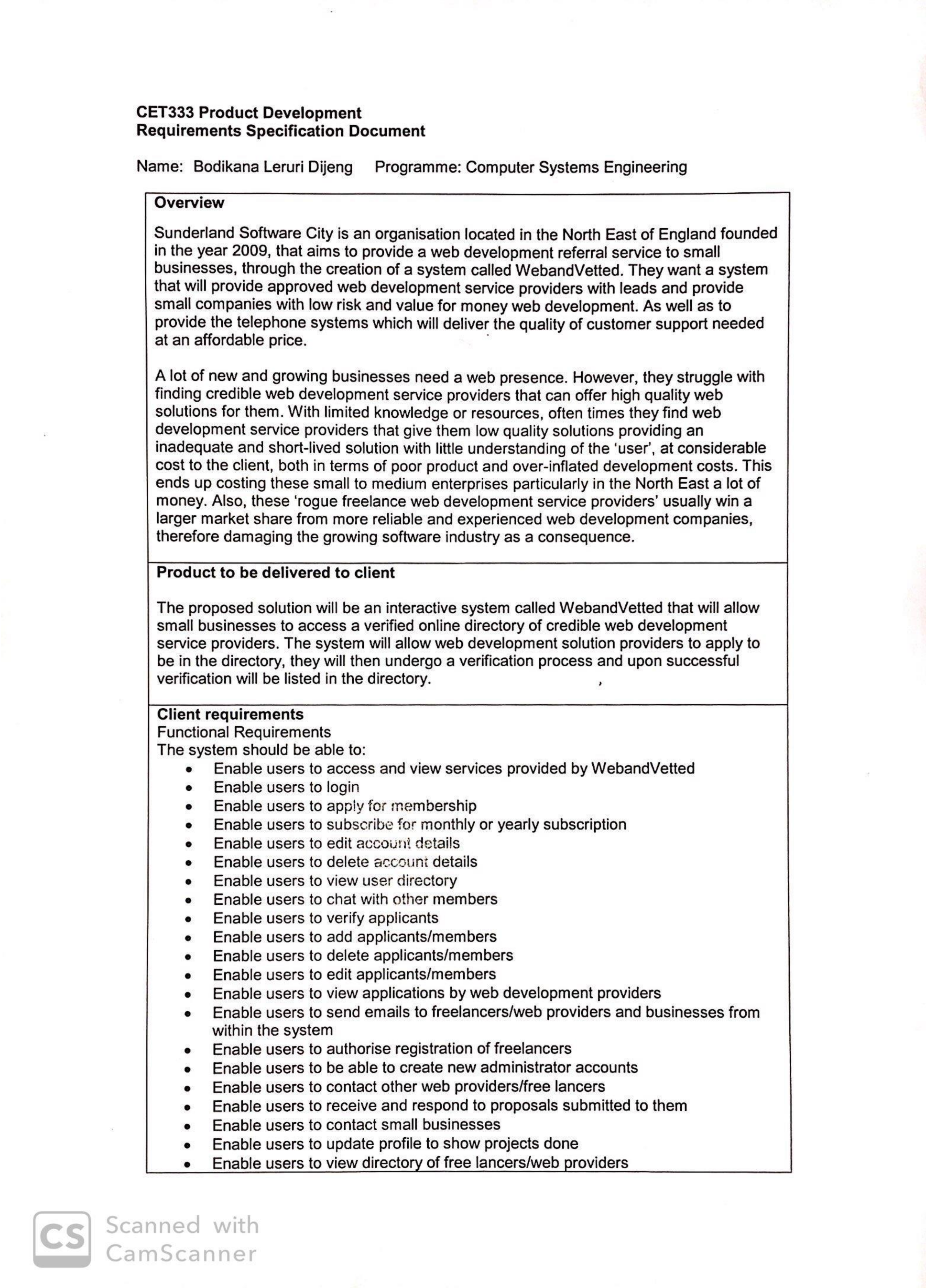
WEB AND VETTED DOCUMENTATION

Bodikana Dijeng
Registration number:
189100139/1

Center: University of Sunderland
Course: Computer Systems Engineering

Contents

Requirements Specification



CET333 Product Development
Requirements Specification Document

Name: Bodikana Leruri Dijeng Programme: Computer Systems Engineering

Overview

Sunderland Software City is an organisation located in the North East of England founded in the year 2009, that aims to provide a web development referral service to small businesses, through the creation of a system called WebandVetted. They want a system that will provide approved web development service providers with leads and provide small companies with low risk and value for money web development. As well as to provide the telephone systems which will deliver the quality of customer support needed at an affordable price.

A lot of new and growing businesses need a web presence. However, they struggle with finding credible web development service providers that can offer high quality web solutions for them. With limited knowledge or resources, often times they find web development service providers that give them low quality solutions providing an inadequate and short-lived solution with little understanding of the 'user', at considerable cost to the client, both in terms of poor product and over-inflated development costs. This ends up costing these small to medium enterprises particularly in the North East a lot of money. Also, these 'rogue freelance web development service providers' usually win a larger market share from more reliable and experienced web development companies, therefore damaging the growing software industry as a consequence.

Product to be delivered to client

The proposed solution will be an interactive system called WebandVetted that will allow small businesses to access a verified online directory of credible web development service providers. The system will allow web development solution providers to apply to be in the directory, they will then undergo a verification process and upon successful verification will be listed in the directory.

Client requirements
Functional Requirements
The system should be able to:

- Enable users to access and view services provided by WebandVetted
- Enable users to login
- Enable users to apply for membership
- Enable users to subscribe for monthly or yearly subscription
- Enable users to edit account details
- Enable users to delete account details
- Enable users to view user directory
- Enable users to chat with other members
- Enable users to verify applicants
- Enable users to add applicants/members
- Enable users to delete applicants/members
- Enable users to edit applicants/members
- Enable users to view applications by web development providers
- Enable users to send emails to freelancers/web providers and businesses from within the system
- Enable users to authorise registration of freelancers
- Enable users to be able to create new administrator accounts
- Enable users to contact other web providers/free lancers
- Enable users to receive and respond to proposals submitted to them
- Enable users to contact small businesses
- Enable users to update profile to show projects done
- Enable users to view directory of free lancers/web providers

- Enable users to update project details before submission deadline

Non-Functional Requirements:

- ***Aesthetically Appealing:*** a vibrant design with clear dialogues to capture attention of the audience
- ***Mobile-Friendly:*** similar look-and-feel on devices with a web browser.
- ***Secure***: sensitive information can be accessed by authorized users only without any security breaches
- ***Visible:*** constantly give feedback to users about freelance jobs of their interest, plus provide feedback about it
- ***Reliable:*** when errors are encountered, users can be instructed to recognise, diagnose and fix the error
- ***Maintainable:*** retrieval and manipulation of data should be updated to all users within 2 seconds

Constraints

- Internet access is not reliable, which may hinder project schedule.
- Lack of necessary skills in php programming may hinder project success.

Resources

Minimum Hardware Requirements:

- 2 Gb RAM
- Quad Core Processor 1.6 GHz
- 500 Gb hard drive

Development Tools:

- Netbeans IDE
- Oracle Database
- NotePad ++

Programming:

- PHP version 5.4.19
- HTML 5, CSS 3, Javascript
- AngularJS 5

Evaluation

The pilot system will be tested using various test plans. These will include functional testing, usability testing, interface testing, compatibility testing, performance testing and security testing. Functional testing will check all the links in web pages, database connection, forms used for submitting or getting information from the user in the web pages. Usability testing will be for testing the human-computer interaction characteristics of the system and any weaknesses are identified for correction. Interface testing will be where the server-side interface will be tested. This is done by verifying that communication is done properly. Compatibility of the server with software, hardware, network, and the database will be tested. Compatibility will test the system's compatibility with different devices, operating systems and browsers. Performance testing will test if the web application can sustain heavy load. This testing will include Web Load Testing as well as Web Stress Testing.

The last kind of testing that will be done is Security Testing, this will include tests for Password cracking, Network Scanning and Virus Detection.
The aim of these tests will be to check all aspects of the web application to ensure high standard product will be delivered to the client.

Client Sign-off

Signature (this indicates acceptance of the scope of the practical component of the project) **Date** 01/03/19

PLANNING DOCUMENT (SCHEDULE)

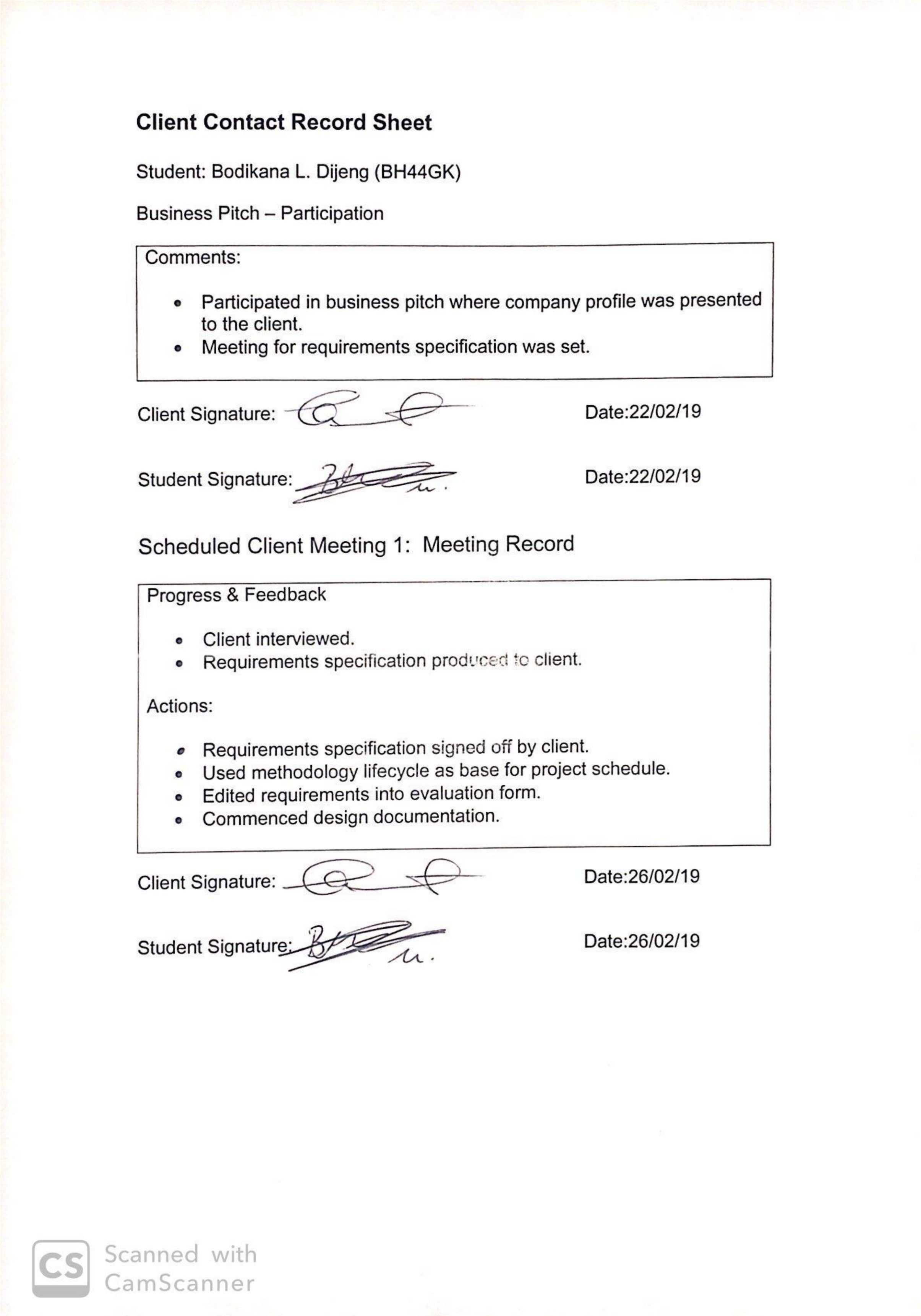
Task	Hours	Estimated Start-Date	Estimated End Date	Actual Start Date	Actual End Date	Deliverable
1. DEVELOP OVERALL MODEL	**18**					
1.1 Define Problem	5	04/02/2019	06/02/2019	04/02/2019	05/02/2019	Objective and scope of project
1.2 Interview client	1	07/02/2019	07/02/2019	06/02/2019	06/02/2019	PowerPoint presentation
1.3 Analysis of requirements	5	07/02/2019	10/02/2019	07/02/2019	15/02/2019	Notes
1.4 Determine methodology to use	6	11/02/2019	13/02/2018	18/02/2019	26/02/2019	Requirement Specification Document
1.5 Client meeting and requirements specification signoff	1	14/02/2019	14/02/2018	28/03/2018	01/03/2018	Signed Requirement Specification Document
2. **BUILD FEATURES LIST**	**21**					
2.1 Administrator	4	15/02/2019	15/02/2019	18/02/2019	18/02/2019	Complete admin feature list
2.2 Regular User	1	18/02/2019	18/02/2019	19/02/2019	19/02/2019	Complete regular user feature list
2.3 Identifying freelancers	3	18/02/2019	18/02/2019	20/02/2019	20/02/2019	Complete freelancer feature list

