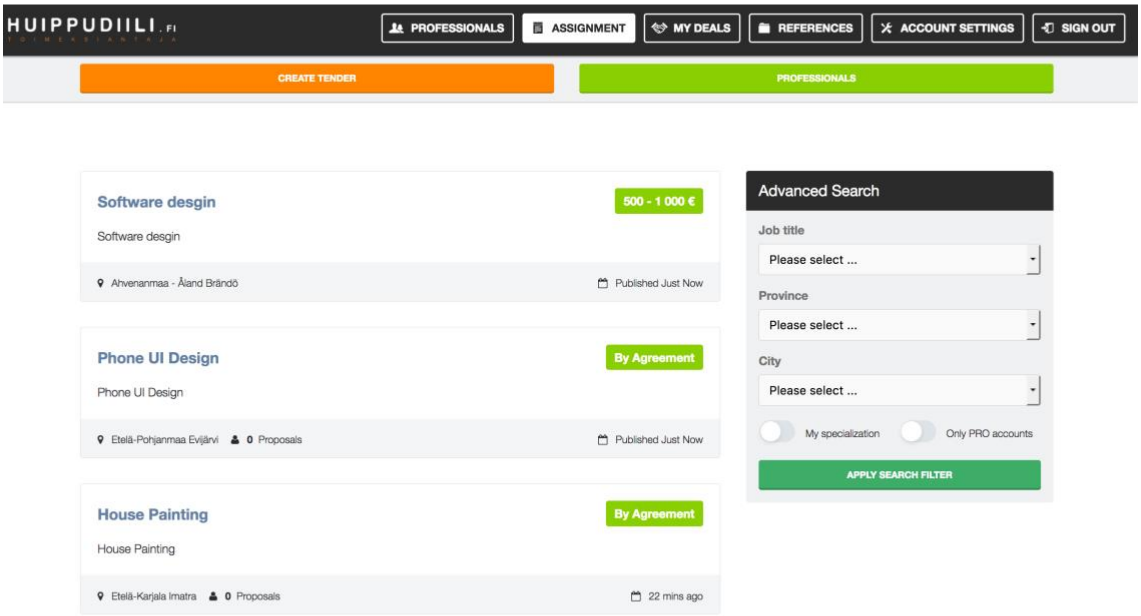
In the project publishing component, a flexible Ajax function calls multi times the entity INSERT, UPDATE to shorten the loading time and to enhance the user experience.

3.6.2 Project listing and searching

Project listing and searching is an open page for all users. It will list all the published project which is ready for bidding and in the bidding stage. From each project card, it displays the project location, the short of bidding amount, and the project offer range or discuss on the way (offer settings during the project publishing), and also project publish time which has been calculated before loading.

To list eligible projects out is required at times for the SQL query optimization, especially, on the Key Words searching. There was earlier a Key Words search function but it was discarded during the developing. Although the SQL query from the title and description was tried to be optimized. It is still not enough to quickly locate or guessing locate the customer's "Key Words". Therefore the "Advance Search" was presented. With the Advance Search, the user can quick locate the projects with either location or type.

This feature provides a good usability, and a user-friendly experience. On the other hand, the categories of project types also have to be improved. The more specialized, detailed list of the types needs to be prepared well in advance. (That is one of the demands of the Admin panel). Through the collection, screening, integration of the new type of list, the project can be more accurately be defined. Below Figure 13 shows the Listing page, every project card has the brief information include the title, the description, the location, the bidding proposals, the calculated published time, and the offer type by the price range or the agreements.



