CollaBoard Monetization Strategy

# CollaBoard Product Offerings

|  |  |  |  |
| --- | --- | --- | --- |
| Free | Personal | Student & Non profit organizations | Enterprise |
| -Max 3 Projects | -Unlimited Projects  DropDown where he can choose the number of subscriptions the user needs. Like when he buys for his team.  -14 Day free trial | -All features of CollaBoard personal  -Discount for students  -Discount for non-profit-organizations | -Surface Hub Version  -Large interactive display version  -Enterprise  Features  --User Management  --On Prem Installation  --Custom azure tenant |
| 0 $ | 7$ | 3$ | PRICES NOT DISPLAYED IN THE POP UP:  8$ / user  On-prem: starting from 10’000  Custom azure tenant: starting from 10’000 |

|  |
| --- |
| CollaBoard Large touchscreen version (Surface Hub) |
| 30-Day trial |
| 30$ / Month |

# Application startup (no license available)

At startup the application will check the kind of license applied to current user or machine, in case **no license** is available the following scenario is expected:

* App is launched on a Surface Hub or a device with **diagonal size of 55 or more inches**
  + In this case an Enterprise license must be acquired so a dedicated popup will open (see 2) then after user has provided the required info a 14 days trial will start with a greeting message (see 3), after trial expired the app won’t run anymore and a popup (see 4) will invite user to buy the license.
  + If registration is completed the application will land at startup screen with Team edition mode button visible.
  + First time it lands to startup screen the Terms of Services and newsletter popup (see 1) will appear, in case TOS are not accepted the application won’t proceed, this dialog will appear just once for logged user.
* App is launched on a device with **diagonal size less than 55 inches**
  + User logs in using actual providers (MS, Google, etc…)
  + App will land at startup screen where an “Upgrade” button and a text indicating remaining free projects will be visible.
  + First time it lands to startup screen the Terms of Services and newsletter popup (see 1) will appear, in case TOS are not accepted the application won’t proceed, this dialog will appear just once for logged user.
  + No limitations or nag screen will be applied to these 3 free projects
  + The user will always be able to join any project is invited to.
  + When user tries to create a brand-new project and no more free projects are available a popup (see X) will appear with the invitation to upgrade.

# Upgrade scenario

Applies when a user decides to upgrade his license, no matter if he has consumed free projects or not.  
A popup dialog (see 5) will open containing 3 options

* **Free**
  + Will just inform the user about features and it will contain text indicating that user is using that mode and the number of remaining free projects
* **Personal or Education**
  + Together with features info will contain an upgrade button and an additional text explaining how to be eligible for an education/NPO discount code, below the text a link to IBV page where the user can request a promo code
* **Enterprise**
  + Together with features a “Request a code” will open the browser to IBV site containing everything related to enterprise licensing.  
    This column will also report the machine Id and will invite the user to make a note of it since it will be required during enterprise license acquisition.

## Payment dialog

The payment dialog (see 6) will collect all the information required for billing, that would include

* User login email
* Payment type (yearly or recurrent monthly)
* Promo code
* Credit card info (code, name, expiration date, security code)

