**BCDV1001 DESIGN PATTERNS FOR BLOCKCHAIN**

****

**MICROFINANCE LOANS**

**Team Members**

Shafin Rizvi

Hennadii Ahanesian (101275688)

Neeraj Bhasin (101263021)

Raymond

Table of Contents

Requirements Analysis…………………………………………………………...............3

Problem……………………………………………………………………………..3

Goal………………………………………………………………………………….3

Stakeholders………………………………………………………………………..3

State data……………………………………………………………………………3

Restrictions………………………………………………………………...............3

Exception……………………………………………………………………………3

Architectural Document……………………………………………………………...…….4

Architectural Design…………………………………………………………….....4

Flow Diagram……………………………………………………………………….4

State Diagram………………………………………………………………...........5

Data Definations……………………………………………………………………5

Project Plan……………………………………………...………………………………….6

Project Decomposition………….………………………………………….. …….6

Time Estimate………………………………………………………………………6

Staffing………………………………………………………………………………7

Decomposition………………………………………………………………..........7

Cost & Time Estimate……………………………………………………………...7

**Requirements Analysis**

**PROBLEM**

Microfinance is a banking service provided to unemployed or low-income individuals. People want to build a platform that provides microfinance loans. However, they have to face two main issues:

1. Expensive traditional audit fee (The amount of loan is small, not worth it for traditional audit).
2. Truthless intermediary.

**GOAL**

The objective is to create a lending and borrowing platform without any third-party interference and with no-collateral deposit required to borrow the loan. It will be a peer-to-peer platform which will have a small gas fee and all the transaction will store in an immutable database.

**STAKEHOLDERS**

|  |  |
| --- | --- |
| NAME | DESIGNATION |
| Shafin Rizvi | Development Team |
| Hennadii Ahanesian | Project Manager |
| Neeraj Bhasin | Development Team |
| Raymond | BusinessAnalyst |
| User | (Lender/Borrower) |

**STATE DATA**

* Debt (type->struct:structure to store debt details).
* Lend and debt history (mapped with debt and lend IDs).
* Loan State (Requested, funded, and paid).
* Storing date&time of transactions.

**RESTRICTIONS**

* Customers cannot approve the request of a loan to own address or address(0).
* Customers can’t get a new loan before paying the previous one

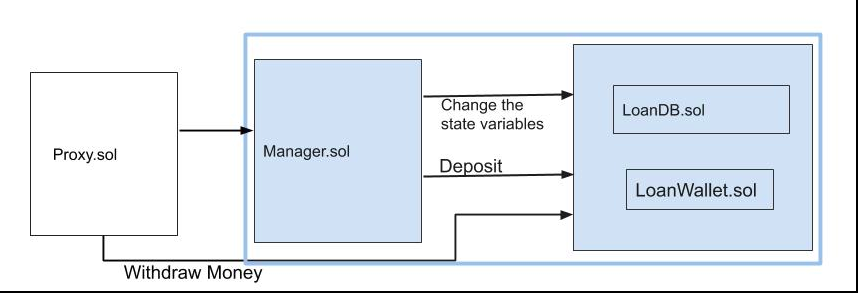
**EXCEPTIONS**

* Reputation point tokens will be decreased if the borrower doesn’t pay the loan back.
* The interest rate can vary according to the number of reputation tokens the customer is holding.

