ABB KANBAN BOARD

PROJECT PLAN

GROUP 1

November 16, 2017.

Introduction

ABB Ports are developing automation and electrification for container terminals. Christoffer Holmstedt is team lead for an engineering team at ABB Ports and they are in need of a digital Kanban board. They have previously used post-it notes on a Whiteboard, but the high maintenance caused the system to fall apart about a year ago.

The team at ABB are working with an application lifecycle management software called TeamForge by Collabnet. While TeamForge has its built-in kanban board, it does not fulfill their requirements. It does not offer them options for custom categories, swimlanes and is not practical to use on daily meetings. The goal with this project is to create a digital Kanban board that use data from TeamForge and visualise it in the Kanban board so that the whole team can track project tasks.

The kanban board should be available as a web application for the team at ABB. The team members need to login to access it. The board will have two types of user roles, team lead and team members. The team lead should be able to modify the board.

The tasks in TeamForge the team are working with are referred to as “artifacts”. The team lead should be able to select what artifacts that show up on the board. The artifacts should be movable using drag and drop, for moving them between different states. The states should not be stored in TeamForge, so our software need to keep the state of cards on the board.

1. Table of Contents

1. Project organization

2.1 Project group (members, contact info, roles or responsibilities)

- **André Caldegren**
  - Developer, User Interface Designer
  - Contact Info: acn15007@student.mdh.se
- **Anton Roslund**
  - Project Manager, Client Communication, Developer
  - Contact Info: Anton.Roslund@minit.nu 072-001 01 80
- **Amer Surkovic**
  - Configuration Management, Developer, User Interface Designer
  - Contact Info: a.surkovic.ba@ieee.org
- **Dzana Hanic**
  - Developer, User Interface Designer
  - Contact Info: dhc17002@student.mdh.se
- **Jawid Nasiri**
  - Developer, User Interface Designer
  - Contact Info: jni15001@student.mdh.se
- **Kevin Oswaldo Cabrera Navarro**
  - Developer, User Interface Designer
  - Contact Info: kco17001@student.mdh.se
- **Robert Duras**
  - Wiki Editor/Organizer, Developer, User Interface Designer
  - Contact Info: rds15001@student.mdh.se

2.2 Organization and communication

Communication

We are using Slack for all our communication. Slack is a messaging platform for teams. With Slack we can create channels for different topics and send private messages. We have a separate channel for announcements, mostly information from the steering group.

Organization

G Suite is used for collaborative editing of documents. It allows everyone to see and edit documents together in real-time. All changes are tracked and version controlled.

Github is used to host our code repository. Different parts are being developed in different branches. When a part of the project is done, it is code reviewed and tested by one or more persons, depending on the size of the part. The persons reviewing should not have been involved in the development of that part. When a part has passed review, it is merged into the master branch.

To track who is doing what we use Github issues. Issues are titled to give an overview of the issue. The title also includes the time spent on the issue as well as the estimated time for that issue. Time estimations are not updated after work has been started on the issue, this is because the steering groups should be able to see how wrong our estimation where. If needed an explanation can be provided in the body of the issue. Within the issue members can have a conversation about the issue, asking for help or giving status updates.

Each issue gets assigned labels to help give an overview of what the issue relates to. A label for the week the issue should be started is also assigned. For each consecutive week the issue is still open a week label is assigned. Github also allows us to assign issues to milestones. We have created milestones for each week of the project which gives us a straightforward way to view and plan our week.

Github also allows us to visualize our issues on a kanban board. We have one kanban board for our entire project. In it we have a “backlog” column, where all issues we haven’t started yet are located. When work has started on an issue it’s moved to “in progress”. Once an issue it solved it’s moved to the “review” column so that the artefact produced can be review. Finally, if it passes the review phase it is moved to “done”.

