|  |  |
| --- | --- |
| UOG FINAL BLUE NEW TAG | **Department of Software Engineering** |
|  |
| University of Gujrat |

**Denouement**



**Session: BSSE Spring 2017-2020**

**Project Advisor: Mr. Saqib Rasool**

**Submitted by**

|  |  |
| --- | --- |
| *Muhammad Mansha Aazar* | *17074198-100* |
| *Uzair Ahmed* | *17074198-107* |
| *Zia Ul Mustafa* | *17074198-059* |

Department of Software Engineering

University of Gujrat

**STATEMENT OF SUBMISSION**

This is certified that **Muhammad Mansha Aazar** Roll No. **17074198-100**, **Uzair Ahmed** Roll No. **17074198-107** and **Zia-ul-Mustafa** Roll No. **17074198-059** has successfully completed the final year project named as **Denouement** at the department of Software Engineering, University of Gujrat, to fulfill the requirement of the degree of **BS in Software Engineering**.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project Supervisor Project Management Office

Faculty of C&IT Evening-UOG

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Coordinator Evening/HOD

**Acknowledgement**

We truly acknowledge the cooperation and help made by **Sir Ikram-ul-Haq**, HOD/Coordinator, C&IT Evening Program, University of Gujrat. He has been a constant source of guidance throughout the course of this project. We would also like to thank **Mr. Saqib Rasool** for his help and guidance throughout this project. We are also thankful to our friends and families, whose silent support led us to complete our project.

1. Muhammad Mansha Aazar
2. Uzair Ahmed
3. Zia-ul-Mustafa

Date: January 18, 2021

**Abstract**

Continuous Integration is a software development technique where developers integrate their work then it is verified by an automated system. Grady Booch first proposed the term CI in 1991. A Continuous Deployment is also a development approach in which functionalities are delivered frequently through automated deployments. CI-CD provides benefits like accelerated time to market, building the right product, improve productivity & efficiency, reliable releases and improve product quality.

DevOps provides automation for these practices. To build a DevOps approach, the required tools are Repository Management Tools, Continuous Integration Tools, Continuous Deployment Tools, Builds Tools and Monitoring Tools. So, they have to hire a new team (i.e. DevOps engineers) to do this stuff. Further DevOps engineers should have Functional Skills, Decision Making Skills, Testing Skills and Advisory Skills. Mostly DevOps cycle is for particular projects using open source tools. This cycle is time-consuming for small companies or group of developers.

Here, we tried to overcome this situation by providing a platform where they can build and deploy their projects having zero or minimum knowledge about DevOps. We limited the scope and developed this platform only for JavaScript apps.

Table of Contents

[Chapter 1: Project Feasibility Report 8](#_Toc71551076)

[1.1 Introduction 8](#_Toc71551077)

[1.2 Project/Product Feasibility Report 8](#_Toc71551078)

[1.2.1 Technical Feasibility 8](#_Toc71551079)

[1.2.2 Operational Feasibility 8](#_Toc71551080)

[1.2.3 Economic Feasibility 9](#_Toc71551081)

[1.2.4 Schedule Feasibility 9](#_Toc71551082)

[1.2.5 Specification Feasibility 9](#_Toc71551083)

[1.2.6 Information Feasibility 10](#_Toc71551084)

[1.2.7 Motivational Feasibility 10](#_Toc71551085)

[1.2.8 Legal & Ethical Feasibility 10](#_Toc71551086)

[1.3 Project/Product Scope 10](#_Toc71551087)

[1.4 Project/Product Costing 11](#_Toc71551088)

[1.4.1 Project Cost Estimation by Function Point Analysis 11](#_Toc71551089)

[1.4.2 Project Cost Estimation by using COCOMO’81 (Constructive Cost Model) 13](#_Toc71551090)

[1.4.3 Activity Based Costing Calculating Value of M: 15](#_Toc71551091)

[1.5 Task Dependency Table 16](#_Toc71551092)

[1.6 CPM - Critical Path Method 16](#_Toc71551093)

[1.7 Gantt chart 17](#_Toc71551094)

[1.8 Introduction to Team member and their skill set 18](#_Toc71551095)

[1.8.1 Uzair Ahmed 18](#_Toc71551096)

[1.8.2 Muhammad Mansha Aazar 18](#_Toc71551097)

[1.8.3 Zia Ul Mustafa 19](#_Toc71551098)

[1.9 Task and Member Assignment Table 20](#_Toc71551099)

[1.9.1 Task durations and dependencies 21](#_Toc71551100)

[1.10 Tools and Technology with reasoning 21](#_Toc71551101)

[1.10.1 Why Node 21](#_Toc71551102)

[1.10.2 Why React 22](#_Toc71551103)

[1.10.3 The development processes. 22](#_Toc71551104)

[1.11 Vision Document 22](#_Toc71551105)

[1.12 Risk List 24](#_Toc71551106)

[1.13 Product Features/ Product Decomposition 24](#_Toc71551107)

[Chapter 2: Software Requirement Specification and Design Document 25](#_Toc71551108)

[2.1 Introduction 25](#_Toc71551109)

[2.2 Entity Relationship Diagram 25](#_Toc71551110)

[2.3 Data Flow Diagram (Functional Model) 26](#_Toc71551111)

[2.3.1 Level 1 DFD 26](#_Toc71551112)

[2.3.2 Level 2 DFD 26](#_Toc71551113)

[2.4 State Transition Diagram (Behavioral Model) 27](#_Toc71551114)

[2.5 Architecture Design 28](#_Toc71551115)

[2.6 Component Level Design 29](#_Toc71551116)

[Chapter 3: User Interface Design 30](#_Toc71551117)

[3.1 Introduction 30](#_Toc71551118)

[3.2 Site Maps 30](#_Toc71551119)

[3.3 Story boards 30](#_Toc71551120)

[3.4 Navigational Maps 30](#_Toc71551121)

[3.5 Trace-ability Matrix 30](#_Toc71551122)

[Chapter 4: Software Testing 31](#_Toc71551123)

[4.1 Introduction 31](#_Toc71551124)

[4.2 Test Plan 31](#_Toc71551125)

[4.2.1 Purpose 31](#_Toc71551126)

[4.2.2 Outline 31](#_Toc71551127)

[4.3 Test Design Specification 32](#_Toc71551128)

[4.3.1 Purpose 32](#_Toc71551129)

[4.3.2 Outline 32](#_Toc71551130)

[4.4 Test Case Specification 33](#_Toc71551131)

[4.4.1 Purpose 33](#_Toc71551132)

[4.4.2 Outline 33](#_Toc71551133)

[4.5 Test Procedure Specification 34](#_Toc71551134)

[4.5.1 Purpose 34](#_Toc71551135)

[4.5.2 Outline 34](#_Toc71551136)

[4.6 Test Item Transmittal Report 34](#_Toc71551137)

[4.6.1 Purpose 34](#_Toc71551138)

[4.6.2 Outline 34](#_Toc71551139)

[4.7 Test Log 35](#_Toc71551140)

[4.7.1 Purpose 35](#_Toc71551141)

[4.7.2 Outline 35](#_Toc71551142)

[4.8 Test Incident Report 35](#_Toc71551143)

[4.8.1 Purpose 35](#_Toc71551144)

[4.8.2 Outline 35](#_Toc71551145)

[4.9 Test Summary Report 35](#_Toc71551146)

[4.9.1 Purpose 35](#_Toc71551147)

[4.9.2 Outline 36](#_Toc71551148)

# Chapter 1: Project Feasibility Report

## Introduction

This chapter is all about project introduction, project development criteria, project resources and costing.

