

Payment

User Story:

As a brand, I should be able to choose between two payment methods: "Pay First" and "Pay as you go" so that I can have flexibility in payment.

Tasks:

Payment Method: Pay First

- Add an option for brands to choose the "Pay First" method.
- Allow brands to set a campaign budget and deposit money into their AIMS account.
- Track and display the deposited money in the brand's account.
- Deduct the agreed-upon amount from the brand's AIMS account when they contract an influencer.
- Transfer the deducted money to the influencer's virtual wallet.
- Record and display all transactions related to the "Pay First" method.
- Create a process for the brand to confirm when the influencer has completed the work.
- Implement a validation step to ensure the work aligns with the contract terms.
- If the work is satisfactory, transfer the money from the influencer's virtual wallet to their actual account balance.
- Provide users with a transaction history log to track all "Pay First" payment activities.

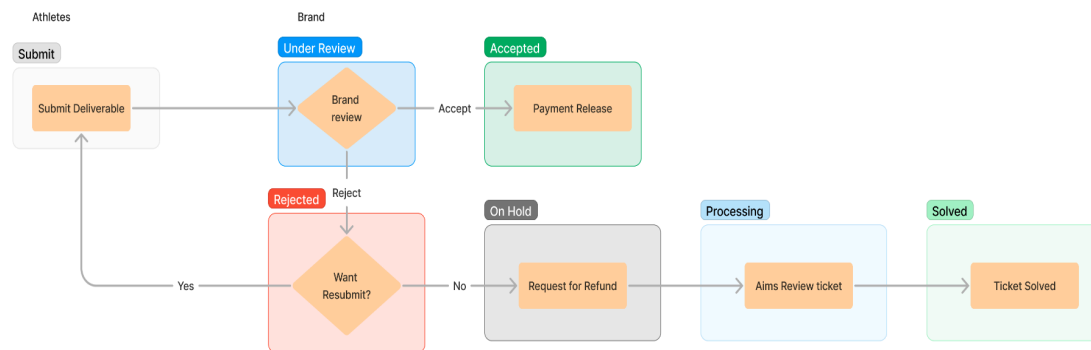
Payment Method: Pay as you go

- Add an option for brands to choose the "Pay as you go" method.
- Implement a mechanism for brands to make payments each time they contract an influencer.
- Deduct the agreed-upon payment amount from the brand's account and update the influencer's virtual wallet.
- Create a process for the brand to confirm when the influencer has completed the work, similar to the "Pay First" method.
- Implement a validation step to check if the work is done according to the contract terms.
- If the work is satisfactory, transfer the money from the influencer's virtual wallet to their actual account balance.
- Offer a transaction history log for both "Pay as you go" brands and influencers to track all payment-related transactions.

Task Submission and Review

Tasks:

1. **Submission**- influencer will submit each task under each deliverable.
2. **Task Review**- Brand will review the task. Each task can be rejected/requested for revision(any number of times).
3. **Task Submission History**- A list of submissions with the appropriate tags(accepted, rejected). For now we will keep the revisions infinite from the brand's side, later we will define the boundary.
4. **Wallet Update**- When the brand finally updates the task as accepted, the payment is ready to withdraw. The brand can not change the accepted status once it is accepted.



Steps for Task submission feature

1. **Submitted**- Athletes submit the task
2. **Under Review**- Brand review the task
3. **Accept**- If the brand agrees it can accept the task.
4. **Resubmit**- If the brand does not agree it can ask for resubmit with the required suggestions i.e. Influencer has a dialogue in the video which is against the brand's value, so the brand can ask for change in the video.
5. **Rejected**- The brand can totally reject the submitted task whatsoever. In this case the brand can ask for a refund from AIMS. Suppose, the influencer could not provide the task on time, so the brand has the right to reject the task and ask for the refund. In this case a refund request will be sent to the AIMS admin panel with the required info - the contract, the submitted things(video, image), submission-resubmission history, the brand and influencer's email and phone number etc.
6. **On Hold**- AIMS will check the refund request from the admin panel. They will communicate with the brand and influencer through mail/phone. Finally the AIMS admin should be able to mark the task accepted or rejected from the admin panel.

For reference admin panel design-

<https://www.figma.com/file/s8JFuWnyKMqmvBeERduyO4/AIMS-Refund-Admin?type=design&node-id=0%3A1&mode=design&t=mlFbIBjZvfV8OaOs-1>

7. **Processing-** When the admin checks the refund request from the admin panel, the status will be processed. After the processing is done, the admin will mark the process either accepted or rejected. If accepted, the money will be updated in the **Available balance** account from the **value of active campaigns** account. If the task is rejected, the amount brand paid for that task would be cut from the **value of active campaigns**. The brand will get a payment update in the payment section. The AIMS admin will then send the money to the brands manually.

