**System Description Documentation**

**For**

**[Artisan Bridge]**

**Document Version: [1.0]**

**Date: [17/09/2021]**

**Contents**

1 Introduction……………………………………………………………………...3

[*1.1* *Contributors* 3](#_Toc387220237)

[*1.2* *Version Control* 3](#_Toc387220238)

[2 OVERVIEW 4](#_Toc387220239)

[2.1 Service Description 4](#_Toc387220240)

[2.2 Data Model 4](#_Toc387220241)

[2.4 Technology 5](#_Toc387220243)

[2.6 Interface 6](#_Toc387220245)

[3 System Requirements 7](#_Toc387220248)

[3.1 User Class and Charactersistics 7](#_Toc387220249)

[3.2 Functional Requirements 7](#_Toc387220250)

[3.3 Non Functional Requirements 10](#_Toc387220251)

[4 Architecture Design 10](#_Toc387220253)

[4.1 Site Map and Navigation 11](#_Toc387220250)

[5 Appendix](#_Toc387220253) 12

5.1 Some Pictures of the webapp interface 12

5.[2 Link To GitHub Repository 14](#_Toc387220245)

5.[3 Link To Artisan Bridge Website 14](#_Toc387220245)

# Introduction

This document provides an overview of the System description for Artisan bridge web application. It includes the purpose, scope, target audience, design approach, main component design and high level system design considerations of the web application.

## *Contributors*

|  |  |  |
| --- | --- | --- |
| **Role** | **Unit** | **Name** |
| Project Manager | All units | Joseph Ajegetina Ajongba |
| Team Head | Front End | Atompoya Evans Yintuma |
| Member | Front End | Arhin Enoch |
| Member | Front End | Addo Tawiah Bernard |
| Member | Front End | Essel Kofi Fosu |
| Team Head | Back End | Obeng Asare Micheal |
| Member | Back End | Ackom Patrick |
| Member | Back End | Asante Kwadwo Frimpong |
| Team Head | DataBase | Solomon Botchway |
| Member | DataBase | Akyere Danso Rosemary |
| Member | DataBase | Addo Bernard Tawiah |
| Team Head | Software Documentation | Asante Kwadwo Frimpong |
| Member | Software Documentation | Akyere Danso Rosemary |

## 

## *Version Control*

Throughout the various development phases of this web application, a lot of changes were made to arrive at the final source code. Also we had members working on different sections of the web application. There was therefore the need to find a way of keeping everything in sync and to have members working concurrently to kill time since we had a deadline to meet. In order to keep track and manage all the changes done to the source code, the team resolved to using a version control system.

Among the various version control systems available on the market, Git was the one settled on, reason being that it is a distributed system and offers more advantages and flexibility than local and centralized version control systems.

A Github link encompassing all the changes and modification made during the development of this web application will be made available in the appendix section of this document.

# OVERVIEW

This document provides a description of the technical design for Artisan-Bridge Web application. The primary purpose is to describe the technical vision for how business requirements will be realized. This document provides an architectural overview of the system to depict different aspects of the system. This document also functions as a foundational reference point for developers.

Please note that this is a baseline document and may be updated as development progresses.

## Service Description

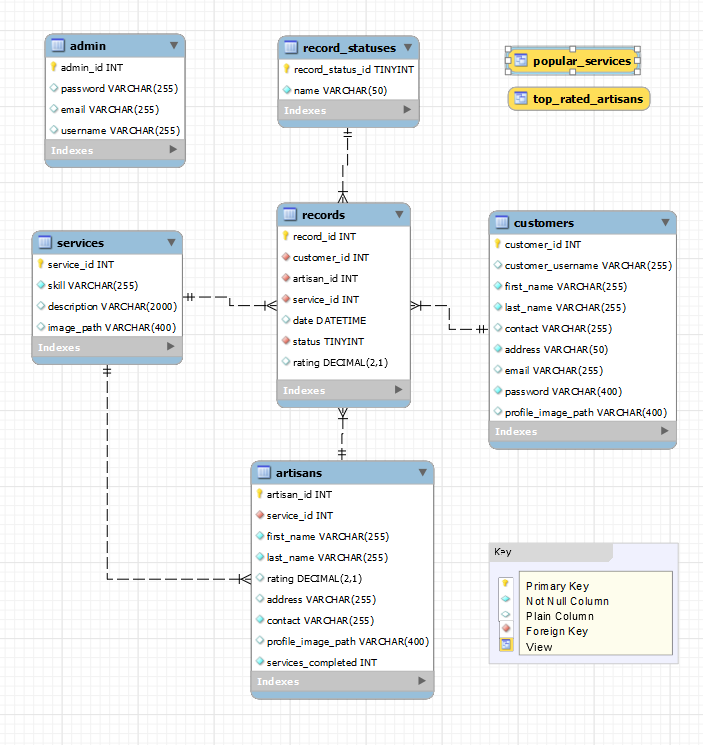
Artisan Bridge app is a bridge between the general public and various artisans in their vicinity.