## Project/Product Feasibility Report

Any developer with or without the knowledge of such complex tools can automate the process of building and deployment of their application. This platform also provides containerization and orchestration for their apps. Our platform first builds the application then do containerization (build app container-image) and then orchestration (deploy on Kubernetes).

There are many types of feasibility:

* + - Technical
    - Operational
    - Economic
    - Schedule
    - Specification
    - Information
    - Motivational
    - Legal and Ethical

### Technical Feasibility

Denouement is basically a web application which build the user application, build the image of that application which is called containerization and then deploy it on Kubernetes which is an orchestration tool. Initially developers of JavaScript language users can interact with our application.

### Operational Feasibility

The end users of the project can easily use our application. They do not need any technical expertise of CI-CD to operate our application. Its Interface is user-friendly.

System Functionalities and how our system provides solution

1. Clone user code from version control system
2. install the all dependencies of application
3. Build the container Image
4. Build the configuration files in YAML format for Kubernetes
5. Deploy it on Kubernetes

### Economic Feasibility

Denouement is an Application that can provide you the quick way of building and deploying the user application on Kubernetes. Our Project provides many benefits economically.

#### Cost Estimation

The overall estimated cost for out project Denouement for is 530,000 PKR (approx.).

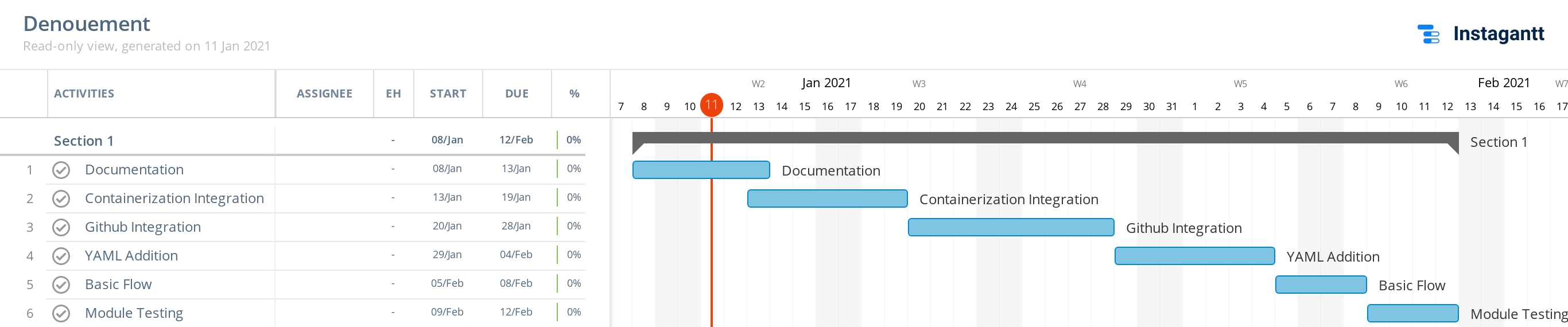
#### Tangible Benefits

This application is free of cost. Late we will decide about such benefits.

#### Intangible Benefits:

We will provide best services to the end user of the application. If application end user finds any bug or difficulty in the system UI or functionality we will launch new version of our Application with bug fix and more user-friendly approach.

### Schedule Feasibility

In our project Denouement, we have defined deadlines at different stages to complete our project on time. we have started our project from 11 jan-2021 and the final completion date for the project is 20 May-2021. Between these days/months we have define different deadlines to complete our project on time. we have also created a Gantt chart in which you can see that at which date which task of our project is going to be complete.

### Specification Feasibility

The requirements of Denouement are clear and definite. Denouement is an Application that can provide you the quick way of building and deploying. In Denouement we have different types of functions and features that are:

* + - * Login
      * User will provide required input for the deployment of their application
      * use have to perform a sql based query
      * he will get a very structured response of system it will be in pretty form
      * User can publish its app on kubernetes

### Information Feasibility

The information we are providing about our project is completed. Reliable and meaningful.

### Motivational Feasibility

According to the Research Paper **Software Project Management and Its Tools in Practice in IT Industry of Pakistan,** In Pakistan more than 50% Software houses using agile methodologies. That welcomes the changes which leads to continuous deployments. So, our platform is good choice to deploy their apps.

### Legal & Ethical Feasibility

We are developing this project for developers. We want to provide rapid solution of such complex layers of automation.

## Project/Product Scope

By using our platform any developer with or without the knowledge of such complex tools can automate the process of building and deployment of their application. This platform also provides containerization and orchestration for their apps. Our platform first builds the application then do containerization (build app container-image) and then orchestration (deploy on Kubernetes).

our Application will provide:

* + - * Login
      * User will provide required input for the deployment of their application
      * user h have to perform a SQL based query
      * he will get a very structured response of system it will be in pretty form

## Project/Product Costing

### Project Cost Estimation by Function Point Analysis

#### Project Costing:

Project Cost Estimation by Function Point Analysis following are five information domain characteristics are determined and counts are provided in here as:

Count of total external Inputs = **10**

Count of total external outputs = **05**

Count of total User inquiries = **05**

Count of total internal logical files = **10**

Count of total external interface files = **01**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Information domain value** | **Low values** | **Average Values** | **High Values** | **FP Count** |
| **No. of external input** | 10 x 3 = 30 | 3 x 4 = 12 | 11 x 6 = 66 | 108 |
| **No. of external output** | 5 x 4 = 20 | 8 x 5 = 40 | 8 x 7 = 56 | 116 |
| **No. of user inquiry** | 5 x 3 = 15 | 5 x 4 = 20 | 1 x 6 = 6 | 41 |
| **Internal logical files** | 10 x 7 = 70 | 2 x 10 = 20 | 3 x 15 = 45 | 135 |
| **External interface files** | 1 x 5 = 5 | 2 x 7 = 14 | 7 x 10 = 70 | 89 |

The following table will give the detailed calculation for total count

|  |  |  |
| --- | --- | --- |
| **1.** | Data communications | 5 |
| **2.** | Distributed data processing | 4 |
| **3.** | Performance | 4 |
| **4.** | Heavily used configuration | 4 |
| **5.** | Transaction rate | 3 |
| **6.** | On-Line data entry | 1 |
| **7.** | End-user efficiency | 4 |
| **8.** | On-Line update | 3 |
| **9.** | Complex processing | 5 |
| **10.** | Reusability | 5 |
| **11.** | Installation ease | 5 |
| **12.** | Operational ease | 4 |
| **14.** | Facilitation change | 3 |

General System Characteristic Value (0-5)

Total = ΣFi = 50

#### Calculate Function Point:

FP est. = Count Total x [0.65 + 0.01(ΣFi)]

FP est. = 489 x [0.65 + 0.01 x (50)]

FP est. = 489 x [0.65 + 0.52] FP est. = 489 x 1.17

FP est. = 562.35

#### For our project

Average Productivity = 32 FP/Per Month

Labor Rate= 16400 / Per Month

Total Estimated Effort = FP est./Productivity

Total Estimated Effort = 562.35/32

Total Estimated Effort = 17.57 pm

#### Now,

Cost/FP = Labor Rate/Productivity

Cost/FP = 16400/32

Cost/FP = 512.5Rs/FP

Total project cost = FP est.\*(Cost/FP)

Total project cost = 562.35 x 512.5

Total project cost = 288204.375

### Project Cost Estimation by using COCOMO’81 (Constructive Cost Model)

#### Introduction:

**Basic:** Is used mostly for rough, early estimates.

**Intermediate:** Is the most commonly used version, includes 15 different factors to account for the influence of various project attributes such as personnel capability, use of modern tools, hardware constraints, and so forth.

