Building motivation one task at a time

High Level Requirements

Group 2:

Xavier Davis, Nabeel Hussain, Fleury Keigni Di Satchou, Shawn Sokoloski

University of Maryland University College

CMSC 495

5/4/2018

Version # 2.0

Document Version Control

Ver	Date	Description
1.0	3/25/2018	Initial requirements documentation with user specifications and use cases
2.0	5/4/2018	Final Updates

Table of Contents

Section 1: Introduction

An application that allows a user to create tasks, and reward payouts upon completion of those tasks. Payout functionality will be executed by use of the Stellar Lumens blockchain technology. Upon completion of the task the original proposer of the task must confirm the completion of the task to initiate payout. Email notifications are used to provide updates to users on task completion statuses.

1.1 Background

A part of a capstone course at the University of Maryland University College, a course requirement was to coordinate in a group environment to develop an application. The nature and concept of the application is to be decided by the group. Project responsibilities are to be divided among the group members to deliver a working product at the end of an eight-week course.

1.2 Purpose

This document is to detail high level requirements and functionality of a task rewards application. Various use cases are included to detail the flow of information between different system components. Detailed information about what the system can do and how the system performs. Project group roles and responsibilities are detailed along with scheduled timelines for the project.

1.3 Project Contacts

Name	Role	Email
Xavier Davis	Project Manager/Software Engineer	x.davis1993@gmail.com
Nabeel Hussain	Project Manager/Integration Engineer	nabeelh21@gmail.com
Shawn Sokoloski	Technical Writer	sasokoloski@gmail.com
Fleury Keigni Di Satchou	Test Engineer	keignif100@gmail.com

1.4 Scope

In Scope

- The project will contain the internet application PinUp.
- The project will use Amazon Web Services to host its source and databases.
- The project will allow the creation and management of user content.
- The project will have a unique and fully functional user interface.
- The project will have adequate documentation and comprehensive user manual.

Out of Scope

- The project will not contain functionality for mobile devices, tablets, or iOS.
- The project will not suggest financial institutions.
- The project will not be deployed into an application market (such as Google Play).
- The project will not contain advertisements or marketing.
- The project will not make use of application notifications. All notifying functionality will employ emails to users.

1.5 Acceptance Criteria Factors

- Must be delivered on time
- Ease of use
- Should look visually pleasing
- Log in/Sign up process should be seamless
- Dashboard features should not contain any errors

1.6 Documentation Organization

This document comprises of the following sections.

Section 1 – Introduction – provides the background, purpose and scope for this document.

Section 2 – Use Cases – provides the high-level use cases, including the actors, for the solution.

Section 3 – System Requirements – provides the high-level functional and supplemental requirements for the solution.

Section 4 – System Specifications – provides the Hardware and software base for the application

Section 5 – Design mockups – provides preliminary mockups for the dashboard layout of the application

Appendix A - Key Decisions Log

Section 2: Use Cases

2.1 Registering User

This use case details the sequence of events a user will encounter during the registration process. The process can also allow a customer to sponsor another customer to link the account for future use.

Actors

- Customer

Flow of Events

1. Registration begins when the “Sign-Up” link is clicked in home page of the application.
    a. All users 13 and older are required to provide email and can sponsor self.
2. The application will display a pop-up menu asking the user for basic information
3. Required input fields will be properly marked.
4. Once form is submitted an email notification will be sent to the user confirming their account creation.
5. Stellar Lumens wallets will also be generated for future transaction use.

6. Concluding the registration process, the user will be redirected to their profile page which will list their basic information and Lumens wallet balance.

Potential Errors

- Duplicate username
- Weak password
- Missing form information

2.2 Link Sub-Accounts

Use case demonstrates the steps needed to link sub accounts to the customers main account.

Actors

- Customer

Flow of Events

1. Clicking the “Add Partnership” button on user dashboard begins the process.
2. The Partnership pop-up form will display.
3. Required input fields will be properly marked on the form.
4. If requested sub-account is a previously established account on system, then:
5. The sub-account’s username
6. Else if the sub-user is 13 years or older without a current system account, then:
    a. Information required includes name and email.
    b. Submission of the form sends an email notification to sub-user.
    c. Sub-user follows flow of events for user registration.
7. Else if the sub-user is younger than 13 years old, then:
    a. The current user becomes the sponsor for the sub-user.
    b. The sponsor provides all the basic information for the sub-user.
    c. No email notifications are activated.
    d. The sponsor retains all admin functions for the sub-user.
8. Sub-account is linked to user.
9. Partnership is displayed on user’s dashboard.

2.3 Log In

Outlines the event process for a user to log into the web application

Actors

- Customer

Flow of Events

1. On application home page a login button is presented.

2. The application displays a login form on screen.
3. The user enters a username and password combination.
4. The system will verify the information.
5. The system will set user permissions based on user account type.
6. The application will display the user's personal dashboard.

Potential Errors

- Incorrect username
- Incorrect password

2.4 Creating a Task Post

This use case outlines the actions need to successfully create a new task post in the roles of a sponsor or sponsored account.