For the artisan, finding a customer is a game of chance. For instance, on campus, we see various artisans such as cobblers, tailors and many others, while others roam the various hostels hoping to find someone who needs their service at that time. It can happen that, at the time of their visit no one needs their service. But at the same instance there is a student in another hostel hoping that same artisan would come around. If it happens that he/she visits that needy student hostel then he can render the service, else if he does not, he loses a customer.

The goal of the app is to bridge this gap between the artisan and their various customers who needs their service. A typical use case is, a student needs the services of cobbler so goes to the app and request that service. The artisan, cobbler in this case is notified by either a phone call or an SMS of the needy customer. The location of the customer is provided and the artisan near to this customer goes to render the service. This helps the artisan by saving his or her energy and time as he or she does not have to roam the various place hoping to find a customer. This also helps the customer by making the artisan available to them at their time of need. Both parties are satisfied and the app in turn serves it purpose.

## Data Model

This section provides an entity relational diagram of the data dependencies and relationship in the Artisan Bridge Database.



## Development Technology

Throughout the development of this web app, some frameworks and underlying technologies were used to deliver the final product.

Frontend: React, CSS

Backend: Flask framework

Database: MySQL

Version Control: Github

## Interface

Each part of the user interface intends to be as user friendly as possible. The fonts and buttons used are intended to be very fast and easy to load web pages. The pages will be sizeable such that they won’t take a long time before they are loaded. The user interface for the software will be compatible with any browser such as Mozilla Firefox, Google Chrome etc. which the user can use to access the system. Screenshot of some of these interfaces will be provided at the appendix section of this document.

# System Requirements

## *User Class and Characteristics*

There are three (3) kinds of users for the proposed system.

• Administrators: Administrators are responsible for carrying out the administration of the business. Administrators can add artisans to the platform as well as regulate payments and view feedbacks given by users.

• Artisans: The artisans are responsible for attending to the requested service from the clients on the app.

• Clients: The clients request for services from artisans and craftsmen. They also give feedback after the service is done either by rating or giving a comment.

## *Functional Requirements*

CLIENT

REGISTRATION:

Before a client can request for any service on the web-application, he first needs to register. This is required to get a good database of users on the platform and to also track the progress of the application with regards to how it is being used by the public. The user registers with Artisan Bridge with the following information;

• Full name

• Phone number

• Email

• Password

• Gender

LOGIN:

After registering, the user’s information is stored in the system’s database for future reference. To log in to the app, the user uses his or her email, phone number and password provided during registration.

PASSWORD RECOVERY:

When the user forgets his or her password, it can be recovered from the system’s database by the user entering his or her email and phone number at the forget password page. If the email and phone number entered do not match any in the database, he or she is then required to create a new account before he or she can use the app.

SERVICE REQUEST:

After logging in, the user sees a wide display of artisans that have registered with the web-app, their ratings and the services they provide. The user can then select an artisan based on his needs and place a request for him.

SERVICE CANCELLATION: Retracting of requested service by clients is allowed.

MODE OF PAYMENT: Payment can be done electronically or by giving physical cash.

FEEDBACK

The app also allows the user to send suggestions, complaints and feedback through the comment box to the administrators. Responds are sent directly to the users through their emails.

ARTISANS

Registration:

Artisans are signed up on the webapp by the administrators. The artisans provide all the details required for registration after reading all the terms and conditions and agreeing to it. The following details are required:

• Name

• Date of Birth

• Nationality

• License Number

• Account Number

• Username

• Profile picture

• Email

• Telephone number

• Account holder name

• Vehicle model

• Vehicle number

• Password

Service Report:

Requested services are marked as complete when its done or incomplete when it is yet to be completed

Online status: An artisan is indicated as offline or online depending on whether he is available for a service or not.

