UNIVERSITY OF ALBERTA

Meeting Agenda and Minutes

Client Requirements Meeting 2
Sept 23, 2022
1:30 pm to 2:30 pm
Remote

Agenda Items

1. New requirements discussion / Q&A

Attendees:

Jensen Khemchandani
Peter Qiu
Ronggang Cui
Matthew Sheydwasser
Leen Alzebdeh
Victor Nguyen
Tomas Peschke
tchristie@zerorampup.com
jotoole@zerorampup.com

New requirements discussion / Q&A

Notes:

- Dev in a box – template of what is getting implemented - product benign sold
- Instance - implementation of a dev in a box (user can have many instances - environment is the same as instance from our perspective I think)
  - From a user's perspective its an ‘environment’ – use instance and environment pretty much the same
- User vs administrator vs developers
  - Users and developers are the same thing?
    - A dev in a box can has many diff users that can access it
    - An administrator of an instance is someone who can carry out CRUD actions
      - Could granular permission settings? Not a requirement would be a nice to have
-

- Office in a box?
    - A different system – can be connected to a dev in a box - don't worry about it
- Cancelling subscription:
    - There is an online/offline state for dev in a box
    - Pause the billing when offline?
    - When you kill an instance – terminating an instance – throwing it away
        - Takes some time to actually delete everything on their end
            - Has a pending status for delete – can still display in this time
            - When 'deleted' status - remove from the dashboard
    - Can you cancel a subscription without terminating a box?
        - No they are super tightly coupled – when cancelled, it means the instance is deleted immediately. – don't need to worry about 'hanging subscriptions' – when a subscription is cancelled, Stripe will give a partial refund based on the time left.
            - Look at Stripe API before making that assumption about the partial refund
    -
- API:
    - getDevInaBoxSolution()
        - Dev in a box is a type of solution - solutions have variables
        - That code represents the structure of one variable: name is name of variable, value is default value of a variable and etc.
        - Match each value in the specified YAML to this structure for each of them
    - How is an instance updatable?
        - What should be updatable by an admin?
            - Number of users
                - Number of developers being assigned to a box
                - We only need to assign number of developers (workspace users) - change billing based on number
                - As admin, do we have to specify exactly which git accounts? No - just the number of people should be configurable on our app – user management will be handled by gitlab
                - Let's say we have 10 users – admin goes in and reduces it to 5 – what happens to the extra 5 that were connected?
                    - Leave that to gitlab

            - Use office in a box
            - Office in a box name

- Organisations:
    - Idea behind an organisation?
        - A person would want to be able to work for multiple groups and companies, and be able to sort / organise their environment by those groups they are part of.
        - Billing is associated with an org !! – not a person!
            - Organisation is linked to an admin – admin will put in the billing information FOR an organisation – different credit cards for different organisations!
        - When someone makes an account – they should automatically have an organisation made for them?
            - Maybe not automatically! – Maybe have an easy way to do one or the other
            - Should be part of a default organisation – that they would need to add payment details to – name it – make a default name whatever
                - Immediately ask for a credit card and such?
            - If someone just wanted to join other orgs – get a user account - be thrown into the default organisation - when logged in, you have to be logged in for a specific org! – (dropdown of all the orgs we are a part of and switch out the stuff on the dashboard!) - could be the admin of one org, but a user of another.
            - We don't need to force people to include billing information right on sign up. We can prompt for it, but can say 'set up later' or something.
        - An org can have multiple admins but at least one! – can promote other users to administrator and bring you down to normal user
            - As an admin I should be able to do anything basically
        - Invite 'links' can work on both existing and non existing accounts – can be just a shared link – still make people make their own default org if they don't already have one.
            - If we create an invitation – make it a one time use link – then make the link inactive or unusable.
- Should we have internal representation of Dev in a box inside our application?
    - If we want to have some sort of persistence (to test associations between users and administrators and such) – maybe in a redis database or something easy just for testing – circle back to this later as we get into the nitty-gritty of developing

- Always create an instance - ensure that call works – before charging the user very last! – any time that we charge the user make sure we do all the 'risky stuff' first

- Colour scheme / theme is undecided – clients will follow up and get back to us on this.
- Do we want a landing page? About section? Corporate?
    - Clients haven't talked about this yet – clients will follow up and get back to us on this

- Leen is done with project overview - PR is up - merge when reviewed
- User stories are completed by Matthew and Tomas for sprint 1
- High level software design is good and merged in dev
- Low fidelity user interface is done – Roman will create branch and PR
- Teamwork document is complete for the most part from Peter's perspective
    - missing Roman's belbin roles (middle 3 and lower 3)
    - Everyone needs to send peter preferred developer role
    - Peter will create branch and PR into develop – will review and merge
- TODO:
    - Sprint planning documentation
    - Github issue creation
    - backlog tasks of issues on Github (At least for sprint 2)
    - Storymap

New action items and responsibilities:

1. Review and merge Leen's Project Overview PR into dev
2. Review and merge Roman Lofi user interface PR into dev
3. Complete teamwork document and create PR / merge into dev
4. Schedule team meeting to complete sprint planning documentation and other related sprint 1 components

Reference documents

- **Client meeting 2 Questions - Google Docs**