CrowdChain Application: Use Case Scenarios

This document outlines detailed use case scenarios for the CrowdChain blockchain-based crowdfunding platform. Each use case includes actors, preconditions, main flow, alternative flows, postconditions, and special requirements.

Table of Contents

---

1. User Registration

Title

New User Registration

Actors

- Unregistered User
- System

Preconditions

- User has accessed the CrowdChain platform
- User does not have an existing account

Main Flow

1. User navigates to the registration page
2. User enters required information:

    - Username
    - Email address
    - Password
    - Confirm password
3. User accepts terms and conditions
4. User completes CAPTCHA verification
5. User submits registration form
6. System validates the input data
7. System creates a new user account
8. System generates a verification email with a unique link
9. System sends the verification email to the user's email address
10. User receives the email and clicks on the verification link
11. System verifies the user's email and activates the account
12. System redirects the user to the login page with a success message

Alternative Flows

- **A1: Invalid Input Data**
    1. System identifies invalid input data
    2. System displays appropriate error messages
    3. User corrects the input and resubmits
    4. Flow continues from step 6
- **A2: Email Already Registered**
    1. System identifies that the email is already registered
    2. System displays an error message
    3. User enters a different email or navigates to the login page
- **A3: Verification Email Not Received**
    1. User requests a new verification email
    2. System generates and sends a new verification email
    3. Flow continues from step 10

Postconditions

- User has a registered account on the CrowdChain platform
- User can log in using their credentials
- User has a basic profile set up

Special Requirements

- Password must meet security requirements (minimum 8 characters, including uppercase, lowercase, numbers, and special characters)
- Email verification must be completed within 24 hours
- User data must be stored securely in compliance with data protection regulations

---

2. User Login

Title

User Authentication

Actors

- Registered User
- System

Preconditions

- User has a registered and verified account
- User is logged out

Main Flow

1. User navigates to the login page
2. User enters their email/username and password
3. User clicks the "Login" button
4. System validates the credentials
5. System generates a session token
6. System redirects the user to the dashboard
7. System displays a welcome message

Alternative Flows

- **A1: Invalid Credentials**
  1. System identifies invalid credentials
  2. System displays an error message
  3. User re-enters credentials or requests password reset
- **A2: Forgotten Password**
  1. User clicks on "Forgot Password" link
  2. System prompts for user's email
  3. System sends a password reset link to the email
  4. User clicks the link and sets a new password
  5. User is redirected to the login page
  6. Flow continues from step 2
- **A3: Account Locked**
  1. System detects multiple failed login attempts
  2. System temporarily locks the account
  3. System displays a message about the account being locked
  4. User must wait for the lock period or contact support

Postconditions

- User is authenticated and logged into the system

- User has access to their account features
- User session is active

Special Requirements

- Session timeout after 30 minutes of inactivity
- Option for "Remember Me" functionality
- Support for two-factor authentication for enhanced security

---

3. Project Creation

Title

Creating a New Crowdfunding Project

Actors

- Project Creator (Registered User)
- System

Preconditions

- User is logged in
- User has completed their profile information
- User has a connected blockchain wallet

Main Flow

1. User navigates to "Create Project" section
2. User enters project details:
    - Project title
    - Project category
    - Funding goal amount
    - Project duration
    - Project description
    - Project images/videos
3. User defines project rewards/perks for backers (optional)
4. User agrees to platform terms for project creation
5. User submits the project for creation
6. System validates all required information
7. System creates the project in "Draft" status
8. System prompts user to define project milestones
9. System displays a success message with next steps

Alternative Flows

- **A1: Incomplete Information**
  1. System identifies missing required information
  2. System highlights missing fields
  3. User completes the required information
  4. Flow continues from step 5
- **A2: Invalid Funding Goal**
  1. System identifies that the funding goal doesn't meet minimum requirements
  2. System displays an error message
  3. User adjusts the funding goal
  4. Flow continues from step 5
- **A3: Save as Draft**
  1. User chooses to save project as draft without submitting
  2. System saves all entered information
  3. User can return later to complete and submit the project

Postconditions

- New project is created in "Draft" status
- Project is ready for milestone definition
- Project is associated with the creator's account
- Smart contract template is prepared for deployment

