CASHTRACK

CONFIGURATION MANAGEMENT PLAN

Version 1.2

24th March 2021

By:
Team Runtime Terror
Ravishankar Amrita (U1822377F)
Datta Anusha (U1822948G)
Kumar Mehul (U1822146E)
Alex Leong (U1921599D)
S Sri Kalki (U1921575L)
Nicklaus Tan (1403385F)
Elliott Ong (U1922981C)
Daniel Loe (U1921408A)

VERSION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Alex Leong, Ravishankar Amrita, Anusha, Daniel Loe, Elliott Ong, S Sri Kalki, Kumar Mehul	14th March 2021	Nicklaus	16th March 2021	Initial Software Configuration Management Plan draft
1.1	Alex Leong	19th March 2021	Mehul	20th March 2021	Revision in Organisation section
1.2	Alex Leong, Ravishankar Amrita, Anusha, Daniel Loe, Elliott Ong, S Sri Kalki, Kumar Mehul	24st March 2021	Nicklaus	24nd March 2021	Final changes & revision for all sections

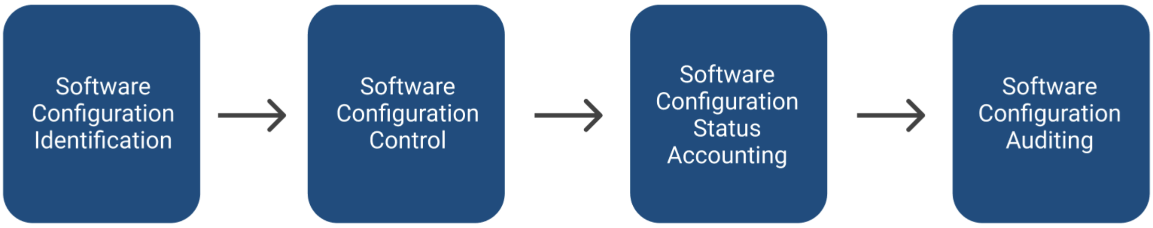
TABLE OF CONTENTS

1 Identification

1.1 Introduction

As a part of the quality control process of the project, the plan of configuration management aims to establish, maintain and ensure the integrity of the project throughout the entire Software Development Life Cycle (SDLC).

The configuration management encompasses identifying and controlling the configuration items and changes for the project which is accomplished by recording the statuses of such configuration items and reporting their changes in activity statuses.


Software Configuration Identification
Software Configuration Control
Software Configuration Status Accounting
Software Configuration Auditing

1.2 Objectives

The core objectives of Software Configuration Management for the CashTrack project are:

- **Delivery**

Ensures that the correct version of the software is delivered.

- **Traceability**

Ensures that the software being produced, Cash Track, meets the stakeholders' requirements.

- **System Integrity**

Ensures the consistency and stability of the software and its supporting materials throughout the Software Development Lifecycle.

1.3 Document Overview

This document contains the software configuration management plan of software (SCMP) of the software application Cash Track, developed by the team Runtime Terror.

The table below lists the sections that will be documented in this document as well as the contents within each section:

#	Section Name	Section Description
1	Identification	Discussion of the abbreviations and technical terms used in the project Descriptions of such terms are elaborated in more details below

2	Organization	Describes how the team will be organised to perform configuration management. The duties and jobs of each team member are listed as well.
3	Configuration Identification	Describes how the Software Configuration Items will be identified.
4	Configuration Control	The changes in configuration management process in the Software Configuration Items in Cash Track will be described here.
5	Configuration Support Activities	The activities which evaluate the degree of compliance towards the procedure and standards in implementing changes to the Software Configuration Items (established within the Software Configuration Management Plan) as demonstrated by the development team will be described here.