2.4 Identifying companies	3	18/02/2019	20/02/2019	20/02/2019	20/02/2019	Complete company feature list
2.5 Identifying important features	3	22/02/2019	22/02/2019	25/02/2019	25/02/2019	Complete important feature list
2.6 Modelling of database	4	25/02/2019	27/02/2019	25/02/2019	26/02/2019	
2.7 Model site architecture	3	27/02/2019	27/02/2019	26/02/2019	27/02/2019	
3 PLAN BY FEATURE	**34**					
3.1 Administrative user	2	01/03/2019	02/03/2019	01/03/2019	02/03/2019	
3.2 Regular user	2	02/03/2019	02/03/2019	01/03/2019	02/03/2019	Project schedule
3.3 Freelancer user	2	05/03/2019	06/03/2019	02/03/2019	06/03/2019	Project schedule
3.4 Company user	2	05/03/2019	07/03/2019	05/03/2019	06/03/2019	Project schedule
3.5 Important features	2	02/03/2019	08/03/2019	06/03/2019	07/03/2019	Project schedule
3.6 Database Model	5	07/03/2019	09/03/2019	06/03/2019	09/03/2019	Project schedule
3.7 Site Architecture	6	06/03/2019	09/03/2019	10/03/2019	12/03/2019	Project schedule
3.8 System Testing and Evaluation	8	09/03/2019	13/03/2019	15/03/2019	16/03/2019	Project schedule
3.9 Documentation of plan	4	02/03/2019	15/03/2019	19/03/2019	19/03/2019	Project schedule
4 DESIGN BY FEATURE	**26**					
4.1 Administrative user	3	15/03/2019	15/03/2019	20/03/2019	20/03/2019	Activity diagram, use case diagram, & uml diagram for administrator

4.2 Freelance user	3	14/03/2019	14/03/2019	16/03/2019	16/03/2019	Activity diagram, use case diagram, & uml diagram for freelancer
4.3 Company user	3	15/03/2019	17/03/2019	16/03/2019	16/03/2019	Activity diagram, use case diagram, & uml diagram for company
4.4 Regular user	2	15/03/2019	16/03/2019	17/03/2019	17/03/2019	Activity diagram, use case diagram, & uml diagram for user
4.5 Website architecture design	4	13/03/2019	18/03/2019	15/03/2019	20/03/2019	Blueprint for sitemap
4.6 Database ERD model design	6	17/03/2019	20/03/2019	17/03/2019	22/03/2019	Evaluation sheet
4.7 System testing and database evaluation of functional and non-functional requirements	5	26/03/2019	30/03/2019	23/03/2019	29/03/2019	Client Notes
5 BUILD BY FEATURE	**61**					
5.1 Administrator	8	02/04/2019	09/04/2019	30/02/2019	10/04/2019	Admin fully functional user interface
5.2 Freelancer	6	03/04/2019	10/04/2019	04/04/2019	12/04/2019	Freelancer fully functional user interface
5.3 Company	7	02/04/2019	13/04/2019	08/04/2019	15/04/2019	Company fully functional user interface
5.4 Regular user	5	13/04/2019	18/04/2019	15/04/2019	19/04/2019	Regular fully functional user interface
5.5 Detailed sitemap	4	23/04/2019	26/04/2019	27/04/2019	27/04/2019	Fully complete detailed sitemap
5.6 Website database	6	27/04/2019	02/05/2019	30/04/2019	03/04/2019	Functional website database
5.7 System code	14	30/04/2019	09/05/2019	30/04/2019	14/05/2019	Fully functional website

5.8 Client Meeting	2	10/05/2019	10/05/2019	10/04/2019	10/04/2019	System design diagrams
5.9 System requirements evaluation	4	11/05/2019	15/05/2019	17/05/2019	17/05/2019	Complete testing requirements and evaluation document
5.10 Documentation and final submission	5	14/05/2019	18/05/2019	25/05/2019	25/05/2019	Documentation of product
TOTAL HOURS	**160**					

CLIENT CONTACT RECORD SHEET AND SIGN OFF



Client Contact Record Sheet

Student: Bodikana L. Dijeng (BH44GK)

Business Pitch – Participation

Comments:

- Participated in business pitch where company profile was presented to the client.
- Meeting for requirements specification was set.

Client Signature: Date:22/02/19

Student Signature: Date:22/02/19

Scheduled Client Meeting 1: Meeting Record

Progress & Feedback

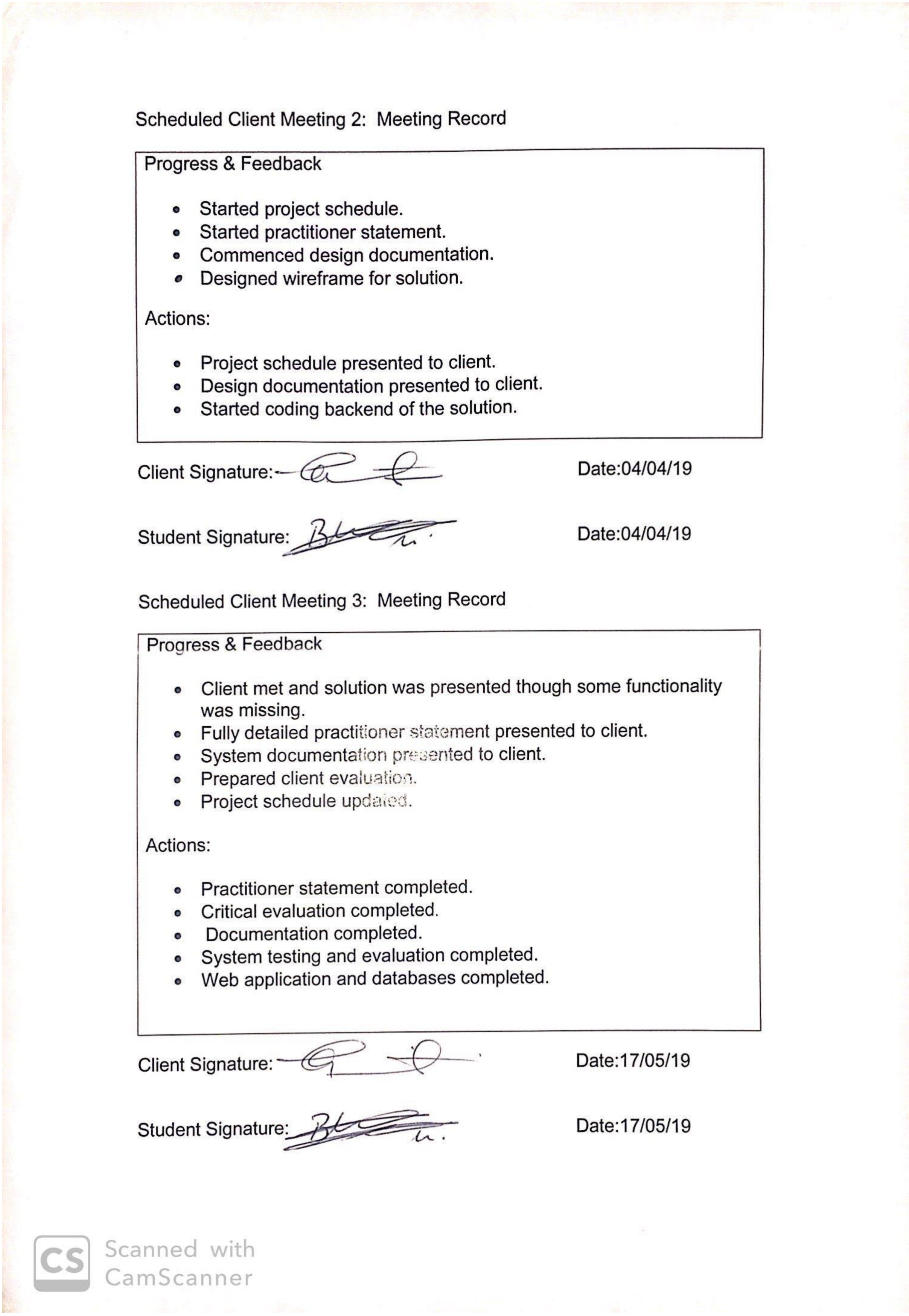
- Client interviewed.
- Requirements specification produced to client.

Actions:

- Requirements specification signed off by client.
- Used methodology lifecycle as base for project schedule.
- Edited requirements into evaluation form.
- Commenced design documentation.

Client Signature: Date:26/02/19

Student Signature: Date:26/02/19



Scheduled Client Meeting 2: Meeting Record

Progress & Feedback

- Started project schedule.
- Started practitioner statement.
- Commenced design documentation.
- Designed wireframe for solution.

Actions:

- Project schedule presented to client.
- Design documentation presented to client.
- Started coding backend of the solution.

Client Signature: Date:04/04/19

Student Signature: Date:04/04/19

Scheduled Client Meeting 3: Meeting Record

Progress & Feedback

- Client met and solution was presented though some functionality was missing.
- Fully detailed practitioner statement presented to client.
- System documentation presented to client.
- Prepared client evaluation.
- Project schedule updated.

Actions:

- Practitioner statement completed.
- Critical evaluation completed.
- Documentation completed.
- System testing and evaluation completed.
- Web application and databases completed.

Client Signature: Date:17/05/19

Student Signature: Date:17/05/19

Practitioner Statement

Introduction

This segment of the portfolio comprises of systems, techniques, instruments and methodologies of the implemented in the Web and Vetted development life cycle. Right off the bat I chose to find out about the Extreme programming system and the Test-Driven Methodology which were different to me since I have never utilized them for any task advancement. The discoveries I made for both of them demonstrated that they are not perfect to carry out on the Web and Vetted project. Since both of them did not meet the client's wants, I chose not to proceed with any of them. Utilizing my past experience, I gained from the System Development module and Project management module, I decided to do this project utilizing the Agile approach since it aims to hand over the project in complete and with great quality well in time when contrasted with other two choices referenced previously.

Methodology

Agile Methodology is perfect for this venture since it points on conveying the undertaking in a concurred time and in great quality and due to that I use the methodology. I have discovered that the real test in software development is producing a complete web application in time and in great quality. For this a great technique that is centred around planning, evaluation, deploying and testing is essential. With this in mind, the agile approach demonstrated to be proficient in my Project Management module. With this in mind it was an insightful move to utilize it to deal with the Web and Vetted site development.

The Agile methodology stresses on isolating tasks into smaller segments and a solution is given alongside the corresponding project stage until full completion of the project. Before employing this method, the percentage completion of weekly tasks was low. Utilizing the agile method, I isolated tasks into smaller sub tasks that I had the option to oversee effectively which greatly increased my productivity. Looking at the results convinced me of the impressiveness of this methodology. I then decided to utilize the methodology for completing the whole Web and Vetted project.