Special Requirements

- Minimum funding goal of 0.1 ETH (or equivalent)
- Maximum project duration of 90 days
- Project description must be at least 200 characters
- At least one project image is required
- Project must comply with platform guidelines and legal requirements

---

4. Milestone Definition

Title

Defining Project Milestones

Actors

- Project Creator
- System

Preconditions

- User is logged in

- User has created a project in "Draft" status
- Project basic details are complete

Main Flow

1. User navigates to the milestone definition section of their draft project
2. System displays guidelines for creating effective milestones
3. User adds first milestone:
    - Milestone title
    - Description
    - Deliverables
    - Completion criteria
    - Funding percentage allocation
    - Estimated completion date
4. System validates milestone information
5. User adds additional milestones (repeating steps 3-4)
6. System ensures total funding allocation across milestones equals 100%
7. User reviews all milestones
8. User confirms milestone plan
9. System saves the milestone information
10. System updates project status to "Ready for Review"
11. System notifies platform administrators for project review

Alternative Flows

- **A1: Invalid Milestone Information**
    1. System identifies issues with milestone information
    2. System displays specific error messages
    3. User corrects the information
    4. Flow continues from step 4
- **A2: Funding Allocation Mismatch**
    1. System identifies that total funding allocation doesn't equal 100%
    2. System displays an error message with the current total
    3. User adjusts milestone funding percentages
    4. Flow continues from step 6
- **A3: Save Progress Without Completing**
    1. User saves current milestone information without finalizing
    2. System saves the partial milestone plan
    3. User can return later to complete milestone definition

Postconditions

- Project has a complete milestone plan
- Each milestone has defined criteria and funding allocation
- Project status is updated to "Ready for Review"
- Project is queued for administrative review

Special Requirements

- Minimum of 2 milestones required
- Maximum of 10 milestones allowed
- First milestone cannot exceed 30% of total funding
- Last milestone must be at least 10% of total funding
- Milestone descriptions must be at least 100 characters
- Clear completion criteria must be defined for each milestone

---

5. Project Browsing

Title

Browsing Available Projects

Actors

- User (Registered or Unregistered)
- System

Preconditions

- User has accessed the CrowdChain platform

Main Flow

1. User navigates to the project browsing page
2. System displays featured and trending projects
3. System shows project categories
4. User browses projects by scrolling through the list
5. User views project cards showing:
    - Project title
    - Brief description
    - Funding progress
    - Time remaining
    - Project image
6. User clicks on a project card
7. System redirects to the detailed project page
8. System displays comprehensive project information:
    - Full description
    - Creator profile
    - Funding details
    - Milestone plan
    - Updates
    - Backer comments

Alternative Flows

- **A1: No Projects Available**
  1. System displays a message indicating no projects are available
  2. System suggests creating a new project or checking back later
- **A2: Filter by Category**
  1. User selects a specific category
  2. System filters projects to show only those in the selected category
  3. Flow continues from step 4
- **A3: Sort Projects**
  1. User selects a sorting option (newest, most funded, ending soon)
  2. System reorders the project list according to the selected criteria
  3. Flow continues from step 4

Postconditions

- User has viewed available projects
- User has accessed detailed information about specific projects of interest

Special Requirements

- Projects should load with minimal delay
- Project cards should display real-time funding progress
- Featured projects should be rotated regularly
- Projects with approaching deadlines should be highlighted

---

6. Project Search and Filtering

Title

Searching and Filtering Projects

Actors

- User (Registered or Unregistered)
- System

Preconditions

- User has accessed the CrowdChain platform
- Projects exist in the system

Main Flow

1. User navigates to the project search page
2. User enters search keywords in the search bar
3. User applies filters:

- Category
- Funding status (% funded)
- Time remaining
- Milestone structure
- Location (if applicable)

4. User selects sorting criteria:
   - Newest first
   - Most funded
   - Ending soon
   - Most backers
5. System processes the search query and filters
6. System displays matching projects
7. User browses through search results
8. User clicks on a project of interest
9. System displays the detailed project page

