



Building motivation one task at a time

Project Design

Group 2:

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

University of Maryland, University College

CMSC 495

5/04/2018

Version # 2.0

Revision Sheet

Release No.	Date	Revision Description
Rev. 0	04/03/2018	Downloaded template and checklist
Rev. 1	04/04/2018	Initial rough draft. Contains logical view (high-level modules only) and development view.
Rev. 2	5/4/2018	Final Updates

Table of Contents

<u>Section 1: Introduction</u>	<u>3</u>
<u>Section 2: Design Goals and Schedule</u>	<u>3</u>
<u>Section 3: System Behavior</u>	<u>3</u>
<u>Section 4: Logical View</u>	<u>5</u>
<u>4.1. High-Level Design (Architecture)</u>	<u>5</u>
<u>4.2. Mid-Level Design</u>	<u>6</u>
<u>4.3. Detailed Class Design</u>	<u>7</u>
<u>Section 5: Physical View</u>	<u>9</u>
<u>Section 6: Use Case View</u>	<u>10</u>
<u>6.1. Registering User</u>	<u>10</u>
<u>6.2. Link Sub-Accounts</u>	<u>10</u>
<u>6.3. Log-In</u>	<u>11</u>
<u>6.4. Creating a Task Post</u>	<u>11</u>
<u>6.5. Complete Payout Process</u>	<u>12</u>
<u>Appendix A. Acronyms and Abbreviations</u>	<u>13</u>

1 Introduction

PinUp approaches the need of a small task and compensation management application through its functionality. Users, being business owners, parents, or any person who requires the completion of small tasks, create accounts with PinUp. Accounts for subcontractors, children, or otherwise individuals performing the required tasks can then be created and maintained by the primary, sponsoring user. Tasks can be posted for all users in the group to see, complete with notes, expiration times, and payout amounts for their completion. Sub-users choose which task to complete, and all group users can see that the task has been initiated. Upon completion, the primary group user can verify its completion, and the payout amount will be transferred from the primary account to the Lumen Wallet of the sub-user who completed the task, via the Stellar Horizon API.

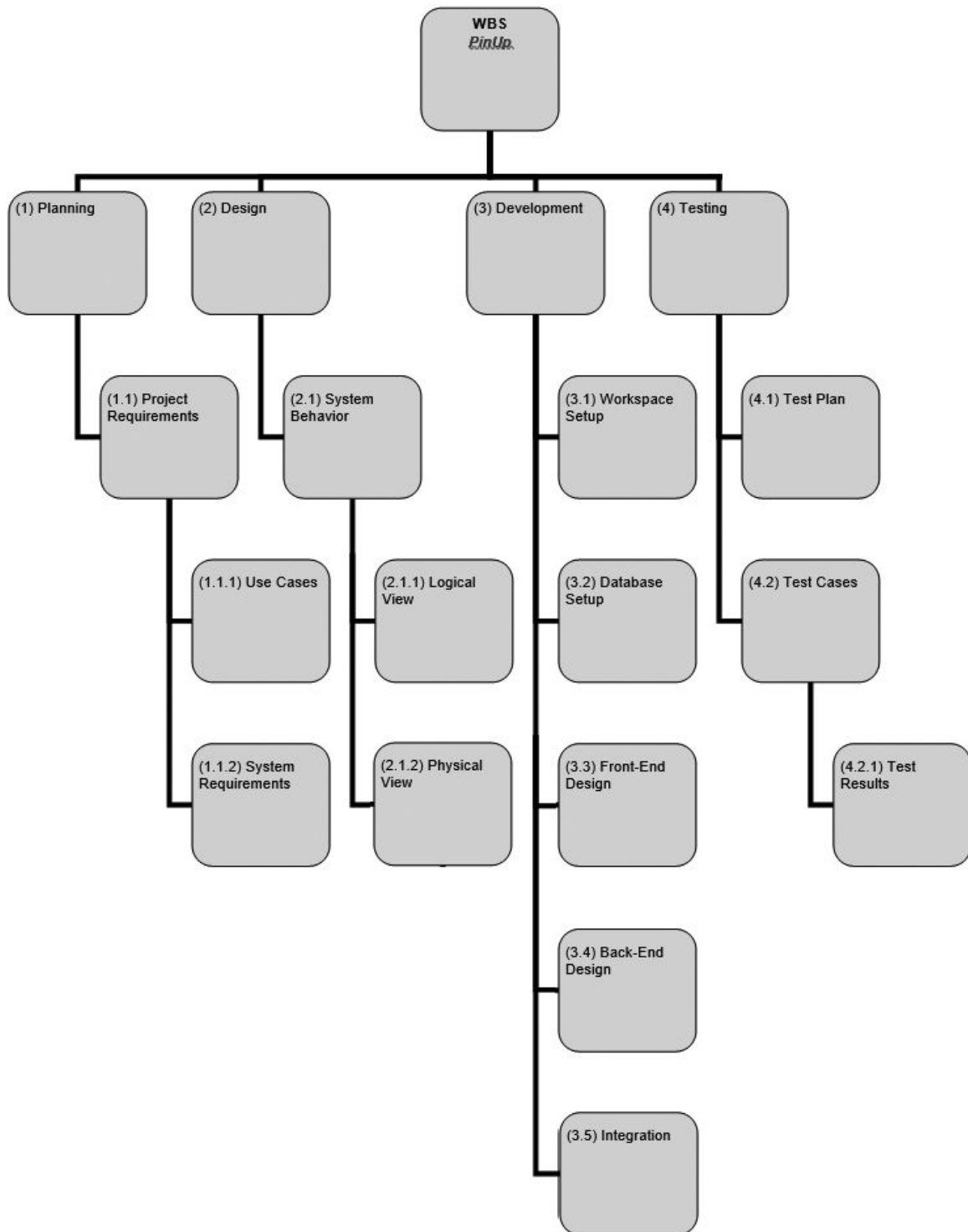
The purpose of this document is to describe the architecture and design of the PinUp application in a way that addresses the interests and concerns of all major stakeholders. For this application the major stakeholders are:

