



Team 1 - trackr

MET CS 673

05/16/2022

Team Members

Team Member Name	Role(s)
Jean Dorancy	Team Leader
Timothy Flucker	Design / Implementation Leader + Requirements Leader
Xiaobing Hou	Security Leader + Configuration Leader
Weijie Liang	QA Leader



Project Overview

- **High-level Description:** A personal Transaction tracker application similar to Mint or TrueBill.
- **Motivation:** An informal and personal way for users to enter information related to their bank accounts and its withdrawals and deposits.
- **Goal:** Allows a user to visualize and understand their spending habits for their bank accounts.

Related Work

- Intuit - [Mint](#)
- Rocket - [Truebill](#)
- These applications are developed by large financial institutions and as such are constantly marketing ads for special bank accounts or credit cards that distract the user from the reason they subscribed: **to understand their spending behavior**

Requirements

- Functional Requirements
 - Essential Features
 - User Management
 - Bank Account Management
 - Transaction Management
 - Desirable Features
 - User Management
 - Optional Features
 - Web GUI (Graphical User Interface)
- Non-Functional Requirements
 - Swagger Document
 - Authentication
 - Authorization

Functional Requirements

Essential Features (1/2)

- User Management

- As a customer, I want to be able to **create a user account** for the application, so that I can have my data associated with my account. Est: 2 - 4 hours

- Bank Account Management

- As a user, I want to **create a bank account record** so that I can track deposits and withdrawal transactions against it. Est: 2 - 4 hours
- As a user, I want to **modify a bank account record** so that I can update its relevant information to be current. Est: 2 - 4 hours
- As a user I want to be able to **deactivate a bank account record**. Est: 2 - 3 hours
- As a user, I want to be able to **view all of my bank account records**, so that I can interact with all of the data I have entered. Est: 2 - 3 hours
- As a user, I want to be able to **view a specific bank account record using a unique identifier**, so that I view its data and take any necessary action against it. Est: 2 - 3 hours



Essential Features (2/2)

- Transaction Management

- As a user, I want to **create a transaction record** linked to a bank account so that my bank account information is up to date. Est: 2 - 4 hours
- As a user, I want to **modify a transaction record** so that I can update its relevant information to be current. Est: 2 - 4 hours
- As a user I want to be able to **void a transaction record** so that I can reverse the transaction and update my bank account. Est: 2 - 3 hours
- As a user, I want to be able to **view all of my transactions against a bank account**, so that I can see all of the activity of that bank account. Est: 2 - 3 hours
- As a user, I want to be able to **view a specific transaction using a unique identifier**, so that I view its data and take any necessary action against it. Est: 2 - 3 hours

Desirable Features

- User Management

- As a user I want all of my **API requests to be authorized using my user credentials (basic authentication)** so that only I can view my data. Est: 5 - 10 hours
- As a user, I want to be able to **reset the password of my account** so that, if I forget my password I do not lose access to my account. Est: 4 - 8 hours

Optional Features

- Web GUI (Graphical User Interface)
 - As a user, I want to have a **web interface to log into the application** so that I do not have to use the APIs. Est: 5 - 10 hours
 - As a user, I want to have a **homepage that shows all of my relevant information** so that I can interact with my data more easily. Est: 20 - 30 hours
 - As a user I want to have a **page where I can add information** such as my bank account, or transaction information so that new information can be added quickly and easily. Est: 20 - 30 hours
 - As a user I want to have a **page where I can edit information** such as my user, bank account, or transaction information so that I can keep my data up-to-date. Est: 20 - 30 hours
 - As a developer, I want to **enable session management for all web pages** so that the application is secured. Est: 10 - 20 hours

Non-Functional Requirements

Non-Functional Requirements

- Swagger

- As a developer, I want a **swagger document which shows how my APIs are designed**, so that I have a relevant technical project artifact. Est: 5 - 10 hours

- Authentication

- As a user I want **all of my API requests to be authenticated using a basic authentication strategy** so that my data is protected from random and unsolicited access. Est: 5 - 10 hours