**ARCHITECTURE DOCUMENT**

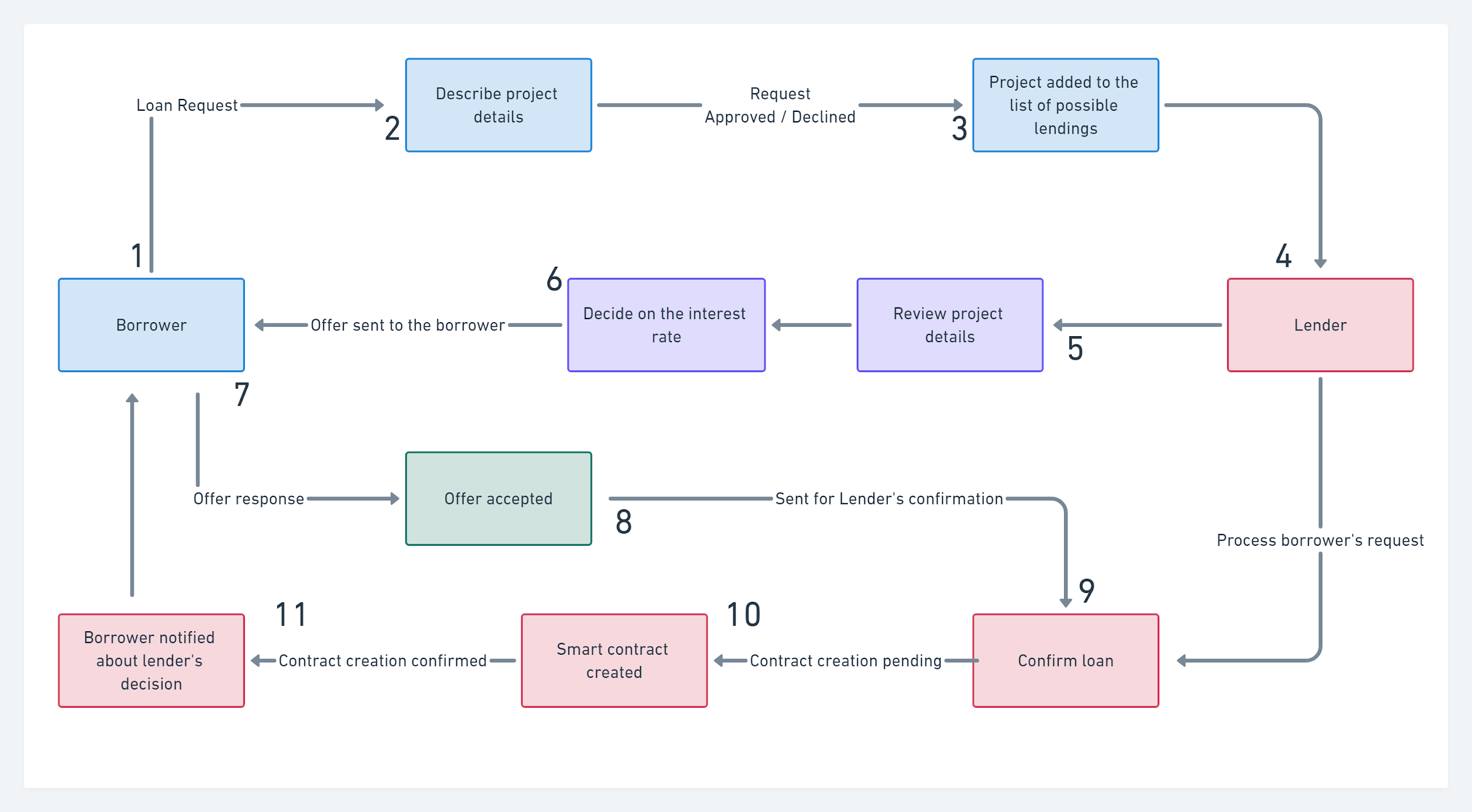
**ARCHITECTURE DESIGN:**

* This the architecture of our smart contracts. Where we have a proxy contract that talks to the user on the behalf of other contracts.
* LoanManager is the decision-maker who checks all the requirements and then approves the payment.
* LoanDB is having all the state variables that record all the information and responsible for storing on a blockchain.
* LoanWallet is responsible for deposit and withdrawal functions.