Project meetings

We have found some time where our schedules do not overlap. For all that time, we book group rooms in the school so we have somewhere to work if we need to. In the planning phase of the project we will spend as much time as possible working together in person, this is to make sure that we are all on the same page. To keep track of when and where meetings are to be held the group uses Google Calendar.

- Monday, 13:00 – 17:00
- Tuesday, 13:00 – 17:00
- Wednesday, 08:00 – 12:00
- Friday, 10:00 – 12:00

Fridays are reserved for our planning sessions where we plan our next week. During the planning, we create backlog issues for the next week and assign people to them. Assigning people in advance is not optimal, but for the course we have to do it.

On Mondays, after our steering meeting, we review the previous week and make any necessary changes to the current week. This is far from optimal. We are the first group to have the steering meeting on Mondays and we are unable to meet before the steering meeting. This forces us to do our planning on Fridays. However, much work are being done during the weekend, which changes to outcome of the week and the plan for the next.

Other routines

For time tracking we use a G suite sheet. Within it we have separate sheets for each week. Within the given week's sheet every member reports how much time they've spent per day and provide a brief description what was done. There is also a sheet with the total time spent per team member per week compared to their planned effort. Outside of time tracking there's also a sheet for absences where we take note of observed absences.

2.3 Planned effort per member for each week in the project

	45	46	47	48	49	50	51	52	1	2	Total
André	6	22	22	22	22	22	22	0	6	6	150
Anton	14	20	20	20	20	20	20	5	5	6	150
Amer	14	20	20	22	22	22	22	0	2	6	150
Dzana	14	26	13/16	21/22	22	22	20	0	5	7	150
Jawid	13	24	20	20	20	20	20	3	5	5	150
Kevin	13	20	20	21	21	21	20	0	5	9	150
Robert	16	17	17	17	17	18	18	10	10	10	150

2.4 Deliverables, deadlines and milestone

Deliverable	**Deadline**
Project plan document	November 16
First version of design description	November 30
First version of product document	November 30
Final version of product document	January 10
Final version of design description	January 10
Final version of project report	January 10

Activities	**Date**
Steering meeting #1	November 8
Steering meeting #2	November 13
Steering meeting #3	November 20
Presentation preparation	November 21
Project presentation	November 22

Steering meeting #4	November 27
Steering meeting #5	December 4
Presentation preparation	December 5
Project presentation	December 6
Steering meeting #6	December 11
Steering meeting #7	December 18
Presentation preparation	January 10
Final presentation	January 10

Key milestones	Start date	End date	Effort
Finished Project Plan Documentation	Nov 7	Nov 16	200
Finished Project Plan Presentation	Nov 17	Nov 22	50
Finished Project Design Documentation	Nov 17	Nov 30	128
Finished Project Design Presentation	Nov 1	Dec 6	50
Finished Environment Setup	Nov 20	Nov 22	80
Finished Teamforge API Integration	Nov 8	Dec 20	88
Finished Login and Logout	Nov 29	Dec 3	28
Finished Kanban board implementation	Nov 22	Dec 20	252
Finished Administrator Implementation	Nov 28	Dec 20	80
Finished Filtering Implementation	Dec 11	Dec 15	128
Finished Project Report Documentation	Dec 15	Jan 11	64
Finished Final Project Presentation	Dec 25	Jan 10	52

