Joining the No-code Low-code Revolution:

Creating an Online Educational Simulation with No Software Development Expertise

Rahul Kapoor and Jaclyn S. Woodward[1]
The Wharton School, University of Pennsylvania
June 4, 2018

Introduction

Businesses in every industry are looking to transform themselves through digitization. However, the proliferation of digitization is creating significant challenges for IT organizations. IT managers are facing increasing pressure in an environment of digital transformation, with rapidly changing technologies but with scarce development resources. This bottleneck has created a need for a new approach to software development, one that opens the doors for a wider range of stakeholders to participate in the digital transformation of their organizations. Enter no-code/low code --- a disruptive trend that aims to democratize software development, and save businesses time and money.

No-code/low-code (NCLC) software development tools seek to enable businesses to keep up in a rapidly changing technology landscape by enabling software development with little or no coding. Although the terminology in this emerging paradigm is not well established, NCLC is generally considered an important part of the movement towards high-productivity rapid application development (RAD). In order to be considered RAD by industry analysts such as Gartner, the tool must emphasize a "model-over-code" development style, but this does not necessarily entail little or no coding. [2] Gartner forecasts that 'citizen developers,' non-IT employees who develop business applications generally leveraging some type of RAD tool, will become a highly impactful trend in the application development domain within the next year.[3] NCLC tools are positioned to empower citizen developers, and make businesses more agile and flexible in the face of increasing digitization.

There are several classes of NCLC tools. A class of NCLC tools focuses on the development of simple mobile applications. Specifically, these tools help develop mobile application screens faster than traditional methods while the rest of the application is developed using conventional coding methods. Another class of NCLC tools evolved from a SaaS application platform that originally solved a specific business problem (e.g., CRM or BPM) into an NCLC tool. The final class includes tools that were designed from the inception to be platforms that provide the capability to build

[1] Correspondence to: Rahul Kapoor, The Wharton School, University of Pennsylvania, Philadelphia, PA 19104, U.S.A. E-mail: kapoorr@wharton.upenn.edu. The authors gratefully acknowledge the valuable guidance and support from Chris Gali, Meredith McGrath, Nitin Chhabra, Vijay Ravi of Graphite GTC towards this project.
[2] Mark Driver, Jason Wong & Mike West, Gartner, Inc., "Market Guide for High-Productivity Rapid Application Development Tools," May 24, 2016.
[3] Mike West, Nathan Wilson & Keith James Mann, Gartner, Inc., "Hype Cycle for Application Development, 2017," August 7, 2017.

applications without the need for conventional coding. There is an important difference between the last two classes of NCLC platforms that are often overlooked during the decision making process. In the former class, if the application being developed does not fall into the business domain suited for the SaaS platform, it is likely that most of development will fall back to conventional coding. However, the latter class of NCLC tools are much more versatile in terms of the applications being developed.

Project Overview

The main drivers of value for NCLC tools are to increase access to software development beyond the traditional IT organization, and to save businesses time and money. Because NCLC is an emerging paradigm, it is difficult to know exactly how well NCLC tools have performed with respect to these goals. However, it is clear that the current over-burdened IT organizations can benefit immensely from this emerging paradigm.

The goal of this project is to explore the utility of NCLC tools for developing complex software applications such as a web-based classroom simulation for business school students. Simulations have become an increasingly important pedagogical instrument in higher education. Educational simulations usually consist of an interest-based, fabricated learning environment where decisions are made by participants and real-time feedback is given. They are an effective way to build critical thinking skills, especially in complex situations.[4] However, educational simulations are expensive and time consuming to create with the cost of development running in tens of thousands of dollars (often more than hundred thousand dollars), and taking as much as a year to be commissioned in the classroom. Thus, the aim of this project is to explore the extent to which the NCLC approach might help lower the barriers for educators interested in creating a highly enriching simulation experience for the students. While many educators invest significant resources in developing and writing case studies, only a handful have pursued the development of their own simulations typically through collaboration with specialized application development firms such as Forio. NCLC tools have the potential to democratize the development of classroom simulations by empowering the educators to build their own simulations at a fraction of the cost and development time, and at the same time allow more students to benefit from the simulation-based learning.

The objective of the simulation, titled Platform Strategy Simulation (PSS), is to teach participants how to create and capture value in a platform-based ecosystem. At the most basic level, participants in the simulation are divided into teams, each team representing a different 'platform,' and each of the teams deciding how to compete for consumers on one side and service providers (e.g., developers) on the other side by charging each of them a price for using the platform. The goal is to maximize profits in this competitive marketplace. PSS offers participants the opportunity to work in teams to create and modify their strategic approach through a series of rounds.

The core team for building the simulation included a Wharton School faculty member and an undergraduate student (Research Assistant), majoring in Marketing and Management, both with no

[4] Kathi Lovelace, Fabian Eggers & Loren R. Dyck, The Academy of Management Learning and Education. "I do and I understand: Assessing the utility of web-based management simulations to develop critical thinking skills," March 2016. Margaret Hilton, Academy of Management Perspectives, "Skills for Work in the 21st Century: What Does the Research Tell Us?," November 2008.

qualifications or experience in software development. The NCLC tool used for the development of the PSS simulation is Graphite Studio created by Graphite GTC. Graphite Studio was chosen since it provided the capability to build applications for any business domain, and was not confined to only mobile or SaaS-based applications. The undergraduate student was primarily responsible for building the simulation, with the simulation design and requirements provided by the faculty member. The key steps in building the simulation included learning to develop software applications using Graphite Studio, translating the simulation design and requirements into software specifications, building the simulation on Graphite Studio, and, finally, running the pilot. The core team was supported by Graphite Studio, most notably a Graphite Studio developer who had five years of experience in software development but only 5 months on Graphite Studio when this project was initiated. Exhibit 1 details the overall simulation development flow highlighting the key steps and tasks, the number of hours spent, and the involvement of the Graphite GTC staff in each step.

Project Implementation

Learning to Use Graphite Studio (NCLC Tool)

During the initial orientation, Graphite GTC staff presented the logic of the different elements in Graphite Studio. For example, the staff explained the underlying concepts behind object design. This background information was very valuable once the student began working with the tool. Graphite Studio separates the front-end from the back-end elements of software development. On the front-end, a preview of the user interface (UI) is updated real-time as the user is creating it, making it easy to show to others and collaborate on changes. On the front-end and the back-end, logical processes within the application are created using a flow chart styled process designer. These flow charts include different types of functionalities that are defined in Graphite Studio and are able to be dragged-and-dropped as needed within the simulation routine.

Graphite GTC provided the student with a comprehensive training document, which included detailed instructions on multiple practice projects using the tool. The practice projects in the training document became increasingly more complicated and time consuming. These initial projects covered many of the tool's capabilities and the basic skills necessary to start building the simulation. One example of a practice project was a web application where a user could add, edit, and delete contacts. At first, the training document seemed very intimidating, but over time, the student realized how easily application development on Graphite Studio could be broken down into manageable steps. After completing several practice projects, using the tool became intuitive enough that the student felt ready to start developing the simulation.

Creating Simulation Specifications

Once the basic training on Graphite Studio was completed, the focus moved onto the process of creating PSS. There was a significant amount of planning involved in translating the simulation design into a multi-round interactive web application in which several participants provide their platform pricing strategies and observe outcomes in terms of profits and market growth. Graphite GTC staff provided the knowledge and skills necessary to create a comprehensive plan of exactly how to build the simulation. They also emphasized how important it is to take the time to plan and how often this step gets neglected among software developers. The student found this planning time to be invaluable and, when the time came to start building PSS, it became clear how much easier it was. Especially with an NCLC tool, once the planning is completed, the actual application development happens very quickly. Hence, it was extremely helpful to have the plan clearly outlined prior to the actual application development on Graphite.