ADMINISTRATOR:

Administrators accounts will be fused into the database with administrators accounts having different access levels. Administrators will already have their details consisting of their:

• Full Name

• Emails

• Password

• Contact

CLIENT ACCOUNT VIEWED FROM ADMINISTRATOR DASHBOARD

Client accounts can be viewed from dashboard. Details of account that can be viewed consists of:

• Personal information

• Emergency Contact

• Activity logs

• Service Requests

• Service Payments

ARTISANS ACCOUNT VIEWED FROM ADMINISTRATOR DASHBOARD

All artisans account can be viewed from dashboard. Details of account that can be viewed consists of:

• Personal Information

• Rating

• Emergency Contact

• Services Requested

• Earnings

• Activity Logs

All Administrators are given the privilege of either activating or deactivating an artisans account from the dashboard.

Requested Services

Administrators can view all service request that have been made on the application. All completed, requested, ongoing, cancelled and scheduled requests with details of clients information, artisans information, service price, service payments, dates and times can be viewed by the administrators. Every requested service has a unique id.

Payments Records

Administrators can view payment records. Customer payments with details like references, dates etc. are viewed.

## *Non Functional Requirements*

Apart from the functional part of the software, there are non-functional parts that do not exactly perform a particular action but are crucial nevertheless. These consists of attributes such as security, performance, usability etc.

Performance: On a scale of one to ten, the performance of the web application can be rated as an 8.5. This is very pleasing since there is a guarantee that there won’t be frequent failure of the application especially when it is being accessed by a number of people.

Supportability: Since the applications are going to be used on a wide range of devices, it is in a format that supports all these different devices such that users can access features and application without any issues.

Legal: The applications are legal since no laws are broken in the creation and development of the system.

Maintainability: The applications give room for maintenance after the development of the applications. Updates can be done to the applications as and when needed.

Security: The system must automatically log out users after a period of inactivity. The system’s data will only be accessible by authorized administrators. Sensitive data will be encrypted before being sent over insecure connections.

Portability: The application will be using HTML and scripting languages so the application is portable on the end – user’s face as any system using any web browser should be able to access all features of the web application.

**4. Architectural Design**

**4.1 Site Map and Navigation**

No No No Yes Yes

Get Registration information

Valid?

Successful Registration?

User Visits Artisan Bridge Website

User Role

Manage feedbacks

Register Artisans

Submit Feedback

Manage your orders

Get Login Information

Contact Us page

Are you logged in?

