VCU College of Engineering

CS 25-327 XState Builder

Preliminary Design Report

Prepared for

Mahesh Nair & Jaquelyn Dellinger

Capital One

By

Sohum Dharamsi

Neil Randeri

Bryan Wheeler

Thien Dang

Under the supervision of

Irfan Ahmed

Date

December 12th, 2024

Executive Summary

This project aims to develop a web application that facilitates the creation of XState files to define customers' event-driven workflows. The platform will enable call center agents to fill out an intake form via Slack, generating XState definition files based on their responses.

The primary objectives are as follows:

1. Web Application Development
    a. Create a user-friendly interface for displaying available XState definitions
    b. Enable users to trigger events within each XState definition.
2. Database Integration
    a. Establish database tables to manage relevant data, associating tasks with specific workflow instances.
3. Slack Integration
    a. Design an intake form within Slack to collect user input.
    b. Ensure the form poses relevant, user-friendly questions to simplify the creation of XState definition files.
    c. Automate the opening of pull requests in a GitHub repository, reflecting the newly generated XState definitions.

This project will streamline the process of defining customer workflows allowing agents to more effectively manage a higher volume of cases.

Table of Contents

Section A. Problem Statement

Industry Context

The project falls within the field of customer service management and workflow automation, specifically focusing on call centers. Call centers are critical in delivering customer support across various industries, including telecommunications, finance, healthcare, and e-commerce. The effectiveness of a call center is largely determined by its ability to handle customer inquiries promptly and accurately. As businesses increasingly rely on technology to enhance service delivery, the demand for efficient, event-driven workflows has grown.

Identifying the Problem

Call center agents often encounter challenges when defining and managing workflows for handling customer inquiries and events. Traditional methods for creating workflow definitions, often involving manual processes or complex coding, can be time-consuming and error-prone.

This inefficiency can lead to:

1. Long Response Times: Slow workflow design leads to delays in addressing customer needs, impacting overall satisfaction.
2. Increased Training Costs: New agents require extensive training to understand existing workflows, which can vary significantly across different systems.
3. Lack of Flexibility: Static workflow definitions can struggle to adapt to evolving customer needs, resulting in lost opportunities for service improvement.

Prevalence and Costs of the Problem

This issue is widespread in the customer service industry. Many call centers still rely on outdated workflow management systems that do not integrate well with modern communication platforms like Slack.

The costs associated with these inefficiencies can be substantial, including:

1. Economic Costs: Reduced customer retention rates, resulting in lost revenue. Studies show that a 5% increase in customer retention can increase profits by 25% to 95%.
2. Societal Costs: Poor customer service experiences can lead to customer frustration and dissatisfaction, affecting brand reputation and customer loyalty.
3. Health and Safety Costs: In high-pressure environments, such as call centers, inefficiencies can contribute to employee stress and burnout, leading to increased turnover rates.

Project Client and Stakeholders

The primary client for this project is the platform stakeholder (Capital One) seeking to improve the efficiency of their call center operations. Key stakeholders include call center agents, workflow managers, and IT departments responsible for implementing and maintaining the technology stack. Additionally, customers who interact with the call center indirectly benefit from improved workflows and quicker response times.

Historical Perspective and Previous Solutions

Historically, workflow automation in call centers has involved complex coding or the use of dedicated workflow management software. Solutions like BPM (Business Process Management) tools have emerged, but they often lack integration capabilities with communication platforms and can be cumbersome to use for non-technical staff.

Previous attempts to streamline workflow design included:

1. Low-Code/No-Code Platforms: Tools like Zapier and Airtable allow users to automate tasks without deep coding knowledge. However, they often lack the specific features needed for event-driven workflows.
    a. Pros: User-friendly and accessible.
    b. Cons: Limited flexibility for complex workflows.
2. Traditional BPM Solutions: Software like IBM BPM or Appian provides robust workflow management capabilities.
    a. Pros: Highly customizable.
    b. Cons: Requires significant training and is often too complex for everyday users.
3. Custom Solutions: Some organizations develop bespoke systems to manage workflows.
    a. Pros: Tailored to specific needs.
    b. Cons: High development and maintenance costs.

Current Project and Innovations

This project aims to improve upon existing solutions by creating an easy-to-use web application that integrates seamlessly with Slack and automates the generation of XState definitions. This approach addresses several unmet needs:

1. Ease of Use: By utilizing a user-friendly intake form, call center agents can generate workflow definitions without needing technical expertise.
2. Integration with Modern Tools: Leveraging Slack as a communication platform allows for real-time interaction and data collection.

In conclusion, the proposed project aims to fill a significant gap in the market by providing a solution that combines ease of use with robust workflow automation capabilities, ultimately advancing customer service technology and improving operational efficiency in call centers. By addressing the identified challenges, this initiative seeks to enhance both agent performance and customer satisfaction, contributing positively to the broader industry landscape.

Section B. Engineering Design Requirements

B.1 Project Goals (i.e. Client Needs)

The primary goal of this project is to develop a web application that enables stakeholders to efficiently create, manage, and review event driven workflows using XState. This application will capture user requirements, automate workflow processes, and ensure integration with various tools like Slack and Github.

- Streamlined workflow creation based on user needs
- Integration of state machines for reliable task management
- User-friendly interface to both technical and non-technical users

B.2 Design Objectives

- Workflow Management: Allows users to create, review, and manage event-driven workflow efficiently
- Task Tracking: Tasks are tracked and updated in real-time within each workflow instances
- Integration: Ensure seamless integration of Slack and Github
- State transition: Incorporate XState to manage precise and predictable state transitions in a workflow
- Data Management: Postgres is used to retrieve and store instances and task data

B.3 Design Specifications and Constraints

Each design objective is mapped to a specification:

- Workflow Management: at least three actions available to user to create, review, and manage cases
- Task Tracking: Software allows for one list of tasks in the user interface that updates as tasks are completed
- Integration: Slack and Github are connected to the software and each case will have exactly one representation in both Slack and Github
- State transition: Each case will have exactly one Xstate file
- Data Management: Postgres will retrieve and store each case

Constraints are as follows:

- Software must integrate into Slack and Github
- Software must smoothly integrate into Capital One's software system
- Format of management must follow Capital One case management standards
- Software should follow the correct order of case management provided by Capital One's standards (Open > review > close, etc.)

Section C. Scope of Work

C.1 Deliverables

- Landing Page
- Workflow page
- Database developed
- Slack Integration
- Github Integration
- Final interface

- All work can be done remotely, with consistent communication from team members

C.2 Milestones

- Configuration of initial vue.js application and docker container
- Creation of landing page
- Creation of workflow page
- Design of database schema
- Configuration of PostgreSQL database
- Slack integration / design of intake form
- Github integration
- Testing

C.3 Resources

- Some resources required by the team:
    - Data or mock data from Capital One to test software

Section D. Concept Generation

Design Concept 1: Modular Web Application for Workflow Design and Execution

Description: This concept focuses on creating a modular web application that allows users to design, execute, and manage event-driven workflows using XState. The application is divided into three core modules:

1. **Landing Page Module:** Users can select their desired workflow action (create new instance, review open instance).
2. **Workflow Page Module:** Users can interact with specific workflows by viewing tasks, creating new tasks, or closing existing ones.
3. **Database Integration Module:** Ensures the persistence of workflow and task data, allowing for efficient instance management.

Pros:

- Highly structured and scalable design.
- Easy for users to interact with workflows through intuitive navigation.
- Provides real-time feedback and updates based on user actions.

