Toyota Motor North America Executive Scrum Board

Team 2 --- Scrum of the Earth

Toyota Project Customer: Troy Morgan

University of Kentucky --- CS 499 --- Fall ’17

September 16, 2017

Authors:

Logan Hickey – editor and author of project plan, title page, table of contents, user interaction, functional requirements, non-functional requirements and appendices

Irsyad Hanif – author of introduction and project overview

Thomas Wheeler III – editor and author of system model and project plan

Jonah Starling – author of development and target environments, feasibility and metrics

Table of Contents

I. Planning and Estimating

1. Size Estimate

We have decided that we will be running 10 one-week sprints with three to five story points per week for a total of 30-50 story points depending on additional customer requests and "the twist". These story points are defined as major tasks that need to be completed such as initial user interface development to card interface features.

2. Risk List

- **Risk 1** - Database leak of critical Toyota information to competitors/ press
  - **Solution**: An extensive testing of database security ensures that it is not a weak link in the Toyota infrastructure should minimize this problem.
- **Risk 2** - Tool fails during important Toyota meeting causing data loss
  - **Solution** - Extensive testing will be put in place to eliminate potentially catastrophic errors. Also, an auto-save feature will be implemented as a stretch goal to minimize data loss due to things out of our control.
- **Risk 3** - Tool is cumbersome and decreases productivity instead of increasing it
  - **Solution** - Have several user trials with actual users at Toyota to receive feedback at multiple stages to ensure that users find the tool intuitive.
- **Risk 4** - We fail to complete the project in time
  - **Solution** - Sticking to the following project plan and good time management will keep us on track to complete the project before the due date.

3. Schedule & Resources Allocation

This section is organized into sprints. Use stories are listed below each sprint to show the high-level progress that we will try to maintain. Our project does not involve a budget and the only resource that must be properly allocated is time. Each member's management focus areas outlined here preceding the schedule. However, it should be noted that all team members will be actively working in all areas of the project.

3.1 Allocation of Management Duties

- Logan Hickey – Customer Point of Contact and Class Assignment Lead
- Jonah Starling – Lead of UI Design
- Irsyad Hanif – Lead of Backend Development
- Thomas Wheeler – Lead of Project Management and Backlog Maintenance

3.2 Sprint Schedule

Sprint 1: September 25 – September 29

-Basic front end

-basic back end

-basic database

Sprint 2: October 2 – October 6

-functional front end

-functional back end

Sprint 3: October 9 – October 13

-basic board functional

-able to move cards around

Sprint 4: October 16 – October 20

-MVP due

Sprint 5: October 23 – October 27

-backend storage

-account creation

Sprint 6: October 30 – November 3

-administrative control

-ui improvements

-card improvements and added features

Sprint 7: November 6 – November 10

-more card features

-Toyota specific scrum details on card

Sprint 8: November 13 – November 17

-trello linking

Sprint 9: November 20 – November 24

-sprint metric system

Sprint 10: November 27 – December 1

-Bug fixes and UI polishing

II. Requirements

1. Introduction

The Toyota Motor North America company currently does not have a great way of collating all of their projects into a single place. Their current solution is a whiteboard with sticky notes, which is not conducive for efficient executive meetings. Sticky notes provide little detail about the project, and a whiteboard is not functional for teleconferencing executives. The TMNA Information Systems group tasked us with creating a solution to help remedy these problems.

The project will be a web app that displays these projects. The projects will be represented as cards, and a user will be able to click on the card to obtain more information about the project. They will have access to the team assigned to it, the progress of the project, any problems the team has encountered, and so on. Users will be able to add, remove, and edit cards, and each edit will update the cards for everyone looking at the board. This main functionality will solve the problems that TMNA has with project management.

This document will go into further detail about the problems that TMNA face, our proposed solution, and how we will implement our solution. The details of our implementation include the hardware and software environments that our solution will run on, our system model, how the user will interact with our solution, functional and nonfunctional requirements, and the feasibility of our solution. This document is meant for both the customer, TMNA, as well as us, the developers. This will serve as a way for the customer to understand our solution and give us feedback, and it will give us a solid document to refer back to during the development lifecycle.

2. Project Overview

In this section, the customers, stakeholders, and users will be identified along with their needs. The problems facing the customer will be described, as well as our solution to the problems and our justification. The constraints that influence our design will also be described.

2.1 Customer, Users and Stakeholders

As mentioned in the introduction, our customer is TMNA Information Systems. The stakeholders of our project include the entirety of TMNA, as they will be affected by our project. The users will be the executives of TMNA and project leaders. The executives will use our project to discuss and gain information about in-progress TMNA projects, and the project leaders will use our project to update the status of their TMNA projects.