Development Method

Since the agile method is the principle approach I utilized, I needed to utilize a reasonable strategy that uses or falls under the agile methodology. The approach stresses more on accomplishing the task in time and in great quality. Basing on the past experience I gained from System Development module, I needed to pick on utilizing Feature Driven improvement (FDD) strategy. The following is a point by point report of how the Feature Driven advancement strategy works and reasons why I utilized it.

Feature Driven Development

Feature Driven Development (FDD) is a gradual and iterative procedure of building up a task or it tends to be characterized as a nimble technique or a light weight for web development. The Feature Driven Development mixes a ton of perceived strategic policies of businesses into a strong entirety. Every one of those practices are constrained from a customer esteemed usefulness perspective. The fundamental reason for this philosophy is to convey a working programming over and over and of high calibre in a concurred time. Feature Driven Development (FDD) is a model that comprises of five essential exercises which are model-driven and short cycle forms executed as pursues:

Develop Overall Model

The FDD model keeps up that groups utilize the appropriate measure of exertion toward the start of the task in order to build up an item model featuring the issue of the space. Feature driven development developing is shared and time boxed. Production of the domain model must be done in detail by small units and after that assessed by companions during demonstration.

Build a feature list:

Learning attained from the development of modelling, a record of highlights are then settled by breaking domain into fields of subjects that has business activity data. All means utilized for each business activity demonstrates highlights in a sorted directory. Features will at that point be communicated as: "activity, results, and article". It is normal that they won't surpass over a period of two weeks to be completed, however in the event that they surpass they have to be broken into smaller segments.

Plan by feature

When the component list has been demonstrated, the accompanying procedure includes assigning many feature sets to the software engineers. This is created as the programmer highlights bearing in mind their practicality, database model, structure, behaviour and design.

Design by feature

Software engineers at that point make a plan bundle for every single component with the assistance of the central developer picking a gathering of highlights that must be built up inside a period of two-weeks. The central software engineer will likewise make full graphs for every single element while the model is being refined. After composition of this, investigation of the plan is completed and prefaces are delivered.

Build by feature:

After assessment of the structure is finished, designers need to design an action for each element and build up the code for their separate classes. At the point when the investigation is done and a unit test is done, the element is then passed to the primary build.

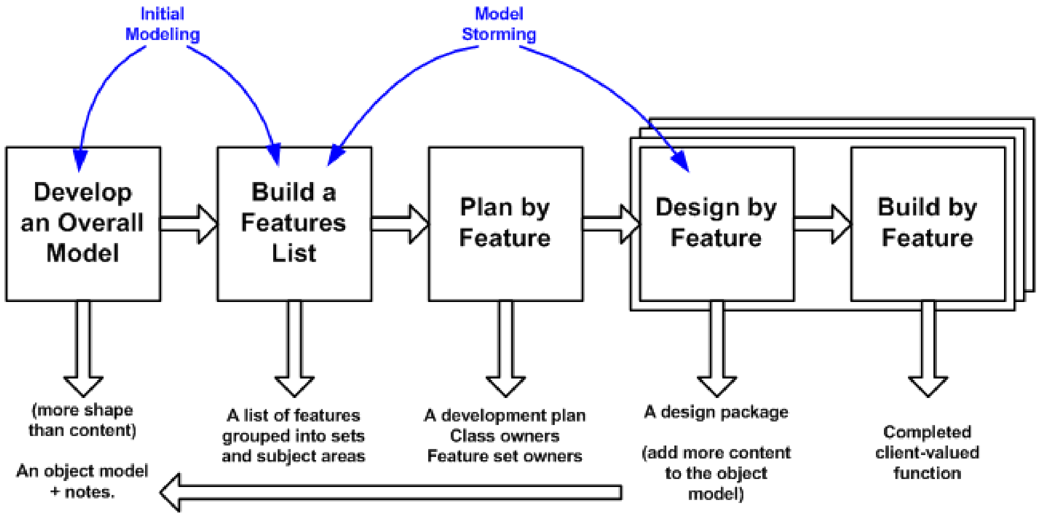

Initial Modeling
Model Storming
Develop an Overall Model
Build a Features List
Plan by Feature
Design by Feature
Build by Feature
(more shape than content)
An object model + notes.
A list of features grouped into sets and subject areas
A development plan Class owners Feature set owners
A design package
(add more content to the object model)
Completed client-valued function

Figure 1: The FDD project lifestyle

The Feature Driven Development lifecycle has methods that are client oriented and utilizes the following practices:

- **Domain Object Modelling:** Investigates and depicts the risky circumstances that must be sold and make a system that can be added to feature list.
- **Configuration management:** Chooses source code for each component and maintains a background marked by the modifications that are done as such that the element groups can improve every one of them.
- **Individual class ownership:** This is where a gathering of code is dispensed to every person. Each person who is allotted code will at that point be responsible for the exhibition, consistency and theoretical trustworthiness of each class.
- **Regular builds:** Are done routinely to guarantee the frameworks cutting-edge and it is exhibited to the customer, this shows mistakes which incorporated on the source code for highlights well in time.

Justification of chosen methodology

Feature Driven Development (FDD) was picked as a perfect strategy for the development of the Web and Vetted web application to deliver inside the approved time span and in high quality. This was in the wake of leading a basic assessment for the Test-Driven Development (TDD), Feature Driven Development (FDD) and the Extreme Programming. Feature Driven Development demonstrates to be a viable methodology when contrasted with Extreme Programming and test-driven methods. This is demonstrated beneath the improvement strategy segment by explaining that it prompts an undertaking being increasingly reasonable since tasks are isolated into little tasks which are progressively sensible and it is more customer centred hence it makes simpler to please the customer. Feature Driven Development guarantees that each achievement of the venture had a deliverable outcome. It likewise ensures that in case after a customer review and there are some changes, it permits that. Feature Driven Development tracks proficient arranging by utilizing project schedule to approach each element.

Feature Driven Development (FDD) is a methodology that is customer based, the customer and the designer cooperate as well as the programmer so as to get the opportunity to comprehend both the functional and non-functional requirements of the application, consequently this will assist me with interacting with the customer more often than not prompting high efficiency. As the engineer I would almost certainly guide the customer where it fits for the application through methods for meetings. The customer will at that point give me a close down as a way of giving the engineer a thumbs up to begin the venture. For each stage and meeting, the customer needs to sign off as this this guarantees I keep to the requirements stipulated by the customer.

Tools and Techniques

In the Web and Vetted project, the following tools and techniques were employed:

Unified Modelling Language (UML)

For the improvement of the Web and Vetted web application the application must be structured and displayed before it could any coding should be possible. Modelling an application is equivalent to giving it its outline of how it’s going to operate and appear. This all alone yields a top-notch application and on time conveyance of the application. UML is an apparatus that will be utilized to display the utilitarian, auxiliary model, social model and the database structural model. Sequence diagrams are known as the behavioural model, USE CASE outlines are named the utilitarian models and ultimately the database engineering model has both the activity diagrams and entity diagrams.

Structural Model

It virtualizes objects structure and their individual procedures, attributes, groups and how they are connected as to the Web and Vetted web application.

Functional Model

The functional model visualizes activities of the Web and Vetted web application however not expressly indicating how those activities were executed. It focuses predominantly on the prerequisites which are useful illustration use case outlines and how the framework is empowering the clients to communicate with one another.

Use Cases

The functional model of the application is visualized by utilizing USE CASES. Use Case model the prerequisites of the Web and Vetted web application by showing how actors in the application connect with the application. USE CASES models help the developer on the most proficient method to build up the application. Class diagrams can be created with the assistance of them.

Entity Relationship Diagram

The graphical portrayal of data of the Web and Vetted web application is made by utilizing entity relationship diagrams by envisioning how the objects relate, freelancer profiles, company profiles and activities inside the framework. The entity relationship diagram is part of the database structural model subsequently showing how data will be stored. Entity Relationship diagrams utilize the CRUD capacities which are (Create, Update, Read and Delete) that work on the database.

Sequence Diagrams

Sequence diagrams show how objects communicate with each from a given situation by utilization of messages.

System Documentation

Introduction

This is a segment that virtualizes every one of the plans of the Web and Vetted in a way that is realistic. This includes dynamic site page being prepared by the web server, the design perspective on the application including the XAMPP architectural model which is perfect for sites which are dynamic during development. Proposed default online visit expected will be virtualized by utilizing a webpage map, Wire frame will be appeared to virtualize how pages of the Web and Vetted interface link with one another. This area further shows UML graphs being sequence diagrams, flow charts, USE cases, stream diagram, entity relationship graph, etc. Below is a diagram that shows how information is being handled on the web server utilizing PHP for the Web and Vetted web application.

