ACME

Oliver Matthew Bowker (220263618)

June 9, 2023

Contents

List of Figures

List of Tables

1 Introduction

ACME (Aston Car Management Enterprises) is a car rental that offers rentals primarily to people in the Aston area. This includes a range of customers from working individuals to the large student population the city has. The company currently uses email and telephone for both customer requests and internal communications.

ACME has realised that the current system is outdated and not a good user experience for the customer. In addition to this the profits of the company have been on a slow decline as newer competition has risen, taking away some of ACMEs customer base.

The aim of this report is to come up with ways in which this problem can be tackled, coming up with a design that can be implemented. This report discusses the stakeholders, business goals and current situation ACME is in. From this use cases and requirements for the new system will be designed, followed by both activity and sequence diagrams to demonstrate how the system could work on a lower level.

The features on the system I will be focusing on in this report will be:

- Adding customer information
- Taking payment
- Handling the return of a vehicle
- Starting a new hire
- Adding a new car to the system

Currently ACME uses an outdated paper-based approach, communicating with customers and other staff members using a combination of telephone and email. The current system has numerous problems such as:

1. Due to the paper-based approach important details such as details about orders, cars and accounting can often go missing causing issues for both customers and internal staff.
2. Backups of data are difficult and time consuming due to the paper-based approach and their risk management for the same reason is near non-existent.
3. Customer experience is not optimal due to the updates coming through telephone calls or email only.

I have used plantuml [1] and LaTeX [2] to create this document. All source code can be found in this **github repository**.

2 Overview

2.1 Business Goals

In addition to this, the system does not contribute to ACMEs current business goals, which are:

1. **Increase profits/customers -** ACME has being seeing a decrease in profits and customers and wants to increase these.

2. **Improve documentation resilience and navigability -** Documents often go missing and are hard to find, by moving away from a paper based system ACME hopes to make this issue less of a problem.

3. **Cater to the student demographic -** ACME wants to take advantage of the large student population in the area, with both Aston and Birmingham university being near by. They are willing to offer better deals to students and offer alternative payment methods to target this demographic.

4. **Automate/speed up time intensive tasks -** Due to the current system, data input is slow and taking and editing rentals is also slow. With a new system ACME hopes this will speed things up, potentially requiring less staff, which will also help increase their profits.

2.2 Proposed changes

In order to fulfill the above business goals ACME has decided to upgrade it's outdated system with a new automated, digital system. The new system will no longer use paper based records, instead opting for a digital solution, which will use cloud computing. The solution should not remove any of the current functionality of the system, however can replace them for more modern alternatives. Some of the planned replacements include:

1. A new system where users can sign up and book rentals, without direct interaction from staff members. This includes a new payment system where customers will be able to pay through the new application immediately without going into the store. The details of both customers and order will be stored in the new database.

2. A new system for staff to add/edit/delete cars in the system, these details will also be stored in a database. This will by speeding up internal workings due to switching to a database instead of the old paper based system. It may also indirectly help with business goal 1 as staff will have more time to do more important things for the company. Another potential is that the number of staff needed could be reduced due to the optimisation, however this would need to be though about due to potential ethical issues.

3. As part of increasing profits and catering to the student demographic, ACME has made a bold plan to try an incorporate cryptocurrency payments into its new system. Cryptocurrency adoption in the UK has been growing in popularity, doubling since 2019 [3]. In addition to this a survey done in Germany showed that '18-to 27-year old survey respondents were three times more likely to own a digital currency' [4] and BanklessTimes wrote an article summarising a Finder report that showed 38% of all cryptocurrency holders in the UK were between the ages of 18-34 [5].

Age Group	Adoption Rate
18-34	38%
34-54	43.5%
55+	20%

Table 1: Table showing adoption rate based on age in the UK [5]

2.3 Software development model

The software development lifecycle (SDLC) usually consists of between 5-7 phases. This isn't a strict rule however with names changing and certain phases often not being included. The figure below shows the SDLC I will be following, **Appendix A** shows a full 7 stage model.

