

# Taking minutes

## Agenda for the team meeting [Project].

Location: Drebbelweg  
Date: 21.02.2023  
Time: 15:45  
Attendees: Alani Rafael, Daga Aryan, Ioannou Andreas, Purice Victor, Rokoma Anastasia, Zambarta Eirini

### *Summary of the meeting:*

No.	Topic	Content Discussed
1	Backlog Review with TA	<p>Difference between epic and user story-</p> <p>Elena: Epic is a bigger feature/ functionality and as she mentioned in the feedback, look at slides. The advanced features are basically the epics we need to build upon.</p> <p>We will have an epic for the basic features and an epic for advanced features.</p> <p>Victor: goals- description or client/server related</p> <p>Epic: needs to be user story related. "As a user I should be able to enter the account without login etc"</p> <p>Don't mention functions like JavaFX or Spring etc.</p> <p>PS: We need a motivation/reason for each user story.</p> <p>Functionality in user stories should be very specific. Don't put in non functional stuff. The user doesn't care if it's in C# or javaFX etc.</p> <p>Order:</p> <ol style="list-style-type: none"><li>1. Title (app title)</li><li>2. Epic &gt;</li><li>3. (contains) user stories</li></ol>

		<p>Epic structure: As in lecture- I.e</p> <ol style="list-style-type: none"> <li>1 epic for Basic Application</li> <li>1 epic for <u>each</u> Advanced Features</li> <li>1 epic for Optional Requirements</li> </ol> <p>Re Split user story into an acceptance criteria.</p> <p>We need to increase the number of user stories.</p> <p>Eg: As a user I want to be able to add specific tasks in order to categorise my tasks. Add a comma and write a short reason as to why the user story is valid.</p> <p>Basically, we need to restructure it a bit.</p> <p>Most user stories should not have an acceptance criteria.</p> <p>Mocks: Elena: We need to have a mock for an epic and NOT for each app or do no mocks at all. For Rafael and Aryan: Let's make the mock for most epics for an excellent.</p>
2	<b>Sprint Backlog</b>	<p>Decided on how we make a structure on how we work. It should be on the basis of the backlog so we need to finish that.</p> <p>Elena: Creating a branch for each issue NOT for each name. We need to have a main branch and also NEED to have a development branch. The main needs to be functional all the time and we need to keep a buffer called the Development Branch and then we merge into Development. And every few weeks we move it to main so that it doesn't fail. Main is basically the final REFINED version of our app.</p>
3	<b>GitLab issue tracker</b>	Write some issues for the GitLab tracker.

		Elena: write some template issues and code based on them. Add the issue to the milestone too. Look over it before and after the sprint. Give every issue it's own branch.
<b>4</b>	<b>To start working</b>	Decided to start working this week (Thursday).
<b>5</b>	<b>How is the git assignment going?</b>	<p>We are positive we can finish it by Friday, we can help each other out and get it done by Friday.</p> <p>Victor: has a huge tree.</p>
<b>6</b>	<b>Should every commit be a merge request?</b>	<p>Elena: We should have commits and then once we're at 30 or so commits we should merge it into Development Branch. Then every now and then create a merge request to merge it in Main.</p> <p>IMPORTANT: before the meeting we should have the main fully merged and up to date. From this week onwards, we will have another Rubrik, which is a potentially functional product. So if we have nothing in the main then there's nothing functional.</p>
<b>7</b>	<b>Time for every merge request</b>	The code should be reviewed in 24 hours of the merge request. So each person will have 24 hours to review every merge request.
<b>8</b>	<b>Spring Injection</b>	<p>You won't be able to write a functional application without spring.</p> <p>But for Dependency injection for testing we should write unit tests like OOP. We will have a lecture for Testing. But for now only unit testing is enough.</p> <p>No threshold for test coverage.</p>
<b>9</b>	<b>Checkstyle Rules</b>	<p>We have to agree on 10 checkstyle rules we need to follow:</p> <p>We made the list and we can generate the .xml file based on that.</p>
<b>10</b>	<b>What kind of policy do we</b>	Already done.

	<b>adopt for Git Development</b>	
<b>11</b>	<b>Type of application?</b>	We need to have a native application. Not a web application
<b>12</b>	<b>Team making</b>	For this week:  JavaFX team: Victor, Aryan and Eirini  Spring team: Antreas, Rafael and Anastasia
<b>13</b>	<b>Feedback</b>	Good in detail meeting 👍
<b>14</b>	<b>Next few meeting</b>	Week 4: Chair: Aryan, minute taker: Eirina Week 5: Chair: Eirina, minute taker: Anastasia Week 6: Chair: Anastasia, minute taker: Antreas Week 7: Chair: Antreas, minute taker: Rafael I'm sorry if I made spelling errors.