***COEPD Project 1 – submission by Aditi Gupta***

Online Agriculture Products Store

1. Identify Business Process Model for Online Agriculture Store – (**Goal, Inputs, Resources, Outputs, Activities, Value created to the end Customer**)

**Solution:**

1. **Goal:** Make a user-friendly mobile / web app to enable direct purchase transactions between farmers in rural areas and companies selling Agri-products
2. **Input**: Details of Agri products from manufacturers, purchase requests from farmers, internet connectivity and mobile/web devices to access internet, delivery channels, payment gateways, user information
3. **Resources:** APT IT team, product and sellers listing and information, investors
4. **Output:** On-time delivery of Agri products to farmers
5. **Activities:** Display information of various products, farmers browse and choose products, place it in cart, confirm order, choose payment method, complete payment, book the order, Order ID creation, fulfillment and delivery with order status tracking. Help and FAQs for support.
6. **Value created to the end Customer:** One-stop-shop that saves money and time: Ease of access to product and seller information on any device, access to variety of products and fair price-points, door-step delivery.
7. **SWOT analysis for Mr. Karthik**

* **Strengths:** Experienced team, funding, support from Mr. Henry.
* **Weaknesses:** Uncertain market demand, competition, serviceability to remote locations
* **Opportunities**: Addressing a significant need in the market, expanding to other regions.
* **Threats:** Technological challenges, regulatory issues, adoption by farmers

1. **Feasibility study on doing this project in Technology (Java)**

**Considerations:**

**Hardware/software requirements** –Java is a robust, scalable and secure language for app development, payment gateway integration and server deployment. Current office hardware can handle the tasks without issues.

**Availability of skilled resources** – 1 PM, 1 senior Java developer, 5 Java developers, 2 testers and 1 BA

**Budget** – INR 2 Cr is sufficient to kickstart the project

**Time frame** – 18 months is enough time to deliver the project

1. **Gap Analysis**

**AS-IS**

* Farmer face difficulties in procuring fertilizers which are very important for farm. They face the same problem in-case of buying seeds for farming certain crops. Lack of pesticides which could help in greatly reducing pests in crops.
* Limited options, traditional methods of procuring Agri products which might not be the best quality.
* Considerable time, efforts and money spent by farmers due to lack of accessibility and information.
* Limited connection with direct sellers.

**TO-BE**

* An online web/mobile app that gives access to information on multiple sellers and options of farming products.
* Price points, varieties, product information
* Door-step delivery
* Scalable service and application
* Direct procurement from manufacture
* Saves time, money and efforts for farmers

1. **Risk Analysis**

**BA Risks: If BA responsibilities are not executed properly:**

* Requirement gathering
* Stakeholder mapping
* Prioritization
* Elicitation techniques
* Project planning and execution
* Scope creep
* Testing & Implementation

**Project Risks:**

* Adoption by intended users
* Technical challenges
* Internet accessibility
* Limited budget and time
* Change in project team / stakeholders
* Correct information from all stakeholders

1. **Stakeholder Analysis (RACI Matrix):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stakeholder** | **Requirement Gathering** | **Design & Development** | **Testing** | **Deployment & Support** | **Decision Making** |
| **SOONY Committee (Mr. Henry, Mr. Pandu, Mr. Dooku)** | I | I | I | R | D |
| **Project Manager (Mr. Vandanam)** | A | C | C | C | R |
| **Business Analyst (Pratik Bartake)** | R | R | C | C | C |
| **Farmers (Peter, Kevin, Ben)** | C | C | I | I | I |
| **Delivery Head (Mr. Karthik)** |  | R | R | R | C |
| **Java Developers (Ms. Juhi, Teyson, Lucie, Tucker, Bravo)** |  | R |  |  |  |
| **Network Admin (Mr. Mike)** |  | R |  | R | C |
| **Database Admin (John)** |  | R |  | R | C |
| **Testers (Mr. Jason, Ms. Alekya)** |  |  | R | R | C |
| **Fertilizer/Seed/Pesticide Manufacturers** | C | I |  | I | I |

*SOONY Committee:* Provides overall project direction, approves budgets, and makes key decisions.

*Project Manager:* Oversees the project execution, coordinates resources, and ensures project deliverables meet deadlines and budgets.

*Business Analyst:* Leads requirement gathering, examines user needs, defines functionalities, and documents specifications.