- Users and the customer – desire assurances that the design of the application will provide ease of use, a quality experience navigating the system and reliability with minimized downtime.
- Developers – the system architecture will provide developers with a simplistic coding environment.
- Project Manager - the project manager is responsible for assigning tasks and coordinating development work. He or she wants an architecture that divides the system into components of roughly equal size and complexity that can be developed simultaneously with minimal dependencies. For this to happen, the modules need well-defined interfaces.

2 Design Goals and Schedule

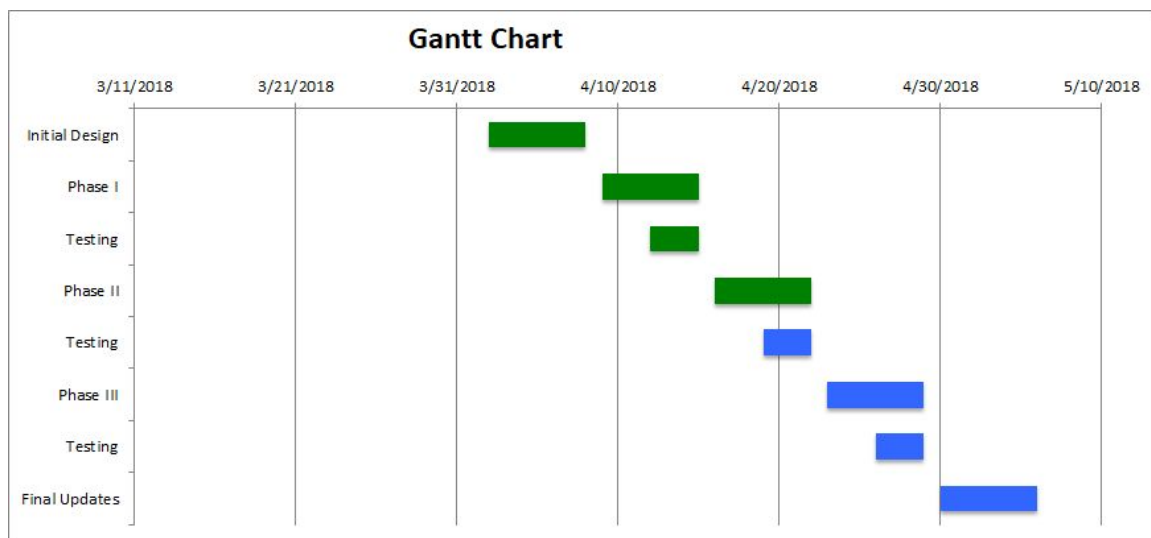
The design priorities for the PinUP application are:

- The design should minimize complexity and development effort.
- The design should emphasize ease of use.
- The design should provide effortless navigation.
- The design should compartmentalize system dependencies and components.
- The design should enable seamless system upgrades with minimize downtime.

Work Breakdown Structure (WBS)

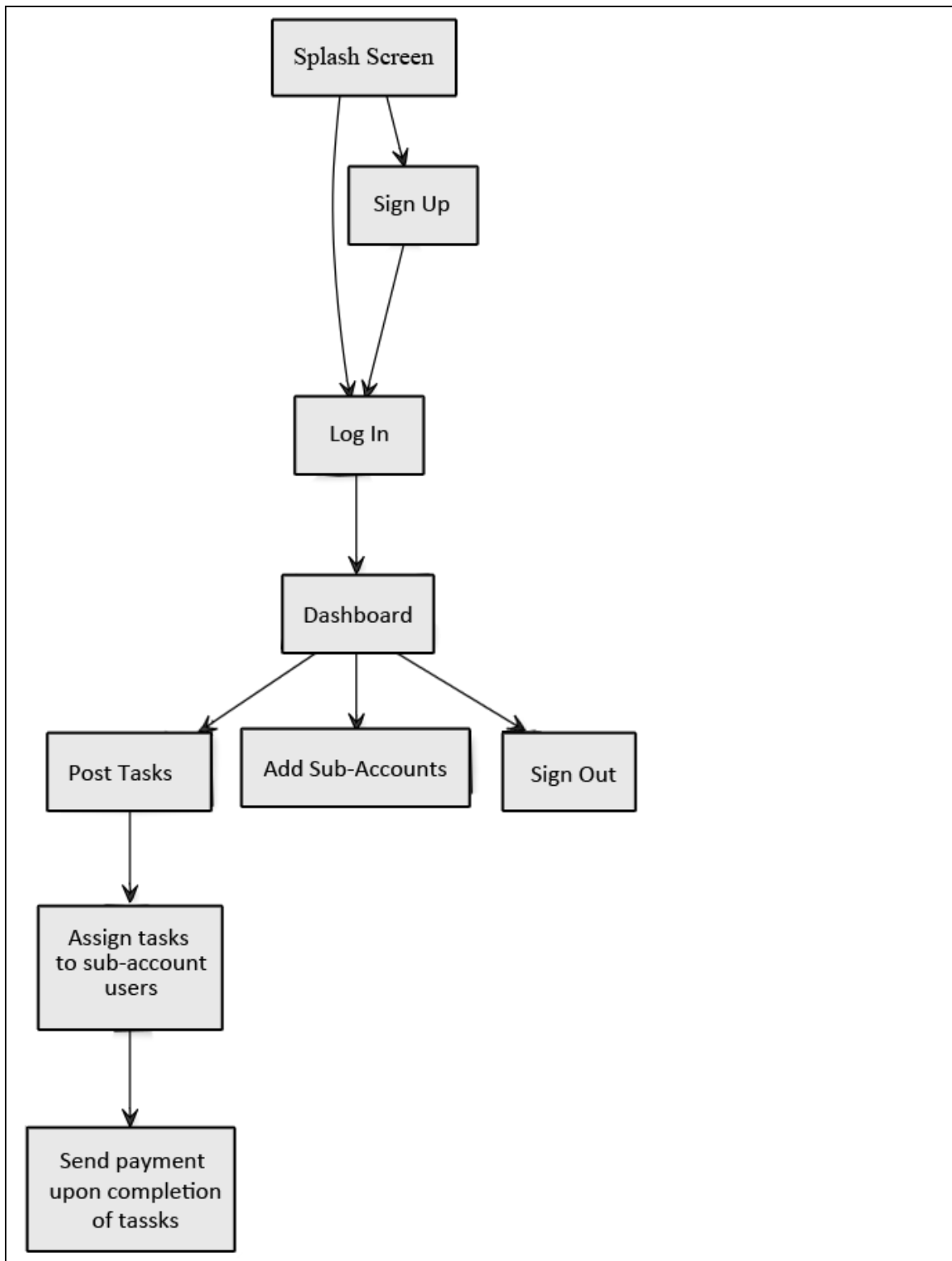
Gantt Chart

Task Name	Start	End	Duration (days)
Initial Design	4/2/2018	4/8/2018	6
Phase I	4/9/2018	4/15/2018	6
