Software Requirements Document

Project Title: FiscalOptima

Shashi Kumar Kadari Mallikarjuna #015092348

Hamza Mekouar #016099159

Sukriti Agarwal #016280171

Table of Contents:

1. Introduction

This section gives an overall description and a detailed overview of all the contents in the whole document. It includes purpose, the intended audience for our web application, project scope, definitions and a list of references.

1.1. Purpose

The purpose of this document is to give an elaborate description of the requirements for the “FiscalOptima” web application. This document contains the features offered by the application, constraints, and user interface. This document is primarily intended to be proposed to a customer to get their approval and use it as a reference for developing the alpha version of the web application.

1.2. Definitions, Acronyms and Abbreviations

Term	**Definitions**
User	Someone who interacts with the web application.
Administrator	The system administrator who is responsible for running and giving permission for the establishment and leading system.
Business Owner	Someone who runs a business and wants to be an active user of the web application.
Stakeholder	Anyone impacted by/involved in the project.
Web Application	An online application that is designed to provide multiple useful features to the user.
Database	A collection of tables and rows in which a set of data can be stored.
Web Server	It is a program that uses HTTP to serve the files that form Web pages to users.

Transaction	An instance of buying or selling goods or services.
Monthly statement	A statement which is generated on monthly basis. It has the complete details of business transactions.
Invoice	A list of goods sent or service provided, with a statement of the sum due for these.
Investment	The action of spending money to buy something hoping to make a profit.
Expenditures	The amount of money spent by a businessman.
Income	The total money received by/from a business transaction.
Profit	Financial gain through the means of business.
Supplier	The person who is selling something or offering some services.
Trader	A middleman who buys goods from the manufacturer and sells it to a retailer.
DESC	Description
RAT	Rational
DEP	Dependency

1.3. Project Scope and Reading Suggestions

The “FiscalOptima” is a Finance based web application which would help businessman and other small business owners. This application would help users to manage their business well. This application would help our users to find their transactions, view their monthly statements and generate invoices upon request. All this data would be accessible only if the users provide their financial details, all their expenditures, and savings as a detailed transaction.

Based on the collected information and the data already present in our database (which is located on a web server), our web application would verify user information to produce an accurate invoice which would reflect upon the business' profits or loss. This application would need access to the Internet and the allocated database to generate results and also produce an update on the user's account balances. This application would work on multiple web browsers like Google Chrome, Internet Explorer and compatible on all versions.

1.4. References

1.4.1. Pressman, R. S., & Maxim, B. R. (2015). *Software engineering: A practitioners approach*. New York: McGraw-Hill.

1.4.2. (n.d.). Retrieved from https://bbcsulb.desire2learn.com/d2l/le/content/502381/viewContent/5458275/View

