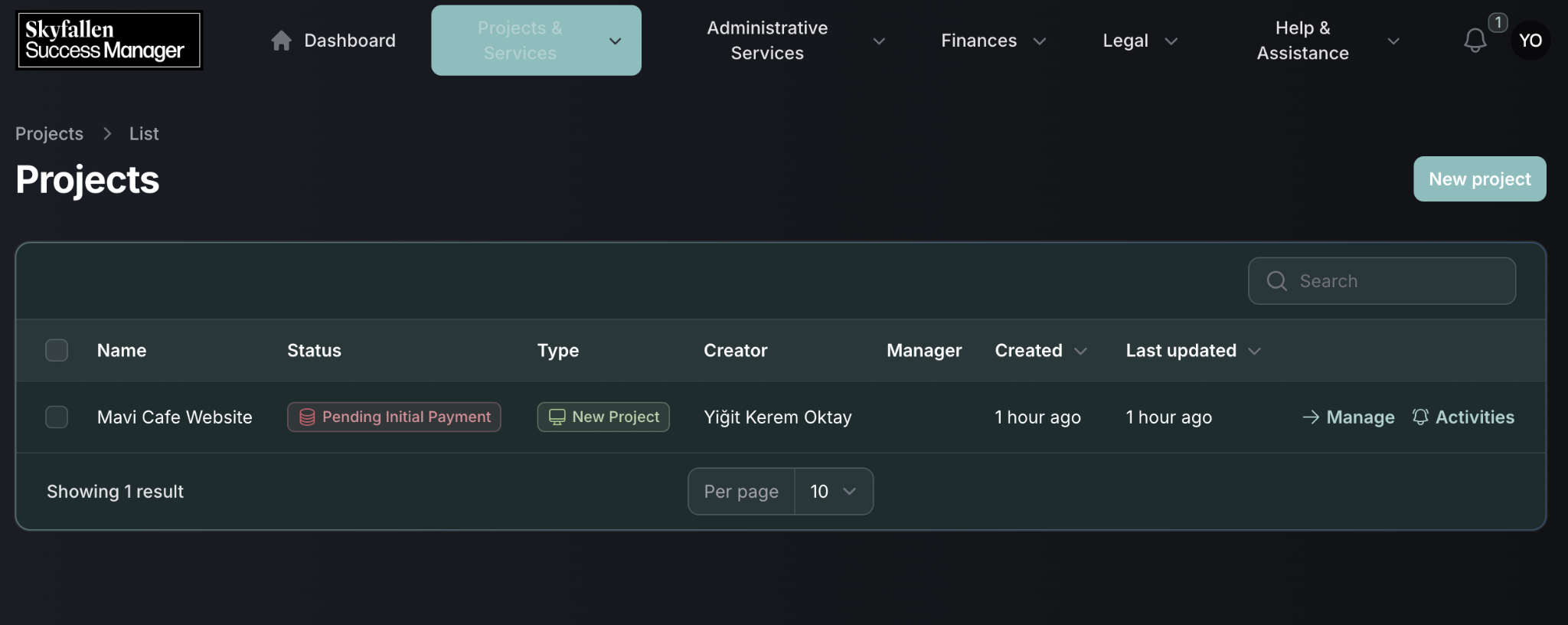
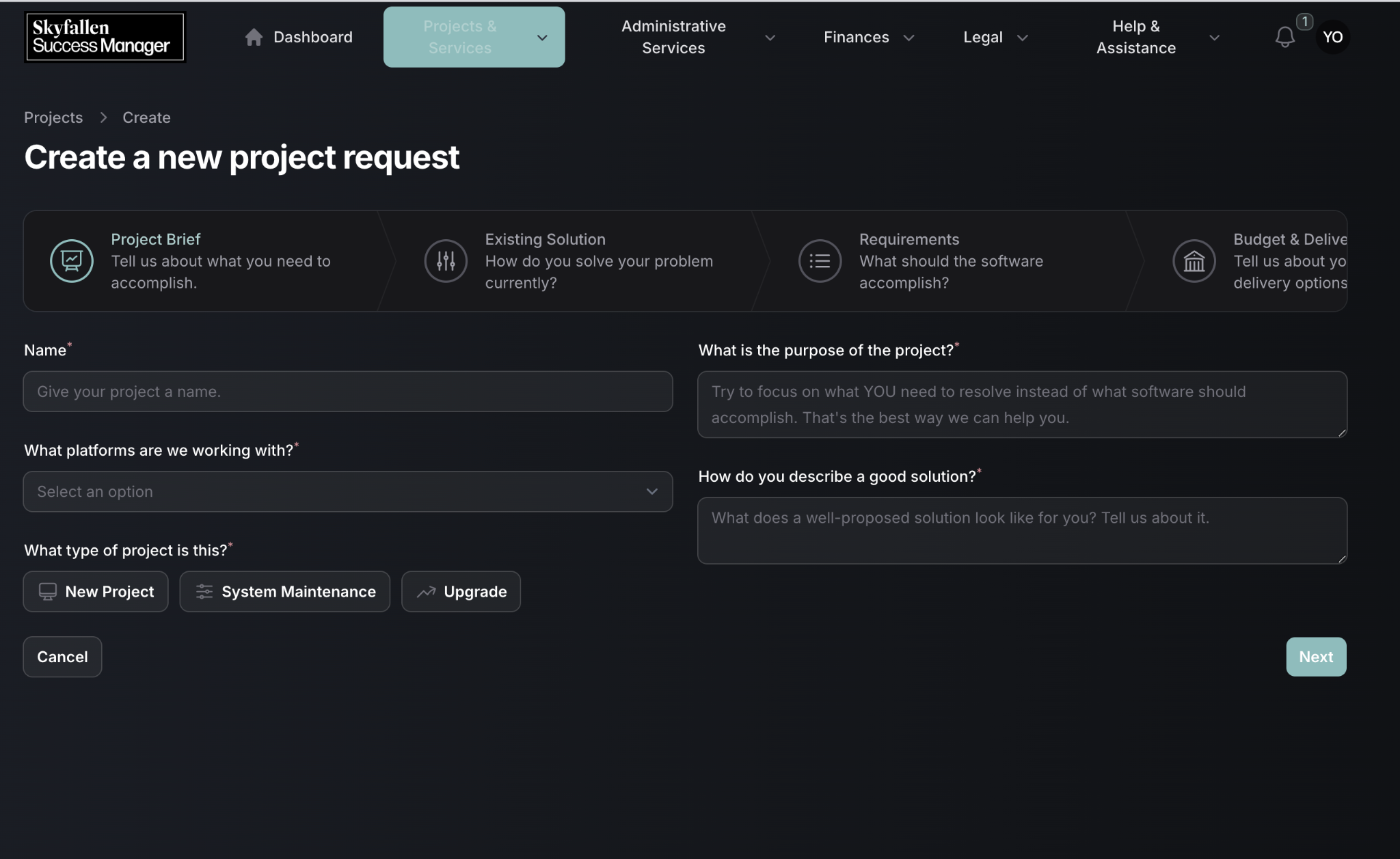
# Mockups