Alternative Flows

- **A1: No Matching Projects**
  1. System displays a message indicating no projects match the criteria
  2. System suggests modifying search terms or filters
  3. User adjusts search parameters
  4. Flow continues from step 5
- **A2: Advanced Search**
  1. User clicks on "Advanced Search" option
  2. System displays additional search parameters
  3. User specifies detailed criteria
  4. Flow continues from step 5
- **A3: Save Search**
  1. Registered user saves search criteria for future use
  2. System stores the search parameters in the user's account
  3. User can access saved searches from their dashboard

Postconditions

- User has found projects matching their criteria
- Search results are displayed according to user preferences
- User can access detailed information about projects of interest

Special Requirements

- Search should support partial keyword matching
- Search results should update dynamically as filters are applied
- Recent searches should be saved for registered users
- Search functionality should be optimized for performance

---

7. Project Backing

Title

Backing a Crowdfunding Project

Actors

- Backer (Registered User)
- System
- Blockchain Network

Preconditions

- User is logged in
- User has a connected blockchain wallet with sufficient funds
- Project is in "Active" status

Main Flow

1. User navigates to a project page
2. User reviews project details and milestone plan
3. User clicks "Back This Project" button
4. System displays backing options:
    - Contribution amount input
    - Reward tier selection (if applicable)
5. User enters contribution amount
6. User selects a reward tier (optional)
7. User clicks "Confirm Backing" button
8. System displays transaction summary and terms
9. User confirms the transaction
10. System initiates blockchain transaction
11. Blockchain network processes the transaction
12. System receives transaction confirmation
13. System updates project funding status
14. System adds user to project backers list
15. System sends confirmation to user
16. System notifies project creator

Alternative Flows

- **A1: Insufficient Funds**
    1. System or blockchain network identifies insufficient funds
    2. System displays an error message
    3. User adds funds to wallet or adjusts contribution amount
    4. Flow continues from step 5
- **A2: Transaction Failure**
    1. Blockchain transaction fails

2. System displays error message with reason
3. User can retry the transaction
4. Flow continues from step 9

- **A3: Minimum Contribution Not Met**
  1. System identifies that contribution is below minimum requirement
  2. System displays an error message
  3. User adjusts contribution amount
  4. Flow continues from step 5

Postconditions

- User's contribution is recorded on the blockchain
- Project funding progress is updated
- User is added to the project's backer list
- User is eligible for selected rewards
- User gains voting rights for milestone approvals
- Smart contract holds the contributed funds in escrow

Special Requirements

- Minimum contribution amount of 0.01 ETH (or equivalent)
- Transaction gas fees clearly displayed before confirmation
- Real-time exchange rate information for cryptocurrency
- Clear explanation of milestone-based disbursement
- Cooling-off period of 24 hours for large contributions (optional)

---

8. Milestone Completion Submission

Title

Submitting Milestone Completion Evidence

Actors

- Project Creator
- System

Preconditions

- User is logged in as the project creator
- Project is successfully funded
- Project has active milestones
- Current milestone is in progress

Main Flow

1. Project creator navigates to project management dashboard
2. Creator selects the active milestone
3. Creator clicks "Submit for Completion" button
4. System displays milestone submission form
5. Creator provides completion evidence:
    - Detailed description of completed work
    - Images/videos of deliverables
    - Links to external resources/repositories
    - Any additional documentation
6. Creator certifies that all milestone criteria have been met
7. Creator submits the milestone completion request
8. System validates submission completeness
9. System changes milestone status to "Pending Approval"
10. System notifies all project backers about the submission
11. System initiates the voting period for milestone approval

Alternative Flows

- **A1: Incomplete Submission**
    1. System identifies missing required information
    2. System displays error messages
    3. Creator completes the missing information
    4. Flow continues from step 7
- **A2: Save Draft Submission**
    1. Creator saves submission as draft without submitting
    2. System saves the draft
    3. Creator can return later to complete and submit
- **A3: Milestone Deadline Extension Request**
    1. Creator identifies need for deadline extension
    2. Creator includes extension request with justification
    3. System flags the submission as including extension request
    4. Backers vote on both completion and extension

Postconditions

- Milestone completion evidence is submitted
- Milestone status is updated to "Pending Approval"
- Voting period for backers is initiated
- Project backers are notified about the submission
- Submission timestamp is recorded

Special Requirements

- Submission must include at least 3 pieces of evidence
- Submission description must be at least 200 characters

- All submitted evidence must be accessible to backers
- Submission must address all criteria defined in the milestone
- System must maintain immutable record of all submissions

