ARTYFY

Project Team - 5

COMP8117 Fall 2021

**Sprint - 4 Report**

# **Objectives of the sprint**

The objective of the sprint is to integrate the frontend with the backend and to conduct a dry run of the system. This sprint includes detailed working of View list of artists feature, i.e., View or fetch artists and their arts from the database and also, to create UI for the same. We have also focused on creating a more user friendly registration UI. By the end of this sprint, we have completed the below tasks related to the features:

* View list of Artists feature
* Art Finder Algorithm
* Payment Services
* Integration of Art Finder Algorithm
* Fix internal Bugs

# **Tasks**

* **Scrum meetings**

Conducted the scrum meetings in the team to gather the requirements which are provided by the customer and discussed the progress and implementation throughout the sprint.

* **Meeting with customer**

Customer gave related inputs and validation is done for every screen present and for milestone meeting, required presentations, live demo & feedback is taken at end of sprint.

* **Spécifications**
* **View list of artists.**

**Introduction:**

To view the list of artists and their art that are saved in the database and integrating it with UI for further purposes.

**Inputs**

User will click on the artist list and the details of all the artists available will be fetched from the database, accordingly.

**Processing**

Basing on the selection of the artist by user, a list of arts done by the artist choose will be fetched from the database and integrating it so that we have reusability of components of that list. This integration is done with other screens and backend so that the data that is added by user, is processed accordingly.

**Output**

Users can view the list of artists and their respective art by providing some details and that can be further used for future purposes.

* **Payment Service**

**Introduction:**

Once the user decides on buying a particular piece of art, they will be redirected to payment and billing.

**Inputs:**

User selects a piece of art that is available to buy in our application.

**Processing:**

When the user selects a piece of art that is available to buy in our application, and click on buy,this will be added to the cart and they will be directed to various payment systems available in our applications such as paypal. In the backend we will use our algorithm to process their requests.

**Output:**

The user will be able to buy a. particular item.

* **Art Finder**

**Introduction:**

To take the image provided by the user and find similar images in the same that can be processed using algorithm.

**Inputs:**

User uploads a picture from the UI screen and then, this image reaches the algorithm for further processing.

**Processing**:

When user uploads a picture from the screen, it reaches the Art Finder algorithm for further processing in which it undergoes through loading the image and passing the image through firebase. The algorithm checks for the image and then it is saved to the database for future references. The algorithm further provides similar images in that genre for user to view.

**Output:**

Various images similar to the one user uploaded will appear on the screen with information on that genre.

**Miscellaneous Tasks:**

There were miscellaneous tasks that includes integration issues with integration bugs, completing backlogs, planning sprin-5 and testing the current sprint.

**Design** -

We are using location of the user to facilitate nearby artists . So user must accept location access. Picture quality art is dependent on hardware specification of device of artist. Authenticity of details must be validated if the scope of project is extended, current scope is restricted to authenticity of artist.