The planning was broken down into two categories: (1) Functional Specifications and (2) Technical Specifications and Wireframe Design. It was most effective to start by thinking more broadly about what the simulation was meant to achieve, what kind of user interfaces would work best to achieve these goals, and how a user would move through the simulation. Through close examination of the simulation guidelines and many conversations between the student and the faculty member about the goals, the Functional Specifications document that included a very comprehensive plan of all of these aspects was put together. Simultaneously, a user interface (UI) Specialist at Graphite GTC helped create a wireframe that captured the vision of the simulation that was detailed in the Functional Specifications. Having a specialist help teach the student best practices around UI design was very valuable. This wireframe served as a reference point throughout the entire development process.

After completing the Functional Specifications, the next step was to create the Technical Specifications. The Technical Specifications included the object model needed to create the data model for PSS. The student was trained on designing the object model during the initial learning phase. The purpose of these specifications was to identify the exact means by which the functionalities required in the simulation flow will be built using the Graphite Studio tool.

Building Simulation

Once the detailed plans were completed, it was time to start building PSS using Graphite Studio. The Graphite GTC developer helped with some of the more complex tasks. The first screens to be built, Administrator Home and Create Game, were the ones most similar to the Manage Contacts practice project. Exhibit 2 shows some of the screenshots from Graphite Studio while the simulation was being built. Once the initial screens were completed, the student branched out to learning new skills in Graphite Studio, such as building tables and graphs, which were especially important for the participants and the faculty member (the administrator) to observe the outcomes of the different participant strategies. Overall, the administrator and participants interact with the simulation through 15 different screens, samples of which are provided in Exhibits 3 and 4. The Graphite GTC developer helped the student understand how to structure the backend logic of the project, and reinforced the best practices in building a software project.

By the time, PSS was developed, 15 different screens and the equivalent of 218,171 lines of codes had been created in about 100 hours of work by the student with not a single line being explicitly coded. PSS also included 9 different graphs and 6 different types of tables to portray the results of the simulation over the different rounds. In under one month, the entire simulation was built and ready for a soft launch.

Soft Launch and Future Improvement

A pilot run was attempted with a group of senior executives in an Executive Education program. It was an extremely successful pilot, with several participants mentioning the importance of the simulation to their overall learning experience. By leveraging the power of Graphite GTC's NCLC tool, the faculty member was able to offer an extremely compelling simulation experience and improve the participants' understanding of the complex dynamics underlying a platform ecosystem strategy. In future, the faculty member plans to introduce additional functionalities in PSS, and expand the learning opportunities for students with respect to strategies in platform-based ecosystems.

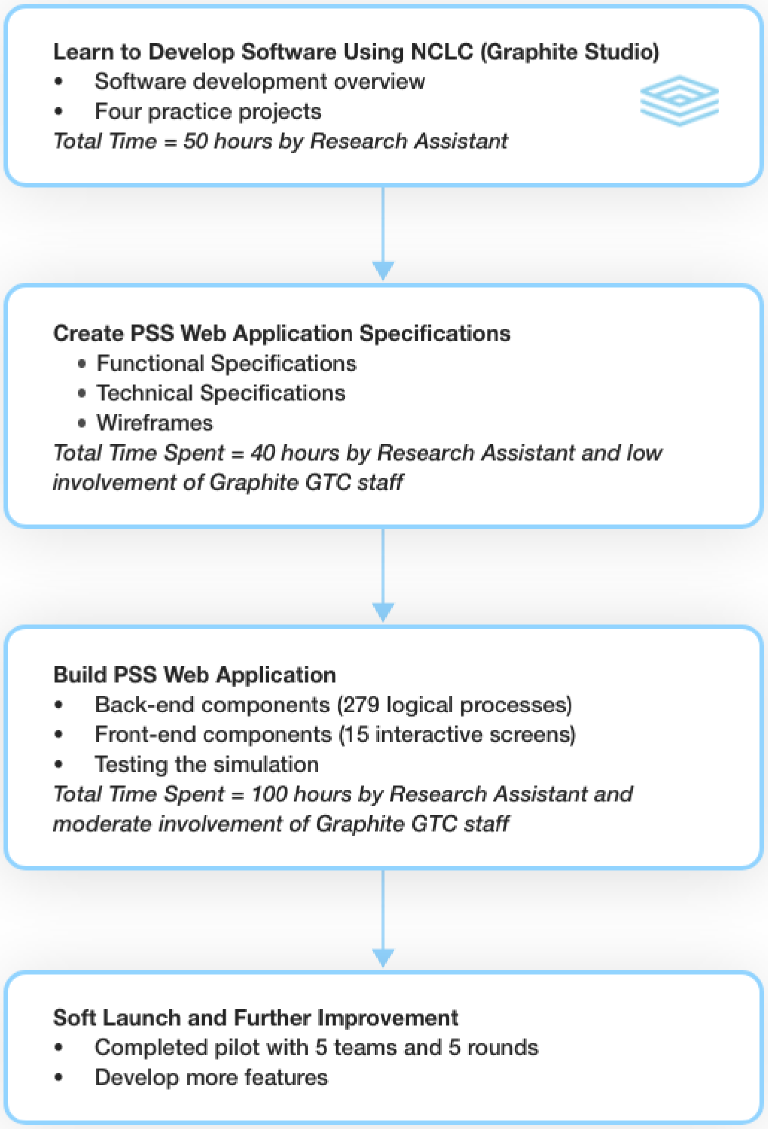
Reflections on NCLC Experience

The objective of this project was to understand the utility of an NCLC tool like Graphite Studio for developing a complex software application such as a web-based educational simulation to be used in a classroom. There are three main ways how the NCLC experience satisfied this objective: (1) it created an approachable means for a person with no experience in software development to build a high-quality complex web application, (2) it offered a significantly lower cost and faster approach to developing web applications without requiring full-time involvement of experienced software developers, and (3) it allowed the faculty (i.e., the business process owner) and the student (i.e., the developer) to communicate and iterate on a regular basis than what might have been possible in the traditional software development process.

NCLC tools have the potential to democratize the software development process, and the successful development of the simulation is an important case in point of the impact that it can have on businesses. The simulation was built by an undergraduate student, with no experience or qualifications in software development. The student was able to learn the NCLC tool and build a web-based application involving 15 interactive screens and 279 logical processes in less than 200 hours, with not a single line being explicitly coded. From the standpoint of the faculty member who was managing this project, the NCLC approach also enabled close collaboration with the student throughout the development process. It offered more opportunities to share updates during the course of the application development, which allowed the application development process to benefit from constant feedback and reduced the likelihood of miscommunication.

In sum, the NCLC tools stand to democratize the application development process, and save businesses that are undergoing digital transformation, significant amount of time and money. This project provides a valuable case study of the enormous potential of the NCLC tools, and the important purpose they can serve in the environment of increasing digitization.