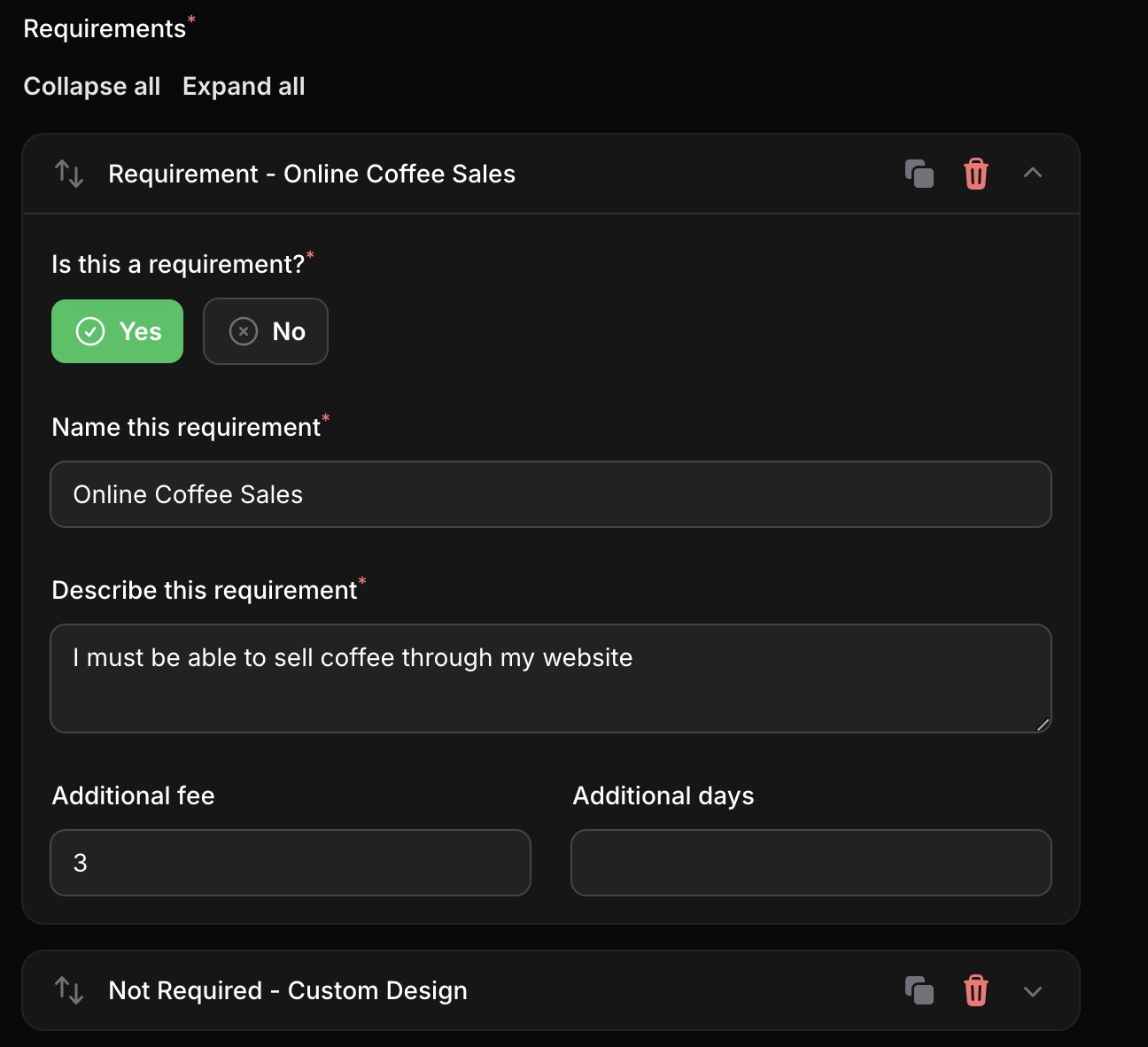
1. Projects List

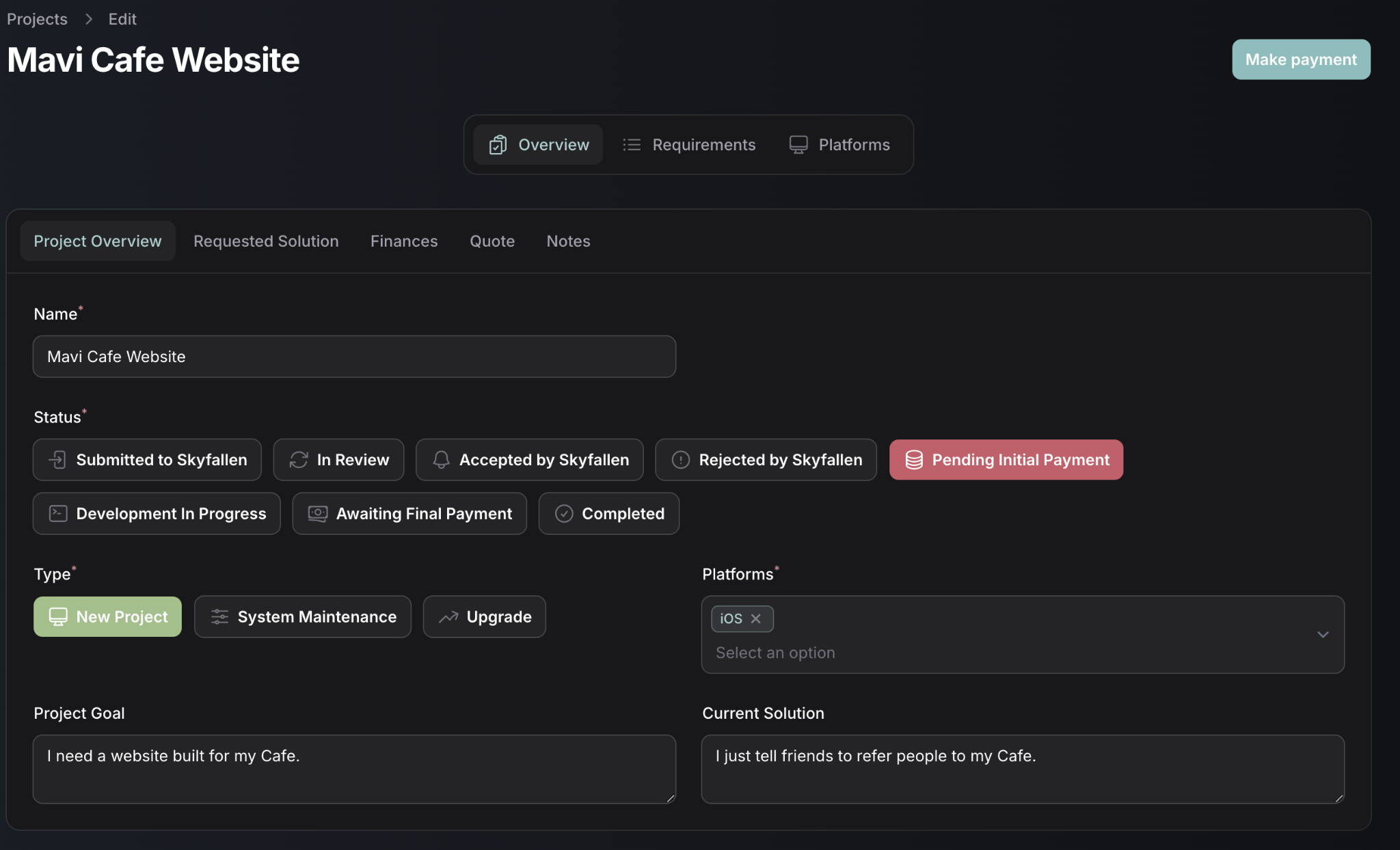


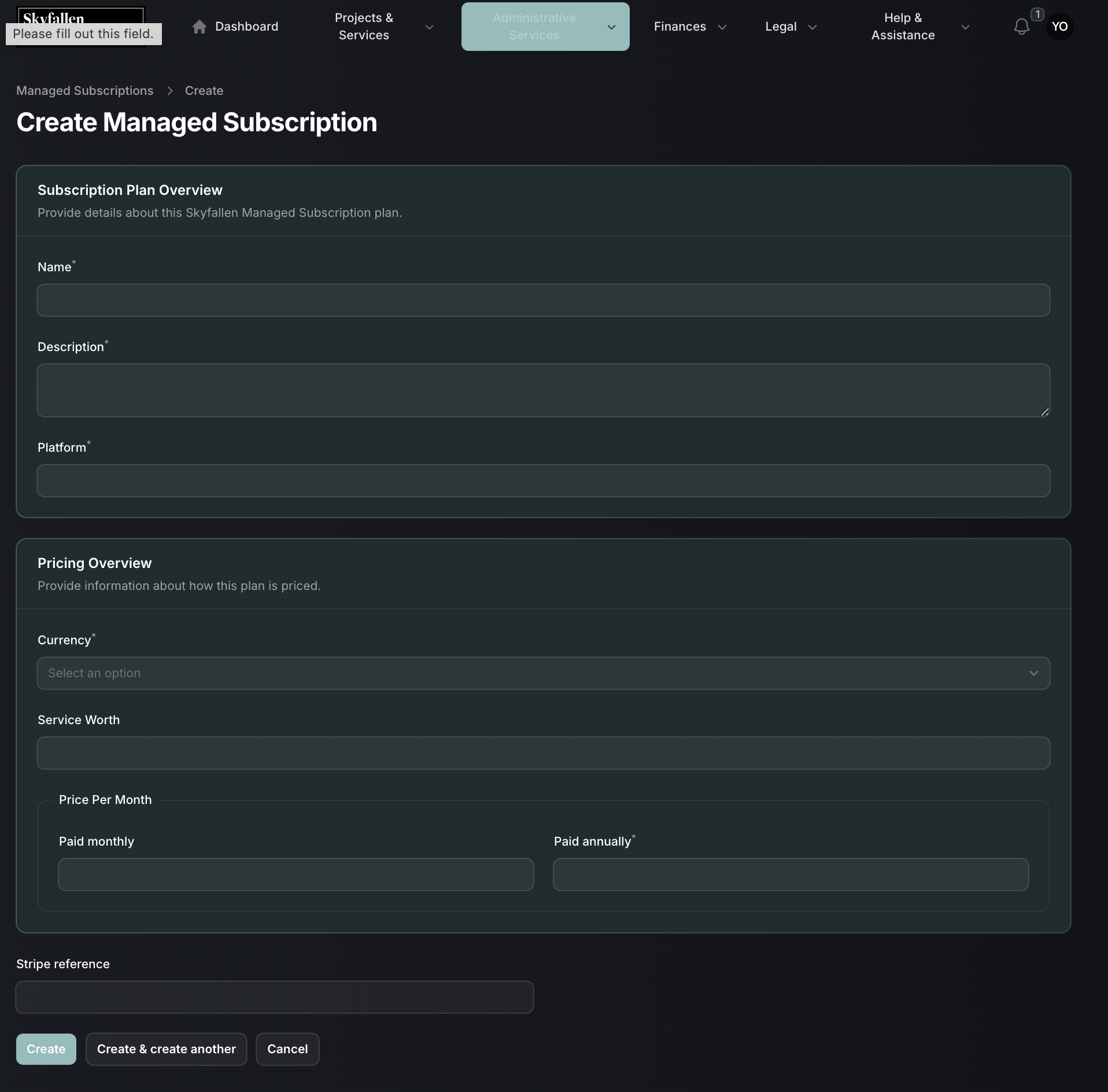
1. Project Request Wizard



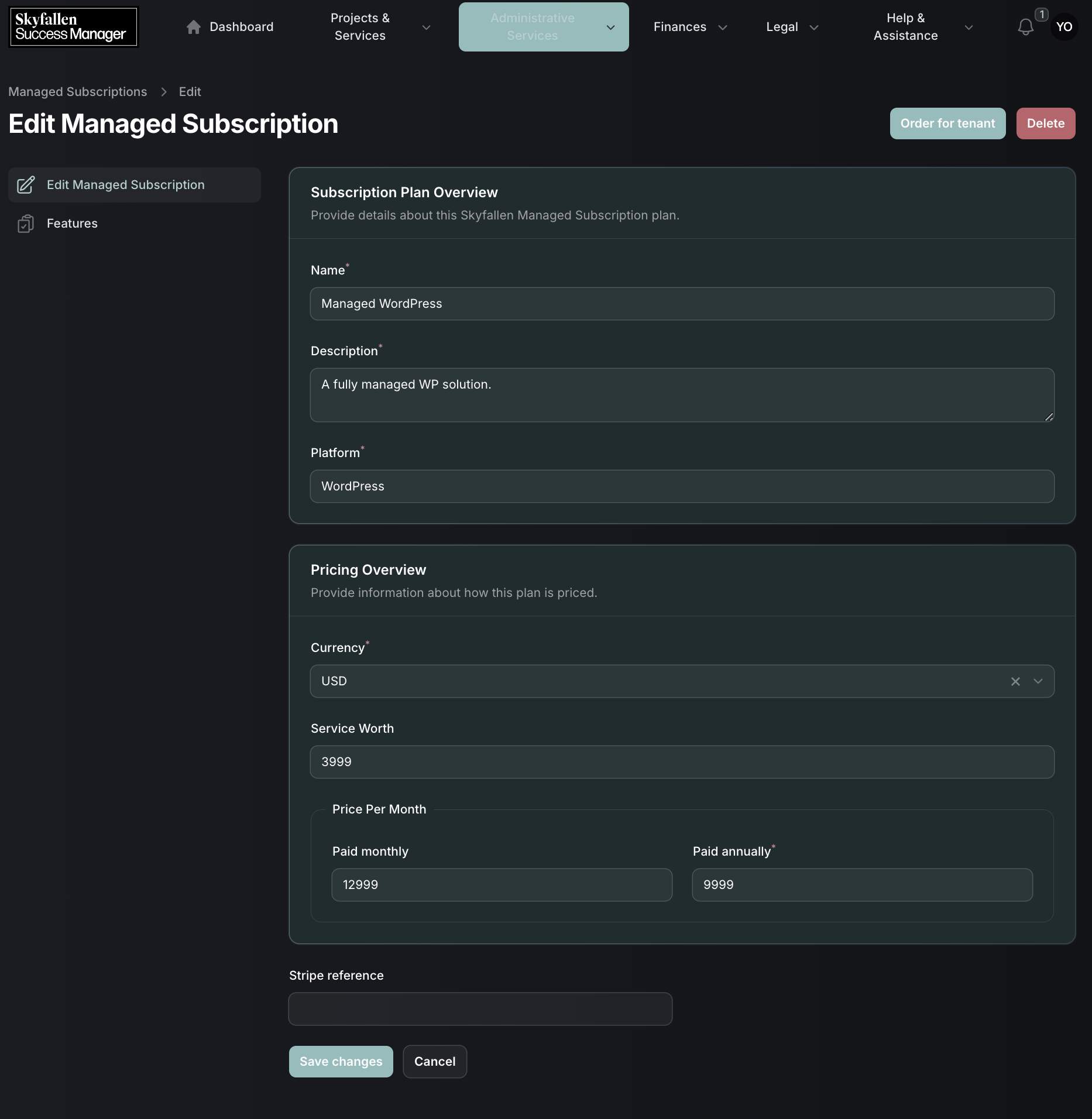
1. Project Requirement Wizard



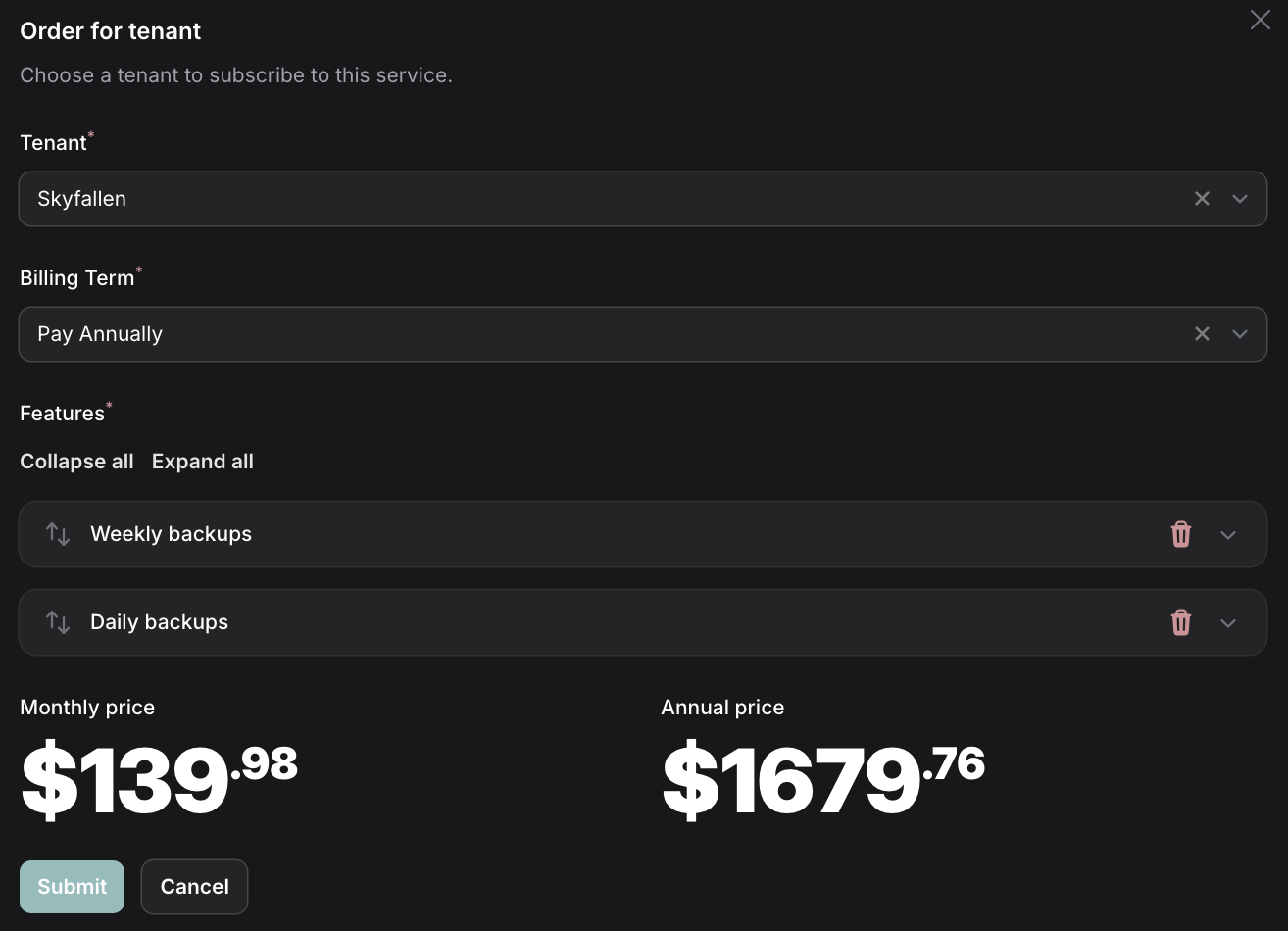
1. Edit Project Page
2. Create subscription template