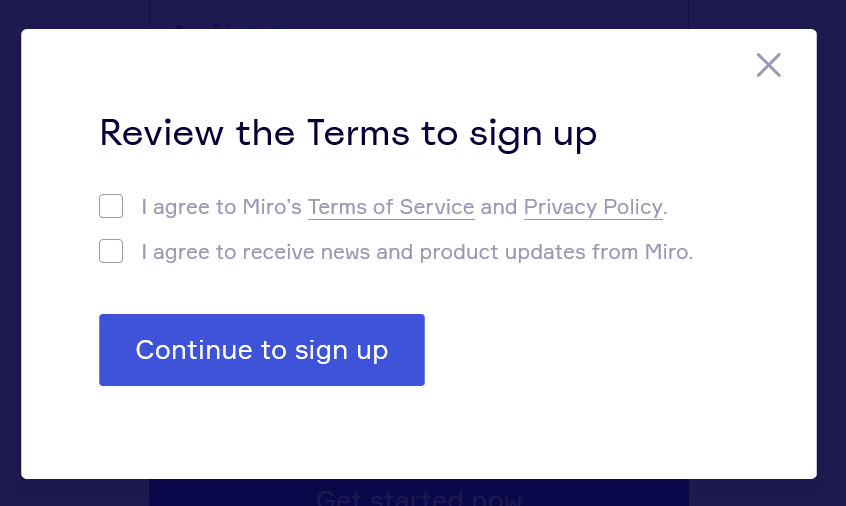
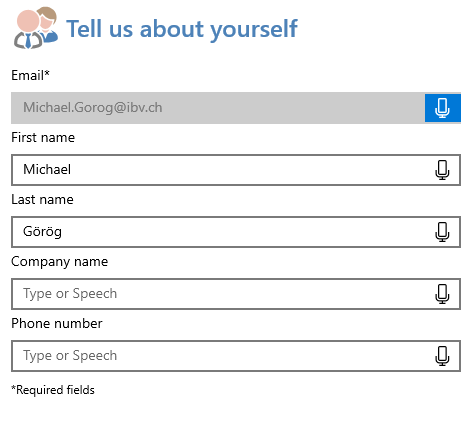
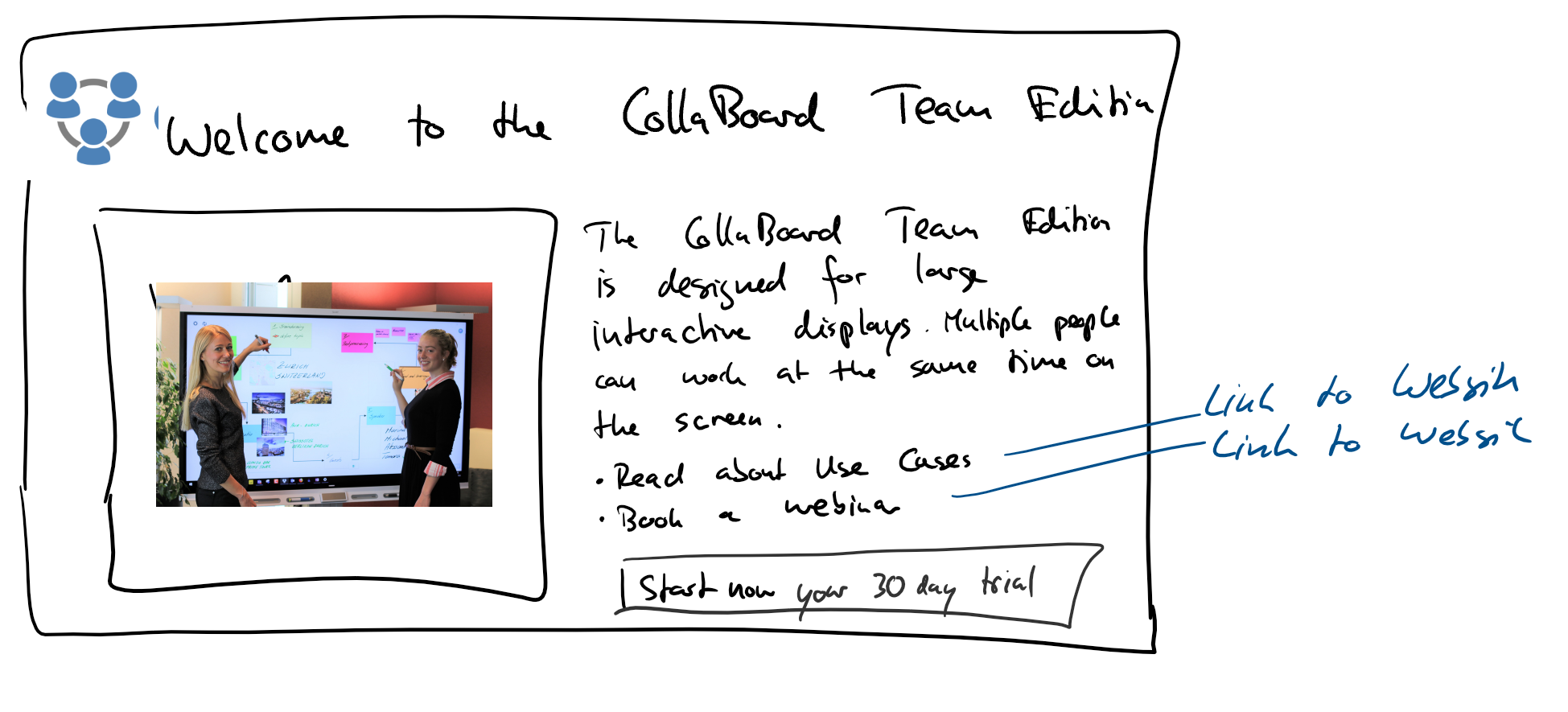
If payment fails the dialog will display an error message, otherwise a thank you message will be shown and the user, when the user will close payment screen, the payment dialog will also close displaying the startup screen.  
Being application licensed the upgrade button won’t appear anymore until the license expires.