1) Question marks because it’s dependant on getting access to TeamForge

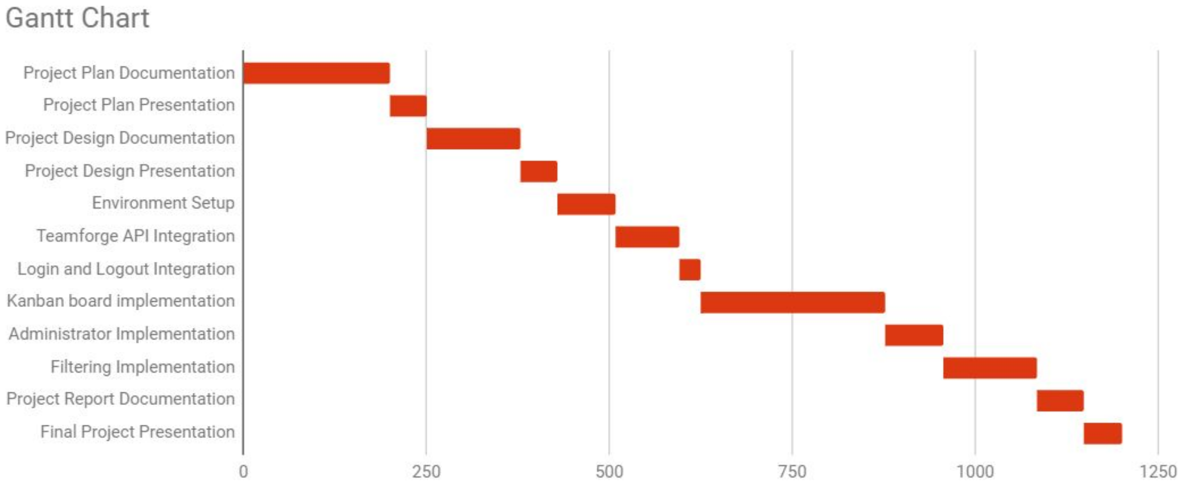

Figure 1. represents Gantt Chart based on key milestones and calculated effort.

2.5 Quality assurance (any planned activities or routines for ensuring quality, both regarding the final product and other deliverables)

“Quality assurance is a way of preventing mistakes or defects in manufactured products and avoiding problems when delivering solutions or services to customers” [1].

To assure that our product satisfies customer specific requirements we are planning to apply several methods. At this early stage, we are focused on understanding customer’s requirements.

One of the methods we are planning to use is *code review*. This method will include multiple participants within our team who are involved in the implementation of the *Kanban Board*. To make sure that this step is conducted successfully, a person responsible for reviewing will check reasonable lines of codes per day, so the person is focused and can provide valuable feedback to our team. Correcting errors at early stages tends to reduce time and cost of further development and maintaining of the system in the later stages. In addition to the code reviewing, we will also do *documentation reviewing*. That includes rereading existing documents and overviewing their state in the *Github Kanban board*.

The second method we are going to use is *black box testing*. If we have the time the product will be tested both on the client and randomly selected persons. The randomly selected person will either receive a task that they need to achieve within our system. Or they just get to have a look around and tell us what their opinion is.

The third and final method of quality assurance will be *acceptance testing*. We will take a close look on the specified requirements that were given when we started. And we will then make sure that we have achieved the requirements that were agreed upon. An evaluation of how well we managed to achieve the requirements will be presented in some form at the later stages of this project.

2. Description of the system to be developed

3.1 High level description of the domain and the problem

A Team of engineers at ABB ports currently use TeamForge by Collabnet for tracking and managing tasks for their projects. Although TeamForge offers Kanban-like boards for tracking and managing team tasks, its features are not enough to satisfy the working engineers. This led them to use post it notes for grouping tasks and managing responsibilities between them. Tasks grew in numbers so the regular whiteboard became insufficient for the team lead to track all of the project tasks. The reason why they did not use TeamForge for the actual work is that it does not give so many options in customizability of swimlanes and categories of work. TeamForge tasks are still updated but after some time passes by. They are not actually used in daily meetings.

A solution to the mentioned domain of the problem would naturally be some kind of kanban board management tool. One of the most famous ones is *Trello*. However, Trello or similar tools cannot be used because tasks need to be stored in TeamForge. This means that the solution to the described issue would be a Kanban board which has items imported from the TeamForge database used by ABB ports engineers.

3.2 Description of existing systems

This project will extend TeamForge. TeamForge is an application lifecycle management (ALM) platform for collaborative software development.

