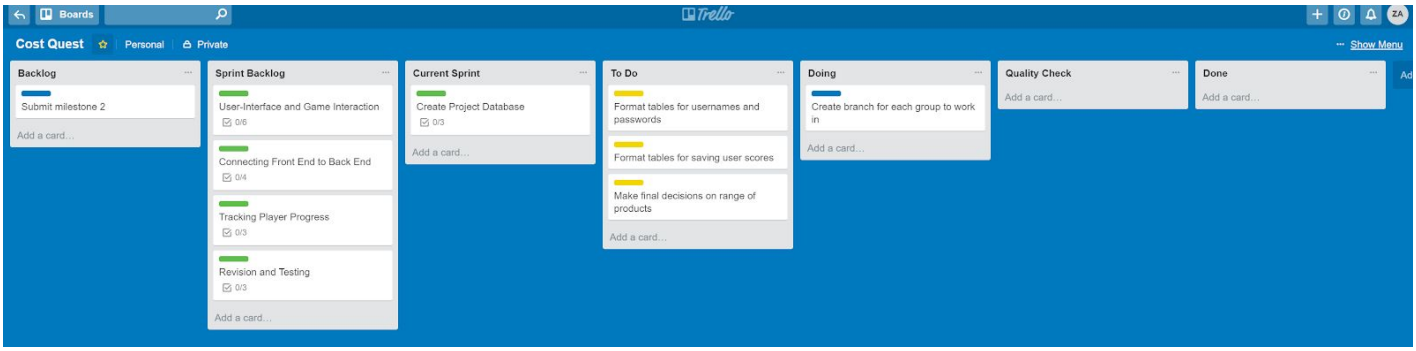
CSCI 3308 Milestone 2
CostQuest

Rebekah Haysley
Binpeng Wu
Zach Asmussen
Haotian Zheng
Theodore Margoles
Jorge Pulido

Project Management Tool Dashboard

Cost Quest Trello Screenshot



This screenshot is an overview of our Project Management Tool in Trello. We have broken up the scheme into sections and differently tagged cards. It is broken up into 7 different sections; Backlog, Sprint Backlog, Current Sprint, To Do, Doing, Quality Check, and Done.

Backlog are a group of miscellaneous tasks that need to be put into to do for various reasons. Blue tagged cards are the miscellaneous tasks to be completed like, in the screenshot, creating branches in git for each group to work with. Green tagged cards are the sprints and yellow tagged cards are subtasks of the sprints. Sprint backlog contains the list of cards that show all of our future sprints spanning to testing the final product. Current sprint is the sprint(s) we are currently working on. To Do lists tasks that are being hard pressed to be started from sprint tasks to miscellaneous tasks. Doing are the card tasks which are currently being done. Quality check are cards which have been finished, but need to be checked over by group members for quality before being put into the done section that contains all of our completely finished and reviewed tasks.

CostQuest Requirements

Functional Requirements: "What" - (Users' view)

User Story ID	As a *<type of user>*	I want to *<perform some task>*	So that I can *<achieve some goal>*
1	Player	View my score and progress	Feel accomplished and validate my reasons for playing the game.
2	Player	Be prompted for a username/email and password	Ensure that my player profile is secure and unique.
3	Player	Create a user account to play the game if I do not already have one.	Log my scores and track my progress.
4	Player	Be challenged with many different unique products	Improve my product recognition and price estimation skills.
5	Developer	Be aware of due dates and schedule	Maintain a good project timeline and be able to meet the project deadline.
6	Developer	Learn HTML, MySQL, PHP, and JavaScript	Increase my ability as a software developer and become a more desirable hire to companies.
7	Developer	Practice teamwork skills and engineering project methodology	Build my character and accountability.
8	Professor	Receive Milestone reports	Make sure that the students are staying on task and developing a worthwhile project.
9	Professor	View a final presentation	Learn about the project that the students worked on all semester.

Non-Functional Requirements: “How” - (Developers’ view)

User Interface Requirements

- Prompt user for a email and password.
- Display the product and description, then prompt user to enter a guess for the price.
- Display user score.