Customer Dashboard

Admin Dashboard

Order Service

Signup Page

Login Page

Browse Services

Home Page

yes Customer Admin No Yes

Manage your account info

Order Confirmation

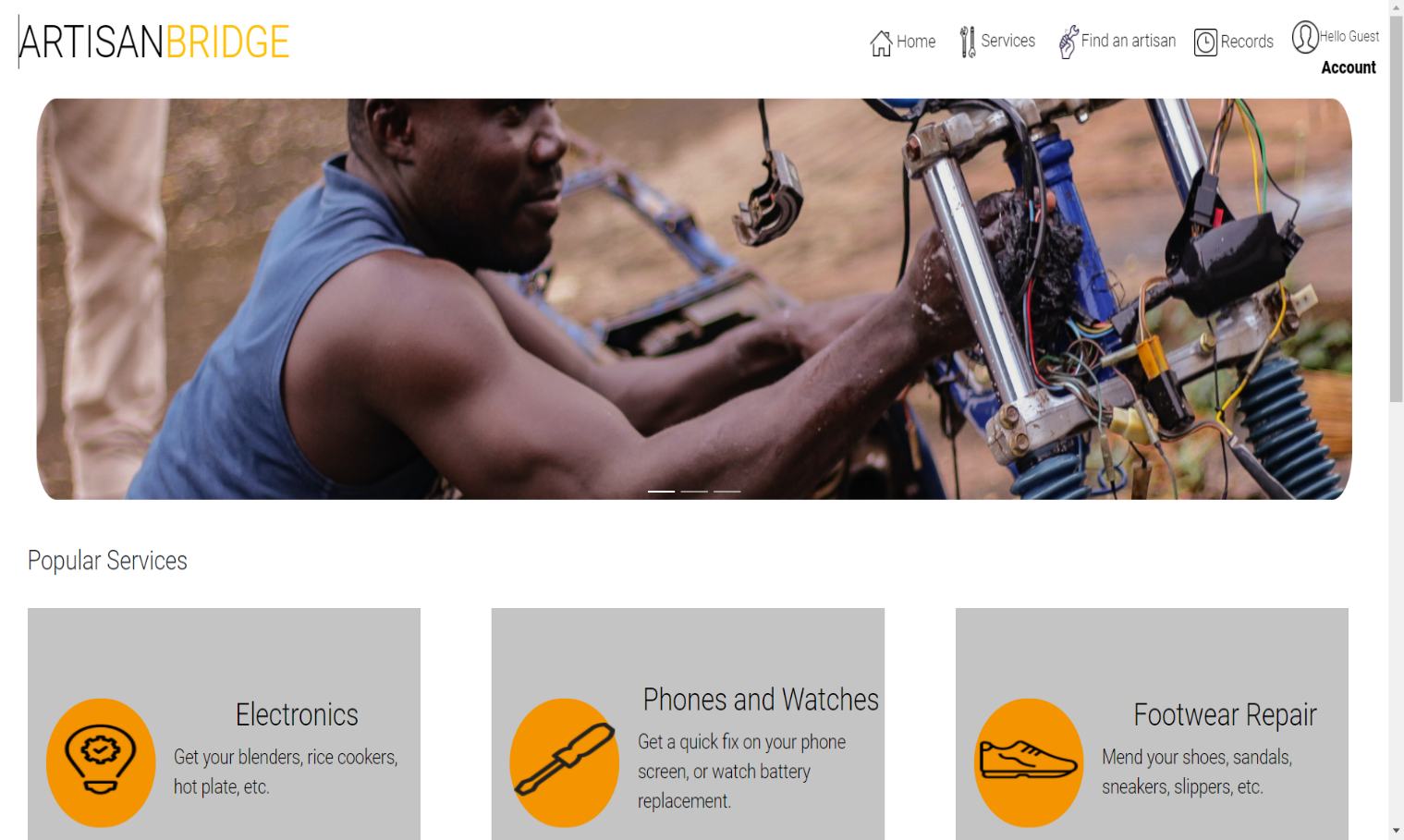
Continue using website?