**Detailed:** Accounts for the influence of the different factors on individual project phases: design, coding/testing, and integration/testing. Detailed COCOMO is not used very often.

Each level includes three software development types:

1. **Organic:** Relatively small software teams develop familiar types of software in an in-house environment. Most of the personnel have experience working with related systems.
2. **Embedded:** The project may require new technology, unfamiliar algorithms, or an innovative new method
3. **Semi-detached:** Is an intermediate stage between organic and embedded types.

#### Basic COCOMO

|  |  |  |
| --- | --- | --- |
| **Type** | **Effort** | **Schedule** |
| Organic | PM = 2.4 (KLOC)1.05 | TD = 2.5(PM)0.38 |
| Semi-Detached | PM = 3.0 (KLOC)1.12 | TD = 2.5(PM)0.35 |
| Embedded | PM = 2.4 (KLOC)1.20 | TD = 2.5(PM)0.32 |

**PM** = person-month (effort)

**KLOC** = lines of code, in thousands

**TD** = number of months estimated for software development (duration)

**(We Supposed KLOC = 80)**

**Organic:**

**PM:** 2.4 × 801.05 = 239.031

**TD:** 2.5 × 239.0310.38 = 20.033

**Semi-Detached:**

**PM:** 3.0 × 801.12 = 406.05

**TD:** 2.5 × 406.050.35 = 20.46

**Embedded:**

**PM:** 2.4 × 801.20 = 461.23

**TD:** 2.5 × 461.230.32 = 17.79

#### Intermediate COCOMO

|  |  |
| --- | --- |
| **Type** | **Effort** |
| Organic | PM= 2.4 (KLOC)1.05 x M |
| Semi-Detached | PM= 3.0 (KLOC)1.12 x M |
| Embedded | PM= 2.4 (KLOC)1.20 x M |

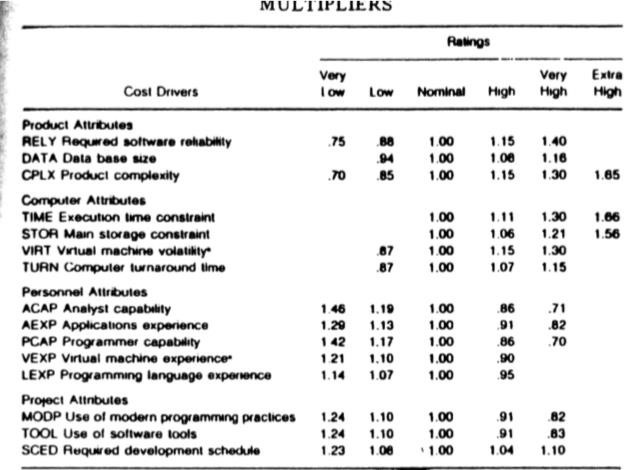
**PM** = person-month

**KLOC** = lines of code, in thousands

**M**: - reflects 15 predictor variables, called cost drivers

#### We Have Selected Organic Mode for Our Project.

### Activity Based Costing Calculating Value of M:



**M =** 1.00 × 0.94 × 0.85 × 1.00 × 1.00 × 0.87 × 1.00 × 1.19 × 0.91 × 0.86 × 1.10 × 0.95 × 0.91 × 0.91 × 1.00

#### M = 0.56

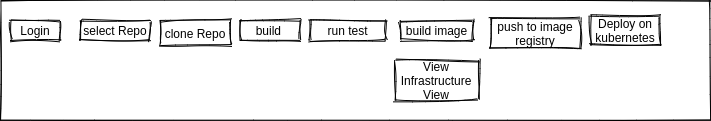
**PM** =2.4 (KLOC)1.05 × M=2.4 × 801.05 × 0.56 = 133.85

**TD(Schedule) =** 2.5 × 133.850.38 = 16.071≈ 17 Days

#### The schedule is determined using the Basic COCOMO schedule equations.

**People Required = =** ≈ 8 people

## Task Dependency Table



## CPM - Critical Path Method

|  |  |  |
| --- | --- | --- |
| **Task** | **Immediate Predecessor** | **Duration** |
| **A.** Requirement’s Gathering / Analysis & Software Design Diagrams | **None** | **2 weeks** |
| **B. Configuration generator** | **A** | **1 Weeks** |
| **C.** Queries View | **A** | **2 Weeks** |
| **D.** Injecting Configuration Files | **A** | **2 Weeks** |
| **E.** Infrastructure files | **B, C, D** | **4 Weeks** |
| **F.** Running on Denouement Platform | **B, C, D** | **4 Weeks** |
| **G. Documentation** | **F** | **2 Weeks** |

**The critical path is:** **A, B, E, F, G**

## Gantt Chart

## Introduction to Team member and their skill set

A brief but a concise introduction of the team members should be provided signifying their skill set. This skill set would especially be representative of the tasks and activities assigned to him.

In Our Team there are three members:

1. Uzair Ahmed
2. Muhammad Mansha Aazar
3. Zia Ul Mustafa

### Uzair Ahmed

I am Uzair Ahmed. I can develop web and android applications. I can do documentation for project.

#### Skill Set Platforms:

1. Windows
2. Visual Code
3. SQL Server Management Studio

#### Programming Languages & Designing:

1. Html, CSS, JavaScript, Java.
2. Design Diagrams
3. Software Costing
4. Software Testing
5. Requirement Gathering
6. React
7. Gatsby
8. Docker

### Muhammad Mansha Aazar

I am Muhammad Mansha Aazar. I can develop mobile applications in Android and web applications in Nodejs.

#### Skill Set Platforms:

1. Visual Code
2. MySQL Workbench
3. Microsoft SQL Management studio
4. Linux
5. Windows
6. Docker
7. Kubernetes

#### Programming Languages & Designing:

1. C++
2. Html, CSS, JavaScript
3. Nodejs
4. SQL, Database Management
5. Interface Designing
6. Design Diagrams
7. React

### Zia Ul Mustafa

I am Zia Ul Mustafa. I can develop web application in React. I have also developed hotel management system and many other small projects.

#### Skill Set Platforms:

1. Visual Code
2. Windows

#### Programming Languages & Designing:

1. Html, CSS, JavaScript, Java
2. Design Diagrams
3. Software Costing
4. Software Testing
5. React

|  |  |  |
| --- | --- | --- |
| *Members Name* | *Skill Set* | *Tasks* |
| Muhammad Mansha Aazar | ***Platforms:***   1. Visual Code 2. SQL Server Management Studio 3. Containerization 4. Orchestration   ***Programming Languages***   * 1. C++   2. Html, CSS, JavaScript  1. SQL, Database Management 2. Interface Designing 3. Design Diagrams 4. Nodejs | Software Designing  Interface Designing  Database Mgmt.  Software Programming  Backend Coding  Database Coding  Prototype Designing  Software Integration  Debugging |
| *Zia Ul Mustafa* | ***Platforms:***   1. Visual Code   ***Programming Languages***   * 1. Html, CSS, JavaScript,   2. Design Diagrams   3. Software Costing   4. Software Testing   5. Diagram Designing   6. Requirement Gathering | Software Costing  Software Documentation  Requirement Gathering  Software Testing  Software Debugging  Database Designing  Prototype Designing |
| *Uzair Ahmed* | ***Platforms:***  1. Visual Code   1. SQL Server Management Studio   ***Programming Languages***   * 1. Html, CSS, JavaScript, Java.   2. Design Diagrams   3. Software Costing   4. Software Testing   5. Requirement Gathering | Software Testing  Software Logic Planning  Website UI Designing  Software Quality Assurance  Requirement Gathering  Database Designing  Front End Logic |

