TMPay

Project Documentation Summary - Team Marc Transaction Manager

Project Overview:

The TMPay Transaction Manager is an application designed to streamline and automate the deposit and withdrawal process for agents and players within a casino network. The app allows users to submit deposit and withdrawal requests, and the admin has the authority to approve, reject, or request more information for each transaction. Additionally, the app facilitates a banking management system, enabling users to store their bank and e-wallet account details for easy selection during transactions.

▼ **Key Features and APP workflows and overview.**

1. **User Manager**

   - Allow users to sign up and select their role as either "Agent" or "Player."

     ○ Flow

     ○

   - Entries (Fullname, email, contact number, , team name, username and password. Provide a dropdown menu for users to choose if an agent or player that will serve as values to validate against a preloaded database (from the downloaded reports on the casino platform in csv or excel format)

or by server side api call to get hierarchy based from the user input (key: username:string, isAgent:true/false) as a query parameter.

- Admin (owner can set the limitations)
- Owner panel

1. Payment Manager**:**

2. **Admin Dashboard:**

   - As the admin, you have a dashboard to view all pending, approved, and rejected transactions.

   - Implement filtering and sorting options to easily manage and track transactions and reports.

   - Add a feature to request more clarification from the user if needed, with a notification system for communication.

3. **Approval Workflow:**

   - Enable you as the admin to approve, reject, or send back (request more information for a selected transaction)

   - For approved deposits, automatically transfer the chips to the user's casino affiliate account.

   - For approved withdrawals, process the payment to the specified bank or e-wallet using either manual sending (if the user selected a bank account from the listed accounts of the admin) or online automatic transfer if the user selected a payment method with a payment gateway integrated in the system and then update the user and admin about the transaction status.

4. **Integration with Casino agents Platform and Payment Gateways**

   💡 Integrate the app with the casino affiliate platform by rest api to manage the digital chips and user accounts efficiently. For more detailed info about casino platform interegration, please click the link below.

   💡 <u>Refer on the documentation of casino on this page:</u>

- Api for gethierarchy will be used on the following features:

☐ note that the get hierarchy function is mapping all user in the casino including those who are outside of my team without the auth token that is provided to me that is limited to use to do deposit and withdraw on my networks. Team Marc consist of 3 accounts on different levels with no connection to each other so it means that i have 3 separate auth tokens

- Verification of a user if a member of team marc and to which account that username belongs to. I have 3 agent accounts so we have to set a loop of conditions to identify which token we will use for the transactions

- identification of user if agent or player

- identification of user if allowed to do transaction due to rebates

- validation if it is allowed for a requested username by an agent who might request a transaction for a certain player that is not on that agent network (fraud agents)

💡 Integrate the app with the payment gateway providers by rest api to manage the automatic payments seamlessly and efficiently.

The documentation of the payment gateway and the postman collection is attached below.

GoVIPCenter-CashFlex-API-Gold-V_1_1_0.pdf

GoCashFlex (Gold) V4 2.postman_collection

5. **Real-time Notifications:**

- Implement real-time notifications for users and admin to receive updates on their transaction status.

6. **Security and Privacy:**

- Prioritize security measures, such as encryption and authentication, to safeguard sensitive user data and transactions.

- OTP and Sign Up features

7. **Transaction History and Reporting:**

- Provide users with a transaction history section to view past transactions and their statuses.

- Generate reports for owner and admin to analyze transaction trends and performance.

8. **Payment Method Management:**

- Allow the admin to manage the payment methods (manual or online) and update them as needed.

9. **Support:**

- Integrate a chat support feature to facilitate communication between users and the admin for immediate assistance. (I have a chatbot to integrate on this feature)

- Option to integrate smtp (sms,email, internal app messaging)

- Option to use OTP (using An api call to send message to the agent/player account on the casino platform side), this is also one line on security mesures to ensure that the requesting user is not fraud. This is due to the limitation of the casino platform that I have no access to view their password on the casino platform for more secured validation.

Additional Features:

1. **User Banking Management:**

- All users (agents and players) can access a "Banking Management" section in their profile.

