

TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Mihkel Väli 194089IADB

Repair job management application for electronics repair company

Web applications with C# course project

Supervisor: Andres Käver

Tallinn 2024

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Mihkel Väli

04.03.2024

Table of contents

1.1 Currently used solution.....	5
1.2 Proposed solution	5
1.3 MVP features	6

List of figures

Figure 1. Entity relationship diagram	7
Figure 2. Application login view sketch.....	8
Figure 3. Technician logged in home view	9
Figure 4. Technician add new repair view	10

1 Introduction

The company is specialized in repairing and servicing devices made by Apple Inc. [1] be that a laptop, smart watch, or a phone. At this moment Apple has more than 2.2 billion active devices worldwide [2] each of which could possibly need a repair at some point in its lifespan. Each repair has its own peculiarity although to a bystander they might seem just alike but before every repair there are certain aspects which must be considered by the repair technicians to offer the best possible solution to the client. For example, when a phone has been dropped resulting in a broken back glass it's utmost important to know whether the back glass has been replaced before or not since that will greatly affect the complexity and time cost of the repair and therefore plays an important role in giving out accurate estimated repair time that the client is usually the most interested in. This is just one of the reasons why it is crucial to have a convenient way to manage and keep track of each repair.

1.1 Currently used solution

At this moment the information of each repair is stored in excel tables. This is by no means a handy and good solution long term. It takes a lot of time to insert data about a repair task this way and is also prone to human error. The current solution also lacks a way for the company's clients to see their device's service history and the progress of an ongoing repair job.

1.2 Proposed solution

New proposed solution will benefit both, the company's technicians carrying out the repairs and the clients who want fast and convenient service. The solution is an application that conforms to the constraints of REST architectural style commonly known as REST API [3]. As mentioned above the new solution will have two primary user groups: repair technicians and clients. As the time to develop this application is limited due to the course duration let it be noted here that the features benefitting technicians are

more favoured since they are the user group suffering from the current solution the most and the company's clients will benefit from this new application even if there are no visible changes to them.

1.3 MVP features

Application's desired functionality is represented by features that are divided into categories by their importance.

Must have features (MVP):

As a technician I want to log in to my account.

As a technician I want to add a new repair job.

As a technician I want to see older repair jobs and filter them by device serial number.

As a client I want to see the status of my ongoing repair jobs.

Nice to have features (not necessary for MVP):

As a client I want to see the history of my repair jobs.

As a client I want to see the total cost of my repairs.

As a technician I want to see how much time providing a certain service has taken me on average.

2 Entity relationship diagram

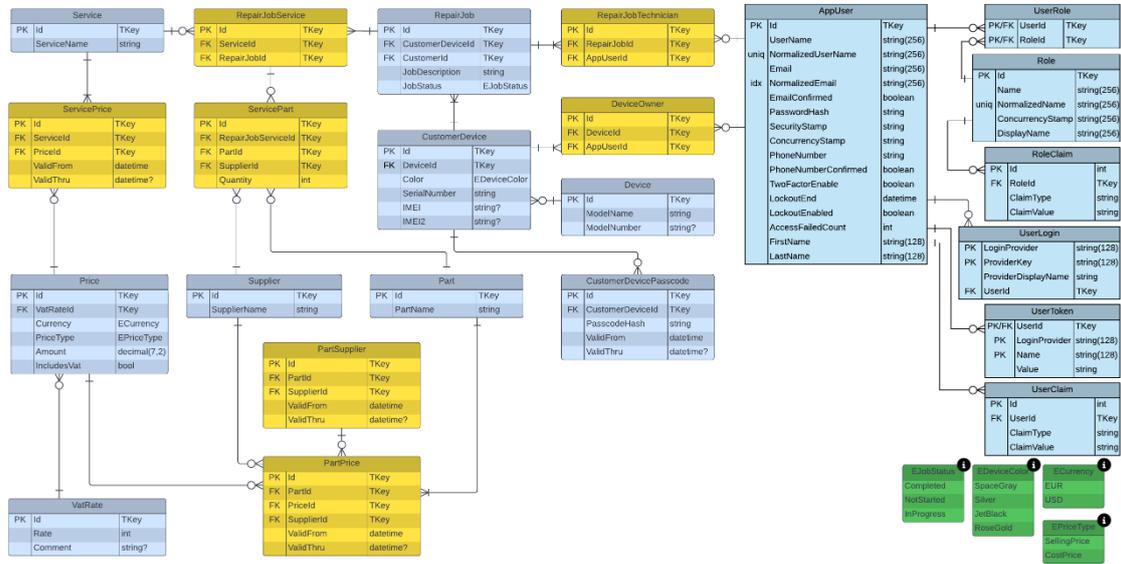


Figure 1. Entity relationship diagram

This diagram is not final and some parts of it may change over the duration of the project. Yellow coloured tables are many: many in-between tables that are not counted as separate entities in this scope. Striped, blue tables are considered entities. Blue tables are for identity and green tables for enums. Enum lists in the diagram are not complete and have selected values just to illustrate their meaning.