1. Edit subscription template



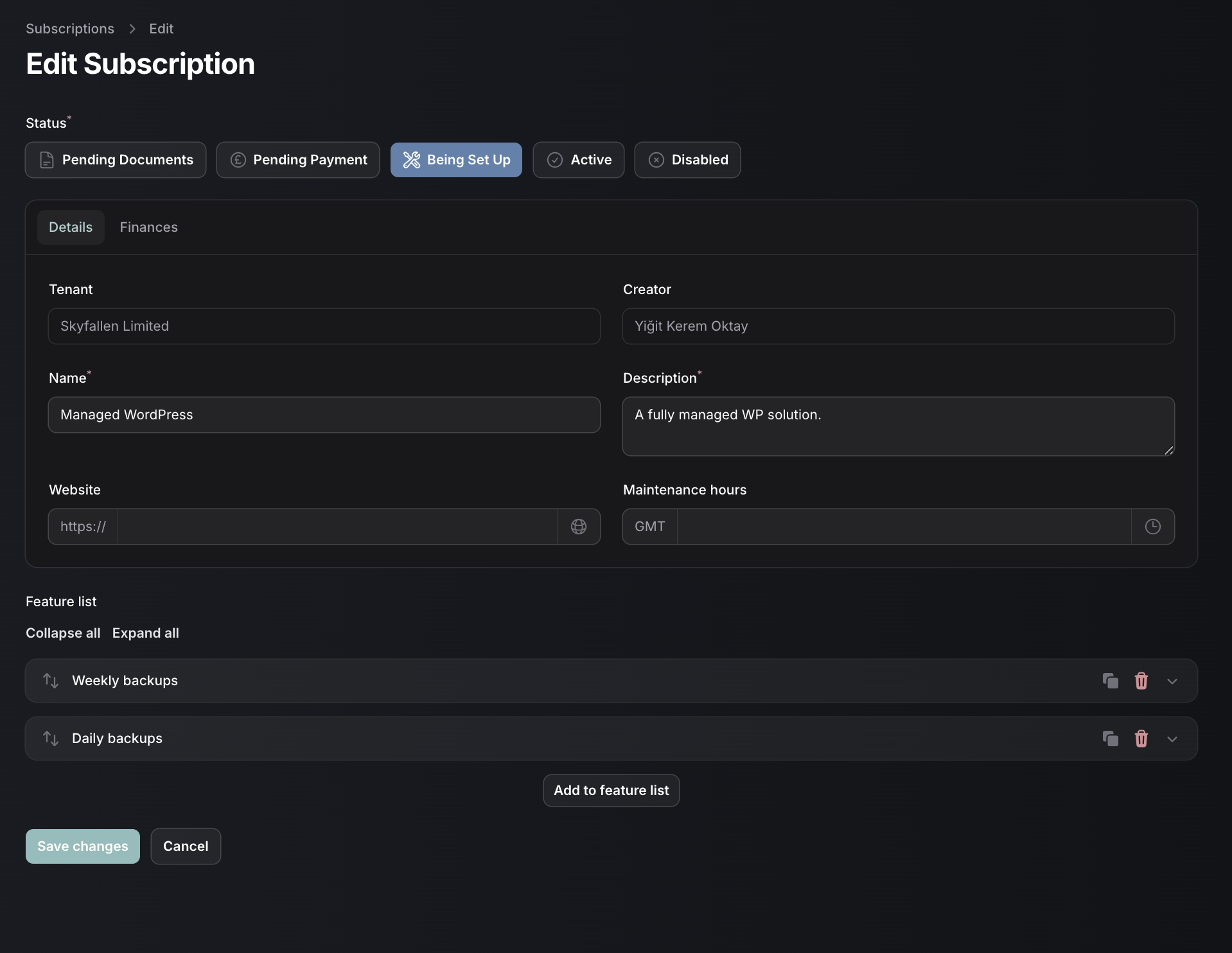
1. Subscription order modal

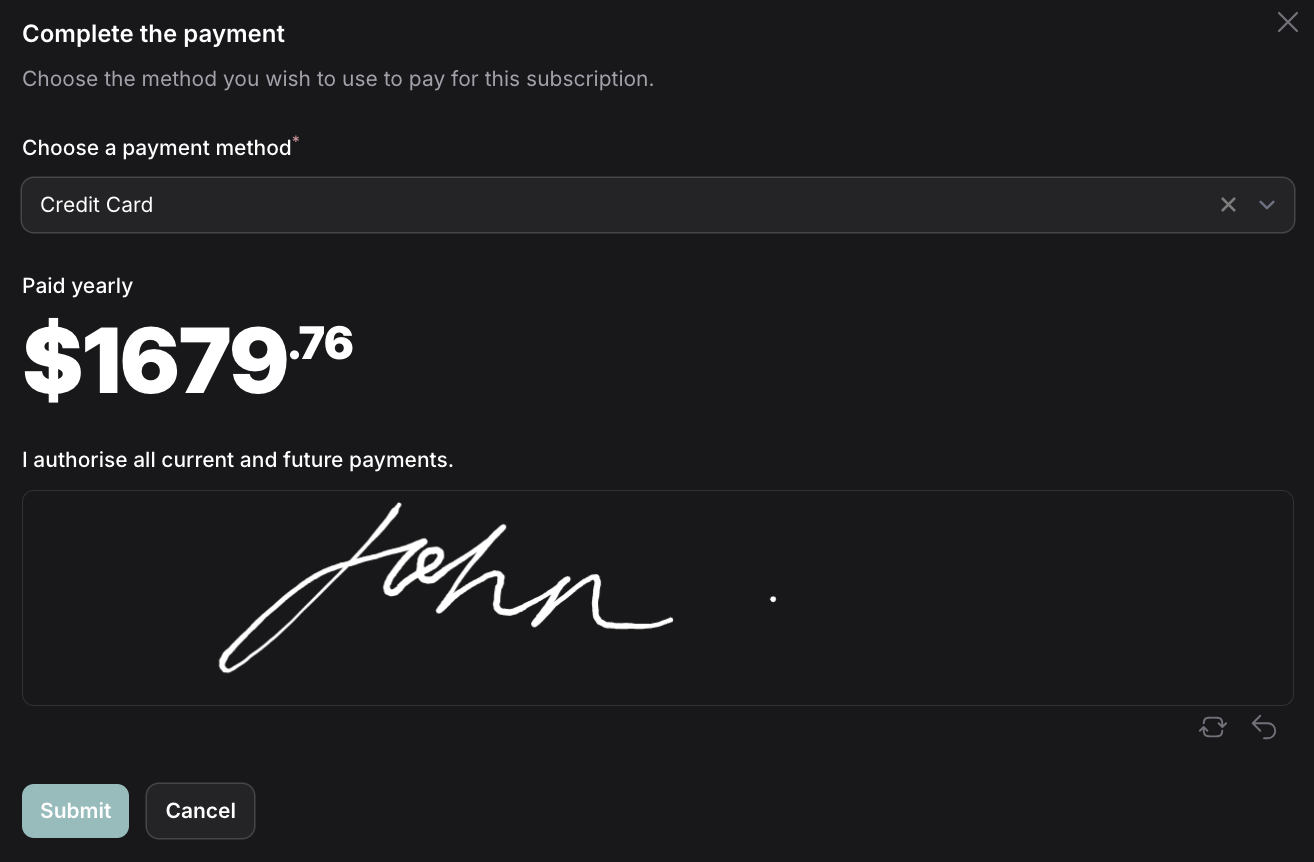


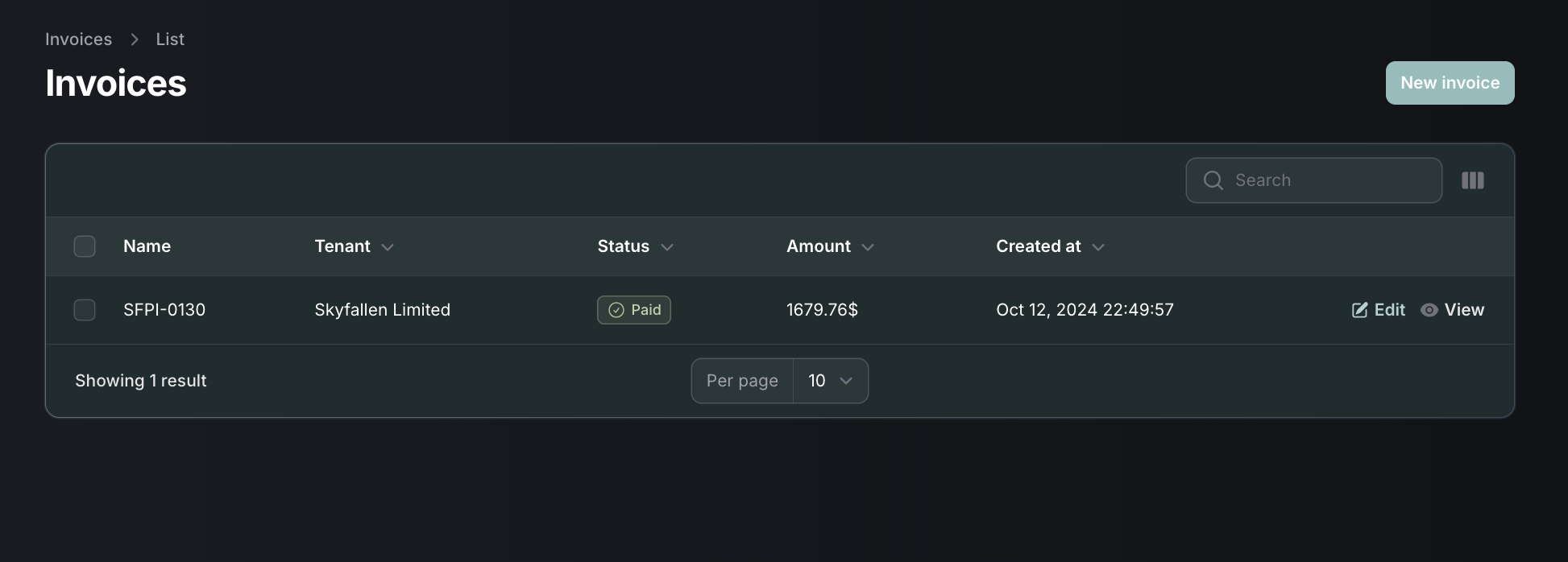
1. Subscription plan features

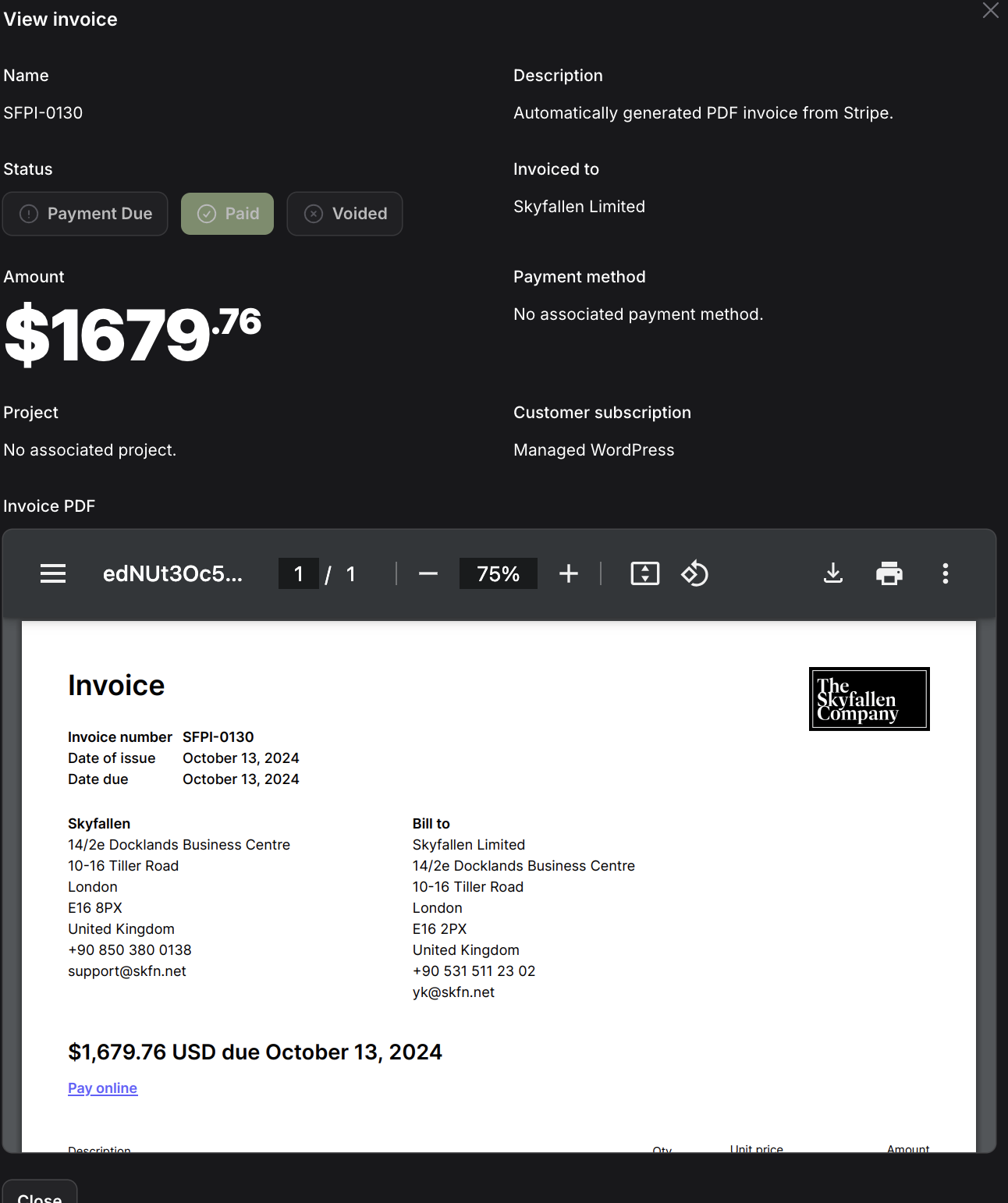