The Client uses TeamForge internally for development. TeamForge has functionality for Kanban boards, but it’s not sufficient for our client’s needs. In TeamForge, our client has artefacts which they want displayed as cards in a Kanban board. We are able to get access to the artefacts through TeamForge REST API.

We plan on developing using the free trial of TeamForge. However, we have not been able to get a free trial. When requesting a free trial you receive an email from a TeamForge Enterprise Account Specialist. We have emailing them back without getting a response.

3.3 High-level description of the desired functionality

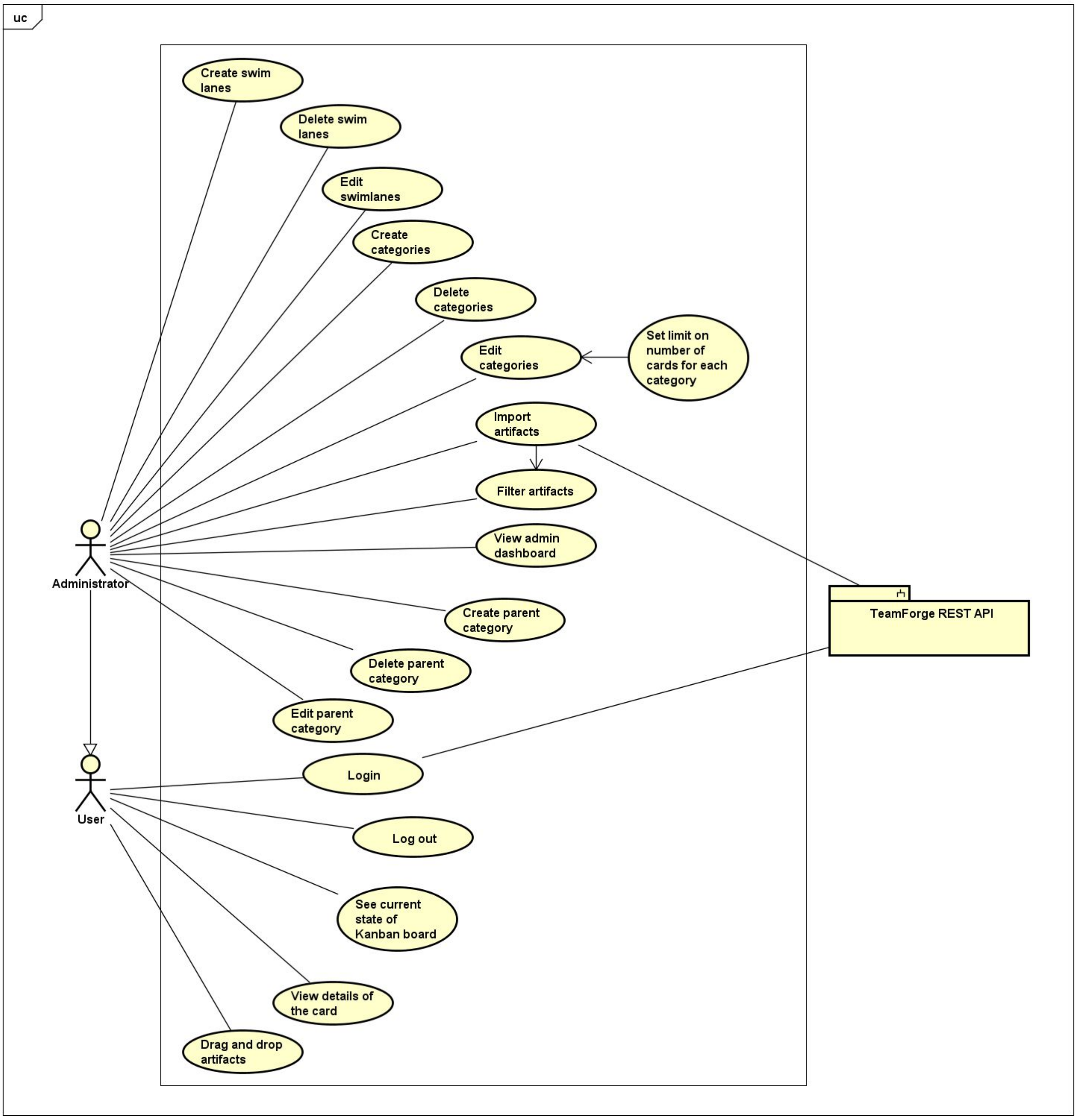

Figure 2: Use Case Diagram

*Login and logout***:** Both users and administrators must log in to access any content. Once they do, they’re able to do the following:

- *User* is able to see the current state of the Kanban board that's associated with you and drag and drop artifacts. They are also able to see details of the kanban board card.
- *Administrator* gets an access to the administrator dashboard and he is able to create/edit/remove categories and parent categories, create/edit/remove swimlanes, filter artifacts imported from TeamForge REST API as well as choose which project to import and which cards to put on the kanban board.

Kanban Board: board where the users will be able to see what team leader currently finds to the important and must be done. It's a view of the current project that the team leader has created via the admin dashboard.

*Admin Dashboard***:** board where the administrator can edit the Kanban board so the team members can see it. The available actions that the administrator can take are the following:

- *Set card limit* - They can set a limit on a selected column which restrict how many cards can be present within it.
- *Import and filter artifacts* - They can import artifacts from the TeamForge server and filter them down to the ones they want the team to see.
- *Create Category* - They can create a custom category. The custom category can contain several columns.
- *Edit/Remove Category* - They can modify or remove selected category.
- *Create Swimlane* - They can create a custom swimlane on the board allowing them to separate different subsets of cards visually.
- *Edit/Remove Swimlane* - They can modify or remove selected category.

4.4 Additional Functional Requirements

The system must gather data from the ABB Ports Teamforge server, and update it in an interactive Kanban Board. The administrator should be able to configure the kanban-board rows and columns using navbar buttons representing the models (following the model-view-controller architecture). These models are: swimlanes, parent-categories and categories. When one of this buttons is clicked, all the data should be gathered by the controllers in the system using Laravel. The data configurable by the user are name and sort-number, while the system will assign the id automatically to each of these models. This kanban board must be configurable for a single project in the first phase of the development. In later phases if there is no delays in the development, the system will be able to work for several projects in the same kanban board.

In order to start the use of the kanban board functionalities, the administrator should use the admin panel inside the select-artifacts view. In the panel, he has to request the date from Teamforge using buttons connected to the REST API (https://forge.collab.net/apidoc/swagger/). In the select artifacts view, the admin can decide which artifacts to display in the kanban board. Each artifact should contain information on person assigned, title, description, project affiliation and date. No syncs back to Teamforge should be done.

4.5 Additional Non-functional Requirements

Kanban board has to be available as a web application for the team at ABB. The web application must be able to run on Linux server (using web server like *nginx*). The system should reflect automatically the changes made by the user on real time. interaction with the system can be done using a touch screen display and it always must be available during working times of ABB. The Kanban board should be easy to understand and not be time consuming. The sort number for each category and swimlane must be configurable either by creating, editing or using the buttons in the views made by laravel-php and html.

3. Initial project backlog

4.1 Functional requirements

The following tables will show more details about functional requirements. Each table contains user story, prerequisite, input, validation concerns, processing, output, client priority, development priority, and, effort estimation measured in *person-hours*.

