CS 353 Project Final Report

Group 8

Maintenance Data Management System - FixItUp

Web Page Link: https://github.com/Seftali/CS-353-Project-Reports

Berk Ataç -	**21200623 - Section 1**
Eren Aytüre -	**21200559 - Section 1**
Umut Balkan -	**21401911 - Section 1**
Derviş Mehmed Barutcu -	**21302589 - Section 1**

1. Overview

FixItUp is a project that helps companies' repairment operations and interactions between them and their customers. The companies can add their employees in two category: customer service and technical staff. Their workloads are listed in the following paragraphs and the customers can sign in to the system and make requests.

The customers can request repairment about their products and they can track their processes and their status. They can read the reports written by technical staff and make decisions about them. Finally, they can accept the repairment or complain about the repairment.

Technical staff sees the repairment requests. After they examine the product they should fill the preliminary report and the final report to inform the customer regarding to their product and the parts that need to be changed.

Customer service engages when the repairment is done and the customer is not satisfy with the repairment. After customer reads the final report, if they are not satisfy with the service they click the complain button and related information will pass to the customer service. Product name and the problem, customer informations and the technical staff's information that repair that product. All in all, the customer service should contact with the customer and understand why s/he is not satisfy with the service.

2. Problems And Their Solutions During Implementation

•Biggest problem we faced during our implementation was our ER diagram. We needed to revise it before creating our tables. We could not make out relations and how to get the data we need with our old ER diagram. We also moved around some of the attributes and changed the overall structure of the diagram. Below is our new diagram

•Another problem we faced was how to store "status" attribute in our database and also how make change of status to have its own life cycle, meaning status of of product needs to have a certain order, user can not make it what s/he wants by simply editing it, system needs to do it on its own. We solved this problem by storing status attributes as integer values (1 to 8 to be exact). We than, using php if statements, defined what these status are with respect to their int values.

•Same faced a similar problem for userType attribute which determines which user (customer, cust Service, tech staff) is logged in. We again stored int values for figuring out which user is logged in; 1 is for customer, 2 is for customer Service and 3 is for tech staff.

•Another problem we faced was about Reports. In our site, we have two reports which are both written by tech staff for customer to see. One preliminary report and one final report. At first we stored reports as attributes under Repairment table thinking “every pairments has a report”, but this thinking was wrong and we decided to make reports a new table instead, with its own attributes. This can be seen in our revised ER diagram. This way we stored reports more efficiently and just like we did for userType, we added an attribute to Reports table called repType(int) which determines if a report is preliminary(1) or final (2). In our gui we also separated these reports.

•Another problem we faced was how to prevent users (a customer for exp) to enter another users page(to a tech staff page). Our sites runs locally, so just typing the page name to address bar causes one user to be able to access any page any time. Basically we faced an authorization problem. We solved this by checking the userType of the current user logged in by finding it using the userID in the current session. We used php for coding this.

If for example a users type is 2, which shows that user is a customerService and s/he tries to enter the customer page, user is rejected and directed to login/sign up page (index.php). We wrote this check for every page.

• For retrieving specific data from our tables, we used “fetch_array” function, for inserting data or finding data, we used insert_id (which selects the last id, since we used auto increment for our primary keys.

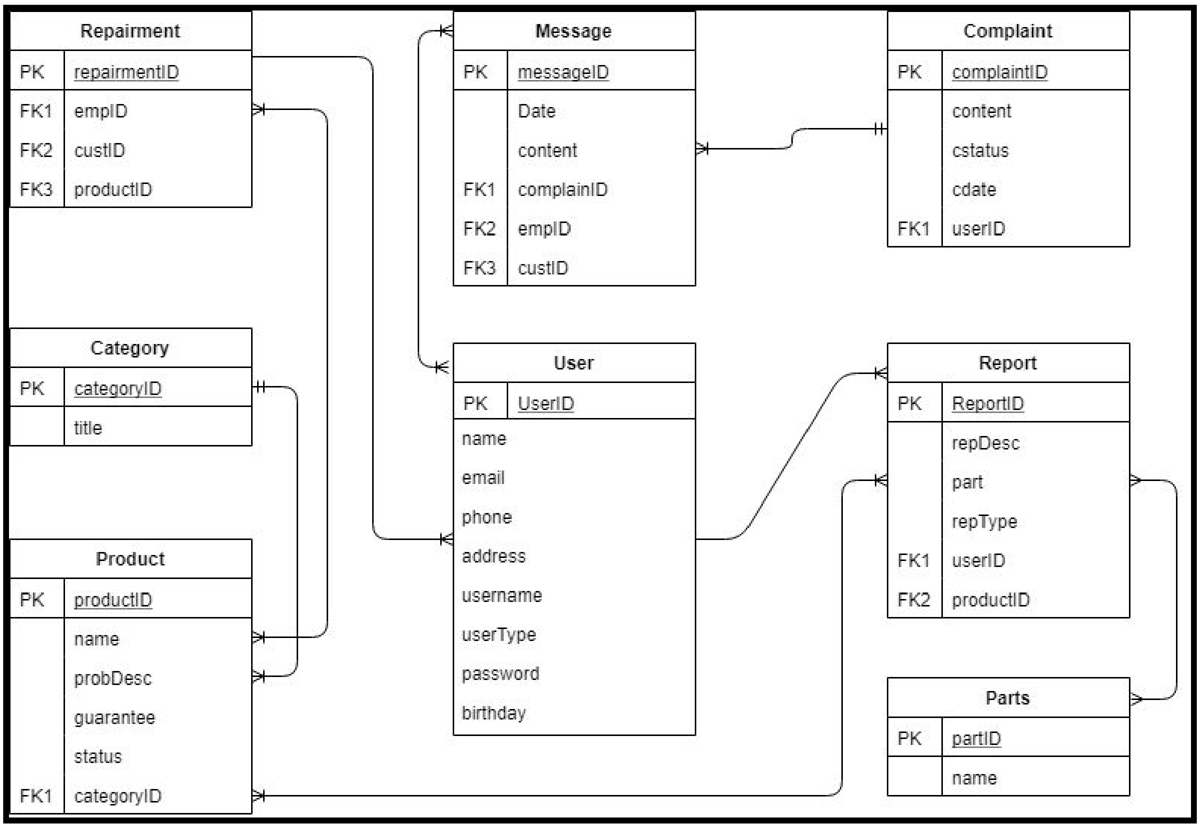
•We used “implode” function for storing separate product parts with “,” between them to product table.