Cons:

- Requires significant effort in front-end and back-end integration.
- Complex testing needed to ensure data consistency and workflow transitions.

Potential Risks:

- User confusion due to poorly designed interfaces.
- Performance issues in handling large-scale workflows or complex event-driven logic.

Design Concept 2: Guided Workflow Definition Using a Slack Integration

Description: This concept integrates Slack as a user-friendly interface for workflow intake. Users interact with a guided form via Slack to answer structured questions about their workflow requirements. The system generates an XState definition file based on their responses and opens a pull request to a designated GitHub repository.

Pros:

- Leverages a familiar platform (Slack) to simplify the intake process.
- Provides a conversational and guided experience for users.
- Automates the process of creating and storing workflow definitions.

Cons:

- Dependence on Slack for critical functionality, limiting standalone application use.
- Limited to the capabilities and constraints of Slack integrations.

Potential Risks:

- Workflow definition inaccuracies due to incomplete or ambiguous user responses.
- Potential delays or failures in Slack-GitHub integration.

Design Concept 3: Intelligent Workflow Builder with Machine Learning Assistance

Description: This concept adds a machine learning component to the workflow builder to provide users with dynamic recommendations. As users define workflows, the system suggests potential states, transitions, and actions based on similar workflows or historical data. The user can accept, reject, or modify these suggestions.

Pros:

- Reduces user effort in defining workflows by offering pre-built suggestions.
- Improves accuracy and consistency across similar workflows.
- Encourages innovation by exposing users to potential workflow designs they may not have considered.

Cons:

- Requires significant time and resources to develop and train the machine learning model.
- Increased complexity in the application architecture.

Potential Risks:

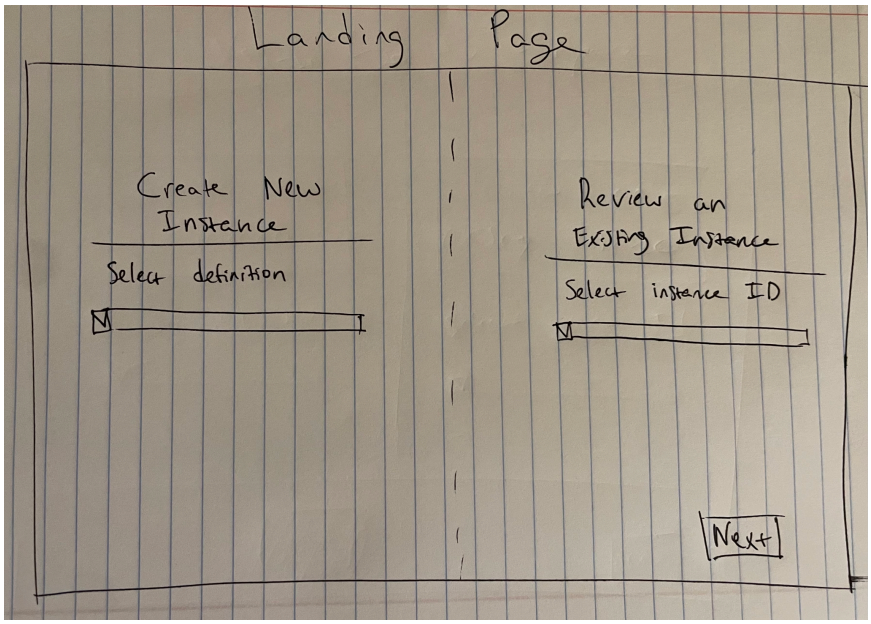
- Incorrect or irrelevant recommendations may frustrate users.
- Over-reliance on the machine learning system could lead to less user understanding of the workflows.

Sub-Problems and Additional Design Concepts:

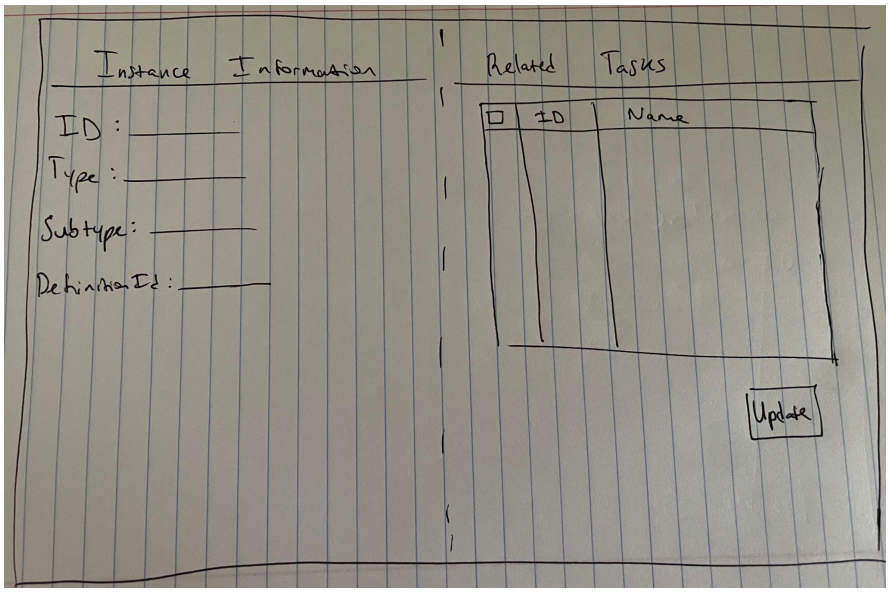
1. **Database Schema Optimization:** Optimize the schema for Instance and Task tables to support scalability and minimize query response time.
2. **Version Control and Collaboration:** Expand GitHub integration to allow multi-user collaboration on workflow definitions, tracking changes, and resolving conflicts.
3. **Real-Time Workflow Visualization:** Provide a live, interactive visualization of workflow states and transitions to improve user understanding.

Initial Design Sketches and Models

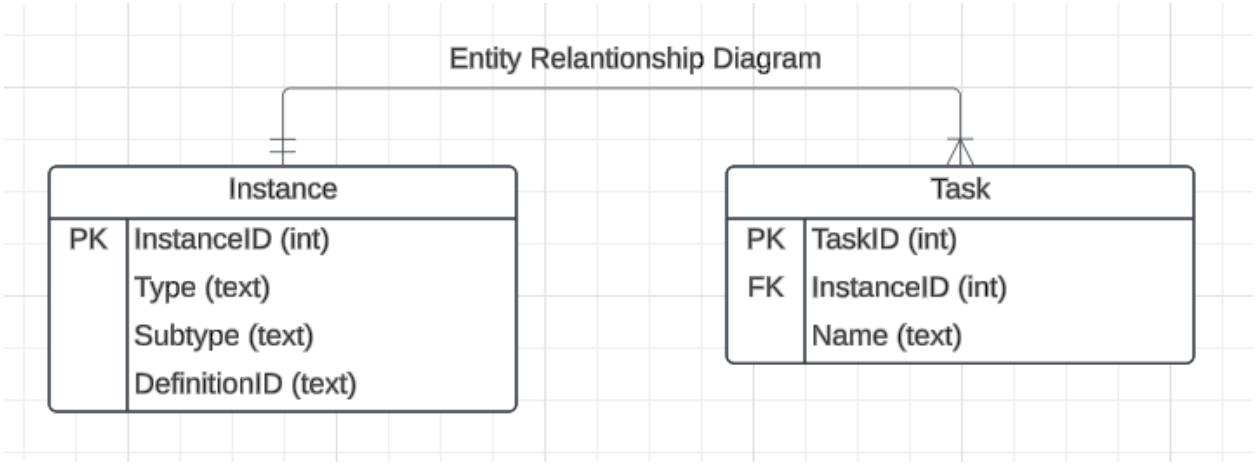
- **Landing Page Flow Diagram:** Visualizing user navigation between creating new instances and reviewing open instances.



