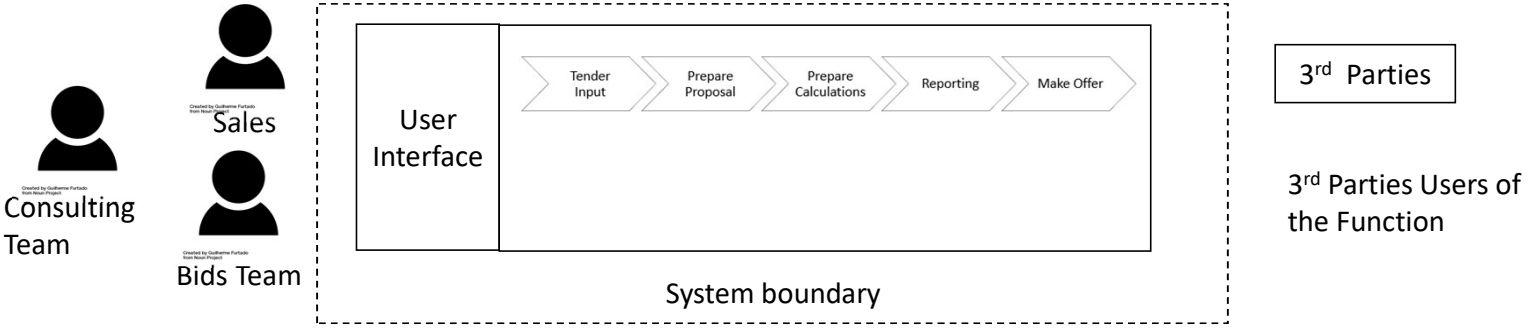
Bid and Proposals Management System

Architecture Overview

Context | Scope | Customer Exp | Goals & System Objectives | Key Entities| Scenarios | Layered View | Interactions

Business Context : Bid and Proposals System

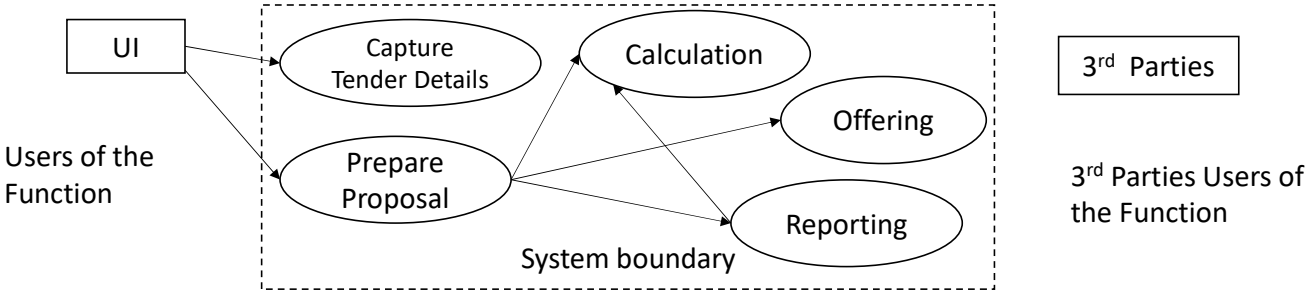


Bid and Proposal Management System

The Request For Proposal is prepared by the organization raising the tender. The Sales, Consulting and Bids Team work together to make the best bid for the Request. The objective of the Sales Team is to win the bid. The bids team focus is on rightly pricing it, while sales engineer collects the initial data and the consulting team is focused on supporting them to succeed with the bid.

The details are stored and associated with a RFPID. We start preparing a proposal with details of base case and additional details of Revenue, tax, Expenses and Calculate the possible values of NPV and IRR for the project. Positive NPV in 3 years indicates high probability of project success. Multiple proposals are prepared with different combinations and calculations done. The best is selected. The Comparison report shows the comparison across proposals. It is left for us to select one Proposal.