*Farmers:* Provide user insights, participate in testing, and represent the target audience.

*Delivery Head:* Oversees development activities, manages the development team, and ensures technical feasibility.

*Development Team:* Responsible for building the application based on defined functionalities.

*Network & Database Admins:* Configure and maintain the network infrastructure and database for the application.

*Testers:* Conduct quality assurance testing to identify and report bugs.

*Fertilizer/Seed/Pesticide Manufacturers:* Provide product information for the platform and participate in testing vendor functionalities.

1. **Business Case Document - Online Agriculture Products Store**

|  |
| --- |
| **1. Executive Summary:** |
| Mr. Henry proposes the creation of an Online Agriculture Products Store to address the challenges faced by farmers in procuring agricultural products. This project aims to provide easy access to fertilizers, seeds and pesticides to farmers located in remote areas. This initiative is in line with SOONY's CSR goals and is entrusted to APT IT SOLUTIONS for execution. |
| **2. Project Overview:** |
| • Project Name: Online Agriculture Products Store • Sponsor: Mr. Henry (SOONY) • Project Committee: Mr. Pandu, Mr. Dooku, Mr. Henry • Project Manager: Mr. Vandanam • Delivery Head: Mr. Karthik • Duration: 18 months • Budget: 2 Crores INR |
| **3. Project Objectives:** |
| Build an online platform for farmers to purchase agricultural products. Facilitate direct communication between farmers and product manufacturers. Enhance accessibility and availability of fertilizers, seeds, and pesticides. |
| **4. Business Need:** |
| Farmers in remote areas face challenges in procuring essential agricultural products. The Online Agriculture Products Store aims to address this need, providing a convenient and efficient solution for both farmers and product manufacturers. |
| **5. Project Scope** |
| Development of a user-friendly web and mobile application. Integration with manufacturers for product listings and updates. Secure online transactions and delivery logistics. |
| **6. Stakeholder Analysis** |
| Key stakeholders include Mr. Henry, the Committee, APT IT SOLUTIONS team, farmers (end-users), and product manufacturers. Their roles and responsibilities are detailed in the RACI Matrix. |
| **7. Risks and Mitigation:** |
| Technical Risks: Potential challenges in development and implementation Mitigation: A thorough feasibility study and ongoing technical support Market Risks: Uncertainty in user adoption and market demand. Mitigation: Conducting market research, user feedback, and agile development for adaptability. |
| **8. Feasibility Study** |
| Considerations for feasibility include: Hardware and software requirements. Availability of trained resources. Budget constraints. Timeframe for development. |
| **9. Project Execution** |
| SDLC Methodology: Agile SDLC Models: V Model Phases: Requirements Gathering, Analysis, Design, Development, Testing, UAT, Deployment, and Implementation. |
| **10. Billing Model** |
| Fixed Bid model selected, with funds released based on timesheets submitted every two weeks. Quarterly audits planned for project progress assessment. |
| **11. Business Benefits** |
| Improved access to agricultural products. Increased efficiency in procurement processes. Enhanced economic opportunities for farmers. |
| **12. Conclusion:** |
| The Online Agriculture Products Store aligns with SOONY's commitment to social responsibility, addressing a critical need in the agricultural sector. The project's success will contribute to the well-being of farmers and the sustainable growth of the community. |
| **13. Approval:** |
| Approved by Mr. Henry, Mr. Pandu, and Mr. Dooku. |

1. **Four SDLC Methodologies**

**Sequential:** Linear, each phase must be completed before the next.

**Iterative:** Repeated cycles of development and testing.

**Evolutionary**: Continuous improvement and feedback-driven.

**Agile:** Incremental development, flexibility, and collaboration with stakeholders.

1. **Sequential (Waterfall) Methodology**

The Sequential Model, also known as the Waterfall Model, is a traditional and structured approach to software development. It follows a linear, step-by-step process where each phase must be completed entirely before moving on to the next. Here's a breakdown of its key characteristics:

**Phases in the Waterfall Model:**

Requirement Gathering: This phase involves collecting and documenting all the functionalities and features the project needs to deliver. Close collaboration with stakeholders (farmers, manufacturers) is crucial in this phase.

System Design: Based on the gathered requirements, a detailed system design is created, outlining the overall architecture, components, and functionalities.