Exhibit 1: Platform Strategy Simulation (PSS) Development Process using NCLC Tool



Learn to Develop Software Using NCLC (Graphite Studio)

- Software development overview
- Four practice projects

Total Time = 50 hours by Research Assistant

Create PSS Web Application Specifications

- Functional Specifications
- Technical Specifications
- Wireframes

Total Time Spent = 40 hours by Research Assistant and low involvement of Graphite GTC staff

Build PSS Web Application

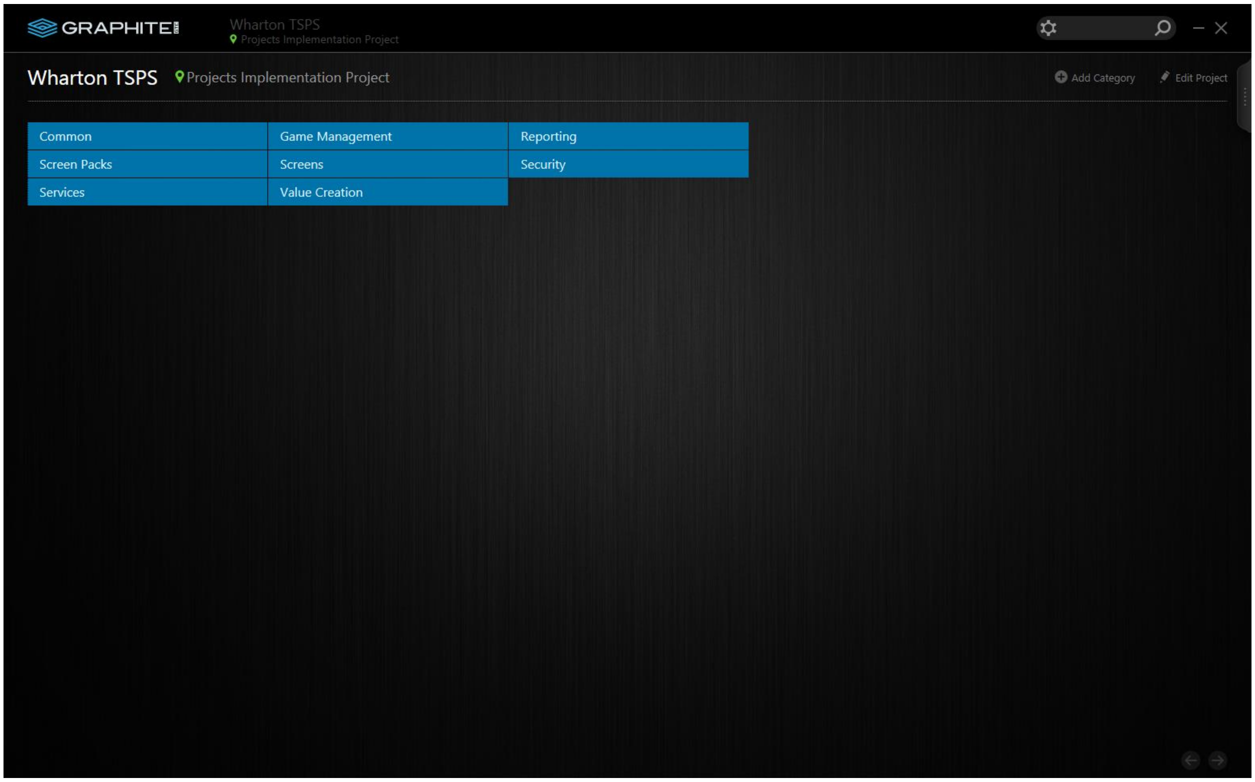
- Back-end components (279 logical processes)
- Front-end components (15 interactive screens)
- Testing the simulation

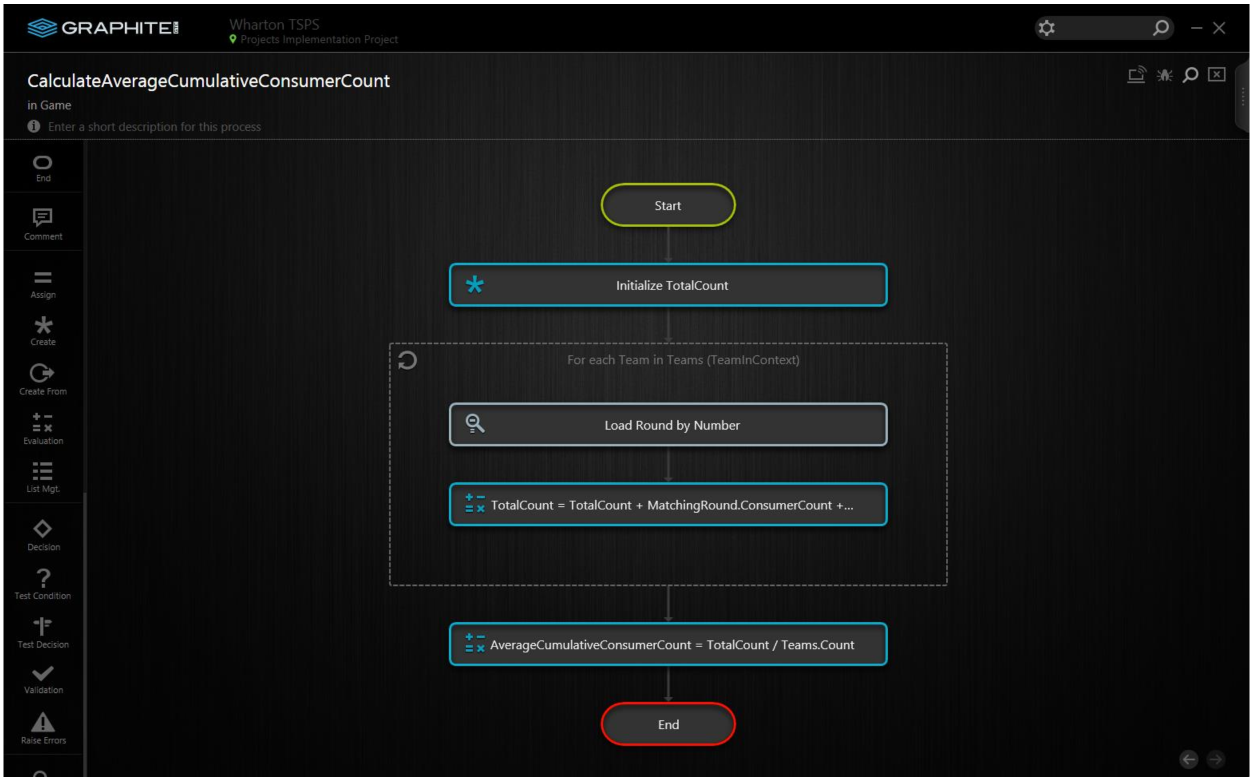
Total Time Spent = 100 hours by Research Assistant and moderate involvement of Graphite GTC staff