2. Overall Description

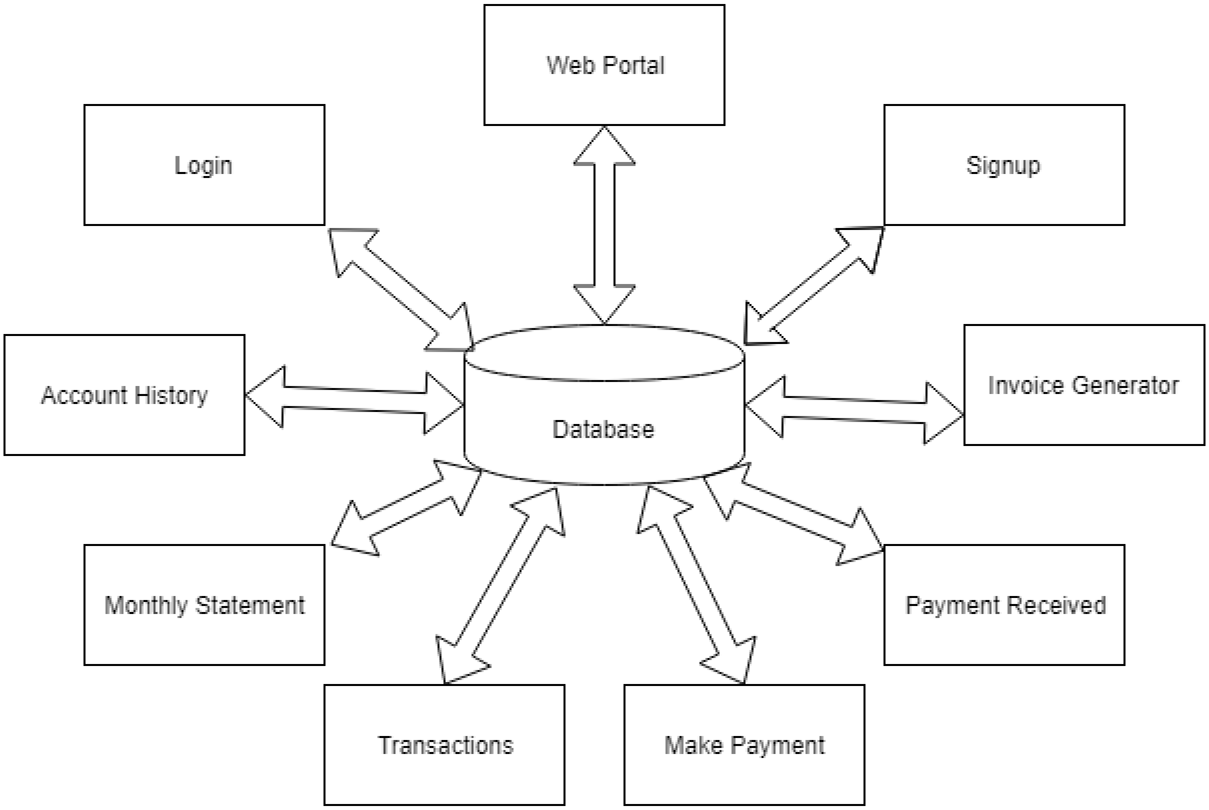
This section gives a detailed overview of the application as a whole. The whole perspective of data retrieval, data storage, and the recorded transaction will be explained in detail. This section will also explain the basic functionality of the web application. An elaborate block diagram would explain the perspective along with listing some of the product features in detail. It would also list all the users and stakeholders of the website listing the available functionality for each person using the website. Some constraints of the applications along with user manual for the website.

2.1. Product Perspective

The web application will be used to track all transactions only for verified users. The application will help to view information about their account balances along with their most recent transactions. The web hosting site along with the database would help to retrieve and manage user's information to view all the records as a whole. The application would need to communicate with the Internet along with the database to find the user and verify his identity.

At first, a user would sign up for the web application, thus making an account on our application. Next, this data would be stored in database, so that whenever a user logs in the application the data would be verified from the database. After the verification is successful the user can browse through the website to see his/her account history, transactions, view monthly statements and also generate invoice. If and whenever data needs to be retrieved from database, that happens only if verification is true. Database retrieval generates queries and if the queries result is good then the request completes. Thus, giving an output on the screen to the User Interface.

Since this is a data-centric product with internet access, thus data needs to be stored in the database at every click. The web application will communicate with the database at all time, but in different ways depending on which functionality the user chooses.



2.2. Product Features

With this application, the users will be able to smartly manage their businesses. The outputs would be generated based on the user inputs and the already existing data in our databases. Upon the verification of the user, the application would search through several criteria until it finds the output desired by the user. The project entails the following features.

2.2.1. **Payable Feature**: This feature would allow businesses to track their outstanding payments, to make sure payments are made correctly to the appropriate suppliers. This feature would ensure that all the payments by the users are made at the right time. The payable feature would deduct from the existing balance and also makes sure the user is not charged an overdraft fee. This transaction would be a part of the monthly statement as an expense.

2.2.2. **Receivable Feature**: This feature would effectively manage clients' activities and would constantly update the owner's account details with the new updated balance. There could be instances when a payment is not received on time and thus, as soon as a payment is made, the user is updated. This transaction would be a part of the monthly statement as the user's income.

2.2.3. **Accounting Feature**: This feature would simply integrate all the purchase and sale statements, incorporate the profits and losses and provide monthly financial reports. This feature accounts all the valid transactions inputted in the web application and also the existing from the database.

2.2.4. **Monthly Statements Feature**: This feature would produce a financial statement on the basis of the accounting feature, which would display all transactions, expenditures, savings, and profits. Expenditures of the month are mostly based on the input provided by the user. Income would be a part of the statement only if the whole payment is received before the statement is generated if not the transaction is listed in the next month. Profits are based on verified payments only. This statement is auto-created and does not need to be requested at the end of each month.