1.4 Abbreviations and Glossary

1.4.1 Abbreviations

Abbreviation	**Description**
SCMP	Software Configuration Management Plan
SCM	Software Configuration Management
SCI	Software Configuration Item
SCR	Software Change Request
SRS	System Requirements Specification
SDLC	Software Design Life Cycle
VDD	Version Description Document

1.4.2 Glossary

Glossary Term	**Description**
Software Configuration Management (SCM)	The discipline of identifying the configuration of a system for controlling changes to this configuration as well as maintaining the integrity and traceability throughout the software Development Life Cycle.
Software Configuration Management Plan (SCMP)	A reference document for the SCM process that includes the work breakdown structure and description of each section of each of the respective processes.

Software Configuration Item (SCI)	Any service component, infrastructure element, or other item that needs to be managed in order to ensure the successful delivery of services.
Software Change Request (SCR)	Software change request is a tool used for requesting, approving, and documenting changes to the project. A change request is a declarative document, stating what needs should be accomplished while leaving out how changes should be carried out.

1.5 References

1.5.1 Project References

#	Document Identifier	Document Title
1	[R1]	Annual Survey on Infocomm Usage in Households and by Individuals for 2019. Singapore: Infocomm Media Development Authority, p.10. Available at: https://www.imda.gov.sg/-/media/Imda/Files/Infocomm-Media-Landscape/Research-and-Statistics/Survey-Report/2019-HH-Public-Report_09032020.pdf?la=en
2	[R2]	A.B., M.-C.N., P.B., G.T., R.C., & C.T. (2020). The 2020 McKinsey Global Payments Report. McKinsey & Company. https://www.mckinsey.com/~/media/mckinsey/industries/financial%20services/our%20insights/accelerating%20winds%20of%20change%20in%20global%20payments/2020-mckinsey-global-payments-report-vf.pdf
3	[R3]	Clymo, R. (2020, May 12). More than 6.1 billion people will use digital payments by 2023. TechRadar. https://www.techradar.com/sg/news/more-than-61-billion-people-will-use-digital-payments-by-2023
4	[R4]	Survey conducted by CashTrack! Team https://docs.google.com/spreadsheets/d/1aMxRc6aY7G7RhIqGwnpnQmCOoYjvbHSfg7pkqz1Sh5Y/edit?usp=drive_web&ouid=115403635418018121990

1.3.2 Standard and regulatory References

CashTrack will be storing multiple users personal expense records and shared expense records data between friends, therefore information security is of utmost importance to protect all users' data. Hence, to do so, Team Runtime Terror shall be adopting the International Organization for Standardization ISO/IEC 27000 series of information technology security techniques, to protect the privacy of all users and their data.

#	Document Identifier	Document Title
1	[STD1]	IEEE Standard for Software Configuration management Plans - IEEE Std 828 -1998
2	[STD2]	ISO/IEC 27000:2018 Information technology — Security techniques — Information security management systems — Overview and vocabulary. (2019, February 4). International

		Organization for Standardization. https://www.iso.org/standard/73906.html https://standards.iso.org/ittf/PubliclyAvailableStandards/c073906_ISO_IEC_27000_2018_E.zip
3	[STD3]	ISO/IEC 27001:2013 Information technology — Security techniques — Information security management systems — Requirements. (2020, December 16). International Organization for Standardization. https://www.iso.org/standard/54534.html
4	[STD4]	ISO/IEC 27002:2013 Information technology — Security techniques — Code of practice for information security controls. (2020, December 16). International Organization for Standardization. https://www.iso.org/standard/54533.html
5	[STD5]	ISO/IEC 27003:2017 Information technology — Security techniques — Information security management systems — Guidance. (2017, March). International Organization for Standardization. https://www.iso.org/standard/63417.html
6	[STD6]	ISO/IEC 27004:2016 Information technology — Security techniques — Information security management — Monitoring, measurement, analysis and evaluation. (2016, December 16). International Organization for Standardization. https://www.iso.org/standard/64120.html
7	[STD7]	ISO/IEC 27005:2018 Information technology — Security techniques — Information security risk management. (2020, December 16). International Organization for Standardization. https://www.iso.org/standard/75281.html
8	[STD8]	ISO/IEC 27032:2012 Information technology — Security techniques — Guidelines for cybersecurity. (2012, July). International Organization for Standardization. https://www.iso.org/standard/44375.html
9	[STD9]	ISO/IEC 27040:2015 Information technology — Security techniques — Storage security. (2015, January 13). International Organization for Standardization. https://www.iso.org/standard/44404.html
10	[STD10]	ISO/IEC 27701:2019 Security techniques — Extension to ISO/IEC 27001 and ISO/IEC 27002 for privacy information management — Requirements and guidelines. (2020, October 2). International Organization for Standardization. https://www.iso.org/standard/71670.html

