DEFINITION OF DONE DOCUMENT: APARTMENT RENT BIDDING SYSTEM

PROJECT INFORMATION

Project Name: Apartment Rent Bidding System
Submission Date: November 1, 2022
DoD Version: 1.0
Client: Damen Tomassi

PROJECT TEAM

Team Member	**Role**	**Document Responsibilities**
Gregory Zacharko	Product Owner	Common Terms and Definitions Within Project Scope and Definition Section; Definition of Done: Checklist; Acceptance Criteria; Proofread DoDD
Lucas Adams	Scrum Master	Burn-Up Chart
Alizsa Johnson	Developer	Project Scope & Definition; Definition of Success
Emma Dougherty	Developer	Production Environment; User Documentation

PROJECT SCOPE AND DEFINITION

The Apartment Rent Bidding System is a Web application that allows apartment administrators to auction off apartment units to potential renters.

The administrators create apartment buildings, populate those buildings with apartment units, and set the minimum going price and bid rate. The apartment administrator gives the bidders the apartment unit ID. Then the bidders start placing bids and continue bidding until the auction ends. At the end of the auction, a winner is selected and both the winning bidder and the apartment administrator are notified.

The scope of the Apartment Rent Bidding System is to create a Web application where apartment administrators can hold auctions for apartment units and invite bidders to bid on the rent.

For Administrators

- Allow administrators to sign up for a special user account that allows them to create apartment buildings and display available units.
  - Once an administrator creates an apartment unit, it is assigned a unique identifier that is visible to the administrator. This identifier allows approved bidders to access the auction.
- Allow administrators to set the minimum bid and minimum bidding increment.
- Allow administrators to add apartment details such as lease length and amenities.

For Bidders

- Allow bidders to sign up for an account that lets them participate in auctions.
- Allow bidders to bid on apartment units.
- Allow bidders to use an auto bidding feature, where they set the max amount they'd like to spend and the bidding system will bid on the users behalf until no other competing bids are placed or until the bidders max price is reached.
- Allow bidders to be notified when they've been outbid and when they've won the auction.

Deliverables

This project will yield a fully functional Web application that allows apartment administrators to auction off apartment units and allows bidders to access and participate in the auctions.

Milestones

- Sprint 0:
  - Decide what kind of pages need to be present on the web application and create the GUI wireframes in Figma.
  - Research the features of Bubble.io.
- Sprint 1:
  - Recreate the Figma diagrams in Bubble.io and begin adding functionality to the web pages.

    - Set up the database for the apartment listing details so that it stores admin input and displays it on the web page.
    - Verify that administrators can create listings.
- Sprint 2:
    - Set up additional databases that will hold the account information for the bidders and administrators and verify that both user types are able to sign up and log in.
    - Create the bidding feature that will allow bidders to participate in the auction.
- Sprint 3:
    - Create the auto-bidding feature and decide how both bidders and the administrator will be notified of significant events related to the auction.
    - Look into using a plugin to send out email updates to the auction participants.
- Sprint 4:
    - Begin bug testing the web application. All of the key features will be implemented so the next step is to verify that the application is working as intended.
    - Begin implementing stretch goals.
- Sprint 5:
    - Test any added features and confirm that they are working properly.

COMMON TERMS AND DEFINITIONS:

1. SPONSOR: The owner/stakeholder of the product.
2. PROFESSOR: Third-party overseer of the project and the team developing it.
3. ADMINISTRATOR: A person that creates a listing for an apartment unit using the product.
4. BIDDER: A person that uses the product to bid on an open and available apartment unit.
5. APARTMENT UNIT ID: A randomized, unique sequence of letters and numbers that bidders use to access the specific apartment listing they want to bid on.
6. BUBBLE.IO: A software engineering platform that allows developers to quickly build high-quality, fully functioning, powerful desktop/Web and mobile applications without having to type out thousands of lines of code.
7. ELEMENT: A component on the website/Web application that the user can interact with (e.g. buttons, links, text input boxes) or that displays information to the user (e.g. dynamic text displays).
8. ELEMENT PALETTE: The place where a developer can pick an element to drag-and-drop onto the page. It is where all possible elements are listed.

9. WORKFLOW: The different user interactions that can occur between the administrators and bidders, and the website/Web application. It is the sequence/flow of events and interactions when an administrator and/or bidder uses the product.
10. FIGMA: A software platform that allows developers to create a wireframe for their project/product’s GUI.

DEFINITION OF SUCCESS

Success for our team is defined by a functioning Graphical User Interface (GUI) that allows both an administrator and user to enter information in the Apartment Rent Bidding System. An administrator would be able to upload information about a specific apartment such as price, pictures of the apartment, and other specifications. A user would be able to create an account and log in to a specific area in which the desired apartment would be listed. The user could then place a bid on the apartment and be notified if their bid was accepted.