Database Requirements

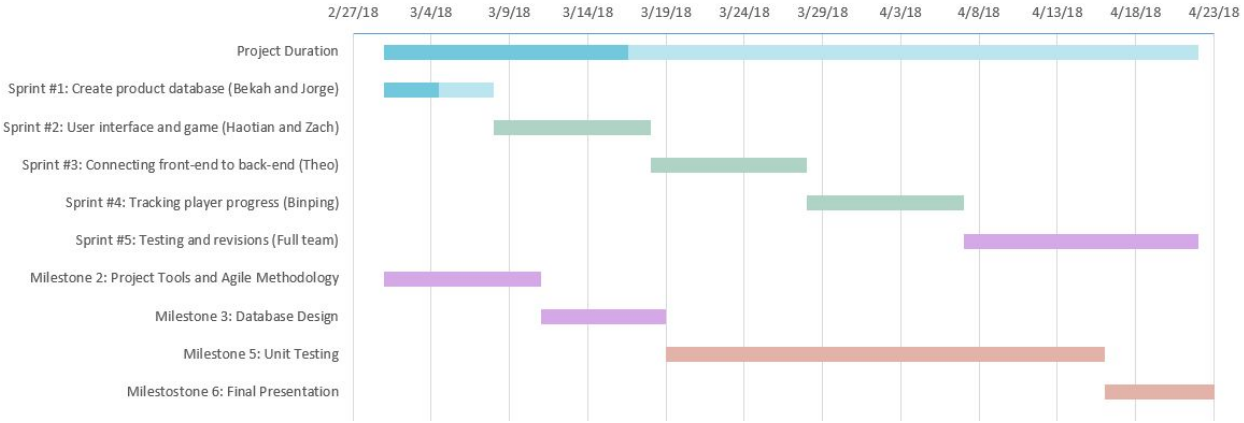
- Store the image URL of an actual product in the product database.
- Store the product ID, product name, product description, and product price in the product database.
- Store the username/email, password, and player score in the player database.

Class Requirements

- Work in a team of 6 people to develop a working software application with a front-end user interface, a back-end database, and layers that connect them.
- Give a group presentation that reinforces the creativity and usefulness of the project.
- Document the design, development, and testing in a GIT repository.
- Turn in Milestones 1-8 on time.
- Use software development methodology and tools.

Timeline / Gantt

Task Name	Start Date	End Date	Duration (Days)	Days Complete	Days Remaining	Percent Complete
Project Duration	3/1/2018	4/22/2018	52	15.60	36.40	30%
Sprint #1: Create product database (Bekah and Jorge)	3/1/2018	3/8/2018	7	3.50	3.50	50%
Sprint #2: User interface and game (Haotian and Zach)	3/8/2018	3/18/2018	10	0.00	10.00	0%
Sprint #3: Connecting front-end to back-end (Theo)	3/18/2018	3/28/2018	10	0.00	10.00	0%
Sprint #4: Tracking player progress (Binping)	3/28/2018	4/7/2018	10	0.00	10.00	0%
Sprint #5: Testing and revisions (Full team)	4/7/2018	4/22/2018	15	0.00	15.00	0%
Milestone 2: Project Tools and Agile Methodology	3/1/2018	3/11/2018	10	0.00	10.00	0%
Milestone 3: Database Design	3/11/2018	3/19/2018	8	0.00	8.00	0%
Milestone 5: Unit Testing	3/19/2018	4/16/2018	28	0.00	28.00	0%
Milestostone 6: Final Presentation	4/16/2018	4/30/2018	14	0.00	14.00	0%



Sprint Descriptions

Sprint 1 Tasks: **Create Product Database (Jorge and Bekah)**

- Format tables for usernames and passwords as separate table from items in database (common security practice).
- Format tables for saving user scores in a table that can be joined to user table with a key, or at very last matched with a common user key for ease of use (OR could be part of 1 main user table, communicate with Binpeng on details before his sprint, #4).
- Make final decisions on range of products we want in the database, as well as final decisions on breaking products up into different tables by subgroup OR keeping all products in same table.

Sprint 2 Tasks: **User-Interface and game (Haotian and Zach)**

- Build user interface that asks user if they want to 1) create account, or 2) login to existing account. (do not worry about connecting to back end, just make html form in proper manner).
- Implement proposed game architecture in HTML and javascript, under the assumption that the queries have been successfully gotten from backend, build the game in a way that can be tested using hand-entered queries.
    - Build module for showing item picture.
    - Build module for entering guessed price.
    - Build module for storing entries to be guessed on in each game.
    - Build module for switching to next picture (incrementing index in array or linked list of database entries)
    - Keep track of all guesses and then display results at the end of game.