- **Workflow State Diagram:** Displaying states, transitions, and events for a sample workflow.



- **Database Entity-Relationship Diagram (ERD):** Showing relationships between Instance and Task tables.



- **Intake Form Sample Questions:** Should ask relevant, user-friendly questions to build out the definition easily based on general user requirements
  (Could potentially use ML here):

1. What is the instance type?
2. What is the instance subtype?
3. What is the flow for your definition? Let's go through each state (Loop through these until flow is complete with all tasks closed and instance closed):
    1. What is this state called? * (Input)
    2. What is the description of this state? (Input)
    3. (Second loop) What event will your definition listen for in this state? *
        1. InstanceCreated
        2. InstanceClosed
        3. CreateTask
        4. TaskCreated
        5. TaskClosed
        6. No More
    4. What action will need to be taken for this event? *
        1. CreateTask
        2. CloseTask
        3. CloseInstance
        4. Log
        5. None
    5. (End second loop) What is the target state for this event? *
        1. CreateTask
        2. CloseTask
        3. CloseInstance
        4. Closed
        5. None

Section E. Concept Evaluation and Selection

Selection Criteria

The following criteria were used to evaluate the design concepts:

1. **Performance:** Ability to meet functional requirements and handle event-driven workflows effectively.
2. **User-Friendliness:** Ease of use and accessibility for the target audience.
3. **Scalability:** Capacity to handle increasing complexity and workload over time.
4. **Implementation Cost:** Resources required for development and deployment.
5. **Reliability:** Consistency and accuracy in workflow execution and management.
6. **Risk:** Potential for failures or issues during implementation and use.

Decision Matrix

Criteria	**Weight**	**Web Application**	**Slack Integration**	**ML Assistance**
Performance	0.3	4.5	3.5	4.0
User-Friendliness	0.2	4.0	4.5	3.0
Scalability	0.2	4.5	3.0	4.5
Implementation Cost	0.1	3.5	4.0	2.5
Reliability	0.1	4.5	3.5	3.5
Risk	0.1	4.0	3.5	2.5
Total Score		**4.3**	**3.7**	**3.5**

Selected Concept

The project will adopt a hybrid approach that incorporates elements from all three design concepts.

Rationale for Selection

This mixed approach leverages the strengths of each concept while mitigating their individual limitations:

- The **Modular Web Application** will provide the foundation, ensuring a scalable, reliable, and user-friendly system for workflow management.
- The **Slack Integration** will serve as an additional interface for workflow intake, offering a conversational and guided experience for users already familiar with Slack.
- The **Machine Learning Assistance** component will be selectively integrated to enhance the user experience by providing intelligent recommendations during workflow definition.

By combining these concepts, the project achieves a robust, scalable, and user-centric design. This approach allows flexibility in addressing various user needs and opens the door for future enhancements based on feedback and evolving requirements.

Section F. Design Methodology

F.1 Computational Methods

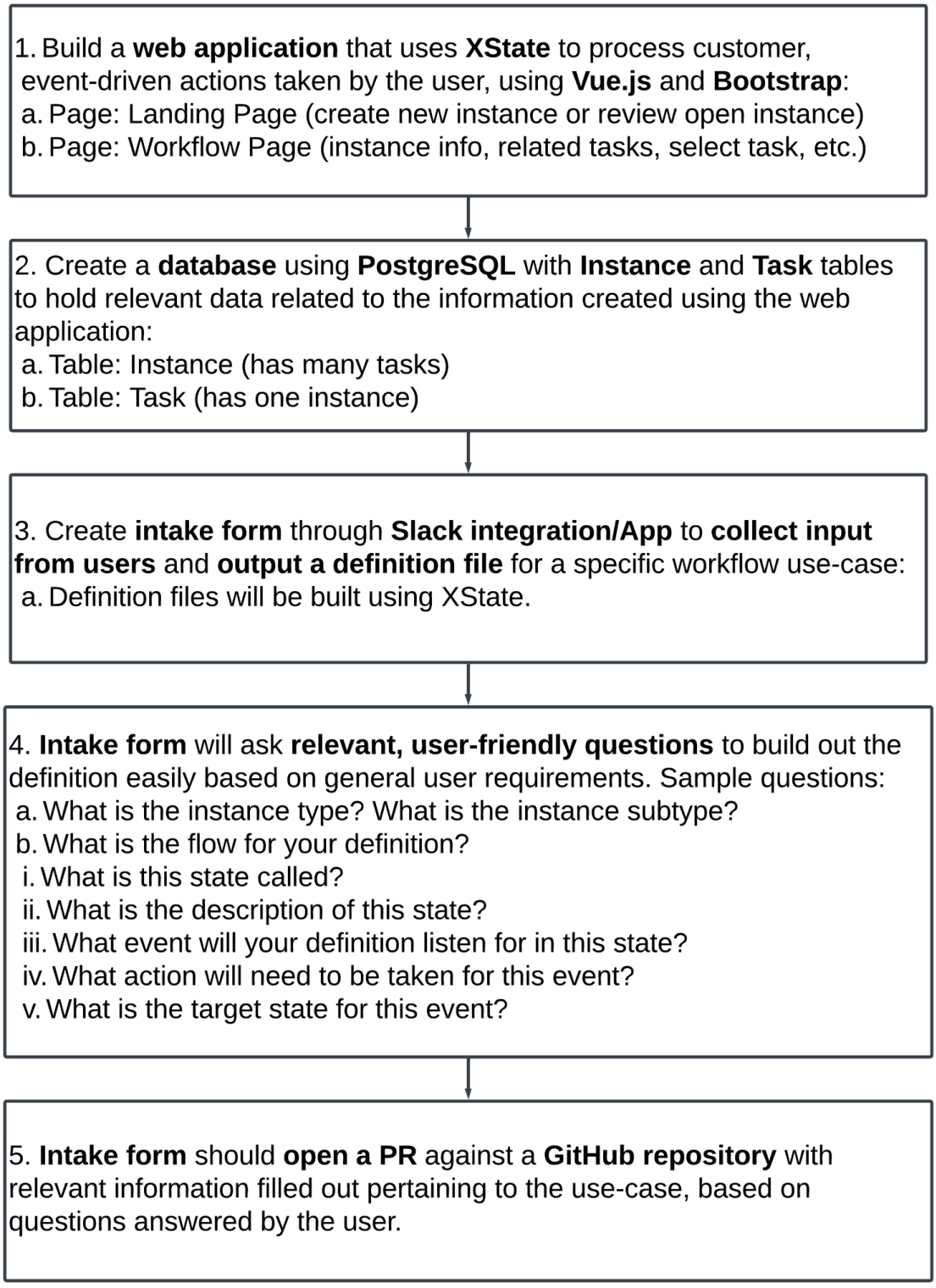
The project will utilize computational tools to model and simulate workflow transitions and database interactions. Tools like XState Inspector will be used to debug and visualize state transitions. Database schema design will be validated through SQL performance testing tools such as pgAdmin and Postman for API calls.

F.2 Experimental Methods

Prototype testing will be conducted in stages:

1. **Front-End Testing:** Use tools like Cypress and Jest to ensure the modular web application works as intended and delivers a smooth user experience.
2. **Slack Integration Testing:** Simulate user interactions in a controlled Slack workspace to validate intake forms and workflow generation.
3. **Machine Learning Testing:** Evaluate the accuracy and relevance of recommendations using test datasets derived from mock workflows.