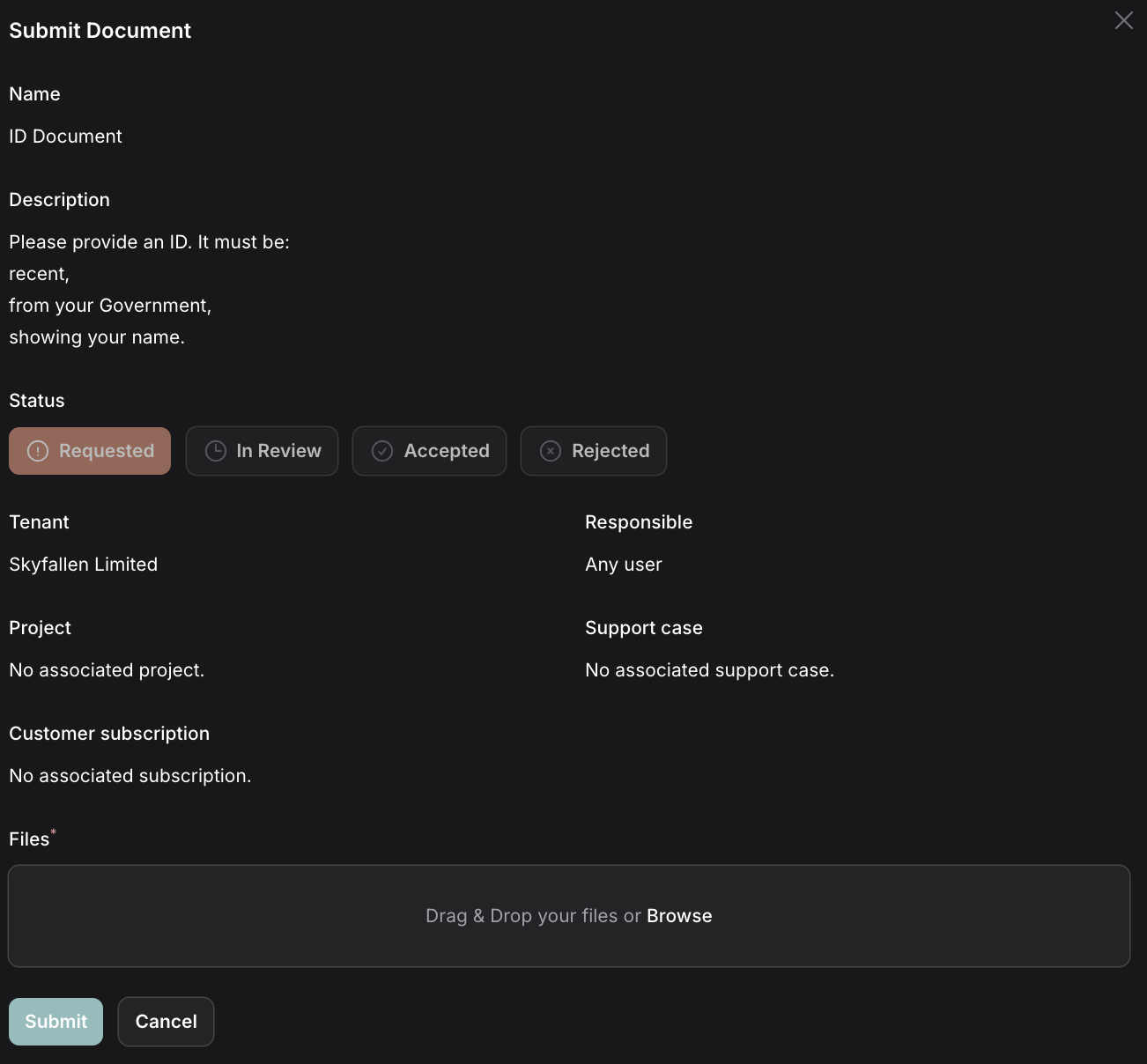
1. Customer subscriptions list
2. Customer subscription details



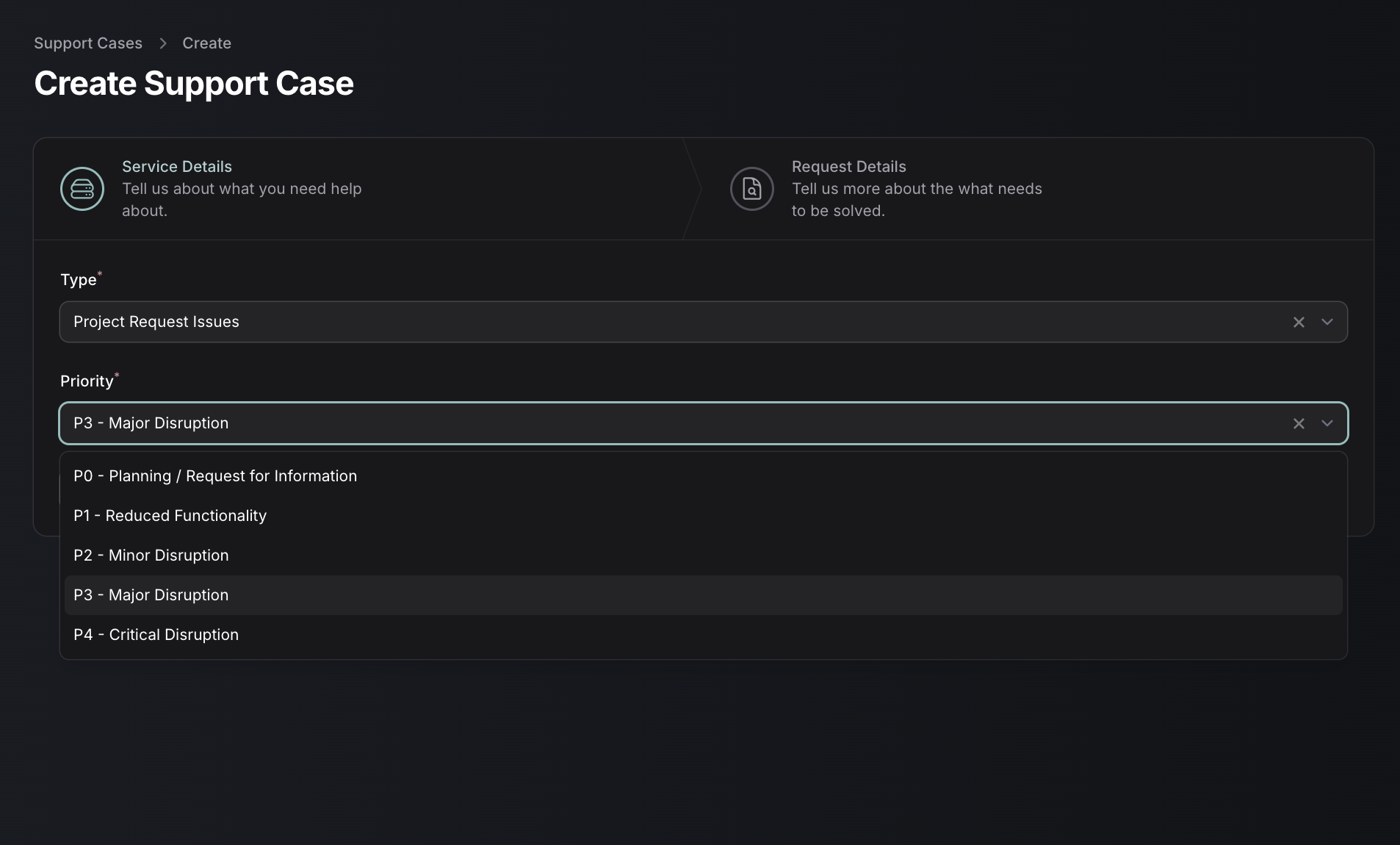
1. Customer subscription payment interface
2. Invoice list



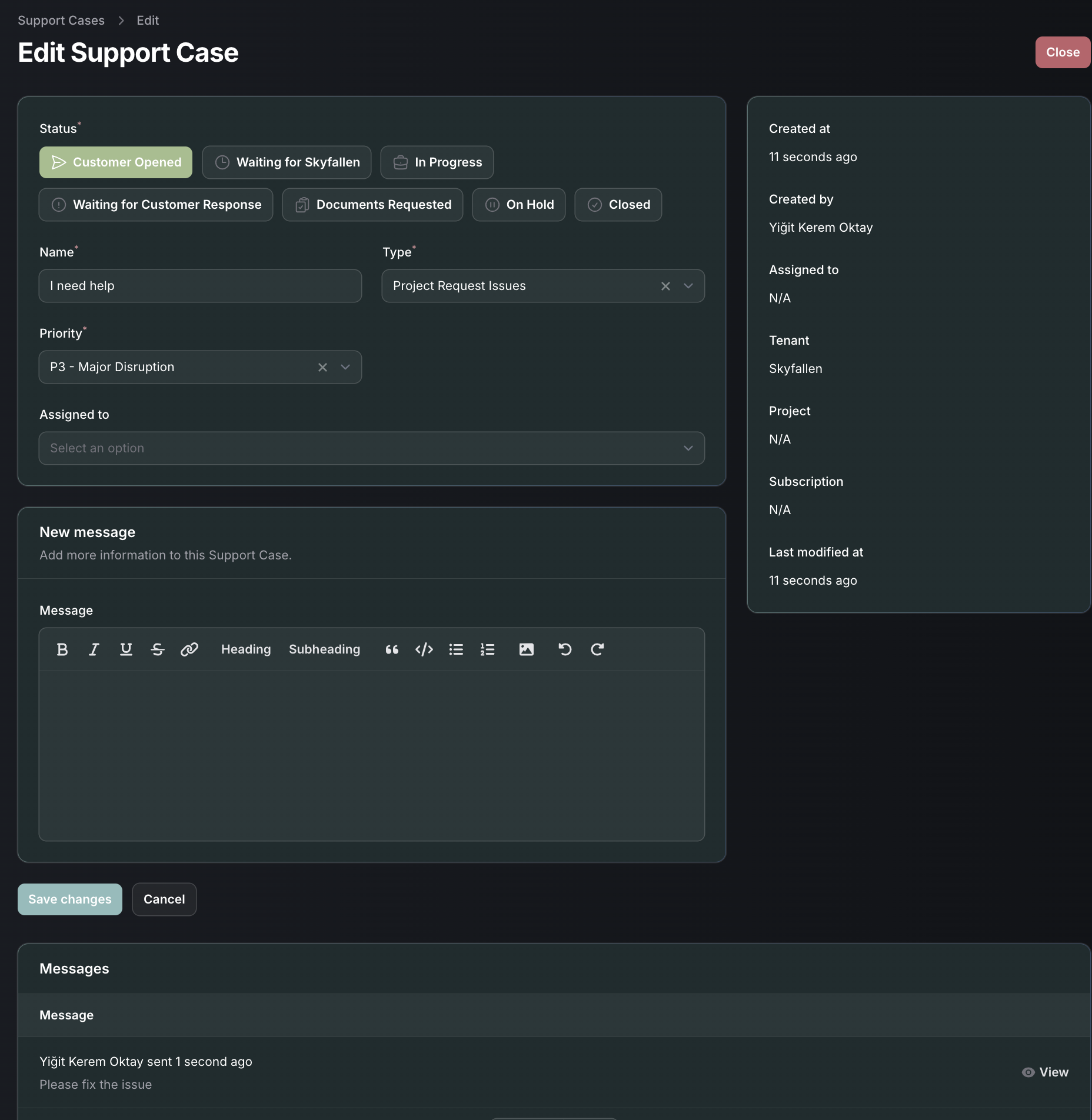
1. Invoice preview
2. Document submission form