•For connecting our database to our php files we created settings.php with server info and included it to our every page.

•PHP, MySQL, JavaScript
•Pingendo(GUI), ATOM, PhpStorm, VisualStudio

3. Final E/R Diagram

3.1. E/R Diagram



Repairment	
PK	repairmentID
FK1	empID
FK2	custID
FK3	productID

Message	
PK	messageID
	Date
	content
FK1	complainID
FK2	empID
FK3	custID

Complaint	
PK	complaintID
	content
	cstatus
	cdate
FK1	userID

Category	
PK	categoryID
	title

User	
PK	UserID
	name
	email
	phone
	address
	username
	userType
	password
	birthday

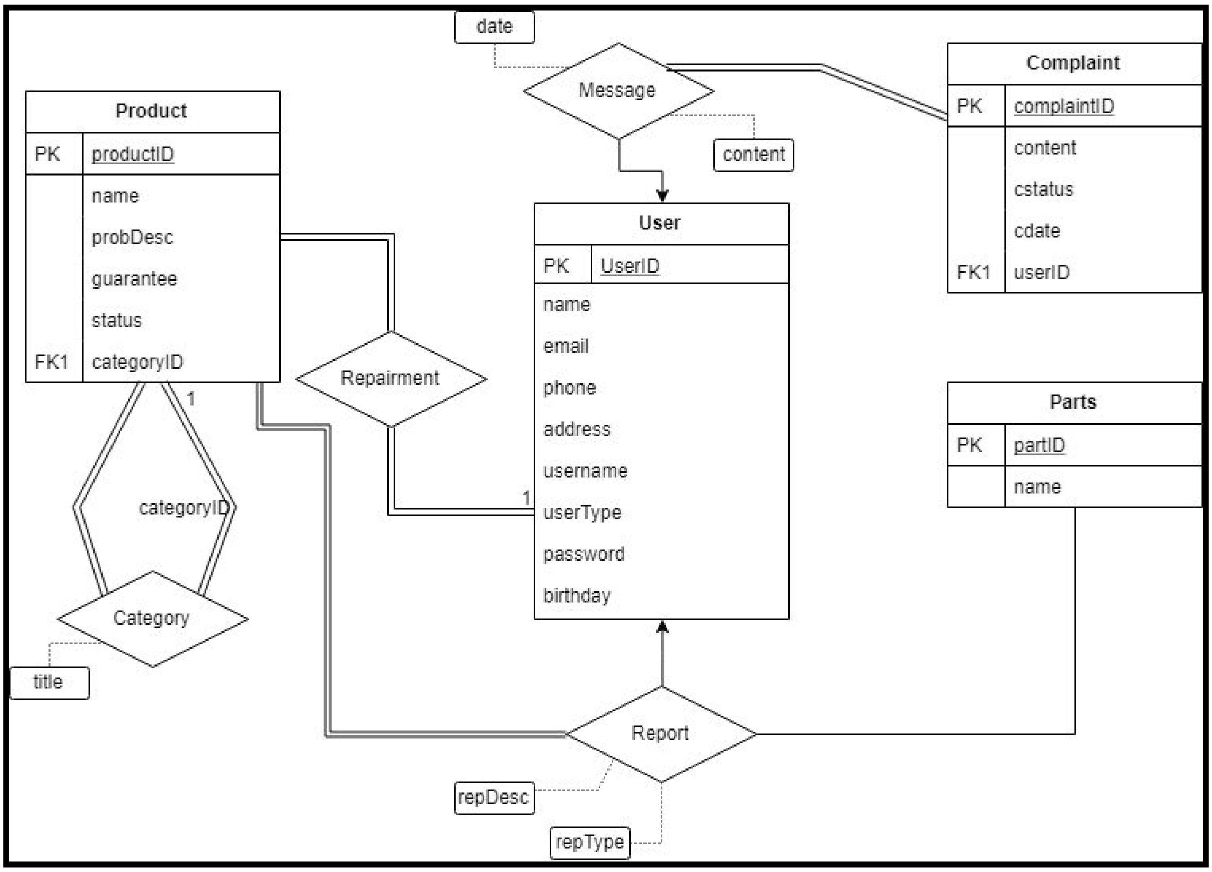
Report	
PK	ReportID
	repDesc
	part
	repType
FK1	userID
FK2	productID