2.2.5. **Invoice Generator Feature**: This feature creates an invoice for all the requested transactions. The invoice will be generated only if the user requests the generation. The invoice generation would be possible only if there is at least one transaction to record and show. This transaction could be an expenditure or an increase in income recorded for a particular user.

2.3. User classes and Characteristics

There are two different kinds of users that are going to be interacting with our application: business owners and administrators. Each of these two types of users has different use of the system so each of them has their own requirements.

The business owners will be using the web portal to use the features provided by our web application that will help their business. In the web portal, the business owners will manage information about their everyday business transactions, generate an invoice and monthly statements, etc.

The administrator will be the stakeholder of the web application. The administrators also interact with the web portal and the database too.They are managing the users of our web application. The administrator can add, update or delete a user of our application.

2.4. Design and Implementation Constraints

Internet connection is a constraint for the web application. Since the web application can be viewed on any web browser, compatible on all versions. Thus, each browser could have a different viewing or minor changes in the user interface of the website.

Also, internet connection is used to fetch data from the database which happens over the Internet too. Therefore, if the internet is down then the users or the admins would not be able to view the web application. Hence, Internet connection is a must for the application to function.

The web application needs access to the database too and each database has a limited capacity. Thus, the capacity of the database could be a constraint too. There could be several queries to the database and thus how the queries are queued up would make a difference in how the result is generated. The more the number of queries, the longer it would take to fetch data from the database.

3. System Features

3.1. System Feature 1

The “FiscalOptima” web application needs a Session Timeout, because if a user is inactive on a certain page for a really long time. Thus to ensure the security of the financial information timeout feature is implemented.

This application is supposed to contain a lot of financial information this feature is needed to reduce the online threats. The attacker/hacker could gain access to the prolong inactive sessions viewing their private financial information or even modify a user’s account (like changing a transaction or total account balance). Thus, to prevent the inactivity, the session timeout feature is implemented.

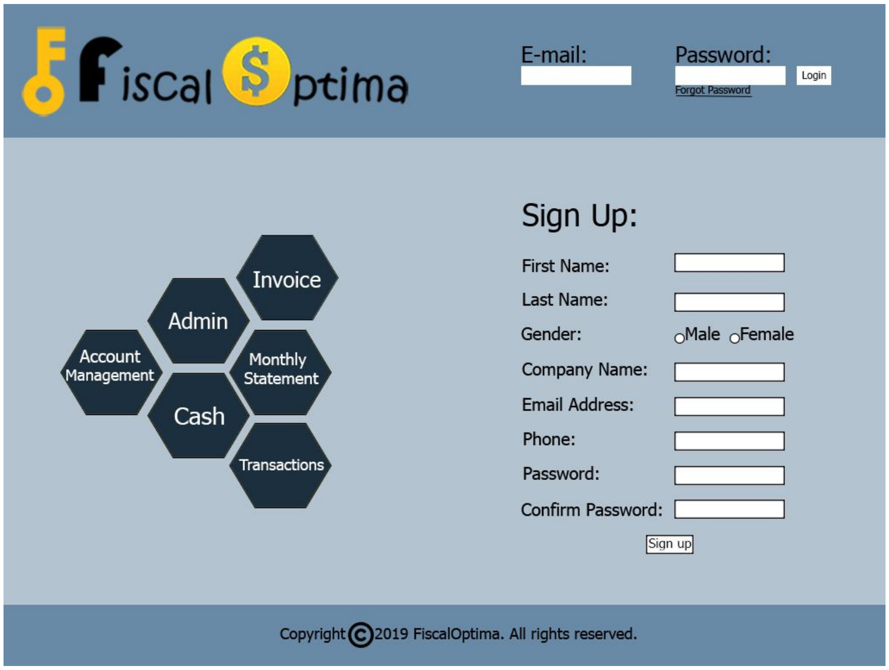
3.2. System Feature 2

Our application is a finance based application, meant to help users to manage their business better. Thus, our application will have to deal with a lot of calculations and even a slight miscalculation could result in huge errors. One of the features of our systems is to calculate and round up all the calculations up to 3 decimal places so that all the calculations could be a little more precise.

4. External Interface Requirements

4.1. User Interface

The main page has two functions; a first time user is able to create an account using the sign up section. Additionally, an account holder can sign in to the web application through the main page. Figure 1.