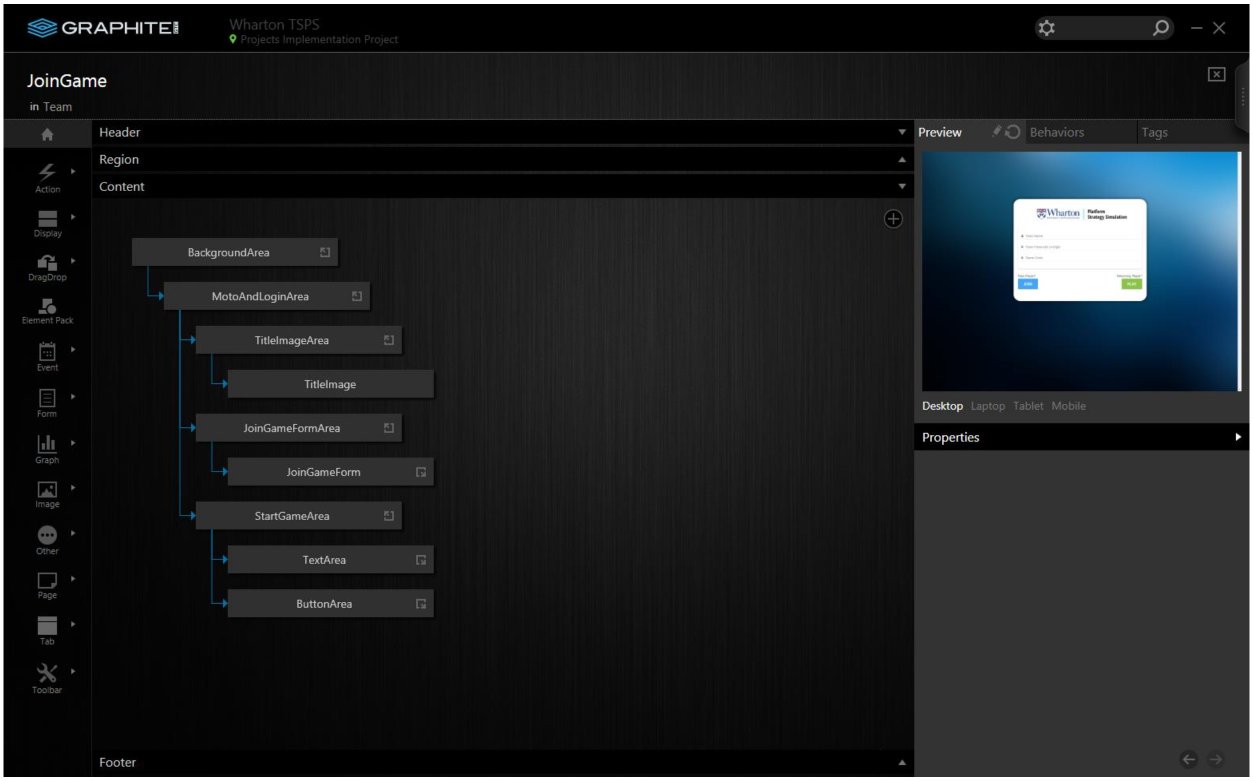
Soft Launch and Further Improvement

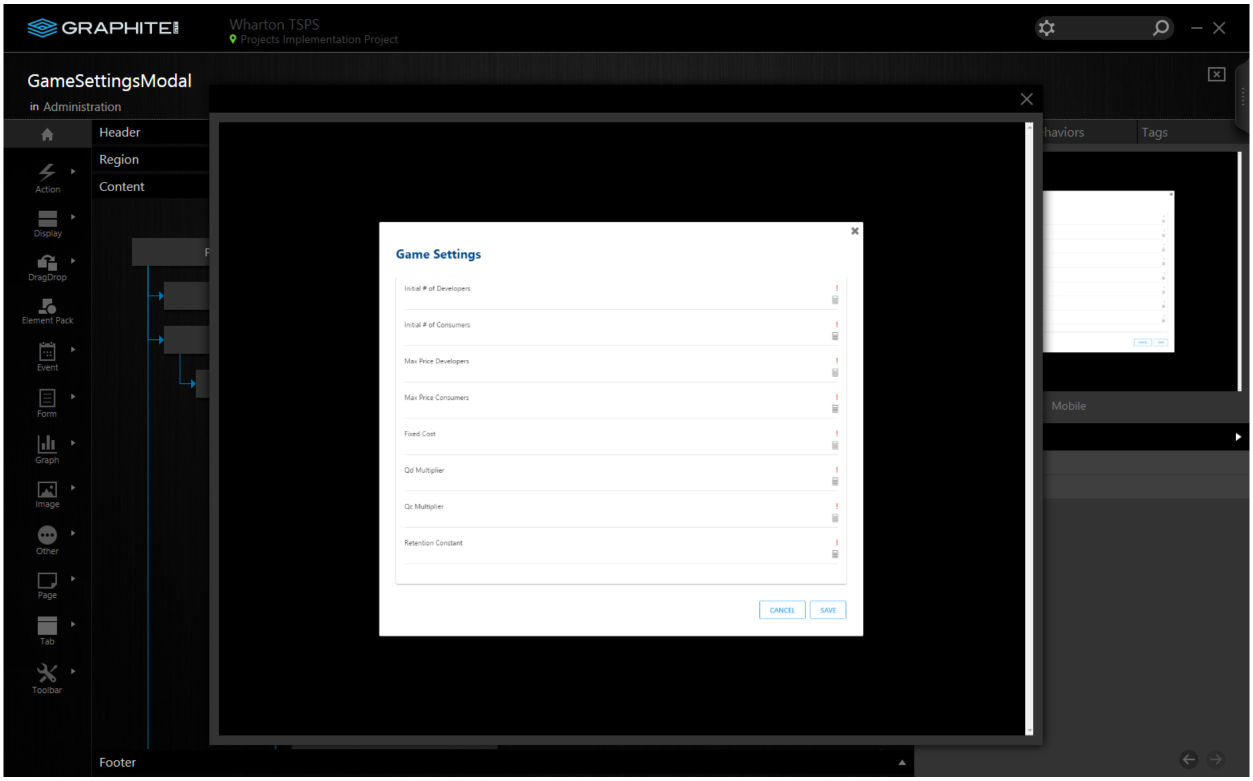
- Completed pilot with 5 teams and 5 rounds
- Develop more features

Exhibit 2: Developing Platform Strategy Simulation in **Graphite Studio**




GRAPHITE
Wharton TSPS
Projects Implementation Project
CalculateAverageCumulativeConsumerCount
in Game
Enter a short description for this process
End
Comment
Assign
Create
Create From
Evaluation
List Mgt.
Decision
Test Condition
Test Decision
Validation
Raise Errors
Start
Initialize TotalCount
For each Team in Teams (TeamInContext)
Load Round by Number
TotalCount = TotalCount + MatchingRound.ConsumerCount +...
AverageCumulativeConsumerCount = TotalCount / Teams.Count
End