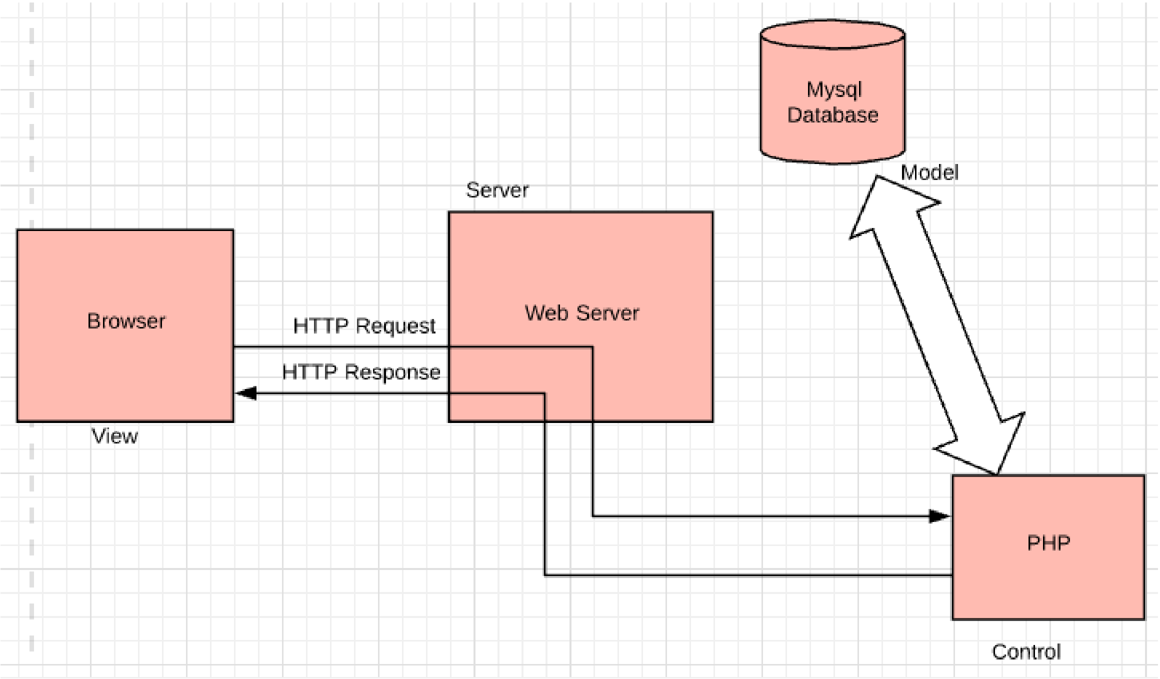


Figure 2Design Architecture with MVC Model

Web and Vetted Web Application Site Map

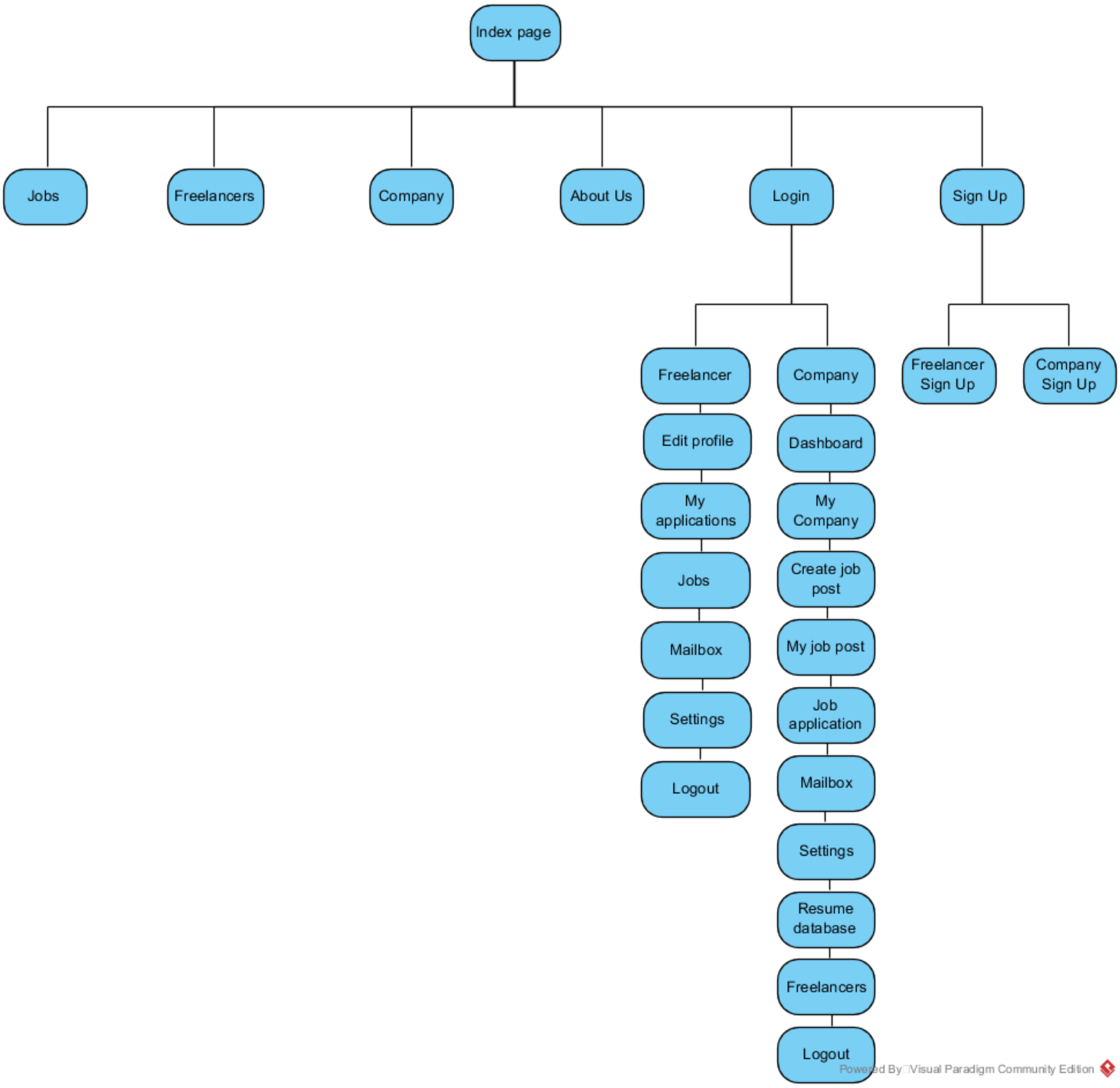


Figure 3: Web and Vetted Site Map

The above diagram shows the sitemap of the Web and Vetted web application. It is a visual representation of how users will navigate through the system. It comprises of the home page that shows the general overview of what the web application does, it has the sign-up feature where a user can register as a freelancer and or as a company. A user may also login as a freelancer or a company. Once logged into the system both users (freelancers & companies) may perform a variety of functions unique to each user profile.

Wireframe

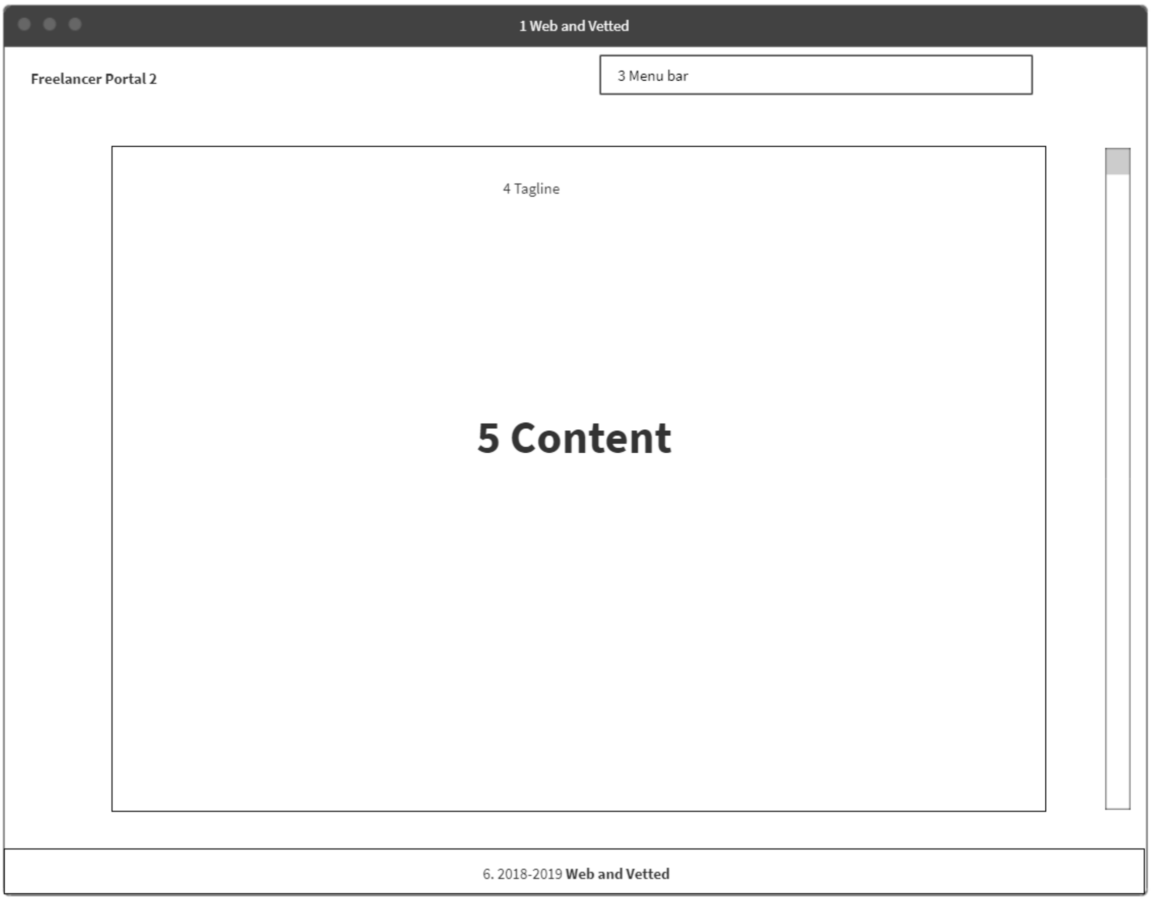


Figure 4: Wireframe of Web and Vetted

1. Shows the title of the Web and Vetted website.
2. This is where the logo is placed.
3. This is the menu bar.
4. This is where you find the tagline.
5. This is where the site content is situated.
6. This is where you find the website footer.