2.2 Problem Statement

At TMNA, the current solution for an overview of all their projects is rudimentary at best. Toyota categorizes project progress into three categories. A circle indicates a project that is on time, a triangle indicates a project that needs work or is falling behind, and an X indicates a project that is seriously behind or failing. TMNA's current solution is to create the three sections on a whiteboard and place sticky notes with the project name within the three sections. This solution hurts meetings: an executive cannot gain clarification about why a project is falling behind or failing without the project leader explaining. Also, executives cannot see the board well. During a teleconference, a person must point a camera at the board for the rest of the members who are not physically present to see.

2.3 Proposed System

Our proposed system is to take the whiteboard and place it on a website or a web application. During a meeting, the web application can be projected onto a TV or Surface Hub for discussion, or an executive can have it open on his laptop. The users can expand projects, represented as cards, and obtain more information about the project. The card overview will also change depending on the category. A card in the circle category will just display the name of the project. A card in the triangle or X category will also display why it is in that category, with cards in the X category going into more detail. Users will also be able to add, edit, and delete cards on the application. For security, a user will have an account and will be able to access only the boards he has been invited to.

2.4 Project Constraints

The justification for this solution is that Toyota requires specific details for their project management board that no other product has. When we were first introduced to the product, the development team felt like they wanted us to remake Trello or JIRA. However, as they detailed

their required functionality, we started to understand that Trello did not have all of the features that they needed. For example, the project board needed to be something that they could display on a big TV in their office. Trello and JIRA does not have this feature. Also, the three columns that represent project status is unique to Toyota, and while Trello can categorize projects, it is not built around this functionality.

3. Development and Target Environments

This section begins by presenting the challenge that Toyota's cross platform hardware systems bring to the program. This is followed by a discussion of our choice of software framework, Electron, and reasons why it will be a effective solution moving forward.

3.1 Challenge of Cross Platform Hardware Systems

Creating a large-scale app that will be used companywide at a major company presents quite a few questions and challenges from the start. First, what kind of platform will we need to develop this product for so that it can accomplish its desired goals? Second, what is the demographic of platforms that our potential users are using? Third, based off the answers to the first two questions, how do we develop the software for the platform that we saw best fit for development? These questions have a lot of weight as the final product will go into an environment that already has a working solution and our solution must prove to be not only more efficient and easier to use but also a solution that can be used by all without restrictions. When talking with the customer about what hardware platforms most of the employees used we found a varied response. Some use Mac and others use Windows, some use iPhones and some use Androids, some executive meetings use surface hubs and others use apple mirroring and some don't have a dedicated system that moves with them. With these factors in mind it was clear that we could not simply choose a single platform to develop for.

3.2 Electron Framework Solution

Looking at the goal of what the product should do it was clear the product that we are developing should be cross platform on Mac, PC, and Web. To accomplish this in a timely manner we came to the conclusion that Electron, a JavaScript based platform, would serve as the quickest and most efficient vessel for our task set forth. Electron is free to use and we have unlimited amount of memory (within reason) for our project purposes. Electron will be completely functional on any of the platforms mentioned above allowing maximum flexibility and use for executive meetings and project lead updates. This solution will offer the customer a viable product for early adoption across the board.

4. System Model

This section starts with an explanation of the current system that Toyota is using during its executive review meetings. Following is a description of how our proposed solution will provide solutions and new opportunities to Toyota.

4.1 Existing v. Proposed Solution

Toyota has engaged us to build a tool for their executives to use to monitor the status of projects. This tool previously existed only on whiteboards, and we are creating a digital solution. It will involve cards representing each ongoing project arranged into three columns: on schedule/budget, falling behind, and needs immediate intervention. The cards will contain details

on each project including links to other Toyota scrum tools for the project, but only a short overview will be initially visible to the user before expanding the card. The cards can be moved between the columns as needed by individuals assigned to the project or by upper management. This enables management at Toyota to keep a close eye on individual projects while still being able to track progress across multiple projects in the region.

4.2 System Layout

Figure 1 is system user flow diagram that represents the different ways a user can interact with our proposed system. It functions in a similar manner to the existing white board solution. The user begins at the login screen for the tool. From there the user is either taken to a selection screen to determine which board is viewed or to the account creation screen if the user does not already have an account. Toyota will handle authentication and privilege assignment. From the account creation screen, the user will be able to view all of the boards he or she has access to. Once a board is selected, the user will see all of the current projects on that board displayed as movable cards. The cards are arranged according to Toyota's current classification system. Which ranks projects that are on target, slipping behind schedule, and those that need immediate intervention. From the board selection screen the user, can adjust their settings such as which board is displayed by default. Once on the selected board, the user can click on a project card to see the expanded card view which shows a detailed description of the project, the people associated with the project, and the scrum details on the project. All of this information is pulled from the database and is push updated in real time. Also on the board display, the user can access board settings to add or remove users from the board, add or remove cards from the board, or adjust notification settings for that particular board. From the expanded card view, the user can access the settings for that particular card.