- Authorization

- As a user I want **only API requests I send or from a system admin to be authorized to access my data** so that only I and a system admin can view my data. Est: 5 - 10 hours



Management Plan

- Objectives and Priorities
- Process Model
- Communication Plan
- Risk Management
- Timeline

Objectives and Priorities

1. Everyone actively contributes and practices Scrum rituals.
2. Opportunities for everyone to learn.
3. Focus on best practices to maintain high quality.
4. Bottom up unit-testing approach.
5. Complete all essential features without compromising software quality.
6. Complete some desirable features.

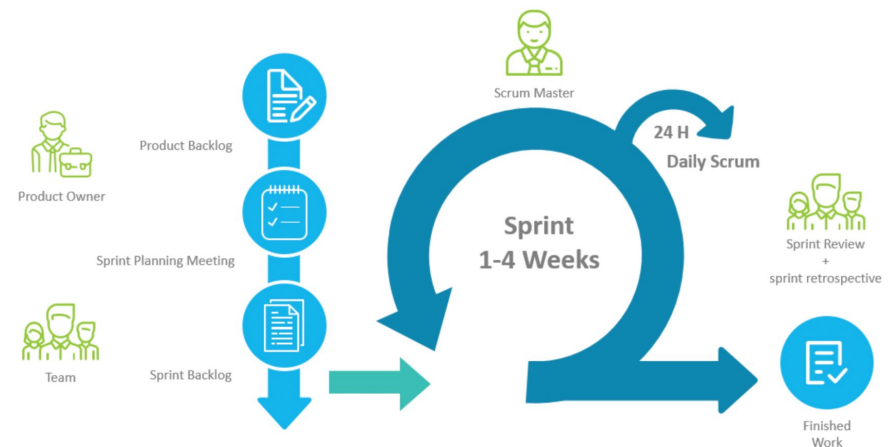
Process Model

- **Sprint:** Iteration length.
- **Daily Scrum:** Async in Discord chat.
- **Sprint Retro and Planning:** Discuss and prioritize goals for the sprint.

Note: Stakeholders (professor and facilitator) feedback will be async.



AGILE SCRUM PROCESS



Communication Plan

- Zoom for synchronous meetings.
 - Sprint rituals
 - Team meetings
 - Meetings are recorded and uploaded to the team folder
- Discord channel for instant messaging.
 - Daily Scrum Thread
 - Pull Requests Thread
- Pivotaltracker for managing requirements and tasks.



Risk Management

Risk Title	Priority	Plan
Constant requirements change	8	Favor generic solutions that are adaptable and embrace change
Unclear requirements	12	Constant communication with stakeholders.
Lack of motivation or responsibility	15	Everyone to work on things they are interested in and practice Scrum rituals.
Scope creep	16	Explore alternative solutions.

Timeline

Sheet with detail estimates for each iteration: [here](#)

Iteration	Requirements	Estimated (man-hours)
1	<ul style="list-style-type: none">- User Management- Bank Account Management	50
2	<ul style="list-style-type: none">- Transaction Management	35
3	<ul style="list-style-type: none">- User Management- Create a Swagger document for APIs- Web GUI	165

Configuration Management Plan

- Tools
 - Dev Tools
 - Management Tools
 - Test Tools
- Code Commit Guideline and Git Branching Strategy
- Deployment Plan
 - Platform
 - Multiple Environments