Flowchart

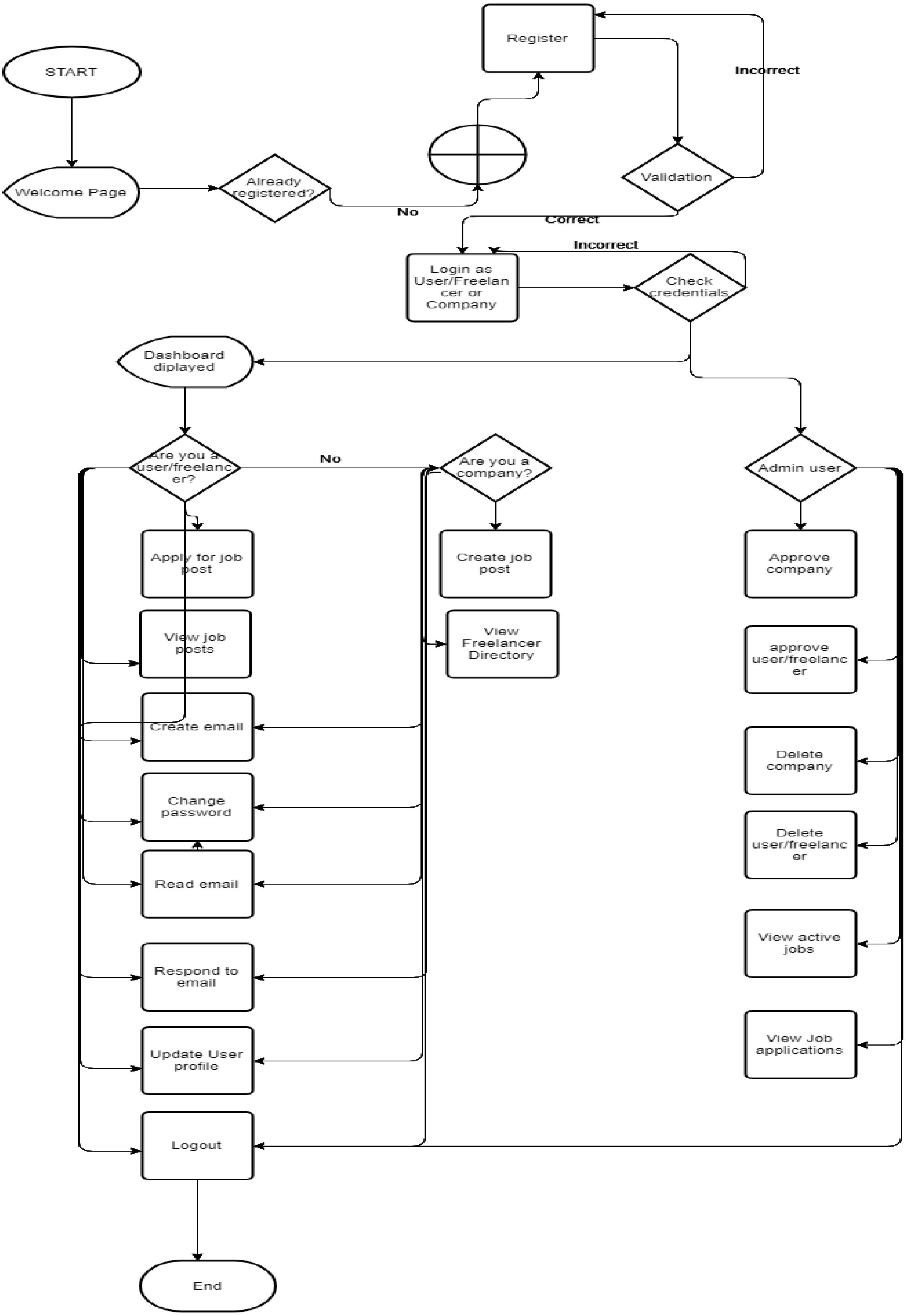
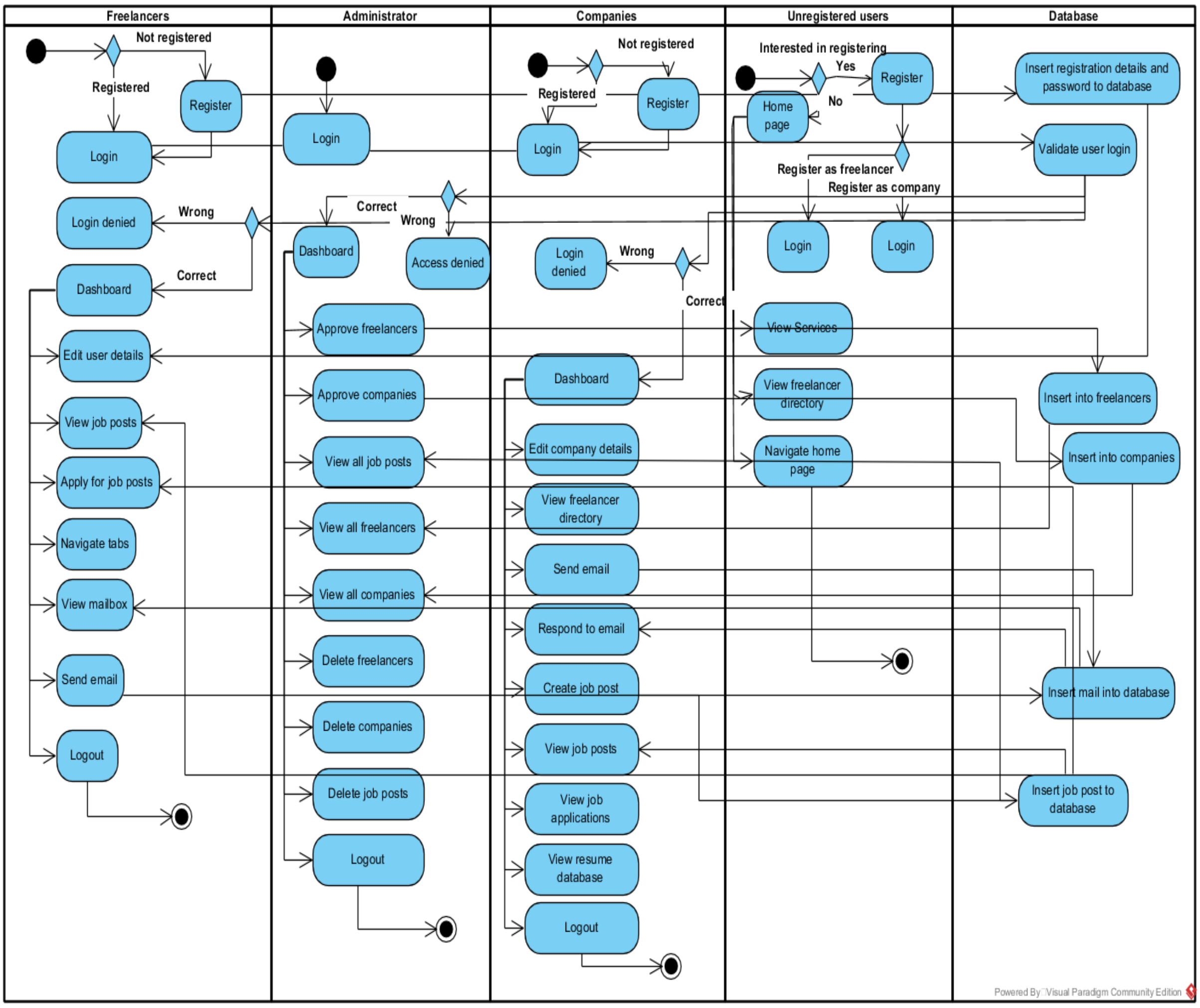
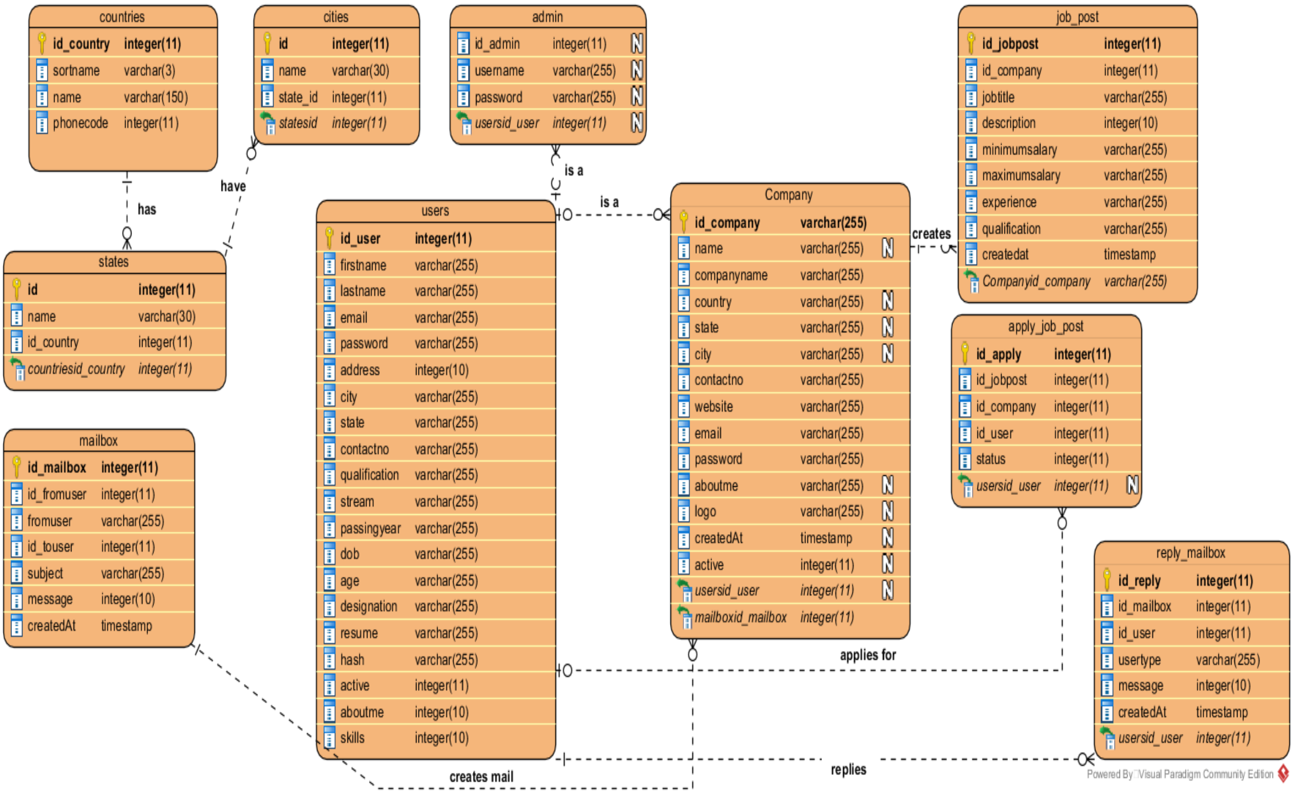


