Tony Vinayak

tony.vinayak@gmail.com

Abstract

Functional specifications for Roark (code name for PSA product)

psa ‘Roark’  
Features description

Contents

1 Time Tracking 2

1.1 Consultant logging time 2

1.2 Time Approval 2

1.3 Reporting 2

2 Expense Reporting 3

2.1 Consultant Entering Expense Report 3

2.2 Managerial Review and Approval of Expense Report 3

2.3 Billing Expense Reports to Customers 4

3 Billing 4

4 Customer Relationship Management 4

Appendix A: Wireframes 6

# 1 Time Tracking

There are two major aspects to time tracking component:

* Make it VERY EASY for consultants to track their time. Make the interface brain-dead simple. It should be accessible on the web as well as mobile devices
* Approval process, once the consultant has logged in their time.
* Reporting component, to churn out timesheets that are required every now and then – e.g., weekly status reports to be sent to customers, or timesheets that need to accompany invoices.

## 1.1 Consultant logging time

For Consultants to enter their time, the interface should allow them to:

* Choose the **Customer Name** from a searchable picklist. As they type in characters, pick list continues to shortlist the matching customer names
* Choose the **Project Name** for the chosen customer name. Again this should be a searchable picklist. The picklist will show only the currently “open” projects. Projects are “closed” once they have been fulfilled and invoiced (projects can be re-opened by Manager role)
* Choose **Service Item** from a searchable pick list. The pick list only shows the service items that are defined for that particular project. This tracks all the way back to the opportunity definition where service items would have been defined. For example, a project may include 100 hours of “Standard Consulting”, 20 hours of “Project Management”, and 8 hours of “Instructor-led training”. Service items are defined in Price Book, along with their billable rates.
* Enter a **comment** that describes what work was done
* Enter **Time**. Granularity of time should be a configurable. For example, some business will want it to be in minimum one-hour chunks, while others may tolerate fractional hours (e.g. 30 minute) chunks too.
* Choose whether this time is **billable** or not. By default, this option is selected to be billable.

The time-entry form also displays overall status of the project that includes:

* Various Service Items that are included for that particular project
* Maximum permissible hours for each of those service items
* How many hours remain to be consumed.

## 1.2 Time Approval

When the Consultants submits the time entry form, it triggers an email to the designated Project Manager. Project Manager can log into the system and review the timesheets that need to be reviewed and approved. Approval is recorded for each time entry. Only approved time entries can be subsequently billed to the customer. There is no need for any further workflow between Project Manager and the Consultant for the approval process. That is, just keep it simple – Project Manager needs to record their approval of each time entry.

## 1.3 Reporting

Reporting on timesheets should be fairly standard functionality – pick the customer and project, specify a date range (e.g., time logged for previous week or previous month), or include all dates, and generate report. Report can be previewed on the web page, exported in CSV format, or saved as a PDF file. Report elements and formatting would be pretty standard:

* Customer name in the header
* Project name in the header
* Indicated the total budgeted time for each Service Item that is on the SOW
* Include chronological details of time logged by various consultants for various service items
* Also include rate and total amount for each time entry
* Include subtotals and grand totals

# 2 Expense Reporting

There are three major aspects to expense reports:

* For consultants to enter their expenses. Keep the interface DEAD SIMPLE, and available via web and mobile devices
* For Manager to review and approve the expense reports
* For Admin to include the billable data from expense reports into customer invoices

## 2.1 Consultant Entering Expense Report

The interface – both on the web and on mobile devices – should include:

* Pick Customer Name from a searchable picklist
* Pick Project Name from a searchable pick list
* Specify date
* Specify amount
* Upload image of receipt. For mobile app, offer option to take a picture of the receipt.
* Specify category. This includes standard pre-defined categories such as Transportation, Meals, Lodging, Other. Etc.
* Specify currency. Default to USD.
* Specify if the amount is reimbursable to the consultant. For example, if it has been paid with a company credit card, then it won’t be reimbursable to the consultant
* Specify if the amount is billable to the customer

Once the consultant submits the expense report, the designated project manager receives and email alert for reviewing and approving the report