Sign Up:

First Name:

Last Name:

Gender: ○Male ○Female

Company Name:

Email Address:

Phone:

Password:

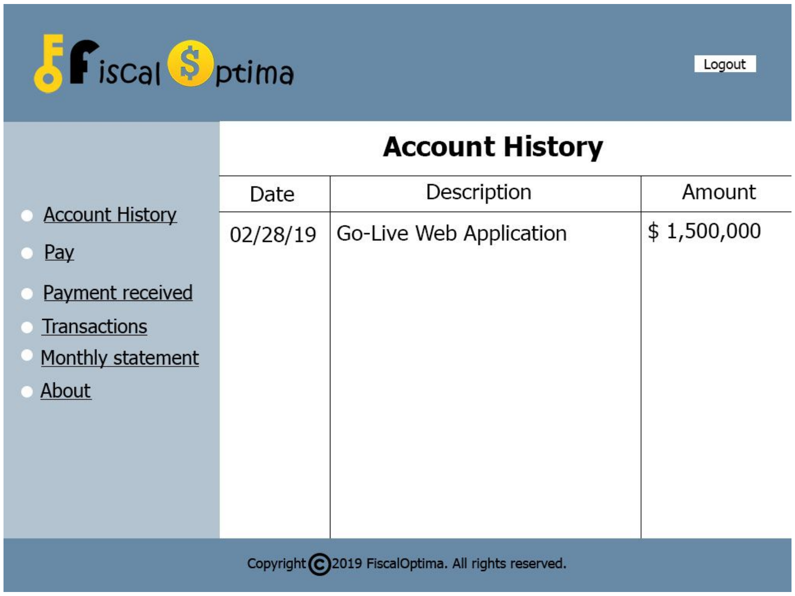
Confirm Password:

Copyright © 2019 FiscalOptima. All rights reserved.

Figure 1

In Figure 2, this interface shows menu options that allows the user to access their account history, pay, payment received, transactions, monthly statement, etc. Account history shows combination of all transactions that were made in the past. As figure 2, demonstrate the details of how the account history page. For example, in figure 2 we see the description of the transaction, when it was processed and how much money it cost.

Figure 2



Account History

Date	Description	Amount
02/28/19	Go-Live Web Application	$ 1,500,000

Copyright © 2019 FiscalOptima. All rights reserved.

Pay option will let the user make a payment towards all expenses and investments. Therefore, the buyer will be able to see all his transactions that were processed into the account. The user can also receive money from other businesses or private parties; payment received will result all the deposit and refunds made into the account. Transactions can be modified or cancelled throughout the transactions option on the menu. Transaction interface will have add, modify, and delete options, see figure 3. Also each transaction will be described with 6 different attributes:

Date: which is the day that the transaction was made.

Sold to: to which company or private customer was the substance sold to.

Description: illustration of the products being sold.

Quantity: the amount of the articles being sold.

Price/Qty: how much did it cost the customer to buy one piece of the products.

Total Price: is the total amount that the customer paid for the total substances.

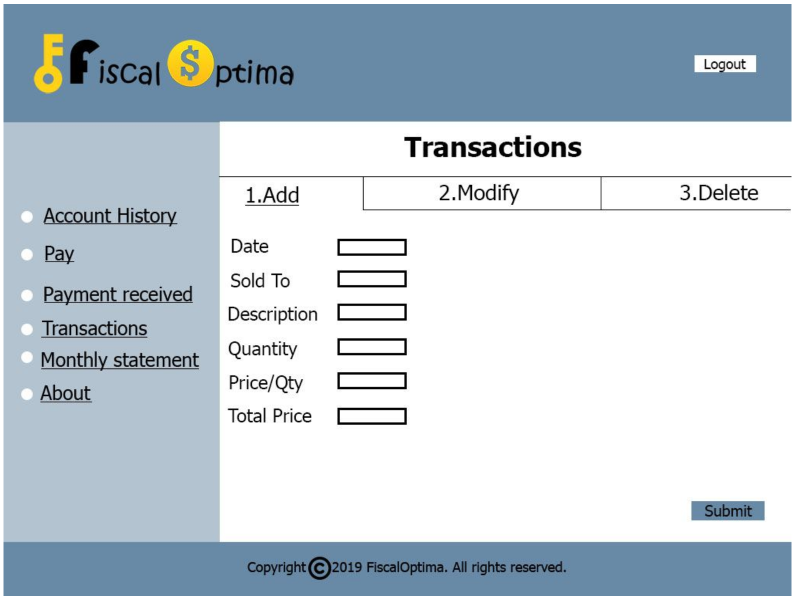


Figure 3

By the end of each month the account owner will be permitted to generate monthly statement in order for him to see all transactions in the past month. Last but not least, there is also an about option that provides users to learn more about our web application and how it can improve managing their business or multiple businesses.