Game Settings

Initial # of Developers

Initial # of Consumers

Max Price Developers

Max Price Consumers

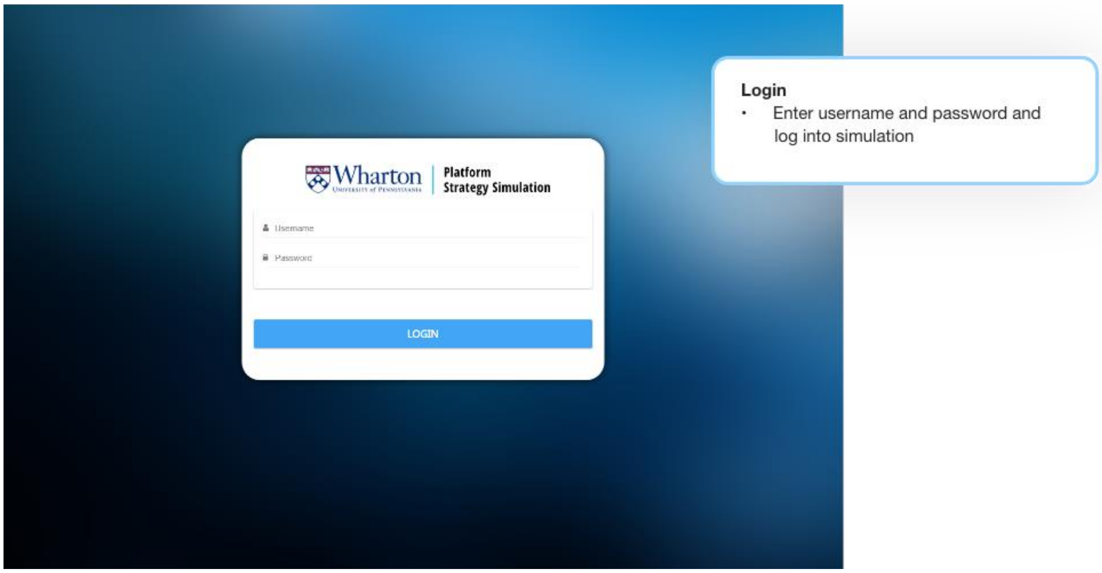
Fixed Cost

Qd Multiplier

Qc Multiplier

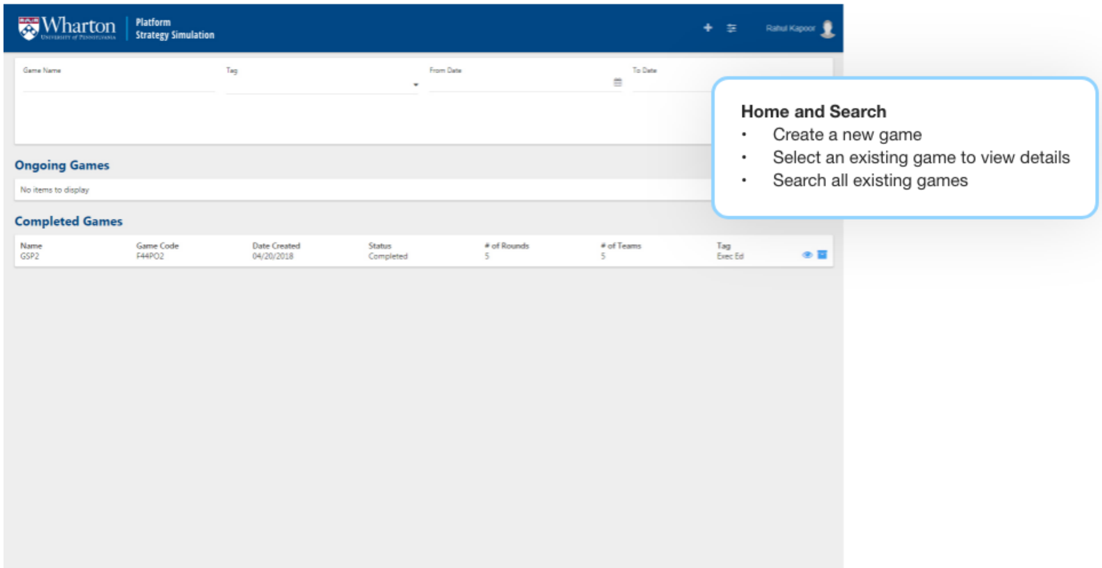
Retention Constant

Exhibit 3: User Interfaces for Administrator


Wharton University of Pennsylvania
Platform Strategy Simulation
Username
Password
LOGIN

Login

- Enter username and password and log into simulation


Wharton University of Pennsylvania
Platform Strategy Simulation
Rahul Kapoor
Game Name
Tag
From Date
To Date

Ongoing Games

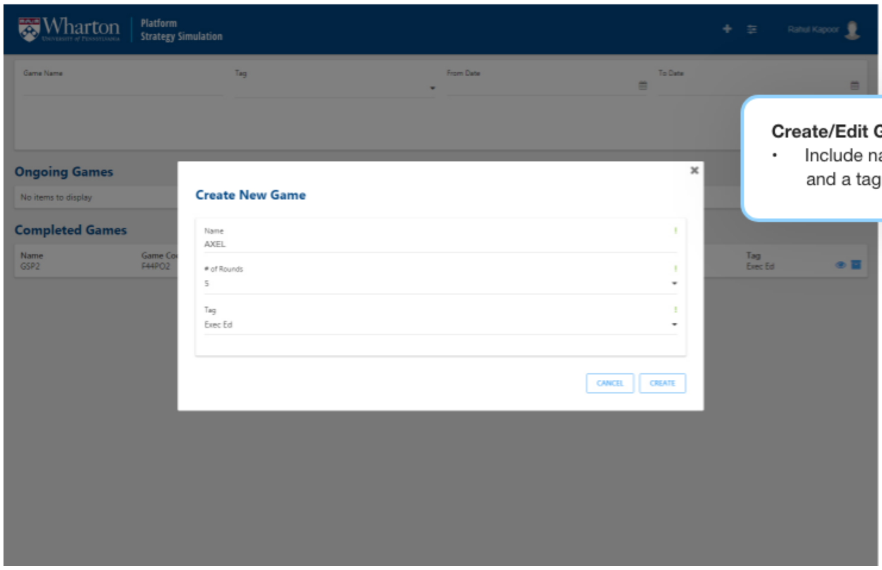
No items to display

Completed Games

Name	Game Code	Date Created	Status	# of Rounds	# of Teams	Tag
GSP2	F44PO2	04/20/2018	Completed	5	5	Exec Ed

Home and Search

- Create a new game
- Select an existing game to view details
- Search all existing games



Game Name	Tag	From Date	To Date

Ongoing Games

No items to display

Completed Games

Name	Game Co	Tag
GSP2	F44PO2	Exec Ed

Create New Game

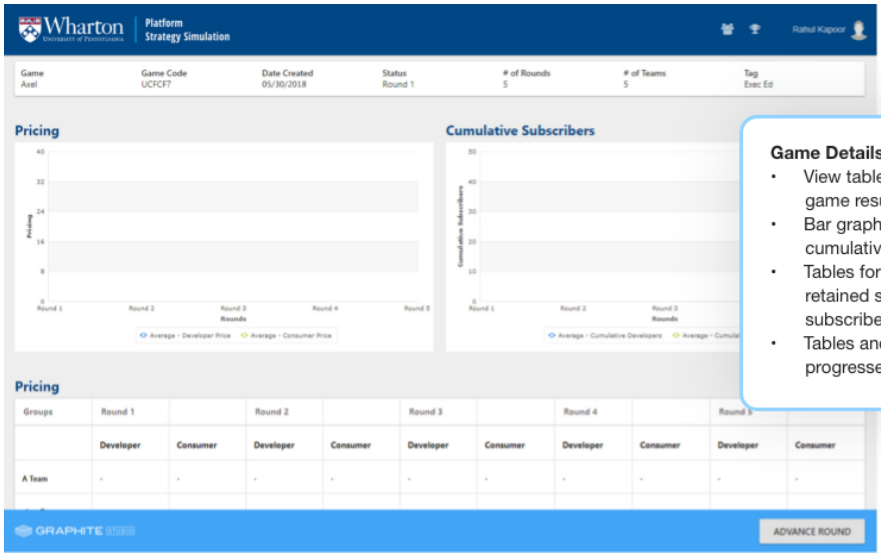
Name
AXEL

# of Rounds
5

Tag
Exec Ed

Create/Edit Game

- Include name, number of rounds, and a tag (improves search capability)



Game	Game Code	Date Created	Status	# of Rounds	# of Teams	Tag
Axel	UCFCF7	05/30/2018	Round 1	5	5	Exec Ed

Pricing

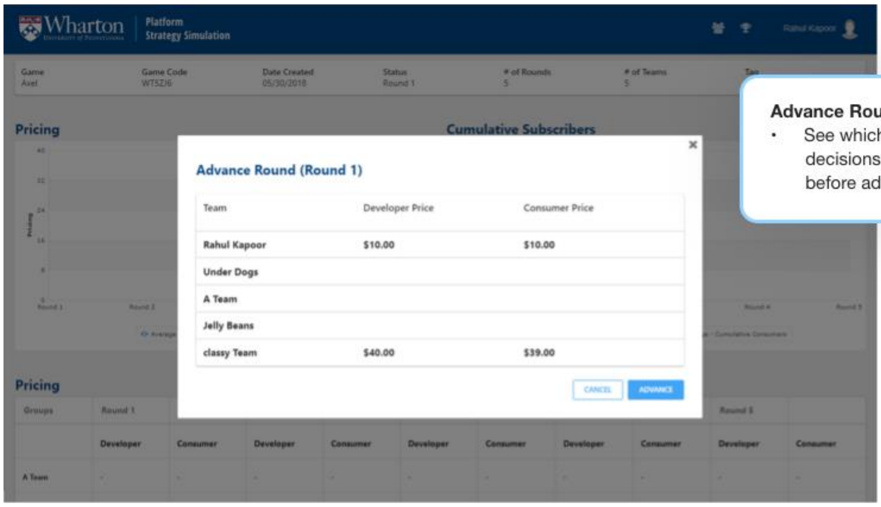
Cumulative Subscribers

Pricing

Groups	Round 1		Round 2		Round 3		Round 4		Round 5	
	Developer	Consumer	Developer	Consumer	Developer	Consumer	Developer	Consumer	Developer	Consumer
A Team	-	-	-	-	-	-	-	-	-	-

