Development Process Documentation

Notes 1

Requirements 2

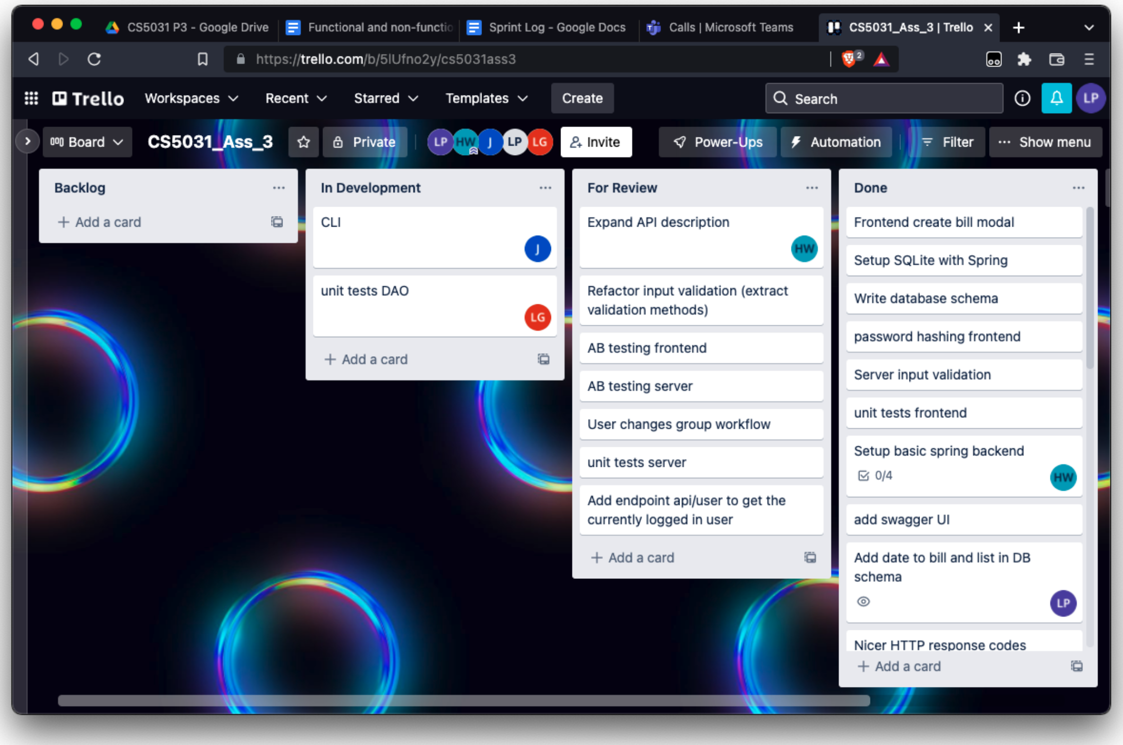
Functional requirements 2

Non-functional requirements 2

Sprint Log 3

Notes

- Sprint log contains rough outline of sprint cycles, the to-dos of each cycle, and the review of previous to-dos
- Meetings were (almost exclusively) held on Teams
- Work allocation and ad-hoc communication was done on Teams
- Backlog and granular todos were managed on Trello:



Backlog

+ Add a card

In Development

- CLI
- unit tests DAO

+ Add a card

For Review

- Expand API description
- Refactor input validation (extract validation methods)
- AB testing frontend
- AB testing server
- User changes group workflow
- unit tests server
- Add endpoint api/user to get the currently logged in user

+ Add a card

Done

- Frontend create bill modal
- Setup SQLite with Spring
- Write database schema
- password hashing frontend
- Server input validation
- unit tests frontend
- Setup basic spring backend 0/4
- add swagger UI
- Add date to bill and list in DB schema
- Nicer HTTP response codes

+ Add a card

Requirements

Functional requirements

- User shall have their own account to which they can login repeatedly
- Users shall be able to create their account themselves
- Users shall be able to create groups that correspond to their flats
- Users shall be able to see all bills in their group, the bills outstanding for them, the amount outstanding and their account total.
- The users shall be able to set the relative percentages due by each member of the bill individually (It does not have to be 50/50 if the bill has 2 users)
- There shall be one admin in a given group - the admin can ~~create recurring payments and~~ add members to the group, as well as changing the role of another user to admin
- The User that sets up the bill shall be the owner of the bill - they can set the amount due, the people that have to pay it back, and the percentages the user have to pay, and a means of payment.
- The users that are not the owner of the bill can only view the bill and pay it back
- The owner of a bill may also add the bill to an existing shopping list
- Users shall be able to mark any outstanding bills as paid. A user can only pay their share of the amount due in full.
- Users shall be able to create shopping lists
- Users can only be part of a single group
- Users shall have a view of their total due payments
- Users shall have a view of their payment history
- (A/B testing): Users shall be able to pay off all their payments at once on the overview page

Non-functional requirements

- The system shall use an SQLite database
- Vue.js client
- Terminal client
- Java Spring backend

Sprint Log

Meeting	21/03
Description	First meeting with brainstorming of ideas
Todo	Pick an idea

Meeting	22/03
Description	Decided to build a system that lets flatmates coordinate their expenses, shopping etc.
Todo	- Determine (non-) functional requirements - Decide on work allocation

Meeting	23/03
Description	- Establish (non-) functional requirements - Agree on basic API design and endpoints - Design database
Todo	- Start first sprint cycle

Sprint cycle	23/03 - 25/03
Todo	- Basic setup of vue.js client - Basic setup of Spring application
Review	n/a

Sprint cycle	25/03 - 28/03
Todo	- Integrate SQLite database on the server including a data access object - Setup API endpoints with restricted access - Implement Admin and User views in the frontend - Setup Gitlab CI workflow
Review	- Vue client has been set up - Spring has been set up including web security

Sprint cycle	29/03 - 01/04
Todo	- Frontend-client / server integration - Add lobby view to client - Unit tests for Server - Begin implementation of terminal client - Fix gitlab CI workflow
Review	- SQLite fully working - Majority of API endpoints integrated - Gitlab CI workflow not working yet - Swagger UI implemented

Sprint cycle	02/04 - 05/04
Todo	- Finish unit tests for Server - Add more extensive testing for DAO - Implement A/B testing functionality - Server refactoring (style, comments) - Finish CLI implementation
Review	- Frontend and server fully integrated; frontend meets all functional requirements - Server meets all functional requirements - Unit tests finished

Sprint cycle	06/04 - 07/04
Todo	- Write group report - Prepare project for submission - Submit project
Review	- Unit tests finished and all passing - Server and clients meet (non-) functional requirements