## Task and Member Assignment Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Duration (days)** | **Dependencies** | **Members** |
| T1 | 29 |  | M1, M2 |
| T2 | 15 | T1 | M2, M3 |
| T3 | 16 | T1(M1) | M1, M2 |
| T4 | 16 | T1 | M2, M3 |
| T5 | 28 | T2, T3, T4 (M2) | M1, M2, M3 |
| T6 | 25 | T5 (M3) | M1, M3 |
| T7 | 14 | T6 (M1) | M1, M2, M3 |

### Task durations and dependencies

#### Activity Bar Chart

#### Allocation of People to Activities:

|  |  |
| --- | --- |
| **Task** | **Members** |
| T1 | M. Mansha Aazar, Uzair Ahmed, Zia Ul Mustafa |
| T2 | M. Mansha Aazar, Uzair Ahmed |
| T3 | M. Mansha Aazar |
| T4 | M. Mansha Aazar, Uzair Ahmed |
| T5 | Zia Ul Mustafa |
| T6 | Uzair Ahmed, M. Mansha Aazar |
| T7 | Zia Ul Mustafa |

## Tools and Technology with reasoning

In this Project, we are going to develop a application using Node

### Why Node

Node.js is a server-side JavaScript run-time environment. It's open-source, including Google's V8 engine, libuv for cross-platform compatibility, and a core library. Notably, Node.js does not expose a global "window" object, since it does not run within a browser.

It's an environment, and back-end frameworks run within it. Popular ones include Express.js (or simply Express) for HTTP servers and Socket.IO for WebSocket servers.

### The development processes.

What tool support is required to effectively work?

For example, if the organization decide to employ an iterative development process, it is necessary to automate the tests, since you will be testing several times during the project.

* We will use Visual Code for both Node
* Programming Languages and Tools for the development of application
* JavaScript
* Docker
* Kubernetes
* Tekton
* buildpacks
* Node

## Vision Document

Anyone wants to automate their software development process from developing, building, and deploying, they have to learn CI-CD. Which involves multiple tools like Continuous Integration Tools, Continuous Deployment Tools, Version Control System and have follow some best practices of CI-CD to achieve automation. If you are going to containerize your application, then you have to learn some more tools like containerization tools, container orchestration tools. So, to do all this stuff you have to be so focused in each automation layer. Because their working environment is totally different from each other’s. So, if you are a small development team or a large team you have to train some developers to do this stuff or hire new developers.

By using our platform any developer with or without the knowledge of such complex tools can automate the process of building and deployment of their application. This platform also provides containerization and orchestration for their apps. Our platform first builds the application then do containerization (build app container-image) and then orchestration (deploy on Kubernetes).

**Our Application will provide:**

1. Registration
2. Login
3. Dashboard
4. Repository Catalog
5. Repository Selection View
6. Form for inputs
7. User will provide required input for the deployment of their application
8. We will return file which will contain all code of their app
9. User can publish its app on our platform

**Have you fully explored what the "problem behind the problem" is?**

Yes, we have fully explored that the developers facing the problems. To automate a development process requires multiple tools and their expertise

**Is the problem statement correctly formulated?**

Yes, the problem statement is correctly formulated. Developers are facing problems because of the complexity of such tools.

**Is the list of stakeholders complete and correct?**

Yes, the list of stake holders is complete and correct

**List of Stake Holders**

1. Universities Concerned Persons
2. Users
3. Students
4. Developers
5. Development Team

**Does everyone agree on the definition of the system boundaries?**

Yes, everyone agrees on the definition of the system boundaries.

**If system boundaries have been expressed using actors, have all actors been defined and correctly described?**

Yes, all actors have been defined and correctly described

**Have you sufficiently explored constraints to be put on the system?**

Yes, we have explored constraints to be put on the system.

**Have you covered all kinds of constraints - for example political, economic, and environmental?**

Yes, we have covered all kinds of constraint.

**Have all key features of the system been identified and defined?**

Yes, all key features of the system been identified and defined

**Will the features solve the problems that are identified?**

Yes, the features solve the problem that is identified.

**Are the features consistent with constraints that are identified?**

Yes, the features are consistent with constraint that is identified.

## Risk List

Denouement has multiple risks during the development.

Following are the risks:

1. No Custom grip
2. Integration problems
3. Files Structure problems
4. Infrastructure as code file’s structure

## Product Features/ Product Decomposition

Denouement is a web application and platform which provide a simple and easy to understand automation to developers through a web system. They can easily deploy their application to Kubernetes and can enjoy the features of Kubernetes without learning it.

**Features:**

* + - 1. Login
      2. User will provide required input for the deployment of their application
      3. use have to perform a sql based query
      4. he will get a very structured response of system it will be in pretty form
      5. User can publish its app on kubernetes

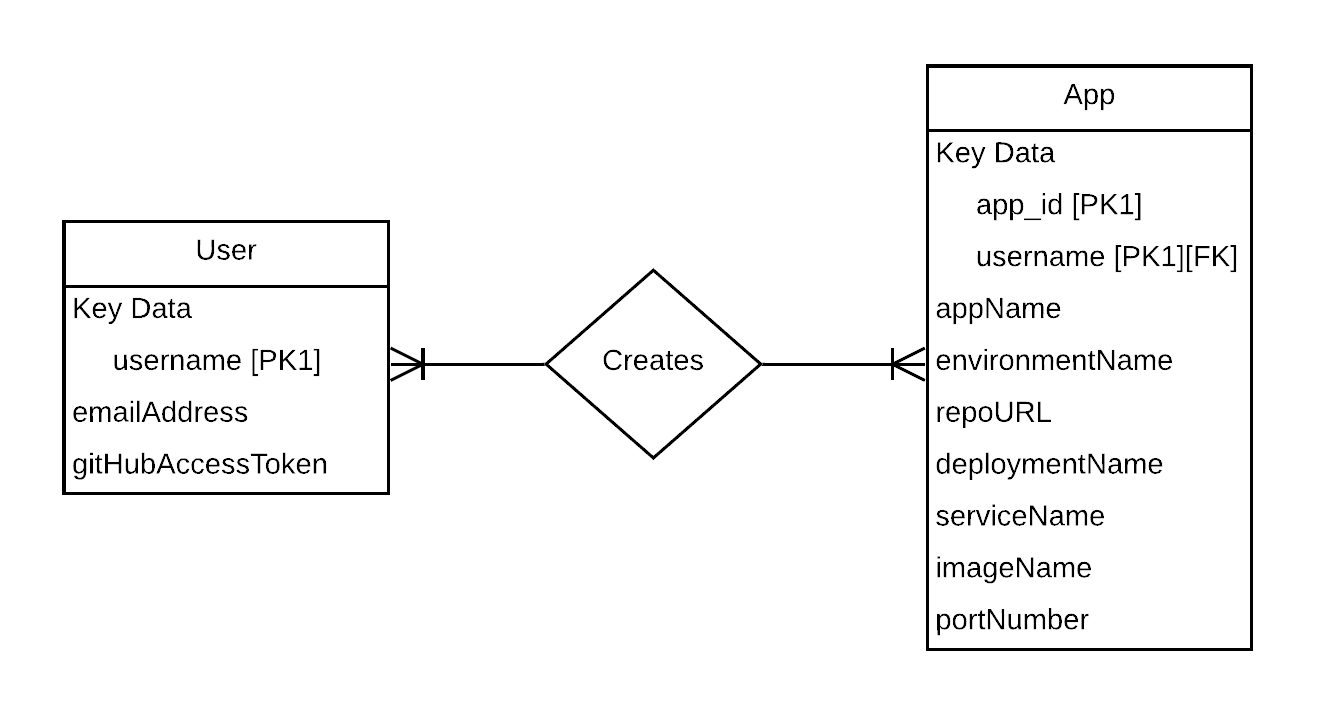
# Chapter 2: Software Requirement Specification and Design Document

## Introduction

This chapter contains the following design models of our project.