Inside application info tile the expiration date will be always be visible, current features like License key, register or evaluation are no longer necessary.

# Application startup (license available)

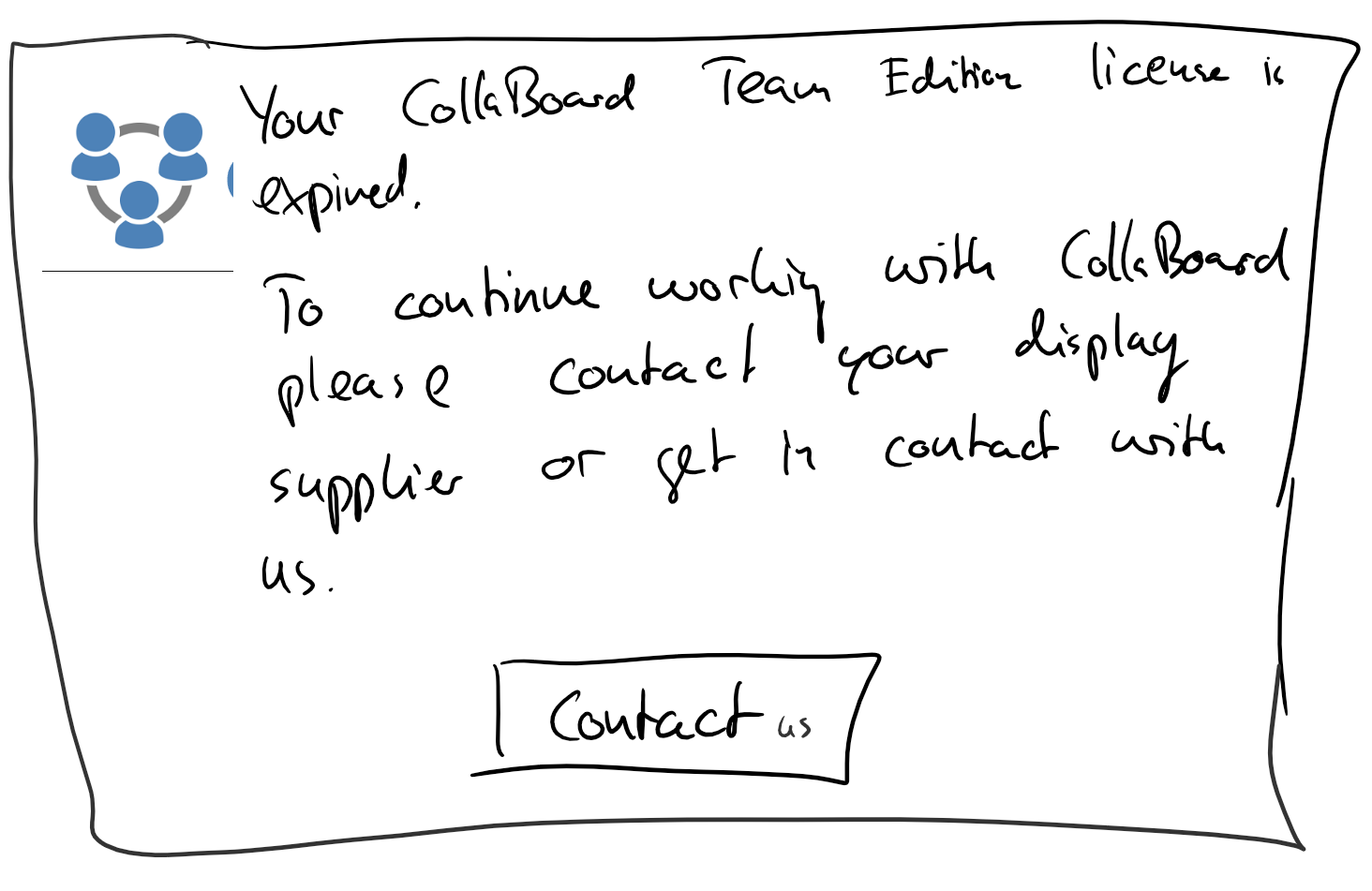
If a license is available Collaboard will work normally with the exception that Upgrade button won’t be visible.

