



**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

CZ3002 Advanced Software Engineering Project:

Test Plan - CashTrack

Version 1.3

Team Members:

Ravishankar Amrita (U1822377F)

Datta Anusha (U1822948G)

Kumar Mehul (U1822146E)

Alex Leong (U1921599D)

S Sri Kalki (U1921575L)

Nicklaus Tan (19403385F)

Elliott Ong (U1922981C)

Daniel Loe (U1921408A)

Lab Group: B2

School of Computer Science and Engineering (SCSE)

Revision 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	15th March 2021	Initial Change Management draft
1.1	Elliot Ong	18th March 2021	Kalki	19th March 2021	Revision to Appendixes C, D, E
1.2	Kalki	21st March 2021	Nicklaus	22nd March 2021	Revision in Responsibilities
1.3	Alex Leong, Ravishankar Amrita, Anusha, Daniel Loe, Elliott Ong, S Sri Kalki, Kumar Mehul	24th March 2021	Nicklaus	24th March 2021	Final changes & revision for all sections

Content Page

1 Test Plan Identifier	4
2 Introduction	4
2.1 Purpose	4
2.2 Description and Scope	4
3 Test Items	5
4 Features to be Tested	5
5 Features not to be Tested	7
6 Approach	7
7 Item Pass/Fail Criteria	9
7.1 Unit Test Level	9
8 Suspension Criteria And Resumption	11
9 Test Deliverables	12
9.1 Testing Evaluation Summaries	13
9.2 Incident Logs and Change Requests	13
10 Testing Tasks	14
11 Environmental Needs	16
11.1 Hardware	16
11.2 Software	16
12 Staffing, Responsibilities And Training Needs	17
13 Schedule	19
14 Risks And Contingencies	20
15 Approvals	21

1 Test Plan Identifier

Test Plan Identification Number: CashTrack_MTP_0103

2 Introduction

2.1 Purpose

This document outlines the testing plan with regards to testing the functionality of CashTrack System. It will support the following objectives:

- Identify the project components to be tested
- Outline the high level suitable test requirements
- Identify the deliverables of each testing phrase
- Identify the testing approach to be engaged
- Identify the effort and resources required

2.2 Description and Scope

The scope will describe the system testing after the integration of the subsystems and components. Unit testing will be conducted through black box testing, white box testing as well as boundary test. This will determine the feasibility and performance of the subsystem as well as the system as a whole.

Internal Interfaces to be tested:

1. Register and Login
2. Personal Expenses
3. Shared Expenses
4. Dashboard
5. Invite
6. Group
7. Chat

External Interfaces to be tested:

1. Mongo database
2. Google API

Performance metric to be tested:

1. Response time to login to CashTrack
2. Response time to send invitation

3. Response time to load dashboard (analytics)
4. Response time for user to receive notification regarding new group expense

3 Test Items

The following items define the scope to be test:

Items	Document Version
Functional Requirements	System Requirement Specification Version 1.2
Non-functional Requirements	System Requirement Specification Version 1.2
Process Requirements	System Requirement Specification Version 1.2
Input Requirements	System Requirement Specification Version 1.2
Output Requirements	System Requirement Specification Version 1.2
Use Case Description	Use Case Descriptions Version 1.3
Test cases	Test Cases and Requirements Test Coverage Report v1.0

4 Features to be Tested

Features	Risk Level (L, M, H)
User Authentication	
Login	M
Sign Out	M
Dashboard	
View Summary of Transactions	M
View Spending Pattern Analysis	L
View Personal Spending Limit	L

Personal Expense	
Create New Expense	H
Edit Expense	H
Delete Expense	H
Shared Expense	
Create New Expense	H
Edit Shared Expense	H
Delete Expense	H
View Summary of Transactions	M
Add comments to Shared Expense	M
Settle Up Shared Expense	H
Acknowledge Shared Expense	H
Friends	
Add New Friend	L
Delete Friend	L
View Friend Details (Data analytics, Transactions)	M
Groups	
Create new Group	M
Add Members to Group	M
Remove Members from Group	M
Delete Group	M
View Group Details(Data analytics, Transactions)	M
Chat	
Create new Chat	L