Development: The actual coding and development of the application takes place based on the approved design specifications.

Testing: Once development is complete, rigorous testing is conducted to identify and fix bugs and ensure the application functions as intended.

Deployment: The final tested application is deployed to the production environment and made available to users (farmers).

Maintenance: Even after deployment, ongoing maintenance is required to fix bugs, address security vulnerabilities, and potentially implement new features as needed.

**Advantages of Waterfall Model:**

Clear Structure: The defined phases provide a clear roadmap for the project, making it easy to track progress and identify milestones.

Easy Project Management: Due to its structured nature, project management tasks like scheduling, resource allocation, and budgeting become more straightforward.

Suitable for Well-Defined Projects: For projects with clearly defined requirements and minimal anticipated changes, the Waterfall model can be efficient and effective.

**Disadvantages of Waterfall Model:**

Lack of Flexibility: Adapting to changes in requirements mid-project can be complex and disruptive. Reworking earlier phases might be necessary, leading to delays and cost overruns.

Limited User Feedback: User feedback is primarily incorporated during the initial requirement gathering phase. This can lead to potential issues with usability or functionality being discovered later in the development process.

Not Ideal for Evolving Requirements: If the project requirements are likely to change significantly throughout development, the Waterfall model might not be the best choice.

1. **Iterative Methodology**

The Iterative Model is a software development lifecycle (SDLC) methodology that emphasizes delivering working software in smaller, incremental releases. Unlike the Waterfall model's linear approach, the Iterative Model focuses on a more cyclical and adaptable process.

**Process:**

Requirement Prioritization: Project requirements are prioritized based on importance and feasibility.

Iteration Planning: A small set of high-priority features is selected for development within a defined timeframe (iteration).

Development & Testing: The development team focuses on building and testing the chosen features within the iteration.

Delivery & Feedback: The completed functionalities are delivered to stakeholders (farmers, SOONY committee) for feedback.

Iteration Review: Feedback from stakeholders is used to evaluate the delivered features and refine requirements for the next iteration.

**Advantages of Iterative Model:**

Early User Feedback: This approach allows for early user feedback after each iteration, enabling course correction and ensuring the delivered product aligns with user needs.

Adaptability: The Iterative Model is more flexible than the Waterfall model and can accommodate changes in requirements or priorities throughout the project lifecycle.

Reduced Risk: By delivering functionalities in smaller chunks, the project becomes less susceptible to major issues arising from unclear requirements later in development.

**Disadvantages of Iterative Model:**

Scope Creep Potential: The iterative nature can lead to scope creep if feature prioritization and iteration planning are not well-managed.

Documentation Challenges: Keeping documentation updated with evolving requirements can be challenging.

Requires Strong Project Management: Effective project management is crucial to ensure successful iteration planning and timely delivery of features.

1. **Evolutionary Methodology**

The Evolutionary Model in software development combines elements of both the Waterfall (Sequential) and Iterative Models. It offers a structured approach for projects with a core set of well-defined requirements but also the potential for future enhancements or evolving needs.

**Process:**

Initial Core Features: Similar to the Waterfall model, the initial phase focuses on defining and developing a solid foundation of core functionalities for the project. This initial development might follow a more sequential approach.

Incremental Delivery: Once the core features are established, the project transitions to an iterative approach. New features and functionalities are prioritized and delivered in smaller increments based on user feedback and evolving requirements.

Continuous Improvement: The project continuously evolves through these iterations, incorporating feedback and potentially adding new features as needed.

**Advantages of Evolutionary Model:**

Balance Between Structure and Flexibility: This model provides a structured foundation while allowing for ongoing adaptation to changing needs.

Suitable for Projects with Core Functionality: It's ideal for projects with a well-defined core but the possibility of future enhancements based on user adoption or market trends.

Reduced Risk in Core Features: By focusing on core functionalities initially, the project mitigates the risk of major issues arising from unclear requirements later.

**Disadvantages of Evolutionary Model:**

Clear Planning Required: Careful upfront planning is necessary to define the core functionalities and ensure their effective development.

Risk Management is Key: Managing the evolution of requirements and prioritizing new features effectively is crucial to avoid scope creep.

Potential for Delay: If the core functionalities are not well-defined initially, there's a risk of delays during the iterative phase.

1. **Agile Methodology**

Agile methodology is a software development approach that prioritizes flexibility, collaboration, and continuous improvement. It emphasizes delivering working software in short, iterative cycles known as sprints (typically 1-4 weeks).

**Core Principles of Agile:**

Individuals and Interactions over Processes and Tools: Agile values the human element and collaborative teamwork over rigid methodologies and tools.

Working Software over Comprehensive Documentation: Priority is given to delivering functional software early and often, with documentation evolving alongside the project.

Customer Collaboration over Contract Negotiation: Agile fosters close collaboration with stakeholders (farmers, SOONY committee) throughout the development process to ensure the project aligns with their needs.

Responding to Change over Following a Plan: While some planning is necessary, Agile embraces the ability to adapt to changes in requirements or priorities as the project progresses.

Agile Development Process:

Project Backlog: A prioritized list of all project requirements and features is created.

Sprint Planning: A team collaboratively selects a set of high-priority features for the upcoming sprint and defines acceptance criteria.

Development and Testing: The development team focuses on building and rigorously testing the chosen features within the sprint timeframe.

Daily Stand-up Meetings: Short daily meetings are held to keep everyone informed about progress, identify roadblocks, and adjust plans if needed.

Sprint Review: At the end of the sprint, the completed functionalities are demonstrated to stakeholders, and feedback is incorporated for future iterations.

Sprint Retrospective: The team reflects on the sprint's successes and challenges to identify areas for improvement in the next iteration.

**Advantages of Agile Methodology:**

Highly Adaptable: Agile readily accommodates changes in requirements or priorities, making it suitable for projects with evolving needs.

Early and Continuous User Feedback: Frequent feedback loops ensure the project stays on track and delivers value to users.

Increased Collaboration: The Agile approach fosters close collaboration between developers, stakeholders, and end-users.

Reduced Risk: Delivering working software in short sprints helps identify and address issues early, minimizing overall project risk.

**Disadvantages of Agile Methodology:**

Requires Strong Team Skills: Successful Agile implementation requires a team with good communication, self-organization, and problem-solving skills.

Not Ideal for Fixed Requirements: If the project requirements are completely fixed and unchangeable, Agile might not be the most efficient approach.

Documentation Challenges: Agile prioritizes working software over extensive documentation, which might need to be adapted throughout the project.

1. **SDLC Models:**

**Waterfall:** Linear, sequential, and rigid.

**RUP (Rational Unified Process):** Iterative and flexible.

**Spiral:** Iterative with risk assessment at each cycle.

**Scrum:** Agile, iterative, and collaborative.

**Waterfall**: A waterfall model is a traditional model in IT Company, the waterfall model is a classical model used in system development life cycle to create a system with linear and sequential approach. In this model software development done from one phase to another phase in download manner, output of one phase used as a input for next phase, every phase has to completed before next phase starts and here is no overlapping of the phases. it is a progressive implementation of the project which is divided into different phases of SDLC. As waterfall models have few limitations, still it was used earlier on a wide range.

**RUP Model:** Stands for Rational Unified Model This is a software development process from rational, a division of IBM, it divides the development process into four distinct phases that each involve business modelling, Analysis and design, Implementation, testing and deployment, In RUP there are four project life cycles

A) Inception

B) Elaboration

C) Construction

D) Transaction

**Spiral:** This phase starts with gathering of business requirements in the subsequent spirals as the product matures identification of system requirement are done in this phase. This also includes understanding of system requirement by continual communication between customer and the analyst at the end of the spiral the product is deployed

Design: Design phase starts with the design in the baseline spiral and involves architectural, logical design of modules, physical product design and final design in the successive spirals.

Construct: Construct phase refers to development of the final software product at every spiral. In the spiral when the product is just thought and the design is being developed, a Proof of Concept (POC) is developed in this phase to get the users’ feedback. Then in the successive spirals with higher clarity on requirements and design a working model of the software called build is developed with a version number. These versions are sent to the users for feedback.

Evaluation and Risk Analysis: Risk analysis includes identifying, estimating, and observing technical feasibility such as schedule slippage and cost overrun. After testing the build, at the end of first iteration, user evaluates the software and provides the feedback. Based on the customer assessment, development process enters into the next iteration and afterwards follows the linear approach to implement the feedback provided by the user. The process of iterations along the spiral carries on with throughout the life of the software.