Software desgin
500 - 1 000 €
Software desgin
Ahvenanmaa - Åland Brändö
Published Just Now

Phone UI Design
By Agreement
Phone UI Design
Etelä-Pohjanmaa Evijärvi 0 Proposals
Published Just Now

House Painting
By Agreement
House Painting
Etelä-Karjala Imatra 0 Proposals
22 mins ago

Advanced Search
Job title
Please select ...
Province
Please select ...
City
Please select ...
○ My specialization ○ Only PRO accounts

FIGURE 13. Projects Listing and Searching

3.6.3 Projects bidding

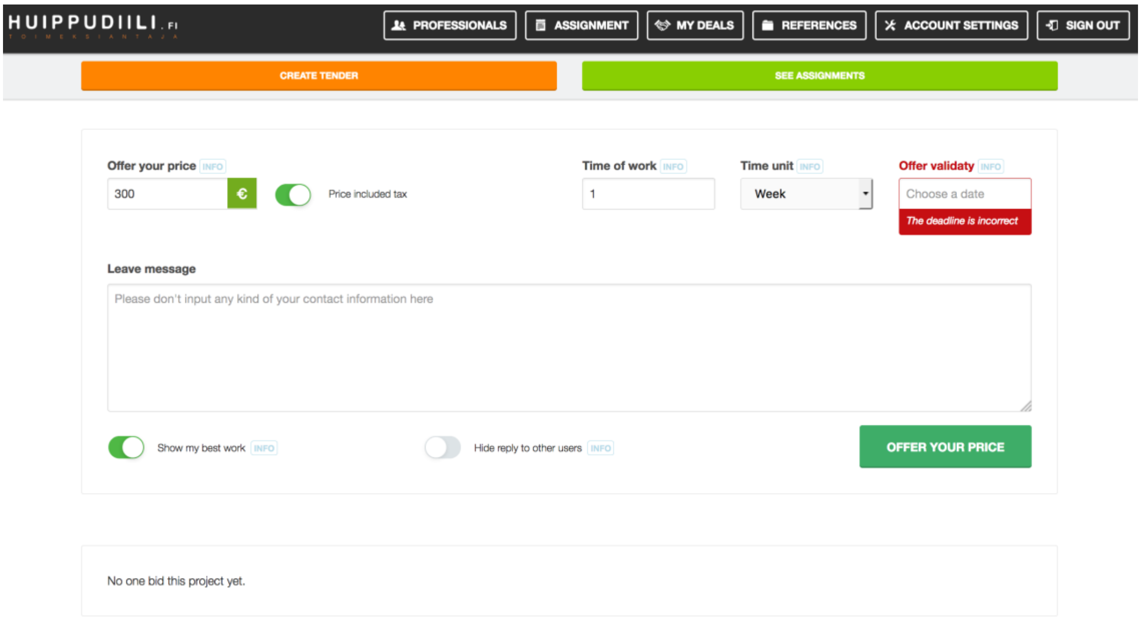
The projects bidding component follows the double election, a win-win principle. It can be divide into four actions. Before listing the actions, the bidding working flow needs to be cleared at first.

1. The “Freelancer” views the Published Project and offers a bidding. It can cancel this offer any time before the project goes to next status. Then it is waiting.
2. The “Employee” views all biddings and decides to approve one. The “Approved” status will notice the Freelancer. Then it is waiting.
3. The “Freelance” accepts the deal or abandons the offer. If the deal is accepted, the deal is made and the project goes to the next stage.

The actions include, participating in the bidding, cancelling or abandoning the offer, approving the bidding or accepting the offer, making the deal or completing the project.

The Figure 14 below shows a bidding form, the features can be easily used. Here are just shortly listed out some main features and techniques used.

- Form validation,
- Datatime picker is a third-party JS lib
- “Show my best work” is related to the freelancer reference. The Freelancer can upload or record the previous projects.
- “Hide to the other user”, is an interesting feature which lets the Freelancer do an anonymous bid. Only the project publisher can see the details. It can control the visibility of all the files in this form.
- “Leave message” provides a simple communicate window. If the “Hide to the other user” is turned on, the message will also be hidden.



Offer your price: 300 €

Price included tax: on

Time of work: 1

Time unit: Week

Offer validaty: Choose a date

The deadline is incorrect

Leave message

Please don't input any kind of your contact information here

Show my best work: on

Hide reply to other users: off

No one bid this project yet.