---

9. Milestone Voting

Title

Voting on Milestone Completion

Actors

- Project Backers
- System
- Blockchain Network

Preconditions

- User is logged in as a project backer
- User has contributed to the project
- Milestone completion evidence has been submitted
- Voting period is active

Main Flow

1. Backer receives notification about milestone submission
2. Backer navigates to the project page
3. System displays milestone submission details and evidence
4. Backer reviews the submission materials
5. Backer selects voting option:
    - Approve milestone completion
    - Reject milestone completion (with reason)
    - Abstain from voting
6. Backer submits their vote
7. System records the vote on the blockchain
8. System updates voting statistics
9. System displays confirmation of recorded vote
10. System continues collecting votes until voting period ends
11. At the end of voting period, system tallies final results
12. System determines outcome based on voting threshold
13. System updates milestone status based on voting outcome

Alternative Flows

- **A1: Request Additional Information**
    1. Backer requests additional information before voting

2. System notifies project creator
3. Creator provides additional information
4. Backer receives notification and reviews new information
5. Flow continues from step 5

- **A2: Change Vote**
  1. Backer decides to change vote during voting period
  2. Backer navigates to voting page
  3. System allows vote change if voting period is still active
  4. Backer submits new vote
  5. System updates vote record
  6. Flow continues from step 8
- **A3: Voting Deadline Extension**
  1. System identifies low voter participation
  2. System extends voting period by 48 hours
  3. System notifies all backers about extension
  4. Flow continues from step 10 after extended period

Postconditions

- Backer's vote is recorded on the blockchain
- Voting statistics are updated
- After voting period, milestone status is updated to "Approved" or "Rejected"
- If approved, fund disbursement process is triggered
- If rejected, project creator can address issues and resubmit

Special Requirements

- Voting power proportional to contribution amount
- Minimum voting threshold of 51% of total backing value
- Voting period of 7 days
- Immutable voting record on blockchain
- Anonymous voting with transparent results
- Clear explanation of voting impact on fund disbursement

---

10. Fund Disbursement

Title

Milestone-Based Fund Disbursement

Actors

- System
- Blockchain Network
- Project Creator

- Project Backers (passive)

Preconditions

- Milestone has been approved by backers
- Voting period has ended with positive outcome
- Smart contract holds sufficient funds

Main Flow

1. System identifies milestone approval has reached threshold
2. System initiates fund disbursement process
3. System calculates disbursement amount based on milestone percentage
4. System triggers smart contract function for disbursement
5. Blockchain network processes the transaction
6. Smart contract transfers funds to project creator's wallet
7. Blockchain network confirms transaction completion
8. System records disbursement details
9. System updates milestone status to "Completed"
10. System updates project funding status
11. System notifies project creator about successful disbursement
12. System notifies backers about disbursement
13. If more milestones exist, system activates next milestone

Alternative Flows

- **A1: Transaction Failure**
  1. Blockchain transaction fails
  2. System records failure reason
  3. System retries transaction up to 3 times
  4. If still failing, system alerts administrators
  5. Issue is resolved manually
  6. Flow continues from step 4
- **A2: Insufficient Gas**
  1. System identifies insufficient gas for transaction
  2. System allocates additional gas from platform reserve
  3. Flow continues from step 5
- **A3: Final Milestone Completion**
  1. System identifies this is the final milestone
  2. System updates project status to "Completed"
  3. System initiates project completion process
  4. System collects final feedback from backers and creator

Postconditions

- Funds for the milestone are transferred to project creator
- Milestone status is updated to "Completed"

- Project funding status is updated
- Transaction details are recorded
- Next milestone is activated (if applicable)
- Project status is updated to "Completed" if final milestone

Special Requirements

- Disbursement must occur within 24 hours of approval
- Transaction fees covered by platform or deducted from disbursement
- Complete transaction record maintained for audit purposes
- Automatic handling of currency conversion if necessary
- Compliance with relevant financial regulations
- Failsafe mechanisms for transaction issues

---

11. Project Updates

Title

Posting Project Updates

Actors

- Project Creator
- System
- Project Backers (passive)

Preconditions

- User is logged in as project creator
- Project is in active status (funding or development phase)