Game Details

- View tables and graphs detailing all game results
- Bar graphs for average pricing and cumulative subscribers
- Tables for pricing, new subscribers, retained subscribers, cumulative subscribers, revenue, and profits
- Tables and graphs update as the game progresses



Game	Game Code	Date Created	Status	# of Rounds	# of Teams	Tag
Axel	WT5ZJ6	05/30/2018	Round 1	5	5	

Pricing

Cumulative Subscribers

Advance Round (Round 1)

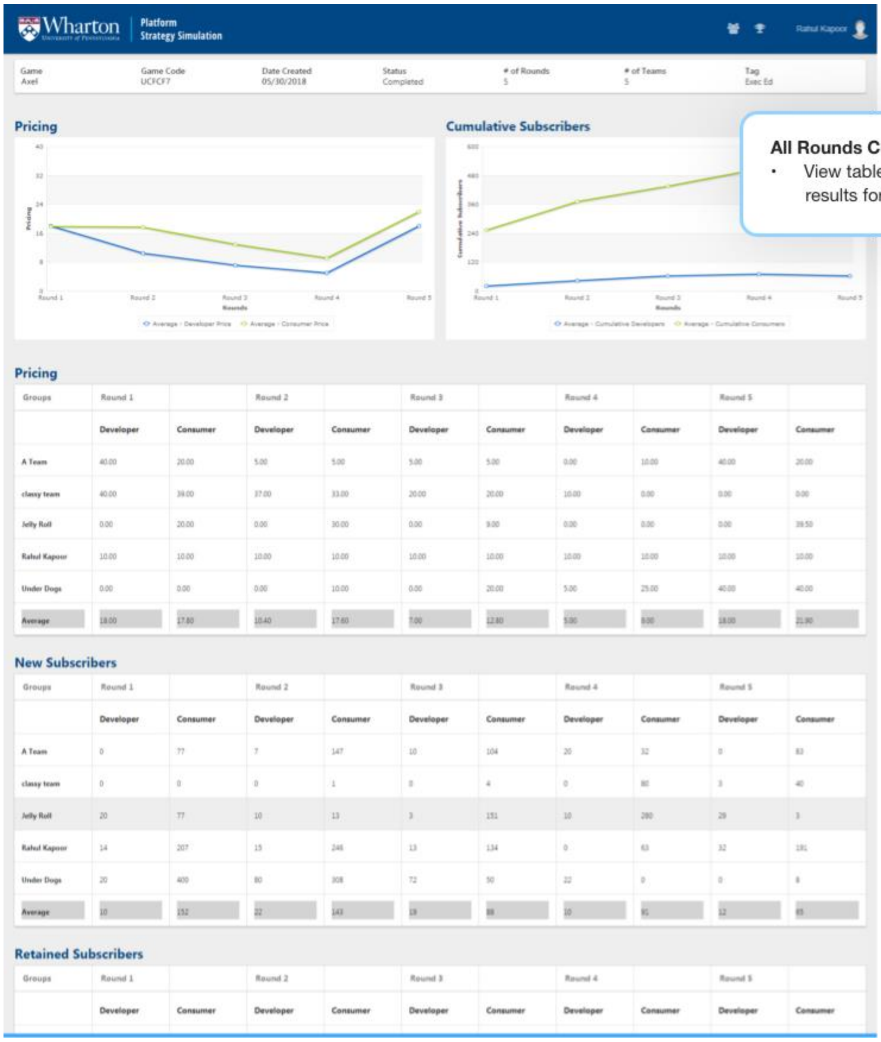
Team	Developer Price	Consumer Price
Rahul Kapoor	$10.00	$10.00
Under Dogs		
A Team		
Jelly Beans		
classy Team	$40.00	$39.00

Pricing

Groups	Round 1								Round 5	
	Developer	Consumer	Developer	Consumer	Developer	Consumer	Developer	Consumer	Developer	Consumer
A Team	-	-	-	-	-	-	-	-	-	-

Advance Round

- See which teams have submitted their decisions and which ones haven't before advancing the round



Game	Game Code	Date Created	Status	# of Rounds	# of Teams	Tag
Axel	UCFCF7	05/30/2018	Completed	5	5	Exec Ed

Pricing

Groups	Round 1		Round 2		Round 3		Round 4		Round 5	
	Developer	**Consumer**	**Developer**	**Consumer**	**Developer**	**Consumer**	**Developer**	**Consumer**	**Developer**	**Consumer**
A Team	40.00	20.00	5.00	5.00	5.00	5.00	0.00	10.00	40.00	20.00
classy team	40.00	39.00	37.00	33.00	20.00	20.00	10.00	0.00	0.00	0.00
Jelly Roll	0.00	20.00	0.00	30.00	0.00	9.00	0.00	0.00	0.00	39.50
Rahul Kapoor	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00
Under Dogs	0.00	0.00	0.00	10.00	0.00	20.00	5.00	25.00	40.00	40.00
Average	18.00	17.80	10.40	17.60	7.00	12.80	5.00	8.00	18.00	21.90

