

Projet Proposal on Local Tiles Online Order





Submitted By: Binod Rana Submitted To:

Batch: 22 C Sudeep Lal Bajimaya

NCC ID: 00175028 Module Leader

Softwarica College of IT & E-Commerce Dilibazar, Kathmandu

Submitted Data: April-09-2019

Table of Contents

[Chapter 1: Introduction 3](#_Toc5700838)

[1.1 Introduction 3](#_Toc5700839)

[1.2 Background of the projects 3](#_Toc5700840)

[1.3 Problems Statements 4](#_Toc5700841)

[1.4 Description of the project 4](#_Toc5700842)

[1.4.1 Features of the project 4](#_Toc5700843)

[1.5 Overview of the project 4](#_Toc5700844)

[Chapter 2: Scope of the project 5](#_Toc5700845)

[2.1 Scope 5](#_Toc5700846)

[2.2 Limitation 5](#_Toc5700847)

[2.3 Aims 5](#_Toc5700848)

[2.4 Objective 5](#_Toc5700849)

[2.5 Overviews of the scope 6](#_Toc5700850)

[Chapter 3: Development Methodology 7](#_Toc5700851)

[3.1 Description of Methodology Chosen 7](#_Toc5700852)

[3.2 Design Pattern 10](#_Toc5700853)

[3.3 Architecture 11](#_Toc5700854)

[Chapter 4: Project Planning 12](#_Toc5700855)

[4.1 Work Breakdown Structure (WBS) 12](#_Toc5700856)

[4.3 Grant Chart 15](#_Toc5700857)

[Chapter 5: Risk Management 17](#_Toc5700858)

[Chapter 6: Configuration management 20](#_Toc5700859)

[Chapter 7: Conclusion 21](#_Toc5700860)

[References 22](#_Toc5700861)

**List of Figure**

[**Figure 1: Waterfall Model ……………………………………………………………………. 7**](#WaterfallModel)

[**Figure 2: Waterfall Methodologies Vs Agile ……………………………………….. 9**](#WaterfallMethodologiesVsAgile)

[**Figure 3: MVC Design pattern …………………………………………………….. 10**](#MVCDesigpattern)

[**Figure 4: Client-Server Architecture ……………………………………………… 11**](#ClientServerArchitecture)

[**Figure 5: Work Breakdown Structure (WBS)**](#wbs)**…………………………………….. 12**

[**Figure 6: Gantt Chart**](#ganttchart)**……………………………………………………………….. 16**

[**Figure 7: Gantt Chart Timeline ……………………………………………………. 16**](#ganttcharttimeline)

[**Figure 8: Risk Management Life Cycle**](#risk) **…………………………………………… 18**

[**Figure 9: Configuration Management directory structure**](#configuration) **……………………….. 21**

# Chapter 1: Introduction

## 1.1 Introduction

The company was established in 2015 AD at Galkot, Baglung. Since starting it has been providing local tiles at the local customer and for the different customer of big cities of the country. But till now it has been running with the phone calls and physically being presented in the company to make an order and to know details about sizes and prices of tiles as per sizes and this is the most difficult task for customers.

So, Here I will be developing a system named **Local Tiles Online Order** which can make easy to look at the information about tiles and make easy to order. The system will provide different sizes of tiles and the price of each piece of tiles according to sizes available in the company. After looking at all information customer can decide how much do they want and what sizes do they need in numbers and can get cost calculation. As they confirm order they can also select types of vehicles as their needs and wishes.

## Background of the projects

About the background of the projects, there are some system or website which provide online stores of both local and imported marbles or tiles from outside of the country and those systems also provide other online stores along with marbles.

But about my projects, **Local Tiles Online Order** will only promote or sales local tiles online within the country. And the main points about these projects is this **Local Tiles Online Order** is only be used for my company.

## Problems Statements

It was really challenging and difficult tasks to remember details of the customers like company name or customer name, sizes of tiles and number of tiles they have order, types of vehicle they have chosen to transport. All these information were done by phone calls. And if the customers are physically present in the company then orders of customers need to be note done into papers.

## Description of the project

### 1.4.1 Features of the project

The feature of my project are shown below:

* Login and registration.
* Register user.
* View user details.
* Insert, Update, Delete and Retrieve of the tiles details.
* Show types of vehicles.
* List of order.

