Lab File

Software Project Management

[CSE432]

DEPARTMENT

OF

COMPUTER SCIENCE AND ENGINEERING

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted To:

Dr. Misha Kakkar

Department, ASET

AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY

AMITY UNIVERSITY UTTAR PRADESH

NOIDA-201301

Software Requirements Specification

for

TEMPERATURE MONITORING AND ALERT SYSTEM

Version 1.0 approved

Prepared by- Pranav Mendiratta A2305218122

Sahil Gupta A2305218274

Aastha Tyagi A12405218008

Sourik Dutta A2305218128

Pranitha Chellamuthu Velmurugan A2305218040

Amity University, Noida

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

1. Introduction

1.1 Purpose

The government declared the following rules for manufacturing tablets/medicine in all pharmaceutical companies:

• While the manufacturer can maintain the temperature of the tablets between -40 and -30 degrees Celsius, the temperature of the tablets should never remain between -33 and -30 degrees for longer than 20 minutes at a time.
• Also, the manufacturer should maintain a log of when the cooling chamber to produce the tablets is opened.
• These rules should be implemented within a month of their declaration. Failing to comply would lead to cancellation of the pharmaceutical production permit.

These rules were declared to ensure that the medicines produced by all pharmaceutical companies were up to mark and safe for consumption by the public.

1.2 Document Conventions

- *All headings are in bold.*
- *Text will be in Times New Roman and in Italic.*
- *Font-Size for heading is set to 14 and for plain text is 11.*

1.3 Intended Audience and Reading Suggestions

This document is intended to be read by stakeholders of the project to provide an overview of the project plan as well as developers of the project. Stakeholders include the project sponsor, the users of this product and the developers of the product.

The document is organized in accordance with the IEEE SRS template and will provide an overall description of the project with regards to features, user classes, implementation constraints and assumptions followed by details of the system features in the form of Use case descriptions and diagrams and closing with Non-Functional Requirements.

1.4 Product Scope

The problem required to be solved is to manufacture a temperature monitoring device i.e., to make a device that automates the alert system to alert the people responsible for the management of cooling the chamber when the temperature reaches dangerous levels such that our products could get damaged. Because of temperature, going below the lower threshold or going above the upper threshold value will damage the tablets and medicines thus causing loss to our company.

A well-thought-out implementation of the requirements can help in subsiding the losses incurred to the pharmaceutical companies every year due to temperature variations in their storage containers by automating the process and improving their efficiency.

1.5 References

[1] Aditya Grover, Ashish Kapoor, and EricHorvitz.2015. A deep hybrid model for weather forecasting. Proceedings of the 21th ACMSIGKDD International Conference on Knowledge Discovery and Data Mining. ACM,379–386.

[2] Arne Send. 2015.Using Amazon Machine Learning to Predict the Weather. https://arnesund.com/2015/05/31/ using-amazon-machine-learning-to- predict-the weather/.

[3] C. (2018). Indoor temperature prediction in an IoT scenario. Sensors(Switzerland), 18(11), [3610] . https://doi.org/10.3390/s18113610

[4] FedericoMontori, LucaBedogni, and LucianoBononi.2017. A Collaborative Internet of Things Architecture for Smart Cities and Environmental Monitoring. IEEE Internet of Things Journal (2017).

[5] Jeffrey Burt. 2017. Machine Learning Storms into Climate Research. https://www. nextplatform.com/2017/04/18/machine-learning-storms-climate- research/.

[6] Karthik Krishnamurthy, Suraj Thapa, Lokesh Kothari and Arun Prakash, Arduino Based Weather Monitoring System, International Journal of Engineering Research and General Science, vol. 3, issue 2, ISSN 2091-2730, March–April (2015).

[7] N.Adam, V.Atluri, S.YuandY.Yesha, Efficient Storage and Management of Environmental Information, 10th NASA Goddard Conferenceon Mass Storage Systems and Technologies 19th IEEE Symposiumon Mass Storage Systems, April 15– 18, College Park, MA(2002).

[8] P. Kinney, ZigBee Technology: Wireless Control that Simply Works, Communications Design Conference Secretary of ZigBee BoDChair of ZigBee Building Automation Profile WG, 2 October (2003).

[9] R. Bhattacharjee, M. SiGiriya and P. Bhattacharyab

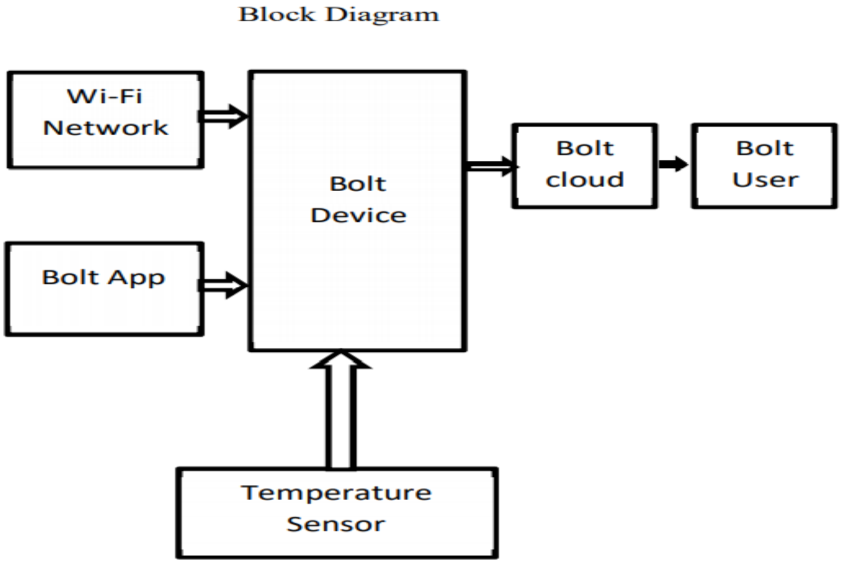
[10] Weather and Environmental Monitoring Sensors, Springer US, pp. 496– 523,(1994).

[11] Weather Forecasting using Arduino Based Cube-SatM.Rahaman Laskara

[12] Y Radhika and M Shashi. 2009. Atmospheric temperature prediction using support vector machines. International journal of computer theory and engineering 1, 1 (2009),55

2. Overall Description

2.1 Product Perspective



This product is a replacement for the current temperature measuring system at a pharmaceutical company. The system helps prevent damage to the companies product by having an accurate temperature measuring system using lm35 sensors that can record temperature in every 30 secs and send alerts in case of any anomalies. It also uses modern cloud technology to store the data that can be used to further improve the system.

2.2 Product Functions

- *The device should be able to measure accurate temperature at an interval of 30 secs and send alerts in case of any anomalies.*
- *The user should just be able to sign into his/her BOLT account from any device and then he/she should be able to view the data along with the temperature graph and also download it into an Excel file and also send it to cloud storage.*
- *The user should be able to work with it remotely by viewing data collected by the sensor on the device via our Bolt cloud accounts logged in on our devices.*

2.3 User Classes and Characteristics

- *The high level users for this product are the companies who want to keep a track of what is going on with the pharmaceutical cooling chambers and want to know if any cooling chamber is hovering over dangerous ranges which can damage the medicine.*
- *The low level and more frequent users that will be working closely with the product are the factory workers in the pharma companies who might or might not have high-school level education.*
- *The alert functionality is to be used by the caretakers of the factory who are available for maximum hours at the factory/ facility and have high school level education thus can easily operate a phone and access the alerts to take the required steps for correction.*
- *The cloud facilities added into the product are to be used by the manager and the data analyst teams of the companies that can easily work with csv data and have decent knowledge of using computers, cloud storages and analytics.*

2.4 Operating Environment

- *This device works in a specific temperature range that is from -20 C to 40 C.*
- *The minimum input is of 10 volts (for wifi modules) hence might damage the device hence wires should be thick and shock and fire resistant.*
- *Runs if high speed internet connection (at least 1 Mbps) is available for low latency and proper working.*
- *Highly temperature sensitive and low latency sensors are used in the device.*
- *Python of the latest version is required to write the program to avoid low latency and microchips should be of temperature resistant material.*
- *The data can be seen online on the user's BOLT account on BOLT cloud platform. Due to the web based nature of the system , it should be largely compatible with any device that is capable of internet access and has a web browser.*

2.5 Design and Implementation Constraints

- *Very high or very low temperatures can be there in the chamber which should be kept in mind while choosing the materials to make the device.*
- *Low latency is required in the system to avoid damage to the product which should be kept in mind while choosing programming languages.*
- *The factory has moderately fast internet speeds since it is outside the city.*
- *Education level of factory workers is not high hence an easy to use interface would be suitable.*

2.6 User Documentation

- *User guides delivered with the product-*
    - *How to interact with the interface.*
    - *Repair guide with specifications of all the parts used.*
- *All the documentation on how to access and see the data in one's bolt account is available on their website.*
- *Twilio API's message alert documentation available on their website.*

2.7 Assumptions and Dependencies

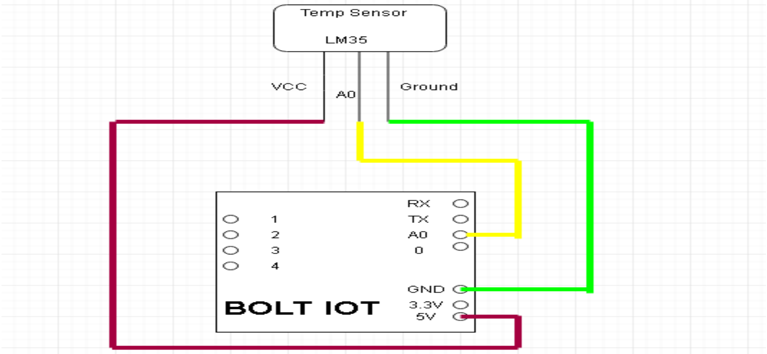
- *Temperature sensors give accurate readings.*
- *Twilio API has less than 3ms latency.*
- *Wifi modules can even work with wifi speeds less than 1mbps.*
- *BOLT cloud platform stores accurate data.*
- *Wires can withstand high voltage.*

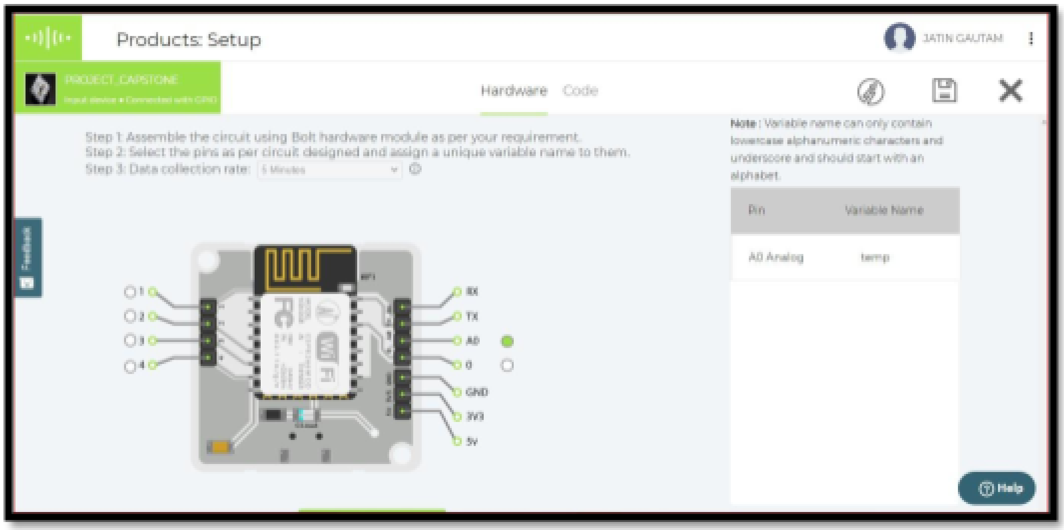
3. External Interface Requirements

3.1 User Interfaces

- Users will have access to the device through the BoLT application available on android store.

- The user will be able to monitor current temperature and will get alerts on app as well as through texts and mail in case of any anomaly.



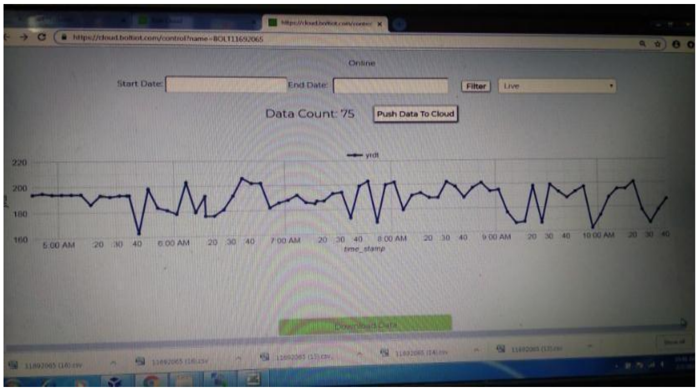


Step 1: Assemble the circuit using Bolt hardware module as per your requirement.
Step 2: Select the pins as per circuit designed and assign a unique variable name to them.
Step 3: Data collection rate: 5 Minutes

Note : Variable name can only contain lowercase alphanumeric characters and underscore and should start with an alphabet.

Pin	Variable Name
A0 Analog	temp

- Graphs- for the recorded data can be viewed by logging into BoLT accounts and predictions and predictions for recordings can also be viewed.



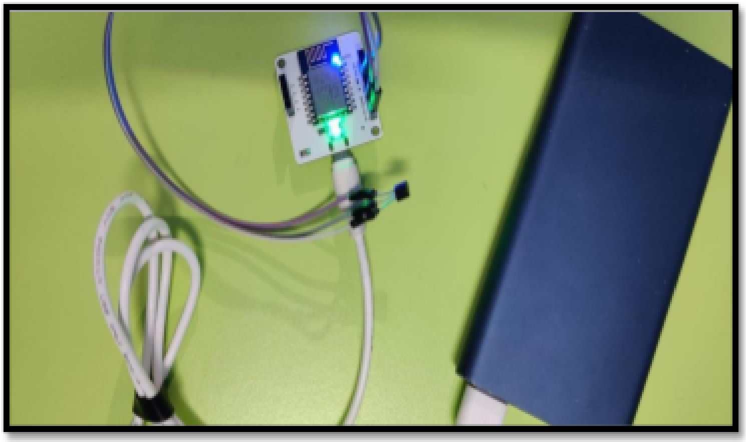
Start Date: End Date: Live

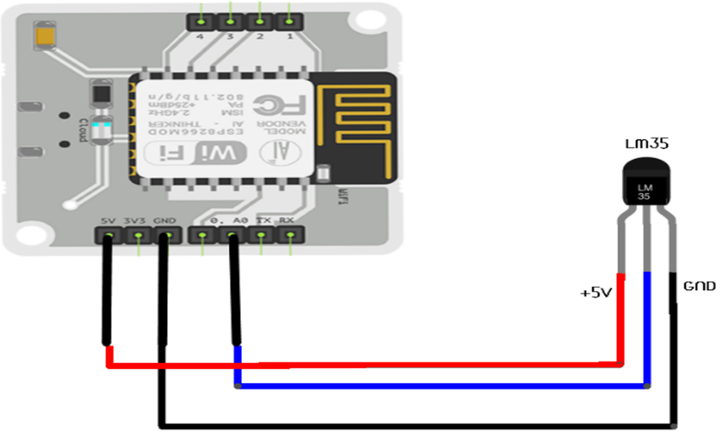
Data Count: 75

3.2 Hardware Interfaces

- Hardware interfaces include BoLT Wi-Fi module LM35 sensor for detecting temperature, jumper wires and power bank for giving power supply.

- Device will start working when the user starts the device by clicking on the start button from the user interface.





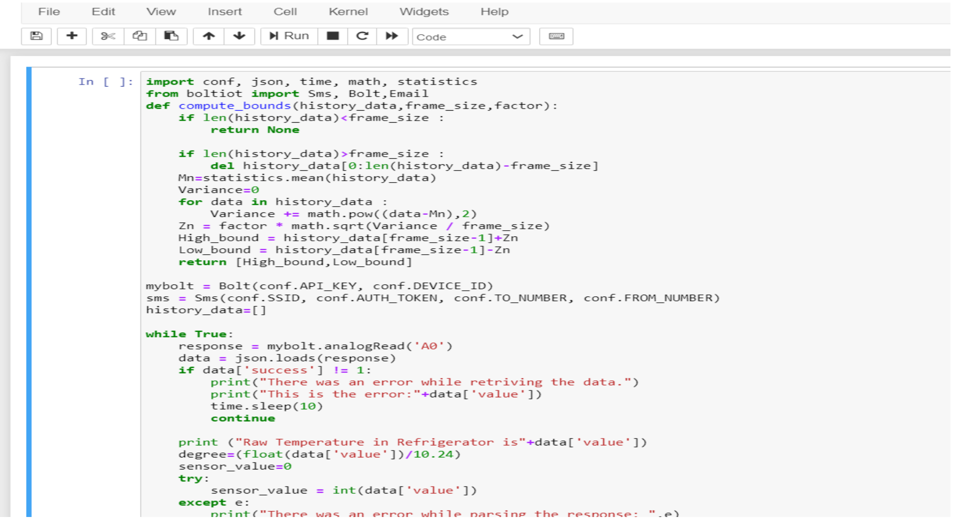
3.3 Software Interfaces

- Jupyter Notebook is used to run code written in python to give the device its functionality.

- Threshold value is specified in the program for sending alerts if temperature crosses threshold.

- The BoLT device API and product key is written in the program after importing the Bolt library to connect it to the device.

- When the program runs it will start measuring temperature every 30 seconds and will check the threshold condition and if temp crosses threshold alert is sent.



```python
import conf, json, time, math, statistics
from boltiot import Sms, Bolt,Email
def compute_bounds(history_data,frame_size,factor):
    if len(history_data)<frame_size :
        return None

    if len(history_data)>frame_size :
        del history_data[0:len(history_data)-frame_size]
    Mn=statistics.mean(history_data)
    Variance=0
    for data in history_data :
        Variance += math.pow((data-Mn),2)
    Zn = factor * math.sqrt(Variance / frame_size)
    High_bound = history_data[frame_size-1]+Zn
    Low_bound = history_data[frame_size-1]-Zn
    return [High_bound,Low_bound]

mybolt = Bolt(conf.API_KEY, conf.DEVICE_ID)
sms = Sms(conf.SSID, conf.AUTH_TOKEN, conf.TO_NUMBER, conf.FROM_NUMBER)
history_data=[]

while True:
    response = mybolt.analogRead('A0')
    data = json.loads(response)
    if data['success'] != 1:
        print("There was an error while retriving the data.")
        print("This is the error:"+data['value'])
        time.sleep(10)
        continue

    print ("Raw Temperature in Refrigerator is"+data['value'])
    degree=(float(data['value'])/10.24)
    sensor_value=0
    try:
        sensor_value = int(data['value'])
    except e:
        print("There was an error while parsing the response: ",e)
```

4. System Features

4.1 Temperature Measurement

4.1.1 Description and Priority

- *This is the main feature of the device and is high priority (9) .*
- *Consistently getting accurate temperature is very important for the device to ensure that the medicines don't get damaged.*

4.1.2 Stimulus/Response Sequences

- *The device prompts temperature checks in regular intervals of 30 secs*

4.2 Alert system using Twilio API

- Twilio's APIs power its platform for communications. Behind these APIs is a software layer connecting and optimizing communications networks around the world to allow your users to call and message anyone, globally.
- This would allow us to create an alert system using the api to communicate easily with the workers.

4.3 User dashboard on BOLT cloud platform

- Bolt IoT platform gives you the capability to control your devices and collect data from IoT devices safely and securely no matter where you are.
- Get actionable insights by deploying machine learning algorithms with just a few clicks to detect anomalies as well as predict sensor values.

Functional requirements:

- *Should measure the temperature of the cooling chambers very accurately.*
- *Should only require USB cable to connect to power source*
- *Easy to operate*
- *Can view the data easily even under bright light*
- *Send Alerts*
- *Protection from short circuiting*
- *Safe*

5. Nonfunctional Requirements*:*

- ■ *Durable*
- ■ *Requires less maintenance*
- ■ *Eco-friendly*
- ■ *Can download data into excel format*
- ■ *View data remotely on his phone*

5.2 Performance Requirements

- *High accuracy (upto 3 precision units) temperature measurement sensors.*
- *Low latency (less than 3ms) IOT sensors and wifi modules.*
- *Upto 1-3 ms response time for the Alert API.*
- *BOLT cloud platform with setup cache and backup service to ensure low latency data retrieval.*

Safety Requirements

- *Prevent bringing very hot or very cold external material near the sensors or else it could generate an alert and cause damage to medicines.*
- *Prevent touching the device with wet hands or it might result in an electric shock.*
- *Touch sensors with clean hands to prevent dust on the sensors causing inaccurate readings thus damaging the medicines.*

5.3 Security Requirements

- *Strong password required during authentication for BOLT cloud account to avoid leakage of company data.*
- *Control over who gets the security alerts.*

5.4 Software Quality Attributes

- *The learning curve must be minimal and be easily operable even by less skilled labor.*
- *The device should be robust and durable to work for a long time in a factory environment.*
- *The sensors must be accurate and reliable.*
- *Some parts such as the sensors of the device should be reusable.*

5.5 Business Rules

- *Only the managers and the caretakers of the factory should get the alerts.*
- *The BOLT cloud account should be accessible by the managers, analytics team and the onground team responsible for maintaining the cooling chambers.*

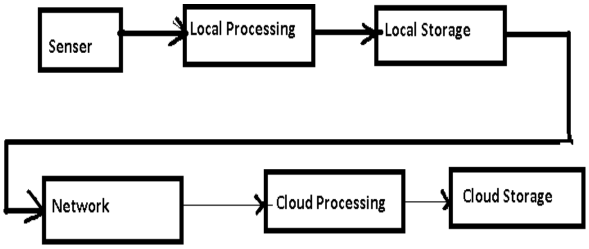
6. Other Requirements

- *Minimum cache storage of 10Mb in the device itself to store program data.*
- *Silicon 10nm microchips.*
- *LED indicators.*

Appendix A: Glossary

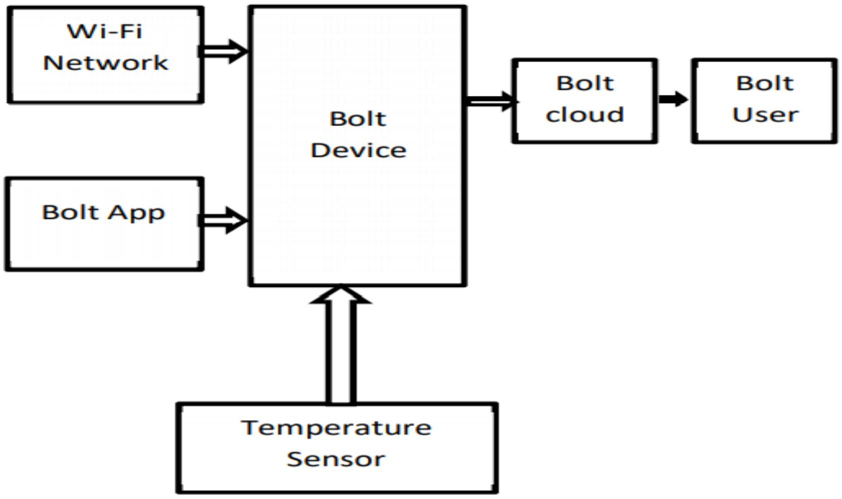
- *API- Application Programming Interface*
- *LM35- The lm35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the celsius (centigrade) temperature.*
- *CSV- Comma Separated Values*
- *IOT- Internet of Things*

Appendix B: Analysis Models

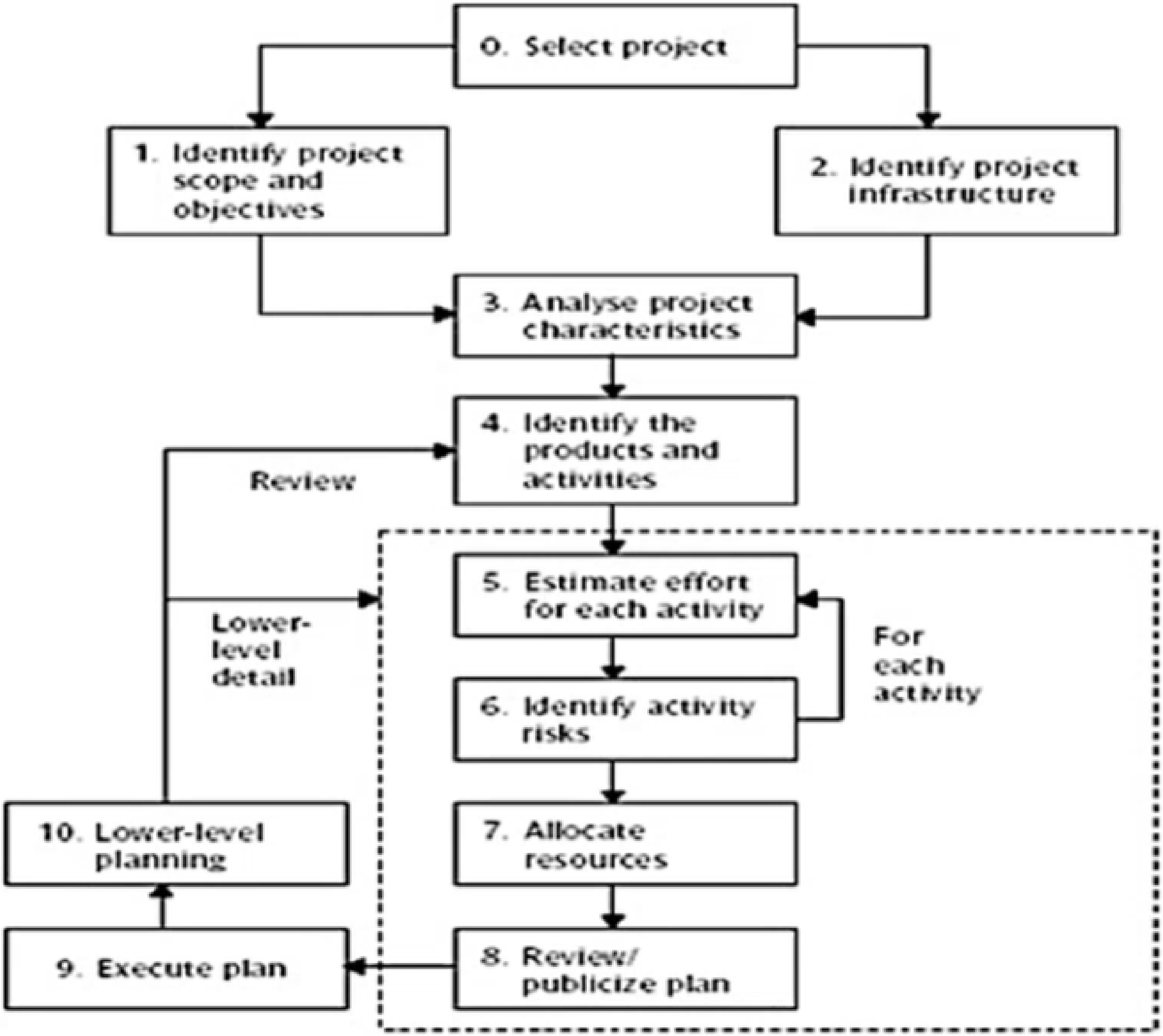


Flow of Data

Block Diagram



Software Project Lifecycle


0. Select project
1. Identify project scope and objectives
2. Identify project infrastructure
3. Analyse project characteristics
4. Identify the products and activities
Review
5. Estimate effort for each activity
Lower-level detail
For each activity
6. Identify activity risks
7. Allocate resources
8. Review/ publicize plan
9. Execute plan
10. Lower-level planning

Activity 0 : Select Project

Project Introduction-

- As we know, farmers of India have been protesting for the past several months against the Farmer laws thinking that it would force them to sell their produce for cheaper rates.

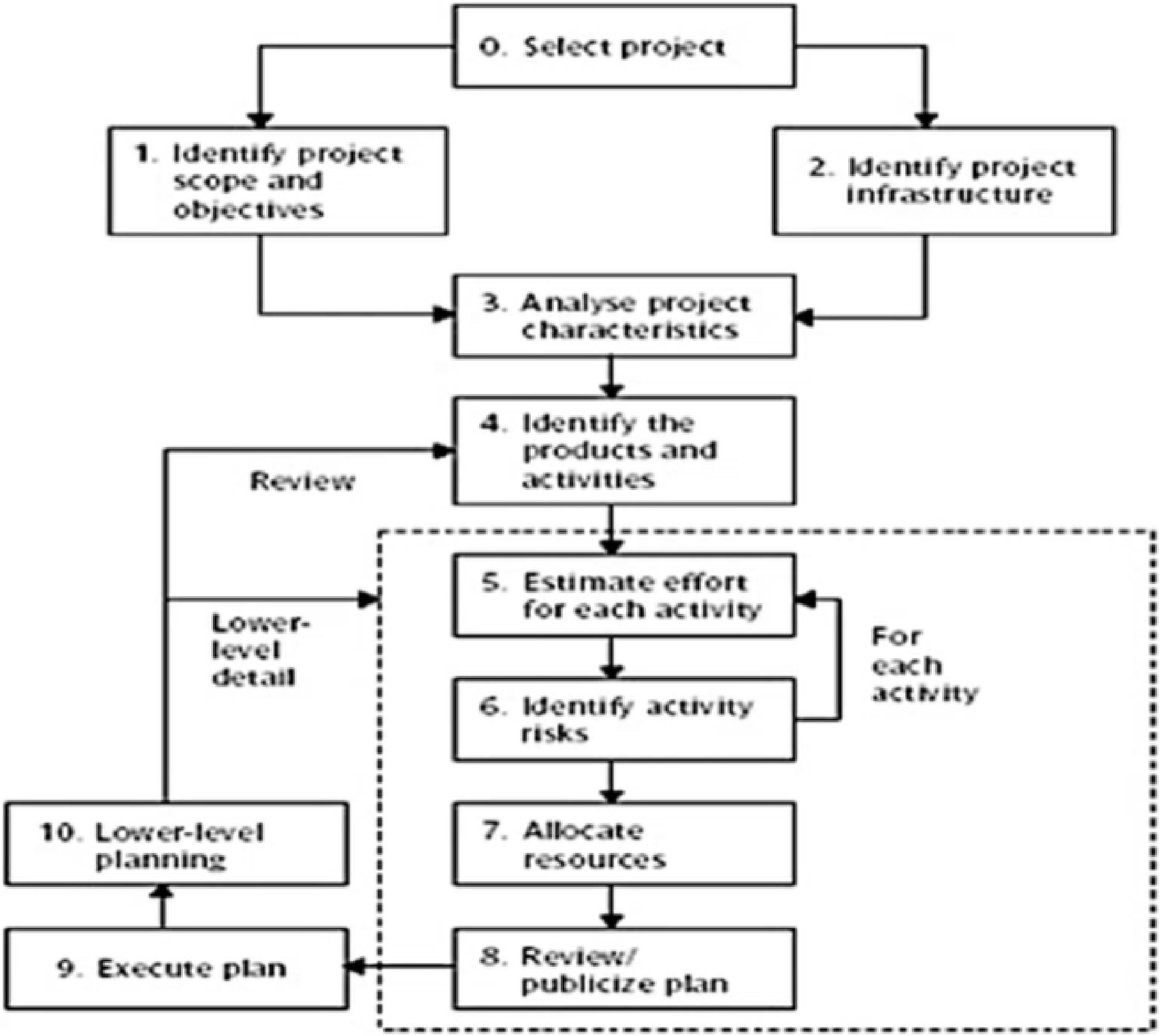
- We will develop a web based ecommerce platform for farmers where farmers can list their produce/ products and can directly sell to the consumers.

- This portal will help farmers to cut out the middleman from the deal and earn higher margins on their produce.

- The portal complies with the prices set by the Government as well as earns the farmers the extra margins that they need thus it is in favour of both the parties.

- It can also help connect farmers to commercial retailers who want to buy produce in large quantities but they can place orders on our portal which stays open for grab by any farmer instead of getting a single farmer under contract that could exploit the farmer.

Software Project Lifecycle


0. Select project
1. Identify project scope and objectives
2. Identify project infrastructure
3. Analyse project characteristics
4. Identify the products and activities
Review
5. Estimate effort for each activity
Lower-level detail
For each activity
6. Identify activity risks
10. Lower-level planning
7. Allocate resources
9. Execute plan
8. Review/ publicize plan

Activity 1 : Identify Project Scope and Objectives

Project Scope-

- Farm and livestock-related interventions have been found to be a successful strategy for poverty alleviation all over the world, and livestock rearing provides a living for a large percentage of the rural population.
- In India, farming is a major livestock enterprise in which small and marginal farmers earn their living.
- The main scope of our project is to improve the growth of the farm producers and consumers relation.
- Farm owners on a small and large scale will benefit.
- We improve the financial situation of both small and large-scale farm owners.
- We want to eliminate the farm owners' poverty.
- This system will assist farmers in selling agricultural products in a convenient manner and provide consumers with simple to use applications.

Project Objectives-

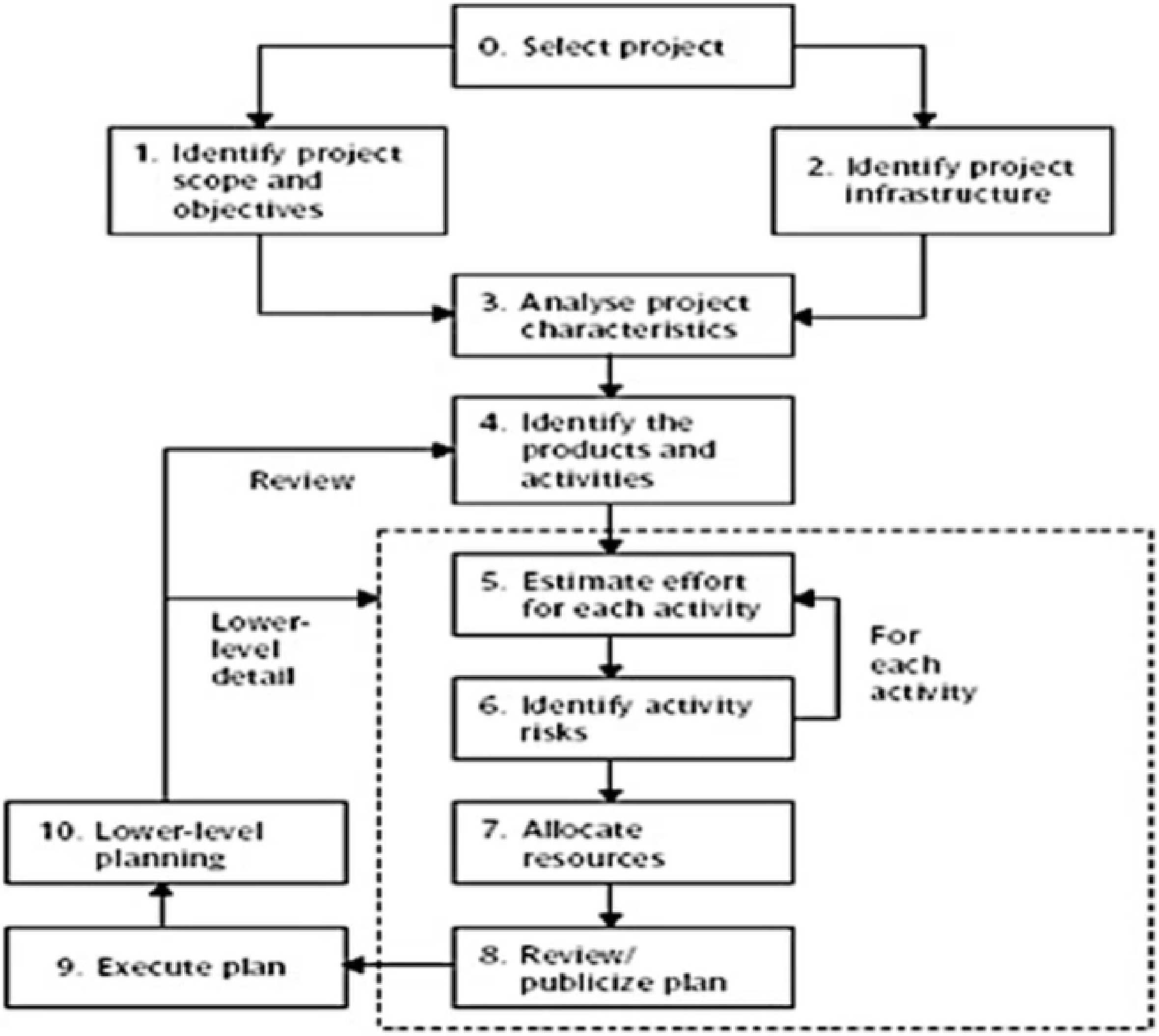
- Have a nice SEO friendly home page with all information about the product and our aim.
- Allow farmers to list their produce on the website for consumers to buy.
- Provide a user friendly dashboard for both the farmers and the customers with authentication features to ensure security on the platforma and avoid scams.

- Aadhar card number or any other government-approved identification should be used to verify sellers.
- Farmers and customers can interact with buying and selling portals with complete product information.
- Customers should be able to write reviews and ask questions, and sellers should be able to upload images and list their products.
- 24 hour on call support for users on the platform.
- Google language translate feature integration on the website for convenience of the farmers as well as the customers.
- Payment gateway on the platform using Razorpay services.
- Auto generated bills on the products and services used/ bought on the platform.

Technologies to be used-

- React.js (Javascript based framework) for the Frontend of the web based platform.
- HTML and CSS for the structure of the website.
- PostgreSQL for database management .
- QA tools to check the application before deployment
- AWS for deployment.

Software Project Lifecycle


0. Select project
1. Identify project scope and objectives
2. Identify project infrastructure
3. Analyse project characteristics
4. Identify the products and activities
Review
5. Estimate effort for each activity
Lower-level detail
For each activity
6. Identify activity risks
7. Allocate resources
10. Lower-level planning
9. Execute plan
8. Review/ publicize plan

Activity 2 : Identify Project Infrastructure

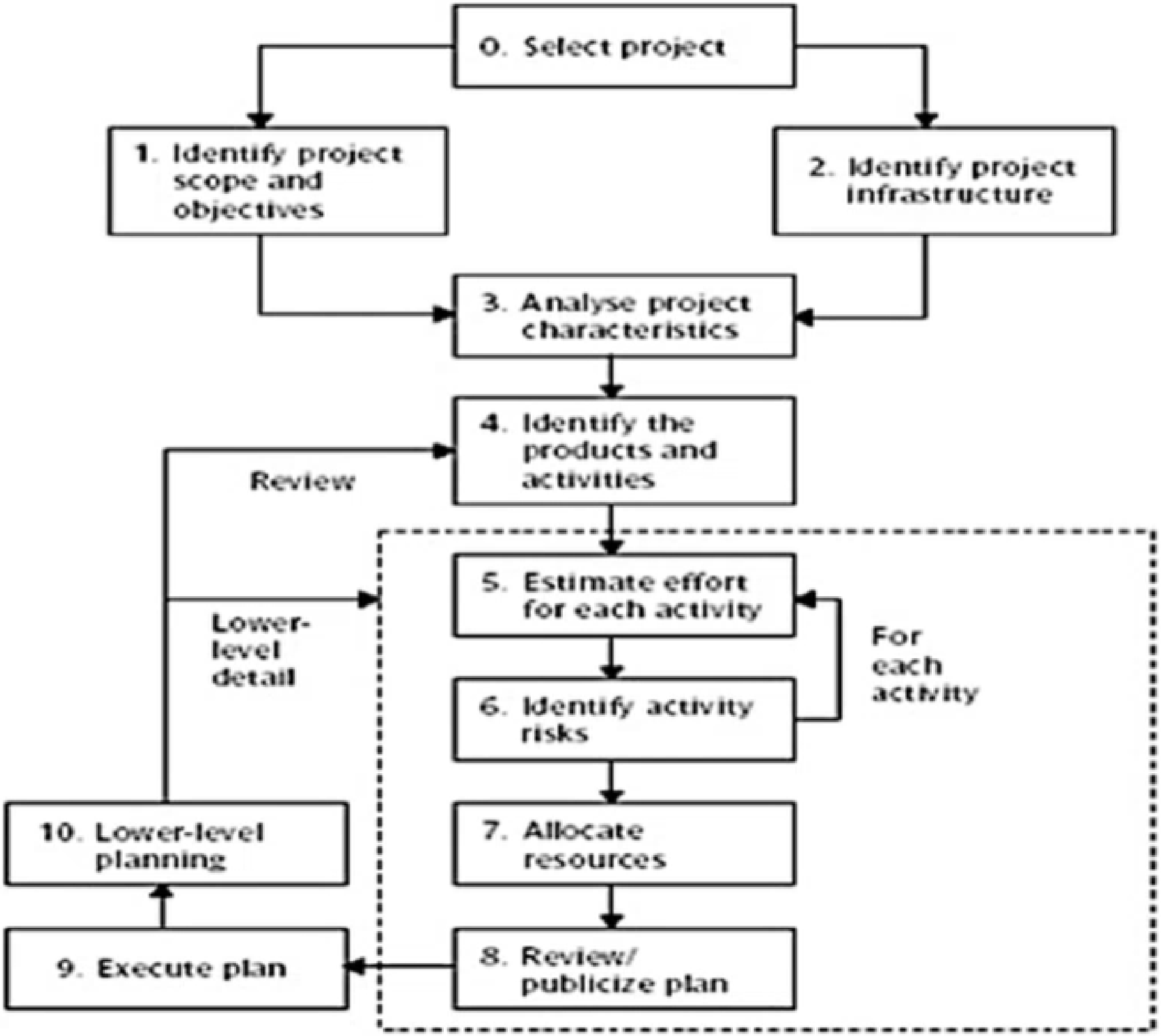
Project Infrastructure Planning-

- We would be using the Waterfall model of project development.

- This model is used only when the requirements are very well known, clear and fixed, product definition is stable, technology is understood.

- Waterfall development allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Software Project Lifecycle

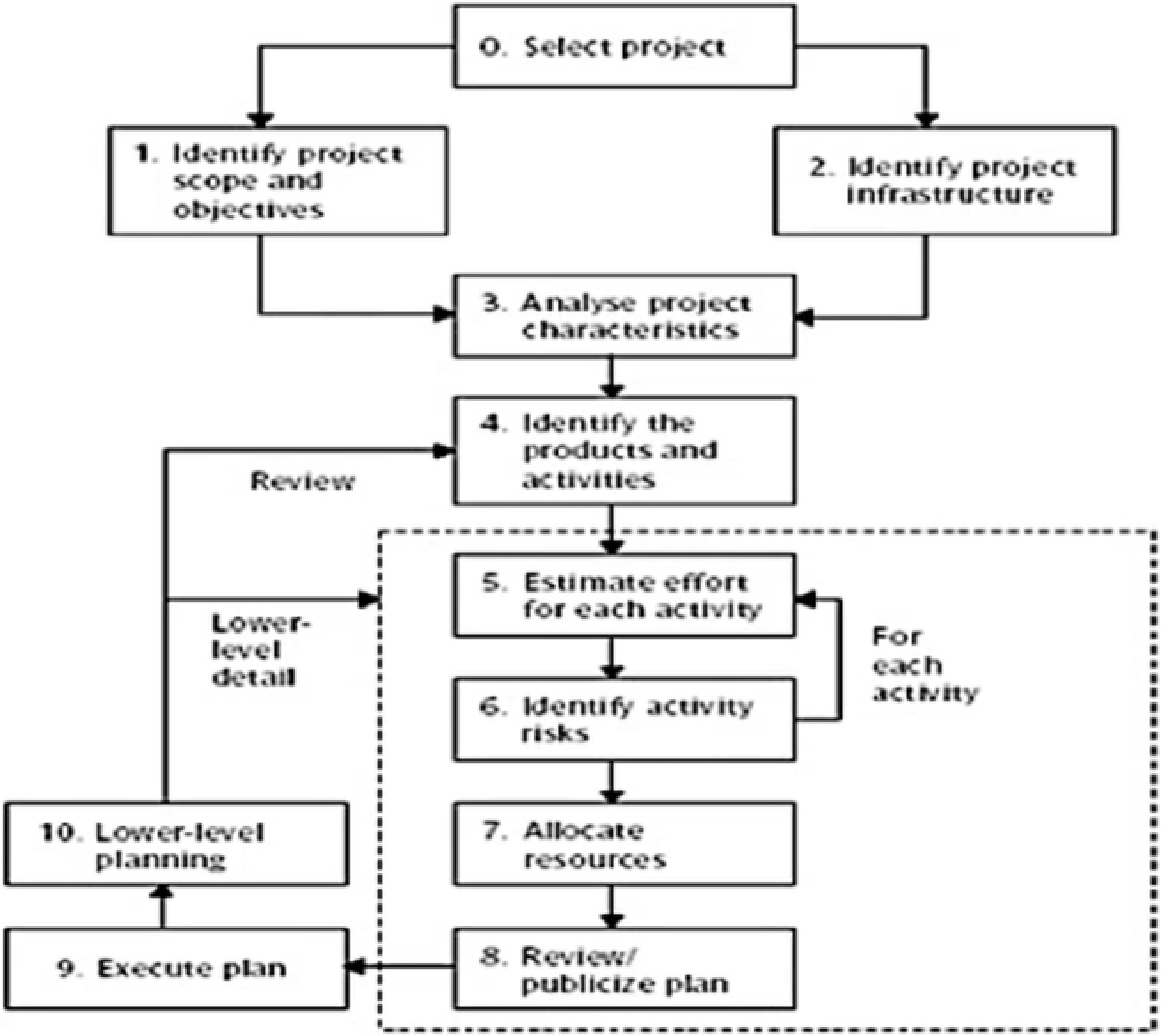

0. Select project
1. Identify project scope and objectives
2. Identify project infrastructure
3. Analyse project characteristics
4. Identify the products and activities
Review
5. Estimate effort for each activity
Lower-level detail
For each activity
6. Identify activity risks
7. Allocate resources
10. Lower-level planning
9. Execute plan
8. Review/ publicize plan

Activity 3 : Analyse Project characteristics

Project Characterization-

- Our Ecommerce platform for farmers is data oriented.
- System is not safety critical.
- Our project follows a C2C model. (Consumer to Consumer)
- Our platform will only provide application specific services.
- System follows a defined framework laid out by all the stakeholders.