Testing	4/12/2018	4/15/2018	3
Phase II	4/16/2018	4/22/2018	6
Testing	4/19/2018	4/22/2018	3
Phase III	4/23/2018	4/29/2018	6
Testing	4/26/2018	4/29/2018	3
Final Updates	4/30/2018	5/6/2018	6



3 System Behavior

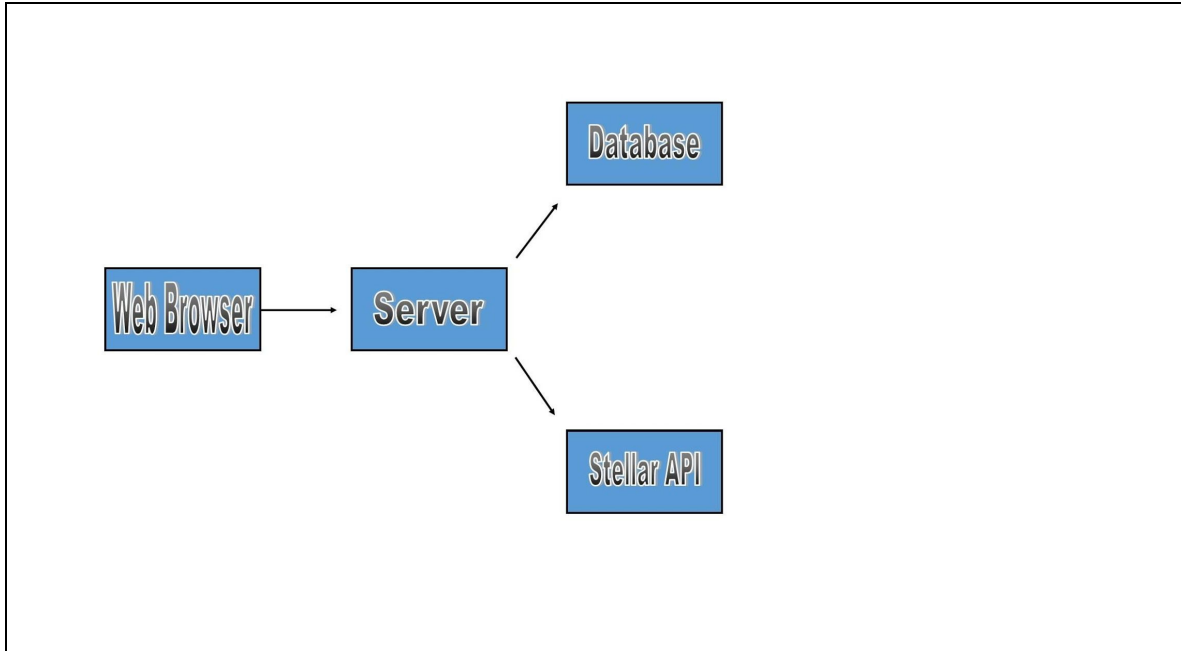
The architecture description presented here starts with a review of the expected system behavior to set the stage for the architecture description that follows. For a more detailed account of software requirements, see the requirements document.