New Subscribers

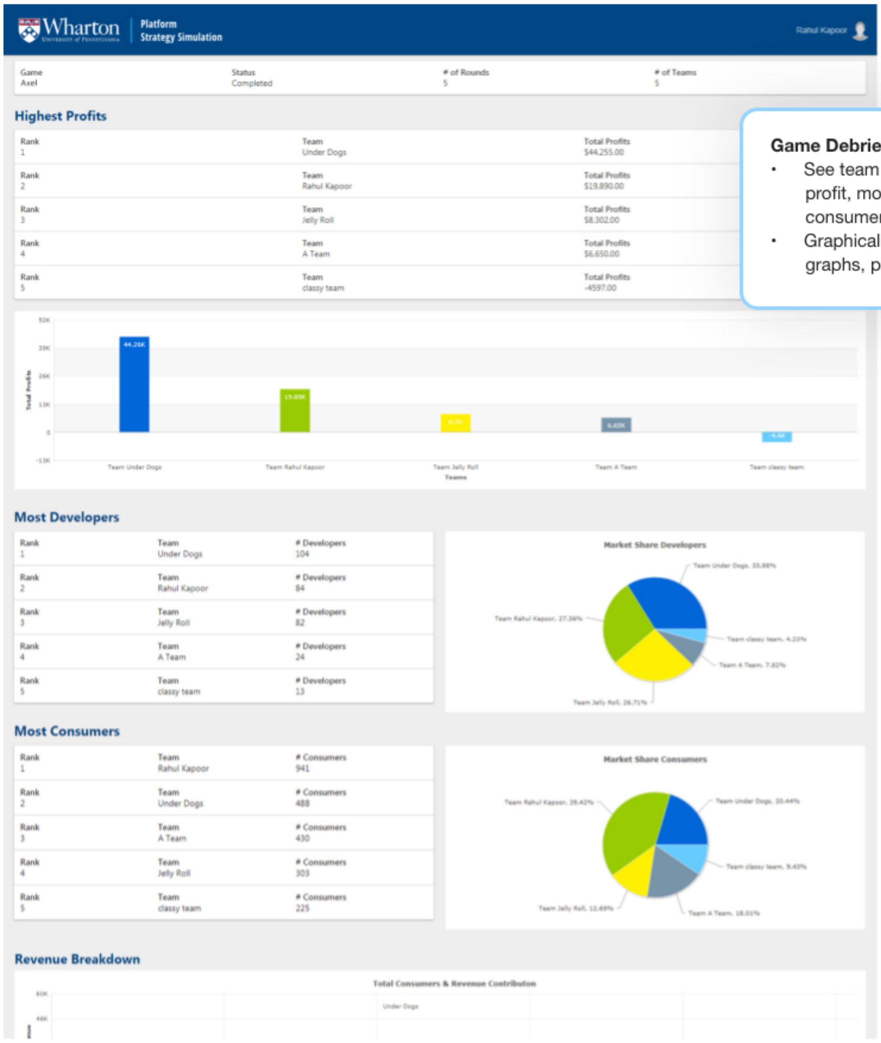
Groups	Round 1		Round 2		Round 3		Round 4		Round 5	
	Developer	**Consumer**	**Developer**	**Consumer**	**Developer**	**Consumer**	**Developer**	**Consumer**	**Developer**	**Consumer**
A Team	0	77	7	147	10	104	20	32	0	83
classy team	0	0	0	1	0	4	0	80	3	40
Jelly Roll	20	77	10	13	3	151	10	280	29	3
Rahul Kapoor	14	207	15	246	13	134	0	63	32	191
Under Dogs	20	400	80	308	72	50	22	0	0	8
Average	10	152	22	143	19	88	10	91	12	65

Retained Subscribers

Groups	Round 1		Round 2		Round 3		Round 4		Round 5	
	Developer	**Consumer**	**Developer**	**Consumer**	**Developer**	**Consumer**	**Developer**	**Consumer**	**Developer**	**Consumer**

All Rounds Completed

- View tables and graphs detailing game results for all the rounds



Game	Status	# of Rounds	# of Teams
Axel	Completed	5	5

Highest Profits

Rank	Team	Total Profits
1	Under Dogs	$44.255.00
2	Rahul Kapoor	$19.890.00
3	Jelly Roll	$8.302.00
4	A Team	$6.650.00
5	classy team	-4597.00

Most Developers

Rank	Team	# Developers
1	Under Dogs	104
2	Rahul Kapoor	84
3	Jelly Roll	82
4	A Team	24
5	classy team	13

Most Consumers

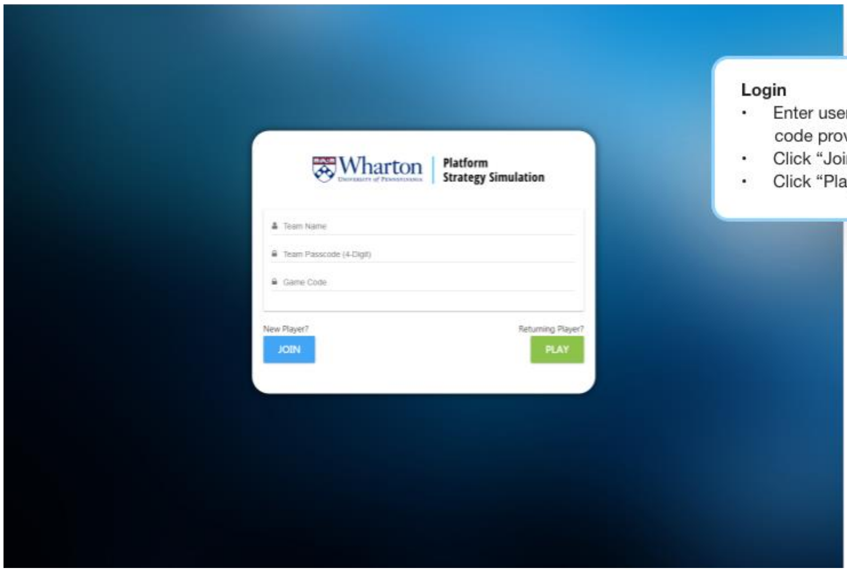
Rank	Team	# Consumers
1	Rahul Kapoor	941
2	Under Dogs	488
3	A Team	430
4	Jelly Roll	303
5	classy team	225

Revenue Breakdown

Game Debrief

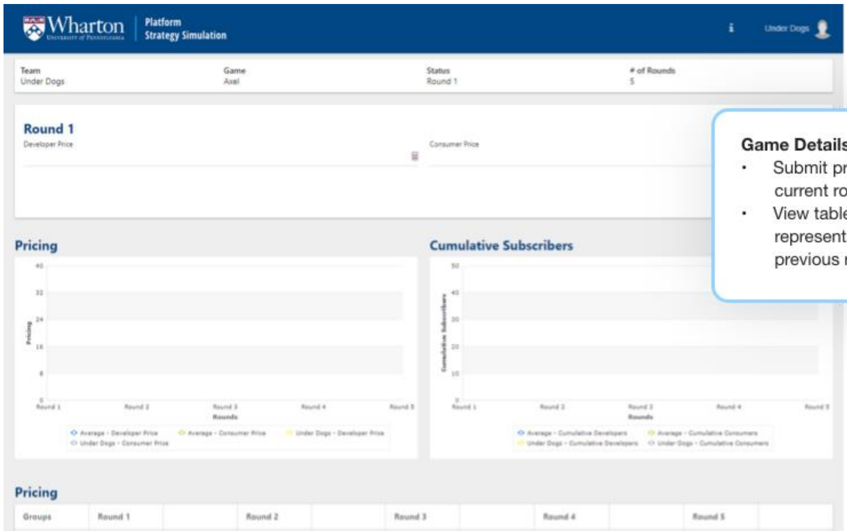
- See team rankings based on highest profit, most developer, and most consumers
- Graphical comparisons using bar graphs, pie charts, and bubble graphs

Exhibit 4: User Interfaces for Participants



Login

- Enter username, password, and game code provided by Administrator
- Click "Join" to finalize team
- Click "Play" to enter the simulation



Team	Game	Status	# of Rounds
Under Dogs	Axel	Round 1	5

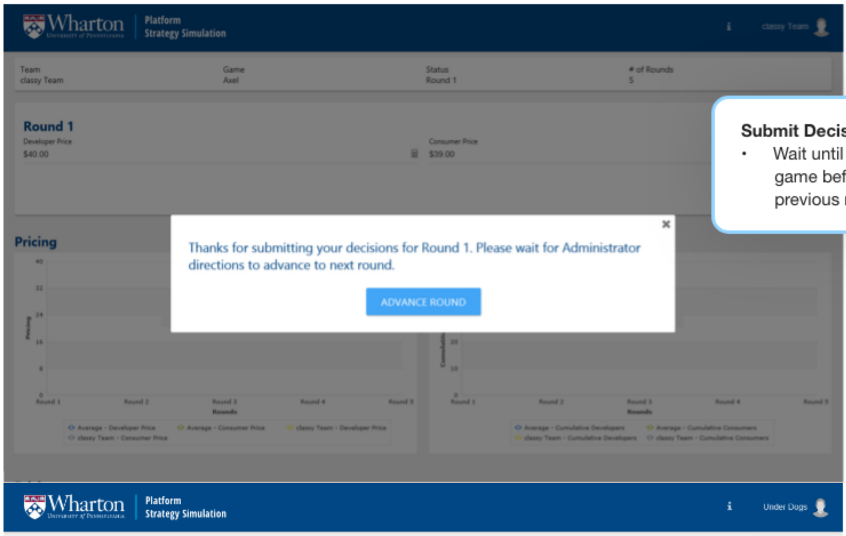
Round 1

Pricing

Groups	Round 1		Round 2		Round 3		Round 4		Round 5	

Game Details

- Submit pricing decisions for the current round
- View tables and graphical representations of results from all previous rounds for all teams