# Screen description

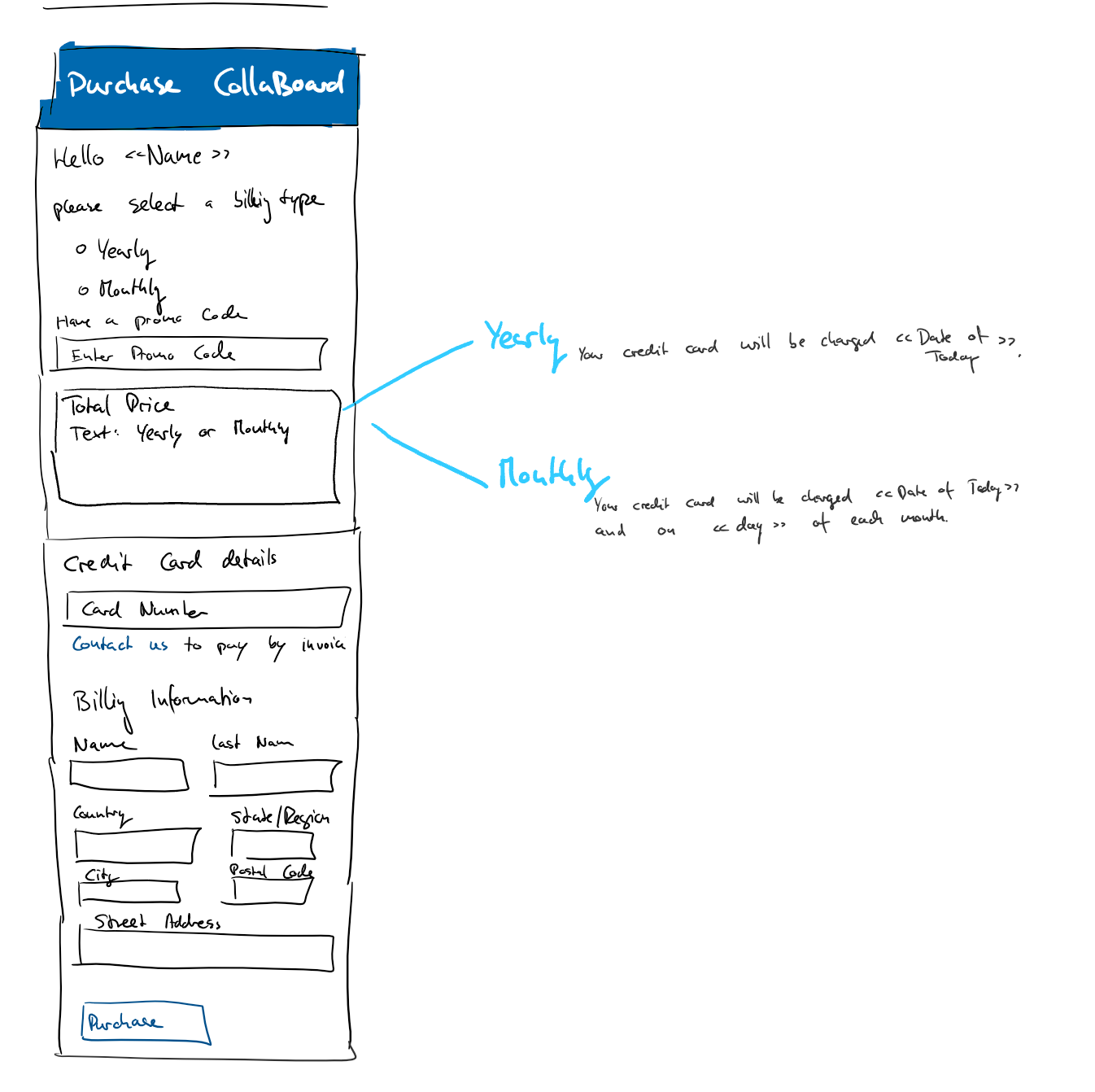
1. Terms review  
   
2. Enterprise Team registration  
   
3. Enterprise Team registration greeting  
   

Use Cases: <https://ibvsolutions.com/en/business-use-cases-for-interactive-displays/>

Webinar: <https://ibvsolutions.com/en/ibv-webinars/>

1. Enterprise Team license expired  
   

Link: <https://ibvsolutions.com/en/contact-us/>

1. Payment dialog  
   

# Payment engine

We decided to go with stripe payment method, an analysis regarding the available developer API and any limitation will be done during the implementation sprint.

<https://stripe.com/ch>

# Technical implementation

In order to implement the above indicated requirements some work must be performed server side on CollaBoard Apis.

At startup the application will invoke a new webapi **GetLicenseStatus** passing following information

*GetLicenseStatus (machineId, username, bool displayDiagonal)*

* + machineId -> Is the unique machine Id
  + username -> optional username (might be null since user isn’t logged yet)
  + displayDiagonal -> The display diagonal  
      
    The API will return an object containing following properties:
  + *Status* (enum)
  + *IsExpired (bool)* indicates that license is expired
  + *ExpirationDate* (nullable DateTime containing the expiration date and time when a license is available)  
      
    *Status* is an enum with following values:   
      
    0: Error  
    1: Free  
    2: Personal  
    3: Enterprise  
    4: Enterprise Team

When a diagonal of 55” or greater is passed just the machine id will be used to identify a license, in other cases the username will be considered.  
The *IsExpired* property allows to move the all the license logic server side and prevent any kind of date adjustment client side.

Another webapi named ***GetFreeProjects****(username)* will return the available number of free projects for provided user (example:ibvuser@outlook.com) please note that count must include also those projects created from a template.

To upgrade a user a new webapi named **UpgradeLicense***(username,licenseType, cardNumber, expirationDate,securityCode,promoCode)* will be implemented, the number of parameter (or object properties if more indicated) will change depending on the info requested on Payment dialog)  
  