4 Logical View

4.1 High-Level Design (Architecture)

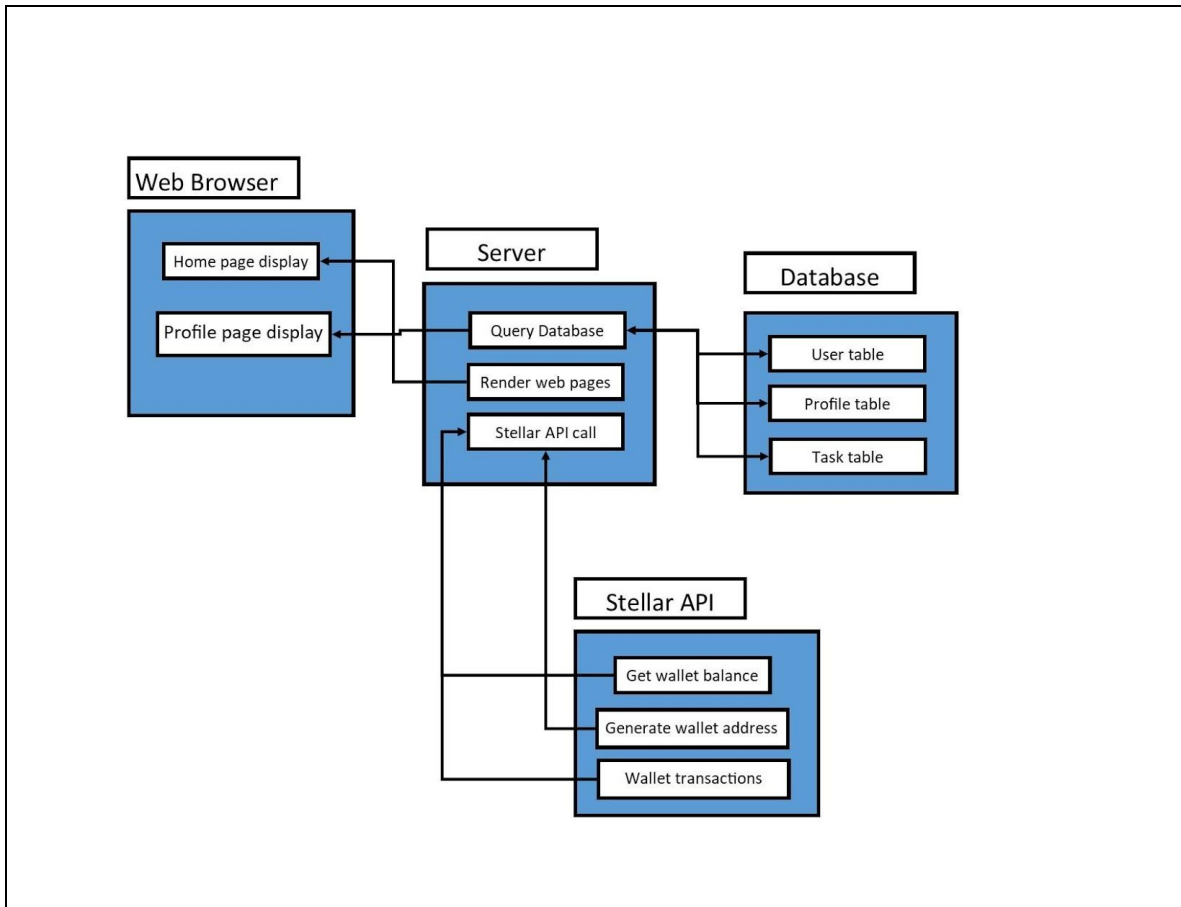
The high-level view or architecture consists of 4 major components:



System Architecture

- The **Web Browser** provides the user with an interface to interact with the application and to send and receive information from the **Server**.
- The **Server** is a central repository for the flow of data within the application,
- The **Database** provide a persistent storage area for application data.
- The **Stellar API** is the third-party system component to achieve user transactions and digital currency information.