****

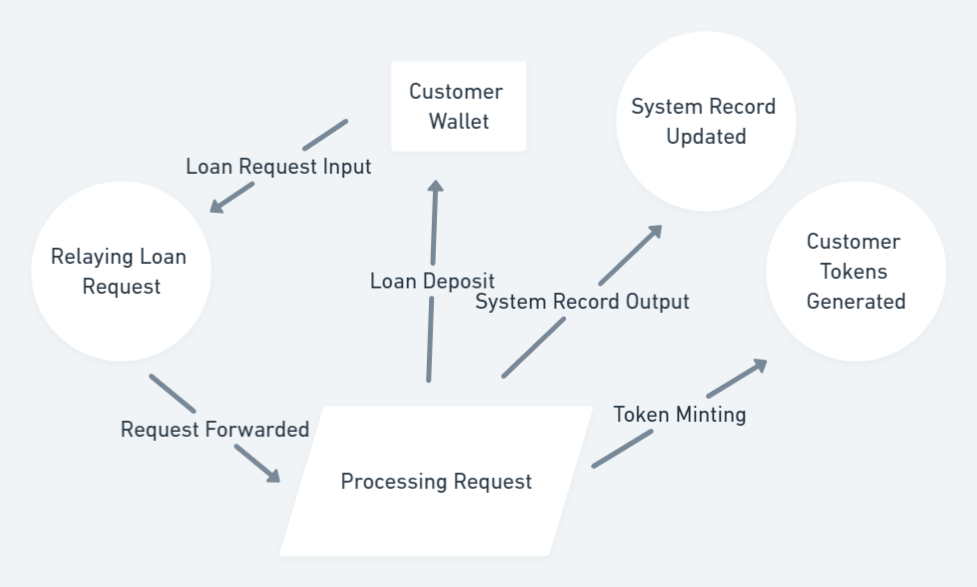
**FLOW DIAGRAM**

This represents the complete flow of our project from starting to end.

****

**STATE DIAGRAM**

This represents the flow of data/information through the state variable. This will be going to store permanently into the blockchain.

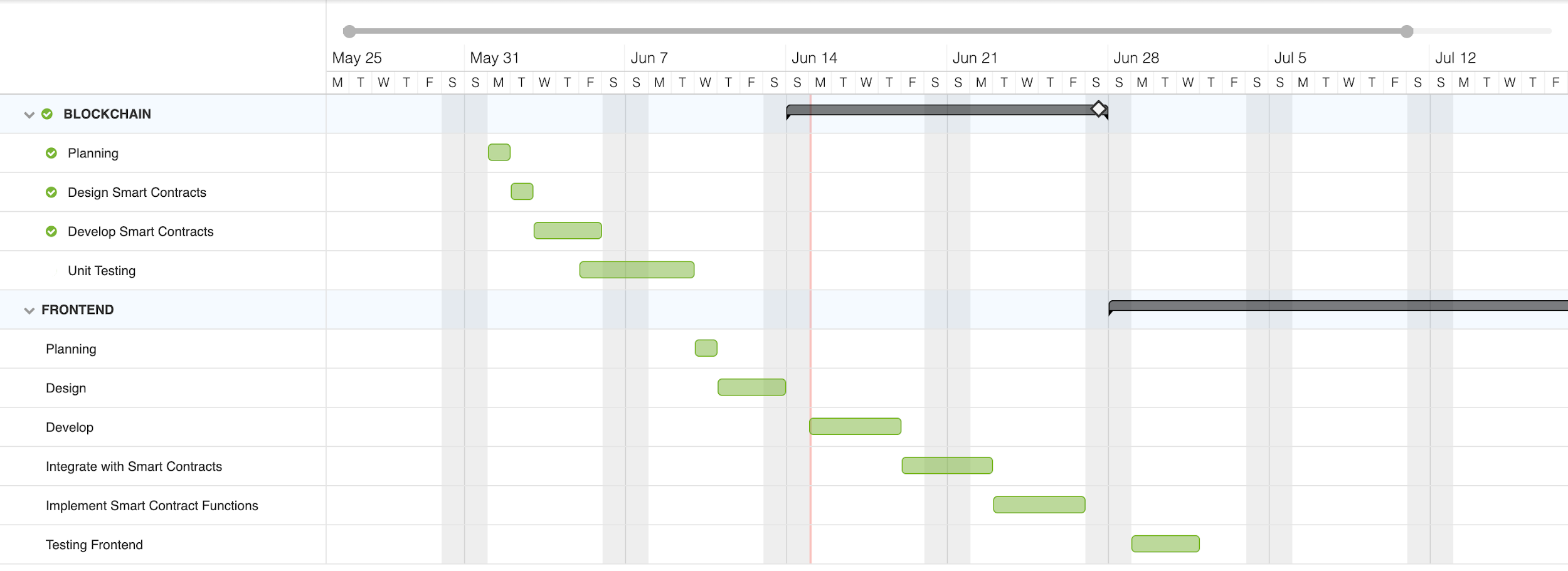


**DATA DEFINITION**

|  |  |  |  |
| --- | --- | --- | --- |
| **NAME** | **TYPE** | **STRUCTURE** | **PURPOSE** |
| Debt | struct | {uint id, address borrower, address lender, uint amount, uint interest, uint loanState} | Structure to store the debt details |
| debtInfo | mapping | (uint => debt) | Mapping of debt with Debt struct |
| debtHistory | mapping | (address => uint[]) | Mapping of borrower address with debt id |
| lendHistory | mapping | (address => uint[]) | Mapping of borrower address with debt id |
| haveDebt | mapping | (address =>bool) | Mapping the borrower address with true/false value |