Scope Diagram : Object Process Model



Bid and Proposal Management System

Tender is identified and details are collected. These include

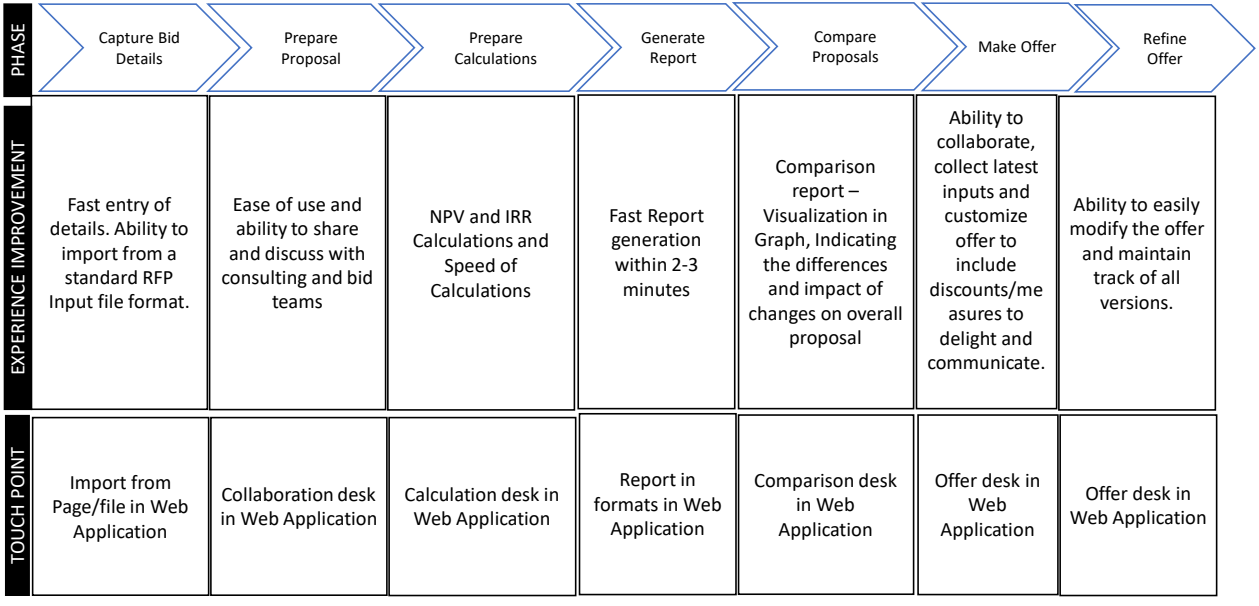
1. Request For Proposal is prepared
2. Details could include Location, Department and Types of Functions to be supported.
3. Acceptance criteria

The details are stored and associated with a Tender Identification Number.

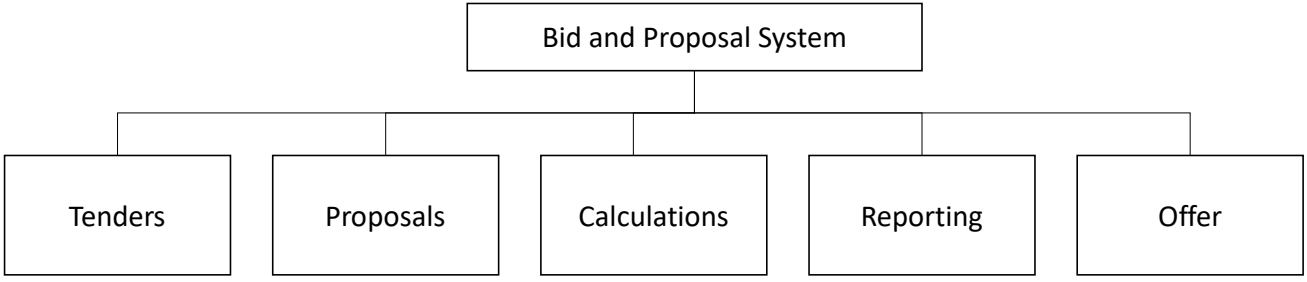
We start preparing a proposal with details of base case and additional details of Revenue, tax, Expenses and Calculate the possible values of NPV and IRR for the project. Positive NPV in 3 years indicates high probability of project success. Multiple proposals are prepared with different combinations and calculations done. The best is selected. The Comparison report shows the comparison across proposals.

It is left for us to select one Proposal.