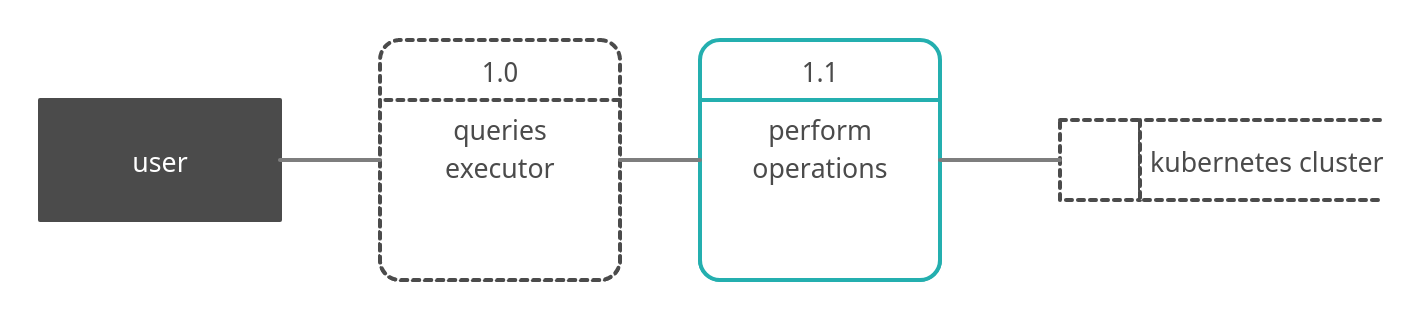
1. Entity Relationship Diagram
2. Data Flow Diagram (Functional Model)
3. State Transition Diagram (Behavioral Model)
4. Architecture Design
5. Component Level Design