To represent the backend management of this system we have included *figure 2* as a database model to deepen our plan with the software itself.

5. User Interaction

This section is broken down to describe how each user of the program will interact with the overall system. To illustrate the web of interactions, we have developed a use-case diagram *Figure 3* is used as the basis of this discussion.

The system administrator (our team) will maintain the back end of data trafficking and UI flow. This will be done by taking feedback from executive and project management from Toyota to fix bugs, develop updates and keep functionality current. These updates will directly affect both the executive viewing board use case and the scrum project card viewing/updating use case.

The executive management team at Toyota will be the main interactive actor on the final product. During scrum review meetings, executives will be able to visualize and jump from project card to project card while on the same live board as executives that may be videoconferencing from across the world. The following success use case scenario outlines the executive managements action outline, following login and board selection, for the program.

5.1 Executive Use Case Scenario

Scope: Toyota Scrum Board

Level: user goal

Primary Actor: executive managers

Step	Actor	Action
1	System	System displays and refreshes project data every 1 minute
2	User	Exec member in KY clicks on a project card
3	System	Project card loads instantly across all opened dashboards
4	User	Comment is made and submitted on project card
5	System	Update is sent to individual project card editor for project team to receive feedback
6	User	Return button is pressed on project card.
7	System	System returns to live dashboard and refreshes project data
8	User	Executive reports a bug and a new metric for the system through report message UI.
9	System	System sends admin team an email with message.

The project leaders will be responsible for frequently updating data in the project card editor to provide an accurate measure for executive review meetings. The project leader will have a variety of parameters to control and report within the program which include but are not limited to scrum score/progress, backlogs, general comments, task distribution, and scrum sprint cycle details. The following success use case scenario outlines the project leads action outline, following login and board selection, for the program.

5.2 Project Lead Use Case Scenario

Scope: Toyota Scrum Board

Level: user goal

Primary Actor: project leaders

Step	Actor	Action
1	System	System stores and displays user data until new save is made
2	User	Project lead marks backlog item as complete and saves.
3	System	System automatically updates scrum score, and product backlog, stores new data into system

4	User	Project lead adds comment to request change to backlog and saves
5	System	System stores comment and updates executive board with symbol on thumbnail card to indicate new comment
6	User	Project lead adds backlog item. Saves. Then removes it and saves.
7	System	System adds new piece of data and automatically updates scrum metrics. New data is stored. Symbol appears on exec board to indicate change to backlog.
8	User	Project lead reports a bug and a new metric for the system through report message UI.
9	System	System sends admin team an email with message.

Specific parameters for individual project cards have not yet been defined. It should be noted that these metrics may create variations in use case scenarios for the project leads and executive managers.

Scrum managers will have a hand in the development of use case scenarios that were described above. They will train and assure that our program aligns with the scrum management process from start to finish in both use case scenarios.

Although integration may not be in the scope of this semester, it is worth noting that the program that a second UK project team is creating could become a more automated and advanced environment for project leaders to update their cards on our program.

5. Functional Requirements

This section seeks to lay the foundation for our user stories or functional requirements for our program. It begins with a description of the chart that organizes our functional requirements. Each section following is ordered by priority of the functional requirements that it contains.

5.1 Functional Requirement Organization

To better organize our functional requirements*, figure 4* was created. This chart is organized by priority, where the first and most important user stories are listed first. The farther down the chart you go the more enhanced the functional requirements become. For example, the backlog voting system is a proposed functional requirement that far exceeds the bare bones functionality of our program.

5.2 Priority 1 Functional Requirements

This section contains requirements to build the skeleton or bare bone version of our program. It includes the basic board and card UI and all associated card setup and can be visualized by *figure 4*. Within the basic UI, a user should be able to create a card, move a card or open/click a card on the board. The user should also be able to add text, edit text, delete text or edit the name of a card in the

card UI. In the backend, the database should be able to store all card names, store text associated with a card and store card position on board. The database must also link to the frontend UI to display name on card, display text on card and display cards on the board.