Manage Orders and Payments

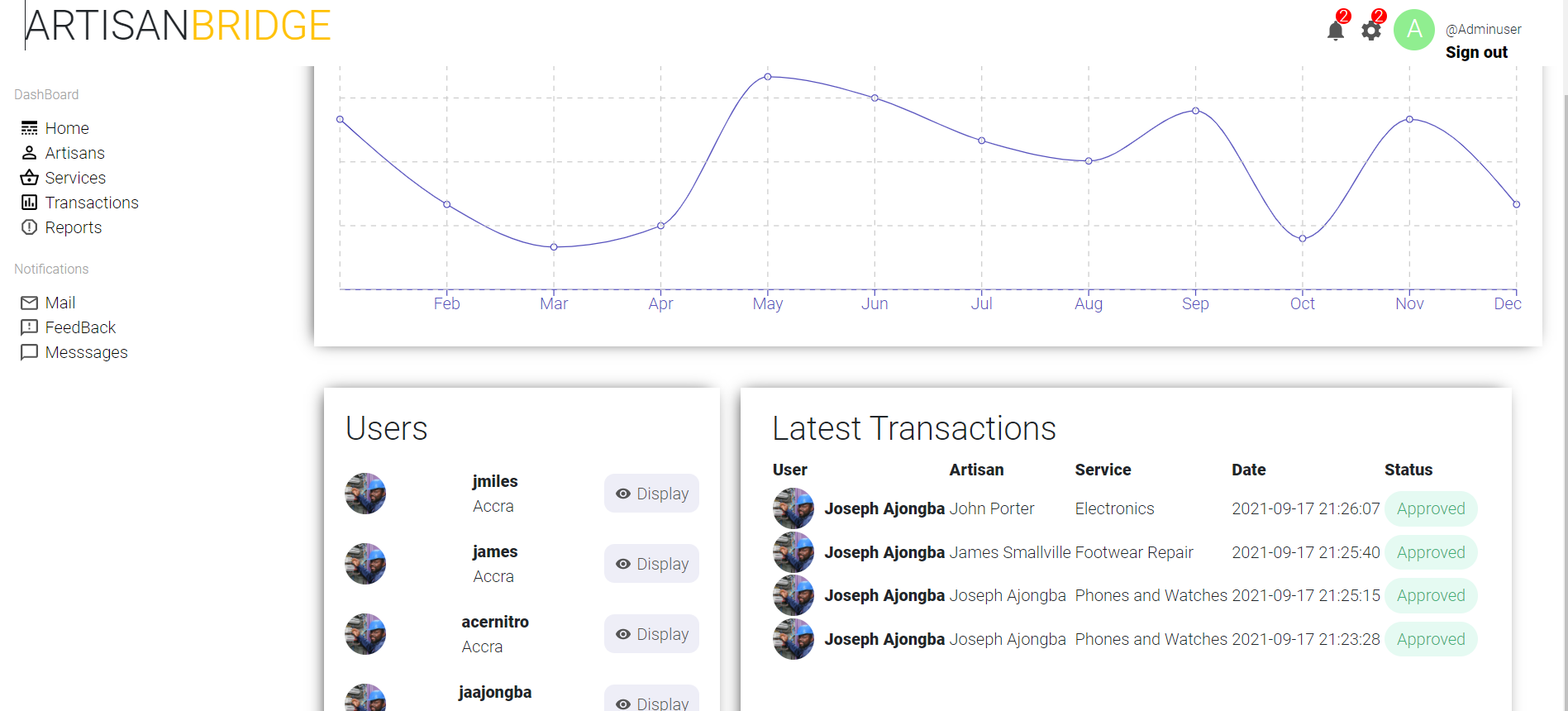
User logs out

**5. Appendix**

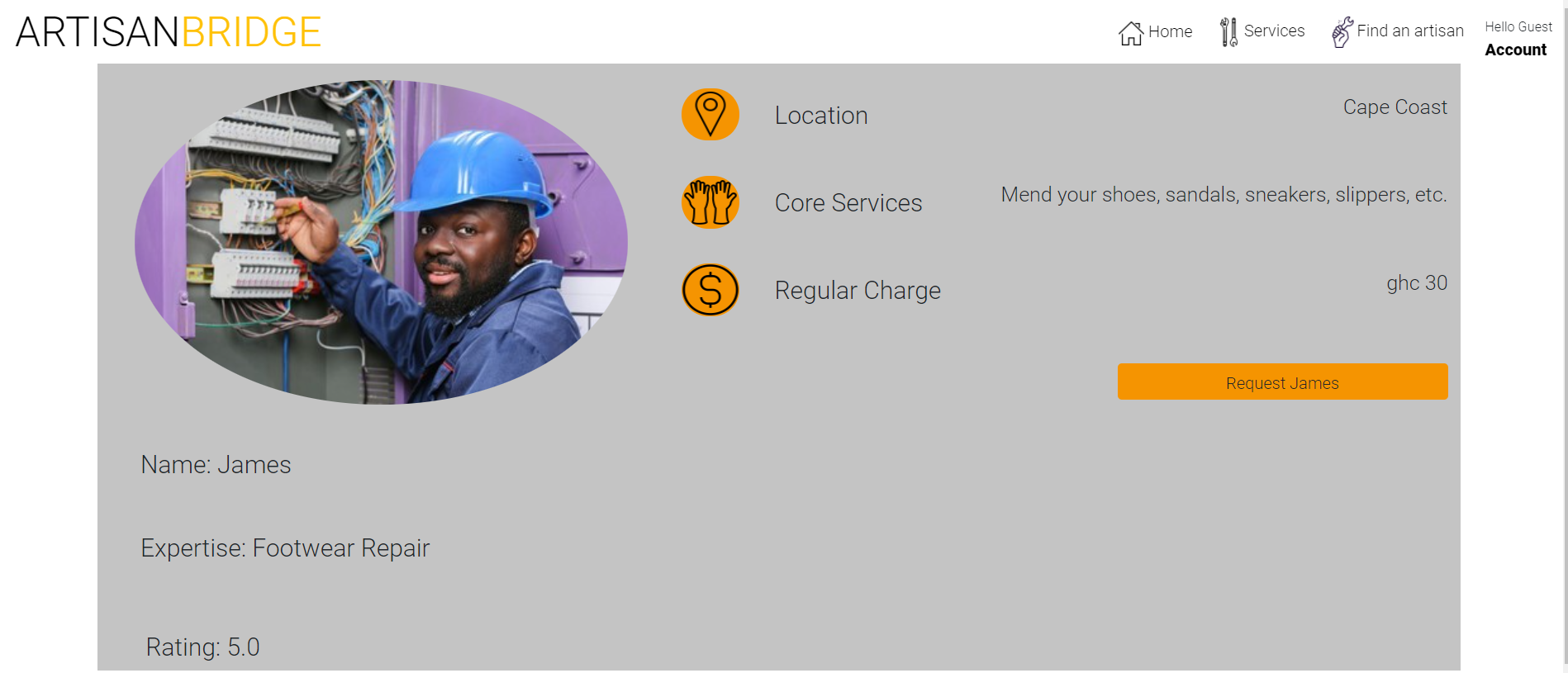
**5.1 Some Pictures Of The Web Application**

