Product Requirements Documentation

Version: 1.0.0 Date: 04/02/2024 Author: Sahil Pahwa

2. Table of Contents:
 - A list of all sections with corresponding page numbers.

Brief overview of the product:
A payment gateway that enables real-time splitting of payments. Combines all transactions made at once and transfers money through "FairSplit" account.

Purpose and goals of the product:
To ease the process of cashless splitting and avoid embarrassment of asking for money later on.

Product Overview:

- ### Introduction:

FairSplit is a cutting-edge payment gateway designed to revolutionise how individuals and groups manage shared expenses. Whether it's splitting a restaurant bill with friends, dividing household expenses with roommates, or managing group expenses for a trip, FairSplit streamlines the process by seamlessly combining all transactions made at once and transferring funds through a shared "FairSplit" account.

- ### Features:

1. Real-Time Splitting: FairSplit allows users to split payments in real-time, eliminating the hassle of manual calculations and subsequent reimbursements.

2. Centralised Transactions: All transactions made within a specified time frame or among a designated group are automatically aggregated and processed together.

3. FairSplit Account: FairSplit provides users with a dedicated "FairSplit" account, where funds are temporarily pooled before being distributed among participants.

4. Customizable Splitting Rules:Users have the flexibility to customise splitting rules based on various parameters such as equal sharing, percentage allocation, ratio or specific amounts per participant.

5. Secure Payment Processing: FairSplit prioritises security and utilises robust encryption protocols to ensure the safety of users' financial information and transactions.

6. Transaction History: Users can easily track and review their transaction history within the FairSplit platform, providing transparency and accountability.

7. Integration with Existing Platforms: FairSplit seamlessly integrates with popular payment platforms and mobile wallets, allowing users to make payments using their preferred methods.

8. Notification and Alerts: Users receive real-time notifications and alerts regarding new transactions, pending payments, and fund distributions, keeping them informed every step of the way.

9. Cross-Border Support: FairSplit supports cross-border transactions, enabling users to split payments with friends or colleagues regardless of their location.

- How FairSplit Works**:**

1. **Initiating a Transaction**: Users initiate a transaction by selecting the option to split a payment using FairSplit within their preferred payment app or platform.

2. **Automatic Aggregation:** FairSplit automatically aggregates all transactions made within a specified time frame or among a designated group of participants.

3. **Funds Pooling:** The total amount owed by each participant is pooled into the shared "FairSplit" account, ensuring a centralised and equitable distribution of funds.

4. **Splitting Calculation:** FairSplit calculates the amount owed by each participant based on the predetermined splitting rules, taking into account factors such as the total amount, number of participants, and allocation preferences.

5. **Transfer and Distribution**: Once the splitting calculation is complete, FairSplit transfers the respective amounts from the shared account to the destination account, facilitating a seamless and equitable distribution of funds.

Benefits:

- Convenience: FairSplit eliminates the need for manual calculations and individual reimbursements, saving users time and effort.

- Fairness: By pooling funds and applying customizable splitting rules, FairSplit ensures fair and equitable distribution of expenses among participants.

- Transparency: Users have full visibility into their transaction history and fund distributions, promoting transparency and accountability within the group.

- Security: FairSplit prioritises the security of users' financial information and transactions, providing peace of mind during the payment process.

Target audience and users:
FairSplit does not aim to limit its audience, but intended to target the youth of the age group 18-27.

Functional Requirements:

1. User Registration and Authentication:
   - Users should be able to register for an account on the FairSplit platform.
   - Authentication mechanisms, such as email verification or 2FA.

2. Payment Splitting:
   - FairSplit should allow users to initiate payment splitting for shared expenses.
   - Users should be able to specify the total amount to split and the participants involved in the transaction.

3. Transaction Aggregation:
   - The system should automatically aggregate all transactions made within a specified time frame or among designated groups.
   - Aggregation should occur in real-time to provide up-to-date information on shared expenses.

4. FairSplit Account Management:
   - FairSplit should provide users with a dedicated "FairSplit" account where funds are temporarily pooled.
   - Users should be able to view the balance of their FairSplit account and track transactions related to the account.

5. Splitting Rules Configuration:
   - Users should have the ability to configure splitting rules based on various parameters, such as equal sharing, percentage allocation, or specific amounts per participant.
   - The system should support customizable splitting rules to accommodate different scenarios and preferences.

6. Payment Processing:
   - FairSplit should integrate with payment gateways and financial institutions to facilitate secure payment processing.
   - Payments should be processed in real-time, and funds should be transferred from participants' accounts to the FairSplit account.

7. Notification and Alerts:
   - The system should send real-time notifications and alerts to users regarding new transactions, pending payments, and fund distributions.
   - Notifications should be delivered via email, SMS, or in-app notifications.

8. Transaction History and Reporting:
   - FairSplit should maintain a comprehensive transaction history for each user, including details such as transaction date, amount, participants, and splitting rules.
   - Users should be able to generate reports and export transaction data for accounting or record-keeping purposes.

10. Integration with External Platforms:

- FairSplit should integrate seamlessly with popular payment platforms and mobile wallets, allowing users to make payments using their preferred methods.
- Integration APIs should be provided to enable third-party developers to integrate with the FairSplit platform.

11. Admin Dashboard:
    - An admin dashboard should be provided to manage user accounts, monitor system activity, and perform administrative tasks.
    - Admins should have access to user management features, transaction monitoring, and system configuration settings.

12. QR Invitation:

-FairShare should enable users to generate and share QR codes to invite others to join shared expenses or events.