Software Project Lifecycle


0. Select project
1. Identify project scope and objectives
2. Identify project infrastructure
3. Analyse project characteristics
4. Identify the products and activities
Review
Lower-level detail
5. Estimate effort for each activity
6. Identify activity risks
For each activity
7. Allocate resources
8. Review/ publicize plan
9. Execute plan
10. Lower-level planning

Activity 4 : Identify the products and activities

Work Breakdown Structure

1. Project Management Activities
    - 1.1 Project Charter
    - 1.2 Project Scope
    - 1.3 Monitoring/Controlling Activities
    - 1.4 Project Use cases
    - 1.5 UP Management Activities Completed

2. Requirements Analysis
    - 2.1 Requirements Gathering
    - 2.2 SRS
        - 2.2.1 Functional Requirements
        - 2.2.2 Non Functional Requirements
    - 2.3 Requirements Analysis Completed

3. Initiation/Planning
    - 3.1 Feasibility Study
    - 3.2 Business Case

3.3 Project Management Planning

3.5 Security Planning

3.6 Initiation/Planning Activities Completed

4. Design

4.1 High-Level Design

4.2 Wireframings

4.3 Technical Specification

4.4 Low Level Design

4.5 Design Completed

5. Frontend

5.1 Home Page

5.2 Sign up & Login Page

5.3 Product Market Place

5.4 Farmer Dashboard

5.5 Customer Dashboard

5.7 Admin Panel

5.7 Payment Portal

5.8 Cart

6. Backend

6.1 Database

6.1.1 Database Design

6.1.2 Relationship development

6.1.3 Database Creation

6.2 Sign up & Login Backend

6.3 Product Market Place

6.4 Farmer Dashboard

6.5 Customer Dashboard

6.6 Admin Panel

6.7 Payment Portal

6.8 Cart

7. QA

7.1 Unit Test

7.2 Function Test

7.3 Integration Test

7.4 Regression Test

7.5 System Test

7.6 Test Completed

8. Deployment

    8.1 Deployment

    8.2 Complete Deployment

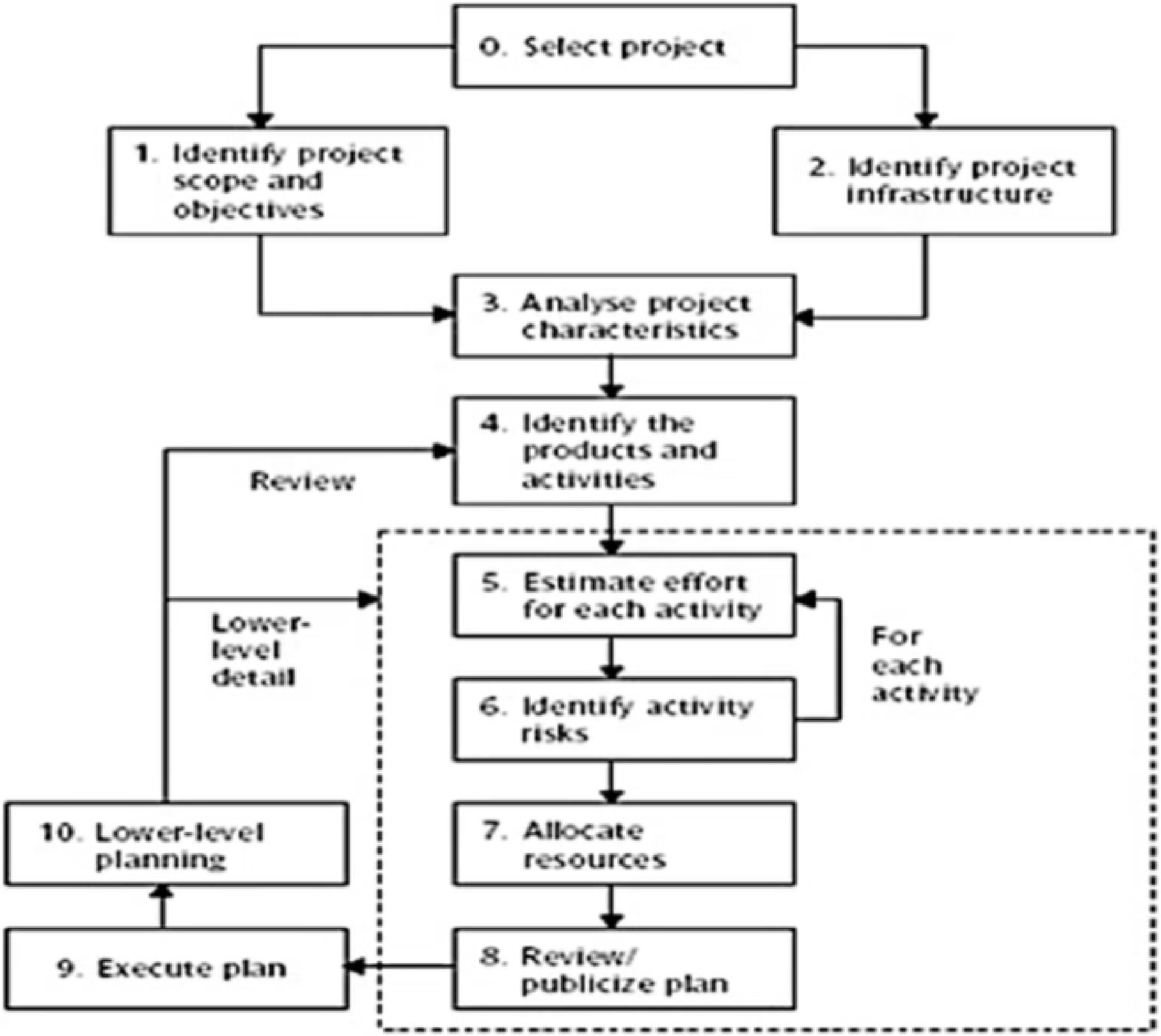
9. Operations & Maintenance

    9.1 Operations Activities

    9.2 Maintenance Activities

    9.3 Operations & Maintenance Completed

Software Project Lifecycle


0. Select project
1. Identify project scope and objectives
2. Identify project infrastructure
3. Analyse project characteristics
4. Identify the products and activities
Review
5. Estimate effort for each activity
Lower-level detail
For each activity
6. Identify activity risks
7. Allocate resources
8. Review/ publicize plan
9. Execute plan
10. Lower-level planning

Activity 5 : Estimate effort for each activity

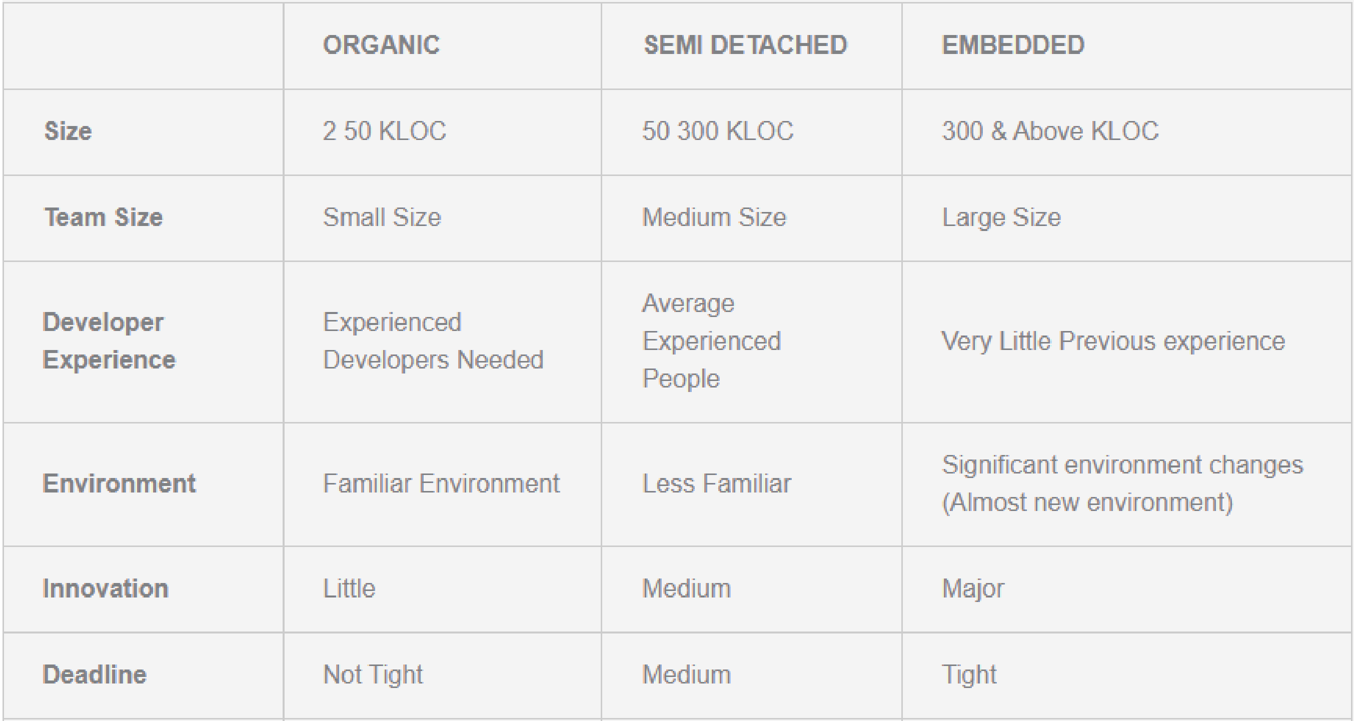
Effort estimation-

Effort estimation is the process of forecasting how much effort is required to develop or maintain a software application. Effort estimation is used to help draft project plans and budgets in the early stages of the software development life cycle.

COCOMO Model-

Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality.

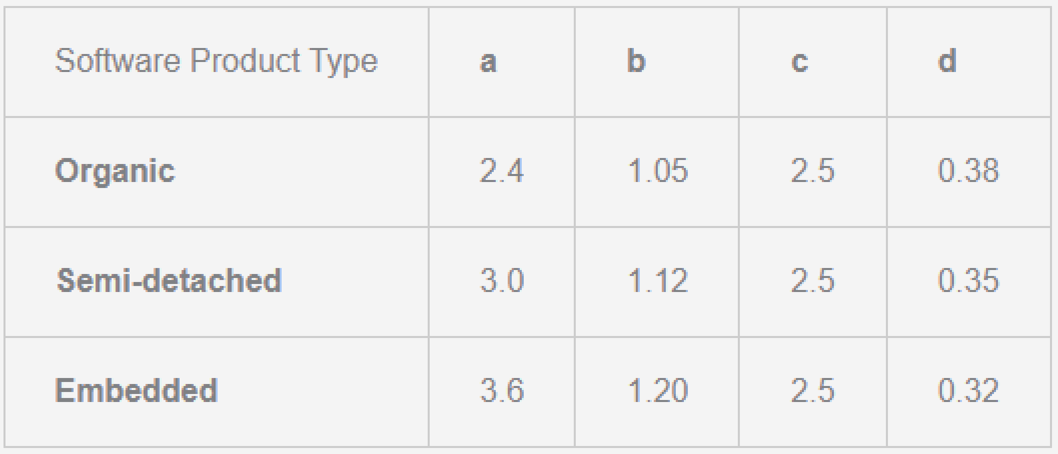
1. Estimated KLOC for the project = 5



	ORGANIC	SEMI DETACHED	EMBEDDED
Size	2 50 KLOC	50 300 KLOC	300 & Above KLOC
Team Size	Small Size	Medium Size	Large Size
Developer Experience	Experienced Developers Needed	Average Experienced People	Very Little Previous experience
Environment	Familiar Environment	Less Familiar	Significant environment changes (Almost new environment)
Innovation	Little	Medium	Major
Deadline	Not Tight	Medium	Tight

2. Based on KLOC and other factors we choose the organic mode of the Basic Cocomo model.

3. Values of constants based on modes-



Software Product Type	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

4. Calculations-

Effort = a*((KLOC)^b)
= 2.4*((5)^1.05)
= 12.98 person-months

Development time = c*((Effort)^d)
= 2.5*((12.98)^0.38)
= 6.6 months

Average Staff Size = Effort/ Development time
= 12.98/6.6
= ceil(1.96)
= **2** persons

Productivity = KLOC/ Effort
= 5/12.98
= 0.38 KLOC/ person-months

Function Points-

5 Types of Functional Units

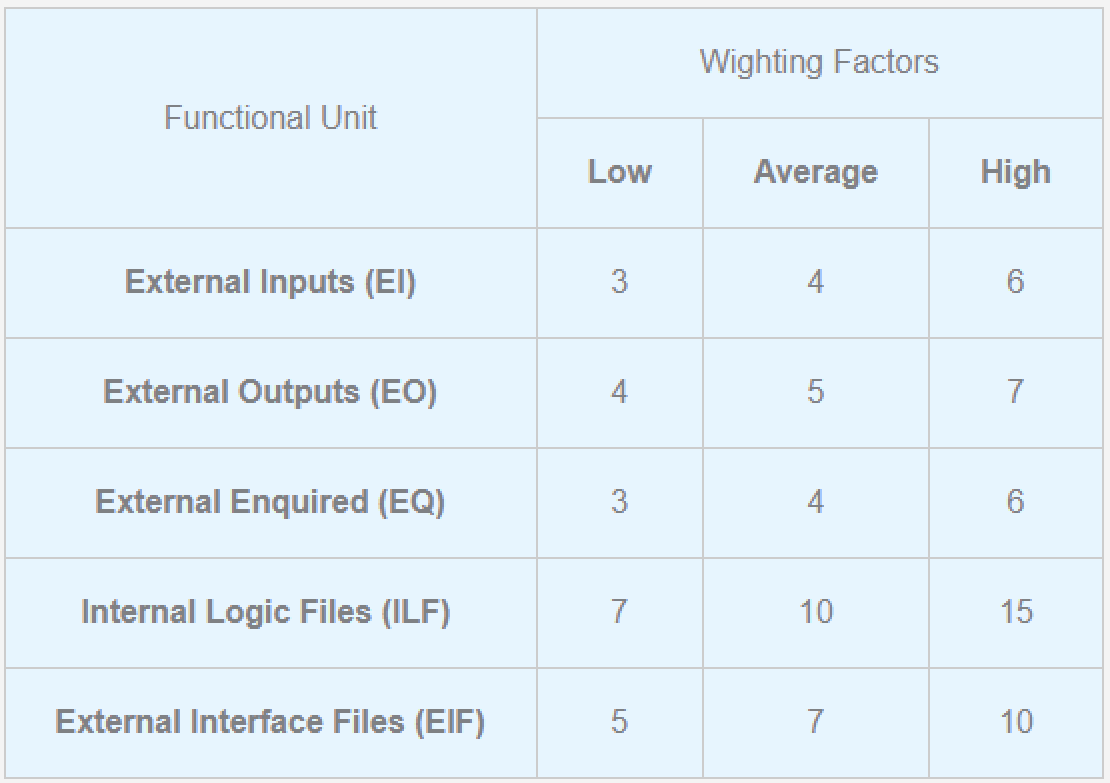
- Internal Logic Files (ILF) – The control info or logically related data that is present within the system.
- External Interface Files (EIF) – The control data or other logical data i.e referenced by the system but present in another system.
- External Inputs (EI) – Data / control info that comes from outside our system
- External Outputs (EO) – data that goes out of the system after generation
- External Enquired (EQ) – Combination of i/o – o/p resulting data retrieval

ILF = 6
EIF = 3
EI = 12
EO = 9
EQ = 3
Final F.P = UFP X CAF

UFP = Unadjusted Function Point

CAF = Complexity Adjustment Factor

Weighing factors-



Functional Unit	Wighting Factors		
	Low	Average	High
External Inputs (EI)	3	4	6
External Outputs (EO)	4	5	7
External Enquired (EQ)	3	4	6
Internal Logic Files (ILF)	7	10	15
External Interface Files (EIF)	5	7	10

5. Taking weighing factors as low we calculate UFP-

UFP = (6*7) + (3*5) + (12*3) + (9*4) + (3*3)
UFP =42+15+36+36+9
UFP =138

6. Calculating Complexity Adjustment Factor

Complexity Adjustment Factor is calculated using 14 aspects of processing complexity. 14 questions answered on a scale of 0 – 5.