Figure 1: Figure showing the phases of SDLC. [6]

1. **Plan -** The planning phase involves, what the project is going to be, the requirements of the project, identifying stakeholders and any feasibility studies that need to be done.
2. **Design -** This phase includes graphical UI/UX designs, but also designs of the software. This can be done using UML.
3. **Develop -** Write the code that the plan.
4. **Test -** Write tests for the written code and perform manual testing.
5. **Deploy -** Deploy to cloud/on prem infrastructure.
6. **Review -** Review the new features/changes added and start the cycle again.

This report covers only covers the plan and some design aspects of the SDLC. Using an SDLC includes benefits such as helping understand requirements, identify risks [6] and speed up delivery of a project. Imagine skipping the plan and design phase above, jumping straight into development. The developers would not know what the system should look like and deliver a subpar final product

For this project I would recommend the use of the agile framework. This methodology is described as:

'The Agile methodology is a project management approach that involves breaking the project into phases and emphasizes continuous collaboration and improvement. Teams follow a cycle of planning, executing, and evaluating.' [9]

By this description our phase mappings would be, Planning = (Plan, Design), Executing = (Develop, Test and Deploy) and Evaluating = (Review). The agile manifesto describes the key concerns of agile.

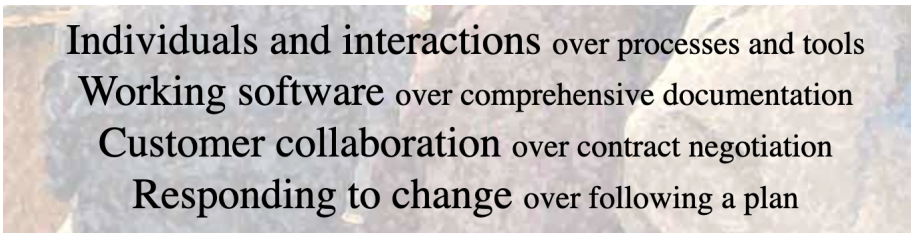

Figure 2: Agile methodology key principles [10]

Approaches such as waterfall can be *'largely dependent upon how much work is done upfront, especially research'* [11]. This research can take a lot of time, and with new students starting in 3 months, a large opportunity could be missed. This point is also linked to the agile manifestos 'Working software over comprehensive documentation' idea. In addition to this, ACME wants to be gain its old customer base back, agile encourages 'Customer collaboration'. Anthill reported that *'71% of customers feel frustrated when an experience is impersonal or company focussed.'* [12]. By interfacing with the customer/user of the application ACME could build good will with it's customer as well as make a better product for the target audience in the long run. In addition to this Ian Somerville states in his book Software Engineering, *'Once a system has been installed and is regularly used, new requirements inevitably emerge.'* [13]. Later I will produce requirements, however with such as large change the likelihood of getting it 100% right first time is small, so being able to adjust is a large benefit.

3 Stakeholders and Use Cases

Before writing requirements it's important to know the stakeholders of the company, as well as who will be affected by the changes that are planned.

3.1 Identifying stakeholders

To help identify stakeholders I have created a viewpoint diagram that can be seen in the below figure.

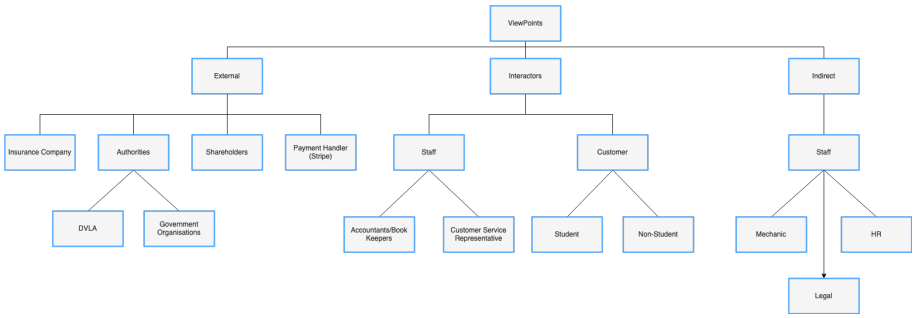

Figure 3: Viewpoint diagram to help identify stakeholders.