## Overview of the project

This project is about an online order of local tiles where the customer can make an order of local tiles, choose types of vehicle as their needs. Payments are done after delivery of products. Customers can also provide feedback about the product. And all these can be done only if the customers register into the system providing valid and required information for the system and signing.

# Chapter 2: Scope of the project

## 2.1 Scope

As every project has its own scope, the scope of my project Local Tiles Online Order is to provide every details information of local tiles online. And by the help of Local Tiles Online Order, people who have faced difficulty for finding information about local tiles will be easy for them to look at the details of local tiles and can order online as their needs.

## 2.2 Limitation

* Online payment is not available.
* Only can be used within the country.

## 2.3 Aims

* Computerized manual systems.
* Design an online system for easy ordering of tiles and choosing vehicle types.
* To increase the productivity of the company.

## 2.4 Objective

* To keep details of tiles like different types of sizes, cost as per sizes and types of vehicle.
* To upgrade and enhance the existing system by increasing it’s effectiveness and efficiency.
* To design a good user interface for efficient and easy use of the system.
* To automatized functionality and features of the company.

## 2.5 Overviews of the scope

The scope of my project is to provide online details of tiles for the customer. But being an online system my system does not provide any online payment and used only within the country. My project aims to change the manual process of keeping details into the computerized system and to achieve those aims I have some objective like to automatized feature and functionality, design good user interface and keep details of tiles online.

# Chapter 3: Development Methodology

## 3.1 Description of Methodology Chosen

This system will be designed for **Shreesh Tiles Company** which is owned by my family. Hence, I have a clear and fixed requirement of the company. So, I will choose the Waterfall model as my software development methodology.

The waterfall model is a linear, sequential approach to the software development life cycle(SDLC). This is mean that another phase in the development process begins only if the previous phase in complete. In this model, the phases do not overlap, goals are set for each phase of development and can not be revisited after completion.

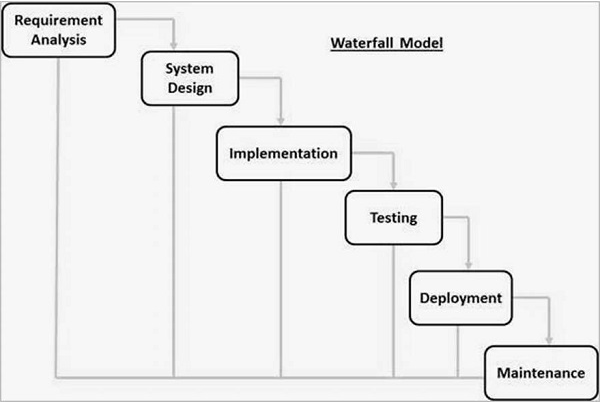


Figure 1: Waterfall Model

The advantage of the waterfall model:

* In the waterfall model, each phase has deliverables and a review process.
* Waterfall model is easy to use and understand and simple.

The disadvantage of the waterfall model:

* This model is not suitable for projects where requirements may be changed in the middle of phases.
* Difficult to measure progress within phases or stages. [(Anon., n.d.)](#SDLC)

Comparison with other Methodology:

* A waterfall is a linear approach to software development. Whereas, Agile is an iterative, team-based approach to development. Waterfall model is a sequential model so it is completed step by step according to its development phases and does not require the participation of customers. But Agile development is an iterative approach. There is a huge interaction between customer and developers during the development of projects and it focuses on customer satisfaction. Hence, I will be using the waterfall model for my project.

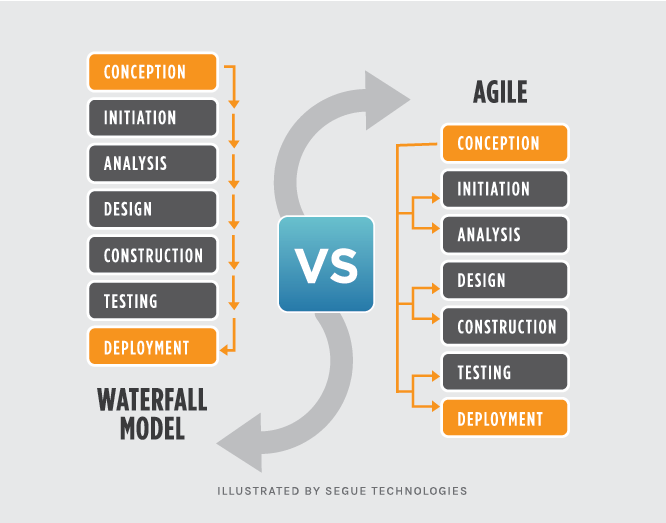


Figure 2: Waterfall Methodologies Vs Agile

## 3.2 Design Pattern

The Model View Controller(MVC) design pattern defined that an application consists of a Model, View, and Controller. The pattern requires that each of these is separated into different objects.