The design-

<https://www.figma.com/file/Egu2IEayC5UZxLtIQxLzV6/Aims-v3-for-Dev?type=design&node-id=1621-23032&mode=design>

Payment Transfer

Tasks:

1. **Wallet Update** - The payment is credited but not made withdrawable to the influencer's wallet once the task is locked by both the parties.
2. **Make Funds Withdrawable-** When the brand finally accepts the task, the money for the task is made withdrawable to the influencer.
3. **Withdrawal-** The influencer will ask for the money in their bank account. AIMS will approve the transfer of the money to the account. (First done manually but later the process will be automated)

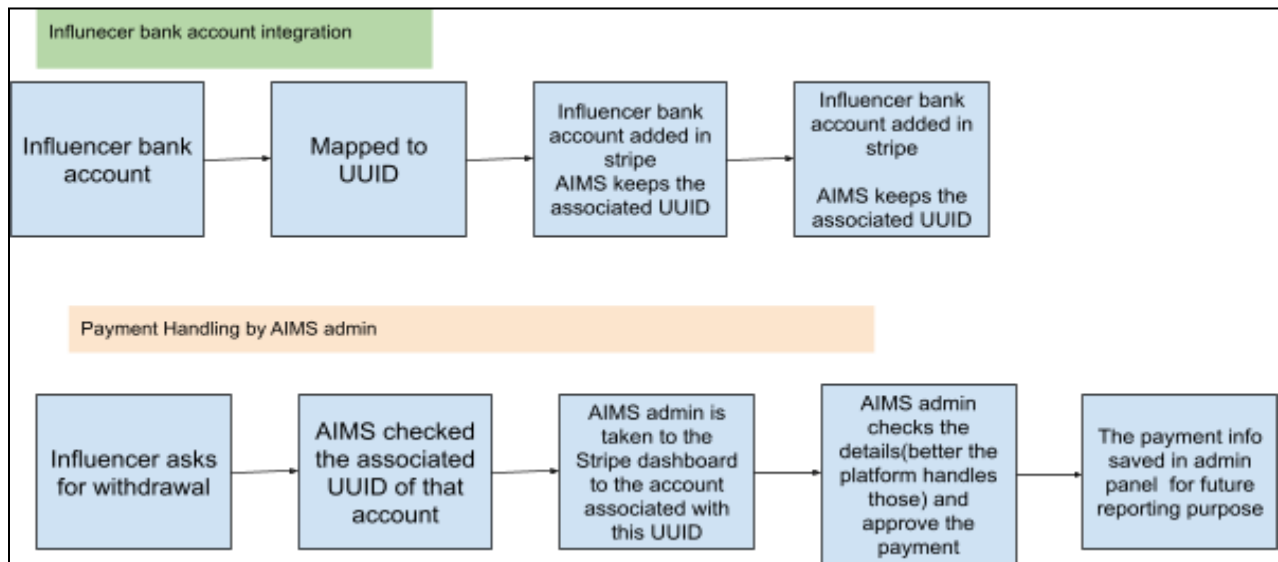
Payment Transfer Automation

1. Influencer Wallet and Withdrawal Request
 - 1.1. Add an option to let the influencer ask for a withdrawal
 - 1.2. AIMS will check if the withdrawal is valid(i.e. Influencer can not withdraw less than 100 dollars and can not request more than twice a month)
 - 1.3. Add an option to the influencer to provide bank information to AIMS(We want to handle this through Stripe APIs)
2. Aims Admin Withdrawal request handling
 - 2.1. Need an interface that will connect to the Stripe's payout API and save the bank information from the influencer
 - 2.2. The AIMS admin can check and review the withdrawal information(the previous withdrawals and the current balance)
 - 2.3. An interface connected to stripe(through iframe or Stripe's API integration) where the admin can approve the payout to the requested accounts. Here the following idea can be useful-

AIMS take the bank account and map it to a UUID → AIMS keep the UUID and send the data to stripe payout dashboard → AIMS gets the UUID when a withdrawal request is provided → AIMS admin is taken to the stripe dashboard against the specific UUID → AIMS admin approve the payment against the bank account of that UUID.

3. Reporting

- 3.1. AIMS will have the transaction reports in the admin panel as transaction history for reporting purposes.
- 3.2. The transaction history must contain the necessary info like the transaction id, amount, date, commission amount.



N.B. The bank information should be taken automatically to the stripe dashboard and the payment must be made through in a more automated way. In my POV, stripe Connect can be a good option for that, please check and confirm if it is applicable.

Notes on Transfer:

- <https://stripe.com/docs/payouts>
- <https://stripe.com/guides/earn-revenue-by-issuing-cards>
- <https://stripe.com/docs/payouts/instant-payouts-banks>
- <https://github.com/konrad-g/simple-marketplace>
- <https://wise.com/>
- <https://www.toptal.com/ecommerce/create-a-marketplace-tutorial>