4.2 Mid-Level Design



4.3 Detailed Class Design

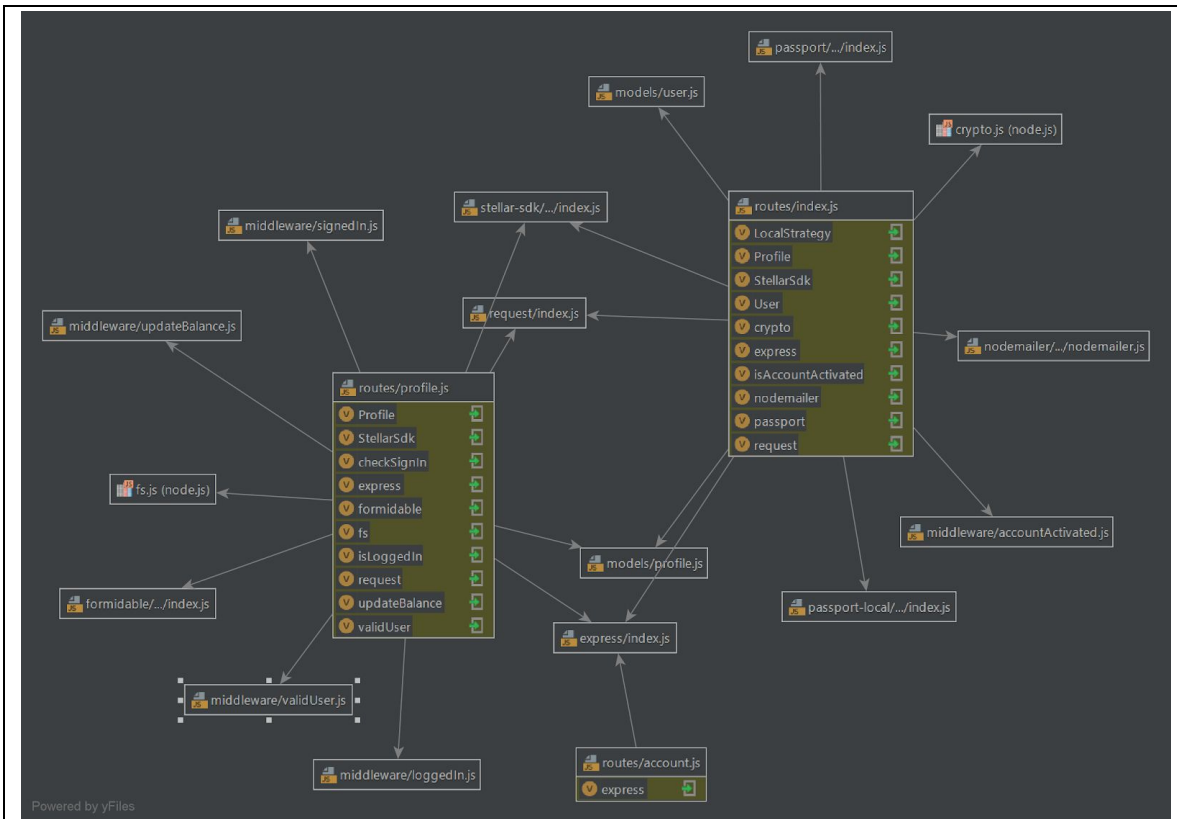


Figure 1. HTTP routes UML