5.3 Priority 2 Functional Requirements

This section contains very feasible secondary requirements of our program that will follow our bare bone skeleton and are outlined in *figure 4*. These features are account creation with all associated backend admin/user management and increased text functions within a card. Within account creation, a user must be able to login, create a new account (add name, add group, request specific permissions), and be taken to the board upon successful login. The backend storage must store initial account data, alter account data, grant/remove permissions to cards and establish admin that is able to add/remove permissions. The user will have access to more specific text features on the cards. They will be able to create a list, delete a list or edit a list and the backend must be able to display and differentiate this data. They will also be able to create a paragraph, delete a paragraph or edit a paragraph of text on a card and the backend must be able to display and differentiate this data.

5.4 Priority 3 Functional Requirements

This section contains requirements that may be reached and act as additional requirements that are not necessary to develop a program that full resolves the problem statement. These requirements are also outlined in detail in *figure 4*. These features include a scrum scoring metric system which will allow the user to display current project score, link a score to associated lists/paragraphs and the backend must store these data and associate the data correctly. The other requirement is a backlog voting system which allows the user to vote on a Trello backlog item and it must be a Fibonacci number which can be saved. It is required that all members of the card to vote on the backlog score before all scores are displayed to the users with access to card. The backend must be able to form all associations between users of same card, store cards and create a new list that displays scores to card UI after all necessary scores are received.

6. Nonfunctional Requirements

This section is split into four sections that Toyota stressed as important for daily use of the program. Each section is a single broad nonfunctional requirement for our program that we narrow down in discussion.

6.1 Server Compatibility

As discussed in the development and target environments section of this paper, the variety of devices and operating systems that must be compatible with our program is vast. To overcome this obstacle, we found the Electron framework which will enable us to create a viable product. An extension of this compatibility problem is the method by which we will make our application accessible to the internal Toyota user accounts. We have a couple options to approach this problem. The first is to store the application within Toyota's server framework. This would increase security for user accounts and provide privacy for our program. This method would also potentially make user account setup more fluid with the server rights that already exist for each user. The second option is to create the application on an open web server so that individuals form an account completely aside from Toyota's server system. The obstacle here is assuring that Toyota's IT team could approve user accounts on this open application. Privacy for our application is also at risk here assuming that trademarking is a

potential option for our program. The technical requirement here is that for either option our program must be secure and accessible to all team members at Toyota that need access to our application.

6.2 Live Response Time

The use case for executive managers forms this requirement. This application will only be highly functional if it is able to refresh all data on the board automatically and in very short intervals. Managers will be actively moving and accessing the data during meetings. Suppose a paragraph is added to Team A's card in the Georgetown, KY executive meeting room during a teleconference with Toyota executives in Ann Arbor, MI. If this change is not updated across all board almost instantly, communication can become confusing and disruptive to the meeting. This application serves the sole purpose to increase fluidity of transferring information during meetings, therefore a live system is a requirement.

6.3 Optimization for Low Crash Rate

It is a requirement that our program must be heavily tested to possess a very low crash rate. Similarly, to the disruption that would be caused by slow refresh times, a buggy system would inhibit managers from communicating easily during meetings. To accomplish this our team must focus on simplifying the application which is meant to be very straight forward. It is required that we remove any unnecessary code/features that create opportunity for system crashes.

6.4 Optimization for High User Volume

It is required that our system be optimized for a massive user environment. It is required that our system be optimized for use of at least 100 users at any given time. This requirement also seeks to avoid system crashes during large and important meetings. An extension of this requirement is that our program can handle the creation of 5000 user accounts. This will provide room for our application to be integrated into Toyota's engineering centers all over the world.

7. Feasibility

Whenever undertaking a project of this size the timeline must be taken into careful consideration and needs to have weight on the features that are included in the product. Scaling the product to fit the timeline given does not give us the result of just one product design but a few. A basic product goal set, or a minimum viable product, allows us to analyze what the basic, bare bone structure and core set of features the product needs and should have. The opposite of that is the penultimate design goal set which takes the route of including all the features the client would like to have as well as others that we would like to include to increase the usability and functionality of the product. This section seeks to attribute which features are absolutely necessary for a functional program within the time that we have to complete the program and which enhanced features could be added if time permits.

7.1 Bare Bone Version

The bare bones version of our product will include the main board with the ability to create, edit, move, and delete cards. Cards will have basic functionality like the ability to add text and links to other important documents and outside material. There will also be basic user management system put into place which will manage access to individual project cards and the main board.

7.2 Enhanced Version

The enhanced version of the product includes having some of the outside features, that the user would normally link, to brought in as built in features. We would also add a voting system for cards and collaboration area on boards and cards to allow the teams to work together smoothly and asynchronously. The enhanced version will also have more support features, management features, and board management that allows the assigning of roles to users. With these two goal sets we will have no problem with our timeline and the inherent flex it will have.

8. Conclusion

A project board is vital for top-level executives. When meetings facilitate without up-to-date knowledge of your projects, the effectiveness of that meeting diminishes. TMNA's solution, while somewhat effective, does not provide on-demand information or up-to-date project statuses. Our project board solution will always provide the most accurate, up-to-date information from the project leaders and allow for more informed business decisions. It will also aid an executive in understanding every project setback and decide whether to continue the project. These decisions will decide the fate of a company.

9. Appendices

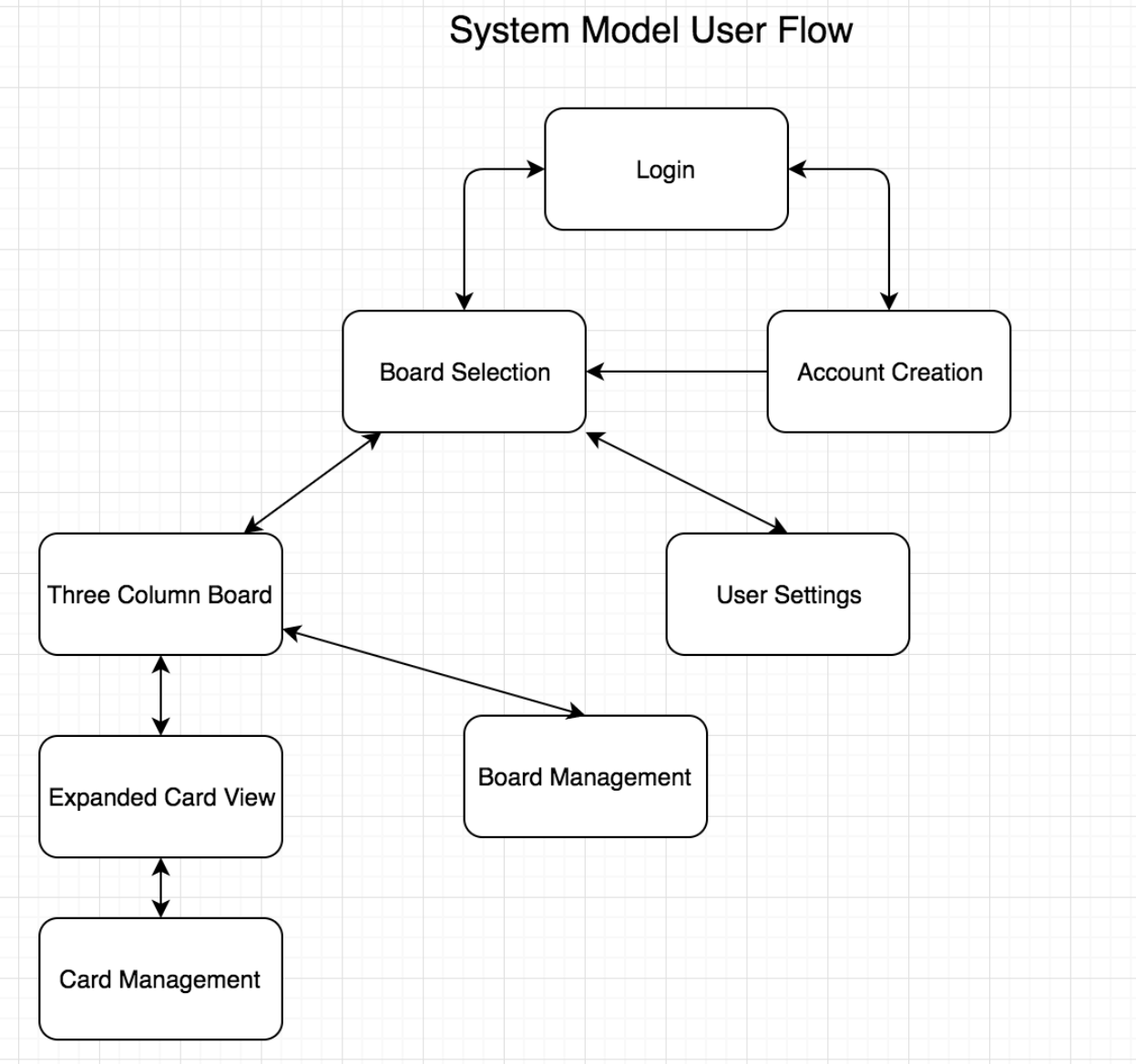


Figure 1 System Model User Flow

This diagram is representative of the system user flow that will be used to guide a user throughout the entire program. It is a high level representation of how the software will be organized to solve Toyota's problems effectively.

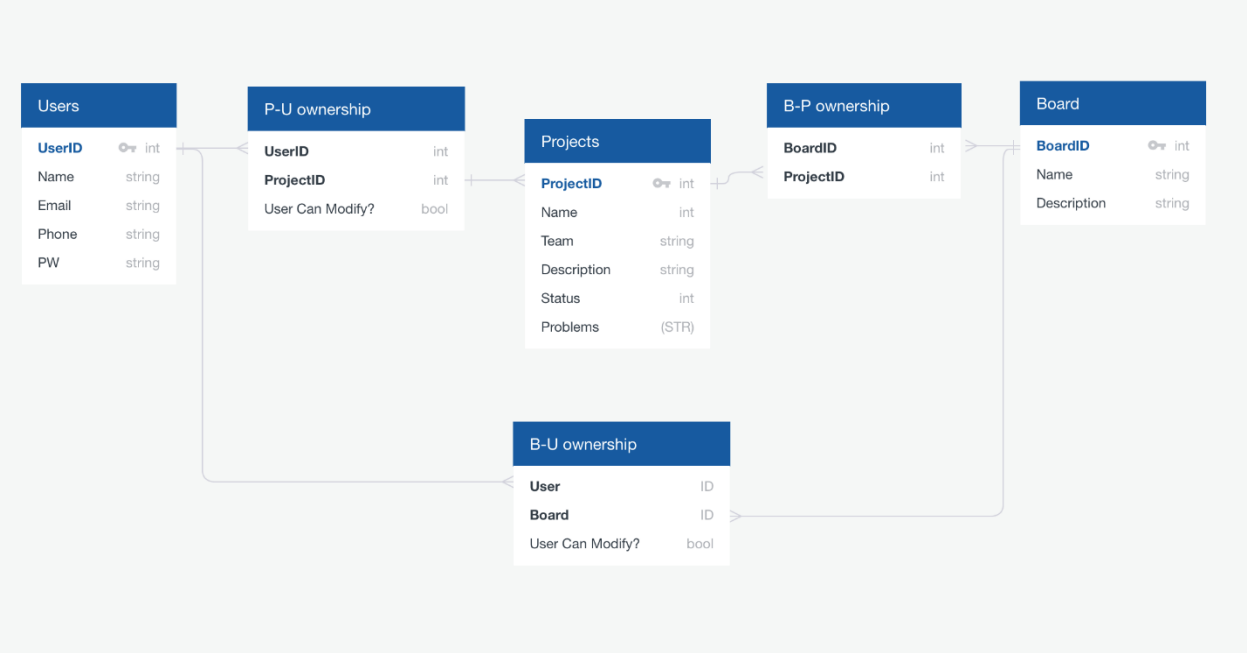


Figure 2 Backend Database Model

This diagram outlines all the variables that will need to be stored on servers to make our program functional. It also shows with arrows how they will need to be linked together to form permissions, user accounts, related text to cards and so on. The P-U, B-U and B-P ownership headers indicate the web of permissions and connections that will be made to control permissions and the network of team cards.

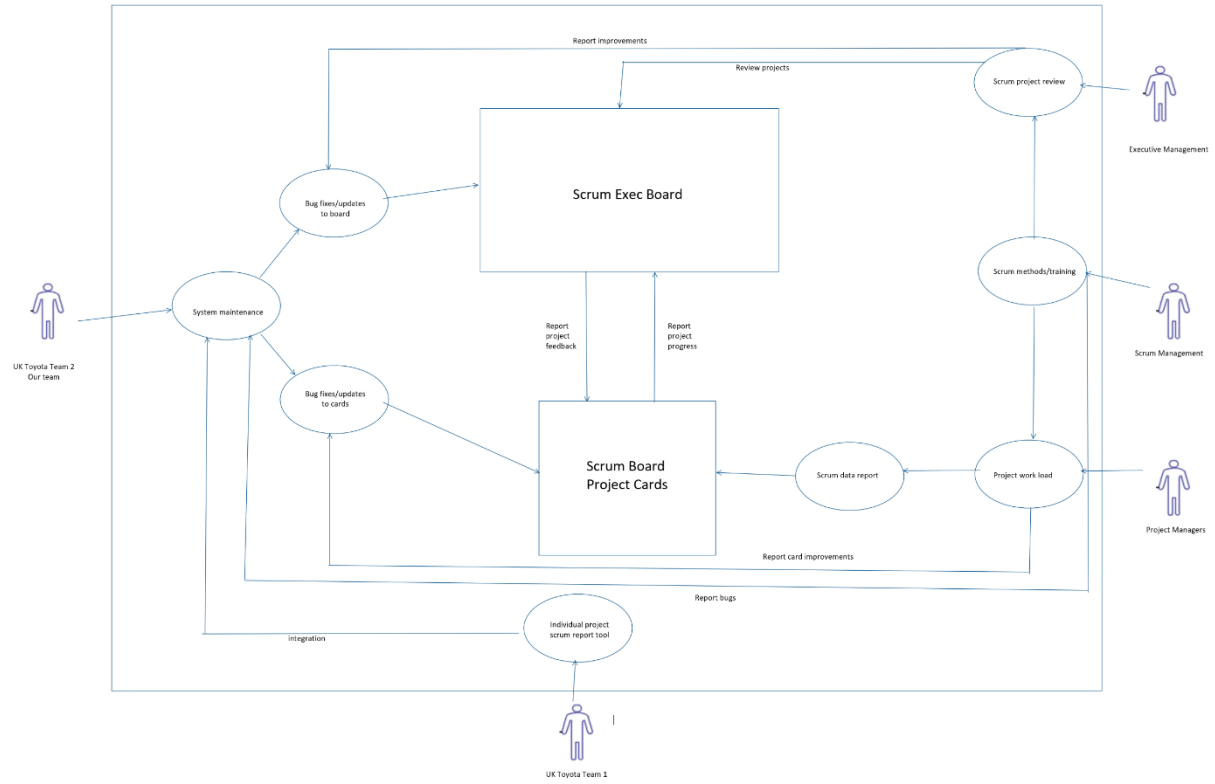
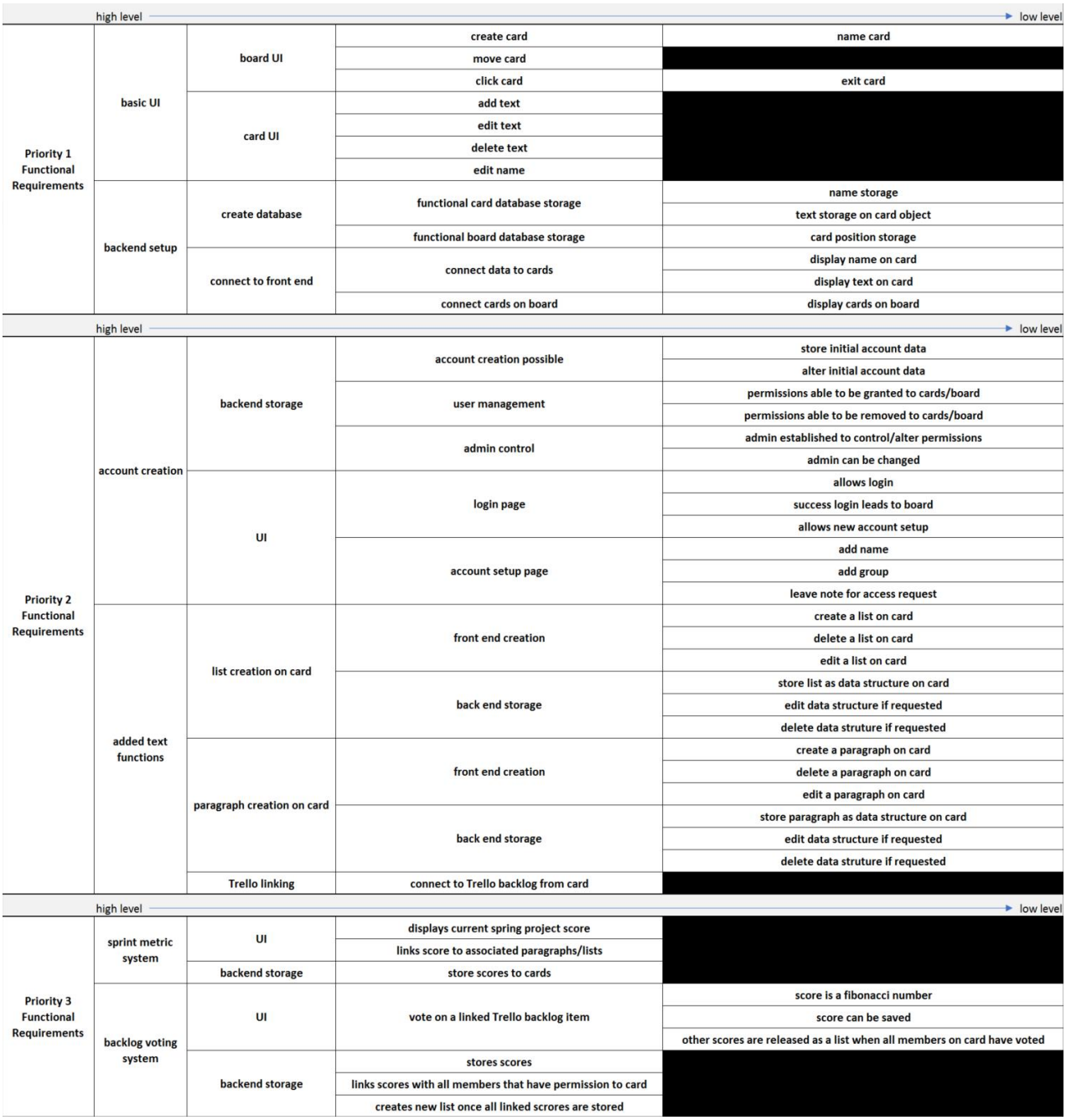