F.3 Architecture/High-level Design



F.4 Validation Procedure

The final design will undergo a comprehensive validation process:

1. **Verification Testing:** Ensure all features meet their specifications. For example, database queries will be benchmarked for performance, and all state transitions will be logged for correctness.
2. **User Validation:** Conduct usability testing sessions with stakeholders and target users to gather feedback on the hybrid approach.
3. **Client Demonstration:** Present the final prototype to Capital One by mid-April, showcasing its modular design, Slack integration, and machine learning recommendations.
4. **Feedback Collection:** Use surveys and structured interviews to capture client and user feedback. Observations of real-time usage will also inform iterative improvements.

Section G. Results and Design Details

G.1 Modeling Results: Non-Applicable

G.2 Experimental Results: Non-Applicable

G.3 Prototyping and Testing Results: Shown below

Landing Page:

Code Blame 93 lines (81 loc) · 2.98 KB

```vue
<template>
  <div>
    <div class="banner d-flex justify-content-between align-items-center p-3">
      <h1 class="navy-blue">Landing Page</h1>
      <img src="../assets/Capital_One_logo.png" alt="Capital One Logo" class="logo">
    </div>
    <div class="container mt-5 pt-5">
      <h2>Create or Review an Instance</h2>
      <div class="d-flex justify-content-center my-4">
        <button class="btn btn-primary mx-2" @click="selectOption('new')">Create New Instance</button>
        <button class="btn btn-secondary mx-2" @click="selectOption('review')">Review Open Instance</button>
      </div>

      <div v-if="selectedOption === 'new'" class="form-group">
        <label for="definitionSelect">Select a Definition:</label>
        <select id="definitionSelect" v-model="selectedDefinition" class="form-control">
          <option disabled value="">Please select a definition</option>
          <option v-for="definition in definitions" :key="definition" :value="definition">
            {{ definition }}
          </option>
        </select>
      </div>

      <div v-if="selectedOption === 'review'" class="form-group mt-3">
        <label for="instanceSelect">Select an Instance ID:</label>
        <select id="instanceSelect" v-model="selectedInstance" class="form-control">
          <option disabled value="">Please select an instance ID</option>
          <option v-for="instance in instances" :key="instance" :value="instance">
            {{ instance }}
          </option>
        </select>
      </div>

      <div class="d-flex justify-content-center mt-4">
        <button class="btn btn-success" @click="handleSubmit">Submit</button>
      </div>
    </div>
  </div>
</template>



<script>
export default {
  data() {
    return {
      selectedOption: '',
```

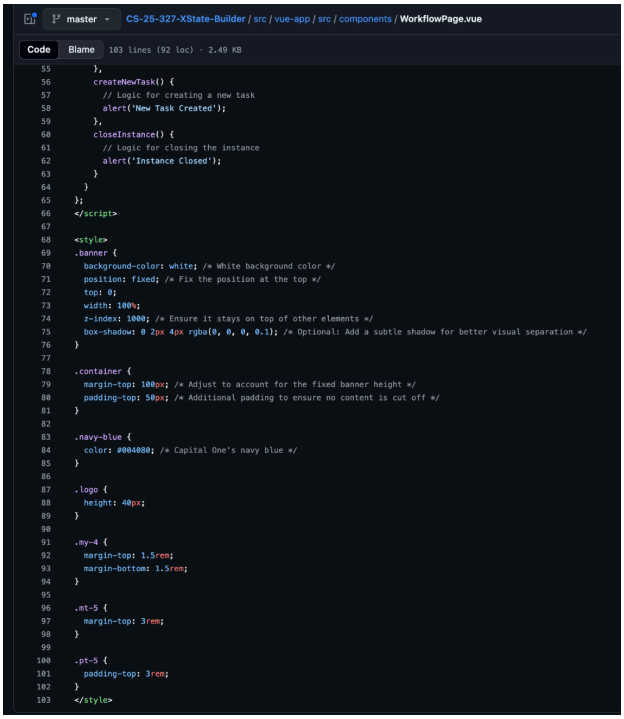
master CS-25-327-XState-Builder / src / vue-app / src / components / LandingPage.vue

Code Blame 93 lines (81 loc) · 2.98 KB

```vue
<script>
export default {
  data() {
    return {
      selectedOption: '',
      selectedDefinition: '',
      selectedInstance: '',
      definitions: ['Definition 1', 'Definition 2', 'Definition 3'],
      instances: ['Instance 1', 'Instance 2', 'Instance 3']
    };
  },
  methods: {
    selectOption(option) {
      this.selectedOption = option;
    },
    handleSubmit() {
      if (this.selectedOption === 'new' && this.selectedDefinition) {
        this.$router.push({ name: 'workflow', params: { definition: this.selectedDefinition } });
      } else if (this.selectedOption === 'review' && this.selectedInstance) {
        this.$router.push({ name: 'workflow', params: { instance: this.selectedInstance } });
      }
    }
  }
};
</script>

<style>
.banner {
  background-color: white; /* White background color */
  position: fixed; /* Fix the position at the top */
  top: 0;
  width: 100%;
  z-index: 1000; /* Ensure it stays on top of other elements */
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1); /* Optional: Add a subtle shadow for better visual separation */
}

.container {
  margin-top: 100px; /* Adjust to account for the fixed banner height */
  padding-top: 50px; /* Additional padding to ensure no content is cut off */
}

.navy-blue {
  color: #004080; /* Capital One's navy blue */
}

.logo {
  height: 40px;
}
</style>
```

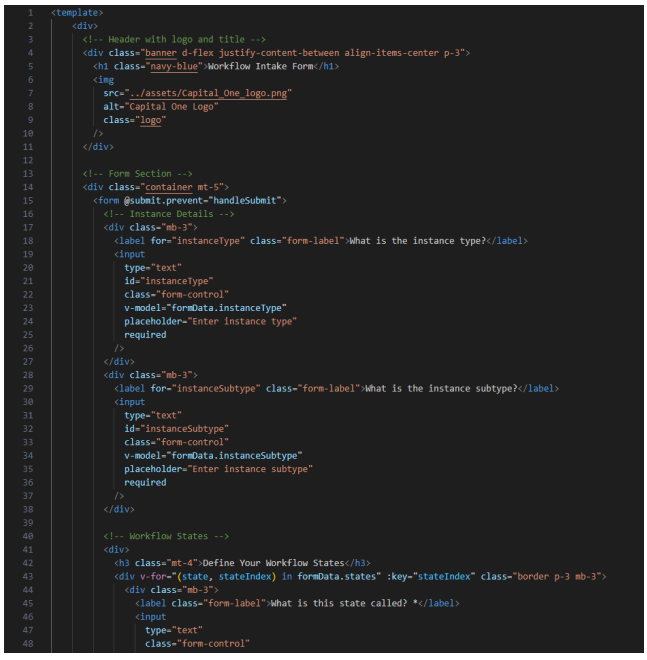
Workflow Page:

```
Code  Blame  103 lines (92 loc) · 2.49 KB

<template>
  <div>
    <div class="banner d-flex justify-content-between align-items-center p-3">
      <h1 class="navy-blue">Workflow Page</h1>
      <img src="../assets/Capital_One_logo.png" alt="Capital One Logo" class="logo">
    </div>
    <div class="container mt-5 pt-5">
      <div class="my-4">
        <h2>Instance Information</h2>
        <p>{{ instanceInfo }}</p>
      </div>

      <div class="my-4">
        <h2>Related Tasks</h2>
        <ul class="list-group">
          <li class="list-group-item" v-for="task in tasks" :key="task.id" @click="selectTask(task)">
            {{ task.name }}
          </li>
        </ul>
      </div>

      <div v-if="selectedTask" class="my-4">
        <h2>Task Details</h2>
        <p>{{ selectedTask.details }}</p>
        <div class="d-flex justify-content-between mt-3">
          <button class="btn btn-danger" @click="closeTask">Close Task</button>
          <button class="btn btn-primary" @click="createNewTask">Create New Task</button>
          <button class="btn btn-warning" @click="closeInstance">Close Instance</button>
        </div>
      </div>
    </div>
  </div>
</template>


<script>
export default {
  data() {
    return {
      instanceInfo: 'Instance information goes here...',
      tasks: [
        { id: 1, name: 'Task 1', details: 'Details for Task 1' },
        { id: 2, name: 'Task 2', details: 'Details for Task 2' }
      ],
      selectedTask: null
    };
  },
  methods: {
    selectTask(task) {
      this.selectedTask = task;
```