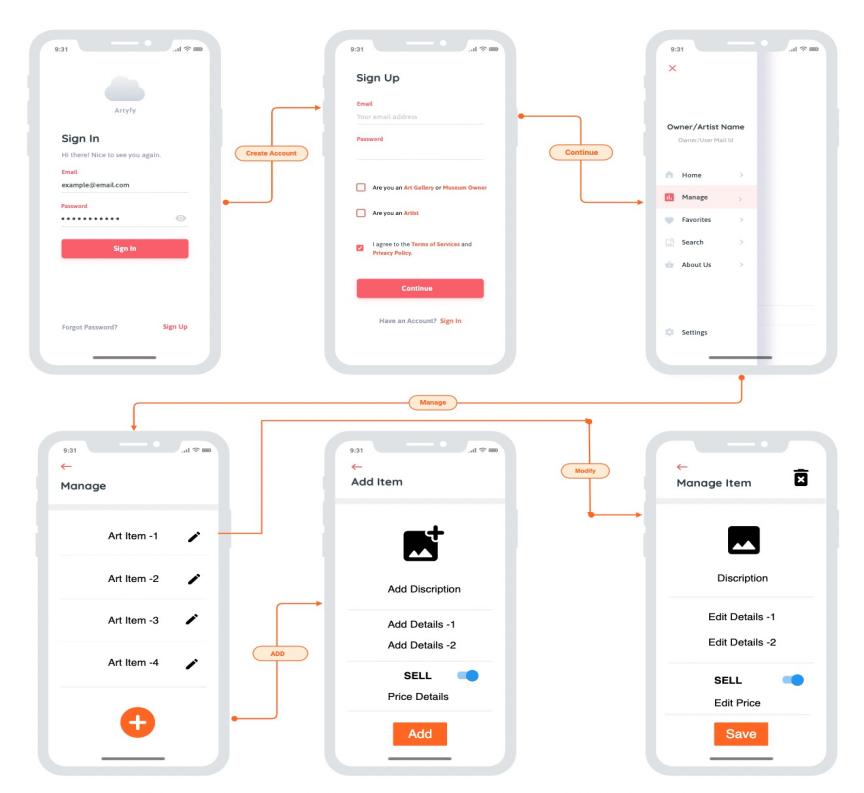
****

Figure : UI Mock UP

**Implementation –**

The artists are able to register themselves in the artist category.This allows them to upload their art into our application. The artist must submit the artist name, their contact information and their location details. The artist can perform activities like add arts, delete arts, sell art or just showcase their art.

For the payment, The user will be able to choose from various payment options available in our application.

**Test**

Testing should be done on validation of payment information, view list of artists feature, visibility of artists and art to end user, operations on art items, synchronizing of changes done and payment failure testing.

1. **Provisional Planning of the Sprint**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task** | **PSD** | **PED** | **Expected Duration** | **Expected Assignee** | **Dependent Tasks** | **Expected artifacts** |
| Art Finder : Algorithm Integration to application | Nov 17 | Nov 19 | 2d 1h | Narayana | 87 |  |
| Art Finder: Improving Accuracy of Algorithm | Nov 17 | Nov 19 | 2d 1h | Shruthi | 88 |  |
| Artists: View List of Artists | Nov 3 | Nov 28 | 4w 1d 2h | Bhargav  Varun  Pravallika  Deepak  Nupur | 89 |  |
| Payment Service | Nov 17 | Nov 18 | 1d 2h | Yuva | 90 |  |
| System Testing: Sprint 4 | Nov 19 | Nov 19 | 7h | Yuva | 95 |  |
| Project Management : Sprint - 4 | Nov 7 | Nov 28 | 3w 1d | Yuva | 143 |  |
| Integration bugs : Sprint 3 | Nov 20 | Nov 20 | 7h | Yuva | 144 |  |

1. **Effective Planning**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Task** | **Related Tasks** | **PSD** | **PED** | **Expected Duration** | **Expected Assignee** | **Dependent Tasks** | **Expected artifacts** |
| Art Finder : Algorithm Integration to application |  | Nov 17 | Nov 19 | 2d 1h | Narayana | 87 |  |
| Art Finder: Improving Accuracy of Algorithm |  | Nov 17 | Nov 19 | 2d 1h | Shruthi | 88 |  |
| Artists: View List of Artists | 1)Artist: registration UI  2)Artist: registration -Backend  3)Artist Schema Validation and creation – Database  4)functional testing  5)Manage Arts of artist: add, delete and update  6) Artist description and list of arts page  7)Integration of Artist Components  8)Artist: View art from artist detailed view | Nov 3 | Nov 28 | 4w 1d 2h | Bhargav  Varun  Pravallika  Deepak  Nupur | 89 |  |
| Payment Service |  | Nov 17 | Nov 18 | 1d 2h | Yuva | 90 |  |
| System Testing: Sprint 4 |  | Nov 19 | Nov 19 | 7h | Yuva | 95 |  |
| Project Management : Sprint - 4 | 1)Sprint Planning  2)Meeting with customer  3)Sprint End :Retrospective  4)Daily Scrum | Nov 7 | Nov 28 | 3w 1d | Yuva | 143 |  |
| Integration bugs : Sprint 3 |  | Nov 20 | Nov 20 | 7h | Yuva | 144 |  |

\*\*w : week

d : day

h : hours

(Our team worked from Wednesday to Sunday in the week, taking leave on Monday and Tuesday)

1. **Review**

* The View list of artist feature is working as expected. We are able to fetch the list artists and of their art and view it in the View artists page as required.
* The integration of artist components and the functional testing of this feature is completed perfectly.
* As discussed in sprint 3, we have worked on integration of art finder algorithm and improving the accuracy of art finder algorithm. This algorithm is running neatly without any errors.
* Payment services have been integrated to the application and the test runs have worked well.
* We will focus on bug fixes after deployment in the next sprint.
* We will also be working on improvising the art finder: art categories or genres for the next sprint.
* Final deployment and release of this application will be done in sprint 5.

1. **Retrospective**

* We organized scrum meetings with all the team members based on their availability and communicated regarding our individual contribution and their respective tasks.
* Based on the customer inputs and suggestions we implemented the features and their functionalities accordingly.
* Tasks allocation and time logging was done in Jira.
* As we focused more on the implementation in this sprint, we will also give more preferences towards the documentation such as sprint report, presentation to the customer and other required documents along with the implementation.
* We will complete the deployment and release of the application in the next sprint.