In order for these functionalities to perform accordingly, certain measures of success would need to be met. This includes completion of all tasks on the Trello Board in the designated period of time. Along with this, the Burn-Up Chart displays a measure of success based on the timeline of the project and the completion of tasks. Most importantly, the sponsor serves as a measure of success in whether they approve of the final product.

To ensure product quality, testing should be performed by the Development Team. This includes unit testing, integration testing, and acceptance testing. Test procedures should be written to ensure each element of the GUI is tested and functions as a whole. Along with this, the entire team should approve of the product’s functionality and capability before finalizing the product with the sponsor. The product should also be tested to meet the product requirements. This includes both the user requirements and the system requirements.

DEFINITION OF DONE: CHECKLIST

Epics:

- Design and Create the Product’s GUIs
    - Creation of preliminary designs in Figma for the Home Page, Account Sign Up Pop-Up Window, Account Login Pop-Up Window, Reset

    - Password Page, Dashboard Page, Add Listing Pop-Up Window, Apartment Listing Page, Enter Bid Pop-Up Window, See Bids Pop-Up Window, and the My Account Page.
    - Transfer, creation, and implementation of the Figma GUI wireframe into Bubble.io so that there are pages to visualize apartment information and so users can easily interact with the system.
- Develop Database and Workflows
    - Researching how the built-in database in Bubble.io works and how to create the Bubble.io workflows to implement user interaction and data processing procedures within the system.
    - Creation of UML Use Case and UML Project Sequence Diagrams to help the Team visualize the product's workflows.
    - Implementation of 2 databases. One with user fillable fields for their account information such as their full name, email address, phone number, and account password, and a separate one with admin fillable fields for the information about an apartment and apartment listing, like the number of bedrooms, number of bathrooms, size of the apartment, and the starting bid price.
    - Connecting the page elements to the workflows and database so that the product is usable and navigable by users.
    - Allowing users to create their own accounts so that they can access the features of the product and so that their personal information is saved by the system. Store personal information in the user information database.
    - Allowing administrators to create an apartment listing by entering into fields the essential information of the apartment. Store apartment listing and its accompanying information in the listing database.
- Build Bidding System and Auto-Bid Functionality
    - Implementation of the main, core system functionality: the bidding system and workflow. Enable users to enter a bid into the system via the Enter Bid pop-up window. Store bids in a field for that listing in the listing database.
    - Create the ability for users to use the Auto-Bid system to bid for them if they so wish. The user sets their Auto-Bid increment to determine how much they want to bid each time, and sets their Auto-Bid maximum to determine the price limit that they want the Auto-Bid system to stop bidding at. The Auto-Bid system bids cumulatively on top of their Auto-Bid increment.
    - Implementation of an email notification system to notify the top/winning bidder and the apartment listing administrator that the auction is over and for the two parties to meet up and finish the deal.

Sprint Increments:

- Sprint 0:
  - First set of Trello board cards written and placed in Product Backlog.
  - Research into Figma and Bubble.io features and development process completed.
- Sprint 1:
  - In Figma, sample GUI wireframe designed and completed.
  - Preliminary Use Case Diagram designed.
  - Created a project in Bubble.io to start developing the product itself.
  - Granularized Trello board cards into cards with more specific requirements on them, and added User Stories, Acceptance Criteria, and Tasks (a “To Do” Checklist) to each card.
  - Conducted more research into Bubble.io’s capabilities and database features.
- Sprint 2:
  - GUI wireframe transferred from Figma to Bubble.io, and design of GUI web pages and pop-up windows tweaked and finalized.
  - Implemented basic, essential elements, components, features, and functionality to the product’s core web pages and pop-up windows.
  - Began development of the product’s databases and fields in Bubble.io.
  - Continued granularization of Trello board cards.
- Sprint 3:
  - Home Page fully completed (both GUI and functionality).
  - Account Sign Up and Account Login pop-up windows and their functionality (Sign Up/Login system) developed completely.
  - Database mostly configured and complete. Connections between the pages/pop-up windows and the database almost fully developed.
  - Create Listing pop-up window and functionality completed.
  - Dashboard Page GUI and functionality largely, but not fully, completed.
  - Apartment Listing Page GUI and functionality mostly completed.
  - Account Information/Profile Page partially developed.
  - Reset Password Page partially completed.
  - Developed and implemented most connections/links between pages.
  - Bid pop-up window and functionality partially completed.
- Sprint 4:
  - Bid pop-up window and functionality fully completed.
  - Database and database connections fully configured and implemented.
  - Dashboard Page feature complete.
  - Apartment Listing Page developed completely.
  - Account Information/Profile Page functional and complete.

