Nihal Rajan (Project Leader)

Dion Peng

Qusai Ghabrah

Shashank Kumar

TAs: Jakob Hain, Rylan Kasitz

CS 30700: Product Backlog - Team 6

Project Name: *Ollie*

1 - Problem Statement

Without an effective management system, it can be difficult to track all the aspects of project stories such as assignee, priority, status, etc. Additionally, discussions about the stories may occur through email, Microsoft Teams, and other platforms which would make searching for conversations about a specific topic from weeks or even days ago difficult. One way to solve this is by having a designated section for comments regarding each ticket as JIRA does. Our project will include a lot of the core functionality of JIRA, but we plan to add other features such as different user roles, email notifications about tickets not recently updated, and scripting where when somebody pushes a commit, a ticket is automatically created for the code reviewer.

2 - Background Information

a)

In large-scale software teams dealing with large software products, it is imperative that teams can view all of the tasks and bugs with a particular service in order to understand and visualize the project’s goals and progress. With no such visualization, projects would take more time and effort to be organized, and some tasks and bugs could be overlooked in the development process.

Our application falls under the domain of an Agile project management tool used to help guide the progress of a software product.

The targeted users would be other software product teams looking for a way of organizing their workflow in a space where the team can holistically view the project and the status of different facets of the product.

b)

Jira, Trello, and Azure DevOps are similar services to our planned application. They are all Agile management services but lack some features that would make them more productive for users.

c)

With our project, we plan to use Git Hooks. This way, when a software engineer pushes changes to GitHub in a certain repository, a Jira ticket will be automatically created notifying the team responsible for that repository to review the pull request. Another such use case of Git Hooks is if the software engineer hyperlinks the Jira story in the pull request description, upon merging the pull request to master, the Jira ticket is automatically closed.

We will also allow for assigning roles to each user on the team. One thing that would be included in our project but not the mentioned services is adding conditions where, for example, tickets will be assigned to different reviewers based on the folder(s) where code is committed.

Another limitation of these services is that sometimes tickets can be forgotten and are not updated during a given sprint. In this case, it would be beneficial to inform team members through an email notification as a reminder to update the ticket. We want to integrate email notification functionality into our JIRA clone that is based on story progress and the users who are a part of those stories.

3 - Requirements (Backlog)

**Functional Requirements:**

| As a... | I would like to... | so that... |
| --- | --- | --- |
| user | add people to the board with view and edit permissions by email | there is some level of security and only people who should have access to the board can see/edit it. |
| user | assign stories to team members (other users) and themselves. | each user will know what to work on. |
| user | view, edit, delete, create, and comment on the stories needed for the completion of a project. | the system will serve as an interactive way to collaborate and communicate regarding tasks. |
| user | synchronize stories across multiple computers (not locally). | all members of the Agile team can see an up-to-date project board. |
| user | assign priority and time estimate for stories. | it is easier to get an idea of how much time should be spent on each story. This will allow for prioritizing certain stories over others as appropriate. |
| user | change story statuses between the four Agile categories: “backlog”, “blocked”, “in progress”, “completed” | all members of the team are up-to-date with all the stories. |
| user | receive real-time synchronization of stories’ status. | refreshing the board is not necessary to see changes coworkers have made. |
| user | receive email notifications as an assignor or assignee i.e. if the assignee of a story changes the status of the story from “In Progress” to “Blocked”, the assigner will receive an email notification. | the assignor and assignee are notified of any story status changes. |
| user | receive reminder emails if a ticket is not updated during a given sprint. (if time allows) | it will be clear what the status is for each of the stories even if the assignee forgets to update the status. |
| user | expose scripting functionality via automation of the ticket creation process, meaning that when somebody pushes a commit, the software automatically creates a ticket for the code reviewer. (if time allows) | it can be ensured that all code is reviewed on time and the progress on the review can be tracked. |
| developer | test changing story statuses, adding stories, adding users to the board, and assigning point values/priority | the basic functionality of the storyboard can be tested before implementing other advanced features. |
| developer | test the storyboard functionality in the viewpoint of the user | we know how users will behave when using the application and what processes need fine tuning for user experience |

**Non-Functional:**

Response time:

Since we are using AWS (Amazon Web Services), we don’t know an exact response time yet, but we are expecting that the response will take 1 second plus the time it takes for AWS to process the request. This hopefully will make it reasonably fast without noticeable delay, like any other website.