Send and Receive Chat messages	L
Delete Chat	L
Personal Account Settings	
Set Personal Spending Limit	L
Set Time Frame for Limit	L
Set % for Reminder	L
Delete Account	M
Notifications	
Receive notifications	H
Informational Sections	
View About Page	L
View FAQ Page	L

5 Features not to be Tested

All features that

- Are not included in this release of the software
- Are low risk, has been used before and is considered stable
- Will be released but not tested or documented as a functional part in the current release of the software

6 Approach

The following members will be performing the tests together with the development team:

- QA Manager
- QA Engineer
- Lead Developer

Appropriate tests will be conducted and cleared. Any issues must be recorded with details including how to replicate the bug, the person who discovered the bug, and the severity of the bug (L, M, H). Developers will then be assigned to fix the bug. QA Manager & Engineers will keep a record of the bugs and how they were solved.

Testing techniques to be used:

1. Functional Testing

Functional testing tests for all the functional requirements of CashTrack and to ensure they are working correctly. For instance, the functionality of adding a shared expense record must be tested thoroughly to ensure all details of the record reflect fully and accurately for all parties involved.

2. Performance Testing

Performance testing tests for all the non-functional requirements of CashTrack. Types of performance testing include testing what happens if large amounts of data are handled, testing security systems, testing backwards compatibility with other existing systems, and testing the reliability of the system. For instance, the response time between when the user inputs their data and Cashtrack outputs a response to that input must be tested to ensure it meets the minimum performance benchmark.

a. Stress Testing

Stress testing tests the system on how it runs under maximum load or beyond, to determine failure behaviour and check for bugs that only occur under heavy loads. For instance, the time it takes for CashTrack to load a new page when the amount of people using it at the same time exceeds the expected amount must be tested to ensure it meets the minimum performance benchmark under load.

3. Regression Testing

Regression testing reruns old tests for parts of the system after changes (bug fixes, module integration), to see if anything was broken due to those changes.

4. Beta Testing

Beta testing is the final stage of testing before release. Cashtrack will be released to a group of users to test in a real world environment. This will help uncover overlooked bugs and obtain feedback from this group of users.

7 Item Pass/Fail Criteria

7.1 Unit Test Level

A pass criterion must be included in its description for every test case. Generally, two key issue categories might be encountered during testing for software defect, they are the following, Severity and Priority:

- **Severity:** The impact of the said defect is the extent of which the defect can affect the software. In other words, it defines the impact of a defect on the software system and can be subdivided into 5 differing severities, Critical, Major, Moderate, Minor, Cosmetic:
 1. **Critical:** Such defects cause cessation of one or more core components of the system which may lead to data corruption in the process. This makes the system ineffectual and there are no substitute methods to attain the objectives of the system.
 2. **Major:** These defects may result in termination or data corruption in the components but differ from the above critical defects in the sense that there is a suitable alternate method to achieve the necessary results.
 3. **Moderate:** The defect that does not result in the cessation of the system but does cause the system to produce inaccurate, inadequate, or unreliable results.
 4. **Minor:** The defect that does not result in the termination of the system nor the serviceability of the system, and the required results can be easily obtained by developing solutions around the said defects.
 5. **Cosmetic:** The defect that is related to the visual enhancement and appeal of the system whereby the changes are related to the User Interface of the application
- **Priority:** The order of importance in which we should resolve a defect.

1. **High:** The resolving of the defect is of high importance and must be resolved as soon as possible as the defect is/might be affecting the application and/or the product severely. As a result, the system cannot be used until such defects have been rectified.
2. **Medium:** The defect should be resolved during the normal course of development activities. Due to its lower importance status, the defect can be resolved after a new build or version is published.
3. **Low:** The defect is classified as an irritant to developers which should be repaired in due time, but the repair can be deferred until all other more serious defects are resolved first.