* **Model:** It does not contain logic describing how to represent the data to a user, only pure application data is contained.
* **View:** Visual representation of the model. The view knows how to access the model’s information, however, it does not know what this data means.
* **Controller:** A controller is exited between the view and the model. Controller handle events triggered by the views and execute the proper reaction to these events and the result of this action is then reflected in the views. [(Anon., n.d.)](#MVC)

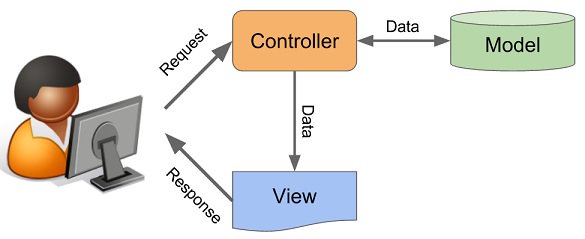


Figure 3: MVC Design pattern

I have chosen the MVC design pattern because it supports the rapid development process, provides multiple views, the modification does not affect the entire model and it supports friendly development platform of the system.

## 3.3 Architecture

Since my system aims and focuses to provide easy services to the customer without any interruption. So, I will be using Client-server architecture. The client-server architecture is a shared architecture where loads of client-server are distributed. The server holds all the resource in a centralized resource system. The client-server gets numerous performance at its edge for sharing resources to its customers when requested. In this architecture, the client and server may be in a network or on the same. Client services will not be interrupted in this architecture since it is service oriented. [(Anon., n.d.)](#client)

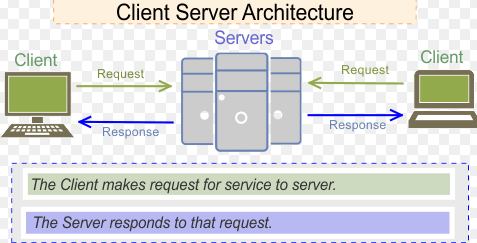


Figure 4: Client-Server Architecture

# Chapter 4: Project Planning

## 4.1 Work Breakdown Structure (WBS)

Work breakdown structure is a way to divide or breakdown of a project into a smaller portion or different stages. It is also a method of organizing and completing work in a project. [(Anon., n.d.)](#WBS)

The purpose of using of WBS are:

* To identify potential risks in a project.
* To make a large project more manageable by breakdown into smaller chunks.
* To quickly developed projects schedule and budgets by allocating price and time for every stage of Work Breakdown Structure.

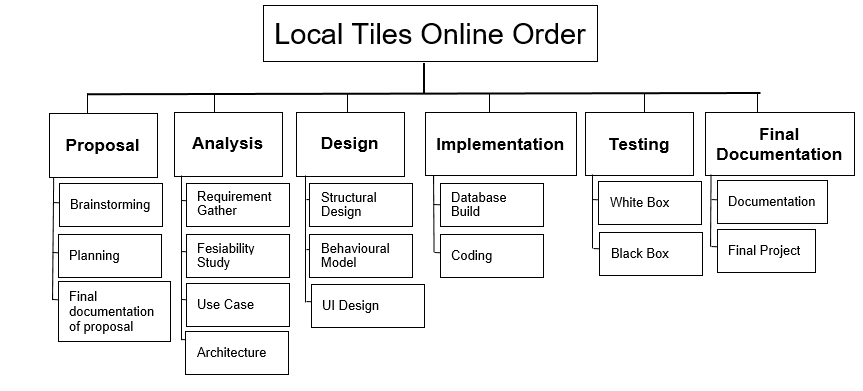


Figure 5: Work Breakdown Structure (WBS)