**Scrum**: Scrum is not a process technique or definitive method, rather it is a framework within which you can employ various processes and technique. It has three roles and every role has clear accountability. The product owner is responsible for maximizing the products value resulting from the development team work.

The Scrum model suggests that projects progress via a series of sprints. In keeping with an agile methodology, sprints are time boxed to no more than a month long, most commonly two weeks. Scrum is a lightweight agile process framework used primarily for managing software development. Scrum is often contrasted with the so-called “Waterfall” approach, which emphasizes up-front planning and scheduling of activities, followed by execution

The scrum models have 5 steps also called phases in scrum.

Step 1: Product Backlog Creation.

Step 2: Sprint planning and creating backlog

Step 3: Working on sprint.

Step 5: Retrospective and the next sprint planning

As a BA I would propose the use of V-Model for the project. As V-model is one of the most important model that is used in the process of software development. V-model is a sequential process in which the next phase begins only after the completion of the present phase. In this model, steps don’t move in a linear way while the steps are bent upwards. It is similar to Waterfall model because we follow V-model from left to right as well as follow a sequential path of execution of processes like as in waterfall model. in waterfall model steps are followed as requirements, design, implementation, verification and finally maintenance. In the same way, the same steps are followed in V-model. We can say that V-model is the alternate evolved version of the waterfall model.

1. **Difference between Waterfall and V-Model**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Waterfall Model** | **V-Model** |
| **Development Approach** | Sequential, linear | Sequential with synchronized testing |
| **Testing Approach** | Testing happens after development phases are complete | Each development phase has a corresponding testing phase |
| **Flexibility** | Less flexible, difficult to adapt to changes | More adaptable than Waterfall |
| **Focus** | Project completion and meeting initial requirements | Verification & Validation (ensuring product meets requirements and functions as intended) |
| **Suitability** | Projects with well-defined, stable requirements | Projects with well-defined requirements and emphasis on testing |
| **Analogy** | Series of steps, one after another | V-shaped ditch, development on one side, testing on the mirror side |

1. **Justification for V-Model**

The V-model emerged as the preferred choice for this project due to its iterative development approach that prioritizes verification and validation at each stage. This stands in contrast to the waterfall model, which follows a more linear sequence. The V-model's emphasis on concurrent testing alongside development phases proved to be a significant advantage. By incorporating testing activities like planning and test design early in the process, the V-model fosters a more iterative and efficient development lifecycle. This is particularly beneficial for projects like this one, where the scope is well-defined and the likelihood of major requirement changes is relatively low.

Furthermore, the V-model's ability to accommodate adjustments mid-cycle offers valuable flexibility. Should new requirements or modifications arise during the development process, the test documents can be readily updated to reflect these changes. This adaptability minimizes rework and ensures that the testing phase remains aligned with the evolving product specifications.

In conclusion, the V-model's emphasis on early testing, iterative development, and adaptability to changing requirements makes it a compelling choice for developing the online agricultural product store. This project's well-defined scope and manageable size further solidify the V-model's suitability, promoting a streamlined development process and increasing the project's overall success rate.

1. **Gantt Chart**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Start Date** | **End Date** | **Duration** |
| Requirements Gathering | 04-01-2024 | 28-02-2024 | 55 |
| Requirements Analysis | 18-02-2024 | 24-03-2024 | 35 |
| Design | 14-03-2024 | 09-05-2024 | 56 |
| Development 1 | 29-04-2024 | 11-07-2024 | 73 |
| Testing 1 | 01-07-2024 | 15-09-2024 | 76 |
| Development 2 | 05-09-2024 | 19-11-2024 | 75 |
| Testing 2 | 09-11-2024 | 23-01-2025 | 75 |
| Development 3 | 13-01-2025 | 29-03-2025 | 75 |
| Testing 3 | 19-03-2025 | 08-06-2025 | 81 |
| Development 4 | 29-05-2025 | 29-07-2025 | 61 |
| Testing 4 | 19-07-2025 | 18-09-2025 | 61 |
| UAT | 08-09-2025 | 09-10-2025 | 31 |

1. **Fixed Bid Vs Billing**

**Fixed Bid**

A Fixed Bid is a project billing method where a predetermined, flat fee is agreed upon upfront for the entire project scope. This means the client pays the agreed-upon amount regardless of the actual hours spent by the service provider to complete the project.

