**Car Insurance Registration Journey**

**Crosure-2022**

# Contents

Introduction 3

goal 3

Customer Requirements 4

Diagrams 5

**Introduction**

Looking for an agency to register any car`s insurance is the main problem as soon as the customer gets a new car. While searching, people lose their time and money as well as not being able to drive their newly bought car.

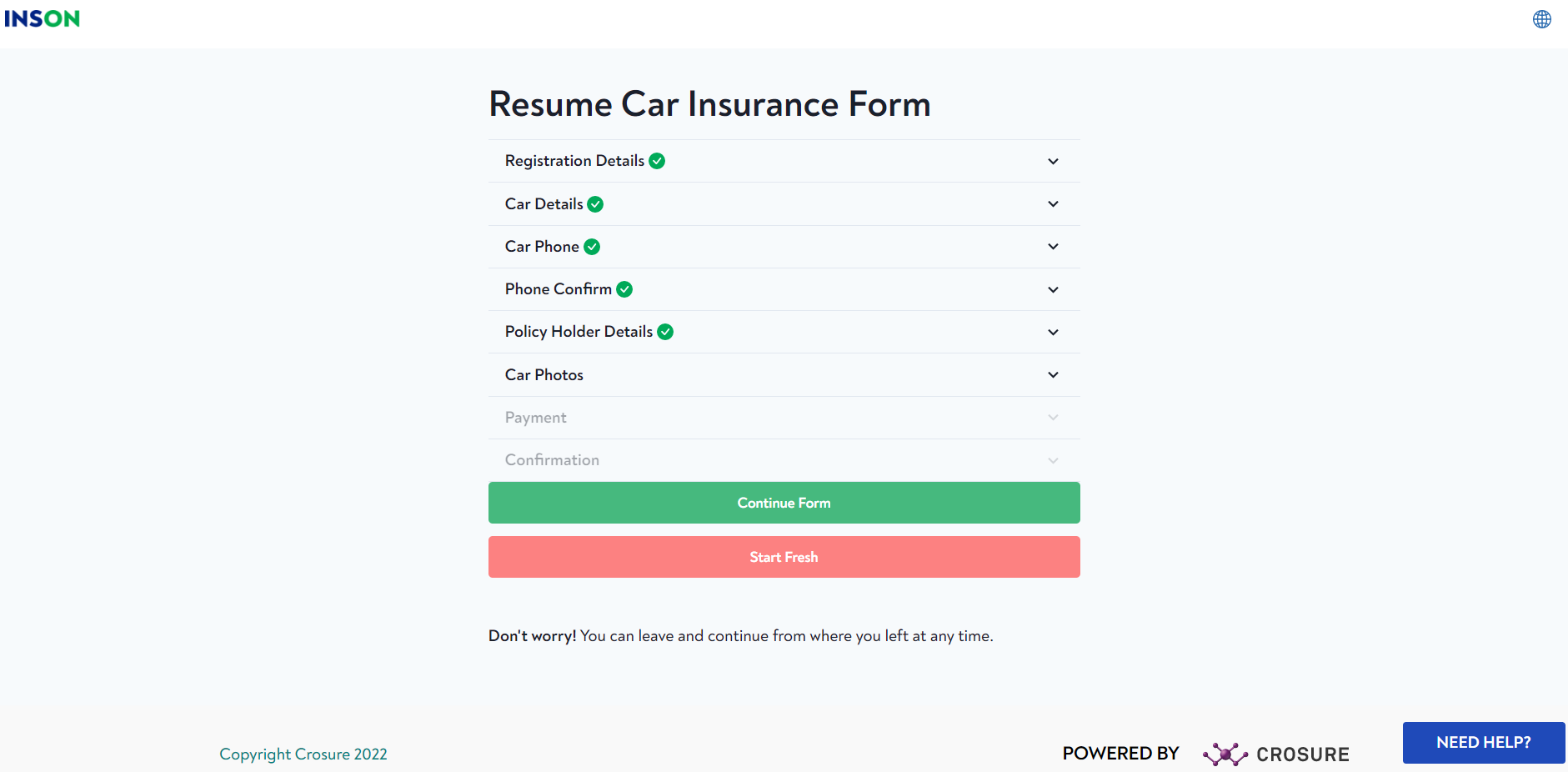
Because of this, Crosure made everyone`s life easy and developed a new product, which offers to customers of the Insurance company to self-register and buy their insurance policy from the company`s website.

**Goal**

Our main task is to develop the best technical adventure possible for the user to buy their Insurance policy without any difficulty while saving their time and money. For this, our team created a specific algorithm to calculate and secure the process of registering without any difficulties. We require the team of developers to develop a form that will be used by customers when they are buying their insurance policy.