FIGURE 14. Project bidding slot

3.6.4 Project management

The project management sections have three sets of view based on the user's role and their permissions. That three set of views are, the "Employee", the "Freelancer" and Both.

The Figure 15 below shows a sample of the Employee's view.

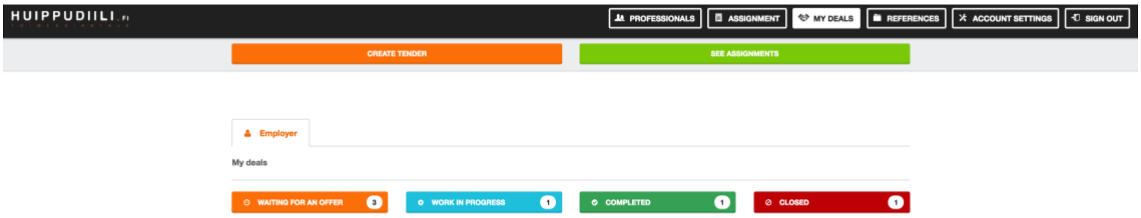


FIGURE 15. Employee Project management

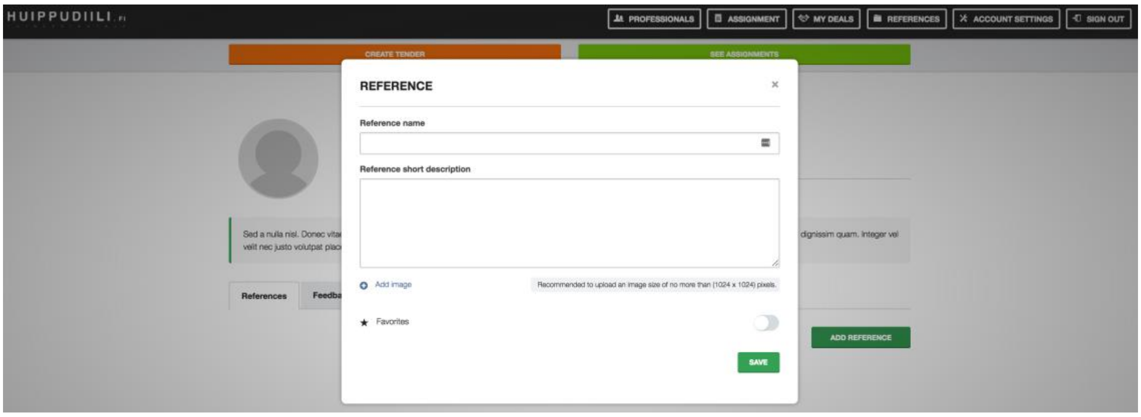
The Employee can easily check the current projects' status. The user-friendly colours are marked at the whole site. There is also the number at the projects status column. The all of columns can be expanded to a list, the list will show the projects card with the action buttons. The action buttons include the status changing actions which it is "Process", "Deal", "Completed" and "Close".

4 THE IMPROVEMENTS

Although the project is kindly implemented, there are still quite a lot of bugs, improvable modules and enhanced features in the backlog list.

4.1 Freelancer Reference

Currently, the “Freelancer” user already has their own “Reference” page. They can upload their best working as they wish. The feature is just an initial implement. Now the user can only upload images and a long text of description. (See Figure 16 below)



Recommended to upload an image size of no more than (1024 x 1024) pixels.

Sed a nulla nisl. Donec vita
velit nec justo volutpat plac

dignissim quam. Integer vel

FIGURE 16. Freelancer Reference

The plan is to make the reference page more valuable. On the home page a slot should be opened for a slider and a project list page should utilize a sider bar for the ”Advertising section”. It still needs from the UI design, the usability testing and the feature implement quite many steps. But if it comes true, this could be a point to attract the user to pay.