*LicenseType* is an enum containing just one element: Personal.  
Even if not strictly necessary at the moment. it will allow us to quickly add other subscription levels in future.  
  
The webapi will take care of interacting with payment services and will return an object with following properties:

* *Status* (enum containing operation result -> Success, Refused, card Expired… TBD)

For the Enterprise Team registration maybe we can use the current **RegisterForEvaluation** Api, but we can reconsider it if necessary.

For recurrent monthly payments, server should handle them independently (if not already provided by payment api) and managed the case of expired card, missing funds etc by updating the license status and expiration date when something fails during renewal process.  
The new API should properly manage the currently active licenses making migration transparent to current users.

In order to manage Terms and Newsletter popup a new set of webapi will be implemented:

* ***GetAcceptedTerms*** *(machineId, username)* will return a bool indicating that Terms and Newsletter has already been accepted.  
  If username is provided returned value must refer to that specific user, otherwise the machine will be considered.

* ***SetAcceptedTerms****(machineId, username, newsletter)* will be invoked when user accepts terms and confirms. In case username is not null, selection must be considered at user level and not at machine level, so if another user launches CB on the same machine for the first time, the popup window will appear.

We have to save this information server side so that we can be consistent with Surface Hub where all local settings are deleted when session is closed.

## Client side refactoring

At the moment all the licensing related stuff is managed by **LicenseManager** class, I propose to refactor it removing all unnecessary stuff and adding what required for the new implementation.  
All stuff related to nag screen (timers, settings, UI stuff will be removed

All client new implementations will be implemented in a new *dev/feat/monetization* branch.

Other client side requirements will be added soon…