Figure 5: Web and Vetted flowchart diagram

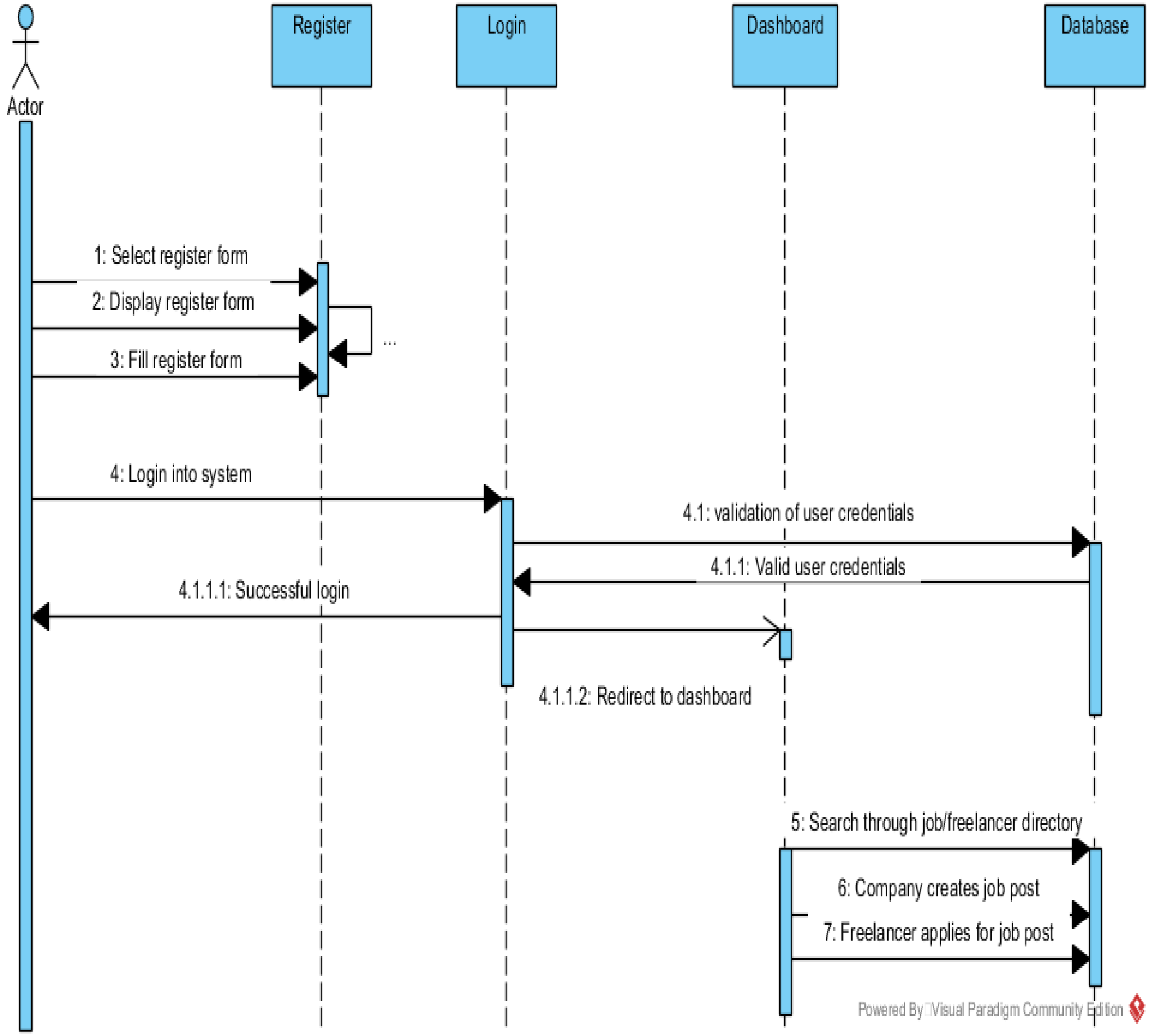
Activity Diagram



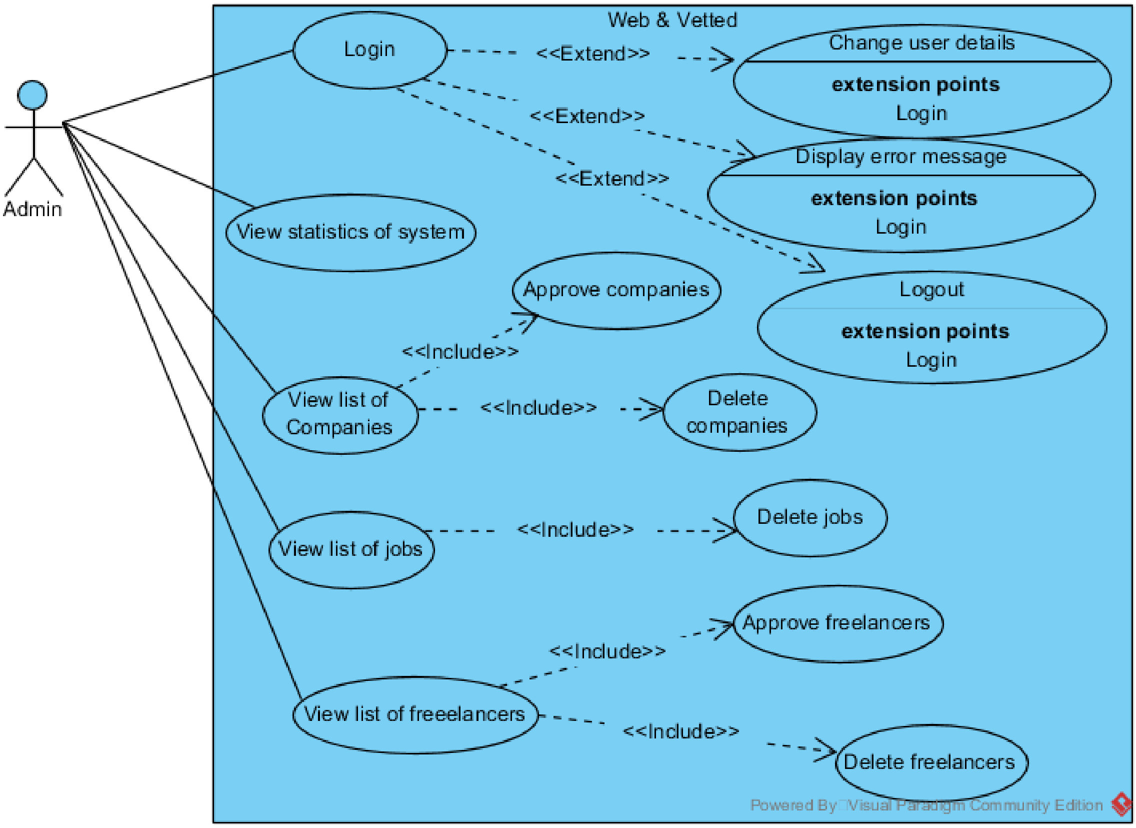
Web and Vetted ER Diagram

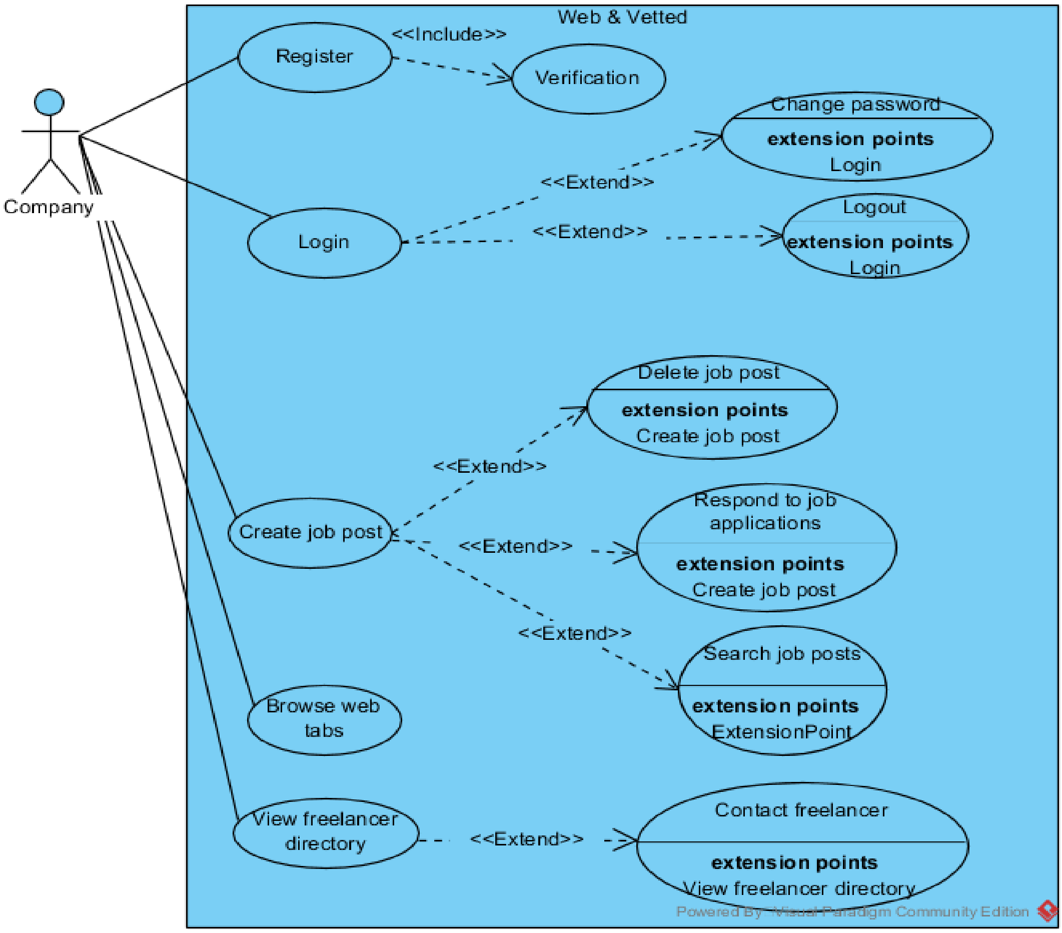


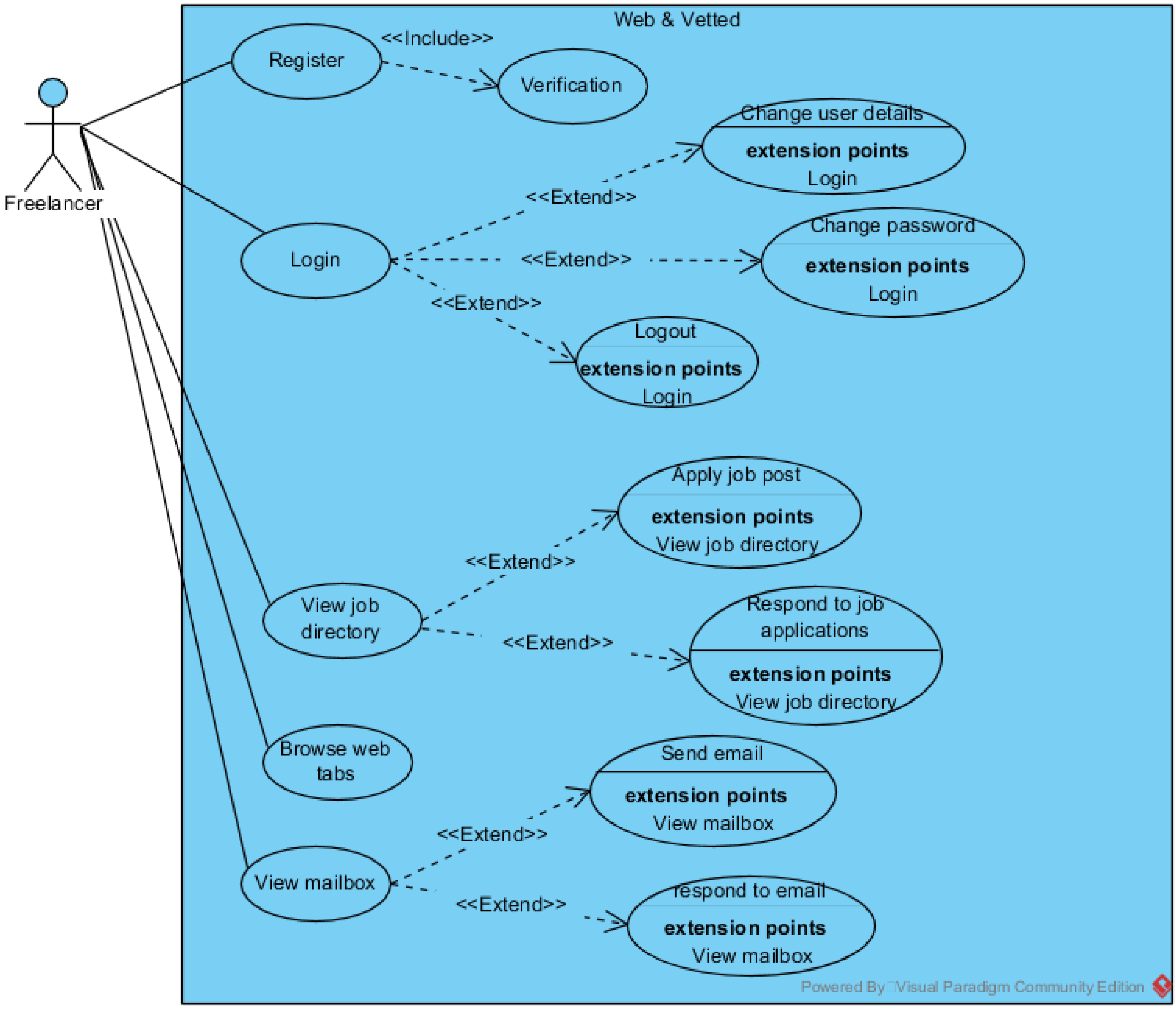
Sequence Diagram


Actor
Register
Login
Dashboard
Database
1: Select register form
2: Display register form
...
3: Fill register form
4: Login into system
4.1: validation of user credentials
4.1.1: Valid user credentials
4.1.1.1: Successful login
4.1.1.2: Redirect to dashboard
5: Search through job/freelancer directory
6: Company creates job post
7: Freelancer applies for job post
Powered By Visual Paradigm Community Edition

Use Cases


Web & Vetted
Admin
Login
<<Extend>>
Change user details
extension points
Login
<<Extend>>
Display error message
extension points
Login
<<Extend>>
Logout
extension points
Login
View statistics of system
Approve companies
<<Include>>
View list of Companies
<<Include>>
Delete companies
View list of jobs
<<Include>>
Delete jobs
Approve freelancers
<<Include>>
View list of freeelancers
<<Include>>
Delete freelancers
Powered By Visual Paradigm Community Edition


Web & Vetted
Company
Register
<<Include>>
Verification
Change password
extension points
Login
<<Extend>>
Login
<<Extend>>
Logout
extension points
Login
Delete job post
extension points
Create job post
<<Extend>>
Create job post
<<Extend>>
Respond to job applications
extension points
Create job post
<<Extend>>
Search job posts
extension points
ExtensionPoint
Browse web tabs
View freelancer directory
<<Extend>>
Contact freelancer
extension points
View freelancer directory
Powered By Visual Paradigm Community Edition


Web & Vetted
Freelancer
Register
<<Include>>
Verification
Login
Change user details
extension points
Login
<<Extend>>
Change password
extension points
Login
<<Extend>>
<<Extend>>
Logout
extension points
Login
View job directory
<<Extend>>
Apply job post
extension points
View job directory
<<Extend>>
Respond to job applications
extension points
View job directory
Browse web tabs
View mailbox
<<Extend>>
Send email
extension points
View mailbox
<<Extend>>
respond to email
extension points
View mailbox
Powered By Visual Paradigm Community Edition

TESTING AND EVALUATION

INTRODUCTION

This part of the portfolio is where assessment and testing of the Web and Vetted site is portrayed in detail. Protocols, testing instruments, customers assessment structure and results for the Web and Vetted site are appeared in this segment. A critical stage in website development, for example, is testing. For the Web and Vetted venture testing was done in all stages in view of the feature driven methodology. To guarantee testing is performed, performance of each component list was recorded by running the code with the end goal that all customers' requirements were met from the on go of the venture. Feature Driven Development technique guarantees that undertakings are isolated into reasonable sub assignments.

Testing is directed with the goal that all customer prerequisites are working before giving to the to the customer, fix a few errors in the event that are recognized well in time just as troubleshooting before giving it to the customer. This additionally should give the customer fulfilment that the venture is of good quality.

Feature Driven Development utilizes different testing levels such framework testing, unit testing just as the useful testing. Every one of these dimensions are completely clarified how they were directed for the Web and Vetted site improvement beneath.

System testing: testing of the entire framework against both functional and non-functional requirements of the Web and Vetted site. This is directed to ensure all requirements of the site are meet before giving it to customer.

Integration testing: this includes testing functionality of site from both the customer and server side with the end goal that there no errors when the site goes live.

Unit testing: This guarantees testing of every product segment of the Web and Vetted site and it is led by the product engineer.

Validation: This type of testing guarantees that the Web and Vetted site fulfils every single indicated functionality. It is through approval to guarantee that all predefined requirement for each client is met. Since this task is utilizing the Feature driven development approach, it is basic to utilize approval type of testing since this technique is more customer valued.