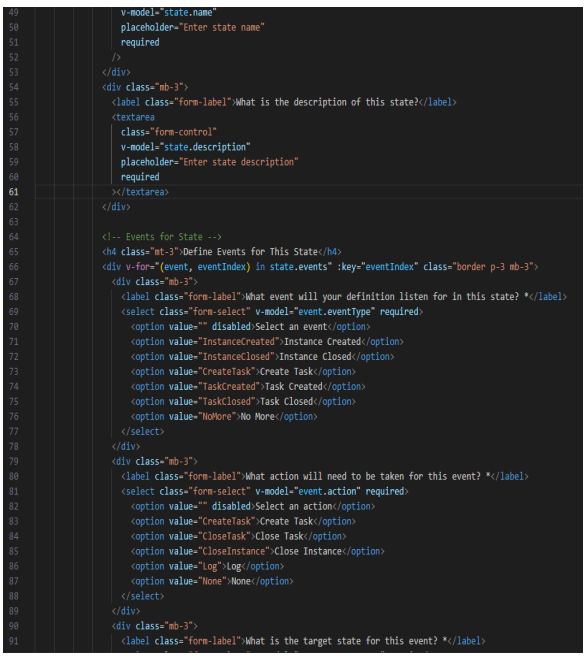


```
master  CS-25-327-XState-Builder / src / vue-app / src / components / WorkflowPage.vue

Code  Blame  103 lines (92 loc) · 2.49 KB

    },
    createNewTask() {
      // Logic for creating a new task
      alert('New Task Created');
    },
    closeInstance() {
      // Logic for closing the instance
      alert('Instance Closed');
    }
  }
};
</script>

<style>
.banner {
  background-color: white; /* White background color */
  position: fixed; /* Fix the position at the top */
  top: 0;
  width: 100%;
  z-index: 1000; /* Ensure it stays on top of other elements */
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1); /* Optional: Add a subtle shadow for better visual separation */
}

.container {
  margin-top: 100px; /* Adjust to account for the fixed banner height */
  padding-top: 50px; /* Additional padding to ensure no content is cut off */
}

.navy-blue {
  color: #004080; /* Capital One's navy blue */
}

.logo {
  height: 40px;
}

.my-4 {
  margin-top: 1.5rem;
  margin-bottom: 1.5rem;
}

.mt-5 {
  margin-top: 3rem;
}

.pt-5 {
  padding-top: 3rem;
}
</style>
```

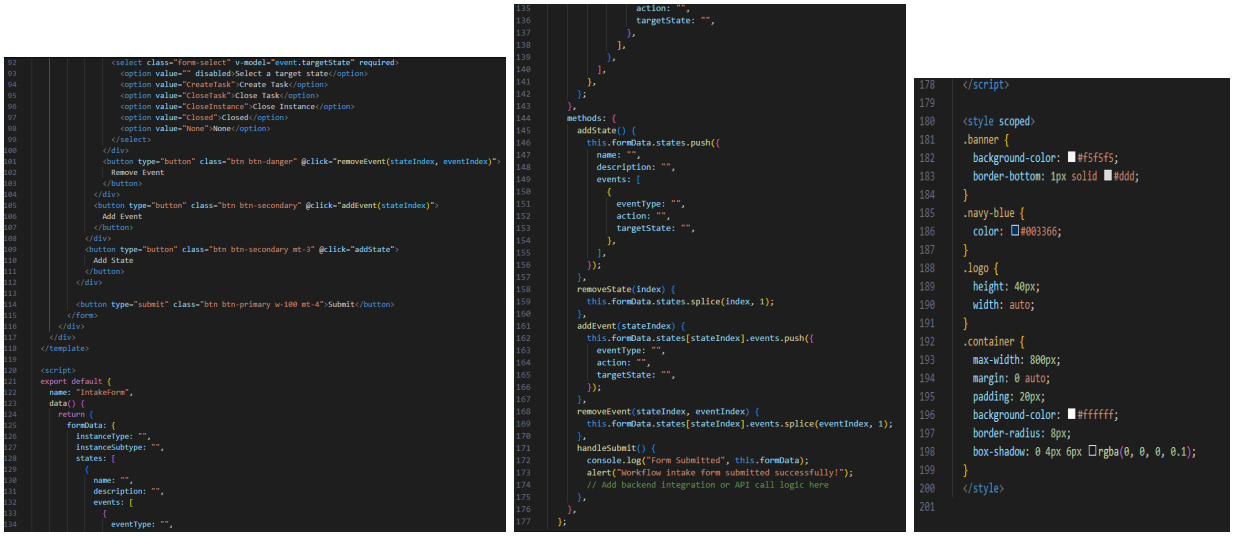
Intake form:



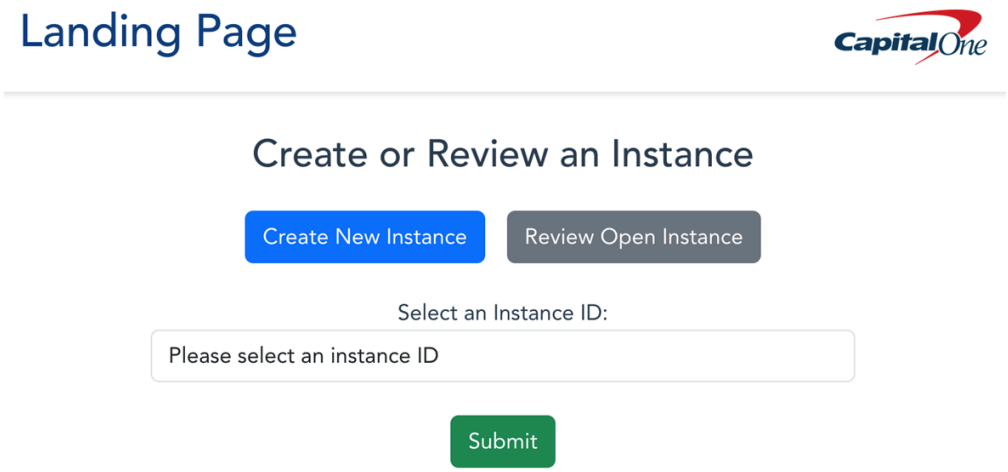
```
<template>
  <div>
    <!-- Header with logo and title -->
    <div class="banner d-flex justify-content-between align-items-center p-3">
      <h1 class="navy-blue">Workflow Intake Form</h1>
      <img
        src="../assets/Capital_One_logo.png"
        alt="Capital One Logo"
        class="logo"
      />
    </div>

    <!-- Form Section -->
    <div class="container mt-5">
      <form @submit.prevent="handleSubmit">
        <!-- Instance Details -->
        <div class="mb-3">
          <label for="instanceType" class="form-label">What is the instance type?</label>
          <input
            type="text"
            id="instanceType"
            class="form-control"
            v-model="formData.instanceType"
            placeholder="Enter instance type"
            required
          />
        </div>
        <div class="mb-3">
          <label for="instanceSubtype" class="form-label">What is the instance subtype?</label>
          <input
            type="text"
            id="instanceSubtype"
            class="form-control"
            v-model="formData.instanceSubtype"
            placeholder="Enter instance subtype"
            required
          />
        </div>

        <!-- Workflow States -->
        <div>
          <h3 class="mt-4">Define Your Workflow States</h3>
          <div v-for="(state, stateIndex) in formData.states" :key="stateIndex" class="border p-3 mb-3">
            <div class="mb-3">
              <label class="form-label">What is this state called? *</label>
              <input
                type="text"
                class="form-control"
```