Main Flow

1. Creator navigates to project management dashboard
2. Creator selects "Post Update" option
3. System displays update creation form
4. Creator enters update details:
    - Update title
    - Update content
    - Images/videos (optional)
    - Update category (general, milestone progress, announcement)
    - Visibility (public or backers-only)
5. Creator previews the update
6. Creator submits the update
7. System validates update content
8. System posts the update to the project page

9. System records update timestamp
10. System notifies all project backers about the new update
11. System adds update to project timeline

Alternative Flows

- **A1: Save as Draft**
  1. Creator saves update as draft without publishing
  2. System saves draft update
  3. Creator can return later to edit and publish
- **A2: Edit Published Update**
  1. Creator selects existing update to edit
  2. System displays update in editable form
  3. Creator makes changes
  4. Creator submits edited update
  5. System updates the post with edit timestamp
  6. System notifies backers about the edit (optional)
- **A3: Scheduled Update**
  1. Creator sets future publication date/time
  2. System saves update for scheduled release
  3. System automatically publishes at scheduled time
  4. Flow continues from step 9

Postconditions

- Update is published on the project page
- Project backers are notified
- Update is added to project timeline
- Update engagement metrics begin tracking

Special Requirements

- Rich text formatting support
- Image and video embedding capabilities
- Update history maintained for transparency
- Minimum update frequency requirement (e.g., at least once per month)
- Character limit of 5,000 for update content
- Support for milestone-specific updates

---

12. User Profile Management

Title

Managing User Profile and Settings

Actors

- Registered User
- System

Preconditions

- User is logged in

Main Flow

1. User navigates to profile settings
2. System displays current profile information
3. User selects information to update:
    - Profile picture
    - Display name
    - Bio/About me
    - Social media links
    - Skills/Expertise
    - Location
4. User makes desired changes
5. User saves profile updates
6. System validates the information
7. System updates the user profile
8. System displays confirmation message
9. User navigates to account settings
10. User updates account preferences:
    - Email notification preferences
    - Privacy settings
    - Wallet connections
    - Two-factor authentication
11. User saves account settings
12. System updates user preferences

Alternative Flows

- **A1: Invalid Information**
    1. System identifies invalid information
    2. System displays error messages
    3. User corrects the information
    4. Flow continues from step 5
- **A2: Change Email Address**
    1. User updates email address
    2. System sends verification email to new address
    3. User verifies new email address
    4. System updates email address in user account
    5. Flow continues from step 8

- **A3: Change Password**
  1. User selects “Change Password” option
  2. System prompts for current password
  3. User enters current password and new password
  4. System validates passwords
  5. System updates password
  6. System sends notification about password change
  7. Flow continues from step 8

Postconditions

- User profile information is updated
- User account settings are updated
- Changes are visible to other users (where applicable)
- User preferences are applied to system interactions

Special Requirements

- Profile picture size and format restrictions
- Bio character limit of 500 characters
- Password strength requirements
- Secure handling of sensitive information
- Option to make certain profile information private
- Blockchain wallet connection verification

---

13. Project Comment and Discussion

Title

Commenting and Discussing Project Details

Actors

- Registered User (Creator or Backer)
- System
- Other Users (passive)

Preconditions

- User is logged in
- User is viewing a project page

Main Flow

1. User navigates to the comments section of a project
2. System displays existing comments in chronological or threaded view

3. User enters a comment in the comment box
4. User submits the comment
5. System validates comment content
6. System posts the comment to the project page
7. System records comment timestamp and author
8. System notifies project creator about the new comment
9. System displays the comment in the comments section
10. Other users can view the comment
11. Project creator or other users can reply to the comment

Alternative Flows

- **A1: Comment Moderation**
  1. System flags comment for potential violation of community guidelines
  2. System holds comment for moderation
  3. Moderator reviews the comment
  4. If approved, flow continues from step 6
  5. If rejected, user is notified about rejection reason
- **A2: Edit Comment**
  1. User selects their existing comment
  2. User clicks "Edit" option
  3. System displays comment in editable form
  4. User modifies comment
  5. User submits edited comment
  6. System updates comment with edit timestamp
  7. Flow continues from step 9
- **A3: Delete Comment**
  1. User selects their existing comment
  2. User clicks "Delete" option
  3. System prompts for confirmation
  4. User confirms deletion
  5. System removes comment from public view
  6. System marks comment as deleted in database

