

Strategic Machines, Inc.

Agreement with Juan Carlos Gonzalez

March 12, 2018

For professional services:

Workitem registration page and process

A working prototype will be developed in accordance with the following specifications:

- Refactor webmachine platform to support a process for an administrator to register new workitems to marketplace. The webmachine routes for api/github will be used by this process
- the completed process will also include updated docs for the administrator to follow. The docs should be written at a high level and provide general guidance on the purpose and process for workitem management
- the capability should permit full CRUD of workitems – supporting the management of workitems across any repo that exists within the authorized account. Note:
 - workitem data captured is defined by the workitem schema. If in the course of this development effort other properties are identified that should be included (or excluded) lets discuss. But the general philosophy is to keep this 'lightweight' and permit developers to locate work suited for their skill level for execution
 - a workitem can only be posted to a repo if the repo (representing projects or skill categories in the strategic market) already exists. If the repo is does not exist, the admin separately through github would need to create it
 - workitems posted to a valid github repository would
 - be recorded in the mongodb db (based on config parameters)
 - updated in the issues on the correct github repo
 - be maintained (changed or deleted) through the same set of webpages
 - be viewable through the workitem reactjs pages that you already developed
- the capability also permits management of 'lifecycle stages' of the workitem. (open > assigned > closed)
- Appropriate tests are needed to ensure that CRUD and lifecycle operations work together in delivering a reliable app that is managing data and transactions across workitem, partner, client and github databases. (note – I think we can simplify the partner schema – lets discuss) Tests would need to demonstrate that workitems progress through the stages, and that they are appropriately rendered on the 'view screens' based on their assigned stages. For example, a workitem assigned, then unassigned would be presented on the assigned webpage, then the open item webpage.
- Assigned workitems should be also provide a link to the partner, and will be a hotlink to the partner gallery page (developed later under a separate workitem)
- All workitems being assigned to a partner would be validated against the partner db that the individual is authorized to take work. If a workitem is unassigned, then the databases are updated accordingly
- A partner completing a workitem has their digital reputation and payments data updated accordingly. An http transaction with payments data is triggered with required information for

making payment to a partner (stub out the http end point. The completion of this transaction cycle will be handled under separately scoped work)

- Items deleted would be marked in the mongo db with a Boolean – and be ‘logically deleted’ from the db – and closed on the github repo (should there be a comment inserted that the issue was closed but not completed to distinguish from closed and completed?).
- All workitem numbers assigned will be consistent with github identifiers, permitting validation that the two sites are synchronized
- For workitems completed and approved, an http transaction will be issued updating the ‘partner database’, workitem database and github accordingly with the completed transaction info (post dates etc). This will need to be carefully managed so that a failure to update in a collection is caught – permitting some kind of resolution on the transaction to avoid referential integrity issues (lets discuss options here to keep this simple)
- A workitem completed and approved can be marked as such from the workitem screen
- A workitem can also be marked as assigned from the workitem screen, capturing the partner identification info and marking the expected delivery date. This http api and route needs to be configured so that it can also be triggered as well by a virtual agent (microservice) where workitem assignment is being handled via text (the virtual agent is separately scoped)
- A simple authentication process is triggered in order to access workitem registration webpages (lets discuss – may be as simple as github oauth – but must be configured so only authorized personnel gain access – will need to be the foundation for all auth processes on the platform)
- The reactjs pages will render information using a clean, crisp format, such as github cards (zeit.co will continue to serve as the ‘style guide’ to ensure the look and feel of the page are lightweight and have a clean look for developers)
- The code will be appropriately documented and structured. App design decisions will be consistent with app architecture of the webmachine platform.
- Tests will be included
- Code will be delivered through pull requests on the strategicmarket github site
- Expected delivery date of final deliverable is targeted for March 26, 2018
- You agree that this is work for hire, work will be done in a quality manner, all deliverables are owned by Strategic Machines, Inc, and the work or concepts will be treated confidentially and not be disclosed to another party without express written permission of Strategic Machines
- This work order can be terminated by either party at anytime

Payment: \$1200 upon delivery of the outcome as specified above

Agreed to:

Date: