SOFTWARE ENGINEERING GROUP PROJECT: AGILE PROJECT

SELF-ASSESSMENT REPORT

As part of your Agile Project submission, you must submit a self-assessment report. One (and only one) report is required per team. Please review the Deliverables section of the Agile Project handbook for more information about how to submit this report.
To prepare your team's self-assessment report, you must fill out the shaded areas of template report accurately (and only the shaded areas. The following information is required:

- The name of your team.
- For each item in the checklist, identify whether the task has been completed.
- For each requirement/user story:
    - Have you completed the requirement in full, partially or not?
    - Your team's submission can only be credited to have met a requirement if the marker is able to find it. If the presence of a requirement may not be obvious to someone unfamiliar with the requirement, use the Where? cell to explain where the marker can find it.
    - Use the Limitations cell to declare any limitations or assumptions you have made when implementing a feature. This can help a marker assess to what extent a requirement has been met.

Team name:

Yuppiedoms

Checklist:

The team has read and understood the list of required Deliverables.	**Yes**
The team has prepared a README.md file that meets the requirements set out in the Deliverables section of the project handbook.	**Yes**
The gradle tasks clean, build, run and test all run from the project root without error.	**Yes**
The team's git repository is linked to Team Feedback and the list of commits recorded in Team Feedback is up-to-date.	**Yes**
All collaborative coding sessions (sessions where two or more members of the team worked together from a single machine) have been registered on Team Feedback by the committer in each session.	**Yes**
All team coordination meetings are recorded in Team Feedback.	**Yes**
The team's Trello board is linked to Team Feedback and the activity recorded in Team Feedback is up-to-date.	**Yes**
All members of the team understand that the Git repository and Trello board must be retained until the start of the next academic year.	**Yes**
The team has assigned ONE team member to complete submission on KEATS.	**Yes**

Requirements completion:

	User story	Done?	Where?	Limitations
1	"I must be able to open a stand-alone desktop application."	***In full***	Gradle build Gradle run	
2	"I must be able to create a new, blank Kanban board."	***In full***	Mainframe.java, Line 148.	
3	"I must be able to add a column, with a given name and role to my current Kanban board."	***In full***	BoardGui.java, Line 111. Lines 104-108 for **name** and **role**.	We assume that you can make a column without a name.
4	"I must be able to remove a column from my current Kanban board."	***In full***	Column_GUI.java, Lines 101-115.	
5	"I must be able to edit the name and role of a column my current Kanban board."	***In full***	Once the column is created, you can directly edit these on the GUI.	
6	"I must be able to add a card, with a given id, title, description and story points, to a column."	***In full***	Column_GUI.java, Lines 55-64, 87-98	
7	"I must be able to edit the title, description and story points of a card."	***In full***	CardGui.java, Lines 103-114, 153-197	
8	"I must be able to move a card from one column to another."	***In full***	Transfer folder, DragPane.java, DropPane.java	We made the assumption that moving cards would be done exclusively with drag and drop.
9	"I must be able to delete a card from a board."	***In full***	CardGui.java, Lines 122-131	
10	"I must be able to exit the application."	***In full***	Mainframe.java, Line 61.	
11	"I want to be able to view a log of all activity on the Kanban board"	***In full***	*Board.java line 60,* *Column,java line 88,*	

			Card.java line 53, BoardGui.java line 212-214	
12	"I want to be able to view the state of a board at a particular time in the project"	***In full***	BoardGui.java, Lines 198-207	*We made the assumption that in order to view the state it was sufficient to take a screenshot of the board*
13	"I want to be able to load and save a board, including its activity log, from/to persistent storage."	***In full***	*CardGui.java lines 212-221. Mainframe.java lines 168-188*	
14	"I want to be able to move cards in the user interface using a drag and drop interface."	***In full***	*Transfer folder, DragPane.java, DropPane.java*	We made the assumption that moving cards would be done exclusively with drag and drop.
15	"I want to be able to move columns in the user interface using a drag and drop interface."	*No*		
16	"I want to be able to see the overall velocity (expressed as story points per week), average lead time (expressed in weeks) and average work in progress (expressed in story points) for the project."	*No*		
17	"I want to be able to see the overall velocity (expressed as story points per week), average lead time (expressed in weeks) and average work in progress (expressed in story points) plotted over time in a line chart."	*No*		
18	"I want to be able to limit the amount of work in progress (WIP), including both a graphical and numerical indicator of the current amount of WIP as a proportion of the maximum WIP."	*No*		
19	"I want to be able to add, edit and remove swimlanes, with dedicated work in progress constraints, in the Kanban board."	*No*		

Other Assumptions Made:

- In testing, for some methods we decided that it is sufficient to test both back end and front end without having an assert case in some tests.

i.e. the SaveTest and LoadTest dont have a back-end check explicitly, but when run one after the other, the board is clearly saved and loaded from memory again.