Key Characteristics:

Predefined cost: The total price is fixed before work begins, based on an estimated project scope and effort required.

Defined deliverables: A clear outline of what will be delivered at the project's end is crucial for accurate cost estimation.

Set timeline: Fixed timelines are usually established alongside the cost, with milestones often included.

**Advantages:**

Predictable costs: Clients know exactly how much they will pay upfront, aiding in budget management.

Faster project initiation: Without lengthy negotiations about hourly rates, projects can potentially start sooner.

Focus on efficiency: Service providers are incentivized to complete the project within the agreed-upon timeframe and budget.

**Disadvantages:**

Risk for service providers: Underestimation of project complexity or unforeseen issues can lead to the service provider absorbing additional costs.

Limited flexibility: Changes to the project scope after the bid is agreed upon might require renegotiation of costs and timelines.

Less suitable for uncertain projects: Fixed Bids are less ideal for projects with evolving requirements or high levels of uncertainty.

Fixed Bids are used when:

Well-defined scope and requirements

Established technology stack

Experienced development team for accurate estimation

Clients who prioritize budget predictability

Alternative Billing Method: Time and Materials (T&M):

**Billing**

In contrast to Fixed Bids, Time and Materials (T&M) billing charges clients based on the actual time spent by the service provider's team on the project.

Billing is used for projects with:

Unclear or evolving requirements

New or unproven technologies

Need for ongoing flexibility and change

Ultimately, the choice between Fixed Bid and T&M billing depends on the specific project characteristics, risk tolerance, and priorities of both the client and the service provider.

|  |  |  |
| --- | --- | --- |
| **Feature** | **Fixed Bid** | **Billing** |
| **Purpose** | Defines the total project cost upfront | Process of sending an invoice for services rendered |
| **Cost Structure** | Flat fee for the entire project | Can be based on Fixed Bid, Time and Materials, or Milestones |
| **Flexibility** | Limited - Changes to scope may require renegotiation | More flexible - Can adapt to changing project needs (depending on billing method) |
| **Risk** | Higher risk for service provider if project complexity is underestimated | Higher risk for client if project scope changes significantly (T&M) |
| **Client Preference** | Predictable costs | May be preferred for ongoing projects with evolving requirements (T&M) |