Tools

Development

Environment

- Java 1.8
- Swagger
- H2 DB
- Maven

Version Control Tools

- Git and GitHub

IDE Tools

- IntelliJ



Project Management

Management Tools

- PivotalTracker
- Google Drive

Meeting and Dissemination Tools

- Zoom
- Discord

Testing

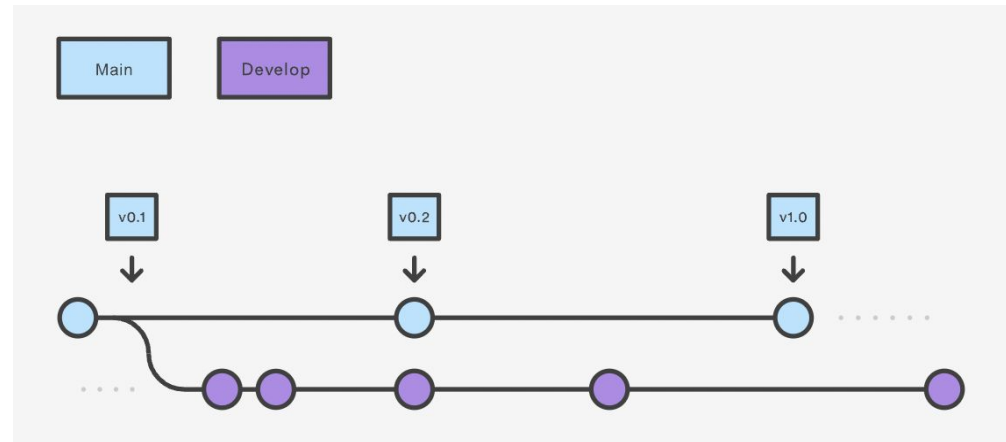
Test Tools

- Postman
- JUnit5
- Selenium

Code Commit Guideline and Git Branching Strategy

Two Protected Branches

- main
 - Releases
 - Production deployment
- development
 - Integrate feature branches
 - Stabilize features integration
 - Development deployment



Note: PR submission, code reviews and approval required before a change can be merged.

Deployment Plan

Platform

- Deployment on Heroku Cloud

Multiple Environments

- **Development:** To deploy and test features during an iteration.
- **Production:** Customer facing environment where we deploy at end of an iteration.

Quality Assurance Plan

- Metrics
- Coding
 - Standard
 - Review Process
- Testing
 - Unit Testing
 - Manual Testing
- Defect Management

Metrics

Metric Name	Description
Number of Features	Number of features that were implemented; shows the complexity of the project.
Number of Defects	Number of defects reported in the project; shows the reliability and completion of the project.
Total Man Hour	The total time spent on preparing for the project. This will allow us to track the total amount of effort spent in support of the project.
Test Passing Rate	Number of tests passed. This metric will be measured during each iteration, and the types of tests include unit tests and manual tests.
User Story Counts	Number of user stories created in Pivot Tracker. The number of completed user stories will be tracked in every iteration.



Coding

Coding Standard

- Use of Google Java coding standard.
- Autoformatting by configuring IntelliJ IDE.

Code Review Process

- All changes to be reviewed and approved via PR.
- Review Checklist
 - Manageability
 - Architecture
 - Maintainability
 - Correctness
 - Invalid input/states
 - Usability
 - Reusability
 - Object-Oriented Analysis and Design Principles

Testing

- **Unit testing:** Unit testing will be conducted by the developer responsible for writing the code for each core method before submitting a pull request.
- **Manual testing:** QA Leader performs manual testing of the core functionality after each iteration. The test methods and results can be recorded in the test report.

Defect Management

- **Defect Type:** High, Media and Low
- **Defects Tracking:** The QA Leader will use Pivotaltracker to track fixed and unfixed defects and the fix rate at the end of each iteration.
- **Defect Resolution Process:** Any team member who finds a defect will create a ticket in Pivotaltracker and assign it to the developer responsible for the feature with the following information.

Questions?



trackr

References

The Ultimate Code Review Checklist

https://www.codegrip.tech/productivity/the-ultimate-code-review-checklist/?utm_source=website&utm_medium=blog&utm_campaign=best-practices-for-code-review-process

Java Coding Standard

<https://google.github.io/styleguide/javaguide.html>

Atlassian Git Workflow

<https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>

Generate Project Name and Logo

<https://namelix.com/>

Agile Scrum Process

<https://powerslides.com/powerpoint-business/project-management-templates/agile-scrum-process/>