Customer Experience



Key Entities/Subsystems Identified



Of these Calculations seem to be more performance oriented and risky to make modifications in the future. In other Cases like tenders, proposals are data capture more and storage. All proposal selection and reporting might use some kind of calculation. Therefore we identify Calculation as a significant element.

It could as well depend on the architect's experience and background. If the architect is highly experienced in a calculation engine, possibly he/she might opt for looking at Reporting as one major element or Proposals and focus on the customer experience at these elements.

List of Scenarios

Scenario Id	Scenario Title	Scenario Description	Expected Result
#1	Version Management of Proposals.	- Multiple Proposal Versions exist and need to be managed and associated with right person and role.	Version Management of Proposal without user manually specifying.
#2	Collaboration and Workflows of Proposals.	- Workflows of assigning a part of proposal to Consulting Engineer. - Approval of Proposal and signoff by all team is essential.	Workflow and which state and person associated with it. Approval and Status displayed.
#3	Responsive Calculations	Calculation needs to be responsive and need to be within 1 minute. 20 Calculations in 1 minute.	All Calculations complete and displayed on page under 1 minute. The Graphs needs to be plotted and ready in 1 minute
#4	Ability to Extend and Add more Calculations	Reports generated and comparisons created in 1 minute. The Formatting in formats of pdf and excel or html to be ready in 1 minute.	Comparisons under 30 seconds. Report generation in rest of 30 seconds
#5	Generate Reports in varied formats	Reports generated and comparisons created in 1 minute. The Formatting in formats of pdf and excel or html to be ready in 1 minute.	Comparisons under 30 seconds. Report generation in rest of 30 seconds