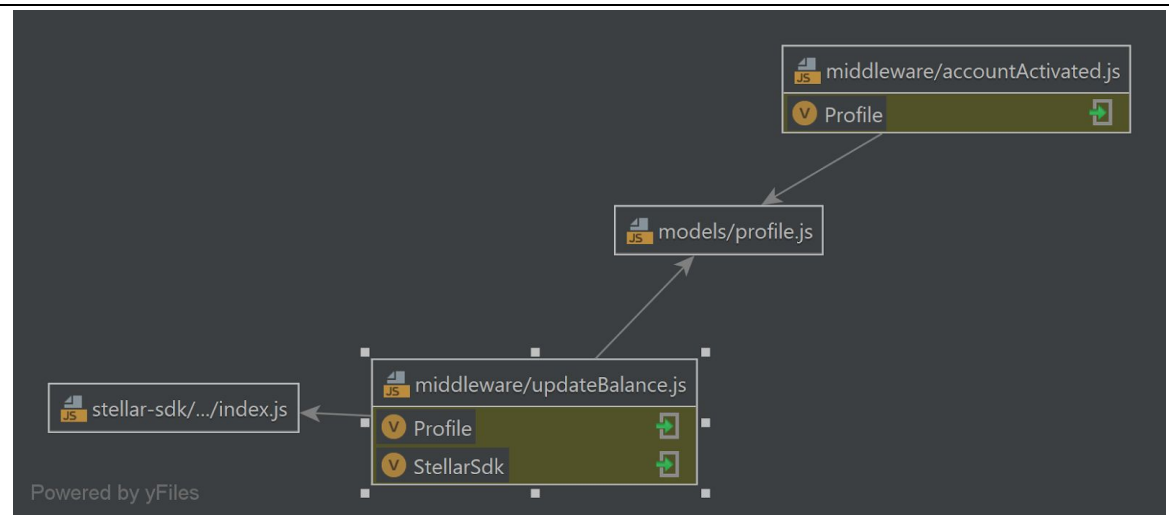
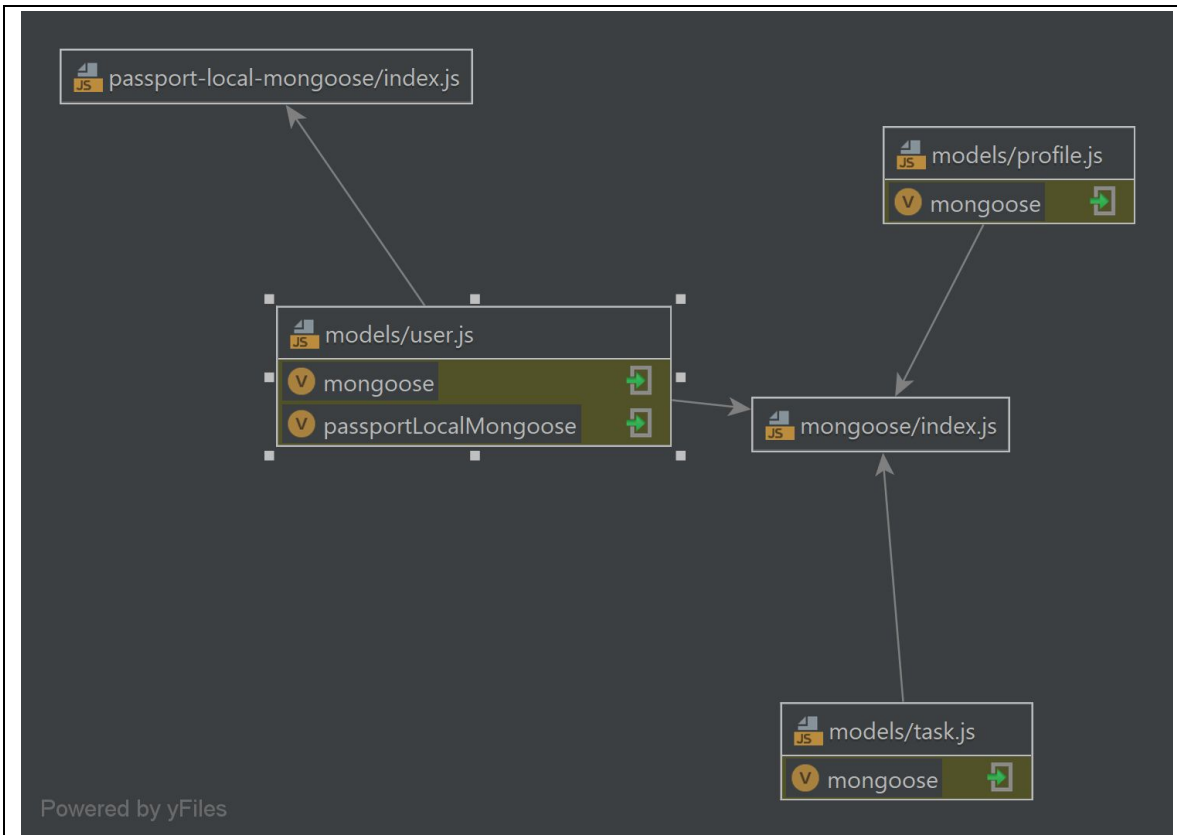
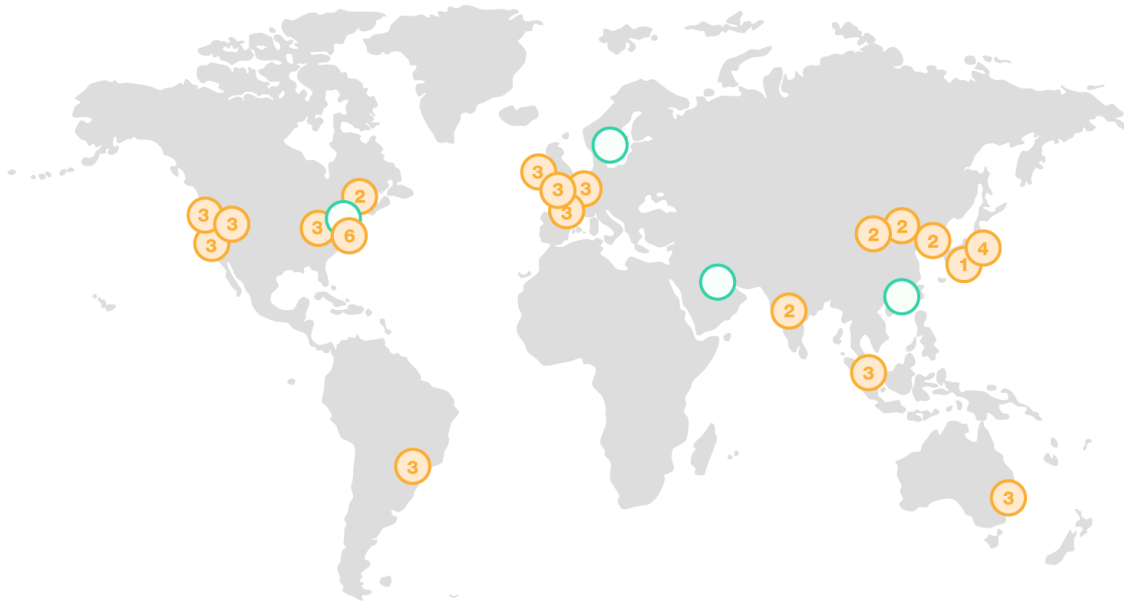