- In this section, users can add, edit, and delete their bank and e-wallet account details.

- Provide a user-friendly UI/UX for adding new accounts, including a selection of banks/e-wallets with the option to upload images for visual identification.

- Users can set a default account for transactions, which will be preselected during withdrawal requests.

2. **Admin Bank Account Management:**

- Each admin has their own login and access to the "Admin Dashboard."

- In the dashboard, each admins can manage the available bank and e-wallet accounts that is assigned to them that users can use as reference or by selecting the activated accounts visible to all users in real time for deposits. And selection of account to be used when appoving requested withdrawals to dynamically update the value (bank/ewallet funds) in the database for efficient report details.

- Admins can add new bank accounts and e-wallets to the system, upload images for visual identification, and set limits for each account.

- Include a switch to activate or deactivate specific accounts when they reach their transaction limit where in of the selected account is in activate mode, all users will see in their dashboard in real time.

- Owner should also have a separate banking management wherein the admins can send funds and vice versa

3. **Withdrawal with Banking Selection:**

- When users make a withdrawal request, an amount input form and a drop down that they can select from their saved bank and e-wallet accounts in the form where the admin will send money upon approval.

- The form will display the user's account details based on their selection, making the withdrawal process faster and more convenient.

4. **Transaction Limit Alert:**

- Implement an alert system for admins when a bank account or e-wallet reaches its transaction limit. Example is an ewallet daily incoming limit is 100k and the outgoing limit is 100k with a monthly limit of 500k on incoming

- Admins can be notified to take appropriate actions, such as activating a new account or updating the transaction limit.

5. **Admin Shift Management:**

- Shift will be recorded based from the log in and all transactions should reference which admin account did the process. Include also a detection of ip address and location that is only visible to the owner user account

6. **UI/UX Design:**

- Ensure the app has a simple, intuitive, and visually appealing UI/UX for easy navigation and interaction.

7. **Testing and Usability:**

- Conduct thorough testing of the banking management features to ensure usability and seamless user experience.

Additional Features:

1. **Owner Access to Transaction Reports:**

- Include a separate "Owner Dashboard" that provides me, as the owner, access to view all transaction reports.

- The owner dashboard should have detailed transaction histories and filtering options to easily find missing money or identify conflicts with the casino platform wallet balance, admin bank accounts and payment gateways bank accounts

2. **Shift Admin Handover and Manual Encoding:**

- Implement a feature for shift admins to create handover reports at the end of their shift.

- In the handover report, use cases like allow admins to manually enter the amount of money sent to colleagues (backup fund) with accompanying screenshots of the online bank/ewallet transaction and the wallet balance on the casino platform dashboard. Access to edit this features can be given to the admin on duty and the admin who will send the fund if in case of emergency that is needed to send even out of duty for bettter traceability.

- The new shift admin should be able to view and verify the handover report on the casino platform wallet at the start of their shift.

- Another use case here is when an outgoing admin received an amount just right before loging out if the system. The incoming admin will see that an

amount was sent to the outgoing shift but the proceess of the approval will be carried out by the incoming shift. Kindly create a flow to solve this issue that might become a conflict on the reports.

- Another use case here is that when an out going admin made a topup request and money has been sent to pay the casino but the topup process had a delay and the shift already ended. So the fund recorded on the database has been deducted but the fund to be received on the admin casino wallet is on pending that only the incoming shift should process the entry manually.

3. **Expenses Field for Withdrawals:**

- Add an "Expenses" field in the admin panel for shift admins to input expenses related to withdrawals, such as bank charges, salary, operation related expenses and a special entry in database when they are sending money to the owner.

4. **Commission from Casino Platform:**

- Include a field in the admin panel where you can manually add the commission received from the casino platform every 15 days.

- The app should update the available funds based on the added commission automatically.

5. advance commision field to also be added with a use cases of if I as owner received an advance commission from the casino and also if some agents requested for a commission advance from me.