I have broken down the stakeholders into 3 categories, External, Interactors and Indirect, which describe their exposure to the new systems.

External - This group of people will not part of ACMEs staff, however do have an interest/role in the system/company. Insurance, and both stakeholders under 'Authorities' are needed for the system to function, insurance for the cars, and authorities to check the details of a car and renter are valid. The payment handler, I give the example of Stripe [15] will also be a part of the system but will not directly interact with it. Finally shareholders of a company always have an interest in change due to the expectation of profit.

Interactors - Interactors are people who will use the new system directly. I have separated these out into staff and customers. Customer is separated into two smaller classes, student and non-student. I did this to reflect one of the business goals of ACME, which is to *cater to the student demographic*, in addition to this stakeholder potentially getting benefits due to their situation. The other branch of interactors is staff. These are the people within the company who will interact with the new systems, like adding cars to the database (Customer Service Representative) and being able to easily query order/financial information (Finance/Accounting).

Indirect - These are employees of ACME who will not directly interact with the new interfaces. Their jobs will stay the same as prior. Despite this I though it a good idea to mention them as they could be the target for future upgrades to systems. In addition to this the due to the increased productivity of the

new system, a mechanic may be expected to do more work, which may not be feasible.

3.2 Use case diagrams

Taking the above stakeholders I have created a use case diagram to *'describe a system's requirements strictly from the outside looking in; they specify the value that the system delivers to users'* [14]. I will discuss each actor individually here but a full use case diagram can be seen in **Appendix B**, including crossovers between actors.

3.2.1 Customer Service Representative

The CSR refers to the member of staff that interfaces with the customer. They will be able to do all the same activities they did before, now using the updated system. I have decided to continue to allow telephone bookings and enquiries to be made. Hopefully the new system will be adopted by most people, however for people who aren't comfortable with it still have this option as *'21% of Britain's population lack basic digital skills'* [16].

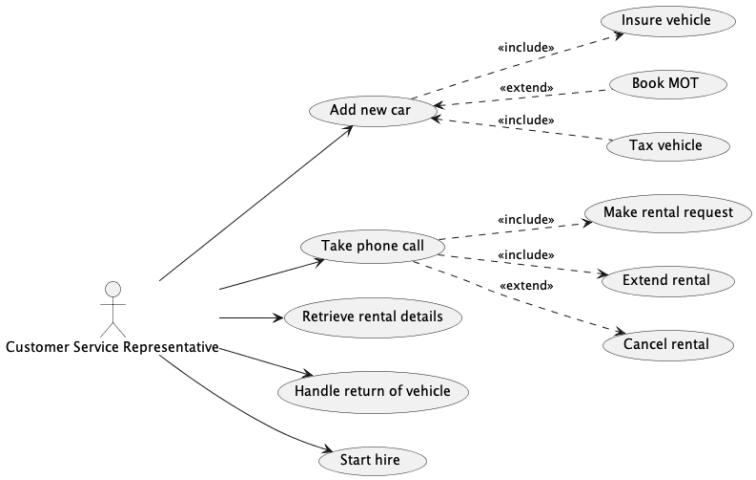

Figure 4: Use case diagram for customer service representative.

3.2.2 Customer

I merged students into a more general category of customer, although they may be able to do different actions in the future that is not the current scope of the changes. The customer can still extend, cancel and pay for rentals, as well as of course renting and returning a car. They will now also be able to login/register on the new system, meaning they will no longer have to give their

details multiple times. They will also be able to browse all available cars for rental on the new system.

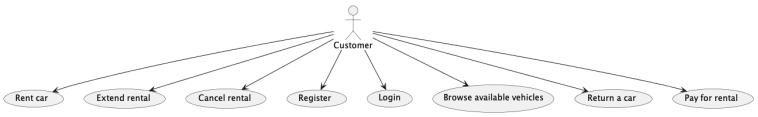

Figure 5: Use case diagram for customer.