4.2 Milestone

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S.N | Works | | | | | | Start Date | End Date | Days |
| 1. | **Proposal** | | | | | | 25 – March | 9 – April | 16 days. |
|  | i. | | Brain Storming | | | | 25 – March | 28 – March | 4 days |
| ii. | | Planning | | | | 29 – March | 1 – April | 4 days |
| iii. | | Final documentation of proposal. | | | | 2 – April | 9 – April | 8 days |
|  | **Analysis** | | | | | | 10 – April | 7 – May | 28 days |
|  | i. | | | | Requirement gather | | 10 – April | 18 – April | 9 days |
|  | ii. | | | | Feasibility | | 19 – April | 26 – April | 8 days |
| iii. | | | | Use-Case | | 27 – April | 2 – May | 6 days |
| iv. | | | | Architecture | | 3 – May | 7 – May | 5 days |
|  | **Design** | | | | | | 8 – May | 1 – June | 25 days |
|  | i. | | | | | Structural design | 8 – May | 16 – May | 9 days |
| ii. | | | | | Behavioral design | 17 – May | 25 – May | 9 days |
| iii. | | | | | UI design | 26 – May | 1 – June | 7 days |
|  | **Implementation** | | | | | | 2 – June | 21 – June | 20 days |
| i. | | | Database build | | | 2 – June | 6 – June | 5 days |
| ii. | | | Coding | | | 7– June | 21 – June | 15 days |
|  | **Testing** | | | | | | 22 – June | 28 – June | 7 days |
| i. | White Box Testing | | | | | 22 – June | 25 – June | 4 days |
| ii. | Black Box Testing | | | | | 26 – June | 28 – June | 3 days |
|  | **Final Documentation** | | | | | | 29 – June | 9 – Jully | 11 days |
|  | Documentation | | | | | | 29 – June | 5 – July | 7 days |
|  | Final Report | | | | | | 6 – July | 9 – July | 4 days |

Milestone is tools used in project management. Until your projects are completed. The milestone should represent a clear sequence of the stage that continuously builds up.

In this way, I will be doing a milestone of the project for 107 days mentioning clearly the stages, sub-stages, start date, end date and total days required for each stage of the project.

**Proposal:** The proposal is proposed document of my project. It covers 16 days which start from March 25. For **brainstorming** and **planning**, I have allocated 4 days and **Final documentation** I have allocated 8 days so that I can properly document my proposal.

**Analysis:** Analysis is the most important stages of any project. So it covers 28 days in total. Inside analysis, I will have **requirement gather** of which I have allocated 9 days because since requirement gather is the key stage in the analysis. For the **feasibility study**, I have allocated 8 days because I will be making this system for my own company so it is feasible. As for **use case** I have already done use case in another project also so I have estimate 6 days. Finally, for **architecture**, I have allocated just 5 days.

**Design:** Design shows how my system will work and look. It covers 25 days with different substages. Structural, Behavioural and UI design will be substages. Among those, I have allocated 9 days for **structural** and 9 days for **behavioral design** because those designs are the main part of the system. Then since **UI design** is the main interface between user and system I have allocated 7 days.

**Implementation:** Implementation means coding and database build of the project. It consists database build and coding as substages which totally cover 20 days. Among these, I have to allocate 5 days for database build because I am familiars with the database build in the previous project. Since I am doing my project using java it will be a little more challenging for coding, so I have allocated 15 days.

**Testing:** Testing means rechecking of code. It covers 7 days. Within 7 days, I have allocated 4 days for **white box** testing because it is slightly challenging test and need to show internal structure. Then for **black box** testing, I have located 3 days because it does not show the internal structure.

**Final documentation:** Final documentation will cover 11 days. Within its documentation and the final report will be included. For **documentation**, I have allocated 7 days because this is one of the most important phases of any project and shows all the information about the entire project and then for a **final report**, I have allocated 4 days because it only shows the summary of the projects.