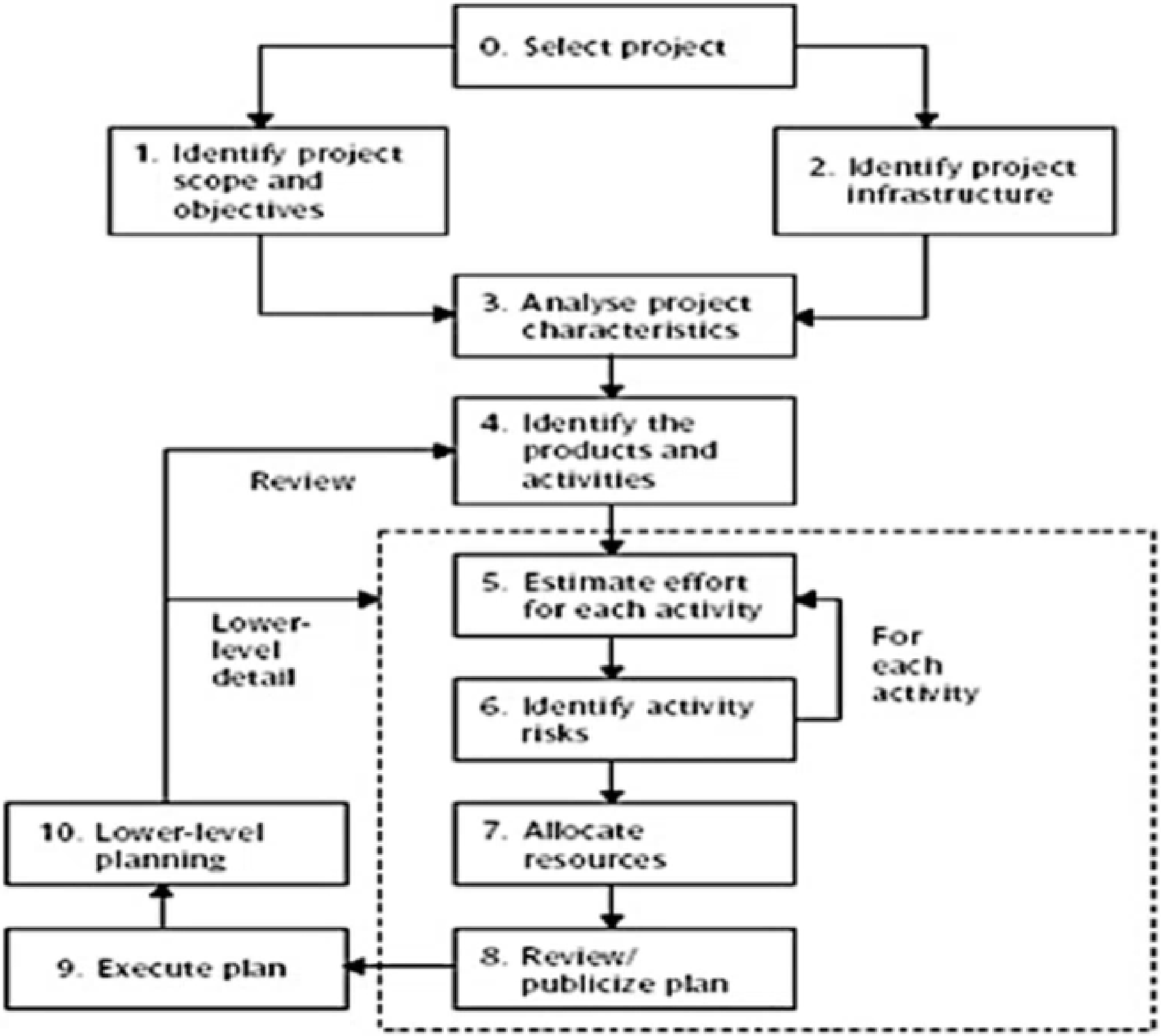
Considering average complexity.

$$\begin{aligned} CAF &= 0.65 + (0.01 * \sum(f_i)) \\ &= 0.65 + 0.01*(14*3) \\ &= 1.07 \end{aligned}$$

FP = UFP x CAF

= 138 x 1.07
= 147.66 function points

Software Project Lifecycle


0. Select project
1. Identify project scope and objectives
2. Identify project infrastructure
3. Analyse project characteristics
4. Identify the products and activities
Review
5. Estimate effort for each activity
Lower-level detail
For each activity
6. Identify activity risks
7. Allocate resources
10. Lower-level planning
9. Execute plan
8. Review/ publicize plan

Activity 6 : Identify activity risk

Risk Identification-

1. Technical Risks-

1.1 Web Server Overload / DDOS attack:

There is always a risk of overload on our application server because of too much traffic on our site or because of a DDOS (Distributed Denial of Service) that aims to send multiple requests to a website to exceed its capacity and prevent it from functioning correctly.

1.2 Security:

Improper security on the application such as lack of firewalls, encryption of customer data and 2 factor authentication and weak database passwords etc can be a huge potential risk in the growth of the platform.

1.3 XSS (Cross site scripting) attacks:

On a website where there are input fields and forms to fill, there is always the risk of XSS attacks where an attacker stores malicious script in the data sent from a website's search or contact form. When the malicious code executes inside a victim's browser, the attacker can fully compromise their interaction with the application.

1.4 SEO:

As we know that Google changes its algorithms quite often. Thus not paying attention to the SEO and digital marketing strategy of the website may result in low rankings and eventually less traffic and sales.

1.5 Bugs in the Software:

If a buggy software is sent into production, it may lead to huge losses to the company in the form of money or customer data.

1.6 Poor Code quality :

Poor code quality can lead to non maintainability or non scalability of a software.

1.7 Performance of Application:

Inefficient code can lead to reduced performance of the software application that might lead to losses to the company. Thus paying proper attention to time and space complexities during development is very important.

1.8 Hardware failure:

If a server or the mainframe crashes it can lead to major setback in the development of the software.

2. Organizational Risk

2.1 Budget:

Wrong estimation or improper allocation of resources can hamper the development of the project.

2.2 Disengaged Stakeholders:

If all the stakeholders are not on the same page then there could be conflicts during development or product launch.

3. Project Management-

3.1 Planning:

Shortfalls in proper planning of the project might cause issues during the development phase.

3.2 Schedule:

Wrong scheduling can have adverse effects on the timely delivery of the project to the customer.

3.3 Estimation:

Inaccurate time and budget estimation can negatively affect the project development and even delay project delivery in some cases.

4. **Operational Risk-**

4.1 Miscommunication:

Miscommunication between developers or stakeholders and project managers or project managers and developers can lead to inconsistencies in project development.

5. **Legal Risks-**

5.1 Legal licenses:

Uncertainty in renewal of ecommerce license can lead to a temporary hold on the business of our application.

5.2 Software quality complies with legal and industry standards:

In case the software does not comply with the legal and industry standard, our application can face unwanted legal actions.

5.3 Copyrighted material on the website:

Upload of copyrighted material on the website can cause the website to be taken down.

Risk Analysis-

Probability of Risk Occurrence

High probability – (80 % ≤ x ≤ 100%)

Medium-high probability – (60 % ≤ x < 80%)

Medium-Low probability – (30 % ≤ x < 60%)

Low probability (0 % < x < 30%)

Risk Impact

High – Catastrophic (Rating A – 100)

Medium – Critical (Rating B – 50)

Low – Marginal (Rating C – 10)

Risk	**Risk Probability**	**Risk Impact**
Technical Risks	Medium-High	High
Organizational Risk	Medium-Low	Medium
Project Management	High	High
Operational Risk	Low	Low
Legal Risks	Low	High

Risk Prioritization-

Priority	**Risk**
Very High	Project Management Plan
High	Technical
Medium	Legal
Low	Organizational
Very Low	Operational

Risk Mitigation Plan-

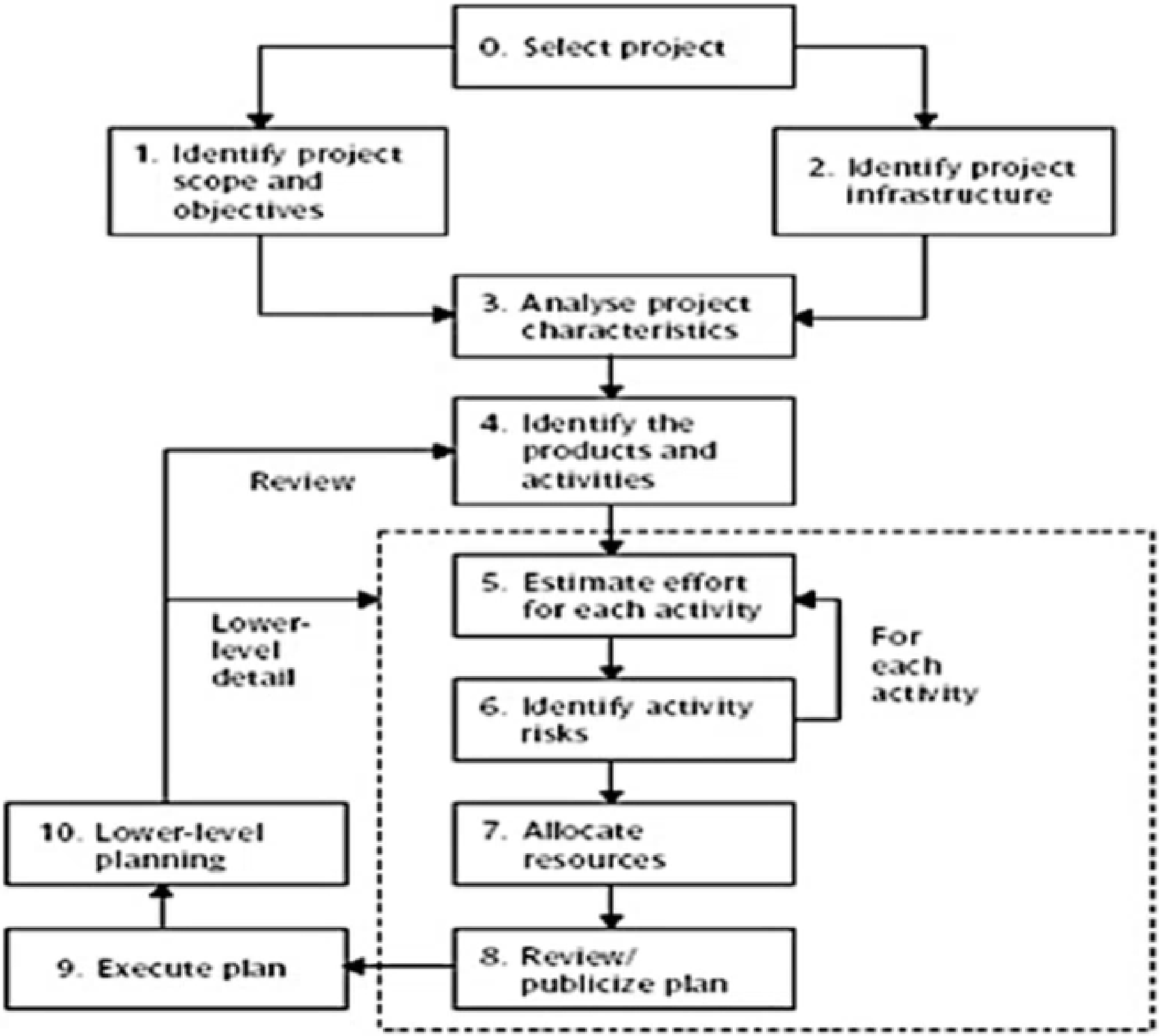
Risk mitigation plans are aimed at the following targets:

- Eliminating the risk
- Lowering the probability of risk occurrence
- Lowering the impact of the risk on the project objectives

Risk Event	**Risk Response**
Technical Risks	● AutoScaling incase of increased number of requests on the website. ● Use Code quality control softwares such as linters. ● Using Performance measuring softwares such as Lighthouse. ● Regular servicing and checkup to avoid hardware failure. ● Regressive testing to ensure bug free software.
Organizational Risk	● Careful analysis of budget and maintaining some flexibility to avoid ● Regular meetings with meeting minutes to ensure all the stakeholders can get written docs of all the important info.
Project Management	● Taking help from professionals for software budgeting. ● Use professional project management softwares such as Jira to track all the steps of a project lifecycle.
Operational Risk	● Regular meetings with meeting minutes to ensure all the stakeholders can get written docs of all the important info.

Legal Risks	● Make sure that the website has all the right legal requirements like: ○ cookie consent forms ○ privacy policies and data storage disclosure ○ terms & conditions ● Bot to detect copyrighted material.

Software Project Lifecycle


0. Select project
1. Identify project scope and objectives
2. Identify project infrastructure
3. Analyse project characteristics
4. Identify the products and activities
Review
Lower-level detail
5. Estimate effort for each activity
6. Identify activity risks
For each activity
7. Allocate resources
8. Review/ publicize plan
9. Execute plan
10. Lower-level planning

Activity 7 : Allocate resources

1. Staffing Matrix (Pranav)-

Roles	Characteristics	Member Name
Team Manager	● Good communication skills ● Management skills ● Understanding ● Conflict resolution	Pranitha
Backend Developer	● Problem solving ● Resourceful ● Knowledge of API integrations ● Knowledge of object oriented programming language	Pranav
Frontend Developer	● Eye for design ● Creative ● Quick learner ● Responsive interface ● Knowledge of Frontend Frameworks ● UI and UX designing concepts ● Wireframing and visualization	Sahil, Aastha
Database Management and Cloud architect	● Knowledge of cloud infrastructure. ● Knowledge of secure, low latency databases. ● Problem solving ● Technical prowess(SQL vs NoSQL)	Sourik

	● System backup and security	
Testing engineer	● Knowledge of white box testing ● Version Control ● Automation scripts ● Strong Logic	Pranav, Sourik, Sahil
QA engineer	● Knowledge of black box and functional testing ● Eye for detail ● Analytical skills	Sahil, Aastha, Pranitha

2. **Staffing Matrix (Sourik)-**

Roles	Characteristics	Member Name
Team Manager	● Good idea on how to break down work ● Management skills ● Conflict resolution	Pranav
Backend Developer	● Problem solving ● Resourceful ● API integrations	Sahil, Aastha
Frontend Developer	● Experience in Web design ● Fast pace	Pranav,Pranitha

	● Responsive interface	
Database Management and Cloud architect(related to IoT)	● Knowledge of secure databases. ● Problem solving ● Ideas about hosting	Sourik, Sahil
Testing engineer	● Knowledge of white box testing ● Version Control ● Automation scripts	Pranav, Sourik, Sahil
QA engineer	● Knowledge of black box and functional testing ● Analytical skills	Aastha, Pranitha

3. **Staffing Matrix (Sahil)-**

Roles	Characteristics	Member Name
Team Manager	● Assigning responsibilities ● Coordinating with others ● Managing tasks ● Conflict resolution	Pranav
Backend Developer	● Designing ● Coding ● Debugging code	Sourik, Aastha
Frontend Developer	● Front page design ● responsible for creating user friendly webpage ● Responsive interface	Sahil,Pranitha
Database Management and Cloud architect(related to IoT)	● Working with AWS and Azure cloud ● Management of data ● Ensuring security of data	Sourik, Aastha
Testing engineer	● white box testing ● Version Control ● debugging code	Sourik, Sahil

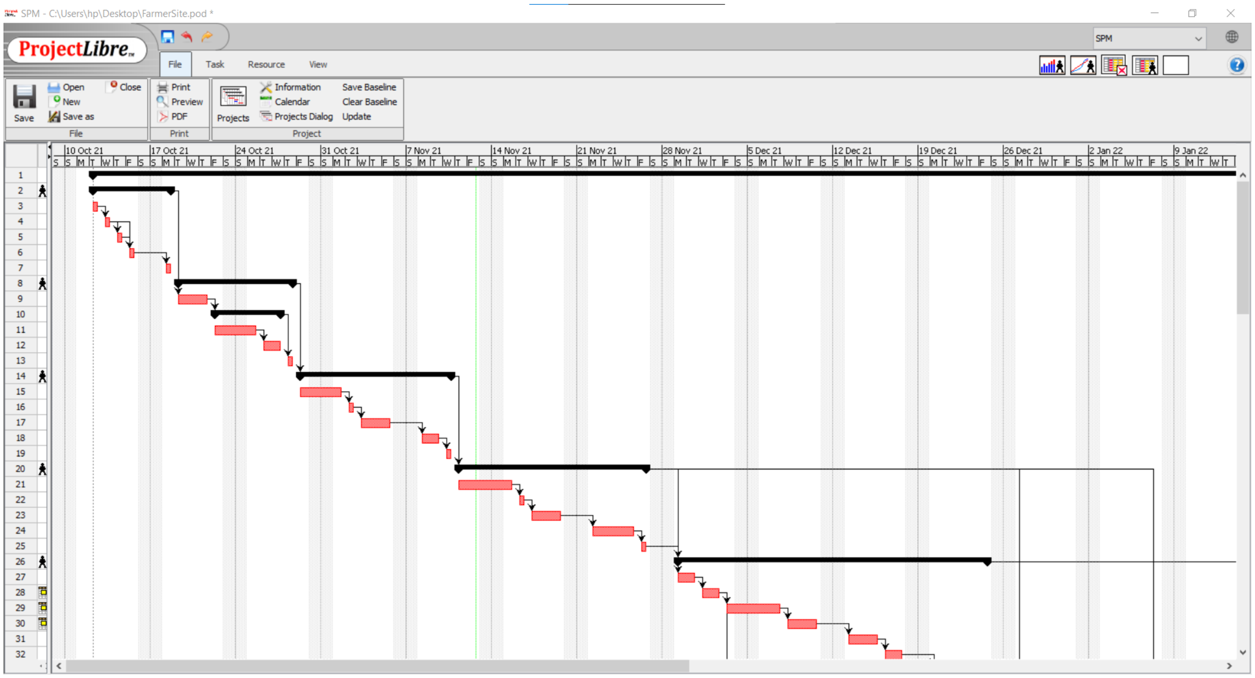
QA engineer	● black box and functional testing ● Improving Quality of code	Pranav, Pranitha

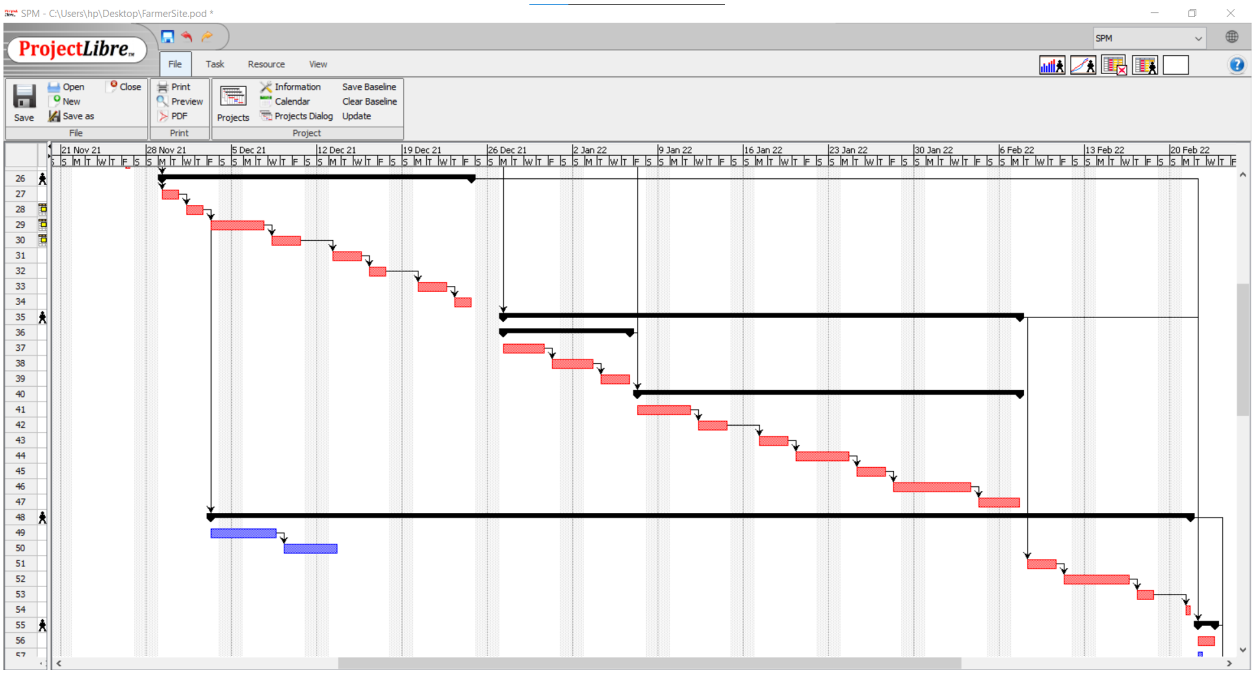
Staffing Matrix (Aastha)

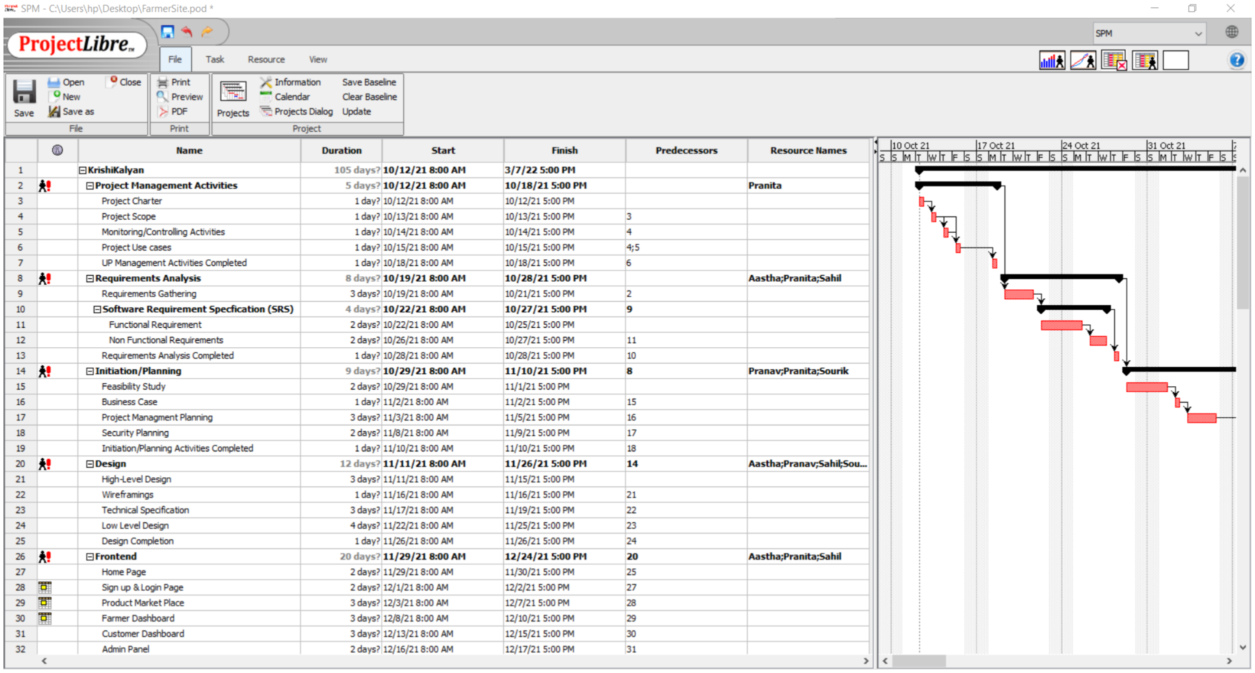
ROLES	CHARACTERISTICS	TEAM MEMBER
Project Manager	Strong leadership skills, ability to coordinate, motivate, and create an atmosphere where a team can successfully exchange ideas, come up with a plan, and complete the project.	Aastha
Back End Developer	Effective time manager, Problem solving skills, End user focused, fast learner, teamwork	Pranav, Sahil
Database Administrator	Maintaining Backend database Strong analytical skills Expertise in SQL Queries	Sourik, Pranita

QA Engineer	Must have the know how of all the working of hardware and software of the products i.e. hardware(Cameras) and software (Managing platform)	Pranita
Front End Developer	Excellent analytical and multitasking skills, Communication skills, good knowledge of HTML, CSS, ability to perform well in a fast-paced environment	Pranita, Aastha, Pranav
Tester	Software knowledge, debugging skills, no compromise on quality, identify and manage risks	Aastha, Sahil

Gantt Chart-

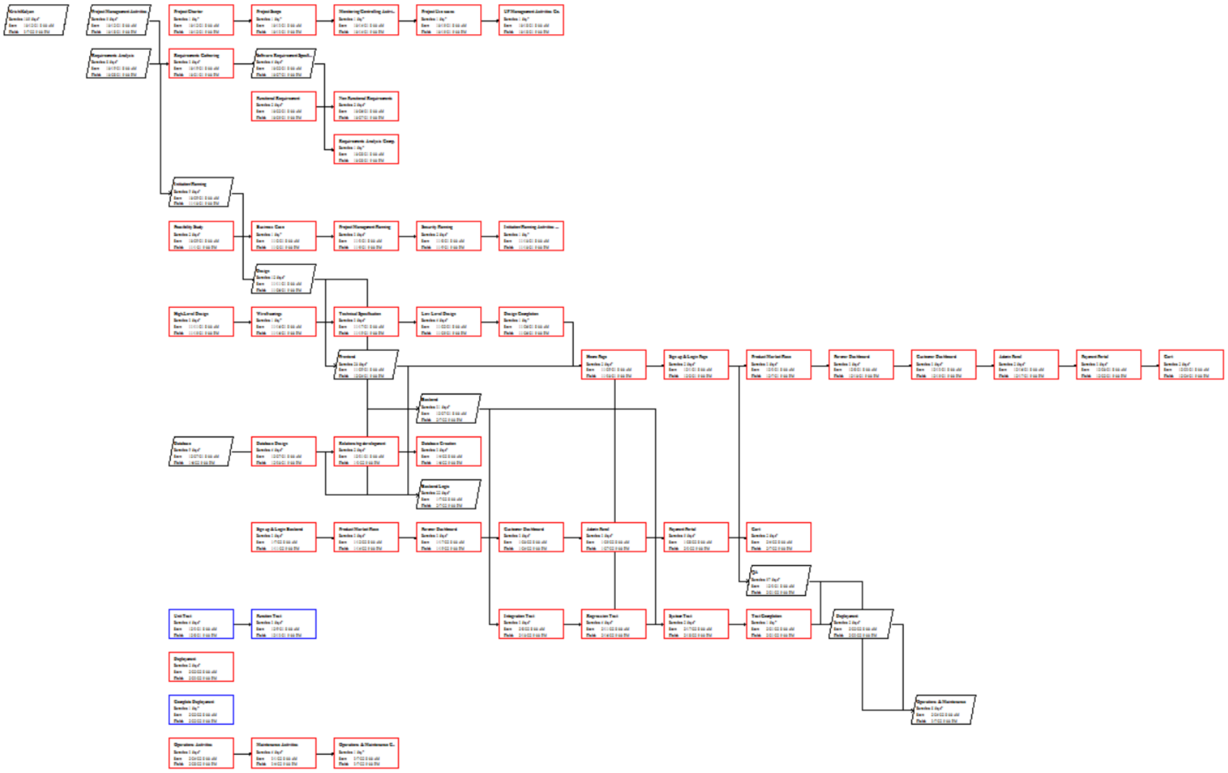

SPM - C:\Users\hp\Desktop\FarmerSite.pod *
ProjectLibre
SPM
File
Task
Resource
View
Save
Open
New
Save as
Close
Print
Preview
PDF
Projects
Information
Calendar
Projects Dialog
Save Baseline
Clear Baseline
Update
File
Print
Project
10 Oct 21
17 Oct 21
24 Oct 21
31 Oct 21
7 Nov 21
14 Nov 21
21 Nov 21
28 Nov 21
5 Dec 21
12 Dec 21
19 Dec 21
26 Dec 21
2 Jan 22
9 Jan 22
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32