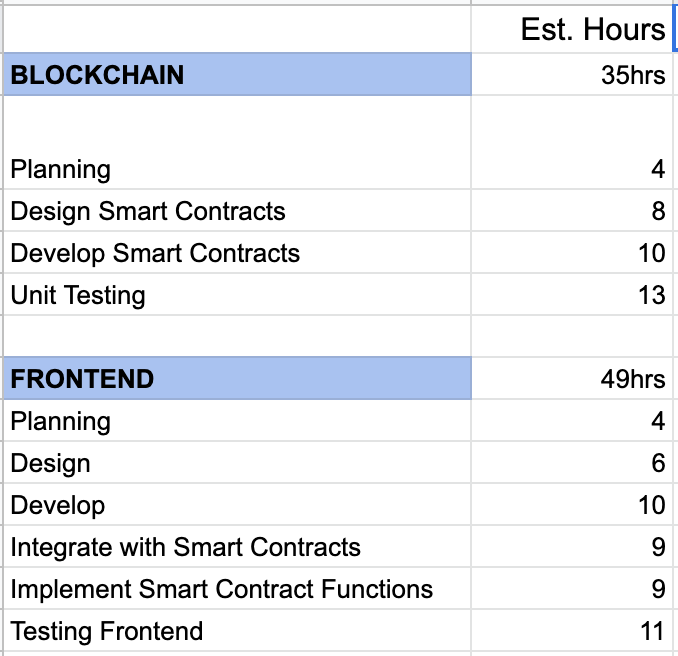
**PROJECT PLAN**

**PROJECT DECOMPOSITION**: Decomposed all the major tasks and sub-tasks. Assigned them with the start and end date for each task.