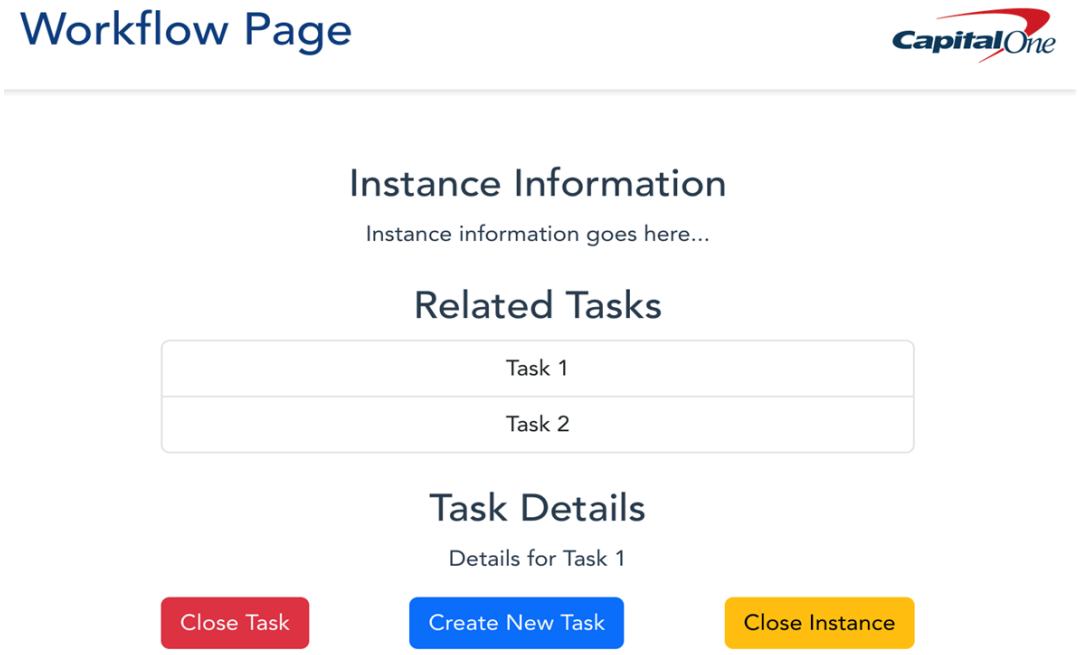
```
                v-model="state.name"
                placeholder="Enter state name"
                required
              />
            </div>
            <div class="mb-3">
              <label class="form-label">What is the description of this state?</label>
              <textarea
                class="form-control"
                v-model="state.description"
                placeholder="Enter state description"
                required
              ></textarea>
            </div>

            <!-- Events for State -->
            <h4 class="mt-3">Define Events for This State</h4>
            <div v-for="(event, eventIndex) in state.events" :key="eventIndex" class="border p-3 mb-3">
              <div class="mb-3">
                <label class="form-label">What event will your definition listen for in this state? *</label>
                <select class="form-select" v-model="event.eventType" required>
                  <option value="" disabled>Select an event</option>
                  <option value="InstanceCreated">Instance Created</option>
                  <option value="InstanceClosed">Instance Closed</option>
                  <option value="CreateTask">Create Task</option>
                  <option value="TaskCreated">Task Created</option>
                  <option value="TaskClosed">Task Closed</option>
                  <option value="NoMore">No More</option>
                </select>
              </div>
              <div class="mb-3">
                <label class="form-label">What action will need to be taken for this event? *</label>
                <select class="form-select" v-model="event.action" required>
                  <option value="" disabled>Select an action</option>
                  <option value="CreateTask">Create Task</option>
                  <option value="CloseTask">Close Task</option>
                  <option value="CloseInstance">Close Instance</option>
                  <option value="Log">Log</option>
                  <option value="None">None</option>
                </select>
              </div>
              <div class="mb-3">
                <label class="form-label">What is the target state for this event? *</label>
```



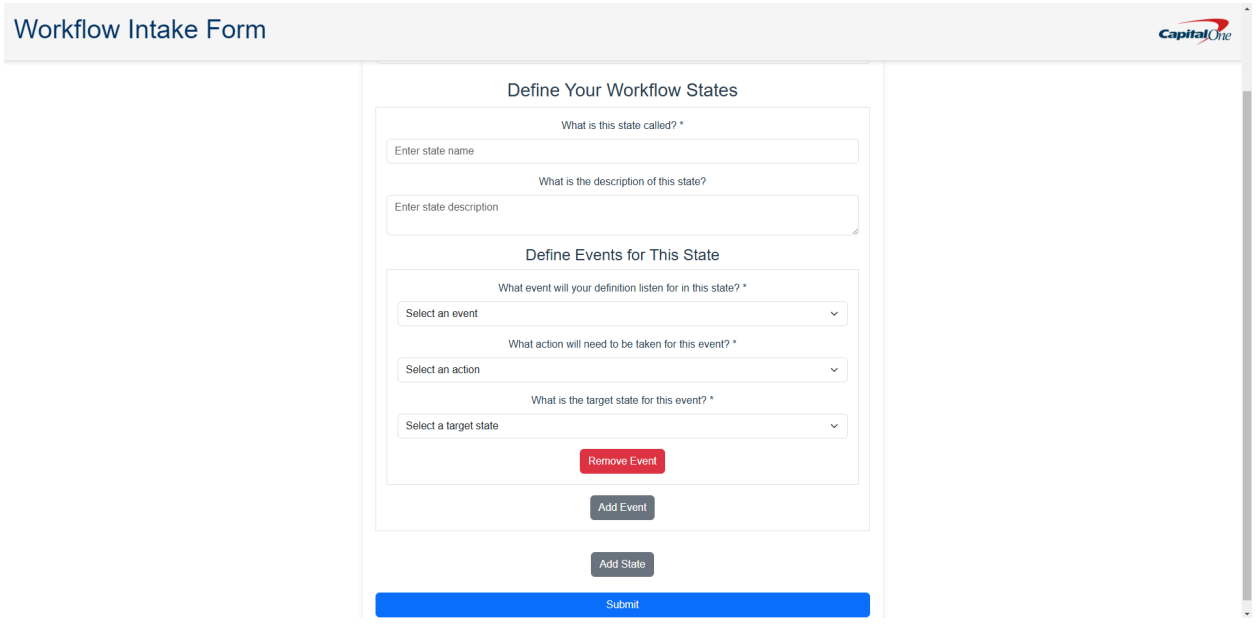
G.4. Final Design Details/Specifications (example subsection)



- Provides users with user-friendly interface with options to create new workflow instances or review existing ones, supported by input validation and seamless navigation
- Incorporates a consistent design and branding of Capital One



- Displays instances and related tasks dynamically, with actionable buttons for closing tasks, creating new ones, and closing workflow instances.
- Supports a clean, scalable design, ensuring easy integration with Slack and Github for future enhancements.



