**Document Title**:  Standard Operating Procedure Document for Development Teams

**Project Title**: Hatching Sparrow CRM

**Document Version**: V1.1 on 3/24/2024.

**Author(s)**: Ryan O’Connor, Full Stack Software Engineer [Name, Title]

**Editor(s):** [Name, Title]

**Company Logo**: [Insert Logo]

**Confidentiality:** Internal

**Table of Contents**

**1. Introduction**

**2. Getting Started: Tools and Setup**

**3. Feature Naming and Documentation**

**4. GitHub Collaboration**

**5. Development and Review Process**

**6. Deadlines and Timelines**

**7. Training and Onboarding for Interns**

**8. Conclusion**

**9. Implementation of the SOP**

**0. Introduction**

**Hello and Welcome to the Hatching Sparrow Project at Up Cancer**

**Organization and Vision:**  
Up Cancer is transitioning from providing web solutions for charities to designing and building a CRM tool called Hatching Sparrow. This project is a collaborative effort among our UI/UX, Frontend, and Backend teams to create a robust, user-friendly CRM platform. Our vision is to develop a tool that empowers organizations to manage their relationships and interactions with clients and stakeholders efficiently.

**1. Getting Started: Tools and Setup**

**Common Tools**

**1. Flock:** Used for team communication and collaboration. It is divided into four channels:

* Frontend
* Backend
* General
* Review

2. **Google Drive:** A centralized repository for all project documents and resources.

* Our main folder: [Google Drive Folder](https://drive.google.com/drive/folders/1pLJ4CszaHxmC894yhK7P7foqmiB1K859?usp=sharing) (commentor permissions)

3. **Zoom:** For meetings and real-time collaboration.

* Organization-wide meetings every Monday at 10pm EST.
* Work sessions are on Wednesday 5pm-7pm EST and Saturday 12pm-2:30pm EST.
* The link to our Zoom Channel (Password: \*\*\*\*\*)

4. **GitHub:** Used for version control and project management. Even non-coders can use the 'GitHub Projects' Kanban boards to track progress.

**UI/UX Tools**

* [**Figma**](https://www.figma.com/)**:** A collaborative interface design tool used for creating and sharing design prototypes.

**Frontend Tools**

* [**React**](https://reactjs.org/)**:** A JavaScript library for building user interfaces.
* [**Git**](https://git-scm.com/)**:** Version control system. Use Git Bash or a similar tool.
* [**VS Code**](https://code.visualstudio.com/)**:** Recommended IDE, though you can use any IDE of your choice.
* [**AWS SDK**](https://aws.amazon.com/sdk-for-javascript/)**:** Used for integrating frontend applications with AWS services.

**Backend Tools**

* [**Node.js**](https://nodejs.org/)**:** A JavaScript runtime for building scalable network applications.
* [**Git**](https://git-scm.com/)**:** Version control system. Use Git Bash or a similar tool.
* [**VS Code**](https://code.visualstudio.com/)**:** Recommended IDE, though you can use any IDE of your choice.
* [**AWS CLI**](https://aws.amazon.com/cli/)**:** Command Line Interface for managing AWS services.

**Security Measures**

Before you start working on our project, please ensure your computer is secure by following these steps:

* **Antivirus Software:** Run a modern antivirus software like [Malwarebytes](https://www.malwarebytes.com/trial) which offers a free 14-day trial. This can be used multiple times to maintain premium protection for free.
* **VPN Software:** If you use public or shared Wi-Fi (e.g., dorms, houses with roommates), consider using a VPN software like [Nord VPN](https://nordvpn.com/) to secure your internet connection.

**2. Feature Naming and Documentation**

**Feature Naming**

* **Major Features:** Sequentially numbered starting from 0.
  + [Requirements Document](https://docs.google.com/document/d/1X0v3Y5NVqMfQ5AXhHxzafGaQFl0bKRdc/edit?usp=sharing&ouid=101066339677844044142&rtpof=true&sd=true)
  + [Feature Specifications Folder](https://drive.google.com/drive/folders/1QHiw5kNTqz4eRE71OY3MUq7sp5Z3h_j6?usp=sharing) containing deep dive documents on each individual feature.
* **Sub-Features:** Indicated with decimals (e.g., Feature 1: Dashboard; Sub-feature 1.1: Widget Adjustment).

**Documentation**

* **Requirements Document:** Each feature is briefly described, followed by detailed UI/UX, Frontend, and Backend requirements.
* **Deep Dive Documents:** Links to Google Drive documents for each feature provide comprehensive details, accessible for in-depth understanding and contributions.

**3. GitHub Collaboration**

**Project Management**

* **Kanban Boards:** Utilize GitHub Projects with Kanban boards tailored to UI/UX, Frontend, and Backend teams. This visualizes tasks and tracks progress across different development stages.

**Ticket Management**

* **Ticket Naming:** Format: [Keyword]: Feature [Number]: [Brief Description] (Example: "Develop: Feature 0: Login/Register Page").
* **Ticket Structure:** Each ticket should include:
  1. A detailed description and acceptance criteria.
  2. Necessary links to documents or resources.
  3. Steps required to complete the ticket.

**Branch Naming Convention**

* **Branch Naming:** Branches created for tickets should follow the format: devname\_featurename\_version (e.g., ryan\_dashboard\_V1).

**4. Development and Review Process**

**Code Development**

* **Commenting:** Extensively comment on all changes or additions to the codebase.
* **Screenshots:** Attach screenshots demonstrating the functionality or completed tests.
* **Supporting Documentation:** Include any additional documents that support the ticket.

**Code Review and Approval**

* **Review Process:** Tickets moved to 'in-review' will be assessed asynchronously by senior team members.
* **Feedback Loop:** If revisions are required, tickets are returned to 'in-progress' with detailed feedback; otherwise, they are moved to 'done' and merged into the main branch.

**5. Deadlines and Timelines**

* **Task Completion:** Aim to complete minor tasks within one week.
* **Review Turnaround:** Strive for a review turnaround time of one week to maintain project momentum.

**6. Training and Onboarding for Interns**

**Initial Training**

* Conduct detailed sessions to walk through the tools and processes mentioned in this SOP.

**Onboarding Sessions**

* Assign mentors to each intern to guide them through their first tasks and ensure compliance with SOP standards.

**Conclusion**

This SOP aims to cultivate a disciplined, efficient, and collaborative environment across development teams. Regular adherence to these procedures is essential for maintaining high standards of work and continuous project advancement.

**Implementation of the SOP**

**Distribution**

* Distribute the SOP during onboarding and make it accessible on the company’s internal knowledge base.

**Review and Updates**

* Periodically review and refine the SOP to integrate new technologies, methods, and team feedback