Product	
PK	productID
	name
	probDesc
	guarantee
	status
FK1	categoryID

Parts	
PK	partID
	name

This E/R diagram demonstrates how our system is built. we use this model because it is easy to implement and looks simpler than the second E/R model.

3.2. E/R Diagram



We added this model because this is how we learned. But we encountered some problems during the implementation phase so we changed it into as we call industrial E/R model.

4. List of Tables

4.1 User

Relational Model:

User(<u>UserID</u>,name,phone,address,username,userType,password,birthday)

4.2 Product

Relational Model:

Product(<u>productID</u>, name, probDesc, guarantee, status, categoryID)

Foreign Keys:

FK: categoryID references Category(categoryID)

4.3 Parts

Relational Model:

Parts(partID,name)

4.4 Complaint

Relational Model:

Complaint(complaintID, content, cstatus, cdate,userID)

Foreign Keys:

FK: userID references User(userID)

4.5 Category

Relational Model:

Category(categoryID, title)

4.6 Message

Relational Model:

Message(messageID, Date, content, complaintID, empID, custID)

Foreign Keys:

FK: empID references User(userID)

FK: custID references User(userID)

4.7 Repairment

Relational Model:

Repairment(repairmentID, empID, custID, productID)

Foreign Keys:

FK: productID references User(productID)

FK: empID references User(userID)

FK: custID references User(userID)

4.8 Report

Relational Model:

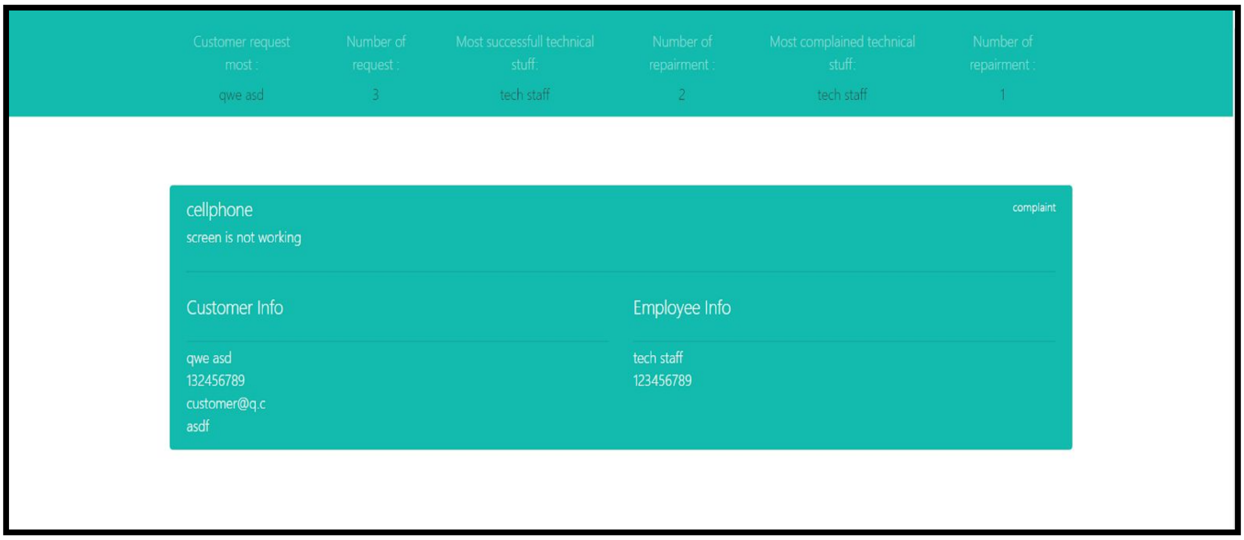
Report(reportID, repDesc, part, repType, userID, productID)

Foreign Keys:

FK: userID references User(userID)

FK: productID references Product(productID)

5. Reports



Customer request most :	Number of request :	Most successfull technical stuff:	Number of repairment :	Most complained technical stuff:	Number of repairment :
qwe asd	3	tech staff	2	tech staff	1

cellphone

screen is not working

complaint

Customer Info

qwe asd
132456789
customer@q.c
asdf

Employee Info

tech staff
123456789

In the picture above, this is the main page of the customer service that handles the complaints.

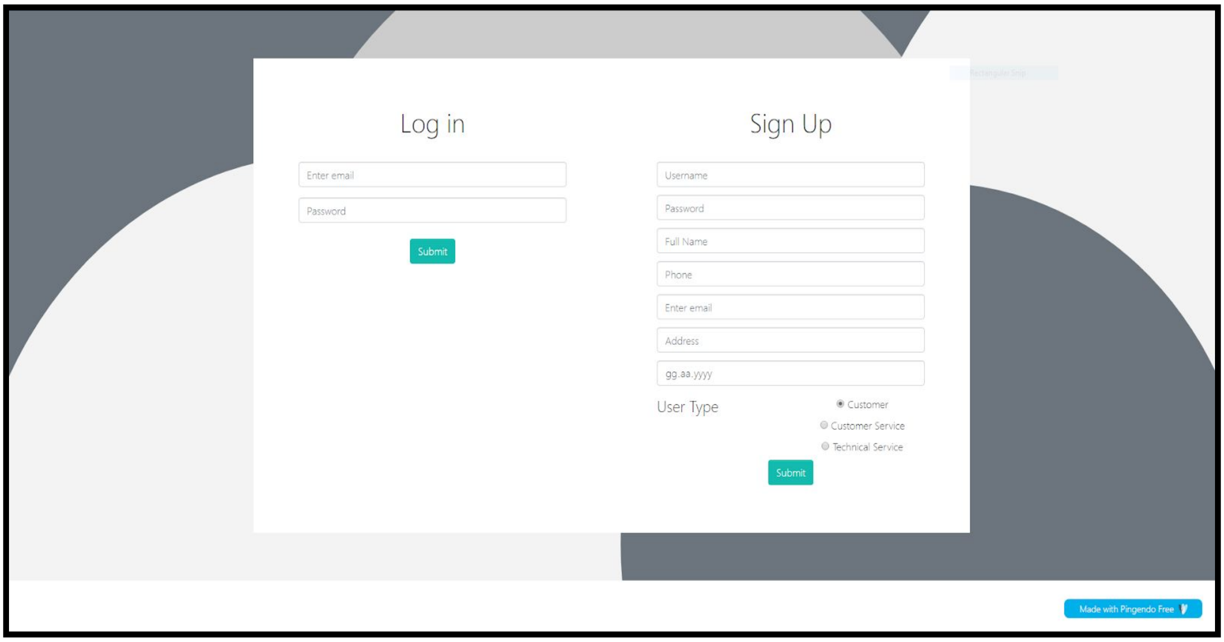
The navigation bar of the page has some statistics about the company. For example, it shows the name of the customer who request repairment mostly and his/her request number, the name of the most successful technical staff and his/her number of repairments and the most complained about technical stuff's name and number of repairments.

Since the customer service handles the complaints, it is her/his responsibility to see and notify their superiors about the successful and complained about technical stuff. That is why they need to keep track of the other personals.

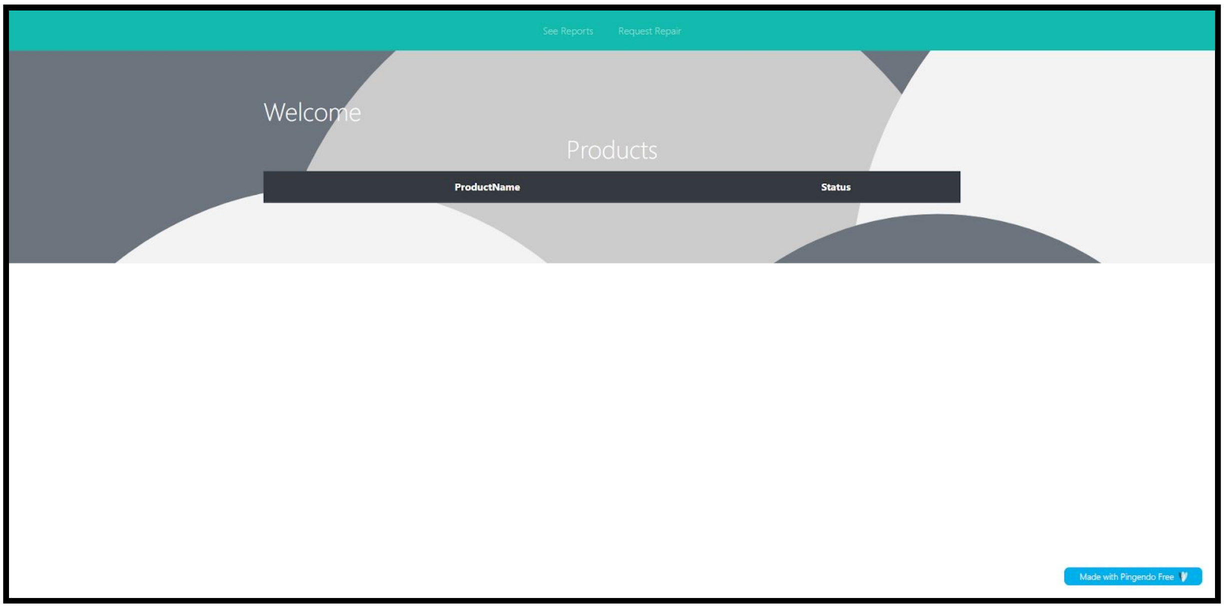
6. GUI and its manuel

6.1. Pages according to process flow

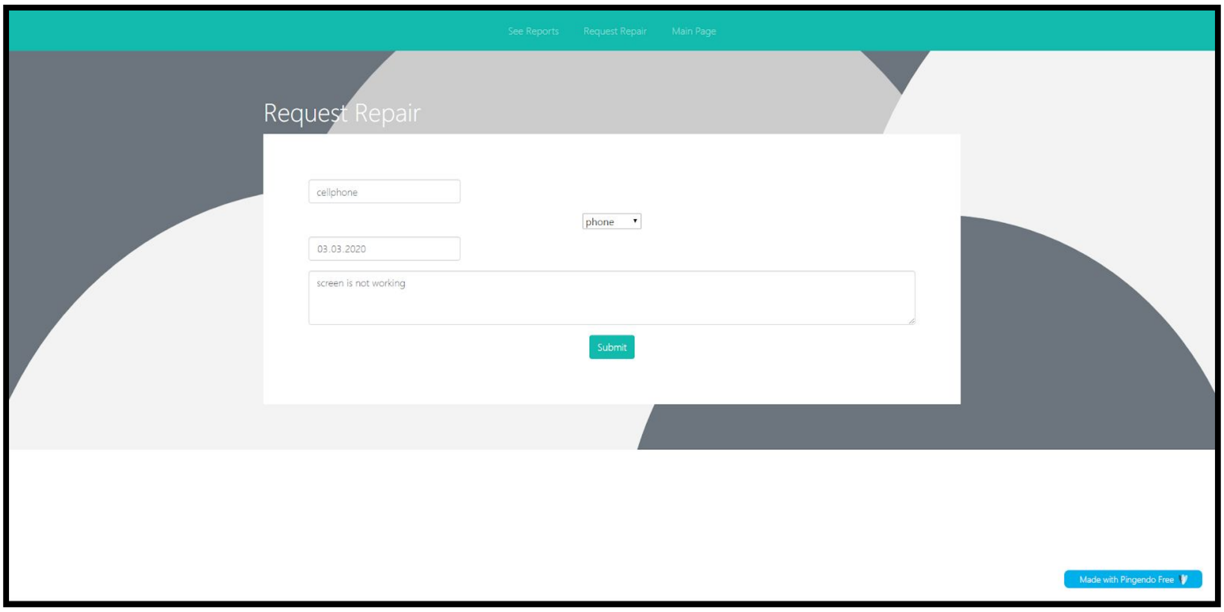
In this page, users can sign up as customer, customer service or technical service. First, they need to fill the information and select related user type, then click the submit button to register to the system, and they can log in with their email address and password.



The below page is **mainpage of the customer** before requesting repairment.



If the customer has a product that needs to be repaired, s/he can **request repairment**. The below picture demonstrates the request repairment screen. S/he needs to fill all the boxes. First, s/he writes the product and selects its category, then s/he selects the end of the guarantee day of the product, and writes description about complaints about why it needs to be repaired.



Request Repair

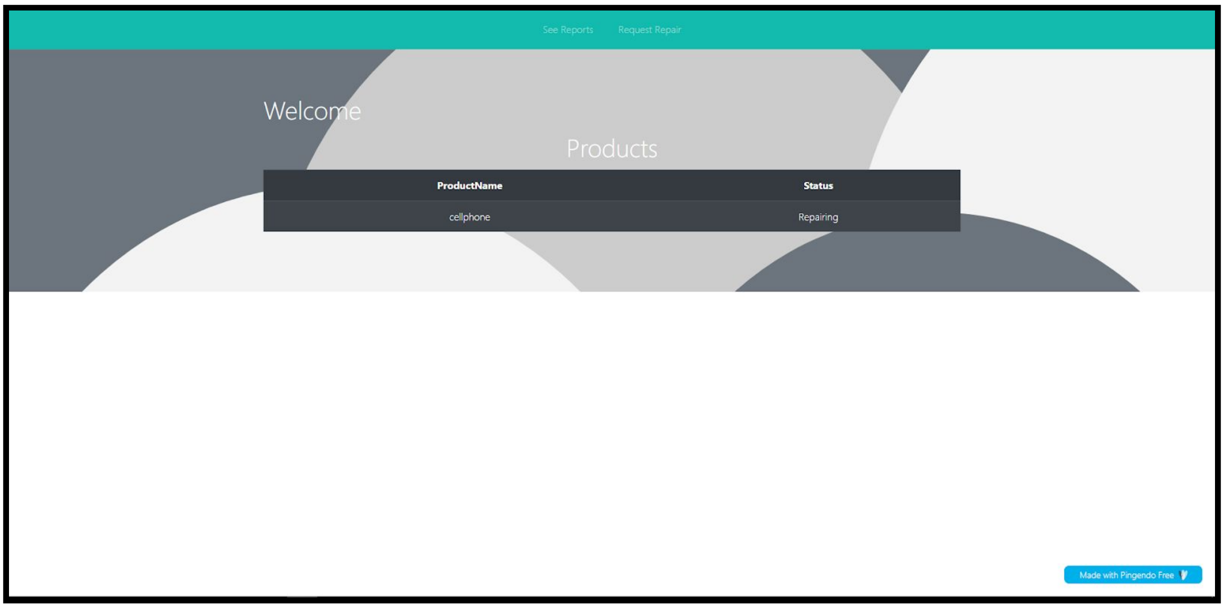
cellphone

phone

03.03.2020

screen is not working

After requesting the repairment, the mainpage becomes like the picture below. Initially the status of the repairment is set to "repairing" and it will change during the repairment process so that the customer can track the process.