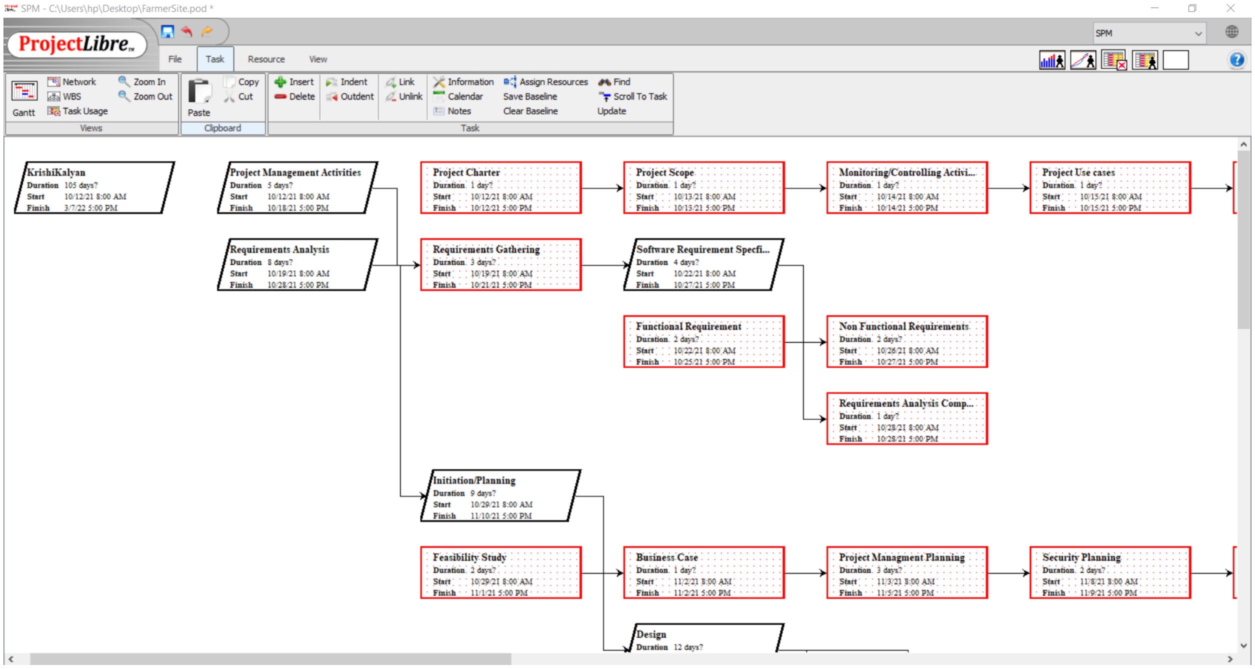

SPM - C:\Users\hp\Desktop\FarmerSite.pod *
ProjectLibre
SPM
File
Task
Resource
View
Save
Open
New
Save as
Close
Print
Preview
PDF
Projects
Information
Calendar
Projects Dialog
Save Baseline
Clear Baseline
Update
File
Print
Project
21 Nov 21
28 Nov 21
5 Dec 21
12 Dec 21
19 Dec 21
26 Dec 21
2 Jan 22
9 Jan 22
16 Jan 22
23 Jan 22
30 Jan 22
6 Feb 22
13 Feb 22
20 Feb 22
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56



		Name	Duration	Start	Finish	Predecessors	Resource Names
1		⊟KrishiKalyan	105 days?	10/12/21 8:00 AM	3/7/22 5:00 PM		
2		⊟Project Management Activities	5 days?	10/12/21 8:00 AM	10/18/21 5:00 PM		Pranita
3		Project Charter	1 day?	10/12/21 8:00 AM	10/12/21 5:00 PM		
4		Project Scope	1 day?	10/13/21 8:00 AM	10/13/21 5:00 PM	3	
5		Monitoring/Controlling Activities	1 day?	10/14/21 8:00 AM	10/14/21 5:00 PM	4	
6		Project Use cases	1 day?	10/15/21 8:00 AM	10/15/21 5:00 PM	4;5	
7		UP Management Activities Completed	1 day?	10/18/21 8:00 AM	10/18/21 5:00 PM	6	
8		⊟Requirements Analysis	8 days?	10/19/21 8:00 AM	10/28/21 5:00 PM		Aastha;Pranita;Sahil
9		Requirements Gathering	3 days?	10/19/21 8:00 AM	10/21/21 5:00 PM	2	
10		⊟Software Requirement Specfication (SRS)	4 days?	10/22/21 8:00 AM	10/27/21 5:00 PM	9	
11		Functional Requirement	2 days?	10/22/21 8:00 AM	10/25/21 5:00 PM		
12		Non Functional Requirements	2 days?	10/26/21 8:00 AM	10/27/21 5:00 PM	11	
13		Requirements Analysis Completed	1 day?	10/28/21 8:00 AM	10/28/21 5:00 PM	10	
14		⊟Initiation/Planning	9 days?	10/29/21 8:00 AM	11/10/21 5:00 PM	8	Pranav;Pranita;Sourik
15		Feasibility Study	2 days?	10/29/21 8:00 AM	11/1/21 5:00 PM		
16		Business Case	1 day?	11/2/21 8:00 AM	11/2/21 5:00 PM	15	
17		Project Managment Planning	3 days?	11/3/21 8:00 AM	11/5/21 5:00 PM	16	
18		Security Planning	2 days?	11/8/21 8:00 AM	11/9/21 5:00 PM	17	
19		Initiation/Planning Activities Completed	1 day?	11/10/21 8:00 AM	11/10/21 5:00 PM	18	
20		⊟Design	12 days?	11/11/21 8:00 AM	11/26/21 5:00 PM	14	Aastha;Pranav;Sahil;Sou...
21		High-Level Design	3 days?	11/11/21 8:00 AM	11/15/21 5:00 PM		
22		Wireframings	1 day?	11/16/21 8:00 AM	11/16/21 5:00 PM	21	
23		Technical Specification	3 days?	11/17/21 8:00 AM	11/19/21 5:00 PM	22	
24		Low Level Design	4 days?	11/22/21 8:00 AM	11/25/21 5:00 PM	23	
25		Design Completion	1 day?	11/26/21 8:00 AM	11/26/21 5:00 PM	24	
26		⊟Frontend	20 days?	11/29/21 8:00 AM	12/24/21 5:00 PM	20	Aastha;Pranita;Sahil
27		Home Page	2 days?	11/29/21 8:00 AM	11/30/21 5:00 PM	25	
28		Sign up & Login Page	2 days?	12/1/21 8:00 AM	12/2/21 5:00 PM	27	
29		Product Market Place	3 days?	12/3/21 8:00 AM	12/7/21 5:00 PM	28	
30		Farmer Dashboard	3 days?	12/8/21 8:00 AM	12/10/21 5:00 PM	29	
31		Customer Dashboard	3 days?	12/13/21 8:00 AM	12/15/21 5:00 PM	30	
32		Admin Panel	2 days?	12/16/21 8:00 AM	12/17/21 5:00 PM	31	

Pert Chart-





KrishiKalyan
Duration 105 days?
Start 10/12/21 8:00 AM
Finish 3/7/22 5:00 PM

Project Management Activities
Duration 5 days?
Start 10/12/21 8:00 AM
Finish 10/18/21 5:00 PM

Project Charter
Duration 1 day?
Start 10/12/21 8:00 AM
Finish 10/12/21 5:00 PM

Project Scope
Duration 1 day?
Start 10/13/21 8:00 AM
Finish 10/13/21 5:00 PM

Monitoring/Controlling Activi...
Duration 1 day?
Start 10/14/21 8:00 AM
Finish 10/14/21 5:00 PM

Project Use cases
Duration 1 day?
Start 10/15/21 8:00 AM
Finish 10/15/21 5:00 PM

Requirements Analysis
Duration 8 days?
Start 10/19/21 8:00 AM
Finish 10/28/21 5:00 PM

Requirements Gathering
Duration 3 days?
Start 10/19/21 8:00 AM
Finish 10/21/21 5:00 PM

Software Requirement Specfi...
Duration 4 days?
Start 10/22/21 8:00 AM
Finish 10/27/21 5:00 PM

Functional Requirement
Duration 2 days?
Start 10/22/21 8:00 AM
Finish 10/25/21 5:00 PM

Non Functional Requirements
Duration 2 days?
Start 10/26/21 8:00 AM
Finish 10/27/21 5:00 PM

Requirements Analysis Comp...
Duration 1 day?
Start 10/28/21 8:00 AM
Finish 10/28/21 5:00 PM

Initiation/Planning
Duration 9 days?
Start 10/29/21 8:00 AM
Finish 11/10/21 5:00 PM

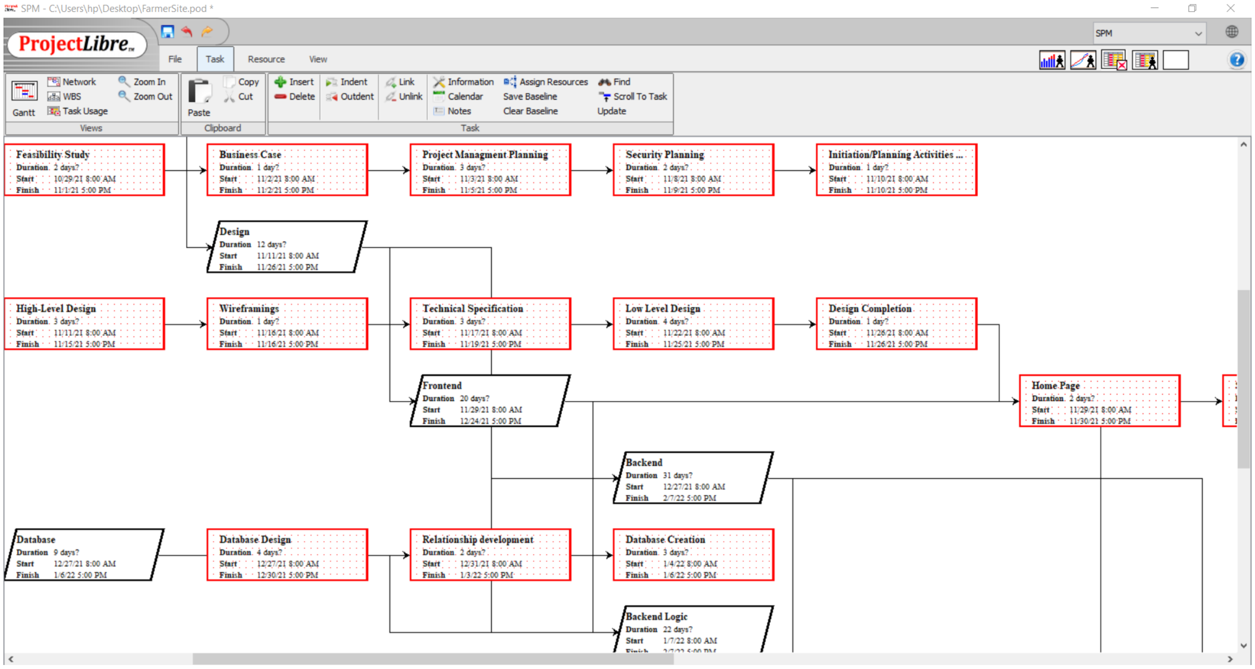
Feasibility Study
Duration 2 days?
Start 10/29/21 8:00 AM
Finish 11/1/21 5:00 PM

Business Case
Duration 1 day?
Start 11/2/21 8:00 AM
Finish 11/2/21 5:00 PM

Project Managment Planning
Duration 3 days?
Start 11/3/21 8:00 AM
Finish 11/5/21 5:00 PM

Security Planning
Duration 2 days?
Start 11/8/21 8:00 AM
Finish 11/9/21 5:00 PM

Design
Duration 12 days?



Feasibility Study
Duration 2 days?
Start 10/29/21 8:00 AM
Finish 11/1/21 5:00 PM

Business Case
Duration 1 day?
Start 11/2/21 8:00 AM
Finish 11/2/21 5:00 PM

Project Managment Planning
Duration 3 days?
Start 11/3/21 8:00 AM
Finish 11/5/21 5:00 PM

Security Planning
Duration 2 days?
Start 11/8/21 8:00 AM
Finish 11/9/21 5:00 PM

Initiation/Planning Activities ...
Duration 1 day?
Start 11/10/21 8:00 AM
Finish 11/10/21 5:00 PM

Design
Duration 12 days?
Start 11/11/21 8:00 AM
Finish 11/26/21 5:00 PM

High-Level Design
Duration 3 days?
Start 11/11/21 8:00 AM
Finish 11/15/21 5:00 PM

Wireframings
Duration 1 day?
Start 11/16/21 8:00 AM
Finish 11/16/21 5:00 PM

Technical Specification
Duration 3 days?
Start 11/17/21 8:00 AM
Finish 11/19/21 5:00 PM

Low Level Design
Duration 4 days?
Start 11/22/21 8:00 AM
Finish 11/25/21 5:00 PM

Design Completion
Duration 1 day?
Start 11/26/21 8:00 AM
Finish 11/26/21 5:00 PM

Frontend
Duration 20 days?
Start 11/29/21 8:00 AM
Finish 12/24/21 5:00 PM

Home Page
Duration 2 days?
Start 11/29/21 8:00 AM
Finish 11/30/21 5:00 PM

Backend
Duration 31 days?
Start 12/27/21 8:00 AM
Finish 2/7/22 5:00 PM

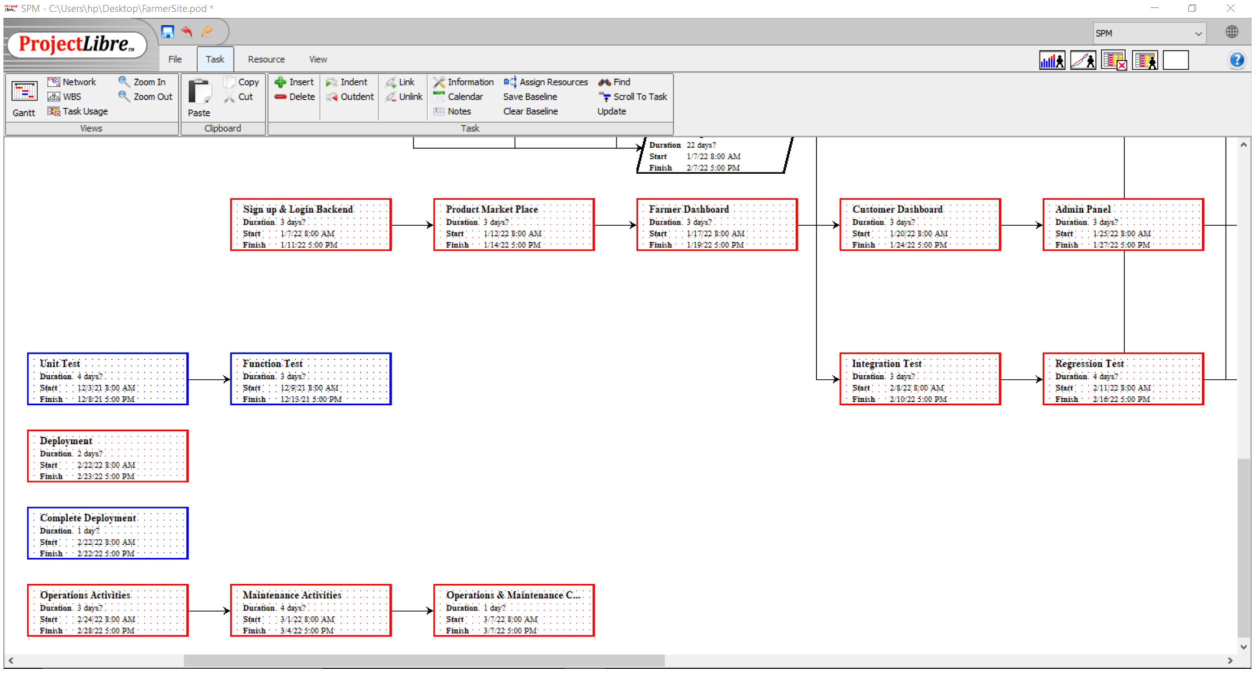
Database
Duration 9 days?
Start 12/27/21 8:00 AM
Finish 1/6/22 5:00 PM

Database Design
Duration 4 days?
Start 12/27/21 8:00 AM
Finish 12/30/21 5:00 PM

Relationship development
Duration 2 days?
Start 12/31/21 8:00 AM
Finish 1/3/22 5:00 PM