Scalability:

We are using AWS to serve our customer base. Depending on how much is paid to AWS, our scalability to support more users and operations from those users will increase. We should be able to support requests in the range of 1000s at the same time from global software engineer teams.

Usability:

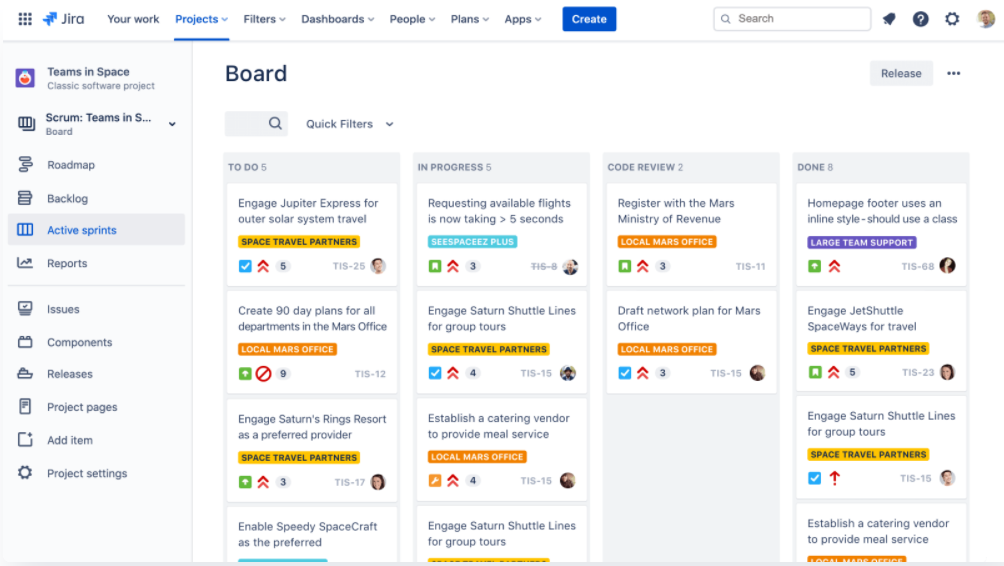
In general, we want to keep the UI and clicks to be as minimalistic and intuitive for users as possible. For example:

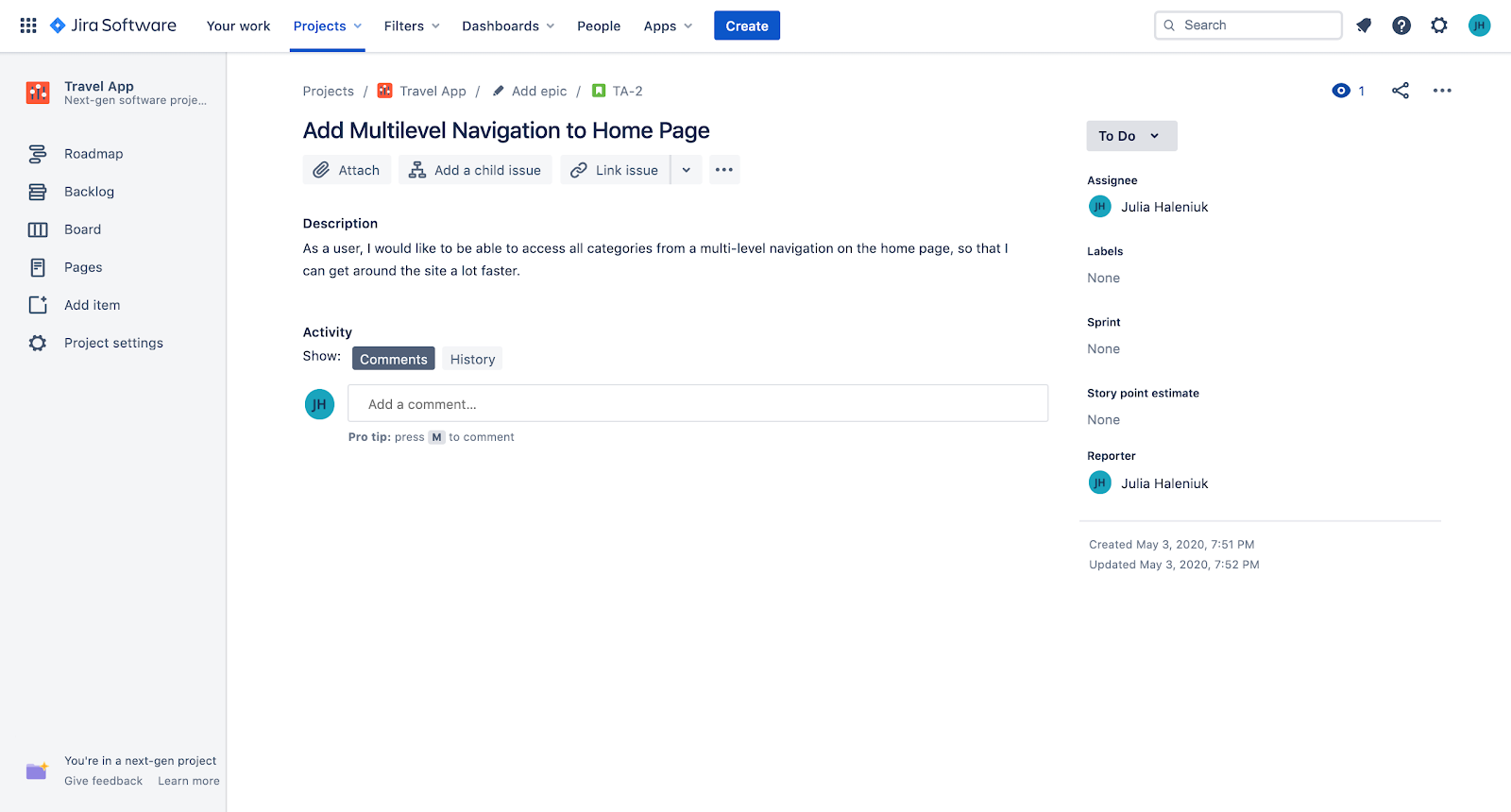
-1 click to create a ticket (click on the "+" button in the top right corner of the window)

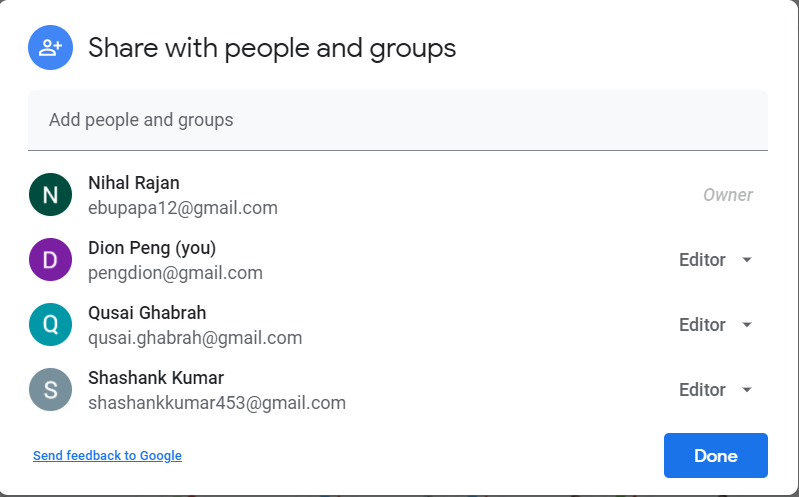
-3 clicks to delete a ticket (click on the "more" button in the top right corner of the ticket, a drop-down appears and one of the options is "delete", you get an "are you sure" confirmation and click yes

In order to create a new story, we intend on creating a “Create” Button. Viewing is possible from the board itself and a more detailed view from clicking on the story tile (Screenshot 1 and 2). In order to edit a story, you can click on the story and a more detailed panel shows up (Screenshot 2), where you can edit the details there. In order to delete, you can drag the story tile into the Trash or from the detailed view panel, click the three dots, which opens a drop-down menu, and click delete.

In order to share, there will be a “Share” button at the top of every team’s board, which will open up a panel like Screenshot 3, where we can set user roles and give people access to the board and send them an email notifying them they are added to the board.







Security:

The stories should only be viewable within the same team, so permissions to access a storyboard can be tied to your email and your manager shares the board with your email during onboarding.