2 Organization

Team Runtime Terror will be managing the software application CashTrack. The tasks of Configuration Management will be shared between the Project Manager, Release Engineer as well as the Quality Assurance Manager & Engineers.

The development team will be using Tortoise SVN and Github for source code management and version control as they are able to record the history of changes and who was responsible for the changes, this allows the team to traceback easily in the event there is a bug / problem found in one of the releases. JIRA will also be used to keep track of CashTrack product development, allowing the development team to keep track of bugs and agile project management

2.1 Software version control and management tools

The tables below shows the activities as well as who will be using these management tools:

Software Tools	Users & Activity
Tortoise SVN	**Release Engineer:** Upload functioning versions of CashTrack as version archive. Upload version documentations. **Project Manager:** View the different release versions of CashTrack.
Github	**Front-end & Back-end developers:** Push and Pull source codes of CashTrack for modifications. **Release Engineer:** Create branches for each release / development phase for version control. These branches will also help to narrow down the location of problematic components and bugs. **QA Manager & Engineers:** Pull codes from branches to carry out quality assurance testing
JIRA	**Project Manager:** Track software development progress and version release status of CashTrack. **Release Engineer:** Track version release status of CashTrack development. **Front-end & Back-end developers:** Update release status

2.2 Activities and responsibilities

The tables below shows the activities as well as who will be responsible for them:

Activities when setting up the project	**Person responsible**
Identification of Software Configuration Items (SCI’s)	Quality Assurance Manager & Engineer
Install the bug repository tool and set up the database	Quality Assurance Manager & Engineer
Install the software configuration repository tool and set up the database	Project Manager
Setup of documentation archives and repository	Release Engineer
Define the configuration processes	Project manager & Release Engineer

Activities during the project lifecycle	**Person responsible**
Export components for modification, test or delivery	Quality Assurance Manager & Engineer
Create version, write version delivery document	Release Engineer

Approve reference configurations	Project manager
Verify version to be delivered and authorise deliveries	Project manager
Creating Backup Spaces	Release Engineer
Do configuration audits	Quality Assurance Manager
Inspect configuration records	Quality Assurance Manager
Monitor configuration records	Quality Assurance Manager & Engineer

Management activities	**Person responsible**
Manage versions and archives for software	Release Engineer
Manage configuration records	Release Engineer
Manage archives for documents	Release Engineer
Manage spaces backup and archive media	Release Engineer
Manage quality reports	Quality Assurance Manager & Engineer

2.2.1 Decisions process and responsibilities

The list of decision processes and responsibilities that will be made through the Configuration Management Plane are listed in the table below:

At the end of an activity of the project

Activities	**Person Responsible**
Perform a configuration freeze	Release Engineer
Present a configuration state of the components impacted by the activity	Release Engineer
Present a documentation state of the components impacted by the activity	Release Engineer

During a configuration management process audit:

Activities	**Person Responsible**
Perform the configuration management process audit	Project Manager
Present the records of the configuration management process	Release Engineer
Present the quality records of the configuration management process	Quality Assurance Manager & Engineer
Present the records of the documentation management process	Release Engineer

3 Configuration Identification

The purpose of Configuration Identification will determine the way software configuration items will be created and identified throughout the Software Development Life cycle (SDLC) of the CashTrack project.

3.1 Identification Rules of Configuration Item

The pre-defined format for identifying Software Configuration Items (SCIs) shall be employed for the scope of the entire SDLC of the CashTrack project. The key SCIs expected to be identified within the product are as follows:

- Angular Frontend Source Code (HTML, CSS and TypeScript Files)
- ExpressJS Backend Source Code (Javascript Files)
- Github (Version Control using Git)
- MongoDB Cloud Database Records (NoSQL Database)
- Documentation (README Files, APIDoc, Architecture Design Doc)
- Node Modules (Dependencies for front-end and back-end)

The format of nomenclature for these items shall be as follows:

[File Name]_[Version Number].[File Format]

Moreover, based on front-end functions and back-end services created during the development phase of the SDLC, these items shall be further organized into different folders directories, which shall intrinsically facilitate configuration item identification.

3.1.1 Version number of a Configuration Item

The attribution of a version number is a prerequisite to any delivery of any configuration item. This number shall be incremented before a new delivery, if the product or its documentation were modified.

The assignment of unique version numbers will reflect the type of changes made, and their corresponding severity. For each major, minor and revision change, the version number assigned will increase by 1.0, 0.1 and 0.001 respectively. This is further elaborated below:

Major	- Additions and changes made to the main features of the Software Application.
Minor	- Bug fixes to main features of the Software Application. - Additions and changes made to the minor features of the Software Application. - Increments of the Minor version number will be reset after each Major version increment.
Revision	- Bug fixes, additions and changes made to the minor features of the Software Application - Increments of the Revision version number will be reset after each minor version increment.

3.2 Identification Rules of Documents

3.2.1 Description of Documents Identifiers

The purposes the CashTrack documents serve throughout the development life cycle of the project and their consequent revision histories shall determine the format for identifying documents. On that note, the comprehensive list of documents associated with the CashTrack project are listed as follows:

1.	Project Proposal
2.	System Requirement Specification (SRS)
3.	Quality Plan
4.	Project plan
5.	Risk Management Plan
6.	Design Report On Software Maintainability
7.	Configuration Management Plan
8.	Change Management Plan
9.	Use Case Model
10.	Class Diagram
11.	Sequence Diagram
12.	Dialog Map Diagram
13.	Database Design
14.	Developer test cases
15.	Test Plan
16.	Release plan
17.	Requirement Test Coverage Report

The document name shall be indicative of the document type, while the version number shall encapsulate the revision history of the document. This may be observed through the pre-defined document nomenclature format below:

[Document Name]_[Version Number].[File Format]

Further, these documents shall be carefully organised into their respective folder directories based on their purposes throughout the SDLC, which shall further facilitate the document identification process.

3.2.2 Version Number of Documents

The assignment of unique version numbers will reflect the severity of the changes made. For each major and minor change, the version number assigned will increase by 1.0 and 0.1 respectively. This is further defined below:

Major	- Significant content modification made to the documentation. These include document content alterations in at least one section. - Must be approved by the Project Manager.
Minor	- Insignificant content modification made to the documentation. These include formatting and grammatical corrections. - Need not be approved by the Project Manager. - Increments in the Revision version number will be reset after each major version number increment.

3.3 Reference configuration Identification

Document Name	Version Number
Project Proposal	1.0
Use Case Description	1.3
System Requirement Specification	1.2
Quality Plan	1.5
Project Plan	1.3
Risk Management Plan	1.3
Configuration Management Plan	1.2
Design Report On Software Maintainability	1.3
Change Management Plan	1.3
Release Plan	1.3

3.4 Configuration Baseline Management

Multiple baselines have been defined for the CashTrack project to ensure proper management and project control throughout the lifecycle of the project.

The purpose of a Configuration Baseline is to serve as a fixed reference throughout the Software Development Lifecycle Cycle (SDLC). It will be documented on the basis of defining change control for various configuration items. It will be considered as a baseline for all the changes that follow.

These baselines is defined as follows:

Baseline	Baseline Description	Period of Establishment	Components
Functional	Defines the functionality requirements based on the system specifications. It will document the system's functionality and capabilities at the minimum benchmark.	During Project Planning phrase and after Deployment & Maintenance Phase	1. Project Proposal 2. System Requirements Specification (SRS) 3. Quality Plan 4. Project Plan 5. Risk Management Plan
Allocated	Defines the design of the functional and interface characteristics that composes the system.	After Design Phase	1. Use Case Model 2. Class Diagram 3. Sequence Diagram 4. Dialog Map Diagram 5. Database Design 6. User Interface Design
Product	Defines the completed and accepted system components and documentation that identifies these products	After Integration and Testing Phase	1. Source Code 2. Test Cases 3. Developer Test Cases 4. Release Plan 5. Requirement Test Coverage Report

4 Configuration Control

During the Software Development Lifecycle (SDLC), part of the Configuration Management domain is Change Management within the Software Configuration Items. With proper control, the team will have to give consent and acknowledge any changes made to the system to avoid any unsupervised changes.

The steps to Configuration Control are:

- Recognising that a change is necessary
- Discuss the change request submitted
- Approval or Disapproval of the change request
- Verification, Implementation and Release of the change request

Only people external to the development team may propose changes, and all changes must be documented using the Change Request Form (CRF). The Change Request Form must be filled up and submitted to the Change Control Board (CCB) for approval. The form must be reviewed by the Lead Developer and the Quality Assurance Manager & Engineers to carry out impact analysis for the requested changes. Change assessment will also be carried out to set a priority of the change, finding out the cost to implement the changes and the budget of the company to implement these changes.

The Change Request Form must include what the change is, explanations on why it is implemented, and which parts of the code will be affected by the change.

The project manager will have a meeting with the Lead Developer and the Quality Assurance Manager & Engineers to discuss the feasibility of these changes and how they will benefit the software and write down their analysis on the CRF.

When the changes are approved by the CCB, the development team members that are working on the parts that are affected by the changes will insert the changes in a separate branch, in order to minimize the effect on the main branch. The branch with the new code will be tested and be cleared of bugs before being merged with the main branch, in order for the main branch to have the latest code.

4.1 Change Management

Upon change approval from the Change Control Board, the change request will be accepted by the project manager and a branch will be created in the Software Configuration Management System (SCM). Software Configuration Items (SCI) will be “Checked out” of the SCM for changes to be made by the assigned development team. When changes are completed by the assigned development team, the codes will be reviewed and audited to make sure that implementation meets the updated requirement. The new SCI will be “Checked in” into the SCM with the Baseline for testing being established. Finally the QA team will carry out final testings to ensure that the product meets the expected quality before the changes can be merged with the main branch.

After QA testing, if there are no bugs reported, the development team can proceed to merge and integrate the new changes with the original branch and the software version will be modified based on the impact of the change (Major, Minor or Minor Revision). The entire development team will then be updated of the new changes made to the application.

To provide details on the changes, the branch name must identify the part of the software being changed, and proper commit details should be added.

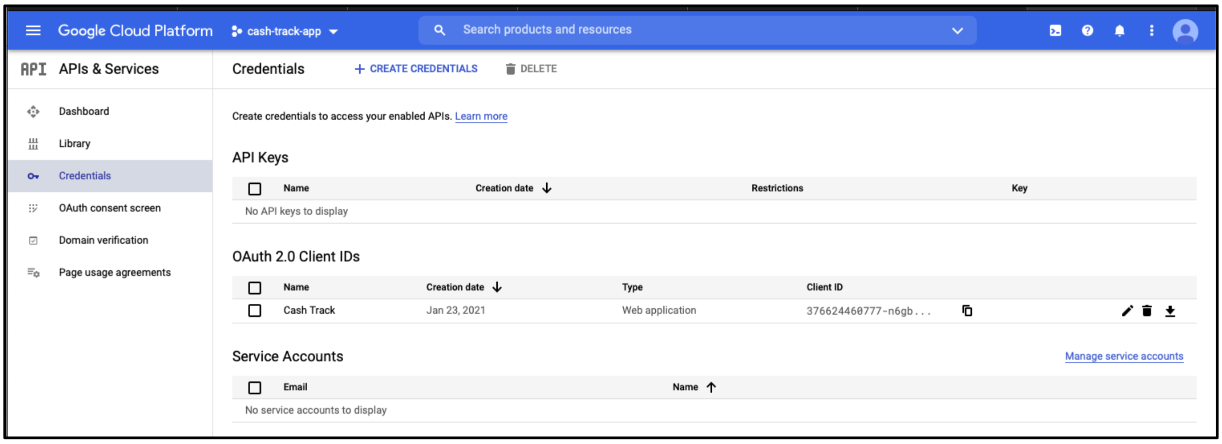
The branch with the new changes must be tested and be cleared of bugs before being merged with the main branch, in order for the main branch to have the latest code.

All commit messages from the new branch must clearly state the changes made and the reason for the modifications. Software version of the main branch will be updated based on the scale of the changes.

4.2 Interface Management

For 3rd party interfaces, CashTrack has integrated Google OAuth for authentication of new and existing users. CashTrack also interfaces with a MongoDB Database Instance where all the data of the web application is stored.

Using the Google OAuth 2.0 service in Google Cloud Platform, the developers of CashTrack have created a unique Client ID to connect the application to Google’s authentication services. This service currently is on Free Tier and has a cap of 100 unique user logins per day, and must be upgraded to Premium Tier when CashTrack is pushed to production. It is very important to ensure that Cashtrack’s unique Client ID and API Key is always active, or the developers are given enough notice to replace the service before it is deprecated.



Credentials

Create credentials to access your enabled APIs. Learn more

API Keys

□	Name	Creation date ↓	Restrictions	Key
	No API keys to display			

OAuth 2.0 Client IDs

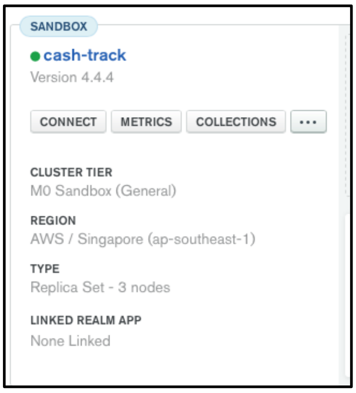
□	Name	Creation date ↓	Type	Client ID
□	Cash Track	Jan 23, 2021	Web application	376624460777-n6gb...

Service Accounts

Manage service accounts

□	Email	Name ↑
	No service accounts to display	

CashTrack is using a MongoDB Cloud Atlas cluster as its database, and the NodeJS Mongo drivers are used to connect the Express Backend to the cloud database. It should be ensured that the cloud database has a low latency and the credentials for accessing the database is not changed without due notice. Currently the cluster is hosted on a Free Tier Sandbox, and must be upgraded to a Premium Tier when the number of read/write operations per day exceeds 5000. In the case of change of database, due notice must be given to all the teams to update the source code in production and all the documentations.



● cash-track
Version 4.4.4

CLUSTER TIER
M0 Sandbox (General)

REGION
AWS / Singapore (ap-southeast-1)

TYPE
Replica Set - 3 nodes

LINKED REALM APP
None Linked

5 Configuration Support Activities

5.1 Configuration Status Accounting

Configuration Status Accounting (CSA) is the process of recording, storing, maintaining, and reporting of Software Configuration Items during the Software Development Life Cycle. During the course of the project, the configuration details must be made current and accessible for preparation, decision-making, and proper change management.

All Software Configuration Items (SCIs) and related documents must be documented in order to ensure traceability.