## Entity Relationship Diagram

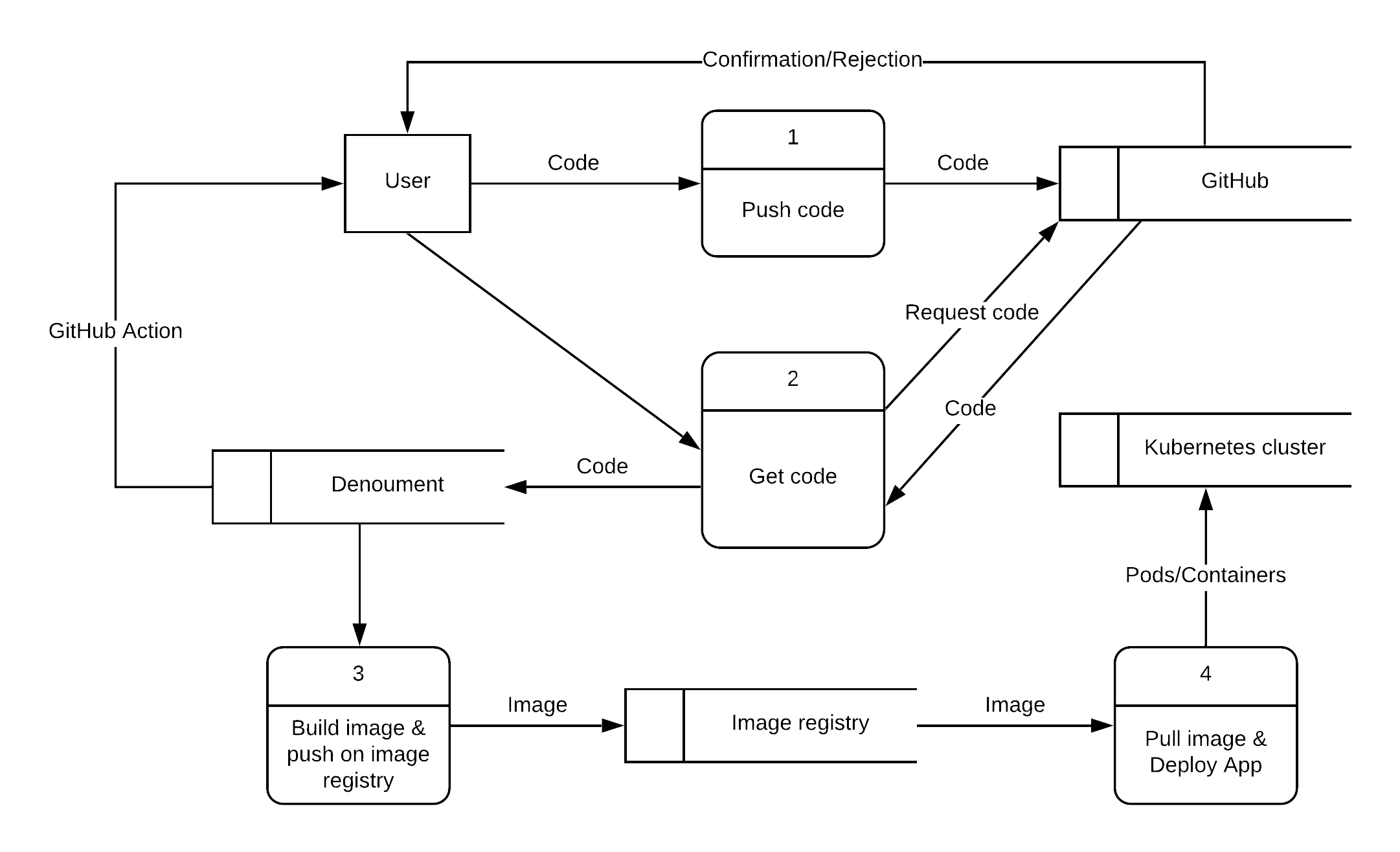


## Data Flow Diagram (Functional Model)

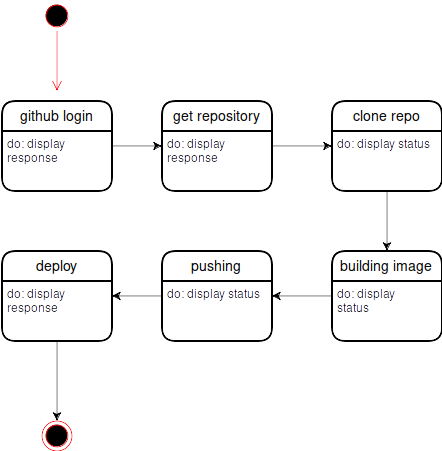
### Level 1 DFD



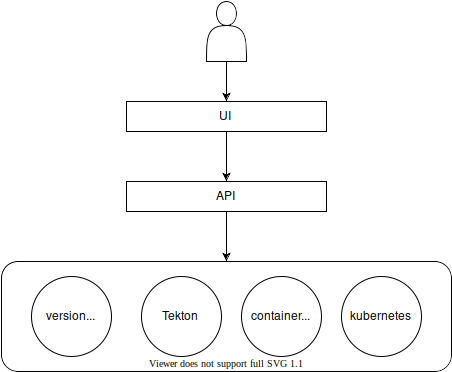
### Level 2 DFD



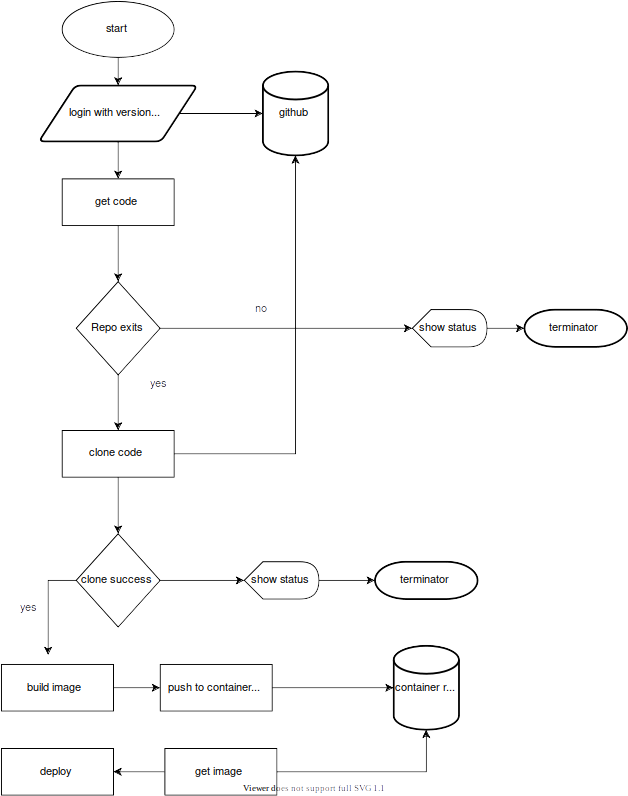
## State Transition Diagram (Behavioral Model)



## Architecture Design



## Component Level Design



# Chapter 3: User Interface Design

## Site Maps

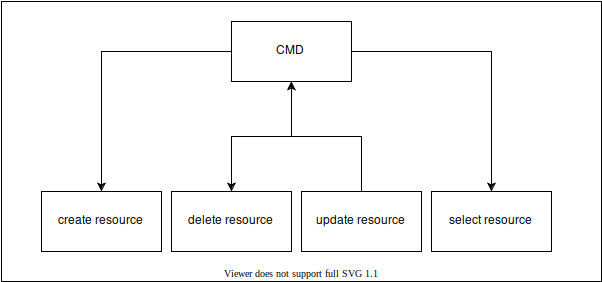


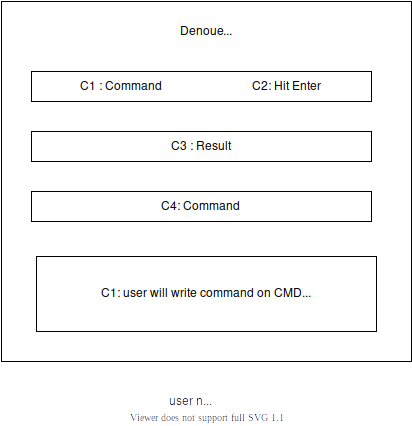
Figure 3. 1: User Site Map

## Story boards

|  |  |  |
| --- | --- | --- |
| **Denouement**     |  | | --- | | C1:Command |  |  | | --- | | C1: Command to create, POST and Delete Kubernetes Resources | |

Figure 3. 2 User Story Board

## Navigational Maps



## Trace-ability Matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SR#** | Para **#** | **System Specification** | **Build** | **Category** | **Use Case Name** |
| 1 | 1.0 | User shall create Kubernetes Resources | B1 | Business | UC\_User\_Create |
| 2 | 1.0 | User shall delete Kubernetes Resources | B1 | Business | UC\_User\_delete |
| 3 | 1.0 | User Shall update Kubernetes Resources | B1 | Business | UC\_User\_Update |
| 4 | 1.0 | User Shall be able to view Kubernetes resources | B1 | Business | UC\_User\_View |
| 5 | 1.0 | User shall be able to view help how to perform command like examples query. | B1 | Business | UC\_User\_help |
| 6 | 1.0 | User shall be able to query Kubernetes resources | B1 | Bisiness | UC\_User\_\_query |

Figure 3. 3 System Specification Tracebility matrics

# Chapter 4: Software Testing

## Introduction

Users want Command line which passed the full cycle of manual testing. it is very important to have the quality. The Test Plan has been created to facilitate communication within the team members. This document describes approaches and methodologies that will apply to the unit, integration and system testing of the " **Denouement**”. It includes the objectives, test responsibilities, entry and exit criteria, scope, schedule major milestones, entry and exit criteria and approach. This document has clearly identified what the test deliverables will Be and what is deemed in and out of scope.

Following are standard artifacts, which must be included in this deliverable:

1. Test Plan
2. Test Design Specification
3. Test Case Specification
4. Test Procedure Specification
5. Test Item Transmittal Report
6. Test Log
7. Test Incident Report
8. Test Summary Report

## Test Plan

### Purpose

* To prescribe the scope, approach, resources, and schedule of the testing activities.
* To identify:
* The items being tested
* The features to be tested
* The testing tasks to be performed
* The personnel responsible for each task
* The risks associated with this plan.

### Outline

#### Test plan identifier

DI-ASM01.

#### Introduction

This is the Master Test Plan for the Denouement . This plan will address only those items and elements that are related to the Denouement, both directly and indirectly affected elements will be addressed. The primary Focus of this plan is to ensure that the Denouement System new system provides the same level of information and detail as the current system while allowing for Improvements and increases in data acquisition and level of details available. The project will have three levels of testing, Unit, System/Integration and Acceptance. The details for each level are addressed in the approach section and will be further defined in the level specific plans. The estimated timeline for this project is very aggressive (six (6) weeks), as such, any Delays in the development process or in the installation and verification of the third party Software could have significant effects on the test plan. The acceptance testing is expected to take one (1) week from the date of application delivery from system test and is to be done in parallel with the current application process.

#### Test items

1. Command for Create K8s resources
2. Command for Update K8s resources
3. Command for Delete K8s resources
4. Command for View K8s resources

#### Features to be tested

* Command for Create K8s resources
* Command for Update K8s resources
* Command for Delete K8s resources

#### Features not to be tested

* Kubernetes

#### Approach

There are following testing types which we are going to use in our project.

|  |  |
| --- | --- |
| **Unit Testing** | Test the smallest piece of verifiable software in the website i.e., Testing of user Id. UNIT Testing will be done by the developer and will be approved by the development team leader. Proof of unit testing (test case list, sample output,) must be provided by the programmer to the team leader before unit testing will be accepted and passed on to the test person. All unit test information will also be provided to the test person. |
| **Integration Testing** | Individual software modules are combined and testing as a group i.e., Website modules integrate and then perform testing. Integration Testing will be performed by the tester and developer. No specific test tools are available for this project. Programs will enter into System/Integration test after all critical defects have been corrected. A program may have up to two Major defects as long as they do not impede testing of the program (I.E., there is a work around for the error). |
| **System Testing** | Conduct a complete, integrated system to evaluate the system’s compliance with its specified requirements. |
| **Acceptance Testing** | ACCEPTANCE Testing will be performed by the actual end users with the assistance of the tester and developer. The acceptance test will be done in parallel with the existing manual process for a period of one week after completion of the System/Integration test process Programs will enter into Acceptance test after all critical and major defects have been Corrected.  A program may have one major defect as long as it does not impede testing of the program (I.E., there is a work around for the error). Prior to final completion of acceptance testing all open critical and major defects MUST be corrected and verified by the User test representative. A limited number of Users will participate in the initial acceptance test process. Once acceptance test is complete. |
| **Regression Testing** | Regression tests are carried out regularly to ensure that the functionalities of website haven’t changed while changes were done in the source code. Functionalities of android application haven’t changed while changes were done in the source code. |

**Test Tools**

The only test tools to be used are the standard provided utilities and commands. A. Jira project program will be used as the source version for testing the features of software working properly. s The tester will meet with developer once every two weeks to evaluate progress to date and to identify error trends and problems as early as possible. These meetings will be scheduled on weeks. Additional meetings can be called as required for emergency situations.