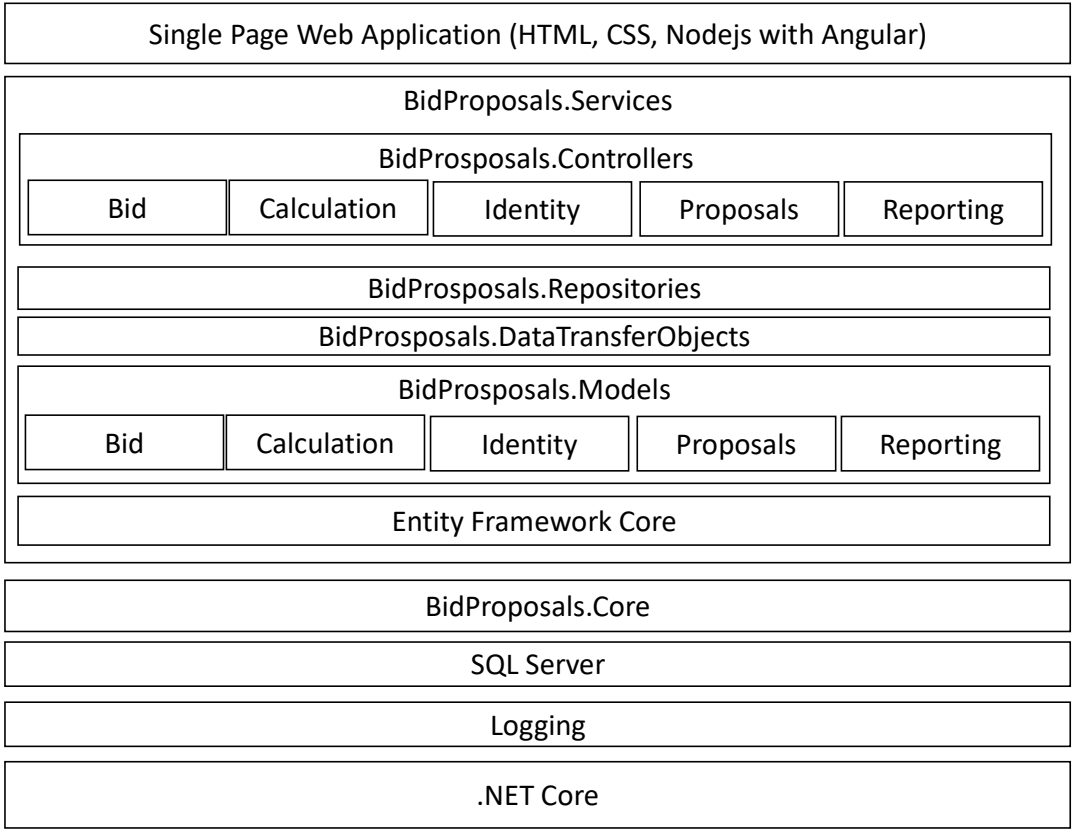
Early Technical Decisions

Scenario ID	Description	Acceptance Measure
Source Control	GITHUB	Lifecyle
Language and Framework	C#, .NET Core	Portability
Standardized/Integration with Systems	Standard RESTful approach	Open Architecture
Build System	GitHub build system	Open , Easy to Use

Technology Decisions in a Microsoft based Approach

Element and Role	Technology Options	Selection and Rationale
Web Application	Traditional ASP.NET Web Application using HTML, CSS, JQuery or Single Page Application with HTML, CSS, Angular or Nodejs	Single Page Application using Nodejs. Highly responsive and modular approach.
Services	WCF RESTFul, ASP.NET Web API	ASP.NET Web API considering the focus on Web and Cloud.
Data Access Layer	ADO.NET Entity Framework	Native with SQL Server.
Database	PostgreSQL, SQL Server, MySQL	SQLServer considering native to Microsoft and also the ease of migration and satisfying the transaction and storage needs.

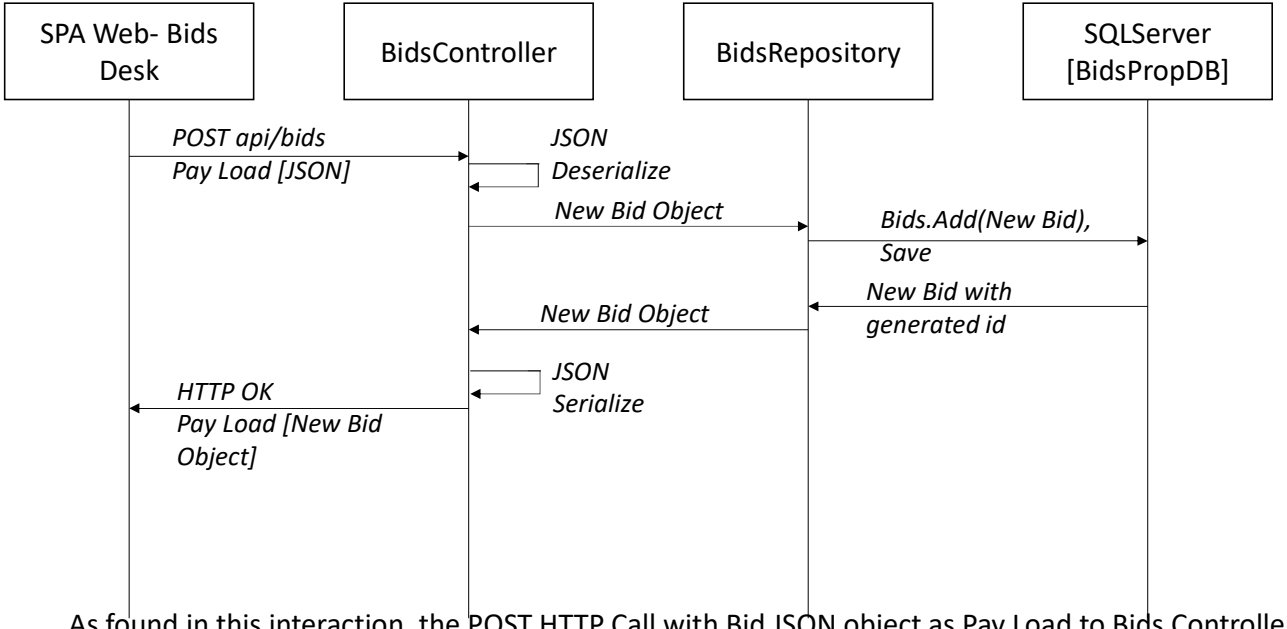
Layered View



The Layered View represents the Structure of the Solution.