Defect Priority-Severity Categories

Combination	Factor Frequency	Pass / Fail
Low Priority & Low Severity	Less than 15% Pass	Pass
High Priority & Low Severity	Less than 5% Pass	Pass
High Severity & Low Priority	-	Fail
High Priority & High Severity	-	Fail

8 Suspension Criteria And Resumption

Testing will be suspended when encountered with critical design flaws that will require an application interface redesigned. Testing will resume when amendments to the code are completed and reviewed.

Severity of Defect	Stoppage Criteria		Resumption Criteria
	Criteria contributing to percentage of defect	Acceptable percentage threshold Level of Defect	
Critical	Defect(s) causing data corruption	2%	Fixed defect(s) causing data corruption of application and acceptable percentage threshold falls below acceptable percentage
Major		5%	
Moderate	Any defect which does not critically impact the development of the system	15%	Acceptable percentage threshold falls below indicated acceptable percentage
Minor		25%	

9 Test Deliverables

The following artifacts shall be provided as testing deliverables available to the stakeholders.

Integration testing requires the following prerequisites to be completed:

- Architectural Design Document
- Detailed Design Document
- Unit Test Plan

Therefore, Specific Integration test only can begin when the following items are delivered:

- Unit testing test reports concerning the components involved (i.e., these components have been unit tested)
- Drivers for this specific integration test
- Input data
- Integration test report
- Integration test output data
- Problem reports
- Test Plan Document
- Test Cases
- Test Design Specifications
- Tools and their outputs
- Simulators
- Static and Dynamic Generators
- Error logs and Execution logs
- Problem Reports and Corrective Actions

9.1 Testing Evaluation Summaries

Test Evaluation summaries will outline the tests conducted and their results. Debugging methods and outcomes will also be recorded. In the event where debugging is unsuccessful, alternative solutions will be proposed in the summaries.

9.2 Incident Logs and Change Requests

Incident log entries will be made for all bugs found during testing. The log will also be used to track the status of the bugs. Any modifications to the requirements must be done through change requests, which will ensure that the proposed changes are fully reviewed before incorporated into CashTrack.

10 Testing Tasks

Listed below are the specific testing tasks required to be performed prior to each release stage of the system CashTrack alongside the estimated time needed to carry out these testing task(s):

Stage of Release	Testing Tasks	Estimated time for Task
First Stage	Authentication and validation of login process	1 Working Day
	Frontend and backend mapping of creating/updating and deleting of personal expense record	2 Working Days
	Check functionality of notification system	1 Working Day
Second Stage	Check reliability of notification system upon nearing prior set expense limit	2 Working Days
	Verify that friends invited to use CashTrack actually gets notification when invited	2 Working Days
	Ensure user is able to add/delete friends to/from friend list	1 Working Day
	Creation/Updating/Deleting of shared expense record	2 Working Days
Third Stage	Creation/Updating/Deleting of groups within the system CashTrack	2 Working Days
	Verify that transaction details displayed between friends are accurate	3 Working Days

	Functionality of additional split bill options and comments	1 Working Day
Fourth Stage	Verify accuracy of the dashboard expense data analytics displayed to user	1 Working Day
	Test functionality of shared viewing of transaction information within a created group	3 Working Days
Fifth (Final) Stage	Test user chat messaging system	2 Working Days
	Test overall reliability of the system documenting defects found if any	5 Working Days

11 Environmental Needs

This section documents the special requirements/environmental needs for this test plan, more specifically, the hardware and software needed to facilitate the overall testing process, from performing the various tests to logging the results of the test.

11.1 Hardware

Hardware used for testing must be consistent with the proposed hardware requirements stated as stated in the System Requirement Specification.