Wharton
University of Pennsylvania
Platform
Strategy Simulation
classy Team

Team
classy Team

Game
Axel

Status
Round 1

# of Rounds
5

Round 1

Developer Price
$40.00

Consumer Price
$39.00

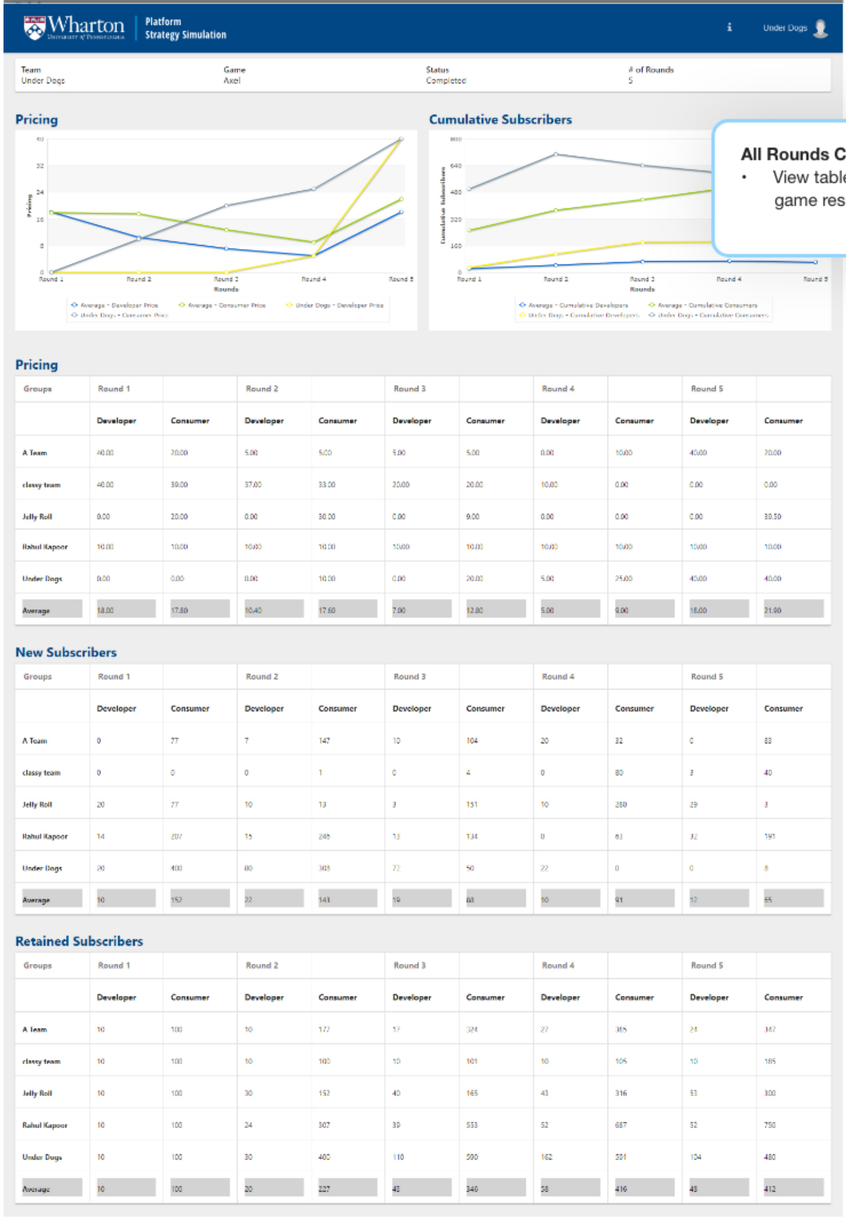
Pricing

40
32
24
16
8
0
Pricing
Round 1
Round 2
Round 3
Round 4
Round 5
Rounds
Average - Developer Price
Average - Consumer Price
classy Team - Developer Price
classy Team - Consumer Price
20
10
0
Cumulative
Average - Cumulative Developers
Average - Cumulative Consumers
classy Team - Cumulative Developers
classy Team - Cumulative Consumers

Thanks for submitting your decisions for Round 1. Please wait for Administrator directions to advance to next round.

Submit Decisions

- Wait until Administrator advances the game before teams can see the previous round results


Wharton
University of Pennsylvania
Platform
Strategy Simulation
Under Dogs

Team
Under Dogs

Game
Axel

Status
Completed

# of Rounds
5

Pricing

40
32
24
16
8
0
Pricing
Round 1
Round 2
Round 3
Round 4
Round 5
Rounds
Average - Developer Price
Average - Consumer Price
Under Dogs - Developer Price
Under Dogs - Consumer Price

Cumulative Subscribers

800
640
480
320
160
0
Cumulative Subscribers
Round 1
Round 2
Round 3
Round 4
Round 5
Rounds
Average - Cumulative Developers
Average - Cumulative Consumers
Under Dogs - Cumulative Developers
Under Dogs - Cumulative Consumers

All Rounds Completed

- View tables and graphs detailing all game results of all the rounds

Pricing

Groups	Round 1		Round 2		Round 3		Round 4		Round 5	
	Developer	**Consumer**	**Developer**	**Consumer**	**Developer**	**Consumer**	**Developer**	**Consumer**	**Developer**	**Consumer**
A Team	40.00	20.00	5.00	5.00	5.00	5.00	0.00	10.00	40.00	20.00
classy team	40.00	39.00	37.00	33.00	20.00	20.00	10.00	0.00	0.00	0.00
Jelly Roll	0.00	20.00	0.00	30.00	0.00	9.00	0.00	0.00	0.00	30.50
Rahul Kapoor	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00
Under Dogs	0.00	0.00	0.00	10.00	0.00	20.00	5.00	25.00	40.00	40.00
Average	18.00	17.80	10.40	17.60	7.00	12.80	5.00	9.00	18.00	21.90

New Subscribers

Groups	Round 1		Round 2		Round 3		Round 4		Round 5	
	Developer	**Consumer**	**Developer**	**Consumer**	**Developer**	**Consumer**	**Developer**	**Consumer**	**Developer**	**Consumer**
A Team	0	77	7	147	10	104	20	32	0	83
classy team	0	0	0	1	0	4	0	80	3	40
Jelly Roll	20	77	10	13	3	151	10	280	29	3
Rahul Kapoor	14	207	15	246	13	134	0	63	32	191
Under Dogs	20	400	80	308	72	50	22	0	0	8
Average	10	152	22	143	19	88	10	91	12	65

Retained Subscribers

Groups	Round 1		Round 2		Round 3		Round 4		Round 5	
	Developer	**Consumer**	**Developer**	**Consumer**	**Developer**	**Consumer**	**Developer**	**Consumer**	**Developer**	**Consumer**
A Team	10	100	10	177	17	324	27	385	24	347
classy team	10	100	10	100	10	101	10	105	10	185
Jelly Roll	10	100	30	152	40	165	43	316	53	300
Rahul Kapoor	10	100	24	307	39	553	52	687	52	750
Under Dogs	10	100	30	400	110	590	162	591	104	480
Average	10	100	20	227	43	346	58	416	48	412