## 4.3 Grant Chart

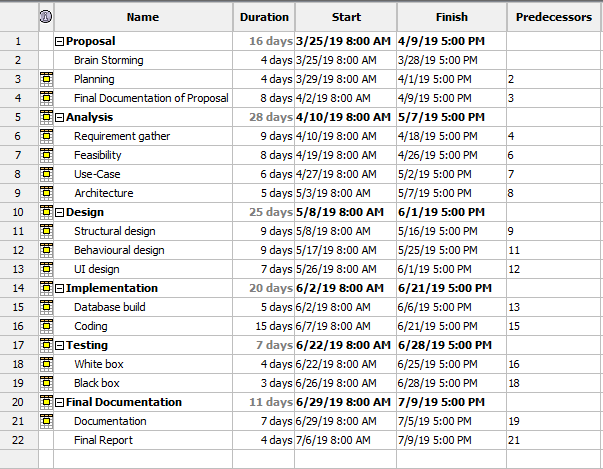


Figure 6: Gantt Chart

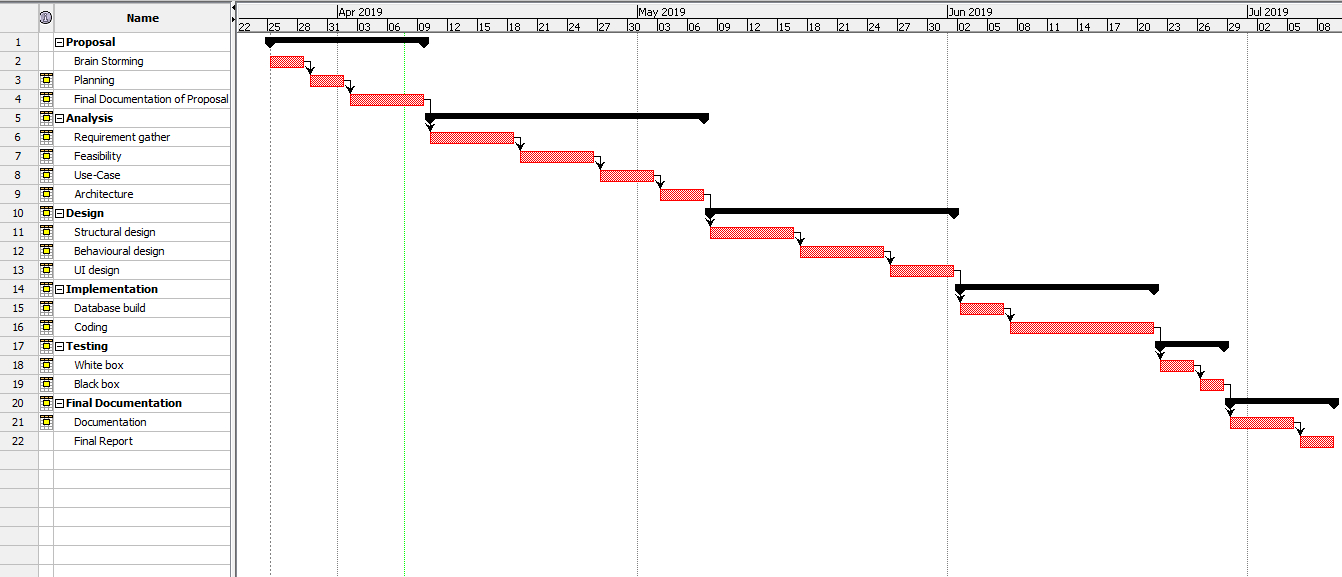


Figure 7: Gantt Chart Timeline

For the successful development of any project time estimation for the stages and sub-stages of the project should be done. Here, I have divided my project into six main stages which are Proposal, Analysis, Design, Implementation, Testing, Final Documentation, and all these main stages have two or more sub-stages. For all these stages and sub-stages I have clearly estimate time frame and show it in the Gantt Chart timeline that will be required to finish them. Which will directly help to finish my project in given time.

# Chapter 5: Risk Management

The unconditional events that directly or indirectly effect to the system being developed are called risk and identifying, managing, accessing and taking action to those risk by defining possible solution is called risk management. [(Anon., 2016)](#risk)

The different risk may occur from different sources like data theft, natural disaster, unauthorized access, legal issues, and others can happen anytime. Some of these are unavoidable whereas some of those can be controlled. So, as to be saved from those threats risk management provides an alternative solution. Which directly helps to increase the success level of projects.



Figure 8: Risk Management Life Cycle

For the estimate of the impact of different risk we use

**Impact = Likelihood \* Probability**

Likelihood of risk and its value

|  |  |
| --- | --- |
| **Likelihood** | **Value** |
| Low | 1 |
| Medium | 2 |
| High | 3 |

The consequence of risk and its value

|  |  |
| --- | --- |
| **Consequence** | **Value** |
| Very Low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very High | 5 |