4.2. Hardware Interface

The web application that we are developing will not have designated hardware. Therefore, there wouldn't be any hardware interface in this project. The connection to the database server is managed by the underlying operating system on the system/server.

4.3. Software Interfaces

The web application that we are developing will mainly be using database to store all the information of the users. The user can update the database by editing, adding, or deleting a transaction without knowing the logic behind it. The communication between the database and the web portal consists of operation concerning both reading and modifying the data.

4.4. Communication

The communication between the different part of the system is very important for the successful execution of the web application. However, the entire communication process is handled by the operating system and therefore we do not have to worry about how the web portal communicates with different programs or different parts of itself for execution.

5. Other Nonfunctional Requirements

5.1. Performance Requirements

The requirements in this section provide a detailed specification of the user interaction with the software and measurements placed on the system performance.

5.1.1. Prominent menu options

TITLE: Prominent menu options.
DESC: The menu options should be prominent and easy to find for the user.
RAT:In order to for the user to find the application featured he wants to use easily.
DEP: none

5.1.2. Usage of the accounting feature

TITLE: Usage of the accounting feature.
DESC: The accounting option should be a part of the menu and it should be easily found by the user. It should be easy to understand as well.
RAT: In order to for a user to look into everyday business transactions.
DEP: none

5.1.3. Usage of the payable feature

TITLE: Usage of the payable feature.
DESC: The payable option should be easy for the user to locate and should be easy to use.
RAT: In order to for a user to pay the dues at the right time..
DEP: none

5.1.4. Usage of the receivable feature

TITLE: Usage of the receivable feature.
DESC: The receivable feature should be easy to locate by the user and the usage of it should be easy and simple.
RAT: In order to for a user to update his total income.
DEP: none

5.1.5. Usage of the invoice generator feature

TITLE: Usage of the invoice generator feature.
DESC: The invoice generator option should be easy for the user to locate. This option should be right next to a business transaction to easily generate a invoice.
RAT: In order to for a user to generate an invoice.
DEP: none

5.1.6. Usage of the monthly statement feature

TITLE: Usage of the monthly statement feature.
DESC: The monthly statement option should be easy for the user to locate and it should display the monthly statements.
RAT: In order to for a user to pay the dues at the right time.
DEP: none

5.2. Security Requirements

To ensure “FiscalOptima” is secure and all the information stays intact, our application has a session timeout feature. The security requirement definitely needs to be included because this application plans to include a lot of user’s private financial information.

The login feature should be secure. The users cannot login through the admin login page or an admin cannot login through the user login page. Users/Admin cannot login with incorrect email or password.

5.3. Software Quality Attributes

5.3.1. Reliability

The application that is being developed needs to be reliable.The reliability depends upon how accurate the application features work. We plan on having our application 100% accurate because numbers play a huge role in business. A small mistake caused during calculations will result in huge loss for the company.

5.3.2. Availability

The final version of the application will be hosted online and will require internet connection for the user to access the web application. The internet will also be required for the application to access the database to retrieve the information of the user.

5.3.3. Maintainability

The first version which is being developed should be done in a manner that will facilitate application extendibility. This means that the code should be written such that it is re-usable while extending the web application by adding new features or while updating the existing features.

6. Other Requirements

This section includes the requirements that satisfy all the fundamental functional actions of the software system, which is the user view for the web application.

6.1. Functional Requirement 1

TITLE: User Registration in the Web Application
DESC: Given that the user has viewed the web application and wants to register for the web application (in other words, wishes to make an account on the website). The user must provide name, business firm name, password, email and an active phone number. After filing in all the fields, user can click on signup button to complete the registration process.
RAT: In order for a user to register on the web application.
DEP: None

6.2 Functional Requirement 2

TITLE: User LogIn in the Web Application
DESC: Assuming that the user has already registered for the application, his/her user email and password are stored in the database. Everytime user tries to logs in their account, a verification query would be sent to database. If the query returns true, user can view his account. If false, user needs to type in the password again.