#### Item pass/fail criteria

The test process will be completed once the initial set of Users has successfully sent in their data for a period of one week. When the team is satisfied that the data is correct the initial set of Users will be set to active and all parallel stopped for those accounts. At this point the next set of Users will begin the parallel process, if not already doing so. Only the initial set of Users must pass the data comparison test to complete the testing, at that point the application is considered live. All additional activations will be on an as ready basis. When a User is ready, and their data is verified, they will then also be activated.

#### Suspension criteria and resumption requirements

When it is obvious that next items or features are dependent on current tested item or feature then next test will be suspended. Resume tests from the beginning.

#### Test deliverables

|  |  |  |  |
| --- | --- | --- | --- |
| **S. NO** | **Deliverable Name** | **Author** | **Viewer** |
| **1.** | Test Plan | Uzair Ahmed | Mansha |
| **2.** | Functional Test Cases | Uzair Ahmed | Mansha |
| **3.** | Daily/ Weekly Status Report | Zia | Mansha |
| **4.** | Test Closer Report | Zia | Mansha |

Table4.1 Test Deliverables

#### Testing tasks

|  |  |  |
| --- | --- | --- |
| **TASK** | **ASSIGNED TO** | **STATUS** |
| Create Acceptance Plan | Uzair Ahmed | Complete |
| Create System/Integration Test Plan | Uzair Ahmed | Complete |
| Define Unit test rules and procedures | Uzair Ahmed | Complete |
| Verify Prototype of Screens | Zia | Complete |
| Verify Prototype of Reports | Zia | Complete |

Table4. 2 Testing Tasks

#### Environmental needs

The following elements are required to support the overall testing effort at all levels within Denouement:

1. Access to both the developer and tester. For development, data acquisition and testing
2. An installed and functional copy System.
3. At least one User with a working copy of this project

#### Responsibilities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Responsibilities** | **Tester** | **Dev** | **Supervisor** | **User** |
| Acceptance test documentation Execution | **X** | **X** |  | **X** |
| System/Integration test Documentation & Exec. | **X** |  | **X** |  |
| Unit test documentation & execution | **X** |  | **X** |  |
| System Design Reviews | **X** | **X** | **X** | **X** |
| Detail Design Reviews | **X** | **X** | **X** |  |
| Test procedures and rules | **X** | **X** | **X** |  |
| Test procedures and rules |  |  | **X** | **X** |
| Change Control and regression testing | **X** | **X** | **X** | **X** |

Table4. 3 Responsibilities Table

#### Staffing and training needs

It is preferred that there will be at least one full time tester assigned to the project for the system/integration and acceptance testing phases of the project. This will require assignment of a person part time at the beginning of the project to participate in reviews etc... And approximately four weeks into the project they would be assigned full time. If a separate test person is not available the developer will assume this role.

In order to provide complete and proper testing the following areas need to be addressed in terms of training.

**A**. The developers and tester(s) will need to be trained on the basic operations of the Crud operation.

**B**. User will require training on the new screens and reports.

**C**. At least developer needs to be trained on the installation and control.

#### Schedule

Time has been allocated within the project plan for the following testing activities. The specific dates and times for each activity are defined in the project plan time line. The persons required for each process are detailed in the project time line and plan as well. Coordination of the personnel required for each task, test team, development team, will be handled by the Supervisor in conjunction with the developer and tester.

A. Review of Requirements document by test team personnel (with other team members) and initial creation of Inventory classes, subclasses and objectives.

B. Development of Master test plan by tester and test with time allocated for at least two reviews of the plan

C. Review of the System design document by tester. This will provide the team with a clearer understanding of the application structure and will further define the Inventory classes, sub-classes and objectives.

D. Development of System/Integration and Acceptance test plans by tester and other essential personnel with time allocated for at least two reviews of the plans.

E. Review of the Detail design document(s) by test team personnel as required. This will provide the team with a clearer understanding of the individual program structure and will further define the Inventory classes, sub-classes and objectives.

F. Unit test time within the development process G. Time allocated for both System/Integration and Acceptance test processes.

#### Risks and contingencies

A. Limited Project staff The system staff currently has two positions. As a result of this staff shortage there may be delays in getting staff to review appropriate documents and to participate in the Acceptance test process. Should User staff become a problem, the appropriate dates for reviews and acceptance testing will slip accordingly. No attempt will be made to bypass any part of the review and testing processes. However, if acceptable to the online agency product delivery system a member if the test team may be available to act as the client’s representative for part of the Acceptance test itself. The reviews of the screens and reports must have Client participation and approval.

#### Approvals

|  |  |
| --- | --- |
| **Project Supervisor** | Mr. Saqib Rasool |
| **Project Manager** | Mr. GulShair |

**Test Cases**

* **Test case ( User can create K8s Resources)**

|  |  |
| --- | --- |
| TEST Engineer | UZAIR AHMED |
| Test case ID | TC1 |
| Dated | 20-April-2021 |
| Purpose | The purpose of this test is to verify user can successfully create Kubernetes resources |
| Pre-req | User must have the install Kubernetes and Denouement system |
| Test Data | User must have to write Command |
| Steps | 1. User will open command line. 2. User will write command. 3. Hit enter |
| Post-req | Successfully create the Kubernetes resources |
| Result | PASS |

Table4. 4 User can create K8s Resources

* **Test case( User can delete Kubernetes Resources)**

|  |  |
| --- | --- |
| TEST Engineer | UZAIR AHMED |
| Test case ID | TC2 |
| Dated | 20-April-2021 |
| Purpose | The purpose of this test is to verify user can successfully delete Kubernetes resources |
| Pre-req | User must have the install Kubernetes, Denouement system and running apps. |
| Test Data | User must have to write Command |
| Steps | 1. User will open command line. 2. User will write command to delete. 3. Hit enter |
| Post-req | Successfully delete the Kubernetes resources |
| Result | PASS |

Table4. 5 User can delete Test case

* **Test case (User can Update Kubernetes Resources)**

|  |  |
| --- | --- |
| TEST Engineer | UZAIR AHMED |
| Test case ID | TC3 |
| Dated | 20-April-2021 |
| Purpose | The purpose of this test is to verify user can successfully update Kubernetes resources |
| Pre-req | User must have the install Kubernetes, Denouement system and running apps. |
| Test Data | User must have to write Command |
| Steps | 1. User will open command line. 2. User will write command to update. 3. Hit enter |
| Post-req | Successfully update the Kubernetes resources |
| Result | PASS |

Table4. 6 User can Update Test case

**5.3 Test item Transmittal Report**

**Purpose**

Test Item Transmittal Report is a document which identifies test items by their status and location information. Purpose of this document is to identify the test items being transmitted for testing. It includes the person responsible for each item, its physical location, and its status.