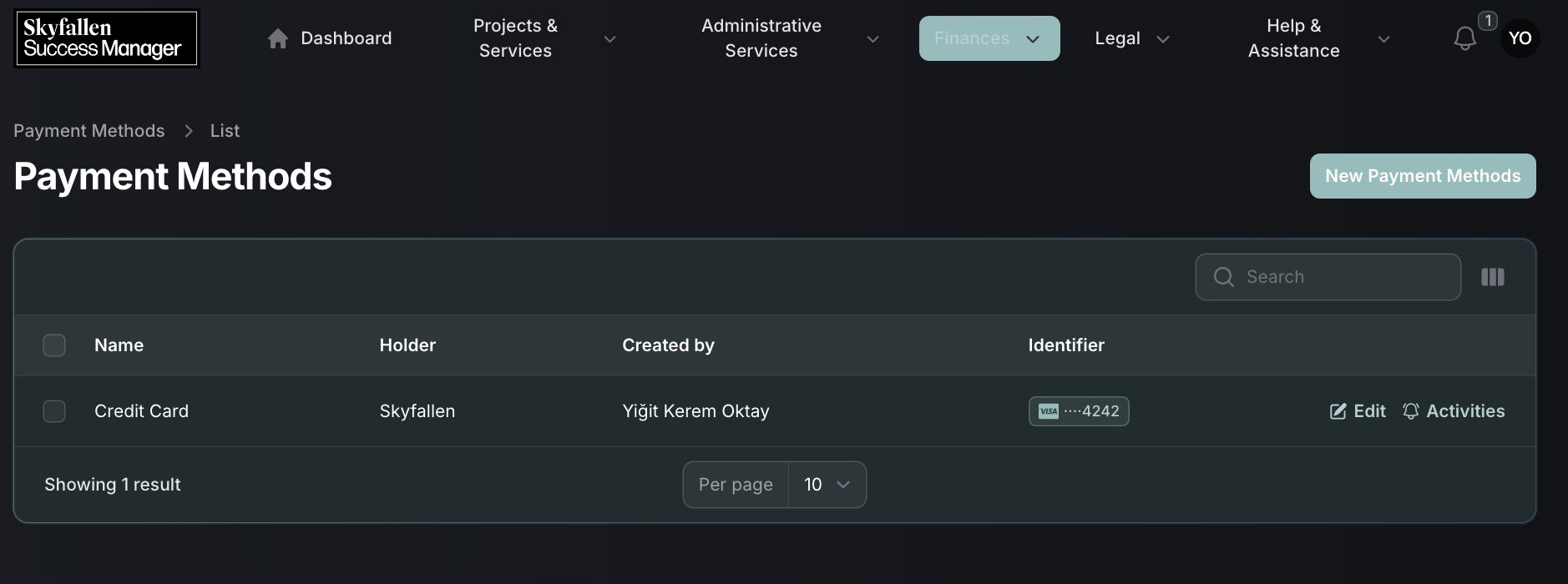
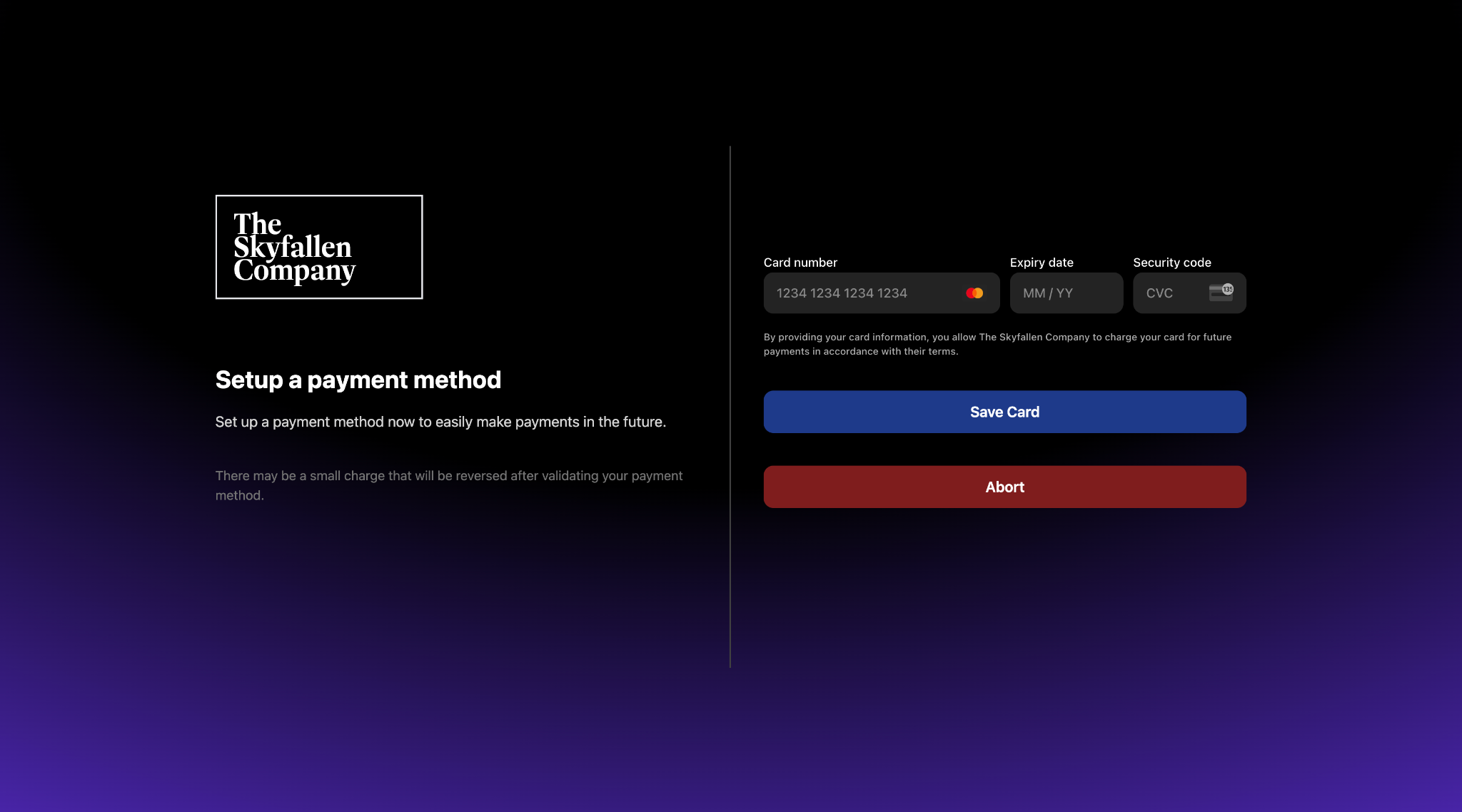