Postconditions

- Comment is posted to the project page
- Comment is visible to other users
- Project creator is notified
- Discussion thread is updated

Special Requirements

- Character limit of 1,000 for comments
- Support for basic formatting (bold, italic, links)
- Threaded replies up to 3 levels deep
- Ability to mention other users with @ symbol

- Ability to include images in comments
- Comment editing allowed within 24 hours of posting
- Community guidelines enforcement

---

14. Administrative Project Review

Title

Administrative Review of Submitted Projects

Actors

- Administrator
- System
- Project Creator (passive)

Preconditions

- Administrator is logged in with appropriate permissions
- Projects are in "Ready for Review" status

Main Flow

1. Administrator navigates to project review dashboard
2. System displays list of projects pending review
3. Administrator selects a project to review
4. System displays comprehensive project details:
    - Project information
    - Creator profile and history
    - Milestone plan
    - Funding goals
    - Reward structure
5. Administrator reviews project against platform guidelines
6. Administrator checks for:
    - Completeness of information
    - Feasibility of milestones
    - Compliance with terms of service
    - Potential red flags
7. Administrator makes decision:
    - Approve project
    - Request changes
    - Reject project
8. Administrator enters decision notes
9. Administrator submits review decision
10. System updates project status based on decision
11. System notifies project creator about the decision

12. If approved, system deploys project smart contract
13. If approved, system changes project status to "Active"

Alternative Flows

- **A1: Request Changes**
  1. Administrator identifies specific issues requiring changes
  2. Administrator details required changes in notes
  3. System updates project status to "Changes Requested"
  4. System notifies creator with specific change requests
  5. Creator makes requested changes
  6. Creator resubmits project for review
  7. Flow restarts from step 1
- **A2: Reject Project**
  1. Administrator identifies serious issues with project
  2. Administrator details rejection reasons
  3. System updates project status to "Rejected"
  4. System notifies creator with rejection reasons
  5. Creator can appeal decision or create new project
- **A3: Escalate Review**
  1. Administrator identifies complex issues requiring additional review
  2. Administrator escalates to senior review team
  3. System updates project status to "Under Extended Review"
  4. Senior review team evaluates project
  5. Flow continues from step 7 with senior team decision

Postconditions

- Project status is updated based on review decision
- Project creator is notified of the decision
- If approved, project is published and smart contract deployed
- If changes requested, project awaits creator modifications
- If rejected, project is removed from active consideration
- Review decision and notes are recorded in system

Special Requirements

- Review must be completed within 48 hours of submission
- Clear documentation of review criteria
- Consistent application of platform guidelines
- Audit trail of all review decisions
- Multiple administrator review for high-value projects
- Legal compliance verification

---

15. Dispute Resolution

Title

Resolving Disputes Between Creators and Backers

Actors

- Project Backer
- Project Creator
- Administrator
- System

Preconditions

- User is logged in as a project backer
- Milestone is in "Pending Approval" or "Rejected" status
- Disagreement exists about milestone completion

Main Flow

1. Backer navigates to project page
2. Backer selects "Raise Dispute" option
3. System displays dispute form
4. Backer enters dispute details:
    - Nature of dispute
    - Specific concerns
    - Desired resolution
    - Supporting evidence
5. Backer submits the dispute
6. System records dispute and assigns case number
7. System notifies project creator about the dispute
8. System notifies administrators about new dispute
9. Administrator reviews dispute details
10. Administrator facilitates communication between parties
11. Parties discuss potential resolution
12. If agreement reached:
    - Administrator records resolution terms
    - Parties confirm acceptance
    - System implements agreed resolution
13. If no agreement:
    - Administrator makes binding decision
    - System implements administrator decision
14. System updates milestone and project status accordingly
15. System notifies all parties about resolution outcome

Alternative Flows

- **A1: Creator Initiates Dispute**
  1. Creator initiates dispute about rejected milestone
  2. System records dispute from creator perspective
  3. Flow continues from step 6
- **A2: Mediation Process**
  1. Administrator determines mediation is appropriate
  2. System assigns case to specialized mediator
  3. Mediator conducts structured mediation process
  4. If resolution reached, flow continues from step 12
  5. If no resolution, flow continues from step 13
- **A3: Escalation to Arbitration**
  1. Dispute requires formal arbitration
  2. System initiates arbitration process
  3. Arbitrator reviews case and evidence
  4. Arbitrator conducts hearing if necessary
  5. Arbitrator issues binding decision
  6. Flow continues from step 14