Database Creation
Duration 3 days?
Start 1/4/22 8:00 AM
Finish 1/6/22 5:00 PM

Backend Logic
Duration 22 days?
Start 1/7/22 8:00 AM
Finish 2/7/22 5:00 PM



Duration 22 days?
Start 1/7/22 8:00 AM
Finish 2/7/22 5:00 PM

Sign up & Login Backend
Duration 3 days?
Start 1/7/22 8:00 AM
Finish 1/11/22 5:00 PM

Product Market Place
Duration 3 days?
Start 1/12/22 8:00 AM
Finish 1/14/22 5:00 PM

Farmer Dashboard
Duration 3 days?
Start 1/17/22 8:00 AM
Finish 1/19/22 5:00 PM

Customer Dashboard
Duration 3 days?
Start 1/20/22 8:00 AM
Finish 1/24/22 5:00 PM

Admin Panel
Duration 3 days?
Start 1/25/22 8:00 AM
Finish 1/27/22 5:00 PM

Unit Test
Duration 4 days?
Start 12/3/21 8:00 AM
Finish 12/8/21 5:00 PM

Function Test
Duration 3 days?
Start 12/9/21 8:00 AM
Finish 12/13/21 5:00 PM

Integration Test
Duration 3 days?
Start 2/8/22 8:00 AM
Finish 2/10/22 5:00 PM

Regression Test
Duration 4 days?
Start 2/11/22 8:00 AM
Finish 2/16/22 5:00 PM

Deployment
Duration 2 days?
Start 2/22/22 8:00 AM
Finish 2/23/22 5:00 PM

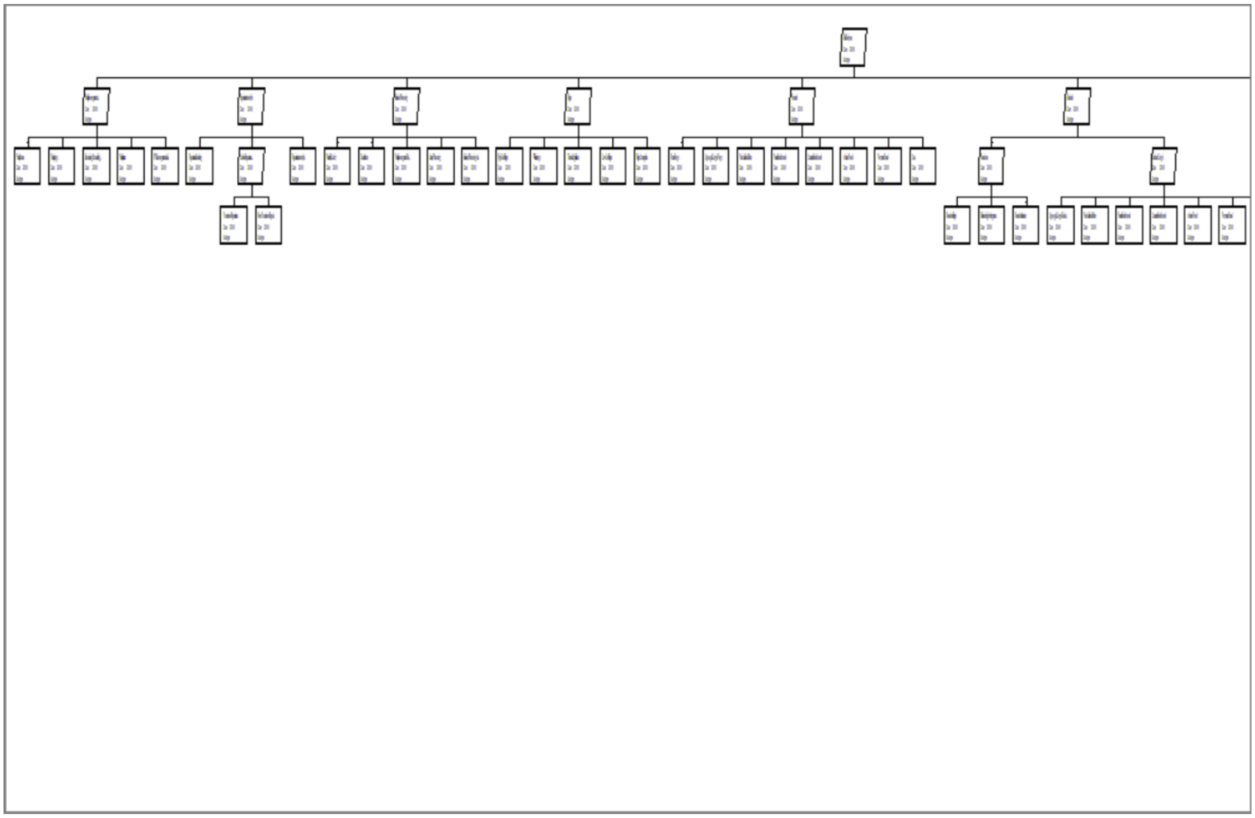
Complete Deployment
Duration 1 day?
Start 2/22/22 8:00 AM
Finish 2/22/22 5:00 PM

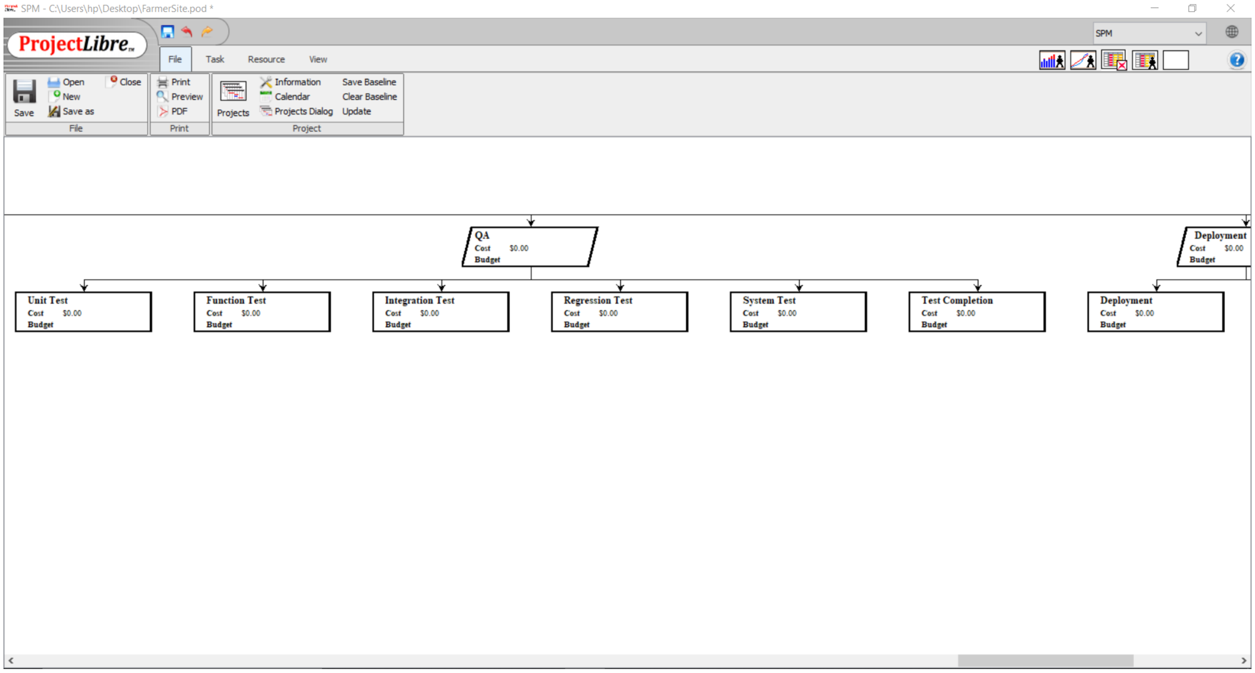
Operations Activities
Duration 3 days?
Start 2/24/22 8:00 AM
Finish 2/28/22 5:00 PM

Maintenance Activities
Duration 4 days?
Start 3/1/22 8:00 AM
Finish 3/4/22 5:00 PM

Operations & Maintenance C...
Duration 1 day?
Start 3/7/22 8:00 AM
Finish 3/7/22 5:00 PM

Work Breakdown Structure-





SPM - C:\Users\hp\Desktop\FarmerSite.pod *

QA
Cost $0.00
Budget

Deployment
Cost $0.00
Budget

Unit Test
Cost $0.00
Budget

Function Test
Cost $0.00
Budget

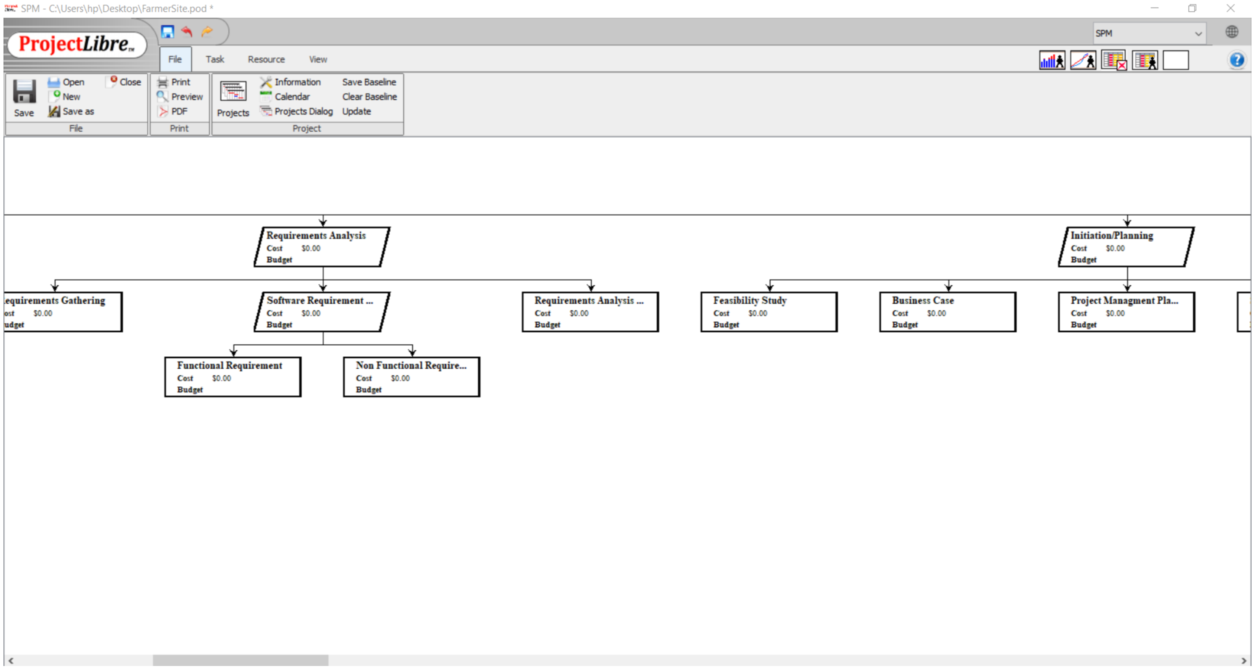
Integration Test
Cost $0.00
Budget

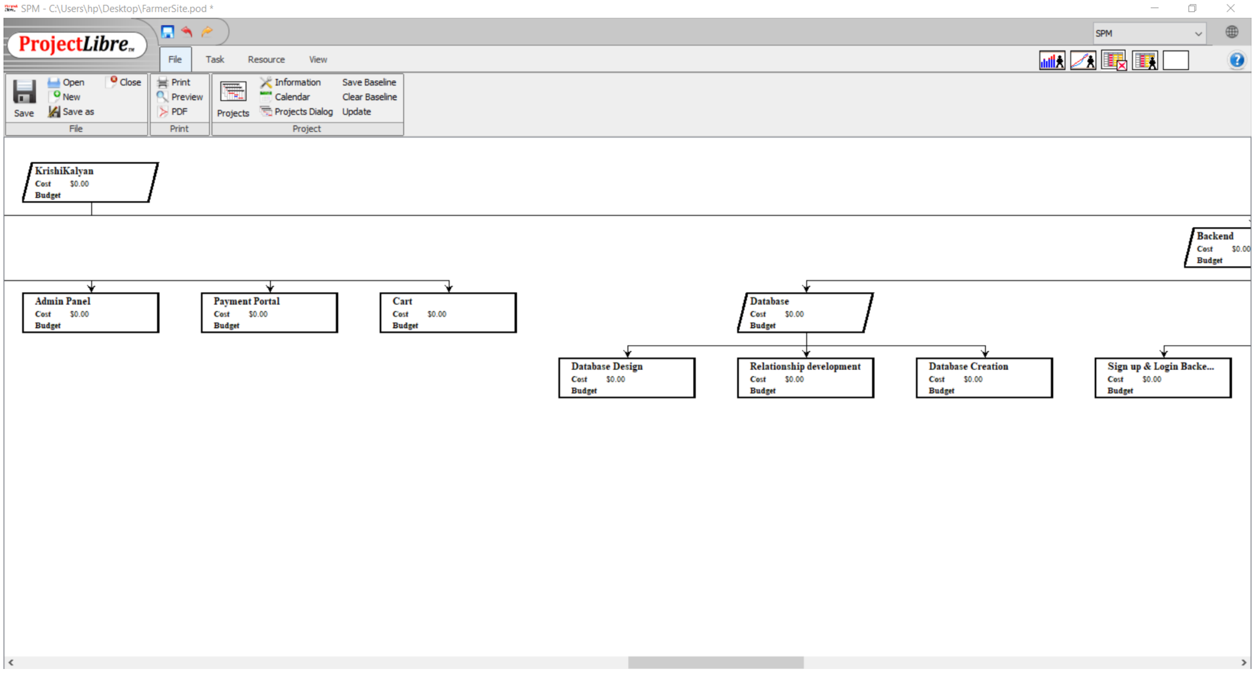
Regression Test
Cost $0.00
Budget

System Test
Cost $0.00
Budget

Test Completion
Cost $0.00
Budget

Deployment
Cost $0.00
Budget


SPM - C:\Users\hp\Desktop\FarmerSite.pod *
ProjectLibre
File
Task
Resource
View
SPM
Save
Open
New
Save as
Close
File
Print
Preview
PDF
Print
Projects
Information
Calendar
Projects Dialog
Save Baseline
Clear Baseline
Update
Project
Requirements Analysis
Cost $0.00
Budget
Initiation/Planning
Cost $0.00
Budget
equirements Gathering
ost $0.00
udget
Software Requirement ...
Cost $0.00
Budget
Requirements Analysis ...
Cost $0.00
Budget
Feasibility Study
Cost $0.00
Budget
Business Case
Cost $0.00
Budget
Project Managment Pla...
Cost $0.00
Budget
Functional Requirement
Cost $0.00
Budget
Non Functional Require...
Cost $0.00
Budget


SPM - C:\Users\hp\Desktop\FarmerSite.pod *
ProjectLibre
File
Task
Resource
View
SPM
Save
Open
New
Save as
Close
File
Print
Preview
PDF
Print
Projects
Information
Calendar
Projects Dialog
Save Baseline
Clear Baseline
Update
Project
KrishiKalyan
Cost $0.00
Budget
Backend
Cost $0.00
Budget
Admin Panel
Cost $0.00
Budget
Payment Portal
Cost $0.00
Budget
Cart
Cost $0.00
Budget
Database
Cost $0.00
Budget
Database Design
Cost $0.00
Budget
Relationship development
Cost $0.00
Budget
Database Creation
Cost $0.00
Budget
Sign up & Login Backe...
Cost $0.00
Budget