Sprint 3 Tasks: **Connecting front end to back end (Theo)**

- Implement finalized database on MySQL server that can serve to remote clients.
- Implement and debug PHP layer that matches our database specs (username/pass and data tables respectively).
- Implement javascript/PHP code embedded in HTML that successfully retrieves our database entries from the backend with the MySql server up and running on a linux machine NOT ON LOCAL HOST OF TEST PC.
    - loop of javascript / PHP that successfully puts *n* random queries into a javascript struct array from remote server. (or other data format specifically requested by Haotian and Zach).

- Debug any concurrent request issues / connection issues / database manipulation problems *as they arise*.

Sprint 4 Tasks: **Tracking Player Progress (Binpeng)**

- Work with Theo, Bekah, Jorge to integrate user progress storage into the database and the PHP code:
    - At beginning of game, when user logs in, if user creates a new account a new spot in database should be associated for tracking that users score associated with a user key, and a javascript variable should be initialized to a score of 0 and saved to be modified by user score at the end of the game. Else → retrieve past user information from the database and store in javascript variable to modify after the game is over.
    - Make executive decisions on storage of user score as a graph, set of graphs, or numerical value or set of numerical values and work with others who have build database to integrate your ideas.

        - If you choose graphs, choose realistic architecture for building graphs that fits into scope.
    - Implement adding scores to database at the end of the game successfully.

Sprint 5 Tasks: **Revision and Testing (Full Team)**

- Analyze core features of the game mechanics and database queries and management for 1)logic errors, 2)syntax errors 3)performance drains
    - Jorge and Bekah → database
    - Theo and Jorge→ PHP / server side
    - Binpeng → user progress
    - Zach and Haotian → user interface
- Review whether we have satisfied all features and/or changed course from original features consistent with agile methodology.
- After review, decide democratically as a group if we want to:
    - A) add new features (as group)
    - B) delete non-functional or obsolete features (as group)
    - C) Assign specific people to revisions on work they are responsible for.
    - D) abort revisions due to lack of time and risk of ending with a non-functional project. (if currently functional at least)

Agile Meeting Description

Date: 3/8/2018
Time: 5:15 pm~6pm
Attendees: Jorge, Binpeng, Haotian, Theo, Bekah

3 Questions:

1. When we working for our project before sprint 3, do we do our project locally or online?
Answer: We can work off of local host to test our functionality until the Sprint where we connect the front and back end in a complete way.

2. Should we divide into pairs or continue to try and program as a large group.
Answer: Theo and Jorge should pair up before Sprint 3 to make sure everyone is on the same page regarding architecture for front end back end connection (PHP, server-client communication).. Binpeng and Haotian should pair up to work on HTML and user interface. Bekah should communicate with Jorge about the database schedule. Zach should work with Binping and Haotian to coordinate the JavaScript game and user interface.

3. At this point in our project do we wish to extend or narrow our scope with respect to any features?
Answer: As long as our sprints are completed within the given time frame, we should be on track to accomplish our project with the original scope, we will have to reassess this decision at other points in the project, including between sprints 2,3,4.

Agile/Sprint Meeting Summary:

During our meeting, we first went around in a circle and said what each person has been working on. Jorge has been working on the php-mysql connection on a local server using LAMPP. Bekah has been putting the database together in excel. Theo has been working on a python tool to potentially automate the process of adding products to the database. Haotian has been working on the html interface. Binpeng has been working on understanding mysql and creating databases. Our main focus of this sprint is getting the database form and structure together. Moving forward, we will continue to work on the product database, the username and password database, and the player score database.

Retrospective Reaction:

Overall, the agile meeting went well because the meeting gave everyone information about what other group mates had done and what they plan to complete in this upcoming week. Furthermore, we have a better understanding about the following important tasks such as security for the password database and the database structure for the high score database and product database. In the future, we need to work on doing better at creating the special features on our website. Our teammates will try to do better at coordinating schedules and making sure the CostQuest project comes together as a group.