

# **Computer Science and Engineering**

---

## **CREATO**

## **Project Management Plan**

**Version 1.0**

Document Number: SPMP-001

Project Team Number: A28

Project Team Members (name and NET\_ID):  
Jason Park (swp274)  
Ben Ju (bj755)  
Danny Kim (hmk353)  
Muriel Wu (yw2966)

## REVIEW AND APPROVALS

Printed Name and Title	Function (Author, Reviewer, Approval)	Date	Signature
Professor Strauss	Approval	11/6/20	On File
Jason Park	Author	11/6/20	On File
Ben Ju	Author	11/6/20	On File
Danny Kim	Author	11/6/20	On File
Muriel Wu	Author	11/6/20	On File

**REVISION LEVEL**

<b>Date</b>	<b>Revision Number</b>	<b>Purpose</b>
Fall 2020	Version 1.0	Initial Release

## Table Of Contents

<b>1. OVERVIEW.....</b>	<b>5</b>
1.1 PROJECT SUMMARY .....	5
1.2 PURPOSE, SCOPE, AND OBJECTIVES.....	5
1.3 ASSUMPTIONS AND CONSTRAINTS.....	5
1.4 PROJECT DELIVERABLES.....	6
1.5 SCHEDULE AND BUDGET SUMMARY .....	6
1.6 EVOLUTION OF THE PLAN .....	6
<b>2 REFERENCES .....</b>	<b>6</b>
<b>3 DEFINITIONS.....</b>	<b>6</b>
<b>4 PROJECT ORGANIZATION.....</b>	<b>6</b>
4.1 EXTERNAL INTERFACES.....	6
4.2 INTERNAL STRUCTURE.....	7
4.3 ROLES AND RESPONSIBILITIES .....	7
<b>5 MANAGEMENT PROCESSES .....</b>	<b>7</b>
5.1 START-UP PLAN.....	7
5.1.1 Estimation Plan .....	7
5.1.2 Staffing Plan .....	8
5.1.3 Resource Acquisition Plan.....	8
5.1.4 Training Plan.....	8
5.2 WORK PLAN.....	8
5.2.1 Work Activities.....	9
5.2.2 Schedule Allocation.....	9
5.2.3 Resource Allocation.....	9
5.2.4 Budget Allocation .....	10
5.3 CONTROL PLAN.....	10
5.3.1 Requirement Control and Traceability .....	10
5.3.2 Schedule Tacking and Adjustment .....	10
5.3.3 Budget Tracking and Adjustment.....	11
5.3.4 Quality Control.....	11
5.3.5 Reporting Mechanisms .....	11
5.3.6 Metrics Collection Plan.....	11
5.4 RISK MANAGEMENT PLAN .....	12
5.5 POST IMPLEMENTATION PLAN.....	12
<b>6 TECHNICAL PROCESSES.....</b>	<b>12</b>
6.1 PROCESS MODEL.....	12
6.2 METHODS, TOOLS, AND TECHNIQUES .....	13
6.3 INFRASTRUCTURE PLAN .....	13
6.4 PRODUCT ACCEPTANCE AND MIGRATION PLAN .....	13
<b>7 SUPPORTING PROCESSES PLANS.....</b>	<b>13</b>
7.1 CONFIGURATION MANAGEMENT PLAN.....	14
7.2 QUALIFICATION (VERIFICATION AND VALIDATION) PLAN .....	14
7.3 DOCUMENTATION (LIBRARY) PLAN.....	14
7.4 QUALITY ASSURANCE PLAN.....	15
7.5 REVIEWS AND AUDITS .....	15
7.6 PROBLEM RESOLUTION PLANS.....	15
7.7 ENVIRONMENT MANAGEMENT PLANS .....	15
7.8 PROCESS IMPROVEMENT PLAN .....	16
<b>8. ADDITIONAL PLANS .....</b>	<b>16</b>

---

<b>9</b>	<b>INDEX .....</b>	<b>16</b>
<b>10</b>	<b>RATIONALE .....</b>	<b>16</b>
<b>11</b>	<b>NOTES.....</b>	<b>16</b>
<b>12</b>	<b>APPENDICES.....</b>	<b>17</b>
12.1	SCHEDULE TRACKING .....	18
12.2	DEFECT TRACKING .....	20
12.3	GANTT CHART/MICROSOFT PROJECT SCHEDULE .....	21

# 1. OVERVIEW

---

## 1.1 Project Summary

Creato is platform where users can not only support their favorite creators but also make profit by trading tokens, which represent the values of the creators.

The purpose of this document is to provide the project management plan for our product, Creato. It will contain a detailed outline of how we will implement our project plans. In addition, this document will also describe technical and management processes.

This document is intended to be viewed by the team members and Professor Strauss.

---

## 1.2 Purpose, Scope, and Objectives

Digital media is on a steep rise each year, and viewers are spending time more on digital content than the traditional media. YouTube, a major digital media content platform, allows users to subscribe to their favorite channels to be able to receive instant notifications whenever the channel uploads new content. Such system started to create fandom among viewers. Therefore, numerous YouTubers have started creating new business with their brand. This means that each YouTube channel can also have its own value that could potentially be measurable. If there is a digitized value of a brand, then we will be able to convert it into tradable assets. Fandom can be a strong influence for attracting buyers. Moreover, viewers can also hope to gain some profit whenever their YouTuber also receives more value.

The purpose of this service is to provide users a way to purchase tokens of their favorite Creators in a way that is similar to how people purchase stocks of their favorite companies using a WTS. The goal of this service is to constantly list new tokens of new Creators and allow customers to have a more variety of tokens to purchase. Also, customers will be able to receive dividends based on the amount of tokens they hold, which will become a good incentive to use our service. We plan to incrementally deliver features based on the client's requirements and expectations flexibly over time. The initial MVP is scheduled to be delivered in May 2021.

---

### **1.3 Assumptions and Constraints**

This service will be created under the assumption that it is usable, efficient, dependable, and scalable. The constraints for the implementation of this project is that the deadlines in this document are met, and the service becomes available to the users as swiftly as possible.

---

## 1.4 Project

- The Product SRS and SPMP will be delivered on Winter 2020.
  - The development of the system will begin in January of 2021, and the MVP is planned to be delivered by beginning of Q2 2021.
  - Further revisions and hotfixes may continue up to end of Q2 2021.
- 

## 1.5 Schedule and

The service will be completed after the following stages:

- Requirements Elicitation Stage
  - Design Stage
  - Implementation Stage
  - Quality Testing Stage
- 

## 1.6 Evolution of the

Any scheduled or unscheduled updates will take place upon the approval of the team members. If a member suggests an update to the initial plan, then the team will review the suggestion together and decide democratically.

---

## 2 REFERENCES

---

Please refer to Team A28 Project Proposal and Team A28 System Requirements Specifications(SRS).

---

## 3 DEFINITIONS

The following terms will be frequently used in this document.

Creators: Youtubers or streamers who participate in creating digital contents.

Viewers: People who view digital contents created by such creators, and users who are potential customers of our service.

---

## 4 PROJECT ORGANIZATION

---



## 4.1 External

Meetings will be held between the software development team and the customers during the requirements elicitation stage and after the quality testing stage.

---

## 4.2 Internal Structure

All four members of the team will be responsible for the software development. Initially, there will only be one main development team that focuses on MVP development and further Quality Assurance. From then on, there will be additional groups such as Devops, Data Engineering, and Growth devs. As the project grow in complexity, additional teams such as Financial Analyst team, Legal team will be added.

---

## 4.3 Roles and Responsibilities

Jason Park and Muriel Wu will be responsible for the front-end development of the software, and Ben Ju and Danny Kim will be responsible for the back-end development. However, the above responsibilities can be shared or interchanged if necessary.

---

---

# 5 MANAGEMENT PROCESSES

---

## 5.1 Start-Up Plan

---

### 5.1.1 *Estimation Plan*

Optional

---

**5.1.2 Staffing Plan**Optional

---

**5.1.3 Resource Acquisition Plan**Optional

---

**5.1.4 Training Plan**

Since the front-end team(Jason Park and Muriel Wu) and the back-end team(Ben Ju and Danny Kim) have proficiency in the programming languages necessary to develop the software, training plan for development is not necessary.

Training for financial regulation and information security may take in place later in the development process.

---

**5.2 Work Plan**

---

**5.2.1 Work Activities**

The work activities for this software are as follows:

1. Elicit user requirements by meeting with customers
  2. Write up the SRS document
  3. Write up the SPMP document
  4. Design and implement the software
  5. Perform quality testing
  6. Release the software to the customers
- 

**5.2.2 Schedule Allocation** Please refer to the Gantt Chart in Section 12.3.

---

**5.2.3 Resource Allocation** The team will devote two hours to eliciting user requirements through discussions with customers. Communication skills will be necessary for this task. The team will devote a total of 16 hours to writing up the SRS document and the SPMP document. The team will devote 50 hours in total to designing and implementing the software, and coding skills and knowledge in data structures and algorithms will be necessary for this task. The team will devote 10 hours to quality testing and 5 hours to releasing the completed version of the software.

**5.2.4 Budget Allocation**      Optional

---

**5.3 Control Plan**

---

**5.3.1 Requirement  
Control and Traceability**

A change log containing the history of all updates made would be the control mechanism for measuring, reporting, and controlling changes to the product requirements. By referencing the change log, the team will be able to visualize what has changed in a certain time and discussions can be made regarding what still needs to be changed.

---

**5.3.2 Schedule Tacking  
and Adjustment**

Github, Slack, and Google Calendar will be the control mechanisms used to measure the progress of work completed. Github will be used to save the team's progress, Google Calendar will be used to assign and keep track of each member's work schedule, and Slack will be used for communication of the team's overall work schedule.

---

**5.3.3 Budget Tracking  
and Adjustment**

Optional

---

**5.3.4 Quality Control**

Quality control of the work processes and resulting work products will be based on user feedback and continuous assessments on the software's functional requirements.

---

**5.3.5 Reporting  
Mechanisms**

Optional

---

**5.3.6 Metrics Collection  
Plan**

The metrics to be collected are source code metrics, development metrics, and testing metrics. The quality of the source code that makes up the software will be measured by counting the number of lines and number of comments to measure the complexity and flexibility of the code. Development metrics will be the measurement of defects or errors within the software, and the source code will be scrutinized for collecting development metrics. Finally, functionality assessments will be performed to collect testing metrics. The metrics will be collected at least once every 10 hours of scheduled work during design & implementation and testing stages.

---

## 5.4 Risk Management Plan

Since this is a platform that deals with financial assets, there may be attempts for money laundering. To prevent such exploitation, we must have Anti-Money Laundering process ready in our platform. Another risk is that there may be malicious trading efforts to somehow increase a tokens' worth to make profit. Hence, there must be some sort of Fraud Detection Monitoring for all the transactions that occur in our platform. In order to enable on-going identification, assessment, and mitigation of risks, the above methods should be available at all times.

---

## 5.5 Post Implementation Plan

Optional

---

# 6 TECHNICAL PROCESSES

---

## 6.1 Process Model

This project will follow a Waterfall model.

There will be three distinct stages in this process:

1. Create the monolithic server with the application layer and the data layer. User, Trading, and Subscription databases should be set up. Initial test data should be injected into the database. Setting up each databases will be the milestones. The front-end team will meanwhile set up web designs in parallel.
2. Set up APIs in the server. After the APIs are set up by the back-end team, the API documentation will be reviewed by the front-end team and discuss whether it is valid. API integration, API documentation will be the milestones at this stage.
3. Create the front-end interface for the system. Meanwhile, the back-end will be conducting further stress tests and integration tests. Front-end development, unit-testing, UX QA will be the milestones for this stage

**6.2 Methods, Tools, and Techniques**

Back-end development of the software will be done using Python, and MySQL will be used as the database platform. Front-end development of the software will be done using HTML, CSS, and Javascript with React.

---

**6.3 Infrastructure Plan**

The team will utilize PyCharm as the IDE for the source code of the software, and the software will be developed on a machine with Windows operating system. Compiling and testing the source code on PyCharm will be the main procedure for analysis and testing of the source code.

Logs will be recorded in AWS Cloudwatch and Sentry.

---

**6.4 Product Acceptance and Migration Plan**

Optional

---

Optional

---

**7 SUPPORTING PROCESSES PLANS**

---



### **7.1 Configuration Management Plan**

The methods used for configuration management would be to collect configuration data and store it in a configuration database. The configuration data will be linked to any issues and changes, and the accuracy of the data will be verified through audits and investigation.

---

### **7.2 Qualification (Verification and Validation) Plan**

Pair programming and peer review will be the methods for qualification verification of the project. Since the team is divided into two subgroups, one that handles the front-end aspect and the other that handles the back-end aspect, each subgroup will work together in a pair, reviewing each other's work. This method will reduce the occurrences of defects and errors during the software development process.

---

### **7.3 Documentation (library) Plan**

The documents involved with the development of this software will be generate by all members of the team, and they will be reviewed and approved by Professor Strauss. The date for the delivery of initial baseline version would be in May 2021.

#### **7.4 Quality Assurance Plan**

The source code will be analyzed and inspected for quality assurance, and as the software is being developed, functionality assessment will be performed for each functionality requirement of the software. This will be done in accordance with the team's progress.

---

#### **7.5 Reviews and Audits**

Peer reviews will be constantly performed as the back-end subgroup and front-end subgroup work together, and user reviews will be collected once the software nears final release. Weekly audits will be performed by the group as a whole.

---

#### **7.6 Problem Resolution Plans**

Since the team will make use of Github for reviewing and revising the source code, if there are any defects or errors, all the members will access the source code via Github and work together to solve the problems. Errors will be categorized according to the level of severity, and the most severe errors will be prioritized for analysis.

---

#### **7.7 Environment Management Plans**

Optional

---

## 7.8 Process Improvement Plan

Optional

---

## 8. ADDITIONAL PLANS

---

---

## 9 INDEX

---

Self Explanatory

---

## 10 RATIONALE

---

None

---

## 11 NOTES

---

None

---

## 12 APPENDICES

---

## 12.1 Schedule Tracking

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SRS – Business Domain	Jason Park	5 hours	5 hours	0 hours
	Ben Ju	5 hours	5 hours	0 hours
	Danny Kim	5 hours	5 hours	0 hours
	Muriel Wu	5 hours	5 hours	0 hours
	Summary for entire team	20 hours	20 hours	0 hours

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SRS – Requirements	Jason Park	5 hours	5 hours	0 hours
	Ben Ju	5 hours	5 hours	0 hours
	Danny Kim	5 hours	5 hours	0 hours
	Muriel Wu	5 hours	5 hours	0 hours
	Summary for entire team	20 hours	20 hours	0 hours

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SRS – Analysis - Complete	Jason Park	5 hours	5 hours	0 hours
	Ben Ju	5 hours	5 hours	0 hours
	Danny Kim	5 hours	5 hours	0 hours
	Muriel Wu	5 hours	5 hours	0 hours
	Summary for entire team	20 hours	20 hours	0 hours

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SPMP	Jason Park	10 hours	10 hours	0 hours
	Ben Ju	10 hours	10 hours	0 hours
	Danny Kim	10 hours	10 hours	0 hours
	Muriel Wu	10 hours	10 hours	0 hours
	Summary for entire team	40 hours	40 hours	0 hours

---

**Cumulative**

Who (individual and team)	Estimated	Actual	Difference
Jason Park	25 hours	25 hours	0 hours
Ben Ju	25 hours	25 hours	0 hours
Danny Kim	25 hours	25 hours	0 hours
Muriel Wu	25 hours	25 hours	0 hours
Summary for entire team	100 hours	100 hours	0 hours

## 12.2 Defect Tracking

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SRS – Business Domain	Jason Park	1 per page	1 per page	0 per page
	Ben Ju	1 per page	1 per page	0 per page
	Danny Kim	1 per page	1 per page	0 per page
	Muriel Wu	1 per page	1 per page	0 per page
	Summary for entire team	4 per page	4 per page	0 per page

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SRS - Requirements	Jason Park	1 per page	1 per page	0 per page
	Ben Ju	1 per page	1 per page	0 per page
	Danny Kim	1 per page	1 per page	0 per page
	Muriel Wu	1 per page	1 per page	0 per page
	Summary for entire team	4 per page	4 per page	0 per page

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SRS – Analysis - Complete	Jason Park	1 per page	1 per page	0 per page
	Ben Ju	1 per page	1 per page	0 per page
	Danny Kim	1 per page	1 per page	0 per page
	Muriel Wu	1 per page	1 per page	0 per page
	Summary for entire team	4 per page	4 per page	0 per page



Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SPMP	Jason Park	2 per page	1 per page	1 per page
	Ben Ju	2 per page	1 per page	1 per page
	Danny Kim	2 per page	1 per page	1 per page
	Muriel Wu	2 per page	1 per page	1 per page
	Summary for entire team	8 per page	4 per page	1 per page

**Cumulative**

Who (individual and team)	Estimated	Actual	Difference
Jason Park	5 per page	4 per page	1 per page
Ben Ju	5 per page	4 per page	1 per page
Danny Kim	5 per page	4 per page	1 per page
Muriel Wu	5 per page	4 per page	1 per page
Summary for entire team	20 per page	16 per page	4 per page

**12.3 Gantt Chart/Microsoft Project Schedule**