

VECKA 3: Designmodell (torrköra User Stories m.h.a. design modell); körbar (ev. i form av tester bara)

Meeting Agenda (Group: Alpha)

Date: 2020-09-16

Chair: Sophia Pham

Participants: Mathias Drage, Benjamin Sannholm, Bashar Oumari, Pontus Nellgård, Sophia Pham

Objectives (5 min)

- Some simple sketches of the GUI.
- Roughly decide on domain model and design model for viewing the map (user story #4).
- Decide on Definition of Done.
 - Decide on strategy for testing.
- Break out tasks into individual cards (notes not issues?) on Scrum board.

Reports (15 min) from previous meeting

- Pontus has made a sequence diagram.
- We have individually been learning about Android concepts and best practices.
- Sophia has drawn some sketches.

Discussion items (35 min)

- Error message because of Firebase for some, including the supervisor. This was resolved.
- Testing strategy
 - We are using JUnit to test.
 - Each person can choose when they want to write their tests (Before or after implementing).
 - Each person will write tests for their own code.
- Definition of done
 - Before starting on task/feature
 - A branch should be created for each self-contained task (user story or task). GitHub flow overview:
<https://guides.github.com/introduction/flow/>
 - In progress → Testing
 - The code should appear to work according to user story requirements.
 - No unnecessary dependencies should be present.
 - Any complicated parts of the code should be documented through comments.
 - All preconditions of methods should be documented through comments above the method.
 - The code infrastructure should strive to follow MVC and other OOP design principles and patterns where appropriate.

- The domain model and design model should be updated as well as possible to reflect the newly implemented code.
- Testing → Review
 - No known bugs should be present.
 - The most important methods in the public interface of each module should be tested well.
 - All public methods that call other methods should be tested.
- When ready for review
 - GitHub flow overview: <https://guides.github.com/introduction/flow/>
 - If you have a local branch, it should be pushed to GitHub.
 - A pull request should be created to compare the feature/task branch with the master branch.
- Review → Done
 - The code should work according to user story requirements.
 - All tests should be passing.
 - The code should be easily comprehensible.
 - The code should be well documented through comments where appropriate.
 - No unnecessary dependencies should be present.
 - The code infrastructure should strive to follow MVC and other OOP design principles and patterns where appropriate.
 - The domain model and design model should be updated to reflect the newly implemented code.
- When review is done
 - GitHub flow overview: <https://guides.github.com/introduction/flow/>
 - The pull request is merged after fixing any merge conflicts locally on the feature/task branch.
 - If the feature/branch is not relevant beyond that instance it should be removed when the merger is done.

Outcomes and assignments (5 min)

- We have decided not to solve the error message issue because it only affects us if we want to use the Firebase sign up feature and it takes too much unnecessary time.
- We have the same assignments as last meeting because the second meeting was just two days after the first meeting.
- We have decided on some preliminary sketches of the GUI.
- We have decided on a preliminary Definition of Done (this document) which after refinement will be uploaded to the official DoD document.
- On the Scrum board, tasks will be broken out into individual cards as needed by whoever is performing the task.
- Deciding on the domain model and design model for viewing the map (user story #4) will be done after designing individually later.
- Discuss short Friday meetings at the next meeting.

Wrap up

- Next team meeting: 2020-09-21 Monday 13:15
- Next supervision meeting: 2020-09-23 Wednesday 13:15 + Team meeting afterwards

Comments from Felix:

1. Just nu så är vår Scrum Board för statisk. Man brukar ha individuella tasks i "in progress" inte hela user stories. Så när man är klar med tasken så skickas de vidare till "Testing".
2. Ha numrerade user stories och numrerade tasks så vet man vilken kategori den tillhör. t.ex. user story #3 och så task #3.1, #3.2 osv.
3. Det är viktigt att vi arbetar vertikalt då vi inte kommer ha något att visa upp ifall alla arbetar med sitt egna horisontellt.