

***Note that you can refine your testing plan as the project development goes. Keep the change log as follow:***

*ChangeLog*

| **Version** | **Change Date** | **By** | **Description** |
| --- | --- | --- | --- |
| version number | Date of Change | Name of person who made changes | Description of the changes made |
| V1 | Feb 17 | Yixi Wu | Test existing finished feature with acceptance test  CI pipeline run automated unit/integration test |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Scope

Scope defines the features, functional or non-functional requirements of the software that **will be** tested.

**Finished Functional Feature:**Track User expense will be tested with acceptance test  
Send records, unit test and integration test exist.

**Unfinished Feature:**  
Transfer money backend part done, unit test and integration test exist  
Hanlde large amount of user: Will be run through load test in the future

## Roles and Responsibilities

| **Name** | **Net ID** | **GitHub username** | **Role** |
| --- | --- | --- | --- |
| Yixi Wu | wuy8 | LimitlessJesse | Developers/Test Manager |
| Michael le | lem2 | mike91827 | Developers/QA |
| Minh Nam Hai Nguyen | nguyemnh | namhai923 | Developers |
| Mohammadafaz Munshi | munshim | afazmunshi49 | Developers |

# Test Methodology

## Test Levels

**Test Levels define the Types of Testing to be executed on the Application Under Test (AUT**). In this course, **unit testing, integraiton tesing, acceptance testing, regression testing, and laod testing** are mandatory. Please describe how will you do these testings. You may skip load testing at this moment. Please revisit it after the related lecture is given.

Unit Test:

API test in the backend for all features: ensure that the API gives correct result and fetch correct information from database. Ensure error handling: API gives proper status code and error message when bad request are given like trying to query an unexisted user.  
Or trying to send expense record when some account’s balance is not enough.

Integration Test:

Send Records: After sending the records, the database should update

Transfer money: Using API test, ensure that the database is updated with correct logic change. When transferring money between two account, account balances change correctly for the two account in the database.

Acceptance Test:

Track expense: User should be able to view their remaining balance and past expense history as a table. And it contains information like amount of the expense and date. And they are able to use filtering to filter their expense

Regression Testing:  
 Setup automated CI pipeline: everytime a PR is created, or a PR is merged, automated unit tests and integration tests will run.

## Test Completeness

Here you define the criterias that will deem your testing complete. For instance, a few criteria to check Test Completeness would be

* 100% code coverage
* Automated test will run everytime new changes are push or PR is created
* acceptance test can succeed
* Test can caught bug and be fixed right away if it was created by new PR

# Resource & Environment Needs

## Testing Tools

Make a list of Tools like

* Automation: Github Actions
* Backend test: Jest, supertest, nodejs
* Frontend test: Jest
* Postman to send request to backend and update database

## Test Environment

* Windows 8 and above
* Github Actions and nodejs

# Terms/Acronyms

Make a mention of any terms or acronyms used in the project

| **TERM/ACRONYM** | **DEFINITION** |
| --- | --- |
| API | Application Program Interface |
| AUT | Application Under Test |