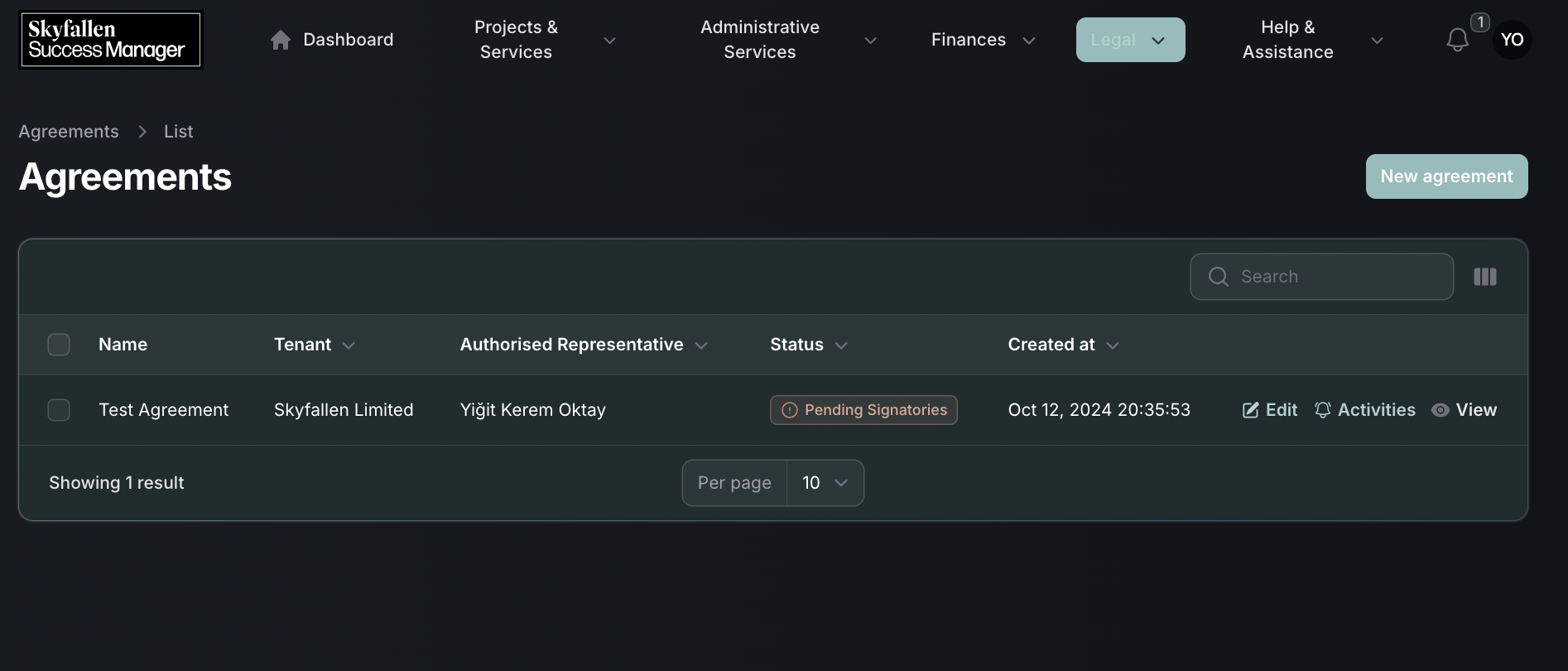
1. Support case creation form



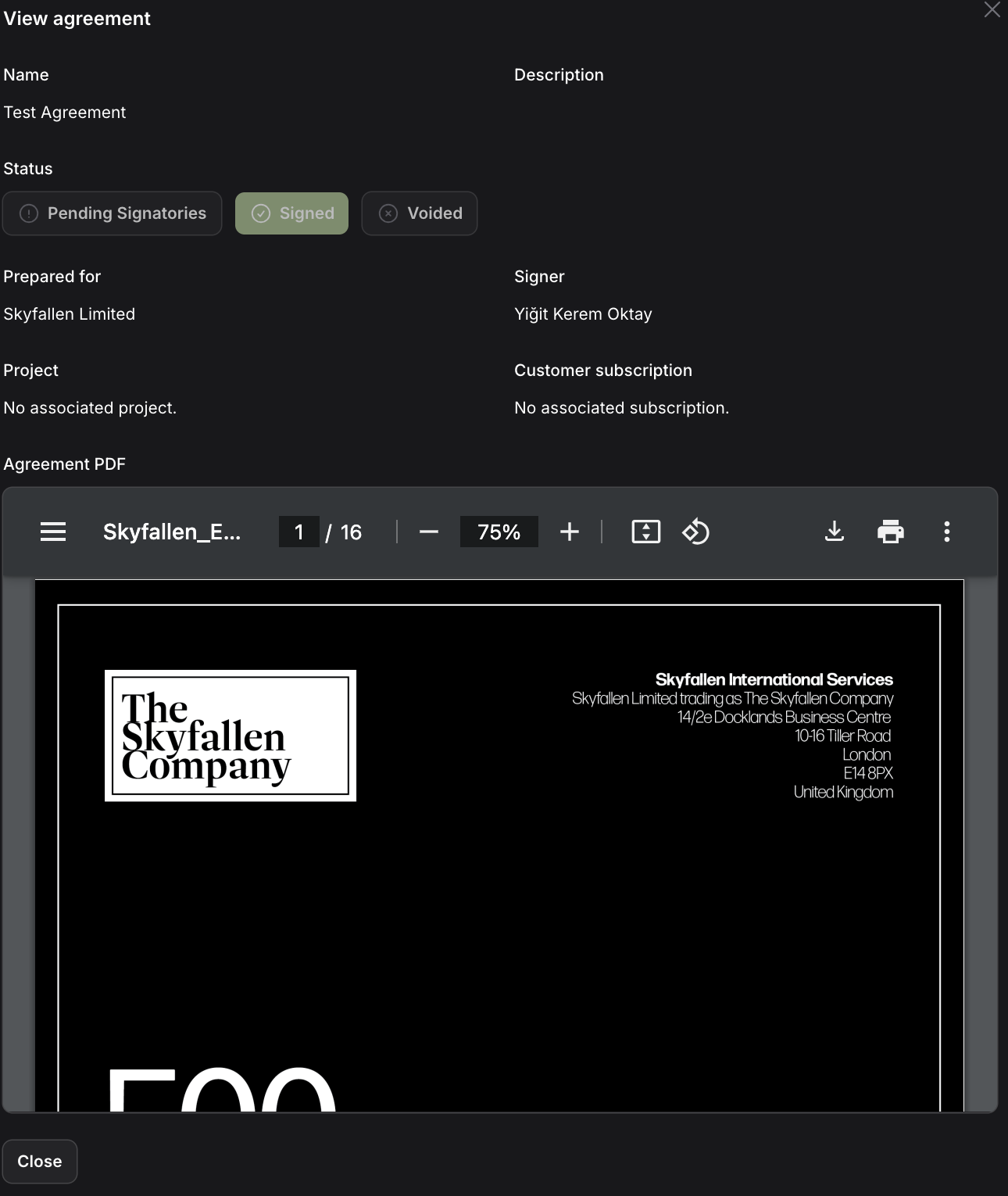
1. Support case overview



1. Payment method list
2. Payment method creation
3. Agreements list

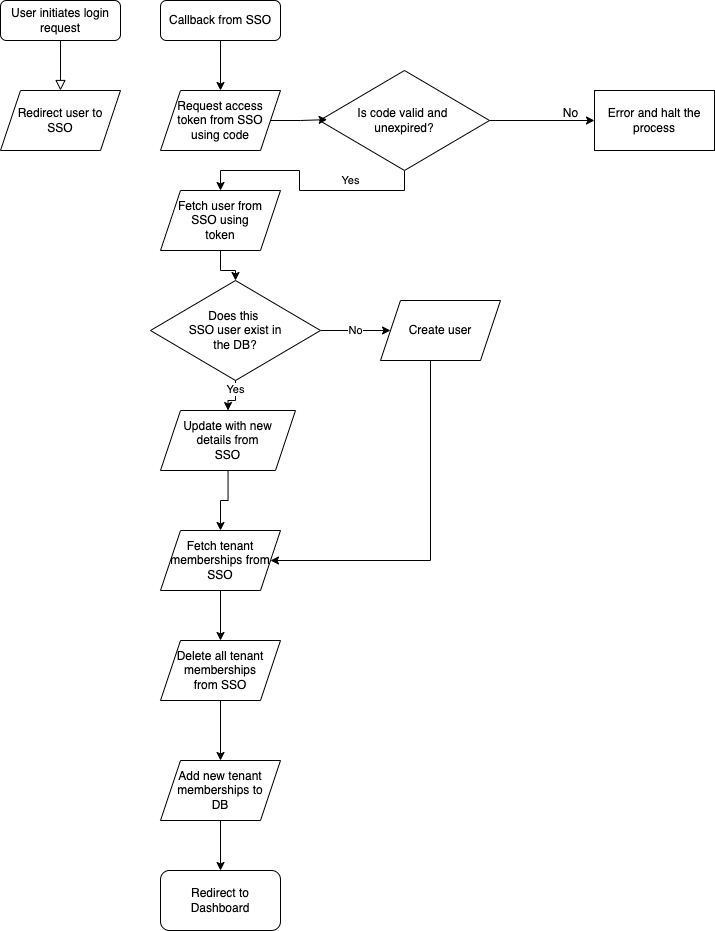


1. Agreement Preview

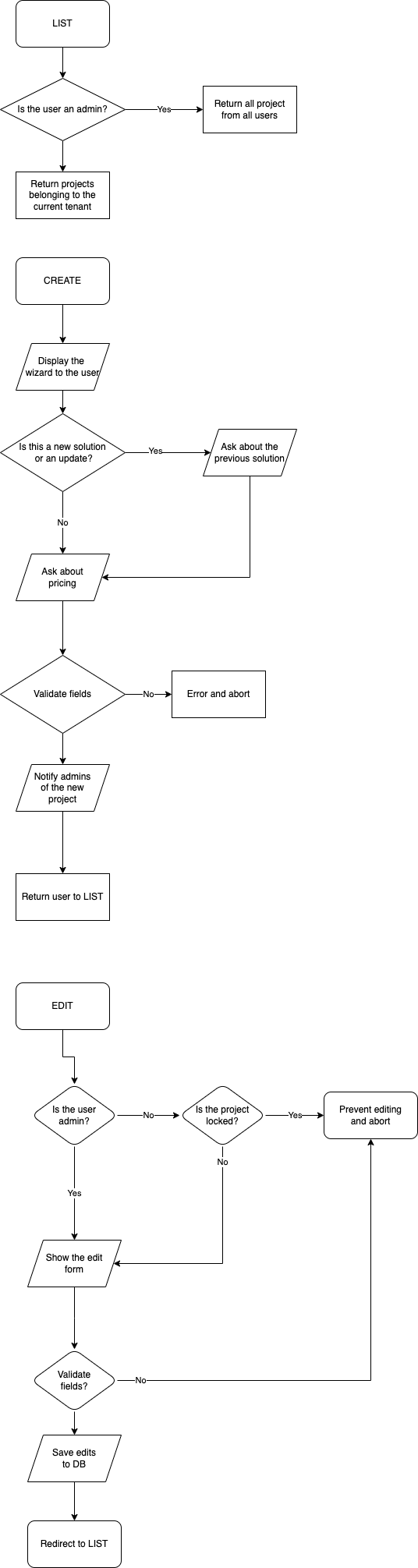


# Flowcharts

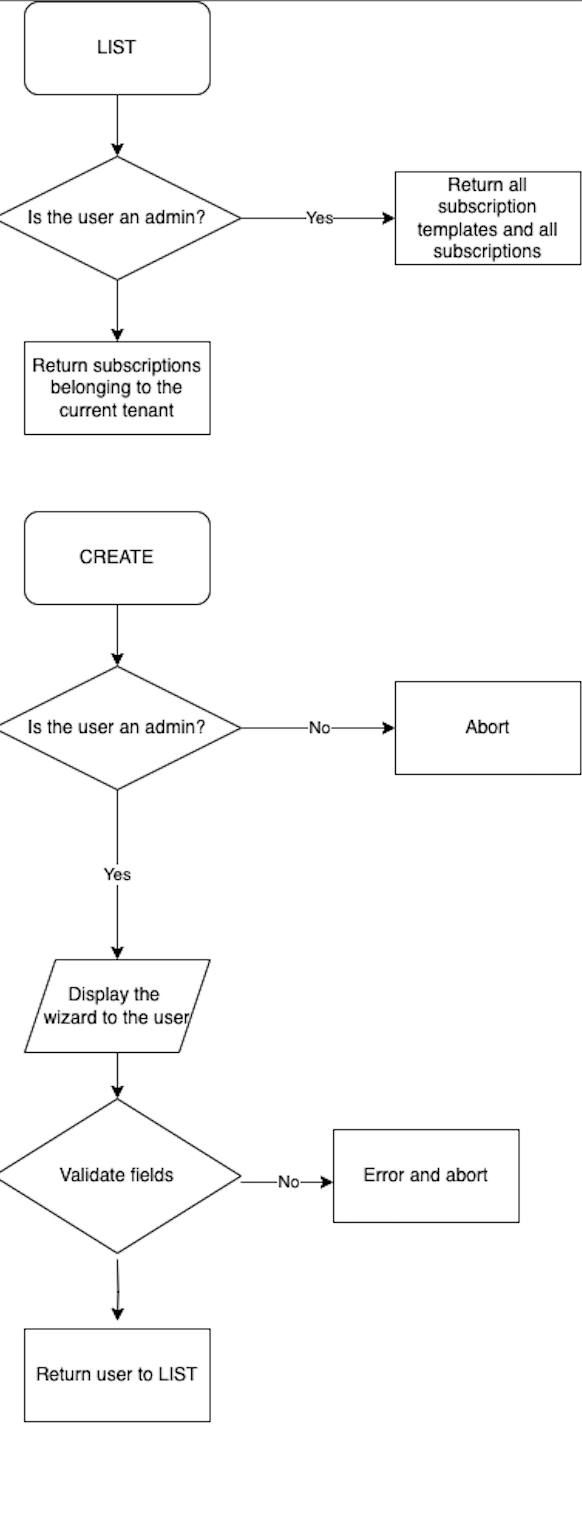
1. Auth Flowchart

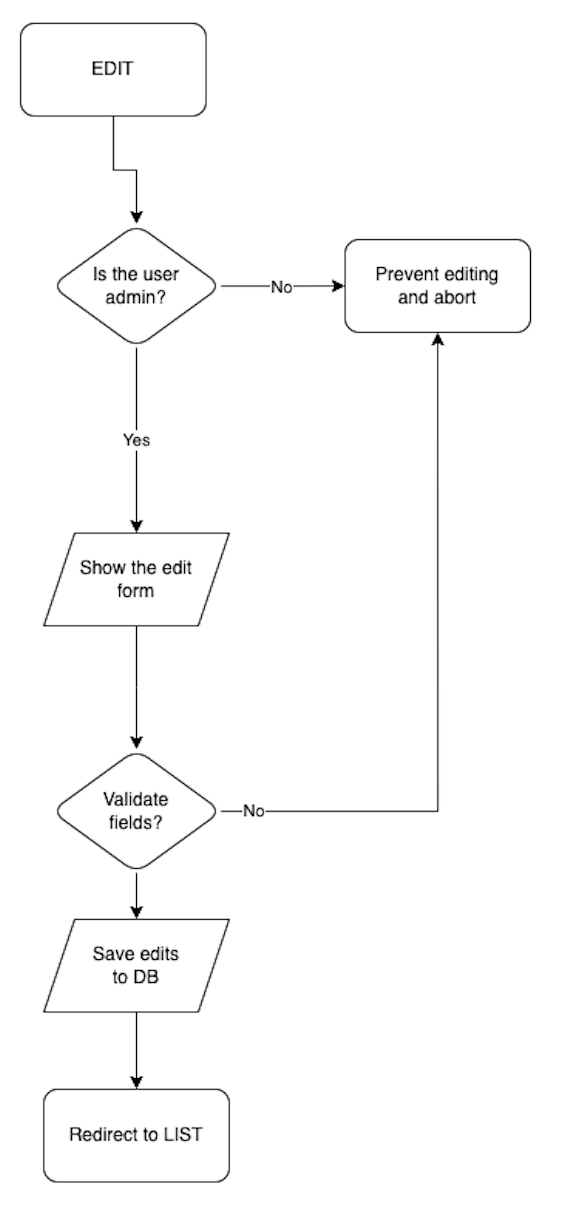


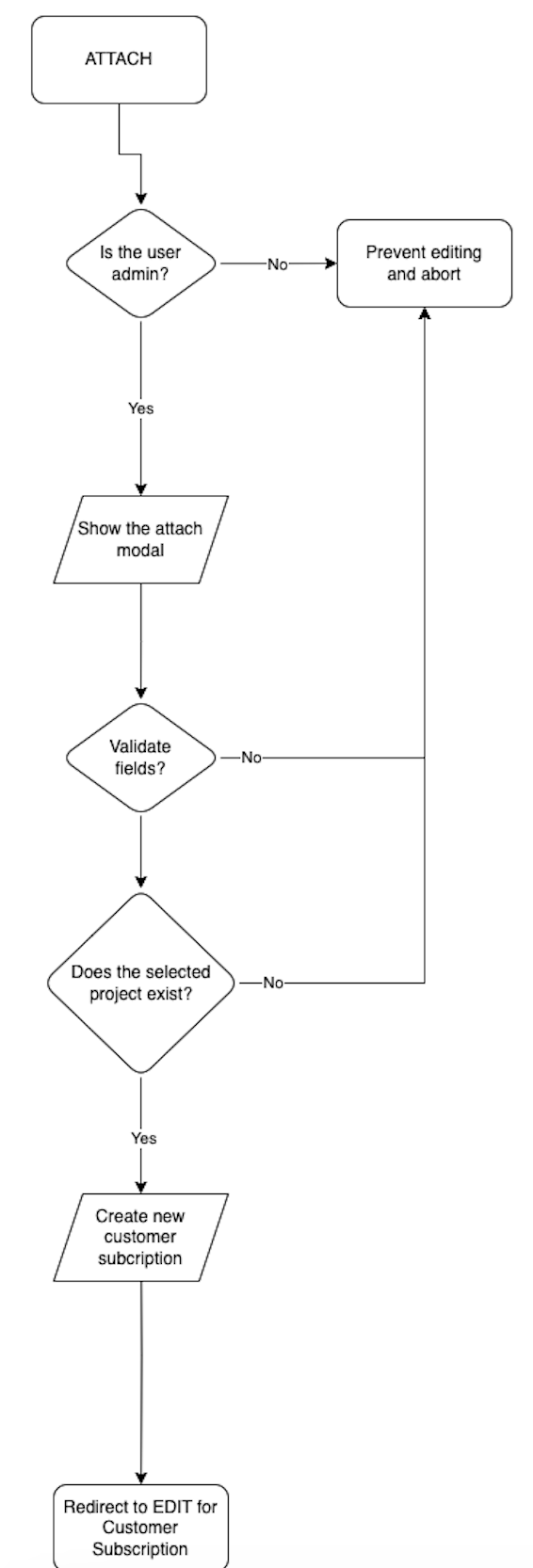
1. Project Flowchart



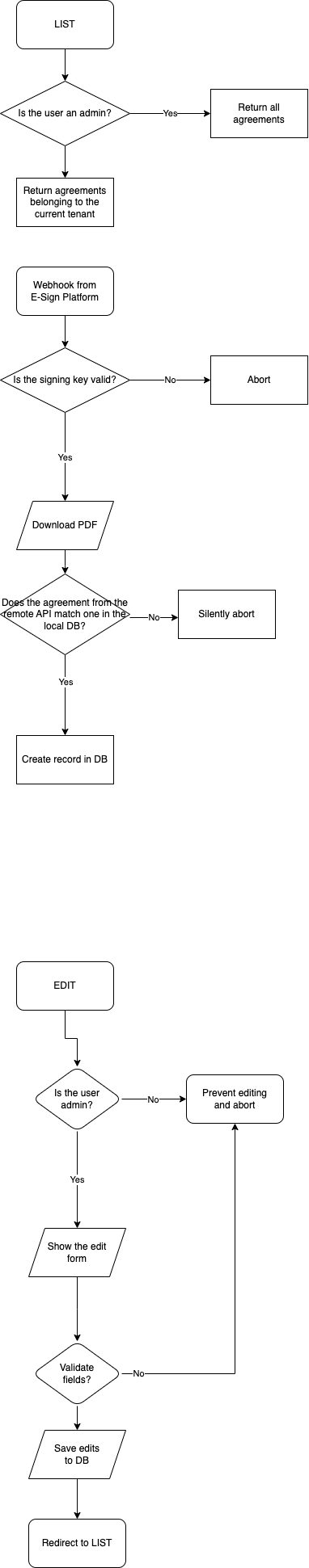
1. Subscriptions Flowchart



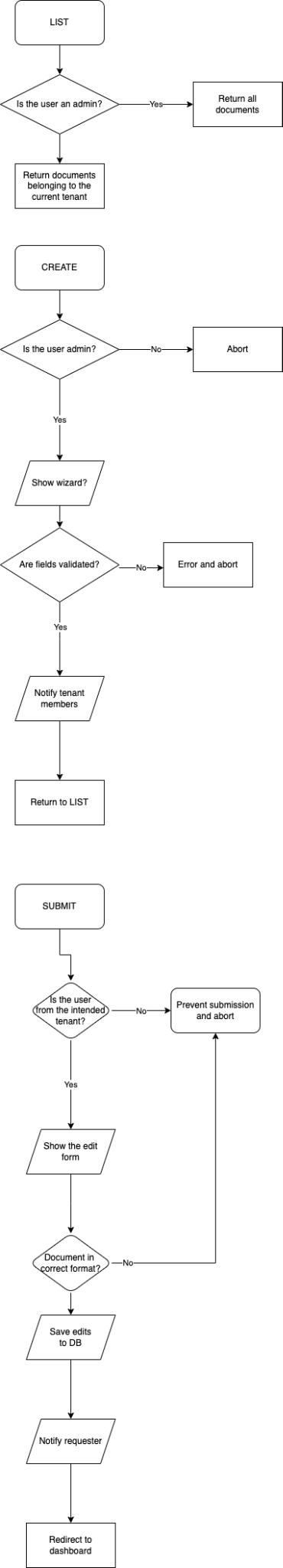




1. Agreements Flowchart



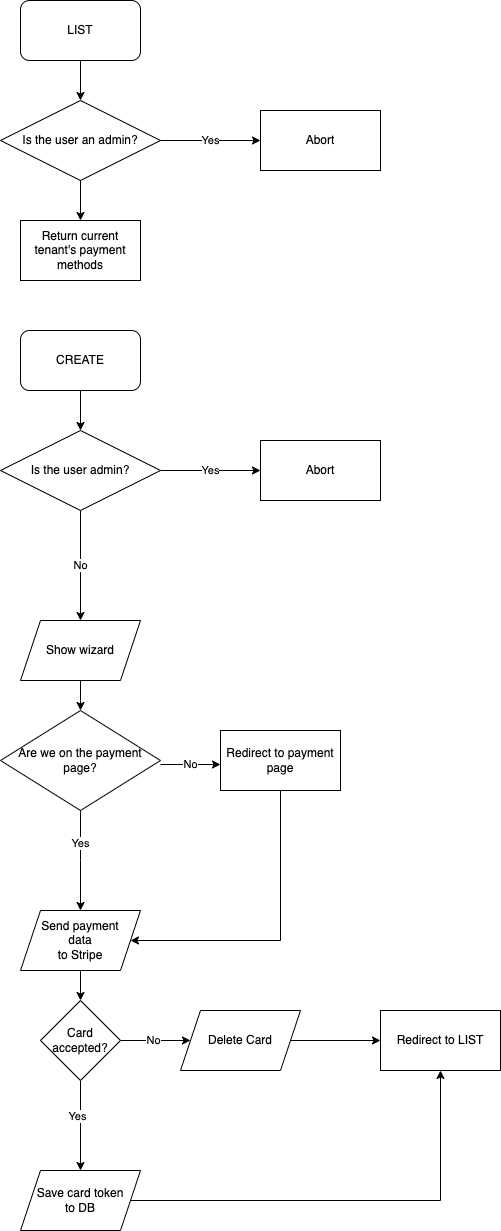
1. Documents Flowchart



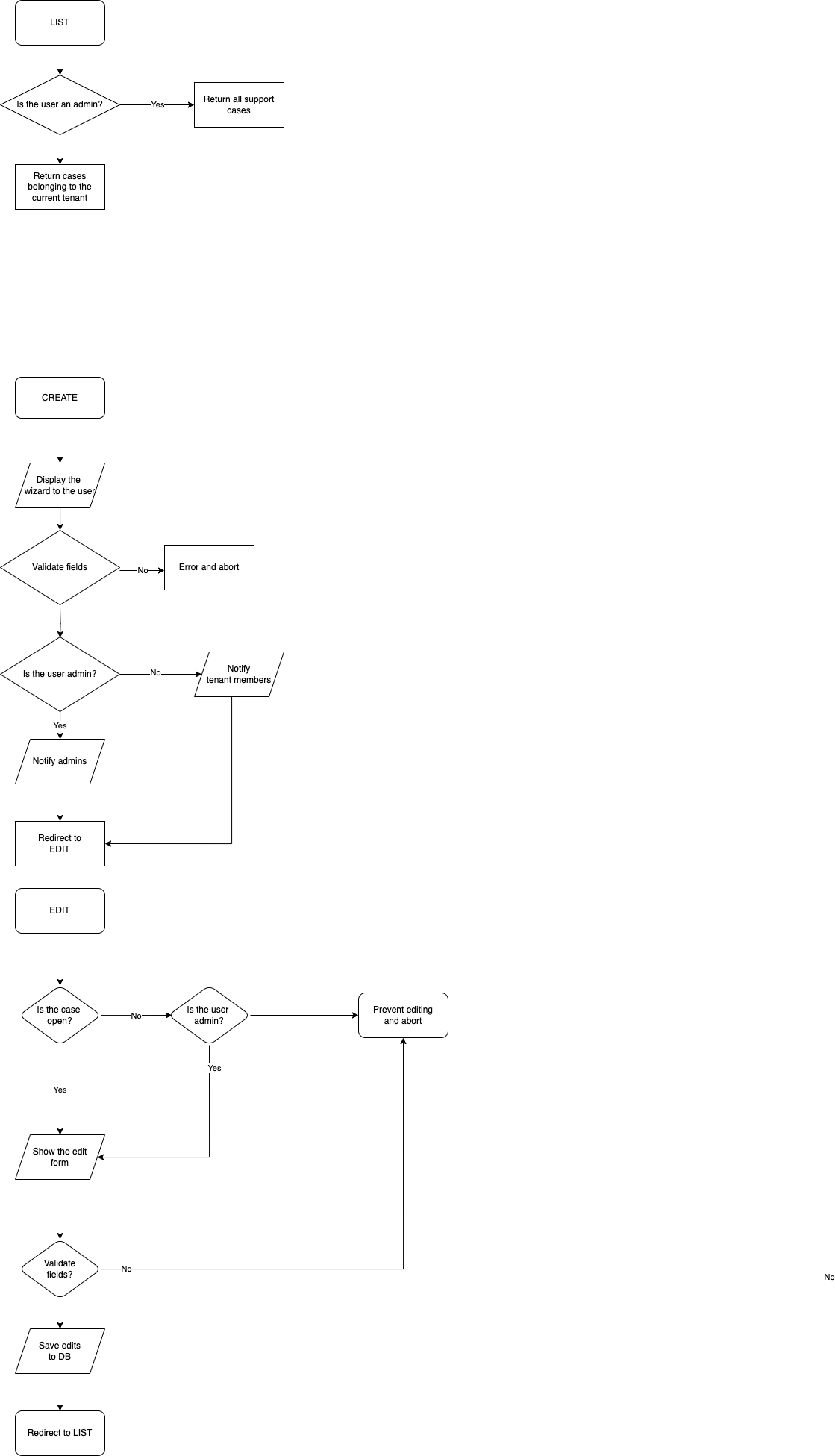
1. Invoices Flowchart



1. Payments Flowchart



1. Support Cases Flowchart



# 

# Database Schema

| **agreements -> Agreement Data** | | |
| --- | --- | --- |
| id | bigint unsigned | Identifying agreements |
| name | varchar(255) | Name of the agreement |
| user\_id | bigint unsigned FK (users, id) | Storing which user is responsible for the agreement |
| description | text | Description of the agreement |
| customer\_subscription\_id | nullable bigint unsigned FK (customer\_subscriptions, id) | Storing for which customer subscription this agreement was generated for |
| project\_id | nullable bigint unsigned FK (projects, id) | Storing for which project this agreement was generated for |
| tenant\_id | nullable bigint unsigned FK (tenants, id) | Storing which tenant this agreement is accessible by |
| sign\_url | nullable text | URL that must be visited to sign this agreement |
| document\_ref | nullable text | Reference number to identify this agreement on external services |
| status | enum('pending','signed','void') | Store the state of this agreement |
| files | text | JSON array of S3 locations of the files related to this agreement |
| created\_at | timestamp | Time record was created |
| updated\_at | timestamp | Time record was last updated |

| **customer\_payment\_methods -> Payment Method Data** | | |
| --- | --- | --- |
| id | bigint unsigned | Identifying payment methods |
| name | varchar(255) | Name of the agreement |
| creator\_id | bigint unsigned FK (users, id) | Storing which user is created the method |
| tenant\_id | nullable bigint unsigned FK (tenants, id) | Storing which tenant this method belongs to |
| name | varchar(255) | Name of the payment method |