5.1.1 Evolutions traceability

The traceability of modifications of items given their types:

Software Configuration Item (SCI)	Traceability
Documentation	Version number within the document and the document file name will be stored. Revisions/changes made to the document will update the version number. The document will be saved with a different file name alongside its new version number.
Source Code	Commit ID, Message/Description and Comments
Software Change Request	Request ID
Design Diagrams	Version Number within file name
Test Cases	Version Number within file name
Version Description Document	Version Number within file name
Database Description	Version control that enforces and monitors check-out/check-in operations on the database

5.1.2 Setting up Configuration status

Throughout the development lifecycle of this project, the Lead Developer will take on the role of the Software Configuration Manager and write the Version Description Document.

Being the Software Configuration Manager, the Lead Developer will set up the state of all versions of each Software Configuration Item consisting of the label, version number, and creation date of the Version Description Document.

5.1.3 Configuration status diffusion

Both the Quality Assurance Manager and the Lead Developer will write the Version Description Document. The Project Manager must then vet and approve the document in order for it to be considered valid.

5.1.4 Configuration status records storage

The records of each SCI are stored in a configuration folder, which contains the following:

- Software Change Requests, sorted by Request ID
- Software Documentation
- Version Description Documents
- Configuration States sorted chronologically

5.2 Configuration audits

The majority of the Configuration Audits in this project are baseline audits. These baselines are an agreed description and definition of attributes of the product, CashTrack at a given point in time that serves as a foundation for future changes.

The most recent status of the SCIs is reviewed and tracked using a standardized checklist so as to ensure that each of their performances is in line with the System Requirement Specifications. The Quality Assurance Manager and Engineer will perform formal reviews of test results of the functional configuration of SCIs to verify that CashTrack has met the requirements identified.

The other types of configuration audits used in this project as as follows:

Configuration Audits	**Audit Description**	**Key Personnel Involved**
Functional	Examines the functional characteristics of a configuration item to verify that its functionality is consistent with the functional requirements as specified in the System Requirement Specification	● Project Manager ● Software Configuration Manager ● Developer ● Quality Assurance Manager
Software	Ensures that the software product CashTrack satisfies the baseline needs so that the product built is the product delivered	● Project Manager ● Software Configuration Manager ● Developer ● Stakeholder

5.3 Reviews

The baseline audit will serve as a foundation for changes to be made in the future, however, before the baseline can be utilised, it has to first be properly formulated.

A project management tool that helps to set up these baselines is known as a review. The table below lists the personnel involved and their responsibilities relating to reviews:

Personnel	Responsibilities
Lead Developer	- Select the objectives and Software Configuration Items for review - Provide approval criteria
Quality Assurance Manager & Engineer	- Planning and deciding a schedule for the review based on procedures - Formulate the procedures to document the flaws within the project as well as the solutions to these flaws
Technical Reviewer (An internal assignment within the project team)	- Have the appropriate domain knowledge or subject matter expertise relating to the work product - Provides technical feedback (including possible defects) on the work product
Recorder	- Documents the issues and key points discussed during the review in a short and concise manner

5.4 Configuration management plan maintenance

Throughout the whole Software Development process, the Software Configuration Management plan will be periodically updated. This way, it can be used to help identify the processes that will be used to implement changes to the identified requirement specifications.

The monitoring of these plans is to be done by the Quality Assurance Manager & Engineer. As for the updating of the Software Configuration Management plan, any changes for the document will have to be assessed and approved by the Quality Assurance Manager & Engineer. Every week, the Quality Assurance Manager & Engineer will collate the approved changes and update the Software Configuration Management plan accordingly.

Once approval for the plan has been obtained, the latest plan is then saved with a newer version number. All older plans will be saved in the configurations folder, this folder can be accessed to help improve traceability as well as facilitate changes to the project schedule.

6 Role Assignment

Role(s)	**Name(s)**
Project Manager	Nicklaus Tan
Lead Developer & Acting Software	Kumar Mehul

Configuration Manager	
Front-End Developer	Ravishankar Amrita
Back-End Developer	Datta Anusha
Release Manager	Alex Leong
Quality Assurance Manager	S Sri Kalki
Quality Engineer	Daniel Loe, Elliot Ong