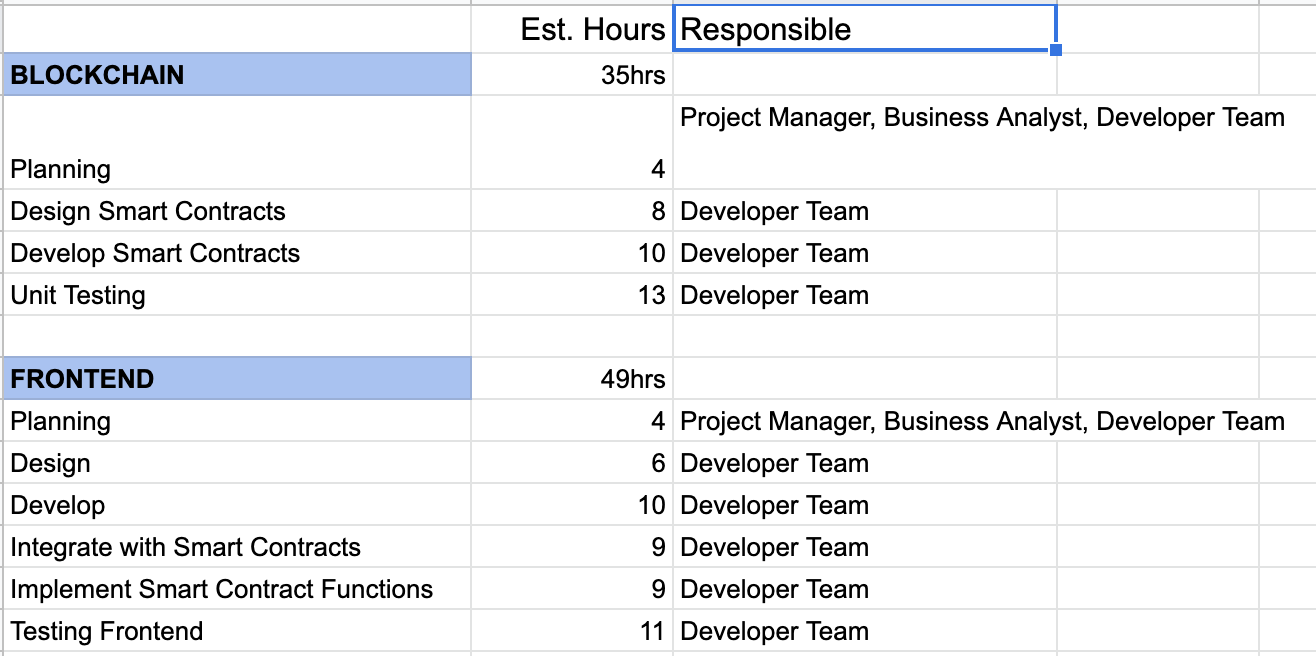
**TIME ESTIMATE**

These are the estimated hours we have calculated for every subtask we require to complete.