| holder | enum('user','tenant') | Info about whether this payment method is registered to the business or its owner |
| type | enum('stripe','bank\_account') | Type of payment method |
| country | nullable varchar(255) | Bank account country |
| account\_no | nullable varchar(255) | Bank account number |
| sort\_code | nullable varchar(255) | Bank account number |
| stripe\_ref | nullable text | Reference number to identify this agreement on payment processor systems |
| brand | nullable varchar(255) | Card brand |
| last4 | nullable varchar(255) | Card last 4 digits |
| created\_at | timestamp | Time record was created |
| updated\_at | timestamp | Time record was last updated |

| **customer\_subscriptions -> Subscription Data** | | |
| --- | --- | --- |
| id | bigint unsigned | Identifying payment methods |
| name | varchar(255) | Name of the agreement |
| creator\_id | bigint unsigned FK (users, id) | Storing which user is created the subscription |
| tenant\_id | nullable bigint unsigned FK (tenants, id) | Storing which tenant this subscription is accessible by |
| name | varchar(255) | Name of the subscription |
| customer\_payment\_method\_id | bigint unsigned FK (customer\_payment\_methods, id) | The primary payment method in charge of paying for the subscription |
| managed\_subscription\_id | bigint unsigned FK (managed\_subscriptions, id) | The template used as a basis of the subscription |
| feature\_list | text | JSON array of features elected from the template |
| website | nullable varchar(255) | Website covered by the subscription |
| description | nullable varchar(255) | Short description of the subscription |
| start\_date | date | Date first subscribed |
| last\_paid\_date | date | Date of last payment |
| next\_renewal\_date | date | Day of the first upcoming renewal |
| currency | varchar(255) | Currency of the fee |
| fee | bigint | Fee of the subscription |
| type | enum(‘monthly’, ‘yearly’) | Subscription term |
| subscription\_id | bigint unsigned FK (subscriptions, id) | Reference to Stripe subscription data provided by Laravel Cashier library |
| status | enum('pending\_documents','pending\_payment','pending\_setup','active','inactive') | Status of the subscription |
| maintenance\_hours | varchar(255) | The hours the customer allowed the client to maintain the website and take it offline |
| created\_at | timestamp | Time record was created |
| updated\_at | timestamp | Time record was last updated |