4.2 User real-time online status display

The user real-time online status displaying tracks the user’s online status. It is part of the communication module. On the current Job Broker Platform, there are less features for building communication between the users. It could be a part that is always paid less attention to compared to a 24/7 customer support, if a channel can be provided for the user to communicate, it is not only saving the support resource at some point, but it also creates a community-based, spontaneous operation, the promotion of a virtuous circle of users. However, it is a long journey, and the real-time online status display feature is a beginning.

4.3 Published Project filter by location

The published project filter by location has not yet fully implemented. The challenge is to implement a smart enough feature to append the user's info as a reference. If the advance search could automatically filter the user nearest to available projects, it would be very convenient and smart.

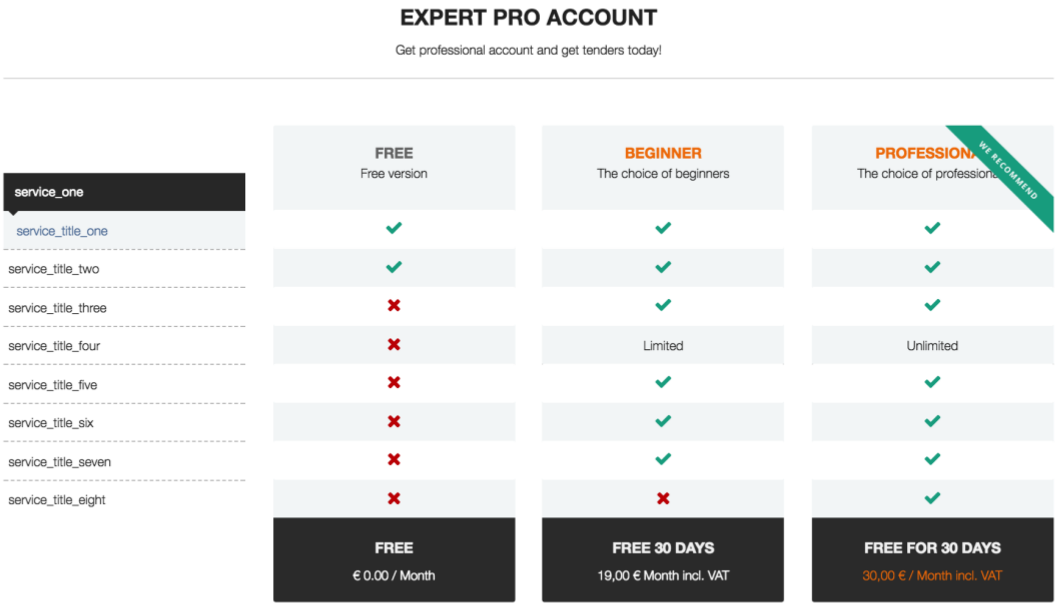
4.4 Monthly service fee and Market exploration

The marketing is always a part of exploring.

How to locate the customer group?

How to define the price list with service?

Below figure 17 is an example price list will basically setup for this project.



EXPERT PRO ACCOUNT

Get professional account and get tenders today!

service_one	FREE Free version	BEGINNER The choice of beginners	PROFESSION The choice of profession WE RECOMMEND
service_title_one	✓	✓	✓
service_title_two	✓	✓	✓
service_title_three	✗	✓	✓
service_title_four	✗	Limited	Unlimited
service_title_five	✗	✓	✓
service_title_six	✗	✓	✓
service_title_seven	✗	✓	✓
service_title_eight	✗	✗	✓
	FREE € 0.00 / Month	FREE 30 DAYS 19,00 € Month incl. VAT	FREE FOR 30 DAYS 30,00 € / Month incl. VAT

FIGURE 17. Draft of Service fee

There were quite a lot of thinking related marketing from the beginning. The service fee also experienced several times huge modification. Till now there are few points summarized.

Do more in-depth and detailed market research, expect to pass the survey can roughly divide the service population of this project. Base on the result we can do more customise improvements.

Regarding to the detail price list, it involved in some trade secrets or it can call different marketing plan and calculation. In this thesis no more in-depth discussion.