- User can define workflow states by adding/removing specific names, events, actions, and target states
- Clean interface with dropdowns and input fields aligning with Capital One branding

Section H. Societal Impacts of Design

H.1 Public Health, Safety, and Welfare:

As of now, there are limited design safety features. However, the team will be implementing features like **Authentication and Authorization** to protect workflows and task data from unauthorized access, **Data Validation and Sanitization** to ensure safe handling of inputs from technical and non-technical users, and **Real-Time Task Tracking** to prevent task duplication or loss. These features will not have a positive impact on public health, safety, and welfare, but will have a positive impact on Capital One by fostering secure and efficient task and workflow management. These features will lead to better productivity and secure data management. By integrating safety considerations like the ones stated above, the project promotes effective software use.

H.2 Societal Impacts:

Our software design leads to improving organizational efficiency, boosting productivity, and increasing accountability across sectors, ultimately leading to better service delivery and economic benefits within communities and society.

H.3 Political/Regulatory Impacts: This section is non-applicable for our project

H.4. Economic Impacts: Our software design for workflow management can impact Capital One's economics by increasing productivity, reducing errors, optimizing resource allocation, decreasing turnaround times, and ultimately lead to cost savings and better visibility into processes, allowing employees to focus on higher-value activities.

H.5 Environmental Impacts: This section is non-applicable for our project

H.6 Global Impacts: Similarly to Section H.4, the economic impacts of our design flow into the global impacts. Capital One's economics can not only grow on a private level, but depending on the continued use of our application, its economics may also improve on a global scale.

H.7. Ethical Considerations: This section is non-applicable for our project

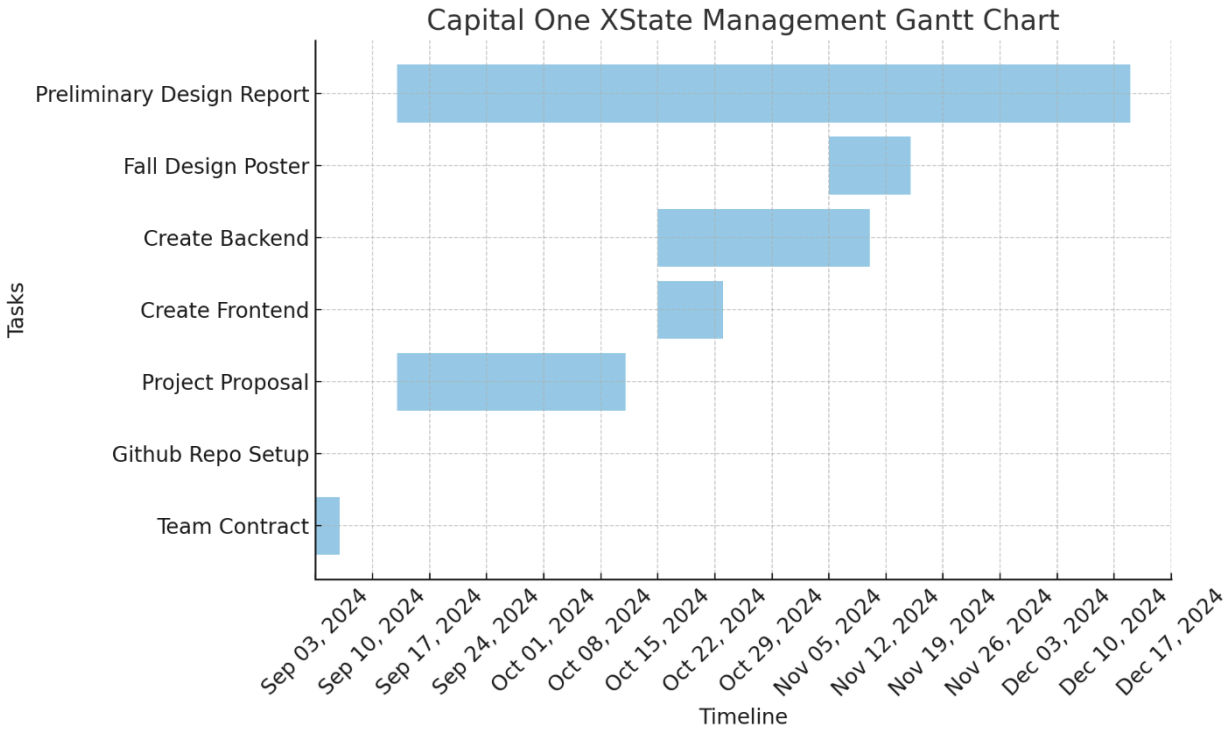
Section I. Cost Analysis

There are no costs involved with our project. As a software project, many open source development applications such as Vue, Github, etc. were used. No manufacturing costs apply, and no physical materials were used. Capital One can apply our design using the same open source software that was implemented in the design.

Section J. Conclusions and Recommendations

Initially, the team outlined the project goals and objectives: creating a web application that leverages XState for reliable state transitions, integrates seamlessly with Slack and GitHub, and provides efficient task and data management using PostgreSQL. The team then focused on creating designs for the user interface. Early designs focused on user requirements like the need for a user-friendly interface accommodating both technical and non-technical users. Throughout development, the team overcame obstacles such as resolving bugs with our web application and improving the accessibility of the user interface. These steps led to the final preliminary design, which features a working user interface, as shown in Section G.4. Future enhancements will include expanding the needed integration with Xstate, Github, and Slack, as well as connection with a database.

Appendix 1: Project Timeline



Appendix 2: Team Contract (i.e. Team Organization)

Step 1: Get to Know One Another. Gather Basic Information.

Task: This initial time together is important to form a strong team dynamic and get to know each other more as people outside of class time. Consider ways to develop positive working relationships with others, while remaining open and personal. Learn each other's strengths and discuss good/bad team experiences. This is also a good opportunity to start to better understand each other's communication and working styles.

Team Member Name	***Strengths each member bring to the group***	***Other Info***	***Contact Info***
Sohum Dharamsi	Industry experience with Git/GitHub Enterprise, Communication, Very Organized	Like doing team projects, always looking to learn something new like XState, excited to work with Capital One.	dharamsiss@vcu.edu 703-398-0897
Neil Randeri	Experience with Git, backend/frontend programming	Have worked on team projects professionally where release dates are hard set	randerin@vcu.edu 571-621-0735
Thien Dang	Experience with front/backend and Git	Prior history in doing team projects alongside having the eagerness to learn and be flexible	dangtp@vcu.edu 910-987-7341
Bryan Wheeler	Experience with backend programming, communication, Git	Have prior experience working in dev team in industry	wheelerbt2@vcu.edu 804-762-1240

Other Stakeholders	***Notes***	***Contact Info***
Mahesh Nair Capital One	Main Capital One POC	mahesh.bahulleyannair@capitalone.com
Jacquelyn Dellinger Capital One	XState Builder Mentor/POC	jacquelyn.dellinger@capitalone.com
Irfan Ahmed VCU	Associate Cybersecurity Professor at VCU	iahmed3@vcu.edu

Step 2: Team Culture. Clarify the Group's Purpose and Culture Goals.