Figure 2. Application middleware JavaScript UML

**Figure 3. Application models JavaScript UML**

5 Physical View

Amazon Web Services (AWS) provides “availability centers” for its customers’ web based needs. In our area, there are six AWS centers in Northern Virginia, referred to as the “US-East” zone of use. Here is a map, taken from the AWS website at <https://aws.amazon.com/about-aws/global-infrastructure/> :



The Stellar Network will also be hosting logs of transactions between PinUp clients. Unfortunately, this network is a very widely decentralized one, and cannot be geographically described. In the words of the Stellar site at <https://www.stellar.org/developers/guides/get-started/> :

“The Stellar network is a worldwide collection of Stellar Cores, each maintained by different people and organizations. The distributed nature of the network makes it reliable and safe.”

6 Use Case View

The following are structurally relevant use cases for the PinUp application. In each case, the Actor involved is the customer using the application.

6.1 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.

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 (form will not allow)
- Missing form information (form will not allow)

6.2 Link Sub-Accounts

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

Flow of Events

1. Clicking the “Add Partnership” button on user dashboard begins the process.
2. A pop-up form is displayed.
3. Required input fields will be properly marked.

4. If requested sub-account is a previously established account on system, then:
5. The sub-user's email address is the only information required to link accounts.
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.

6.3 Log-In

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

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

6.4 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.

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.

6.5 Complete Payout Process

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

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 approval button for the completed task.
 - 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

Appendix A: Acronyms and Abbreviations

A list of the acronyms and abbreviations used in this document and the meaning of each:

App: Application
AWS: Amazon Web Services
API: Application Programming Interface
Sub: Subcontractor, or subordinate