	Minimum Requirements	Recommended Specification
Processor	Intel ® Dual Core ™	Intel ® Core™ i5
Memory	4 GB RAM	
Internet	Broadband internet connection	
Resolution	1024 x 768 minimum display resolution	

11.2 Software

Testing Environment:

- Web Server Database
- User Web Browser

Test Case/Task Management and Logging Tool:

- SVN

11.3 Supporting Documents

Document Name and Version	Location
<i>System Requirement Specification (Version 1.2)</i>	<i>Wiki</i>

12 Staffing, Responsibilities And Training Needs

This section will indicate the people who will be involved to ensure the successful testing of the CashTrack application.

The current size of the development team is small, as such the majority of the team will be involved in the testing process of CashTrack.

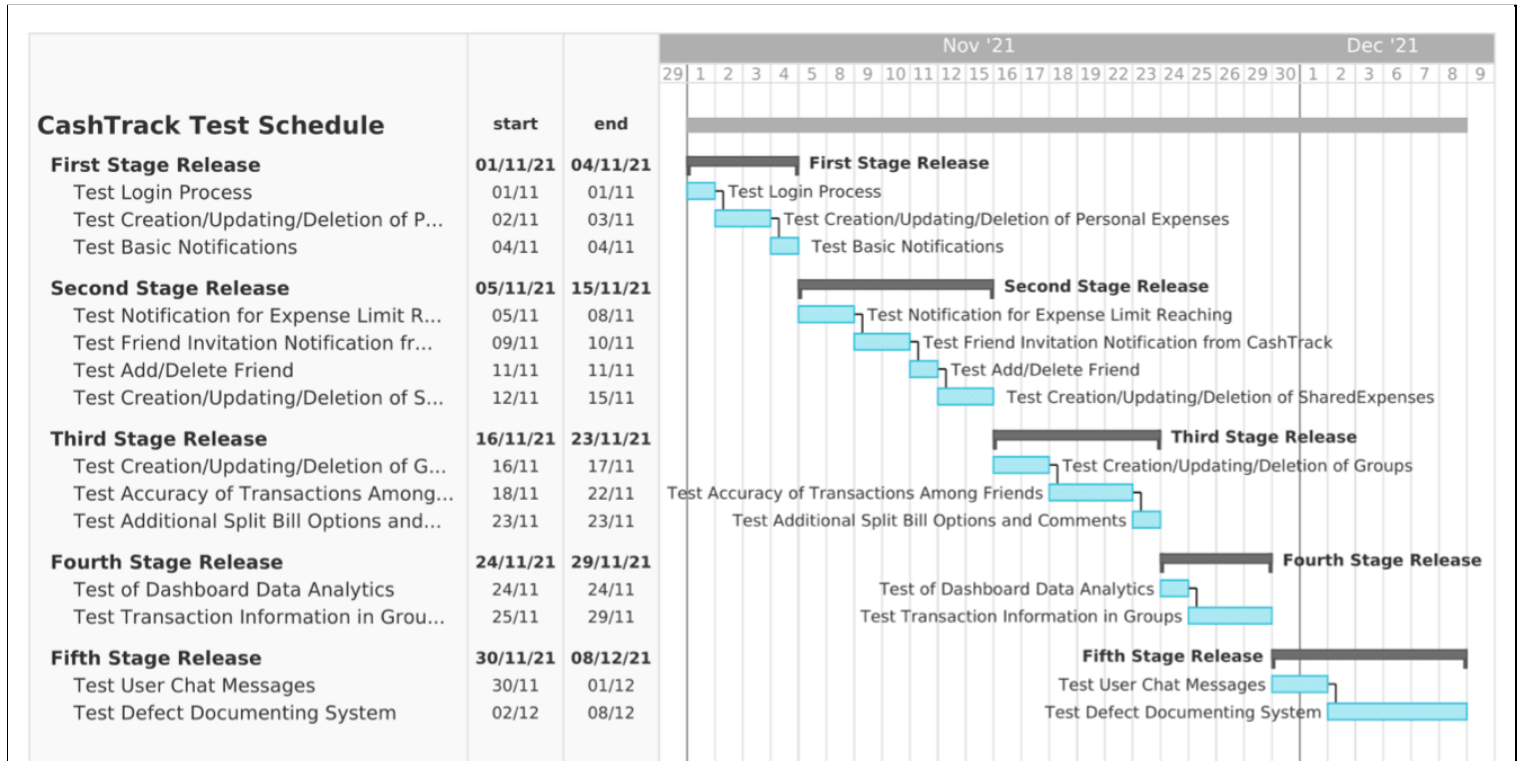
Training will be tailored to each team member's weaknesses so that they will be able to have an all rounded skill set for the development of CashTrack and also allow more crossover of work for each member in the event one team member falls sick, development progress will not be stalled.