5 CONCLUSION

Job broker platform is into bug fixing and deployment stage, and it obtain the initial starting investment. Can not guarantee that this is a hundred percent successful project and not sure how far it could go, the first release will come soon.

The purpose of this project is to provide freelance with more efficient project searching and trading platforms, to provide employers with a wider distribution channels, cost savings and increased economic conversion rates. If the platform itself can also provide economic benefits, it will be a nice win-win.

From this project and several ten thousand code, could be say like that, I grew up with this project. Look backward from now, read again the code, the feature structures, and even SQL queries, they may have hundreds of loopholes, it is a Treasure for me.

REFERENCES

1. CareerFoundry 25.4.2017. The Difference Between UX and UI Design – A Layman's Guide. Cited 18.12.2017 https://careerfoundry.com/en/blog/ux-design/the-difference-between-ux-and-ui-design-a-laymans-guide/

2. Codeigniter 2017. Codeigniter. Cited 8.10.2017 https://codeigniter.com/

3. CodeUp 2017. What is a Full-Stack Developer. Cited 8.10.2017 http://codeup.com/what-is-a-full-stack-developer/

4. Ntu Edu 2010. A Quick-Start Tutorial on a Relational Database Design. Cited 10.10.2017 https://www.ntu.edu.sg/home/ehchua/programming/sql/Relational_Database_Design.html

5. Facebook for Developers 2017. Facebook Login for Apps—Overview. Cited 20.10.2017. https://developers.facebook.com/docs/facebook-login/overview

6. PHP-FIG 2017. PSR-2: Coding Style Guide. Cited 9.10.2017 http://www.php-fig.org/psr/psr-2/

7. ThinkSaaS 2017. PHP 框架 Codeigniter 的六大优势与五大劣势. Cited 10.10.2017 https://www.thinksaas.cn/group/topic/120250/

8. Paypal 2017. What is Paypal. Cited 11.11.2017. https://www.paypal-biz.com/college/buy/registered/paypalintro/3947.html

9. Usama Dar's Blog 2017. PayPal Integration Primer, Quick way to integrate online payments to your website. Cited 11.11.2017 http://usamadar.com/2012/01/24/paypal-integration-primer-quick-way-to-integrate-online-payments-to-your-website/