## 2.2 Managerial Review and Approval of Expense Report

The manager receives a standard email, and logs into the system to review the expense report. Manager has the ability to edit the expense report if needed. Once the Manager approves the expense report, it is marked as “Approved”. Only the approved expense reports are eligible for billing to customers. Expense report also have overall statuses:

* Billable? This is set if there is any line item on the expense report that needs to be billed to a customer.
* Billed? This indicates whether or not the expense report has been included in an invoice to the customer. Actual accounts receivables activity is part of the Financial Accounting hence beyond the scope of this product.
* Paid? This indicated whether or not the expense report has been paid by customer.
* Reimbursed? This indicates whether or not any amount due to the consultant has been paid or not.

## 2.3 Billing Expense Reports to Customers

As part of the billing workflow (in Billing module), when the Admin previews pending billing for a customer for a specified time period any unbilled expense reports for that customer show up along with any unbilled time. The Admin would then go ahead and generate invoice and mark the expense report as “Billed”.

# 3 Billing

The Billing module allows the Admin or Manager to preview billing details and generate invoices. Admin would generate billing details for a given customer for a given time period (see wireframe for an example). Billing details include:

* Any billable time that has not yet been billed
* Any billable expense report that has not yet been billed

The Admin (or Manager) would review the details, make any adjustments (by over-riding any entries) and trigger the billing process by generating invoices. Invoices are generated as PDF files and emailed to customers, and status of each billable item (time or expense) is marked as “Billed”. Subsequently these billable items can be updated as “Paid” once the payment has been received from the customer (will require interfacing with Accounting department). Billing status for each invoice is recorded against each customer (i.e., Invoice number, billing date, billing amount, billing status).

# 4 Customer Relationship Management

This module borrows from the functionality found in popular CRM systems such as Salesforce.com. This includes:

* Maintaining information regarding **Accounts** – name, industry, street address, mailing address, website, etc.
* **Contacts** within those Accounts – name, title, email address, phone number, etc.
* Account **Owner** – typically a sales person
* **Opportunities** for that particular account. An Account can have many opportunities. Each opportunity has a set of items from the Price Book sold to that customer. Each opportunity goes through a lifecycle, defined by various stages:
  + Prospecting (10% chance of closing)
  + Qualifying (25% chance of closing)
  + Engaged (50% chance of qualifying)
  + Finalizing paperwork (75% chance of closing)
  + Closed Won
  + Closed Lost
  + Closed Dead

Main reason for maintaining the above information is to track Opportunities, and answer questions like:

* How much business are we forecasting for next quarter?
* How much business have we already booked?
* How much business did we book last year, and out of that how much has been billed?
* What proportion of the booked business still needs to be delivered?

Roark provides functionality to maintain and report on above information. In a subsequent release Roark may also sync with other CRM systems. Other typical CRM functionality such as case management is outside the scope of Roark.

For each opportunity, some additional information is also maintained:

* Closing date
* Pointer to SOW (in the repository)
* Pointer to Consulting Services Agreement (in repository)
* Pointer to Purchase Order from the customer (stored in repository)
* Next step
* Notes

When an Opportunity is marked as “Closed Won”, it automatically creates a **Project** to track delivery. Project will have the following attributes pre-populated from the Opportunity:

* Customer Name
* Project Name (same as the Opportunity Name)
* Pointer to SOW
* Pointer to Consulting Services Agreement
* Pointer to Purchase Order from the customer
* Service Items, their quantities, and their rates. These are the ones against which the project delivery will be tracked.

# 5 Digital Assets

Roark stores a variety of documents in its underlying database. These are the documents used created during project execution. Some examples:

* Templates for various SOWs and contracts
* Templates for various invoices
* Executed legal documents (signed SOWs, POs etc.)
* Customer-specific deliverables: project reports, software, etc.
* Expense report receipts
* Copies of Invoices send to customers

This repository is indexed and searchable. When a document is added to the repository, certain tags can also be added to assign some additional metadata. Document contents (for .doc, .pdf etc.) are also searchable. Contents from the repository manifest themselves at various parts of Roark – e.g., pointer to SOW

# Appendix A: Wireframes





