Verification: This guarantee the Web and Vetted site gave to the customer satisfy the imposed terms agreed at the start of the development phase of the project through testing and assessment. Since this venture utilizes the Feature Driven Development subsequently confirmation is indispensable approach as it is customer centred procedure that means to meet the customer's desires by making the framework capacities as the customers envision or anticipates.

Testing Protocols

Black-box Testing

Acceptance testing: This makes acceptance tests on the Web and Vetted site agreed requirements utilizing the structure for assessment to check if the site has all requirements initially agreed upon by the customer. Both the customer and the developer need to sign off the evaluation form after completion as proof on paper.

Non-functional testing: This includes testing non-functional requirements features, for example, consistency, security, effectiveness, framework being of good match with genuine world by having a structure that is coordinating errands just as convenience, etc. It doesn't centre around the functional requirements of the Web and Vetted framework.

Functional testing: this part deals with the testing of the functional requirements of the system done by the programmer.

Regression testing: This is a type of testing directed after fixing any errors identified or after any update to the site so changes don't interfere with the code, for an example when the customer need to make alteration to the underlying requirements of an effectively existing code.

Advantages

- It guarantees that missing functions are distinguished and after that fixed.
- It doesn't devour time since it focuses on functionality of the application instead of altogether executing code sections
- It needn't bother the developer to have nitty gritty information of functional requirements of the system.

Disadvantages

- Testing can here and there be intricate when details are not explicitly explained
- Testing results might be overestimated when testing
- There are potential outcomes that a few projects may not be tried

White Box Testing

The white box testing convention accentuation on the rationale inside and inner system of the application by means of reviewing the source code. This is accomplished by utilizing experiments on the code conditions, ways and explanations. The designer needs to play out this test by picking contributions to practice ways in the code so as decide the expected result.

Advantages

- Effective in discovering mistakes and issues
- Required information of internals of the product under test is helpful for careful testing
- Permits finding concealed mistakes

Disadvantages

- Probably won't discover unimplemented or missing highlights
- Requires abnormal state information of internals of the product under test
- Requires code access

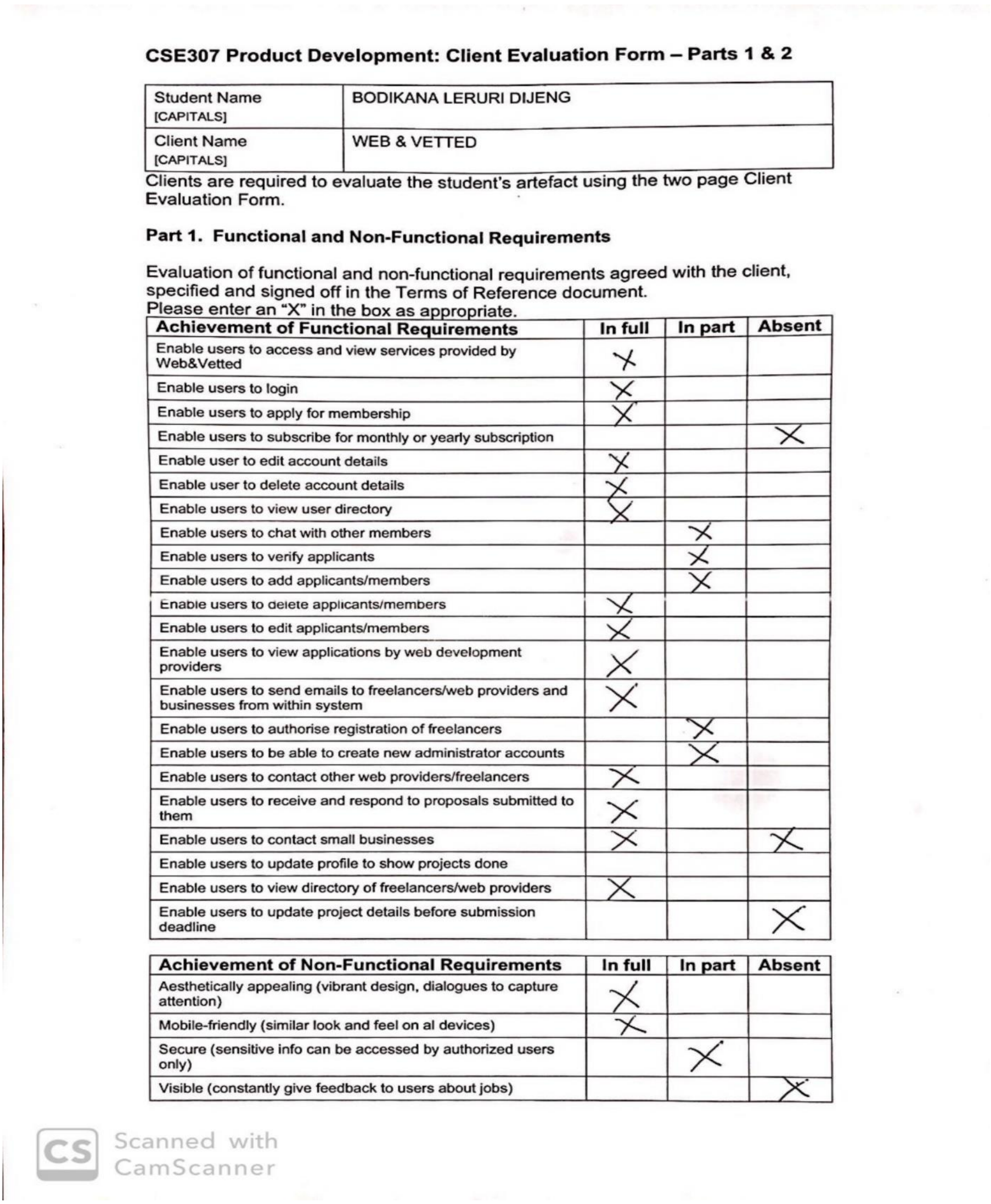
Testing results obtained

Test No	Test Description	Expect Results	Actual Results and Outcomes
1.	**The system should allow users to access and view services provided by Web and Vetted:** the user should be able to view the home page of the website and scroll down to view the services offered.	Home page of the system should load and be displayed.	Outcome positive and home page was displayed.
2.	**The system should enable users to login into the system:** the user clicks on the login button and this should redirect them to the login page. Here they are to choose whether to login as a company or freelancer. Once they have chosen they will the be prompted to login. If the login credentials are wrong then they will be asked to enter correct details. MYSQL database will be used to retrieve login information...	The user has to fill in login information, once filled in they should click on login and if credentials are right then they should be redirected to respective dashboard. If details are wrong then they will be asked to enter right credentials.	Outcomes were positive, once correct credentials were entered user was redirected to corresponding dashboard (freelancer/company). When login credentials were wrong then they were not able to login.
3.	**User registration (Company, Freelancer):** The user clicks on the sign-up button and they are redirected to sign up page. They will then choose to login as freelancer or company. Once they choose they will have to fill out registration information for their respective choice. After this they will click on register. If mandatory fields marked by Asterix are not filled then they will be asked to fill them out before they can register. MYSQL database will be used to store registration information...	Freelancer and company should be able to register. After registration they should both be alerted that they will be able to access their dashboards after approval by the administrator. After approval by admin they both user types should be able to login to the system.	Majority of requirement achieved. Both user types were able to register into system. However, when freelancer was registered, they were not prompted to wait for approval by admin, only the company user was. However, both were then able to login to the system.
4.	**User should be able to apply for membership:** user should be able to apply for membership to the system in the form of registration	User should be able to request to become a member by registering to system.	Requirement achieved and users apply for membership to system by registering to system.
5.	**Users should be able to edit account details:** once logged in, the user should be able to edit their account details using the edit details option on their dashboard	The users should be to change their user details including passwords.	The results are successfully as all users are able to edit their user details.
6.	**All users should be able to delete account details:** the user should be able to deactivate their account should they wish to do so.	User should be able to navigate to settings tab in their dashboard where the option to deactivate account should be visible.	Outcome was positive, user is able to deactivate account.
7.	**Users should be able to view user directory:** users are able to view job and freelancer directories. Freelancer can view job directory, company can view job and freelancer directory and admin can view all directories (company, freelancer, job, job application candidates)	On clicking the appropriate button, users should be able to view respective directories.	Results excellent as all users can view their respective directories.

8.	**Users should be able to chat with other members:** users should have the chat option that will allow them to directly chat with other registered system users in real time.	Upon clicking the chat button there should be a list of online users displayed and user can choose whom to chat with.	Requirement difficult to implement so I chose to enable all users to be able to email each other. I was able to achieve this function fully.
9.	**Users (admin) should be able to verify applicants:** the admin user should be able to approve or reject an applicant's application once they have verified the applicant.	Upon looging into system admin should be able to view list of freelancers and companies. In this list there should be the option to reject or approve next to newly applied members.	Outcome successful as admin can approve or reject newly applied users.
10.	**Users should be able to add applicants:** the admin should be able to add members by approving them from their dashboard.	The admin should be able to approve a new user to the system. Once that user has been approved, the user should now be able to login to the system.	Outcome successful as once a company was approved, they were now able to login to the system.
11.	**Enable users to delete applicants:** the admin user should be able to delete an applicant from the system database subsequently preventing the user to be able to login to the system.	The admin should be able to navigate to the specific user and next to the user be able to click on the delete button which will delete user from database. After this user will not be able to login.	Outcome successful, admin can delete applicants from system.
12.	**Enable users to edit applicants:** users should be able to navigate to edit profile tab on their dashboard which will direct them to page to edit their user details.	Upon clicking edit profile button, user should be taken to corresponding page where the user can edit various user details.	Outcome successful as user can edit their details.
13.	**Enable users to view applications by web development providers**: companies are able to view job applications by web development providers/freelancers**.**	On the company dashboard there should be view my job applications button which will redirect user to job applications submitted for the job post they created.	Outcome successful as company can view job applications submitted to them.
14.	**Enable users to send emails to freelancers and companies from within system:** freelancers and companies should be able to send each other emails using the mailbox on their respective dashboards.	Freelancers and companies should be able to send each other emails using Web and Vetted system.	Outcome successful as both users are able to send each other emails.
15.	**Users should be able to authorise registration of freelancers:** admin users should be able to approve the registration of new freelancers to the system.	In the admin freelancer directory, there should be showing newly registered freelancers, admin should be able to approve or reject the registration.	Admin is not able to fully approve or reject registration of new freelancer.
16.	**Enable user to create new administrator accounts:** the admin user should be able to enter the database where they can add a new admin user to the admin table, which will create a new administrator account.	Admin user should be able to access database and insert new admin user to the admin table therefore creating a new admin account.	Outcome successful. Admin user can be created.
17.	**Enable users to contact other web providers/freelancers:** users should be able to contact freelancers through their mailbox option in the dashboard.	Once a user has navigated to their mailbox, they should be able to send emails to freelancers using the email address the freelancers registered with.	Outcome successful, user is able to contact freelancer using registered email address.
18.	**Enable users to receive and respond to proposals submitted to them:** company users should be able to view applications to their job posts. They should be able to choose a freelancer for the job post.	Company should be able to navigate to job applications tab on their dashboard which will navigate them to the list of applicants for their job posts where they will be able to respond to the job post.	Outcome successful as company can view and respond to job applications.

19.	**Enable users to contact small businesses:** freelancers should be able to contact small businesses/companies through the email feature in the mailbox option.	Once a freelancer has navigated to their mailbox they should be able to send a company an email using the email address the company registered with.	Outcome successful as freelancer can send email to company
20.	**Enable users to update profile to show list of projects done:** freelancers should be able to update their profile to display the projects they have worked on and completed.	Freelancers should be able to update their profile to show their completed projects.	Outcome not successful as freelancer is not able to update project done.