**Outline**

A test item transmittal report shall have the following structure:

a) Transmittal report identifier

b) Transmitted items

c) Location

d) Status

e) Approvals

**Transmittal report identifier**

Transmittal report identifier: Transmittal-report-identifier-01

**Transmitted items**

In this section we identify the test items being transmitted. All the test items are already listed in test plan document. The groups responsible for managing, designing, preparing, executing, witnessing, checking, and resolving the test cases are developers (Uzair Ahmed, Zia, Muhammad Mansha) and supervisor (Mr. Saqib Rasool).

**Location**

We perform all the tests on transmitted items on our laptop.

**Status**

It includes the status of the test items being transmitted. All the test items that are transmitted are successfully pass. Also, we tried to solve the issues that are mentioned in our test incident report.

**Approvals**

Transmittal report will be approved by our supervisor and by PMO.

**Test log**

**Purpose**

Test Log is a document which consists of information about the test cases. Means whether the test case is Passed or Failed.

**Outline**

A test log has the following structure:

a) Test log identifier

b) Description

c) Activity and event entries

**Test log identifier is:**

Test-log-identifier-01

**Description**

This section contains the information about items being tested including their version/revision levels. It also contains the attributes of the environments in which the testing is conducted. We have tested all the test cases that we have developed. We have tested all the test cases on our laptop, and we have enough memory to perform these tests.

**Activity and event entries**

For each event, including the beginning and end of activities, we have recorded the occurrence date along with the tester.

**Execution description**

Execution description includes the information about all personnel present during the execution including testers, operators, and observers. All the group members were present during the testing. Each individual participates in testing process

**Procedure results**

All the test case that we have executed was pass successfully.

**Environmental information**

In this step we Record any environmental conditions specific to this entry (e.g., hardware substitutions). So, we have no environmental conditions in this process.

**Anomalous events**

In this step we record about unexpected event occurred during the execution of test cases. Different kind of UN expected events can occur. e.g., a view profile page display was requested, and the correct screen displayed, but response seemed unusually long. We also Record circumstances surrounding the inability to begin execution of a test procedure or failure to complete a test procedure e.g., a power failure problem. We have also identified the solution about the unexpected event.

**Test Summary report**

**Purpose**

Test summary report is a document which contains summary of test activities and final test results. After the testing cycle it is very important that you communicate the test results and findings to the project stakeholders so that decisions can be made for the software release.

**Outline**

A test summary report has the following structure:

a) Test summary report identifier

b) Summary

c) Comprehensive assessment

d) Summary of results

e) Evaluation

f) Summary of activities

g) Approvals

**Test summary report identifier**

Test summary report identifier: Test-summary-report-01

**Summary**

This section contains the information about items that are tested including their version/revision levels. It also contains the attributes of the environments in which the testing is conducted. We have tested all the test cases that we have developed. All the test cases were successfully passed during the testing procedure. We have tested all the test cases on our laptop and we have enough memory to perform these tests.

**Comprehensiveness assessment**

This involves evaluation of testing and testing process for determining the quality and effectiveness of the testing activities. The main objective here is to define any challenges encountered by the team while implementing or executing it. Our Team has face different challenges while implementing or executing testing process. For example, the main challenge that we have face is to perform all the testing activities within short time. Time management is the main challenge that we have face. All the testing process execute effectively and within the specified time.

**Summary of results**

All the Modules of Our Project that we have developed are Working Successfully.

**Summary of activities**

Testing activities that are involved in whole testing process are reading documents, Test Planning, Test case design or test case writing, Test execution, Reporting your testing progress and Meetings etc.

**Approvals**

Test summary report will be approved by our supervisor and by PMO.

**Test Summary Report:**

All Modules of Our Project Works Successfully

**User Manual**

**available in current directory named as ksql\_user\_manual.pdf**

#### Approvals

**Denouement team:**

* + - 1. Muhammad Mansha Aazar **Signature: Mansha**
      2. Uzair Ahmed **Signature: uzair**
      3. Zia Ul Mustafa **Signature: zia**

**Supervisor:**

* + - 1. Mr. Saqib Rasool **Signature:**

**Final Year Project Evaluation Form (8th Semester)**

Project Title: Denouement

Roll No. 17074198-100 Student Name: Muhammad Mansha Aazar

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Deliverables** | **Marks Distribution** | **Marks obtained** | **Remarks** |
| **Code** (Sub-Section Total: 30) | | |  |
| Complete running project in compliance with requirement and design document | 15 |  |
| Run time code modification and understanding | 15 |
| **Testing** (Sub-Section Total: 20) | | |  |
| Test Plan | 5 |  |
| Test Case Design and implementation | 15 |
| **Overall System and Documentation** (Sub-Section Total: 40) | | |  |
| Project presentation  (Including SRS and Design Document) | 15 |  |
| User Manual | 10 |
| Standard Template | 5 |
| Overall skill set | 10 |
| **Supervisor**(Sub-Section Total: 20) | | |  |
| (Meetings, project progress) |  |  |
| **Project Management Office**(Sub-Section Total: 10) | | |  |
| (Meeting Deadlines, Attending workshops) |  |  |
| **External + Internal Marks** | **90+30=120** |  |  |

**Evaluator 1 (Name, Signature& Date). \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Evaluator 2 (Name, Signature& Date). \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Final Year Project Evaluation Form (8th Semester)**

Project Title: Denouement

Roll No. 17074198-107 Student Name: Uzair Ahmed

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Deliverables** | **Marks Distribution** | **Marks obtained** | **Remarks** |
| **Code** (Sub-Section Total: 30) | | |  |
| Complete running project in compliance with requirement and design document | 15 |  |
| Run time code modification and understanding | 15 |
| **Testing** (Sub-Section Total: 20) | | |  |
| Test Plan | 5 |  |
| Test Case Design and implementation | 15 |
| **Overall System and Documentation** (Sub-Section Total: 40) | | |  |
| Project presentation  (Including SRS and Design Document) | 15 |  |
| User Manual | 10 |
| Standard Template | 5 |
| Overall skill set | 10 |
| **Supervisor**(Sub-Section Total: 20) | | |  |
| (Meetings, project progress) |  |  |
| **Project Management Office**(Sub-Section Total: 10) | | |  |
| (Meeting Deadlines, Attending workshops) |  |  |
| **External + Internal Marks** | **90+30=120** |  |  |

**Evaluator 1 (Name, Signature& Date). \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Evaluator 2 (Name, Signature& Date). \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Final Year Project Evaluation Form (8th Semester)**

Project Title: Denouement

Roll No. 17074198-059 Student Name: Zia Ul Mustafa

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Deliverables** | **Marks Distribution** | **Marks obtained** | **Remarks** |
| **Code** (Sub-Section Total: 30) | | |  |
| Complete running project in compliance with requirement and design document | 15 |  |
| Run time code modification and understanding | 15 |
| **Testing** (Sub-Section Total: 20) | | |  |
| Test Plan | 5 |  |
| Test Case Design and implementation | 15 |
| **Overall System and Documentation** (Sub-Section Total: 40) | | |  |
| Project presentation  (Including SRS and Design Document) | 15 |  |
| User Manual | 10 |
| Standard Template | 5 |
| Overall skill set | 10 |
| **Supervisor**(Sub-Section Total: 20) | | |  |
| (Meetings, project progress) |  |  |
| **Project Management Office**(Sub-Section Total: 10) | | |  |
| (Meeting Deadlines, Attending workshops) |  |  |
| **External + Internal Marks** | **90+30=120** |  |  |

**Evaluator 1 (Name, Signature& Date). \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Evaluator 2 (Name, Signature& Date). \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**