**Customer Requirements**

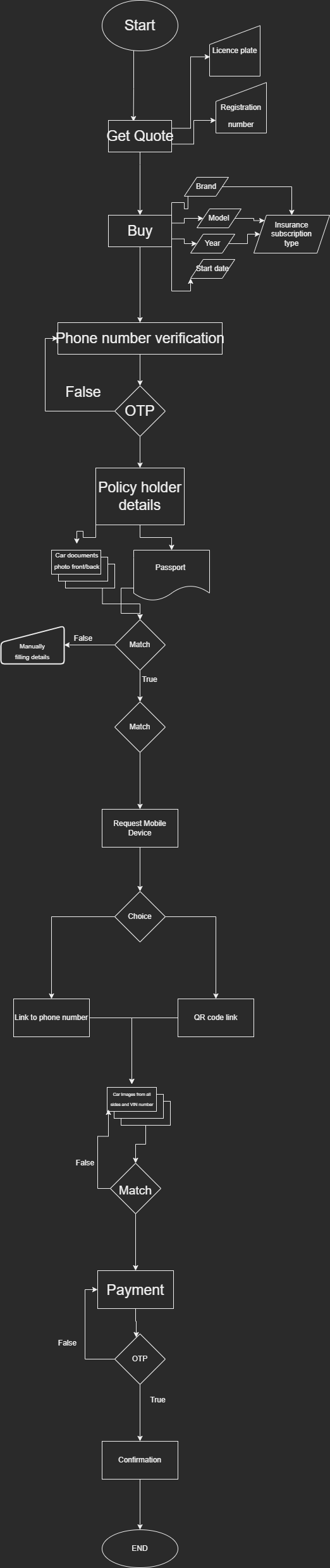
Below are the steps that should be developed to use when buying car insurance.

Our customer is requiring the process will start point, when a user opts to press the start button for registering their car insurance from their company`s website, followed by that users will be required to fill out the car details form, and confirm their phone number and their details. Image 1.1 shows the structure of the process.

**Diagrams**

The diagram below shows how the process of insurance registration should run

Our customer has required to develop a web application, which will be opened by pressing the start button. The application should consist of several stages and will require users to enter their details. Also, the application should save each user`s information while registering and the user should be able to continue a started application.



Our customer wants the first stage as optional and users should be able to fill in their license plate and registration numbers with the skip button available if they prefer. However, the following stages should be mandatory and require checking in each stage.

In the second stage, an application should calculate the sum of the insurance policy by the user`s input as well as the subscription type. Required user inputs should be the brand of the car they want to register, the model of the car by checking available models inside of chosen brand by checking those inputs, the subscription type menu should pup-up and the user should be able to pick one from the list. Finally, the user should be able to choose the starting date(dd/mm/yyyy) of the insurance policy.

In the third and fourth stages, a user should be required to enter their phone number and verify it through an OTP code.

In case of failure, the user should have the option to go back to the third stage(phone number entering) by pressing a button or re-entering the OTP code(the application is required to check the newly entered OTP code without loading the page).

If verification is successful, an application should move to the next stage and ask to upload images of policyholder details. In this stage, users should upload the documents of the car(front and back pages) as well as their passports. Which required to be checked for the details entered in earlier stages.

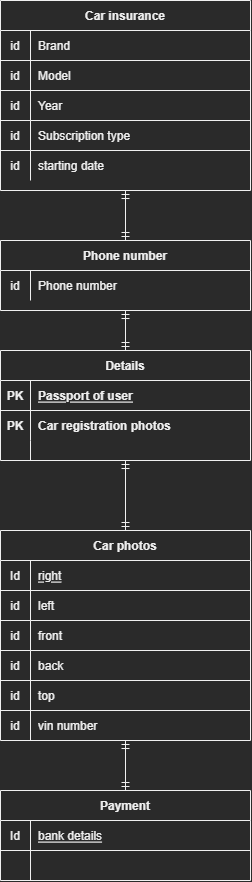
In case of failure, a user should have an option to enter details manually(car brand, model, production year) and they should be checked with the details as well.

In case of success, the application should ask for a mobile device and be continued on that device. Users should have 2 choices either sending a website link to their phone number or showing a QR code which provides a link to the next step in both cases.

A link should open a page of the application, which allows uploading all sides of the car as well as the VIN (should be taken real-time photos, and requires access to a phone camera). All files should be checked before processing to the next stage and in case of failure, they should be required to be re-upload.