    - Reset Password Page functional and complete.
    - All connections/links between, to, and from all pages completed.
    - Finishing touches made to the project.
    - Bug fixing the product and more testing of the product conducted.
- Sprint 5:
    - Final bug fixes, refinements, GUI adjustments, and testing conducted across all buttons, text boxes, links, connections, pop-up windows, pages, picture galleries, text displays, databases, and all other aspects, elements, components, and features of the project.
    - Development of the project fully completed.
    - Fully-featured final product produced, presented, and deployed.

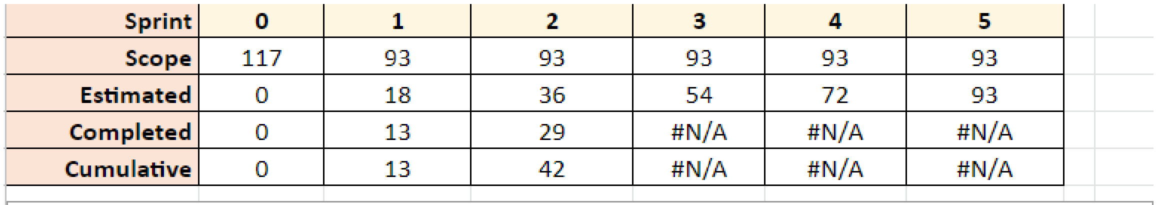
<u>Testing:</u>

- Our Bubble.io project/product will be tested using Bubble.io's built-in "Preview" function and its accompanying debugger. This "Preview" function and debugger allows our Team to see how our product will look and operate when deployed. This way, we can easily test the buttons, text boxes, static text displays, workflows, database changes/calls, pop-up windows, picture displays, and dynamic text displays in a live, real-time testing environment. We will test all of our sprint increments before they are presented and deployed at the end of each sprint.
- Since Bubble.io is a codeless development platform, there will be no peer code reviews in the traditional sense. Instead, our Team will independently and individually test the product as we are developing it, enabling us to assess, catch, and fix problems/bugs as early as possible during this development cycle and sprint. Then, we will discuss our findings, errors, problems, solutions, and anything else we feel the need to discuss together through our Team's Discord server and/or through an in-person or virtual meeting. This in-person/virtual meeting or discussion on Discord will occur multiple times throughout a sprint, as often as needed.
- Our project's design documentation is following a consistent format and style. All written documentation/deliverables will have parity with one another in terms of their structure, form, and writing style. Our style of choice was decided very early on in our product's development lifecycle by all members of our Team.
- Through the convenience and ease of testing through Bubble.io's "Preview" function, our Team can easily perform Functionality Testing and Performance Testing. This is simply done by clicking on all of the buttons, typing into all of our text boxes, clicking on all of our picture galleries, and watching and keeping in mind how long the pages and pop-up windows take to load, how long it takes for fields to update in the database, how long it takes for pictures to upload and

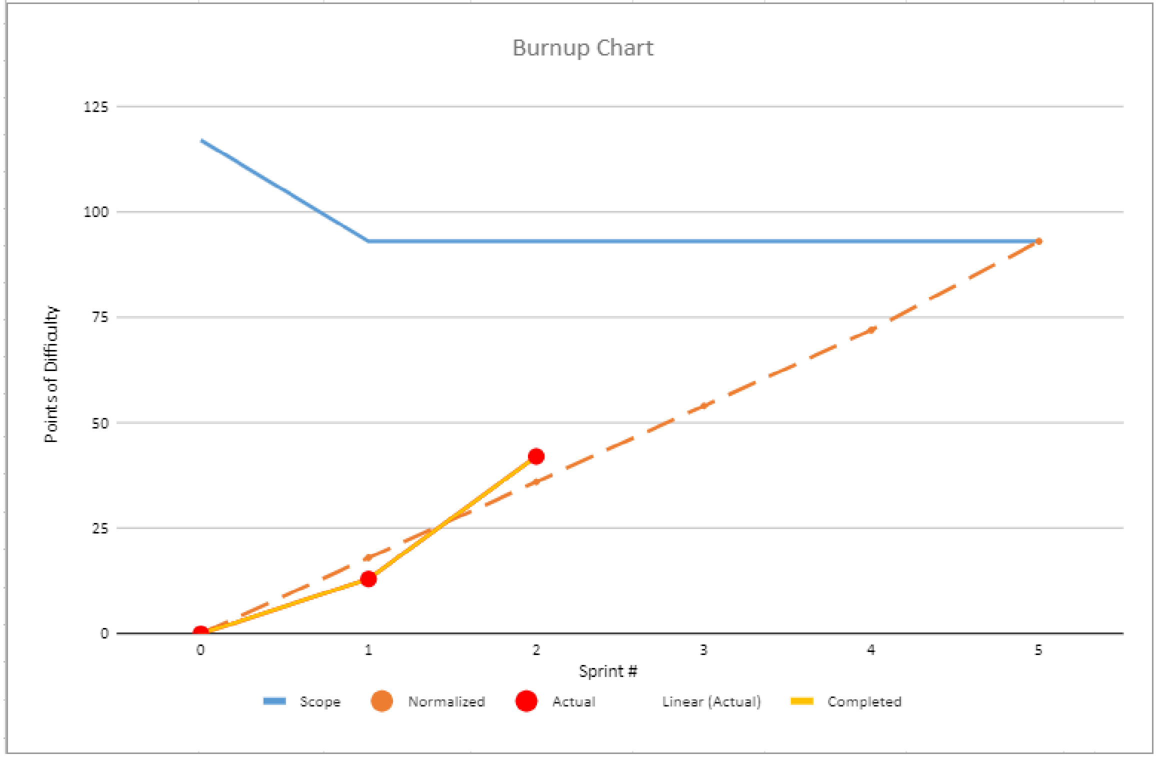
appear, how long it takes for created listings to appear on the dashboard, and seeing if any errors or crashes occur during any of those tests and processes. We can clearly test and visually conclude if all of our elements and workflows are efficiently performing as we expect them to because Bubble.io's capabilities allows us to test in real time, live, within the simulated deployed environment.