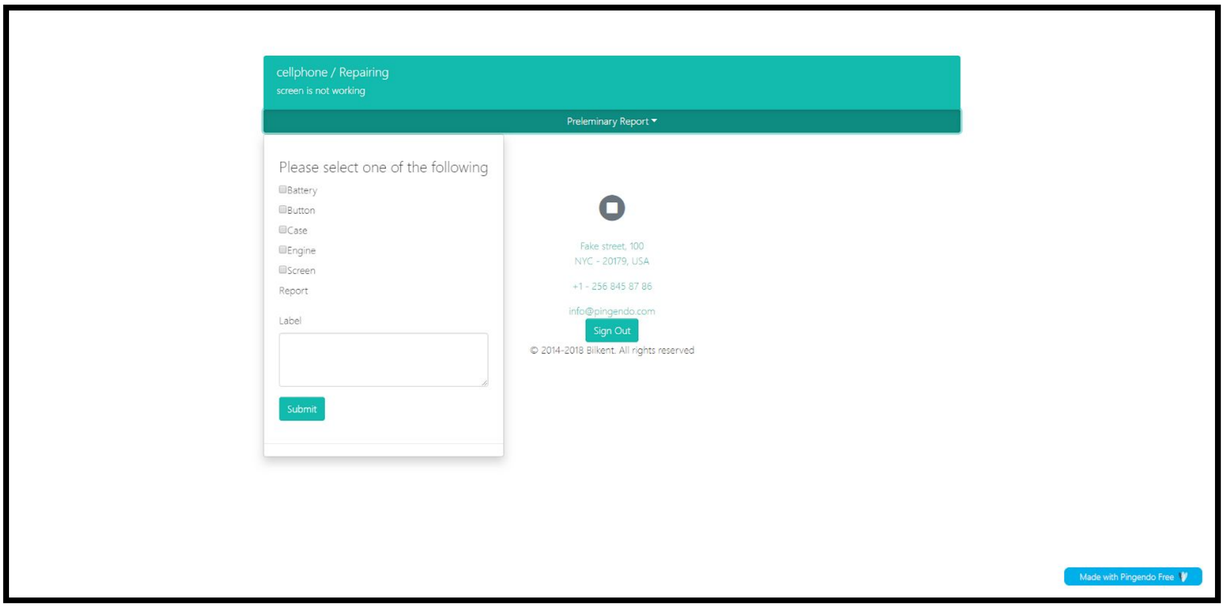


Welcome

Products

ProductName	Status
cellphone	Repairing

After the customer sent a repairment request, random **tech staff** is selected and the request is assigned to s/he. Tech staff's main page is below. After tech staff examine the product, s/he must fill preliminary report about how it will be repaired and which parts are needed to be changed are filled and selected by tech staff and submit.