Risk identification, calculating impact and alternative action for risks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.N** | **Risk** | **Likelihood** | **Consequence** | **Impact** | **Action** |
| 1 | Limited time | 2 | 4 | 8 | Every step of the project should be completed in the assigned time. |
| 2 | Equipment failure | 1 | 5 | 5 | Backup of projects regularly in google drive and GitHub. |
| 3 | Malicious attack | 2 | 3 | 6 | Advanced security and Encryption of data. |
| 4 | Human errors | 2 | 2 | 4 | Testing and review each step of the project. |
| 5 | Unauthorized access | 2 | 5 | 10 | Secure the system with a strong password and encryption of data. |
| 6 | Data Breach | 2 | 4 | 6 | Strong rules, security policy, and implementation. |
| 7 | Cost | 1 | 2 | 2 | Proper analysis of cost before starting the project. |
| 8 | Natural calamities | 1 | 4 | 4 | Safe working environment and backup project daily. |

# Chapter 6: Configuration management

Configuration management is the collection of activities, tools, processes, and methods. That can be used by project practitioners during the project life cycle to manage items. [(Anon., 2016)](#configuration)

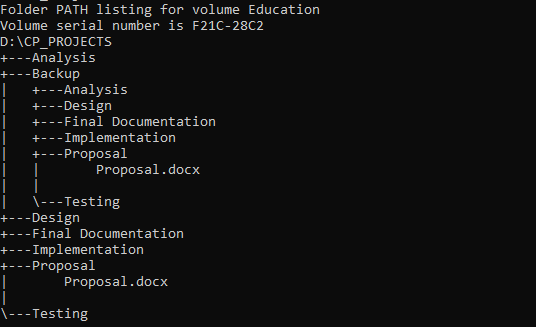


Figure 9: Configuration Management directory structure

**Version Controller**

Version controller keeps the records of changes to a file or group of files so that we can recall specific files later. It allows a return to previous states to the selected files or the entire project if anything happens to the original file of the project. We can also see the changes in files over time. So, as for backup of the project in the process of completing the project, I have used GitHub as for version controller which username is [**ranabinod615@gmail.com**](mailto:ranabinod615@gmail.com)and link for account [**https://github.com/Binodmagar/CP\_Projects**](https://github.com/Binodmagar/CP_Projects)**.**

I have also backup my entire project offline in a folder named Backup. From here also I can revert my project back to the previous state if any unconditional harms happen to the project.

# Chapter 7: Conclusion

By making detail study of the project form the different aspects. I have decided that I will be using structural, behavioral and UI design as for design. For the methodology and architecture, the waterfall model and client-server architecture will be used. As for coding and database build, various types of tools will be used. And, for the successful completion of the project, I have done Work Breakdown Structure(WBS) and time frame estimation for the stages and sub-stages of the project in the Gantt Chart Timeline.

By doing all this activity time required will be less, the cost will not be high, configuration management is good and small to small risk management can be done. So that I proposed the proposal for this project.

# References

Anon., 2016. *definition of risk management.* [Online]   
Available at: [https://economictimes.indiatimes.com/definition/risk-management](%20https:/economictimes.indiatimes.com/definition/risk-management)  
[Accessed 2 4 2019].

Anon., 2016. *introuduction to configuration management.* [Online]   
Available at: <https://www.digitalocean.com/community/tutorials/an-introduction-to-configuration-management>  
[Accessed 4 4 2019].

Anon., n.d. *MVC.* [Online]   
Available at: [https://www.geeksforgeeks.org/mvc-design-pattern/: MVC design pattern](%20https:/www.geeksforgeeks.org/mvc-design-pattern/:%20MVC%20design%20pattern)  
[Accessed 29 3 2019].

Anon., n.d. *SDLC.* [Online]   
Available at: [https://www.tutorialspoint.com/sdlc/sdlc\_waterfall\_model.htm: waterfall model](https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm:%20waterfall%20model)  
[Accessed 25 3 2019].

Anon., n.d. *what-is-client-server-architecture-and-what-are-its-types.* [Online]   
Available at: [http://apachebooster.com/kb/what-is-client-server-architecture-and-what-are-its-types/](%20http:/apachebooster.com/kb/what-is-client-server-architecture-and-what-are-its-types/)  
[Accessed 6 4 2019].

Anon., n.d. *workbreakdownstructure.* [Online]   
Available at: [https://www.workbreakdownstructure.com](%20https:/www.workbreakdownstructure.com)  
[Accessed 1 4 2019].

**Final Documentation**

Documentation

Final documentation of proposal

Final Project

Local Tiles Online Order

Architecture

Use Case

UI Design

Database Build

Planning

Brainstorming

Black Box

Coding

Behavioural Model

Fesiability Study

White Box

Requirement Gather

Structural Design

**Testing**

**Implementation**

**Design**

**Analysis**

**Proposal**