User (main requirements)

1. Login

User story	*As user or administrator I want to login into the system.*
Prerequisite	User is previously registered in the TeamForge CollabNet.
Input	Username and password
Validation concerns	All input data should be entered correctly into the existing input fields.
Processing	User inputs data; System validates entered data; System authenticates user with TeamForge:

	TeamForge responses based on sent data; If TeamForge authorizes user login information, access is granted to the user on web application;
Output	Confirmation about successful login or message about incorrect input data.
Client priority	
Development priority	1
Effort estimation	20

2. Log out

User story	*As user or administrator I want to logout of the system.*
Prerequisite	User is previously logged into the system.
Input	User selects log out button.
Validation concerns	/
Processing	System logs out the user.
Output	Landing page is presented to the user after successful logout.
Client priority	
Development priority	1
Effort estimation	8

3. See current state of Kanban board

User story	*As user or administrator I want to see the Kanban board.*
Prerequisite	User is previously logged into the system.
Input	/

Validation concerns	/
Processing	System presents landing page which contains Kanban board.
Output	Landing page presenting actual state of the Kanban board
Client priority	
Development priority	1
Effort estimation	144

4. Drag and drop artifacts

User story	*As user I want to drag and drop artifacts on Kanban board*
Prerequisite	User is previously logged into the system and there are artifacts on the board.
Input	Click and hold card for drag and drop.
Validation concerns	/
Processing	User clicks and holds the card; User moves the card to the wanted field; Card is moved to the wanted field;
Output	Card is moved to the wanted field
Client priority	
Development priority	1
Effort estimation	108

Administrator (additional requirements)

1. Create swim lane

User story	*As administrator I want to create swim lane on Kanban board.*
Prerequisite	User is previously logged in as an administrator.
Input	Name of the swim lane
Validation concerns	There should not be existing swim lane with the same name.
Processing	User presses button for creating a new swim lane; User enters name of the new swim lane; System creates new swim lane;
Output	System presents new swim lane.
Client priority	
Development priority	1
Effort estimation	25

2. Edit/delete swim lane

User story	*As administrator I want to edit/delete swim lane on Kanban board.*
Prerequisite	User is previously logged in as an administrator.
Input	Updated name of the swim lane.
Validation concerns	There should not be existing swim lane with the same name.
Processing	User presses button for editing selected swim lane or a button for removing it; User enters new name of the new swim lane for editing; System edits/removes swim lanes;
Output	System updates swim lanes.
Client priority	
Development priority	1
Effort estimation	10

3. Create category

User story	*As administrator I want to create a new category on Kanban board.*
Prerequisite	User is previously logged in as an administrator.
Input	Name of the category and the limit of the possible active cards for the named category.
Validation concerns	There should not be existing category with the same name. Number cannot be negative.
Processing	User presses button for creating a new category; User enters name and the limit of the new category; System creates new category;
Output	System presents new category
Client priority	
Development priority	1
Effort estimation	25

4. Edit/delete category

User story	*As administrator I want to edit or remove categories from Kanban board.*
Prerequisite	User is previously logged in as an administrator.
Input	New name of the category or the updated limit of the possible active cards.
Validation concerns	There should not be existing category with the same name. Number cannot be

	negative.
Processing	User presses button for editing or removing the category; User enters new name or the new limit of the selected category for editing; System edits/removes selected category;
Output	System updates categories.
Client priority	
Development priority	2
Effort estimation	10

5. Filter artifacts

User story	*As administrator I want to filter artifacts on Kanban board.*
Prerequisite	User is previously logged in as an administrator.
Input	Selected artifacts from the list of imported objects on TeamForge.
Validation concerns	No artifact is selected from the list.
Processing	System imports artifacts from the TeamForge; User selects artifacts; Selected artifacts are presented on the TeamForge;
Output	System presents selected artifacts
Client priority	
Development priority	2
Effort estimation	128

6. Set limit on number of cards in each column

User story	*As administrator I want to edit limit on number of cards in each column on Kanban board.*
Prerequisite	User is previously logged in as an administrator.
Input	New limit on number of cards
Validation concerns	Number cannot be negative.
Processing	User presses button for editing number of allowed cards for selected category; User inputs new limit on number of cards; System validates input; System sets up new limit on number of cards;
Output	New limit on number of cards is set up and system shows the confirmation message.
Client priority	
Development priority	3
Effort estimation	10

References

[1] *Softlets: Embrace the future*, thesoftlets.com/quality-assurance.php.