cellphone / Repairing

screen is not working

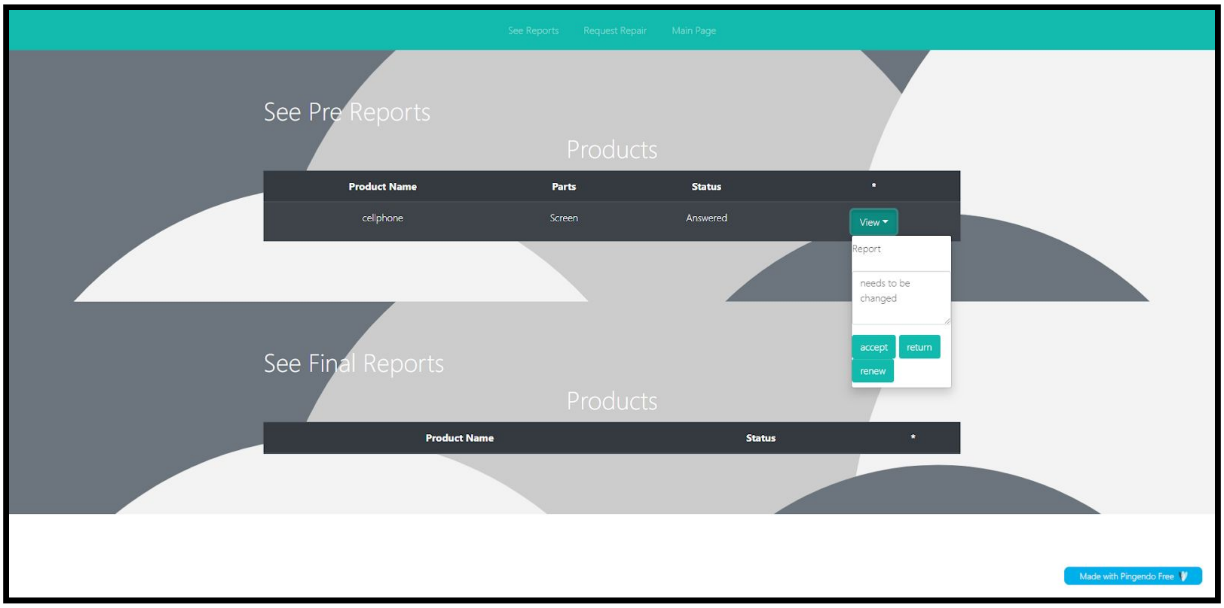
Please select one of the following

☐Battery
☐Button
☐Case
☐Engine
☐Screen

Report

Label

When the preliminary report is submitted, the status of the product is changed as “answered” and customer can understand that tech staff examined the product and wrote a report about it. When the customer enters the see reports page, s/he can see two sections in this page, pre-reports and final reports. When a customer clicks the view button, s/he can see the report and buttons related to her or him decision about repairment.



See Pre Reports

Products

Product Name	Parts	Status	*
cellphone	Screen	Answered	View

Report

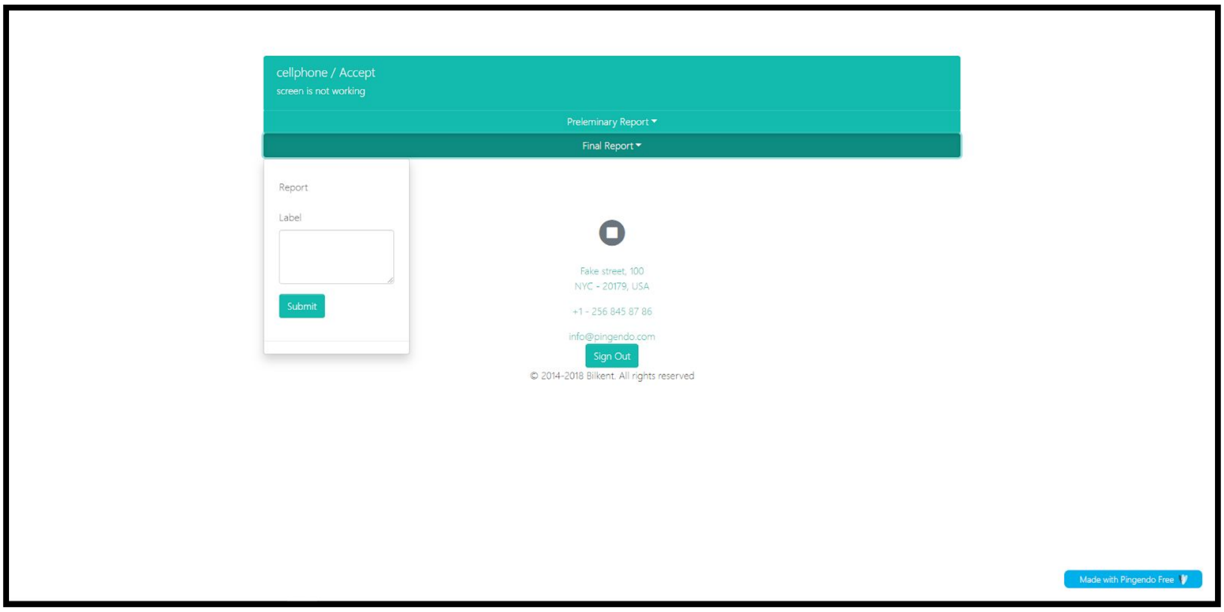
needs to be changed

See Final Reports

Products

Product Name	Status	*

After the customer selects what s/he wants, the final report button will occur in the assigned tech staff’s mainpage. By clicking the final report button, the tech staff can fill the final report about the repairment process and submit to customer.