| **documents -> Document Data** | | |
| --- | --- | --- |
| id | bigint unsigned | Identifying documents |
| name | varchar(255) | Name of the document |
| user\_id | bigint unsigned FK (users, id) | Storing which user is responsible for the document |
| description | text | Description of the document or the request |
| customer\_subscription\_id | nullable bigint unsigned FK (customer\_subscriptions, id) | Storing for which customer subscription this document was requested for |
| project\_id | nullable bigint unsigned FK (projects, id) | Storing for which project this document was requested for |
| tenant\_id | nullable bigint unsigned FK (tenants, id) | Storing which tenant this document is accessible by |
| support\_case\_id | nullable bigint unsigned FK (support\_cases, id) | Storing for which support case this document was requested for |
| status | enum('requested','in\_review','accepted','rejected') | Store the state of this document |
| files | text | JSON array of S3 locations of the files related to this document request |
| created\_at | timestamp | Time record was created |
| updated\_at | timestamp | Time record was last updated |

| **invoices -> Invoice Data** | | |
| --- | --- | --- |
| id | bigint unsigned | Identifying invoices |
| name | varchar(255) | Name of the invoices |
| description | text | Description of the agreement |
| customer\_subscription\_id | nullable bigint unsigned FK (customer\_subscriptions, id) | Storing for which customer subscription this invoice was generated for |
| project\_id | nullable bigint unsigned FK (projects, id) | Storing for which project this invoice was generated for |
| tenant\_id | nullable bigint unsigned FK (tenants, id) | Storing which tenant this invoice is accessible by |
| stripe\_ref | varchar(255) | External reference to identify this invoice on the payment processor systems |
| amount | bigint | The total value of the invoiced goods |
| currency | varchar(255) | Currency of the invoice |
| status | enum('pending','paid','void') | Store the state of this invoice |
| files | text | JSON array of S3 locations of the files related to this invoice |
| customer\_payment\_method\_id | bigint unsigned FK (customer\_payment\_methods, id) | The primary payment method in charge of paying for the invoice |
| created\_at | timestamp | Time record was created |
| updated\_at | timestamp | Time record was last updated |

| **managed\_subscription\_features -> Featured available in a subscription template** | | |
| --- | --- | --- |
| id | bigint unsigned | Identifying features |
| sort | int unsigned | Order of display |
| description | text | Description of the feature |
| name | varchar(255) | Name of the feature |
| extra\_fee | bigint | Extra fees for this feature |
| price\_id | varchar(255) | External payment method reference for the extra fee |
| managed\_subscription\_id | bigint unsigned FK (managed\_subscriptions, id) | The subscription template this feature belongs to |
| created\_at | timestamp | Time record was created |
| updated\_at | timestamp | Time record was last updated |

| **managed\_subscriptions -> Subscription templates** | | |
| --- | --- | --- |
| id | bigint unsigned | Identifying features |
| name | varchar(255) | Name of the template |
| description | text | Description of the template |
| platform | varchar(255) | For what platform does this template apply |
| fee | bigint | Fees for this template if paid annually |
| currency | varchar(255) | Currency of the template pricing |
| stripe\_reference | varchar(255) | External reference in payment processor systems |
| monthly\_fee | bigint | Fees for the subscription if paid monthly |
| worth | bigint | How much this service would cost you if you did not purchase it from the client |
| created\_at | timestamp | Time record was created |
| updated\_at | timestamp | Time record was last updated |

| **project\_platforms -> Platforms suggested for a project** | | |
| --- | --- | --- |
| id | bigint unsigned | Identifying platforms |
| sort | int | Display order |
| project\_id | nullable bigint unsigned FK (projects, id) | Storing for which project this platform was suggested for |
| name | varchar(255) | Name of the template |
| description | text | Description of the template |
| additonal\_project\_fee | bigint | Additional development costs |
| additional\_fee | bigint | Additional fees incurred by the use of the platform |
| additional\_days | bigint | Additional time required to develop for the platform |
| estimated\_cost | bigint | How much this service would cost you if you did not purchase it from the client |
| link | varchar(255) | Link to learn more about the platform |
| hidden | tinyint(1) | Bool deciding whether to show platform in the final project overview |
| logo | varchar(255) | S3 url to platform logo |
| samples | text | JSON array of S3 urls to samples built with the platform |
| created\_at | timestamp | Time record was created |
| updated\_at | timestamp | Time record was last updated |

| **project\_requirements -> Requirements set forth for a project** | | |
| --- | --- | --- |
| id | bigint unsigned | Identifying requirements |
| sort | int | Display order |
| project\_id | nullable bigint unsigned FK (projects, id) | Storing for which project this requirement was suggested for |
| adder\_id | nullable bigint unsigned FK (users, id) | Storing which user proposed the requirement |
| required | tinyint(1) | Bool deciding if this is a requirement or not a requirement |
| name | varchar(255) | Name of the template |
| description | text | Description of the template |
| fee | bigint | Additional development costs |
| additional\_days | bigint | Additional time required to develop the requirement |
| created\_at | timestamp | Time record was created |
| updated\_at | timestamp | Time record was last updated |

| **projects -> Project proposals** | | |
| --- | --- | --- |
| id | bigint unsigned | Identifying requirements |
| sort | int | Display order |
| project\_id | nullable bigint unsigned FK (projects, id) | Storing for which project this requirement was suggested for |
| creator\_id | nullable bigint unsigned FK (users, id) | Storing which user proposed the project |
| tenant\_id | nullable bigint unsigned FK (tenants, id) | Storing which tenant this project is accessible by |
| name | varchar(255) | Name of the template |
| requirement\_description | text | Overall summary of project requirements |
| platforms | text | JSON Array of platforms requested by the customer |
| status | enum('submitted','in\_review','accepted','rejected','upfront\_payment\_pending','in\_progress','pending\_payment','completed') | Storing project status |
| purpose | text | Purpose of the project |
| type | enum('new\_project','maintenance','upgrade') | What type of project is this? |
| existing\_solution | nullable text | How is the problem the project will need to solve currently solved? |
| existing\_infrastructure\_details | nullable text | What infrastructure is used to achieve the current solution |
| buget\_min | bigint | Starting budget |
| budget\_max | bigint | Maximum budget |
| fee | nullable bigint | Development costs |
| upfront\_fee | nullable bigint | Portion of the costs that must be paid upfront |
| additional\_fee | nullable bigint | Fees that must be paid to external vendors |
| currency | varchar(255) | Currency of all project pricing |
| payment\_methods | text | JSON Array of payment method types that are available to the proposer |
| requested\_delivery\_date | date | The date the customer wisher the project completed |
| expected\_delivery\_date | nullable date | When the client expects to deliver the project to the customer |
| delivery\_est\_min | nullable bigint | Minimum days required to deliver project |
| delivery\_est\_max | nullable bigint | Maximum days required to deliver project |
| whitelabel | tinyint(1) | Bool deciding whether white labeling is required. |
| requires\_financial\_services | tinyint(1) | Bool deciding whether financial transactions are involved. |
| customer\_notes | nullable text | Notes added by the customer |
| team\_notes | nullable text | Notes added for the customer by the team members |
| internal\_notes | nullable text | Notes added for the team members by team members, hidden from the customer. |
| attachments | text | JSON array consisting of S3 storage paths of project attachments. |
| project\_manager | nullable bigint unsigned FK (users, id) | Storing which member of the team is going to receive customer’s updates. |
| stripe\_ref | varchar(255) | Reference number of the project in payment processor systems. |
| created\_at | timestamp | Time record was created |
| updated\_at | timestamp | Time record was last updated |

| **signatures -> Signature Data** | | |
| --- | --- | --- |
| id | bigint unsigned | Identifying signatures |
| user\_id | bigint unsigned FK (users, id) | Storing which user created the signature |
| description | text | Description of why the signature was collected |
| customer\_subscription\_id | nullable bigint unsigned FK (customer\_subscriptions, id) | Storing for which customer subscription this signature was requested for |
| project\_id | nullable bigint unsigned FK (projects, id) | Storing for which project this signature was requested for |
| tenant\_id | nullable bigint unsigned FK (tenants, id) | Storing which tenant this signature was created in |
| ip | varchat(255) | IP address of the user providing the signature |
| signature | text | Base64 encoded png data of the signature drawn using computer touchpad |
| created\_at | timestamp | Time record was created |
| updated\_at | timestamp | Time record was last updated |

# Test Plan

| Test Case | Methodology | Data | Expected Output |
| --- | --- | --- | --- |
| Authentication works | Visiting the web application URL and signing in through SSO. | Correct auth credentials. | User reaches the dashboard. |
| Creating a project request | Fill out the details of a project in the new project wizard. | All data filled out and the project type selected as “New Project” or “System Maintenance” or “Upgrade” | Project request created, admins emailed |
| All data but existing solution filled out and the project type selected as “New Project” |
| All data but existing solution filled out and the project type selected as “System Maintenance” or “Upgrade” | Project creation request fails |
| All required fields are not filled out |
| Editing a project request | Open the details of a project, change the requested details and hit save. | All required fields filled out and of the correct type. Project is not locked (either submitted or rejected) or the user is an admin. | Project is updated and notifications are sent. |
| Not all required fields filled out or not all are of the correct type. | Update fails and no notification is sent. |
| All required fields filled out and of the correct type. Project is not locked (either submitted or rejected) and the user is not an admin. |
|
| Paying for a project | Open a project that is pending payment, choose a payment method and sign. Finally, hit pay. | Missing signature or payment method. | No invoice generated. Payment fails and no change is made to the project. |
| Card does not have enough balance or the card transaction fails.   Stripe test card:  4000000000009995  CVC: 100  Exp: Any future date | Invoice generated but is unpaid. No change to the project status. |
| Card charged successfully.  Stripe test card:  4242424242424242  CVC: 100  Exp: Any future date | Payment successful. Project status transitions to the next stage. |
| Paying for a subscription | Open a subscription that is pending payment, choose a payment method and sign. Finally, hit pay. | Missing signature or payment method. | No invoice generated. Payment fails and no change is made to the subscription. |
| Attempt to pay monthly subscription using a payment method that is a bank account. |
| Card does not have enough balance or the card transaction fails.   Stripe test card:  4000000000000341  CVC: 100  Exp: Any future date | Invoice generated but is unpaid. No change to the subscription status. |
| Card charged successfully.  Stripe test card:  4242424242424242  CVC: 100  Exp: Any future date | Payment successful. Subscription status transitions to the next stage. |
| Signing an agreement | Open a pending agreement that is related to a subscription. | Document not signed. | Subscription is pending agreements. |
| Document signed by one of the signatories but there are multiple. |
| All signatories of the document have signed the agreement. | Subscription transitions to payment pending. |
| Creating a support case | Go to the support cases resource, hit create, fill out the details and once again, hit create. | Type is within the category Success Manager and all details are filled. | Notification sent to admins and case created. |
| Type is either within the Project category or Subscription category and a Project or Subscription can be and is selected respectively. |
| Message is too short or no subject. | Case is not created and error is returned. |
| Case is about a Project or Subscription but there are no projects/subscriptions or none are selected. |
| Requesting a document | Go to the Documents section, click New, fill the form and hit create. | User is not an admin. | Request is blocked. |
| Required fields are not filled out. |
| Selected user is not in the selected tenant. |
| All fields filled out and are of the correct type. | Document requested and email sent to customer. |

# 

# Record of Tasks

| Task No: | Planned Action | Planned outcome | Time estimated | Target completion date | Criterion |
| --- | --- | --- | --- | --- | --- |
| 1 | Initial Meeting with the Advisor | A detailed chat transcript is noted down. Requirements of the client understood. | 1hr | 04/24 | A |
| 2 | Success criterion are planned. | A list with all SC is created. | 2hr | 04/24 | A |
| 3 | Relevant frameworks and programming languages are researched. | Programming languages and frameworks suitable for the multiple platforms of the application are chosen. | 3hr | 04/24 | A |
| 4 | Second meeting with the advisor. | Proposed solution is discussed and the SC are finalized. | 1hr | 05/24 | A |
| 5 | Class Diagram | Database structure and possible object fields are planned | 4hr | 06/24 | B |
| 6 | Interface Design | A UI/UX design to bring to the table to the client is created | 6hr | 06/24 | B |
| 7 | Testing Plans | A test plan to validate functionality of the program is created. | 3hr | 06/24 | B |
| 8 | First Prototype | Initial functionality is programmed.No API integration is made but rather manual inputs are used. | 10hrs | 06/24 | C |
| 9 | Second Prototype | A test integration with Stripe is made for testing financial transactions. | 10hrs | 06/24 | C |
| 10 | Meeting with the Client | Confirm the progress in the right direction, add additional test scenarios if needed and update SC. | 2hrs | 06/24 | C |
| 11 | Third Prototype | API integration is completed. Stripe webhooks integrated into the application. | 6hrs | 07/24 | C |
| 12 | Fourth Prototype | User interface is completed and all functionality is made available from the interface.  Authentication is completed. | 10hrs | 07/24 | C |
| 13 | Final Prototype | All permissions are confirmed and added into the project. User role checking, input validation and permission checks are put into place. | 10hrs | 07/24 | C |
| 14 | Release Candidate | An RC version is deployed and made available for the advisor to test. | 4hrs | 07/24 | C |
| 15 | Final Evaluation | A final meeting with the advisor is made to confirm functionality. | 2hrs | 07/24 | C |