Additional Features: Rebates

1. **Admin Wallet Rebates:**

- The app includes a dedicated "Admin Wallet" where rebates will be added manually by the admin.

- Admins can set custom rebates percentages for eligible users (e.g., agents) in the "Rebates Table."

- Whenever an eligible user (e.g., agent) initiates a deposit transaction, the app dynamically calculates the rebate amount based on the user's deposited amount and the predefined rebates percentage.

- For deposit transactions, the app adds the calculated rebate amount to the agent's wallet balance.

- Simultaneously, the app deducts the rebate amount from the casino wallet balance, ensuring accurate cross-referencing with the actual casino platform balance (API integration is not available; rebates are manually managed).

- A dynamic field called "Rebates Given" displays the total rebates given to the eligible user from their deposit transactions.

2. **Rebates Summary:**

- The "Rebates Summary Table" tracks and consolidates the total rebates given and taken for each agent.

- Whenever an agent initiates a withdrawal transaction, the app deducts the rebate amount from the agent's wallet balance (if applicable) and records the value in the "Rebates Summary Table" under "Rebates Taken."

- For deposit transactions, the app calculates the rebates given based on the predefined rebates percentage and records the value in the "Rebates Summary Table" under "Rebates Given."

- The "Rebates Summary Table" ensures an accurate overview of the rebates transactions for each agent.

3. **Rebates Received from Casino:**

- When an admin makes a top-up on behalf of the casino platform, the app records the corresponding rebate amount received from the casino platform in the "RebatesSummary Table" under "Rebates Received from Casino."

- The app ensures proper bookkeeping of the rebates received from the casino for top-up transactions.

4. **Rebates Given to Casino:**

- In case of cash-out assistance requests, the app records the rebate amount given to the casino in the "Rebates Summary Table" under "Rebates Given to Casino."

- The app ensures transparent tracking of rebates given to the casino for cash-out assistance.

5. **Rebates Management:**

- The "Rebates Table" in the admin panel allows admins and the owner to manage the rebates percentages for eligible users (e.g., agents).

- Admins and the owner can add, edit, or delete rebate percentages, ensuring flexibility in managing rebates policies.

Please note that the rebates are dynamically calculated based on the data input, such as deposit, withdrawal, admin top-ups, and cash-out assistance requests. The "Rebates Table" offers an intuitive interface for adding, editing, or deleting rebate percentages. The "Rebates Summary Table" maintains an accurate and comprehensive overview of all rebate-related transactions, ensuring transparency and efficient management.

User Interface Enhancements:

- Design the dashboards with clear visual representations of transaction data, making it easier to spot discrepancies or trends.

- Implement user-friendly interfaces for manual encoding, with intuitive fields for entering information and uploading screenshots.

Security and Data Privacy:

- Prioritize data security and access control to ensure that only authorized personnel have access to sensitive information.

Testing and Iteration:

- Conduct thorough testing to ensure the app's functionality, usability, and security.

- Gather feedback from users, admins, and yourself to iterate and improve the app based on real-world usage.

As the app becomes more comprehensive, working with experienced developers and designers will be beneficial to ensure smooth implementation and a user-friendly experience. Regularly update the app to incorporate improvements and adapt to changing needs.

Remember to implement robust security measures to safeguard user banking information and ensure that only authorized admins can manage the available accounts. The UI/UX design should be user-friendly and responsive to cater to users on different devices.

Follow best practices for data storage, user privacy, and payment processing security. Consider using reputable payment gateways for secure and reliable payment transactions. Testing and iterating the app with a limited user base before full-scale deployment can help identify and fix any issues or improvements needed.

Developing this app required in web development, databases, and mobile app development, to ensure a successful and user-friendly application.

Conclusion:
The TMpay Transaction Manager offers a comprehensive solution for managing transactions, banking details, rebates,   within the casino network. By automating the process and providing real-time updates, the app improves efficiency and transparency while ensuring a secure and user-friendly experience for agents, players, and administrators.

[TM Admin](#)

[Casino API](#)

[User Database (This is just a draft and just my idea)](#)

[Design System](#)

[Untitled](#)