-Invited users should be able to scan the QR code using their mobile devices to seamlessly join the shared expense or event.

13. Phone Number Authentication:

-FairShare should support phone number authentication as an alternative or additional authentication method for user registration and login.

-Users should be able to verify their phone numbers by entering a verification code sent via SMS during the registration process.

14. Payment Search:

-FairShare should provide users with the ability to search for specific payments or transactions based on various criteria.

-Users should be able to search for payments by date range, transaction amount, participant names, or transaction ID.

Non-Functional Requirements:

1. Performance Expectations:

-Response Time: The web application should have a fast response time, with pages loading within 2-3 seconds on average.

-Scalability: The system should be scalable to accommodate a large number of concurrent users and transactions, ensuring consistent performance under varying load conditions.

-Throughput: The system should be capable of handling a high volume of simultaneous transactions without experiencing performance degradation.

Security Requirements:
-Data Encryption: All sensitive data, including user credentials, financial transactions, and personal information, should be encrypted both in transit and at rest using industry-standard encryption algorithms.
-Authentication and Authorization: Implement robust authentication mechanisms to verify the identity of users accessing the system and enforce role-based access control to restrict access to sensitive functionality based on user roles and permissions.
-Security Audits: Conduct regular security audits and vulnerability assessments to identify and mitigate potential security risks and vulnerabilities in the system's architecture, codebase, and configurations.
-Compliance: Ensure compliance with relevant data protection regulations (e.g., GDPR, CCPA) and industry standards (e.g., PCI DSS) to protect user data and ensure regulatory compliance.

2. Reliability and Availability:
- Uptime: The system should have high availability, with a target uptime of 99.9% or higher, to ensure users can access the platform reliably at all times.
- Fault Tolerance: Implement measures to ensure fault tolerance and resilience against system failures or disruptions, such as redundant servers, failover mechanisms, and automatic recovery processes.

3. Compatibility and Interoperability:
-Cross-Browser Compatibility: Ensure the web application is compatible with popular web browsers (e.g., Chrome, Firefox, Safari) across different devices and operating systems.
-Mobile Responsiveness: Design the application to be mobile-responsive, providing an optimal viewing and interaction experience on smartphones and tablets.
-API Compatibility: Provide well-documented APIs and ensure compatibility with third-party systems and devices, enabling seamless integration with external applications, services, and platforms.

4. Usability and Accessibility:
-User-Friendly Interface: Design the user interface to be intuitive, user-friendly, and easy to navigate, minimising the learning curve for users.
-Accessibility Standards: Adhere to accessibility standards (e.g., WCAG) to ensure the application is accessible to users with disabilities, providing support for screen readers, keyboard navigation, and alternative input methods.

5. Scalability and Performance:
-Horizontal Scalability: The system architecture should support horizontal scalability, allowing additional resources to be added dynamically to handle increasing load demands.
-Performance Monitoring: Implement performance monitoring and logging mechanisms to track system performance metrics (e.g., response times, throughput, error rates) and identify performance bottlenecks or issues.
-Documentation and Support:
-Comprehensive Documentation: Provide comprehensive documentation, including user manuals, API documentation, and system architecture documentation, to facilitate system understanding, usage, and maintenance.
-User Support: Offer user support channels (e.g., helpdesk, FAQs, community forums) to assist users with inquiries, issues, and troubleshooting.

Dependencies:

1. Payment Gateway Integration: FairShare relies on external payment gateway services to facilitate secure payment processing. Integration with payment gateway APIs (e.g., PayPal, Stripe) is necessary to handle financial transactions between users and ensure the seamless transfer of funds.

2. SMS Service Provider: The phone number authentication feature of FairShare requires integration with an external SMS service provider to send verification codes to users' mobile devices. Dependence on reliable SMS delivery services is crucial to ensure the effectiveness of the authentication process.

3. Cloud Infrastructure Provider: FairShare relies on cloud infrastructure services provided by platforms such as Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure to host and deploy the web application. Dependence on a robust and scalable cloud infrastructure is essential to ensure high availability, performance, and reliability of the FairShare platform.

4. Database Management System (DBMS): FairShare depends on a reliable database management system (e.g., MySQL, PostgreSQL, MongoDB) to store user data, transaction history, and account information securely. Dependence on a stable and scalable DBMS is crucial for efficient data management and retrieval within the FairShare application.

5. Payment Card Industry Data Security Standard (PCI DSS) Compliance: If handling credit card payments, FairShare relies on compliance with PCI DSS requirements to ensure the secure handling of cardholder data and prevent security breaches. Dependence on adherence to PCI DSS standards is necessary to maintain the integrity and security of financial transactions within the FairShare platform.

6. Third-Party APIs and Services: FairShare may rely on integration with third-party APIs and services to provide additional functionality or features, such as geolocation services for identifying user locations, email notification services for sending transaction alerts, or analytics services for monitoring user behaviour and system performance. Dependence on reliable third-party APIs and services is essential to enhance the functionality and usability of the FairShare platform.

12. Risk Analysis:
- Identification of potential risks and mitigation strategies.

13. Timeline:
- Project milestones and estimated completion dates.

14. Acceptance Criteria:
- Criteria that must be met for the product to be accepted.

15. Testing Requirements:
- Procedures and criteria for testing each feature.

16. Documentation:
- Information on user manuals, help guides, or other documentation.

17. Approval:
- Signature lines for stakeholders to indicate approval.

18. QR invitation

19. Phone number authentication

20. Payment search