Task: Discuss how each team member wants to be treated to encourage them to make valuable contributions to the group and how each team member would like to feel recognized for their efforts. Discuss how the team will foster an environment where each team member feels they are accountable for their actions and the way they contribute to the project. These are your Culture Goals (left column). How do the students demonstrate these culture goals? These are your Actions (middle column). Finally, how do students deviate from the team's culture goals? What are ways that other team members can notice when that culture goal is no longer being honored in team dynamics? These are your Warning Signs (right column).

Resources: More information and an example Team Culture can be found in the Biodesign Student Guide "Intentional Teamwork" page (webpage | PDF)

Culture Goals	***Actions***	***Warning Signs***
Being punctual and on time to every meeting	- Brainstorm meeting times in iMessage GroupChat. Then finalize on Google Calendar - Send reminder text message/e-mail the day before meeting	- Student misses first meeting, warning is granted - Student misses meetings afterwards – issue is brought up with faculty advisor
Consistent communication (e.g. Informing the group of any delays in completing assignments)	- Stay up to date with each other's project responsibilities - Set reasonable deadlines and note when an extension is needed	- Student shows up for weekly meeting with no considerable work done - warning given - Student consistently shows up for meetings with no work done - issue is brought up with faculty advisor
Maintain professionalism with faculty advisor and sponsor mentor/company	- Provide regular updates to both the faculty advisor and corporate sponsor - Always ensure professional and respectful communication - Double check with team before reaching out to sponsor / advisor	- Student fails to follow up with sponsor or faculty advisor as expected - warning given - Student consistently fails to do so - issue is brought up with faculty advisor

Step 3: Time Commitments, Meeting Structure, and Communication

Task: Discuss the anticipated time commitments for the group project. Consider the following questions (don't answer these questions in the box below):

- What are reasonable time commitments for everyone to invest in this project?
- What other activities and commitments do group members have in their lives?
- How will we communicate with each other?
- When will we meet as a team? Where will we meet? How Often?
- Who will run the meetings? Will there be an assigned team leader or scribe? Does that position rotate or will the same person take on that role for the duration of the project?

Required: How often you will meet with your faculty advisor advisor, where you will meet, and how the meetings will be conducted. Who arranges these meetings?
See examples below.

Meeting Participants	***Frequency*** ***Dates and Times / Locations***	***Meeting Goals*** ***Responsible Party***
Students Only	As needed, on google meet or zoom.	Update group on day-to-day challenges and accomplishments Sohum will record these for the weekly progress reports and meetings with advisors.
Students Only	Every Tuesday from 3:30pm - 5:30pm in ERB Atrium.	Will actively work on the project. Sohum will document these meetings by taking notes on shared Google Doc, taking screenshots of command line, etc, then post on EduSourced and update Capstone Report.
Students + Faculty advisor + Project Sponsor	Every Thursday from 4:00pm - 4:30pm on google meet or zoom. *If we end up needing more time we can adjust accordingly.*	Update faculty advisor and get answers to our questions (Sohum will scribe; Bryan will create meeting agenda and lead meeting) Update project sponsor and make sure we are on the right track (Sohum will scribe; Bryan will create meeting agenda and lead meeting; Neil & Thien will present prototype so far)

Step 4: Determine Individual Roles and Responsibilities

Task: As part of the Capstone Team experience, each member will take on a leadership role, ***in addition to*** contributing to the overall weekly action items for the project. Some common leadership roles for Capstone projects are listed below. Other roles may be assigned with approval of your faculty advisor as deemed fit for the project. For the entirety of the project, you should communicate progress to your advisor specifically with regard to your role.

- **Before meeting with your team**, take some time to ask yourself: what is my "natural" role in this group (strengths)? How can I use this experience to help me grow and develop more?
- **As a group,** discuss the various tasks needed for the project and role preferences. Then assign roles in the table on the next page. Try to create a team dynamic that is fair and equitable, while promoting the strengths of each member.

Communication Leaders

Suggested: Assign a team member to be the primary contact for the client/sponsor. This person will schedule meetings, send updates, and ensure deliverables are met.

Suggested: Assign a team member to be the primary contact for faculty advisor. This person will schedule meetings, send updates, and ensure deliverables are met.

Common Leadership Roles for Capstone

1. **Project Manager:** Manages all tasks; develops overall schedule for project; writes agendas and runs meetings; reviews and monitors individual action items; creates an environment where team members are respected, take risks and feel safe expressing their ideas.
   Required: On Edusourced, under the Team tab, make sure that this student is assigned the Project Manager role. This is required so that Capstone program staff can easily identify a single contact person, especially for items like Purchasing and Receiving project supplies.
2. **Logistics Manager:** coordinates all internal and external interactions; lead in establishing contact within and outside of organization, following up on communication of commitments, obtaining information for the team; documents meeting minutes; manages facility and resource usage.
3. **Financial Manager:** researches/benchmarks technical purchases and acquisitions; conducts pricing analysis and budget justifications on proposed purchases; carries out team purchase requests; monitors team budget.
4. **Systems Engineer:** analyzes Client initial design specification and leads establishment of product specifications; monitors, coordinates and manages integration of subsystems in the prototype; develops and recommends system architecture and manages product interfaces.
5. **Test Engineer:** oversees experimental design, test plan, procedures and data analysis; acquires data acquisition equipment and any necessary software; establishes test protocols and schedules; oversees statistical analysis of results; leads presentation of experimental finding and resulting recommendations.
6. **Manufacturing Engineer:** coordinates all fabrication required to meet final prototype requirements; oversees that all engineering drawings meet the requirements of machine shop or vendor; reviews designs to ensure design for manufacturing; determines realistic timing for fabrication and quality; develops schedule for all manufacturing.

Team Member	***Role(s)***	***Responsibilities***
Sohum Dharamsi	Logistics Manager Group Communication Systems Engineer	- Will facilitate all internal and external interactions - Will help to obtain information from sponsor for the group - Keep a detailed record of meeting notes and share with group - Send out weekly emails and other correspondence - Make sure everyone understands what is going on - Will help with GitHub Enterprise - repo versioning, branching, commits, command console, & pull requests
Neil Randeri	Test Engineer Systems Engineer	- Develop backend testing software and execute testing procedures - Keep track of scheduling and testing as the software becomes more detailed - Facilitate improvements in the product and analyze client specifics
Thien Dang	Financial Manager Test Engineer	- Researches technical purchases - Manages procurement - Keeps track of team budget - Establish test procedures and analyze results
Bryan Wheeler	Project Manager Systems Engineer	- Keep track of purchasing and receiving supplies - Develop schedule for project - Keep all team members up to date on project progress, milestones, and deadlines - Backend programming and design

Step 5: Agree to the above team contract

Team Member: Sohum Dharamsi *Signature: Sohum Dharamsi*

Team Member: Neil Randeri *Signature: Neil Randeri*

Team Member: Thien Dang *Signature: Thien Dang*

Team Member: Bryan Wheeler *Signature: Bryan Wheeler*

References

Provide a numbered list of all references in order of appearance using APA citation format. The reference page should begin on a new page as shown here.

[1] *Xstate: Stately*. Stately RSS. (n.d.). https://stately.ai/docs/xstate#:~:text=XState%20is%20a%20state%20management,%2C%20robust%2C%20and%20visual%20ways.