Figure 3 Use Case Diagram

This diagram represents the two main use case scenarios that our program will feature along with three other potential actors in the program. The two main actors are the project leaders and the executive managers which will be directly adding, subtracting and manipulating data in the program from the UI. Our team will act as the admin that fixes bugs and adds improvements to the database and UI flow. Other actors will be scrum managers and the other UK team working with Toyota for a scrum project. The scrum managers will maintain scrum policy within the program indirectly through contact with the executive and project managers and the other UK team may form a product that can be integrated into our program.



high level → low level				
Priority 1 Functional Requirements	basic UI	board UI	create card	name card
			move card	
			click card	exit card
		card UI	add text	
			edit text	
			delete text	
			edit name	
	backend setup	create database	functional card database storage	name storage
				text storage on card object
			functional board database storage	card position storage
		connect to front end	connect data to cards	display name on card
				display text on card
			connect cards on board	display cards on board

high level → low level				
Priority 2 Functional Requirements	account creation	backend storage	account creation possible	store initial account data
				alter initial account data
			user management	permissions able to be granted to cards/board
				permissions able to be removed to cards/board
			admin control	admin established to control/alter permissions
				admin can be changed
		UI	login page	allows login
				success login leads to board
				allows new account setup
			account setup page	add name
				add group
				leave note for access request
	added text functions	list creation on card	front end creation	create a list on card
				delete a list on card
				edit a list on card
			back end storage	store list as data structure on card
				edit data structure if requested
				delete data struture if requested
		paragraph creation on card	front end creation	create a paragraph on card
				delete a paragraph on card
				edit a paragraph on card
			back end storage	store paragraph as data structure on card
				edit data structure if requested
				delete data struture if requested
		Trello linking	connect to Trello backlog from card	

high level → low level				
Priority 3 Functional Requirements	sprint metric system	UI	displays current spring project score	
			links score to associated paragraphs/lists	
		backend storage	store scores to cards	
	backlog voting system	UI	vote on a linked Trello backlog item	score is a fibonacci number
				score can be saved
				other scores are released as a list when all members on card have voted
		backend storage	stores scores	
			links scores with all members that have permission to card	
			creates new list once all linked scrores are stored	

Figure 4 Functional Requirements

The following chart gives a high-level view of all the specific requirements that our product must meet. It is prioritized in alignment with our production schedule. All requirements presented will also be used as potential user stories for our scrum sprints and product backlog.

III. Metrics

a) We estimate that we will have roughly 30-50 story points for our program depending on how many additional requests our customer makes during spring progress/review.
b) Product Effort: We currently have all put forth roughly 12 man-hours each outside of class. This includes product planning meetings, documentation, website building, travel to Toyota and all communications. This accumulates to a total product effort of 48 hours thus far.
c) We do not currently have a real product size or any current defects in our program.

IV. Site URL/Developer Notebook

1. Project Website URL:

https://github.com/irsyadhanif/CS499Team2Wiki/wiki

2. Effort Log:

Travel and Meeting at Toyota…… 3 hours

Planning Meetings…….. 4 hours

Website Building…….. 1 hour

Documentation (this assignment)…….. 4 hours

---

12 hours total

**All members followed this same effort schedule up to this point in time.

**We do not currently have any development notes or necessary assumptions.

**All current design decisions were reported in requirements section of this assignment.

3. Team Meeting Log

Meeting with Toyota – August 31, 2017 – 3 hours – All members present

Team Planning Meeting – September 4, 2017 – 2 hours – All members present

Requirements Meeting #1 – September 14, 2017 – 1 hour – All members present

Requirements Meeting #2 – September 16, 2017 – 1 hour – All members present