

CS673 Software Engineering

Team 1 : trackr

Meeting Minutes

All meeting minutes are kept in this single document. The latest meeting minutes should be at the beginning of the document. For example, meeting 3 minutes is placed before meeting 2 in the document. The team leader should prepare a basic agenda for the meeting and team members should rotate to be the minutes taker. Each group should have at least one meeting per week, and you may have multiple meetings if needed.

Meeting 2

Date and Time: 05/15/2022 7:30 - 8:00 PM EST

Place: [Zoom link](#)

Participants: Jean, Tim, Weijie, Xiaobing and Professor Yuting

Minutes taker: n/a

Timekeeper: n/a

Purpose: Touch base with how the team is doing after the first week about delivery iteration 0 and discuss the software process.

Agenda:

- Stakeholders feedback and contribution
- Determine team progress on delivery Iteration 0
- Discuss presentation rotation so everyone can participate.
- PR approval requires only one approval (to avoid slowing down the team)
- Discuss adopting Agile - Scrum Framework as the software process
- Discuss Progress Report Sheet

Discussions:

- Stakeholders feedback and contribution
- Determine team progress on delivery Iteration 0
- Discuss presentation rotation so everyone can participate (Proposal below)
 - Iteration 0 - Jean
 - Iteration 1 - Tim
 - Iteration 2 - Xiaobing & Weijie
 - Iteration 3 - Everyone
- PR approval requires only one approval (to avoid slowing down the team)

- Discuss adopting Agile - Scrum Framework as the software process
- Discuss Progress Report Sheet

Key Decisions

- One approval per PR
- Invite professor to the planning meeting
- Agile-Scrum
 - Sprint (length of iteration)
 - Sprint Retro & Planning (per iteration)
 - Daily Standup

Action Items:

- Jean prepare the presentation for iteration 0
- Create Scrum meetings that we agreed on

Meeting 1

Date and Time: 05/11/2022 7:00 - 8:00 PM EST

Place: [Zoom link](#)

Participants: Jean, Xiaobing and Timothy

Minutes taker: Timothy

Timekeeper: Jean

Purpose: Team introductions, determine our project, divide sections of SPP, and other miscellaneous meeting items

Agenda:

- Intro + Timezones
- Determine our communication plan
- Project Ideas (everyone bring at least one idea)
- Git workflow and lab 1 review
- Divide and conquer for our deliverables for iteration 0 (SPPP document)
- Find a team name
- Anything else the team wants to address

Discussions:

- Determine our communication plan
 - We are using **Zoom** for live classrooms and are already familiar with it so it makes a natural choice for synchronous meetings.
 - Daily **Virtual Standup** Thread
 - While we don't expect anyone to work on this full-time since folks have other commitments it would be nice to just get status updates.
 - **Update Template Example**
 - What did you do yesterday?
 - What will you do today?
 - What blockers stand in your way?
 - Example:
 - What did you do yesterday? **n/a**
 - What will you do today? **Looking into setting up deployment for dev**
 - What blockers stand in your way? **No blockers**
 - We have a **Discord** channel for instant messaging. I would encourage everyone to install the Discord app on their phones so they are on the loop about what's happening on the channel.
 - We need to **document everything** so we keep records of the work being done and how it's being done.
 - We need to set up **Pivotaltracker** and create tickets and have a board which can give us a view of all of the work being done and status.
 - If we run into **roadblocks**, post in the channel for help and if we can't find a resolution then we reach out to the facilitator and lastly email the professor.
- Project Ideas (everyone bring at least one idea)
 - Make a project selection
 - Get a project skeleton started
 - Decide on DevOps
 - Cloud choice
 - Build a development environment & deployment pipeline
 - Build a production environment & deployment pipeline
- Git workflow
 - Proposed workflow
 - main branch is protected and requires two reviews in order to merge changes