Actors

- Customer (sponsor and/or sponsored)

Flow of Events

1. On each user dashboard, there will be a button to post new tasks for completion.
2. A web form will be presented.
3. Information about the proposed task (i.e. title, description, and payout) will be needed.
4. The user will then verify all information entered on the form is correct.
5. If form is accepted by user, then:
    a. The task will be added to the dashboard for monitoring.
6. If form information is rejected, then:
    a. User is redirected back to form edit for corrections.
7. After form adjustment, step 5 is revisited.
8. Sponsored accounts task proposals follow the same flow of event listed above, but upon final submission of the task a sponsor review is required for final approval.
9. An email notification will be sent out the sponsor account alerting them of the recent proposal.
10. In the email will contain the form information about the proposed task and a link navigating the sponsor back to their dashboard to review the proposed task.
11. During the review process the sponsor can edit the task as needed and insert a payout amount for task completion.
12. If the proposal is accepted, then:
    a. The task is added to the sponsors dashboard like any other self-authored task.
13. With approval the sponsored user can proceed with task completion for posted payout.
14. If the proposal is rejected, then:

a. The sponsored user is notified.

2.5 Complete Payout Process

Outlines event chain for initiating and authorizing the payout for completed posted tasks.

Actors

- Customer

Flow of Events

1) On the employed user's dashboard, a list of all open jobs are displayed.
2) The user clicked the complete button to initiate the payout process.
3) A pop-up window displays confirming the initiation of the payout process.
4) If accepted by the employed user, the employer of the task is notified by email.
5) The employer of the task logs into their dashboard to complete payout.
6) If the employer is satisfied with the completion status, then:
   a) They click the payout button.
   b) A confirmation message is displayed with final details of the payout.
   c) The employer has one last time to approve the Lumens payout.
   d) The amount cannot be changed at this point.
   e) Final confirmation submits the payment to the employed user and the job is archived from the dashboard.
7) If completion status is rejected, the task is still archived from the dashboard, and no payout is initiated.

Potential Errors

- Lack of appropriate funds

Section 3: System Requirements

3.1 Major System Conditions

a) The system must be available on the internet
b) The system must be available 24 hours per day
c) The system must be accessible by computer browser
d) The system must not allow duplicate accounts

3.2 Major System Capabilities

a) The system must allow users to create and login to accounts
b) The system must be able to communicate with users via email

c) The system must allow users to generate 'sub' accounts for subcontractors fulfilling posted tasks
d) The system must allow tasks to be generated and assigned a name, expiration, notes, and a monetary payout value
e) The system must track task progress and completion
f) The system must track expiration on timed tasks, and close them upon expiration
g) The system must archive completed tasks
h) The system must allow users to see tasks which are available, in progress, completed, and closed
i) The system must allow a user to reactivate a completed or closed task
j) The system must allow users to edit task names, notes, expirations, and payout amounts
k) The system must ensure the sponsor's account has enough funds to cover the listed payout amounts of all posted tasks

3.3 Major System Interfaces

a) The system must work with the Stellar Horizon API to access Stellar Core
b) The system must utilize BlockChain utilities to process payments

3.4 System User Characteristics

a) The primary user (sponsor) will have the ability to create unlimited accounts for sub users
b) The sponsor must supply valid email addresses for themselves as well as each sub user in their group
c) The sponsor must be able to validate the completion of each task
d) The sponsor must supply valid institution for supplying funds
e) The sponsor must be able to supply adequate funds to pay out the amount offered on each task posted
f) Sub users must supply valid systems from which to obtain payout funds
g) Sub users must be able to complete tasks and interact with system to indicate tasks in progress and tasks pending completion verification

3.5 Functional Requirements

An actor must have access to an established internet connection with accessibility to the world wide web domain. Using a laptop or desktop hardware, system interactions are possible. Once access to the application is made, the actor can perform various actions on the system. Of which are detailed below.

The frontend of the application provides the user with the interface to navigate and interact with the system. Various task can be executed on the part of the user (i.e. registration, profile initiation, manage wallet transactions). Certain frontend activities trigger additional backend functions to later support user future actions. At the completion of registration and self-containing systems are initiated to generate the public and private keys needed to complete future wallet transactions.

Business Functional Req. ID	Requirement Definition	Priority
1	Adding of new customer to the system. Gathering the needed information to populate into dependency systems.	High
2	Provide a template user profile as a welcome screen for all returning users.	High
3	The ability to create and monitor the process of new tasks within the user's dashboard display.	High
4	Incorporate a digital asset to effortlessly transfer between users.	High
5	Notification system to communicate user activity.	Medium

Section 4: System Specifications

Programming Languages	Dependency Libraries			Cloud Hosting Services
HTML CSS JavaScript	Node.js Request jQuery	Express Passport Bootstrap 4	Mongoose Body-parser npm	Amazon Web Services MongoDB Atlas

Appendix A. Key Decision Log

A.1 Key Decision: Adding sub accounts

Key Area	Decision Detail
Issue Description	When to allow the owner to make sub accounts
Decision	Instead of allowing them to make sub accounts during the registration process, we agreed that they can add sub accounts once they have logged into their account dashboard.
Business / Decision Owner	Xavier Davis, Nabeel Hussain, Shawn Sokoloski, and Fleury Keigni Di Satchou
Date	3/23/2018
Key Decision Author	
Full Description of Issue	
Evaluation Criteria	
Alternative solutions and evaluation results	