Mandatory Client Evaluation



CSE307 Product Development: Client Evaluation Form – Parts 1 & 2

Student Name [CAPITALS]	BODIKANA LERURI DIJENG
Client Name [CAPITALS]	WEB & VETTED

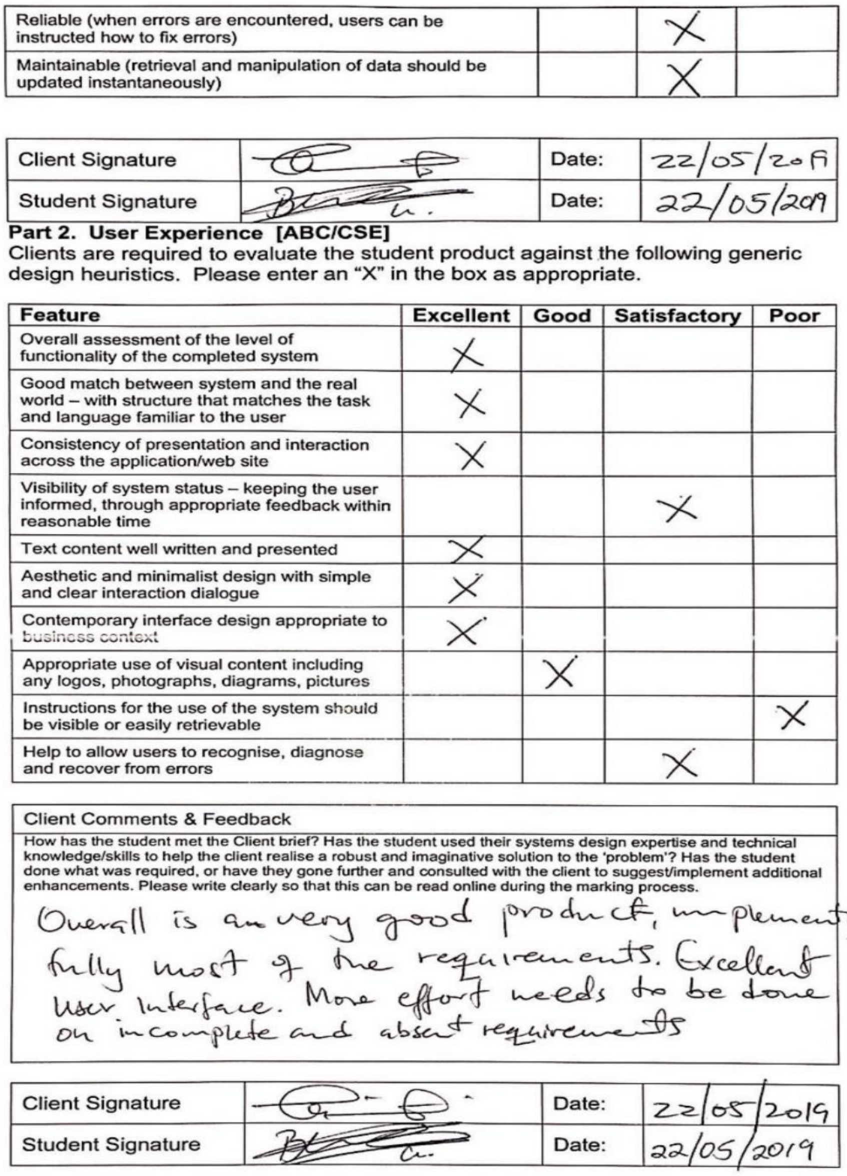
Clients are required to evaluate the student's artefact using the two page Client Evaluation Form.

Part 1. Functional and Non-Functional Requirements

Evaluation of functional and non-functional requirements agreed with the client, specified and signed off in the Terms of Reference document.
Please enter an "X" in the box as appropriate.

Achievement of Functional Requirements	In full	In part	Absent
Enable users to access and view services provided by Web&Vetted	X		
Enable users to login	X		
Enable users to apply for membership	X		
Enable users to subscribe for monthly or yearly subscription			X
Enable user to edit account details	X		
Enable user to delete account details	X		
Enable users to view user directory	X		
Enable users to chat with other members		X	
Enable users to verify applicants		X	
Enable users to add applicants/members		X	
Enable users to delete applicants/members	X		
Enable users to edit applicants/members	X		
Enable users to view applications by web development providers	X		
Enable users to send emails to freelancers/web providers and businesses from within system	X		
Enable users to authorise registration of freelancers		X	
Enable users to be able to create new administrator accounts		X	
Enable users to contact other web providers/freelancers	X		
Enable users to receive and respond to proposals submitted to them	X		
Enable users to contact small businesses	X		X
Enable users to update profile to show projects done			
Enable users to view directory of freelancers/web providers	X		
Enable users to update project details before submission deadline			X

Achievement of Non-Functional Requirements	In full	In part	Absent
Aesthetically appealing (vibrant design, dialogues to capture attention)	X		
Mobile-friendly (similar look and feel on al devices)	X		
Secure (sensitive info can be accessed by authorized users only)		X	
Visible (constantly give feedback to users about jobs)			X



Reliable (when errors are encountered, users can be instructed how to fix errors)		X	
Maintainable (retrieval and manipulation of data should be updated instantaneously)		X	

Client Signature		Date:	22/05/2019
Student Signature		Date:	22/05/2019

Part 2. User Experience [ABC/CSE]

Clients are required to evaluate the student product against the following generic design heuristics. Please enter an "X" in the box as appropriate.

Feature	Excellent	Good	Satisfactory	Poor
Overall assessment of the level of functionality of the completed system	X			
Good match between system and the real world – with structure that matches the task and language familiar to the user	X			
Consistency of presentation and interaction across the application/web site	X			
Visibility of system status – keeping the user informed, through appropriate feedback within reasonable time			X	
Text content well written and presented	X			
Aesthetic and minimalist design with simple and clear interaction dialogue	X			
Contemporary interface design appropriate to business context	X			
Appropriate use of visual content including any logos, photographs, diagrams, pictures		X		
Instructions for the use of the system should be visible or easily retrievable				X
Help to allow users to recognise, diagnose and recover from errors			X	

Client Comments & Feedback

How has the student met the Client brief? Has the student used their systems design expertise and technical knowledge/skills to help the client realise a robust and imaginative solution to the 'problem'? Has the student done what was required, or have they gone further and consulted with the client to suggest/implement additional enhancements. Please write clearly so that this can be read online during the marking process.

Overall is an very good product, implementing fully most of the requirements. Excellent user interface. More effort needs to be done on incomplete and absent requirements

Client Signature		Date:	22/05/2019
Student Signature		Date:	22/05/2019

Technical Deployment of the solution

To deploy the Web and Vetted system I used the following five steps:

1. Preparation:
   There are a few things to consider when you are finalizing a website, and they all depend on what type of deployment you will be completing.

   The three general scenarios of a website deployment is:

   The client has nothing (i.e. this is their first website)
   The client already has hosting and you will be deploying the site on their server
   The client already has hosting but you will be moving to a new server

   I used the first option as it was the first time the client was deploying a web site.

2. Setting up DNS records

   To setup the Web and Vetted website on a new host, the Domain Name server administration management has to be accessed. After accessing the administration management, create an A record which be used to map the domain name to the server's Internet Protocol (IP) or the subdomain record which is used for development of sites which are live. The newly created subdomain is then pointed to new server's internet protocol.

3. Setting up testing site

   This is a stage to test if the Web and Vetted website functions on the live server platform. I recommend setting URL of the subdomain before site deployment. It will be something like test.webandvetted.com that will later be on webandvetted.com. Make sure you do not create a subdomain on the host because it will create another new directory therefore making changes to the Domain Name server. I should be set such that test.webandvetted.com becomes a new separate website. The intension is to make test.webandvetted.com a domain alias.

4. Setting up email accounts

   Setting up an email in website is part of client's priority. The client email address will have to be moved to the new server. This is done by collecting all email address and setting them on the similar mail account of the server that is newly created. In case the client has a mail server that is internal or it is mail host of third –party, then make the Domain Name Server records that deal with mail is right.

Database Configuration:

Database configuration is important before implementing stage five of deployment. The server preferred is Maria databases server which uses the SQL functions. It is robust and flexible. It should be hosted such that host address is of localhost (127.0.0.1) with the server that hosts the Francistown Digital hub and both username and password should match the code defined credentials.

5. Going live

   Even if you are hosting on a new server, take a full backup including any databases of the old website, as you never know when you might need something. After having full control over the Domain name server then alter the record for the domain such that the internet protocol is set to newly webserver and in few minutes the Web and Vetted website will be live.

CRITICAL REFLECTION

This portfolio is based on the development of the Web and Vetted web system and its deliverance to the client. The Web and Vetted system are developed with the aim of giving small businesses a directory of verified freelancers that they can employ to provide web development solutions for them. This initiative was thought of after a lot of companies had complained about hiring freelancers who provided sub-par solutions to them and where no where to be found after handing over the solution to the client.

Initial phase of developing the system was a business pitch where the client was introduced to the company that would be developing their solution. After this there was a meeting where the functional and non-functional requirements were drafted through an interview. All requirements were clear and straight forward and had the aim of creating a system that could easily be used by users. On the 1st of March 2019 an agreement of the requirements specification was made with the client to develop the agreed upon system. The requirements specification form was signed off as a go ahead to begin developing the solution.

After the client signed off a schedule was made so that the solution would be handed over on the 17th of May 2019 and ensuring the appropriate development steps were made during the process. Feature driven methodology affected how the calendar was made, in other words the timetable was finished by ensuring that each element of the necessity is finished at a particular stage. This technique is more customer esteemed along these lines it keeps both the customer and the engineer to follow along if the designer is on the correct route with what concurred with the customer in time. Tasks are broken into smaller sub tasks that are progressively reasonable in Feature Driven Development. All assignments were given a specific measure of hours, assessed dates to be done and genuine dates in which they were completed. The timetable had a depiction of what an assignment is about and furthermore the deliverable segment which indicates what is expected from task.

After research assessment directed on extreme programming, test driven advancement and Feature Driven Development referenced on the philosophy support above of this portfolio, the Web and Vetted site was created utilizing the feature driven development technique which centres generally around breaking complex undertaking into smaller tasks which are sensible and finished well in time. The technique was picked as it is customer centred subsequently it makes both the engineer and customer cooperation to yield a quality item since the customer give criticism of being fulfilled or not. The expert explanation further talks about how I figured out how to pick approach that I utilized, devices and strategies for demonstrating the Web and Vetted framework. UML displaying apparatuses were utilized in demonstrating models, for example, Activity diagrams of the site. The programming language utilized is PHP that utilizes Php grid view framework, utilized MySQL database for capacity and the Apache server of XAMPP to have the Web and Vetted site.

In the structure stage, displaying devices were utilized to draw outlines, for example, utilizes cases and substance graphs with exact portrayals for the Web and Vetted site. Models structured were indispensable since they visualized how the site will resemble. For an occasion the web page map I planned looks practically 100% to genuine site interface I created.

The practitioner statement likewise has models utilized in the development of this site. Both Php and MySQL database are additionally indicated how they will work as it is outlined by the MVC model on the framework documentation.

I utilized black box testing system to test the Web and Vetted site with the customer. The black box testing convention is centred around testing both functional requirements of the site indicated by the customer. This should be done with the customer so they could call attention to where they are fulfilled or not and even give suggestions where system can be improved. I didn't utilize the white box testing since if concentrates more on internal intricacies, that is the source code of the application and individual as engineer is practically inconceivable or tiring to test each line of the code particular if the code has numerous lines. All outcomes got in the wake of testing were recorded to demonstrate anticipated outcomes and genuine outcomes. The customer closed down in the wake of testing the entire framework against determined prerequisites by means of utilization of black box testing. Toward the finish of the task a site that is of extraordinary quality was conveyed to the customer well in time. The quality and time of the undertaking is even confirmed by the assessment structure on the assessment structure area of this portfolio.

The Web and Vetted site has remarkable qualities since both functional and non-functional requirements were incredibly met. The main shortcoming that the customer was not happy with is that I did not manage to implement the approval or rejection option for the freelancer.