3.2.3 Finance/Accounting

The finance/accounting department are responsible for generating tax information. This has been an issue in the past where receipts and order details have gone missing. The new system hopes to fix this by allowing all this data to be inputted into a database. Finance/accounting will be able generate earning reports and profits and loss statements from this.

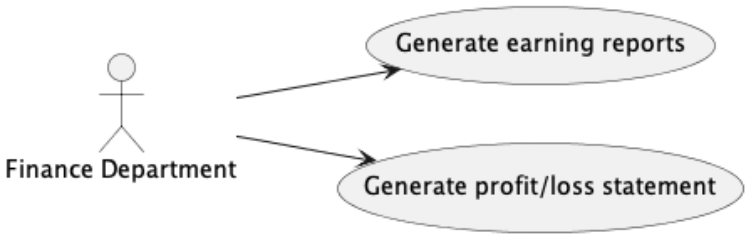

Figure 6: Use case diagram for finance/accounting department.

3.2.4 Insurance Company

The insurance company is an external stakeholder that ACME staff with have to interact with. They are responsible for insuring the cars and are linked to the Customer Service Representative in this task.

Figure 7: Use case diagram for insurance company.

3.2.5 Mechanic

The mechanic is in charge of making sure the cars are suitable to rent/drive. Mechanics can service a car, and repair cars when needed. Sometimes parts

will be ordered when repairing a car, so an optional action has been added to generate a receipt for these costs. This would currently be done paper based, however this could be rolled into the new system to make all expenses/profits stored in the same place, making it easier to do things such as tax returns and profit/loss accounts.

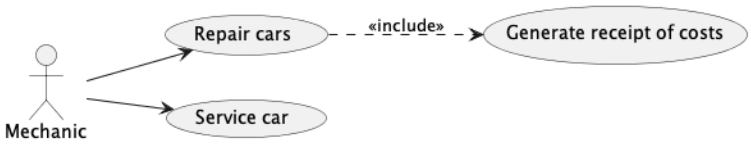

Figure 8: Use case diagram for mechanic.

3.2.6 Payment Service (Stripe)

The payment service is solely responsible for handling payments. They can take a payment for an order and they can validate a payment. When validating a payment there is a chance that it fails, therefore they must inform us if this happens.

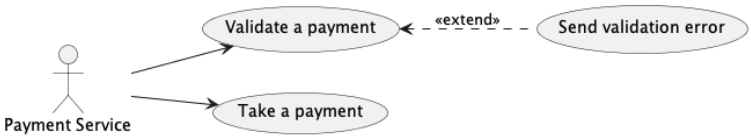

Figure 9: Use case diagram for payment service provider.

4 Feasibility of project

The new system is a big change for ACME, for that reason I will now briefly discuss the feasibility of the changes in different areas of the business.

Economic - As previously mentioned, ACME's sales have been declining which makes money something to heavily consider with the proposed changes. They're not in the 'red' so can afford some expenditure. The main issue currently, is that they don't have the technical staff to create this new system. Glassdoor puts the average software developer at £43,110 a year, £59,137 for a senior [17]. In my opinion ACME could probably hire a senior and two 'mid-level' engineers to do this project, adding up to around £140,000 a year. This is a lot of money, however the new system in the long run could result in less jobs needed in other areas, meaning total outgoings could stabilise. Another option would be to hire contractors, however these are more costly and would need to be called in anytime anything went wrong with system.

In addition to salaries there is also the cost of running the new system. I would recommend a cloud solution instead of a physical server. Not only would the upfront cost be less, you also only pay for what you use. These providers can also provide deals to companies who use their systems to save a bit more money on top.