cellphone / Accept

screen is not working

Report

Label

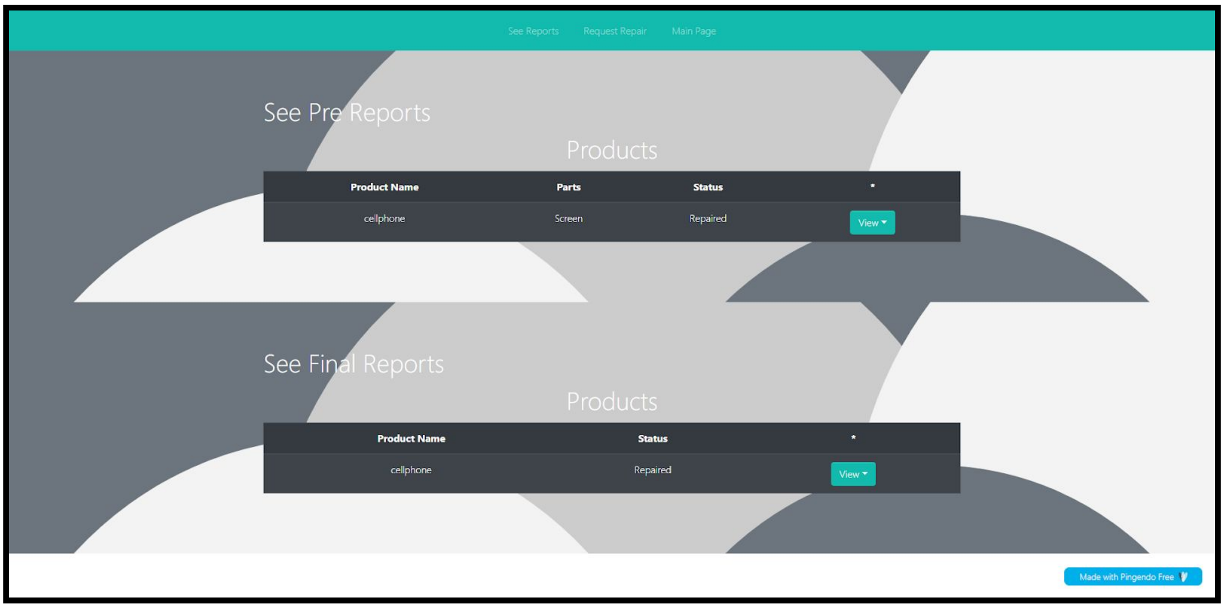
Fake street, 100
NYC - 20179, USA

+1 - 256 845 87 86

info@pingendo.com

© 2014-2018 Bilkent. All rights reserved

When the final report is filled by the tech staff, the customer can see that report from see reports page. In this page, the customer can see both preliminary report and final report about the product. By clicking the view button in final reports section, the customer can see the final report. After reading the final report, the customer can either accept the repairment and process is finished or complain about the repairment.



See Pre Reports

Products

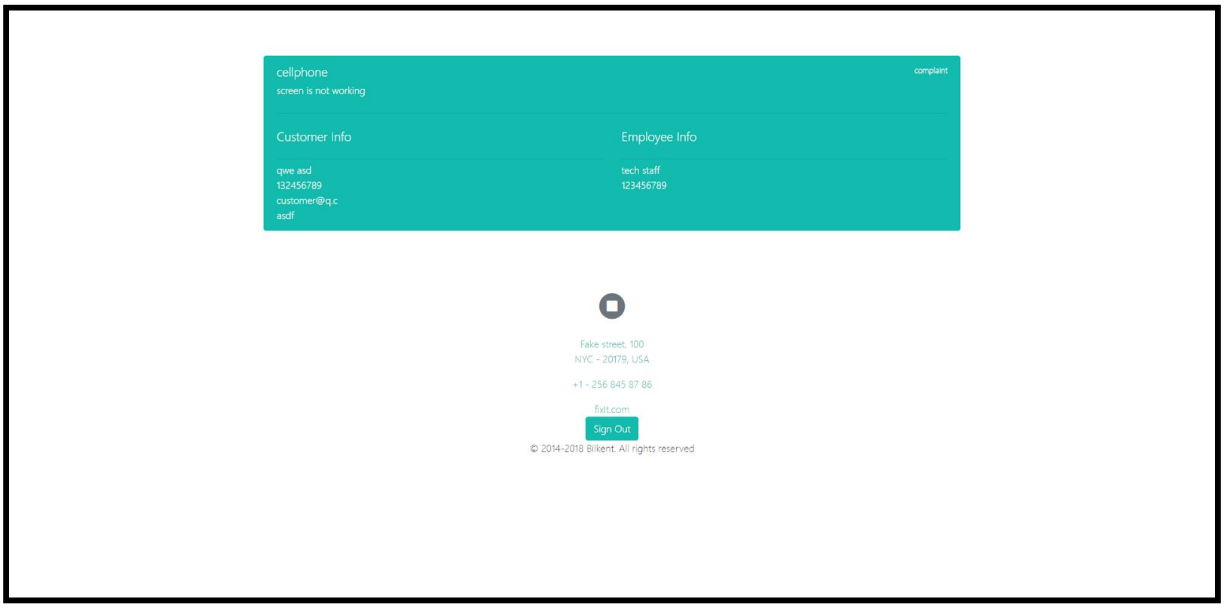
Product Name	Parts	Status	*
cellphone	Screen	Repaired	View ▾

See Final Reports

Products

Product Name	Status	*
cellphone	Repaired	View ▾

If the customer makes a complaint about the repair, that complaint is assigned to a random **customer service** to contact with the customer. The picture below is customer service mainpage. The customer service can see the product and description about its problem, customer information and the tech staff who repaired the product.



cellphone
screen is not working

Customer Info

qwe asd
132456789
customer@q.c
asdf

Employee Info

tech staff
123456789

Fake street, 100
NYC - 20179, USA

+1 - 256 845 87 86

fixIt.com

© 2014-2018 Bilkent. All rights reserved

7. Contributions

Berk Ataç

- GUI
- SQL Statements
- PHP
- ER Diagram

Umut Balkan

- SQL Statements
- ER Diagram

Eren Aytüre

- GUI
- ER diagram

Derviş Mehmed Barutcu

- GUI
- SQL Statements
- PHP
- ER Diagram