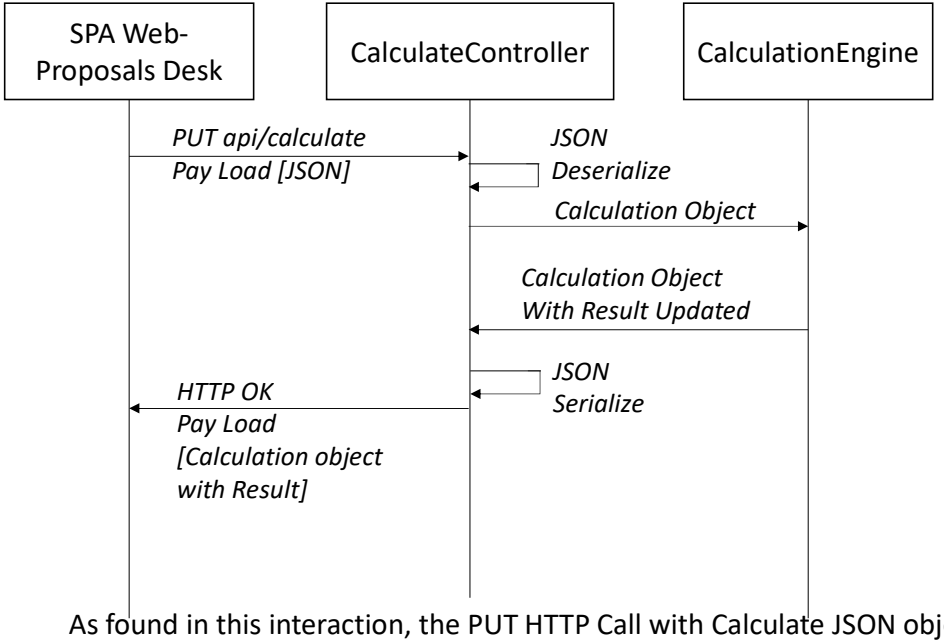
The Layers include SQL Server as Database Layer, Entity Framework serving as Data Access (Object Relational Model).
DataTransferObjects are meant to have the ORM Objects mapped to PresentationObjects. The Repositories provides access to Controllers to perform data operations/ They implement IRepository interface.
The Controllers defined per Service say like BidController provides the endpoint and expose RESTful Services over HTTP.
The Single Page Application using NodeJs and Angular makes calls to Service, and update the page.

Interactions – Sequence Diagram of A New Bid creation



As found in this interaction, the POST HTTP Call with Bid JSON object as Pay Load to Bids Controller leads to deserialization and A Bid domain Object is created and passed to Repository, which in turn calls Add on the Bids collection and invokes Save. The new Bid object is created with a new identifier. This object is further mapped to a Data Transfer object which is prepared and sent together with HTTP 200 as Pay Load in JSON.

Interactions – Sequence Diagram of A Calculation



As found in this interaction, the PUT HTTP Call with Calculate JSON object as Pay Load to Calculate Controller leads to deserialization and A Caculate domain Object is created and passed to CalculationEngine. The Calculation Engine, now once the Result is updated provides the object updated and this is passed as Pay Load together with HTTP OK.

List of Scenarios

Scenario Id	Scenario Title	Expected Result	Evaluation on Candidate Architecture
#1	Version Management of Proposals.	Version Management of Proposal without user manually specifying.	Version Management handled with Database support.
#2	Collaboration and Workflows of Proposals.	Workflow and which state and person associated with it. Approval and Status displayed.	State Management with states managed as part of Entity. Approval Process Workflow in SPA.
#3	Responsive Calculations	All Calculations complete and displayed on page under 1 minute. The Graphs needs to be plotted and ready in 1 minute	Calculation to be responsive would need Asynchronous methods. One possibility is to use Actors Pattern using Akka.NET. A Prototype developed and tried with 2 calculations and tried.
#4	Ability to Extend and Add more Calculations	Comparisons under 30 seconds. Report generation in rest of 30 seconds	Comparison analysis involves analysis of scores and can be handled using async methods.
#5	Generate Reports in varied formats	Comparisons under 30 seconds. Report generation in rest of 30 seconds	Reports are Asynchornously generated and once prepared shared in format.