3 Positive flow screen sketches

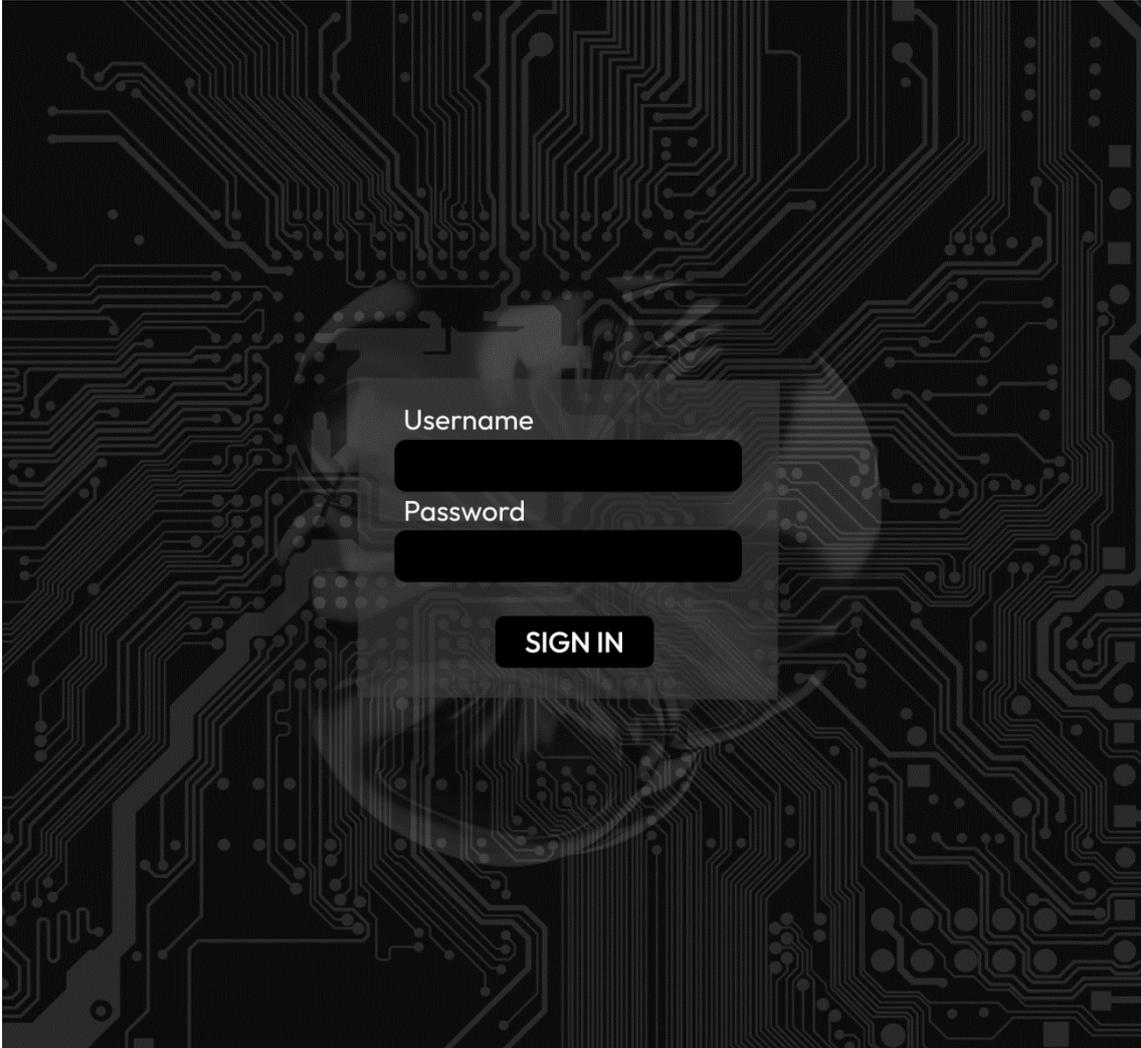


Figure 2. Application login view sketch

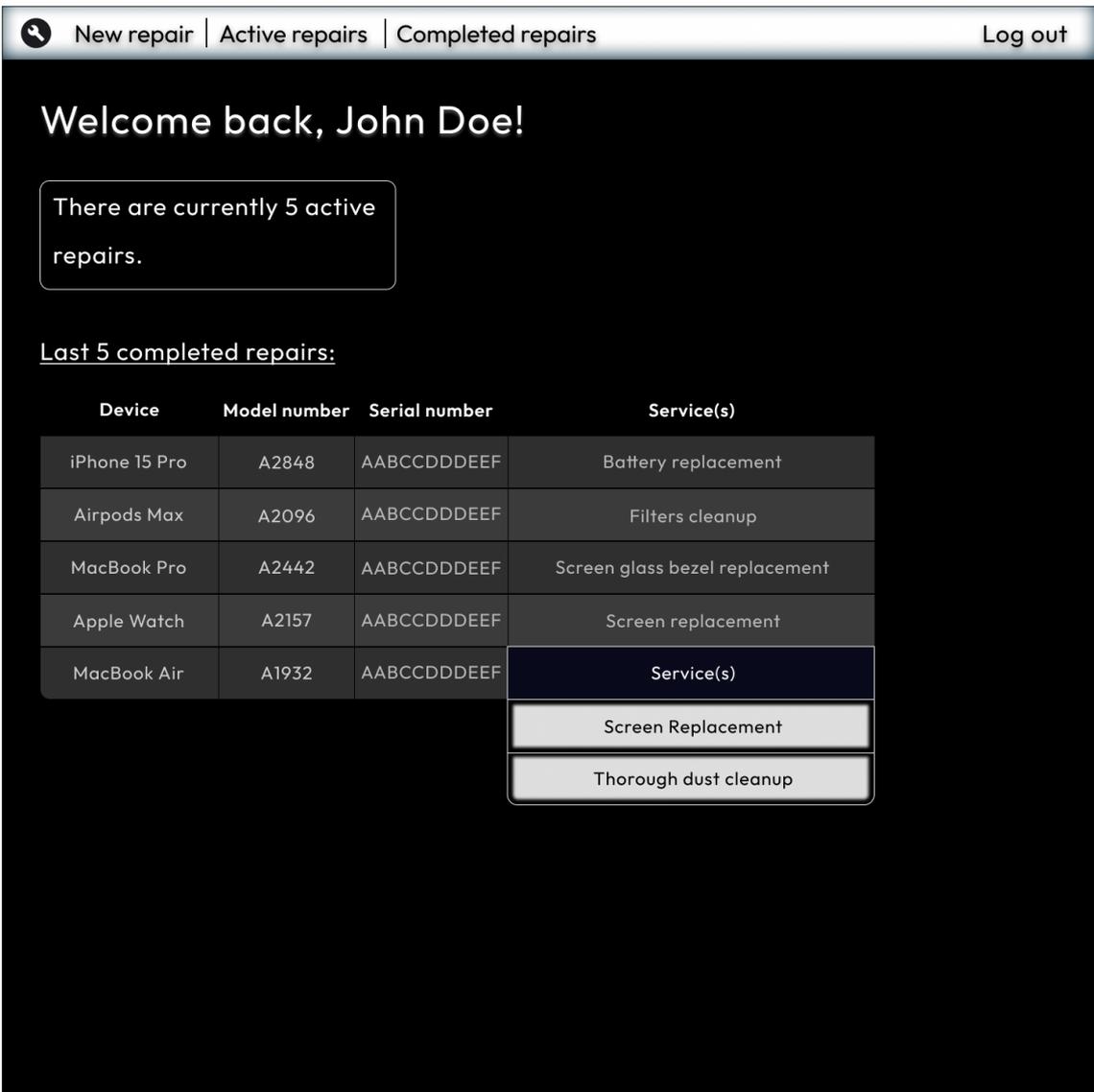


Figure 3. Technician logged in home view

🔑 **New repair** | Active repairs | Completed repairs
Log out

Device details

Serial/IMEI	Device	A-Model	Color	Passcode
CO2LX899F8J4	iMac ▼	A1314	Silver ▼	9753

Client information

Email	Phone number	Client type
john@doe.gmail.com	+37212412321	Private person ▼

Job details

<p>Job description</p> <p>Client is not satisfied with the speed of the iMac and wants us to install a SSD and perform a clean install to MacOS High Sierra.</p>	<p>Additional information</p> <p>Device has minor wear, scratches on the stand.</p>
--	---

Repair Cost

Service/Part	Service/Part price		
SSD Samsung EVO 870 1TB	104.90 €	✓	
			Total repair cost 42.00 €

Added services/parts

Service/Part	Price
Software install	42.00 €

Figure 4. Technician add new repair view

References

- [1] "Apple Inc. - Wikipedia," 2024. [Online]. Available: https://en.wikipedia.org/wiki/Apple_Inc. [Accessed 04 03 2024].
- [2] Apple Inc., "Apple reports first quarter results," 01 02 2024. [Online]. Available: <https://www.apple.com/newsroom/2024/02/apple-reports-first-quarter-results/>. [Accessed 04 03 2024].
- [3] Red Hat, "What is a REST API?," 08 05 2020. [Online]. Available: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>. [Accessed 04 03 2024].