APPENDICES

APPENDICES

APPENDIX 1

INITIATION DOCUMENT OF A BACHELOR'S THESIS

Author ____

Customer ____

Customer’s contact person and information ___________________________________

____

Title ____

Description ____

____

____

____

Objectives ____

____

____

Target schedule ____

____

____

____

____

Date and signatures ____

____

____

```
function socialogin(type, statu) {
    if (type == "googleplus") {
        if (statu == "disconnect") {
            $("#social_disconnect_dialog").dialog({
                modal : true,
                title : text("Disconnect Confirmation"),
                buttons : [{
                    text : text("Yes"),
                    click : function() {
                        var obj = $(this);
                        console.log(gapi.auth.getToken().access_token);
                        var xhr = new XMLHttpRequest();
                        xhr.open('GET',
'https://accounts.google.com/o/oauth2/revoke?token='+gapi.auth.getToken().access_token);
                        xhr.send();
                        $.ajax({
                            url : site_url + "login/socialDisconnect",
                            type : "post",
                            data : { type : "googleplus"},
                            success : function(data) {
                                $(obj).dialog("close");
                                window.location.replace(site_url + "user/settings");
                            }
                        });
                    }
                }, {
                    text : text("No"),
                    click : function() {
                        $(this).dialog("close");
                    }
                }],
                open : function() {
                    $(this).dialog().html(text("Are you sure to disconnect Google+ from
your account"));
                },
                close : function() {
                    $(this).dialog("destroy");
                }
            });
        } else {
            gapi.client.load('plus', 'v1', function() {
                var request = gapi.client.plus.people.get({'userId' : 'me'});
                request.execute(function(res) {
                    res.socialtype = "googleplus";
                    console.log(res);
                    console.log(res.socialtype);
                    if (statu == 'register') {
                        //get user email
                        //user's info
                        if (res.emails) {
                            $(".email_input").val(res.emails[0].value);
                        }
                        if (res.displayName) {
                            $(".nickname_input").val(res.displayName);
                        }
                        $.ajax({
                            url : site_url + "login/gpDataRecord",
                            type : "post",
                            data : res,
                            success : function(data) {
                            }
                        });
                    } else if (statu == "connect") {
                        console.log("connect");
                        console.log(res);
                        $.ajax({
                            url : site_url + "login/socialConnect",
                            type : "post",
                            data : res,
                            success : function(data) {
                                if (data == "ok") {
                                    window.location.reload();
                                } else {
                                    global_warning(data);
                                }
                            }
                        });
                    } else if (statu == "login") {
                        $.ajax({
                            url : site_url + "login/socialLoginCheck",
                            type : "post",
                            data : res,
                            success : function(data) {
                                var obj = json_decode(data);
                                console.log(obj.confirm_user);
                                if($.isNumeric(obj.confirm_user)){
                                    //var obj = json_decode(data);

window.location.replace(site_url+"project/publish_project/confirmP?u="+obj.con
firm_user+"&d="+obj.pid);
                                }
                                if(data == "ok"){
                                    window.location.reload();
                                } else {
                                    $("#login_error_alert").html(data);
                                    $("#login_error_alert").fadeIn("fast");
                                }
                            }
                        });
                    }
                });
            });
        }
    }
    if (type == "facebook") {
        if (statu == "disconnect") {
            $("#social_disconnect_dialog").dialog({
                modal : true,
                title : text("Disconnect Confirmation"),
                buttons : [{
                    text : text("Yes"),
                    click : function() {
                        var obj = $(this);
                        $.ajax({
                            url : site_url + "login/socialDisconnect",
                            type : "post",
                            data : { type : "facebook" },
                            success : function(data) {
                                window.location.replace(site_url + "user/settings");
                            }
                        });
                    }
                }, {
                    text : text("No"),
                    click : function() {
                        $(this).dialog("close");
                    }
                }],
                open : function() {
                    $(this).dialog().html(text("Are you sure to disconnect Facebook
from your account"));
                },
                close : function() {
                    $(this).dialog("destroy");
                }
            });
        } else {
            FB.login(function(response) {
                FB.api('/me', function(response) {
                    response.socialtype = "facebook";
                    //console.log(response);
                    if (statu == "register") {
                        if (response) {
                            if (response.email) {
                                $(".email_input").val(response.email);
                            }
                            if (response.name) {
                                $(".nickname_input").val(response.name);
                            }
                        }
                        $.ajax({
                            url : site_url + "login/fbDataRecord",
                            type : "post",
                            data : response,
                            success : function(data) {
                            }
                        });
                    } else if (statu == "connect") {
                        $.ajax({
                            url : site_url + "login/socialConnect",
                            type : "post",
                            data : response,
                            success : function(data) {
                                if (data == "ok") {
                                   window.location.reload();
                                } else {
                                    global_warning(data);
                                }
                            }
                        });
                    } else if (statu == "login") {
                        $.ajax({
                            url : site_url + "login/socialLoginCheck",
                            type : "post",
                            data : response,
                            success : function(data) {
                                var obj = json_decode(data);
                                console.log(obj.confirm_user);
                                if($.isNumeric(obj.confirm_user)){

window.location.replace(site_url+"project/publish_project/confirmP?u="+obj.con
firm_user+"&d="+obj.pid);
                                }
                                if(data == "ok"){
                                    window.location.reload();
                                } else {
                                    $("#login_error_alert").html(data);
                                    $("#login_error_alert").fadeIn("fast");
                                }
                            }
                        });
                    }

                });
            }, {
                scope : 'email'
            });
        }
    }
```

```
}
```