1. **BA Timesheet for RG, RA, Design, Development, Testing, UAT, Deployment & Implementation at $60/hr**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Requirement Gathering- Timesheet** | | | | | |
| **Sr.** | **Objective** | | **Date** | **Start Time** | **End Time** | **Duration (hr)** | **Gross Pay $** |
| 1 | Stakeholder Identification | | 4-01-24 | 08:00 | 13:00 | 5.00 | 300 |
| 2 | Establish project goal & objectives | | 5-01-24 | 09:00 | 19:00 | 10.00 | 600 |
| 3 | Elicit requirements from stakeholders | | 8-01-24 | 09:00 | 22:00 | 13.00 | 780 |
| 4 | Requirement documentation | | 12-01-24 | 09:00 | 15:00 | 6.00 | 360 |
| 5 | Requirement confirmation & Sign-off | | 13-01-24 | 20:00 | 22:00 | 2.00 | 120 |
| 6 | Requirement prioritization | | 14-01-24 | 09:00 | 17:00 | 8.00 | 480 |
|  |  | |  |  |  | **44** | **2640** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Requirement Analysis- Timesheet** | | | | | |
| **Sr.** | **Objective** | | **Date** | **Start Time** | **End Time** | **Duration (hr)** | **Gross Pay $** |
| 1 | Requirement Verification | | 16-01-24 | 09:00 | 15:00 | 6.00 | 360 |
| 2 | Requirement Validation | | 17-01-24 | 09:00 | 15:00 | 6.00 | 360 |
| 3 | Requirement Architecture | | 18-01-24 | 09:00 | 19:00 | 10.00 | 600 |
| 4 | Requirement allocation- RTM | | 20-01-24 | 09:00 | 14:00 | 5.00 | 300 |
| 5 | Define Solution Scope | | 21-01-24 | 09:00 | 20:00 | 11.00 | 660 |
|  |  | |  |  |  | **38** | **2280** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Design- Timesheet** | | | | | | | | | | |
| **Sr.** | **Objective** | | | **Date** | | **Start Time** | | **End Time** | | **Duration (hr)** | | **Gross Pay $** |
| 1 | Translate BRD into functional requirements | | | 22-01-24 | | 09:00 | | 12:00 | | 3.00 | | 180 |
| 2 | Define UI Requirements | | | 23-01-24 | | 09:00 | | 15:00 | | 6.00 | | 360 |
| 3 | Solution prototype development | | | 24-01-24 | | 09:00 | | 14:00 | | 5.00 | | 300 |
| 4 | Design workshop facilitation | | | 25-01-24 | | 09:00 | | 11:00 | | 2.00 | | 120 |
| 5 | Develop system flowcharts | | | 26-01-24 | | 09:00 | | 13:00 | | 4.00 | | 240 |
|  |  | | |  | |  | |  | | **20** | | **1200** |
|  |  | | |  | |  | |  | |  | |  |
|  | | **Development- Timesheet** | | | | | | | | | | |
| **Sr.** | **Objective** | | **Date** | | **Start Time** | | **End Time** | | **Duration (hr)** | | **Gross Pay $** | |
| 1 | Project planning | | 28-01-24 | | 09:00 | | 11:00 | | 2.00 | | 120 | |
| 2 | Client Meeting- Development updates | | 29-01-24 | | 09:00 | | 12:00 | | 3.00 | | 180 | |
| 3 | Documentation of systems and processes | | 30-01-24 | | 09:00 | | 14:00 | | 5.00 | | 300 | |
| 4 | Change requirement facilitation | | 31-01-24 | | 09:00 | | 10:00 | | 1.00 | | 60 | |
| 5 | Facilitate DevOps doubt clarification | | 1-02-24 | | 09:00 | | 10:00 | | 1.00 | | 60 | |
|  |  | |  | |  | |  | | **12** | | **720** | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Testing- Timesheet** | | | | | |
| **Sr.** | **Objective** | | **Date** | **Start Time** | **End Time** | **Duration (hr)** | **Gross Pay $** |
| 1 | Test Planning | | 2-02-24 | 09:00 | 12:00 | 3.00 | 180 |
| 2 | Test Case Development- Scenarios | | 3-02-24 | 09:00 | 11:00 | 2.00 | 120 |
| 3 | Test Case Review | | 4-02-24 | 09:00 | 12:00 | 3.00 | 180 |
| 4 | Facilitation of Testing and clarifications | | 6-02-24 | 09:00 | 11:00 | 2.00 | 120 |
| 5 | Documenting and interpreting Testing Report | | 7-02-24 | 09:00 | 12:00 | 2.00 | 120 |
|  |  | |  |  |  | **12** | **720** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **UAT- Timesheet** | | | | | |
| **Sr.** | **Objective** | | **Date** | **Start Time** | **End Time** | **Duration (hr)** | **Gross Pay $** |
| 1 | UAT Planning- Participants, test case, documents | | 8-02-24 | 09:00 | 13:00 | 4.00 | 240 |
| 2 | Facilitation & Support- User guide, feedback | | 10-02-24 | 09:00 | 12:00 | 3.00 | 180 |
| 3 | Defect Mgmt.- Prioritise defects and track | | 12-02-24 | 09:00 | 11:00 | 2.00 | 120 |
| 4 | Facilitation of Testing and clarifications | | 13-02-24 | 09:00 | 12:00 | 3.00 | 180 |
| 5 | Documenting and interpreting Testing Report | | 14-02-24 | 09:00 | 11:00 | 2.00 | 120 |
|  |  | |  |  |  | **14** | **840** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Deployment & Implementation- Timesheet** | | | | | |
| **Sr.** | **Objective** | | **Date** | **Start Time** | **End Time** | **Duration (hr)** | **Gross Pay $** |
| 1 | Deployment plan, user documentation, migration | | 01-04-24 | 09:00 | 13:00 | 4.00 | 240 |
| 2 | RTM review and communication to client | | 25-04-24 | 09:00 | 12:00 | 3.00 | 180 |
| 3 | User training manual & user training sessions | | 1-05-24 | 09:00 | 11:00 | 2.00 | 120 |
| 4 | Gather User feedback & report | | 15-08-24 | 09:00 | 12:00 | 3.00 | 180 |
| 5 | Deployment documentation and client signoff | | 31-09-24 | 09:00 | 11:00 | 2.00 | 120 |
|  |  | |  |  |  | **14** | **840** |