RAT: In order for a user to login on the web application.
DEP: None

6.3 Functional Requirement 3

TITLE: Web Application Feature Search
DESC: After verification of username and password, user is given access to the web page which lists a table of features available for the website. This table would be listed as a list of hyperlinks on the left side of the page. The feature user clicks on gets accessed in the right side of the window.
RAT: In order to access a feature provided by the web application.
DEP: None

6.4 Functional Requirement 4

TITLE: Web Application - Switch View feature
DESC: Let us assume that user is active on a single page, but after user is done accessing that feature, the user can click on the hyperlinks provided on the left hand of the web application. Clicking on any of the hyperlink will generate a new view of the application for the user.
RAT: In order to switch the view, the user can click on the hyperlink in the left side of the application.
DEP: None

6.5 Functional Requirement 5

TITLE: Web Application - Sorting and Filtering the results
DESC: When viewing the website, a user enters a particular transaction.The database running in the background filters the generated results and sorts them so the database when generating a query can send the query back in the desired link. For example, an entered transaction is a payment made then the query is going to be saved in the “paid” section of the database. Whereas, if a transaction is about a received payment then it goes in the database in the “received” section of the database.
RAT: In order for a user to view the accurate result, it sorts and filters the data.
DEP: None

7. Project Plan

This section consists of the Gantt chart and the project schedule for the project

Gantt Chart Gantt chart is a graphical representation of the time taken to work on each part of the project. The horizontal bar is used to represent the duration it takes to complete working on each task of the web application. The longer the horizontal bar, the longer it takes to complete.

Spring 2019	January	February	March	April	May
1.Initiate Project					
1.1. Initial Project Document	🟩				
1.2. Statement of Work		🟩			
2. Plan the Project					
2.1. Develop Work plan		🟩			
2.2. Requirements document		🟩			
2.3. Finalize Project plan and Gain Approvals		🟩	🟩		
3. Execute and Control					
3.1. Design Framework and workflow diagrams			🟩		
3.2. Build the framework			🟩	🟩	
3.3 Test the framework				🟩	
3.4. Implement the framework				🟩	🟩
4. Close the Project					🟩

7.1 Project Schedule

The project schedule is a table which contains all the set milestones, task dependencies, task start date, and task duration

Web Application		Effort Estimate in days	Planned Start date	Planned End date
1	**Initiate Project**			
1.1	Develop Project Charter			
1.1.1	Initial Project Document	7	1/24/2019	1/31/2019
1.1.2	Define the scope	2	2/1/2019	2/3/2019
1.1.3	Hold review meetings	0.5	2/4/2019	2/4/2019
1.1.4	Finalize the project and gain approvals.	0.5	2/4/2019	2/4/2015
1.1.5	Statement of Work	4	2/5/2019	2/8/2019
2	**Plan Project**			
2.1	Develop Work Plan			
2.1.1	Develop Project Schedule	3	2/9/2019	2/12/2019
2.1.2	Research and Design the application	3	2/13/2019	2/16/2019
2.1.3	Communication Plan	1	2/17/2019	2/17/2019
2.1.4	Quality Management Plan	1	2/18/2019	2/18/2019
2.2	Requirements Document	10	2/19/2019	3/1/2019
2.3	Finalize Project Plan and Gain Approvals	4	3/1/2019	3/5/2019
3	**Execute and Control Project**			
3.1	Workflow Diagrams	5	3/6/2019	3/11/2019

3.1.1	Define framework stages and activities	2	3/12/2019	3/14/2019
3.1.2	Design framework content formats	1	3/14/2019	3/15/2019
3.2	Build the framework			
3.2.1	Write the framework content	20	3/16/2019	4/6/2019
3.2.2	Review framework content for quality	4	4/6/2019	4/10/2019
3.2.3	Build web tool prototype	10	4/10/2019	4/20/2019
3.3	**Testing the case**			
3.3.1	Test usability web tool and content	5	4/21/2019	4/26/2019
3.3.2	Fix bugs (if any)	4	4/27/2019	5/1/2019
3.4	**Implementation of the framework**	2	5/1/2019	5/3/2019
3.4.1	Presentation of application	4	5/7/2019	5/9/2019
3.4.2	Conduct project review	1	5/9/2019	5/9/2019
4	**Close the Project**	1	5/10/2019	5/10/2019