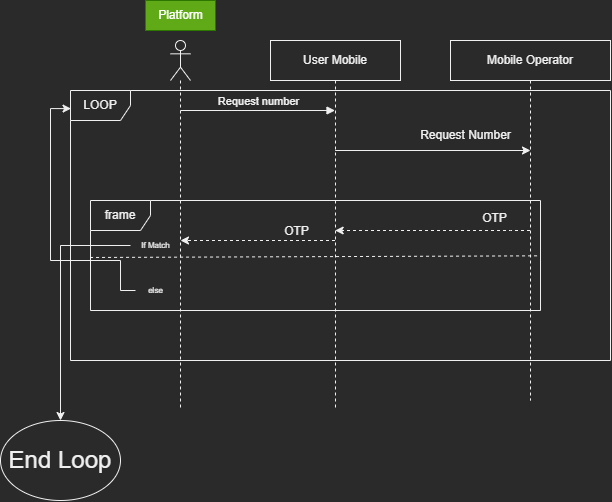
If successful, a user should be able to pay by entering their bank details and CVV. But, before confirmation application should require an OTP code from the bank. And the last stage should be confirmation of the Insurance policy with the details of the customer and the car.

The below diagram explains how Entities are related to each other in the database.



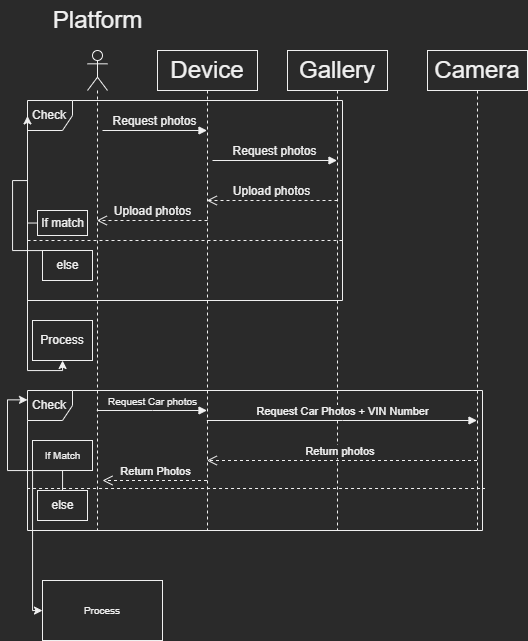
As the diagram shows, the application requires different variables for saving into the database and their relationship to each other. All the details should be saved in the Customer Table and as explained earlier, details entered in the first, second and third stages can be saved as id into that table, which is the Brand, mode and production year of the car as well as insurance subscription type, starting date and customer phone number. However, details of the user(car registration details and passport of user) can be saved as PK or CK when storing them in databases. Followed by that, we can save car photos as IDs and VIN as PK in our database. And finally, bank details can be saved as IDs as they have an expiration date.

The below diagram explains how the platform interacts with external systems



The diagram above explains how the platform should interact with mobile operators while verifying OTP security codes. In the early stages, when the user is entered their phone number platform should send an OTP message request from the mobile operator of that phone number. And as soon as the user receives and enters the sent OTP message platform is required to check if both matches. In case of failure, the user should be able to re-enter their phone number by going to the previous stage and/or re-enter their OTP code and it should continue until they both match. If checking is successful platform should move to the next stage.

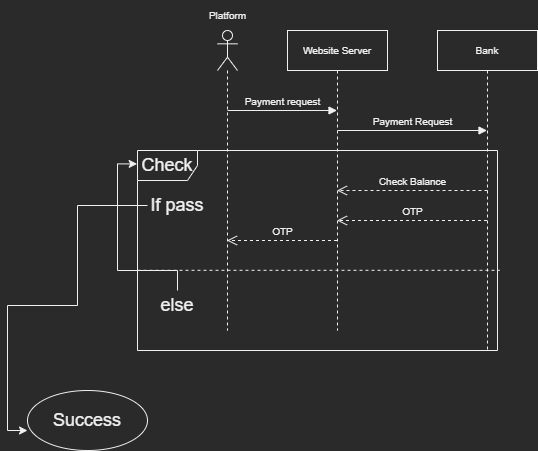
The below diagram explains how the platform interacts with a mobile device



As explained in earlier stages as well as in the diagram, the platform has to interact with mobile devices to confirm details of the car and its photos. For this, when the user reaches the stage when they are asked to upload photo details of their car and passport platform should access(ask permission) the gallery of the device and upload chosen photos. After uploading gets finished platform should check for the details of the car and the passport of the customer. If they fail to match, then request to re-upload or manually fill in the details. If they match the process to the next stage.

In the second part of the diagram, the customer will be required to take photos of their car. For this, our platform should access to devise camera and for each side(top, right, left, front, back) and VIN of the car request a separate photo. After getting all required photos, do check until the photos match, in case of match process to the next stage.

The below diagram explains how the platform interacts with payment systems



In the last stage, the user is required to make payment for their insurance policy. For this our platform should use simple payment logic, the user should fill in their bank account details and should process making a payment and our platform should send a request to the bank through our website server. Bank will check the user`s account balance and if they have a sufficient balance they should send a security code(OTP) which should be asked for by our platform when confirming payment. If they have an insufficient account balance bank should fail to complete payment, or if they fail in the OTP stage we should ask them to re-enter their OTP code.