Postconditions

- Dispute is resolved with clear outcome
- Milestone status is updated based on resolution
- Fund disbursement occurs according to resolution
- Resolution details are recorded in system
- Project timeline is updated to reflect resolution

Special Requirements

- Dispute resolution process limited to 14 days
- Clear documentation of all communication
- Immutable record of evidence and decisions
- Fair and transparent process for all parties
- Option for emergency intervention for serious issues
- Compliance with relevant arbitration regulations

---

16. Project Analytics Dashboard

Title

Accessing Project Performance Analytics

Actors

- Project Creator

- System

Preconditions

- User is logged in as project creator
- Project is in active or completed status

Main Flow

1. Creator navigates to project management dashboard
2. Creator selects "Analytics" section
3. System retrieves project performance data
4. System displays analytics dashboard with:
    - Funding progress over time
    - Backer demographics
    - Traffic sources
    - Conversion rates
    - Milestone completion statistics
    - Engagement metrics
5. Creator reviews overall performance metrics
6. Creator selects specific time period for detailed analysis
7. System updates dashboard with selected period data
8. Creator exports analytics data (optional)
9. Creator uses insights to inform project updates or strategies

Alternative Flows

- **A1: Comparative Analysis**
    1. Creator selects "Compare" option
    2. Creator selects metrics to compare across time periods
    3. System generates comparative visualization
    4. Creator reviews comparative data
- **A2: Custom Report Generation**
    1. Creator selects "Custom Report" option
    2. Creator selects specific metrics to include
    3. Creator configures report parameters
    4. System generates custom report
    5. Creator downloads or shares report
- **A3: Predictive Analytics**
    1. Creator selects "Projections" option
    2. System analyzes current trends
    3. System generates funding and milestone projections
    4. Creator reviews predictive insights

Postconditions

- Creator has accessed project performance data

- Creator has insights into project metrics
- Creator can make informed decisions based on analytics
- Analytics data is recorded for future reference

Special Requirements

- Real-time data updates
- Interactive visualizations
- Data export in multiple formats
- Privacy protection for backer data
- Historical data retention
- Benchmark comparisons with similar projects
- Mobile-friendly dashboard

---

17. Wallet Integration

Title

Connecting and Managing Blockchain Wallets

Actors

- Registered User
- System
- Blockchain Network

Preconditions

- User is logged in
- User has a compatible blockchain wallet

Main Flow

1. User navigates to account settings
2. User selects "Wallet Management" section
3. System displays current wallet status
4. User clicks "Connect Wallet" button
5. System displays supported wallet options
6. User selects wallet provider (MetaMask, WalletConnect, etc.)
7. System initiates connection request
8. Wallet provider prompts user for connection approval
9. User approves connection in wallet interface
10. Blockchain network verifies connection
11. System receives wallet address and verification
12. System associates wallet with user account
13. System displays connected wallet information

14. User can set default wallet for transactions

Alternative Flows

- **A1: Connection Failure**
  1. Wallet connection fails
  2. System displays error message with reason
  3. System provides troubleshooting steps
  4. User resolves issue and retries
  5. Flow continues from step 7
- **A2: Disconnect Wallet**
  1. User selects "Disconnect" for an existing wallet
  2. System prompts for confirmation
  3. User confirms disconnection
  4. System removes wallet association
  5. System updates wallet status
- **A3: Multiple Wallet Management**
  1. User has multiple wallets connected
  2. User sets primary wallet for transactions
  3. System updates wallet preferences
  4. System uses primary wallet for future transactions

Postconditions

- User's blockchain wallet is connected to their account
- Wallet address is verified and stored securely
- User can use wallet for platform transactions
- Wallet status is displayed in user account

Special Requirements

- Support for multiple blockchain networks
- Secure wallet connection protocol
- No storage of private keys or seed phrases
- Clear transaction signing process
- Gas fee estimation for transactions
- Wallet balance display (optional)
- Transaction history integration