- When it comes to Unit Testing our product, our Team will look to the Trello board cards. The Team member will go to the card for the particular part of the product they are testing, and then look at the Acceptance Criteria that is near the top of the card. Then, they will test what they are testing and make sure that it clears each one of the individual acceptance criteria written on the card.
- The Team member that is performing tests is also free to conduct tests that may not be written on the card. This way, testing is dynamic, free flowing, and ever changing, ensuring that all possible errors, problems, and bugs are solved.
- All testing, including Functionality, Performance, and Unit Testing, is performed all throughout the development of the product.

BURN-UP CHART



Sprint	0	1	2	3	4	5
Scope	117	93	93	93	93	93
Estimated	0	18	36	54	72	93
Completed	0	13	29	#N/A	#N/A	#N/A
Cumulative	0	13	42	#N/A	#N/A	#N/A



ACCEPTANCE CRITERIA

Agile Methodology Terms:

- Epic:
  - A larger and more complex version of a User Story that describes a big feature and/or multiple functionalities of the to-be-developed product. And Epic can be made up of one large, long User Story, or several, smaller User Stories bundled together into one package/Trello board card.
- User Story:
  - A small descriptive statement that describes a feature of the product to be developed. User Stories are written using informal, non-technical, simple language that can be easily understood. They provide a relatable scenario from the perspective of the end user to help both non-technical stakeholders and technical developers understand what exactly is the feature being developed and why.
- Acceptance Criteria:
  - A list of requirements on each Trello board card that shows what necessary standards and functionality the feature on that card must meet in order to be considered “Done” or “Complete”. Each bullet point/numbered requirement on the list is made of sentences that are designed to be answered in a binary, yes-or-no kind of way. If the answer to a sentence in the acceptance criteria is yes, then it means the feature meets that requirement. If not, then the answer to the sentence is no, meaning it has failed to meet the necessary requirement and must be worked on and fixed until it clears the requirement in testing. Once all of the acceptance criteria are passed, then that feature and card is “Done”.
- Tasks:
  - A specific, small, individual part of a feature and/or product that is able to be measured and implemented by the Development Team. Tasks on a Trello board card usually take the form of instructions on a checklist or “To Do” list, that can then be checked off as they are completed.

PRODUCTION ENVIRONMENT

In order for the Apartment System to be deployed, Bubble.io requires a subscription to enable the product to function in a live environment. Knowing this, our Apartment System will be previewed to display functionality to the Sponsor. If the product were to be deployed in a live environment, it would be hosted on Bubble.io, allowing a user to take advantage of the Apartment System’s capabilities. The Team will be using the “Preview” functionality to test the Apartment System’s functionalities and features.

USER DOCUMENTATION

Within the user documentation, there will be a brief overview of the product, its purpose, and how the user may use the system. In the overview of the product, it will describe the basic functionality and features of the system that are active once a user has made an account. This includes a Reset Password Page for account access resetting, account authentication, bidding functionalities for bidders, and the ability to create apartment listings for administrators. Along with this, there are separate Dashboards for each user type where they will be able to perform these actions. The purpose of the Apartment Bidding System is to allow users to place bids for specific apartments to rent as well as create listings for apartments to be bid on. This allows for efficient rent establishment for apartments in high-demand areas. Finally, the user may use the system by first creating an account and signing into the system with their email and password. Depending on their account type, they will be brought to their respective Dashboard Page. Within this Dashboard, a user may select a listing to go to so they can place a bid or view the apartment if they are a bidder. A user may create listings for apartments and view bids if they are an administrator. The administrator may select the top bid for the apartment.