- Create a development branch which we can use to stabilize changes from multiple features. It's also a protected branch and also requires two (2) reviews before merging changes.
 - Use personal branches created from the development branch for feature development, experimentation etc.
 - When merging to main or development branches we squash and merge commits so we can have a single commit with a description of the work done. This is particularly useful if we need to revert so we can just revert a commit. Work commits in personal branches can be anything you want.
 - Post reviews in the channel
- Discuss Lab 1 - Set Up Git
 - main
 - development
 - Lab 1 with empty readme.md
- Set up a deadline for everyone to complete the lab and merge to the main branch.
- Divide and conquer for our deliverables for iteration 0: Each group should work on the project documents collaboratively on google drive, and commit them on your Github repository, together with any source code if any. You shall create a release for this submission that include the following items:
 - Readme.md - **done**
 - That's part of lab1.
 - Doc/CS673_presentation0_teamX
 - Looks like the presentation is based on the SPPP (which makes sense)
 - Doc/CS673_SPPP_teamX - **on tack**
 - We will assign different sections depending on areas of focus
 - We will all collaborate on the document to make sure it's complete
 - Doc/CS673_SPPP_RiskManagment_teamX - **on track**
 - Doc/CS673_progressreport_teamX - **To be completed by Monday**
 - Everyone to complete the progress report individually by creating a tab with their name
 - Team to agree on the summary for the group report
 - **Doc/CS673_meetingminutes_teamX - done**
- Find a team name - **done**
- Anything else the team wants to address

Project Idea 1

Transaction Tracker (basic CRUD API application)

Description: Similar to Mint, create an application (Web / APIs) that tracks transactions to a "bank account". Bank Account and Transaction information would be entered manually and transactions would either add / subtract from that bank account's balance.

Users will create an account and will be given a set of credentials that authorize them and allow them to access the Web interface / API interface. A new user will be given a default USER role and only be able to view information relevant to themselves. ADMIN users will be able to see records related to any user.

Essential Features (APIs):

- Add bank account
- edit bank account
- delete bank account
- view bank accounts
 - findAll
 - findById
- create user
- authorize API requests
- add transaction
- edit transaction
- delete transaction
- view transactions
 - findAll
 - findById

Additional Features (nice to have):

- web GUI
 - login page
 - landing page
 - display bank accounts
 - display transactions tied to accounts
 - add bank account page
 - add transaction page
 - edit bank account page
 - edit bank account page

- custom data displays
 - show charts of deposits into an account vs. withdrawal
 - show spending for account within a specified time frame (1 week, 1 month, 6 months)

Technology Stack

- Java 1.8 / Java 11
- Spring Boot Framework
- Eclipse IDE or IntelliJ or VSCode (no preference)
- Git
- Postman (API testing tool, free)
- H2 (in-memory database, for easy to access data)
- Junit5 (unit-testing)
- Swagger (API design / documentation tool, free)

Project Idea 2

Simple app for an ad agency to manage advertising campaigns which can be found [here](#). I'm proposing the same stack as "Project 1" but for the DB we can use PostgreSQL or MySQL if we are deploying to Heroku.

Project Idea 3

Small desktop game (2048). This is a project I did in CS-622 class which display [here](#)

Key Decisions

- Daily Virtual Standup Thread: Post updates in the dedicated thread.
- Pull Request Thread
- Project 1 is selected
- Revisit two reviewers in order to make sure we are not slowing down development time.
- Deadline for Lab 1 is Monday 05/16/2022 at 8AM
- Recurring meeting every Wednesday 7PM - 8PM

Action Items:

- Setup Pivotal Tracker Project and invite team members - **done**
- Create Epic **Iteration 0** in Pivotaltracker - **done**
- Schedule a check-in for Sunday @ 7:30 PM - **done**
- Upload video of the meeting to Drive - **done**

- Email professor about team 1 meeting schedule. - **done**
- Confirm with Weijie that they are able to complete their section - **done**
 - If not, team will divide their section
- Find a team name: Post in the channel. - **done**
- Create a Hello World Java application for basis of project and commit to GitHub repo