****

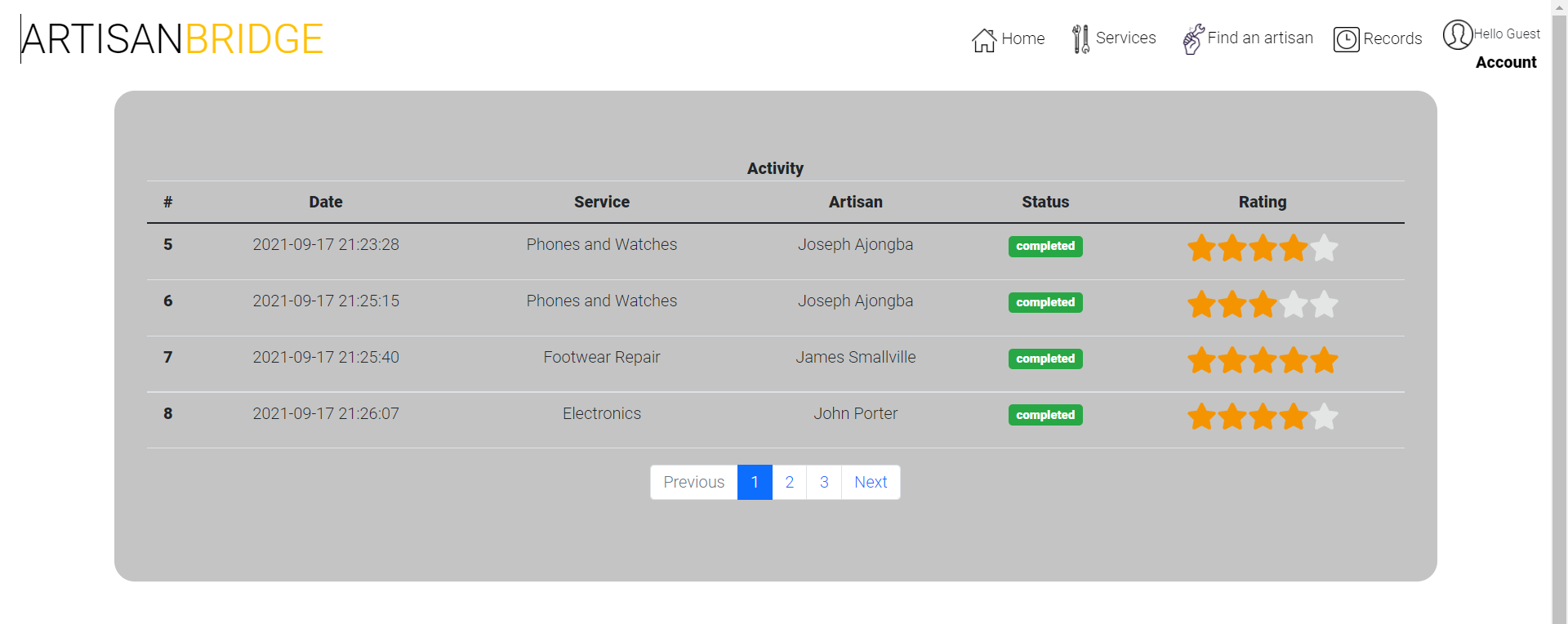
HomePage



Admin page



Artisan Select Page



Records Page

**5.2 Link To GitHub Repository**

<https://github.com/Joseph-Ajegetina/COE356-Codefall_Phantom.git>

**5.3 Link To Artisan Bridge Webapp**

http://artisan-bridge.herokuapp.com/