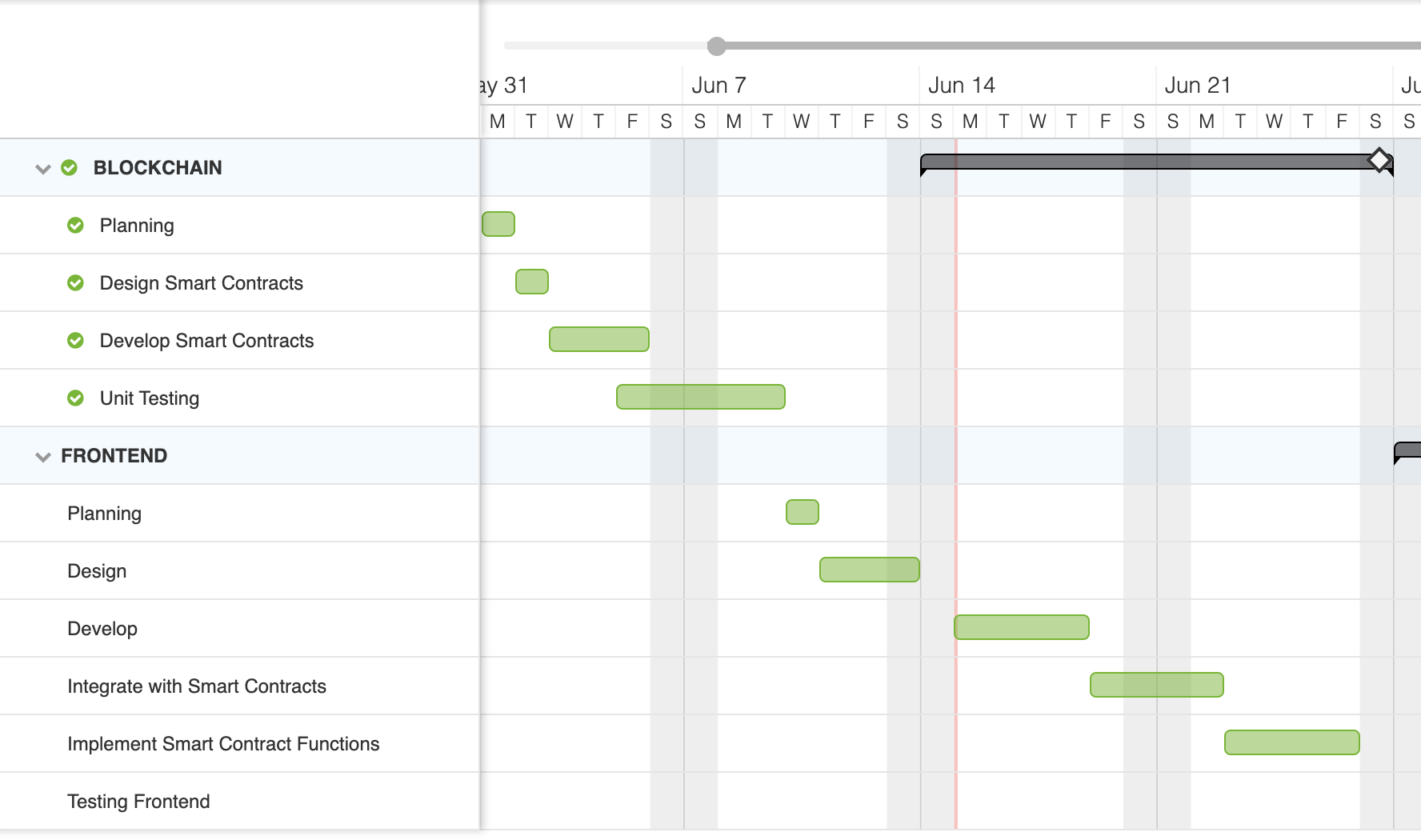
**STAFFING**

The planning phase will be done by all the members and then other phases like designing, development, and testing will be done by the developer team.

****

**DEPENDENCIES**

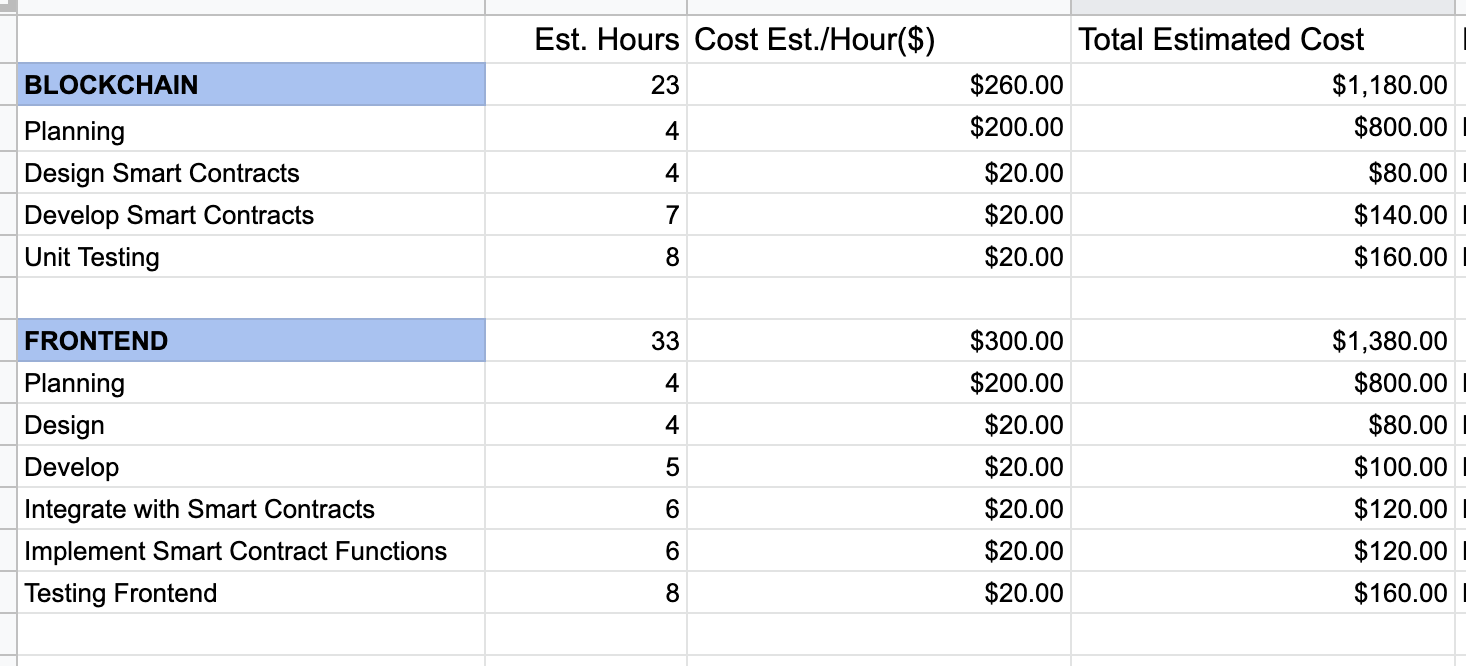
These all are our project tasks on which our project is based.

****

**COST & TIME ESTIMATE**

The hourly wage rate is as follows:

* Project Manager- $120/hr
* Business Analyst- $120/hr
* Developer Team- $20/hr

****