Roles	Responsibilities
Project Manager (NICKLAUS TAN)	<ul style="list-style-type: none">- Identify items to be tested- List down the testing details for each item that needs to be tested- Ensuring proper documentation of change requests- Review testing results and provide feedback to development team
QA Manager (S SRI KALKI)	<ul style="list-style-type: none">- List down testing approach- Designing automatic testing architecture to facilitate testing of CashTrack application- Verify correctness of testing techniques- Review testing results and report the testing results to project manager
QA Engineer (ELLIOTT ONG & DANIEL LOE)	<ul style="list-style-type: none">- Assist QA Manager with the design of the testing architecture- Write test cases- Carry out testing of CashTrack system- Document testing results
Lead Developer (KUMAR MEHUL)	<ul style="list-style-type: none">- Facilitate the implementation of test cases on CashTrack- Identify test classes to be created for front-end and back-end

Front-End-Developer (RAVISHANKAR AMRITA)	- Define test classes to allow for easy front-end testing as required by the testing team
Back-End-Developer (DATTA ANUSHA)	- Define test classes to allow for easy back-end testing as required by the testing team

13 Schedule

The Testing Schedule shown in the Gantt Chart below follows our rigorous testing plan based on every release. This schedule is also in line with our original project plan schedule and project timeline Gantt Chart.



14 Risks And Contingencies

This section will indicate the risks, mitigation and contingencies for the testing process of CashTrack.

Risks	Mitigations	Strategy Contingency
Lack of manpower resource to carry out testing	Assign testing roles on top of development work to development team members so that they can help out with the testing process of CashTrack Use automation testing softwares / tools to lessen the amount of manual testing that testers need to carry out	Hire third party software testers to carry out testing for CashTrack
Lack of availability / late delivery of required hardware, software or tools to carry out testing	Have proper early project planning, listing down equipment, software and tools needed for the development of CashTrack to allow early procurement of these equipment and software tools	Designing in house automation testing softwares tools
Delays in training for automation testing software / tools	Ensure that all development team members who will be involved in testing process attends the training early on how to use such softwares by having proper planning during project planning stage	Hire third party software testers who are already trained in using the automation testing software / tools to help carry out testing for CashTrack
Test cases are not comprehensive enough to cover all scenarios	Ensure team member cross check each others test cases and that they are sufficient to cover all	Ensure test plan is reviewed and modified if necessary

	scenarios	Ensure that test cases are redone and reviewed before testing is carried out
Improper time management leading to delay in testing	<p>Ensure there is a proper planning schedule for testing to be carried out</p> <p>Ensure that the deadlines for testing are to be adhered</p>	<p>Testing of critical components will be completed first to ensure that the main functionality of the application is working</p> <p>Test cases will indications based on their priority and the highest priority test cases will be completed first</p>

15 Approvals

For the initial test plan, the Project Manager (Nicklaus), Lead Developer (Mehul) and QA Manager (Sri Kalki) must approve it.

For subsequent successful test cases, the QA manager (Sri Kalki) and tester must sign off the respective use case requirement as detailed in the use case description document.