Technical - Thanks to use of the agile framework we can take advantage of something called a spike:

> *'Spike A story or task aimed at answering a question or gathering information, rather than at producing shippable product.'* [18]

This would allow the new developers to pinpoint any threats that aren't seen in the planning and requirements gathering beforehand. This spike code can technical be used to help build the final MVP. The main system is a CRUD [18] application and should be very simple to build. It may be worth hiring a security analyst at the end of the project to make sure that the system is secure for both users and staff. The hardest part of the project would be to include cryptocurrency payments. Providers such as MetaMask [20] provide easy to integrate with wallets. However confirming the payment could be a little more challenging, however there are multiple libraries that can help with this. For this issue I would recommend that at least one developer has some experience with blockchain development, otherwise the learning curve could cost time and money due to training. When it comes to final implementation frameworks such as React Native [21], or Flutter [22] can be used for cross device/platform availability, thus speeding up the total development time.

Operational - The new changes may not be welcomed by some staff. This could be because they fear the change or feel like their job is at risk. Where possible ACME should tackle this by keeping an open line of communication with current staff and offering training to current staff for new roles.

As well as staff issues there are potential regulatory issues when it comes to holding personal user data and cryptocurrency as whole. with the latter,

it does seem like some progress is being made in a positive direction. The FCA (Financial Conduct Authority) released a report that seemed very positive towards crypto in the future [23]. In regards to holding user details such as driving license and banking information, there are system like Stripe [15] (for finance) that has a fantastic API [24] that means this data will never be stored by ACME. Driver license also does not need to be stored locally and can be checked and be a simple flag in the database.

5 Requirements

Now we've identified our stakeholders we can start thinking about the requirements for both user and system. In this section I will first identify user requirements, which can be described as *'user requirements are statements, in a natural language'* [13]. Then dig into each outlining the functional and non-functional requirements of that user requirement.

5.1 The system should allow users details to be stored

These requirements fulfill feature - **Adding customer information**.

Requirement	Type	Actors
A user should be able to input personal details (name, age, email)	Functional	Customer, CSR
The system should send a verification email to the user to stop fake accounts	Functional	Customer, CSR
A user should be able to input driving license information	Functional	Customer, CSR
The system should automatically validate drivers license information	Functional	Customer, DVLA/Authorities
A user should be able to login with previously saved details	Functional	Customer, CSR
A user should be able to access their details	Functional	Customer, CSR
A user should be able to recover their password/account	Functional	Customer
Sign up should be a one step process	Non-functional	Customer
No card details or MOT details should be stored	Non-functional	Customer, CSR, Legal

Table 2: Table for user requirement 1.

5.2 The system should allow car list to be edited

These Requirements fulfill feature - **Adding a new car to the system**.

Requirement	Type	Actors
Should allow the addition of a new car	Functional	CSR
Should allow the editing of a car	Functional	CSR
Should allow the removal of a car	Functional	CSR
Should only allow staff to perform operations	Non-functional	CSR

Table 3: Table for user requirement 2.

5.3 The system should accept payments for rentals

These Requirements fulfill feature - **Taking Payment**.

Requirement	Type	Actors
Should still allow in person payments	Functional	Customer, CSR
Should allow card online payments	Functional	Customer
Should allow crypto payments	Functional	Customer
All payments must be put into the new storage systems	Non-functional	Customer, CSR, Accounting/Finance

Table 4: Table for user requirement 3.

5.4 The system should allow the renting of vehicles

These Requirements fulfill feature - **Handling the return of a vehicle and Starting a new hire**. A user starting and returning a vehicle, system wise, is modelled by a member of staff changing the status of the vehicle on the system.

Requirement	Type	Actors
Users should be able to browse available vehicles	Functional	Customer, CSR
Vehicles can have their status updated available/unavailable/rented	Functional	Mechanic, CSR
Rental can be extended	Functional	Customer, CSR
The system should be able to calculate availability dates	Functional	Customer, CSR, Mechanic

Table 5: Table for user requirement 4.

5.5 Other requirements

Requirement	Type
Must use secure protocols (https/ssl)	Non-functional
The new system must be available at all times	Non-functional
Should offer 2fa for security of users	Non-Functional
Should encrypt/hash any sensitive data stored	Non-functional

Table 6: Table for other requirements.

6 Design

Following are, activity and sequence diagrams for some of the proposed changes.
I have colour coded them to show errors/failures. Diagram explanations/justifications can be found in **Appendix C**.

6.1 Adding a new user

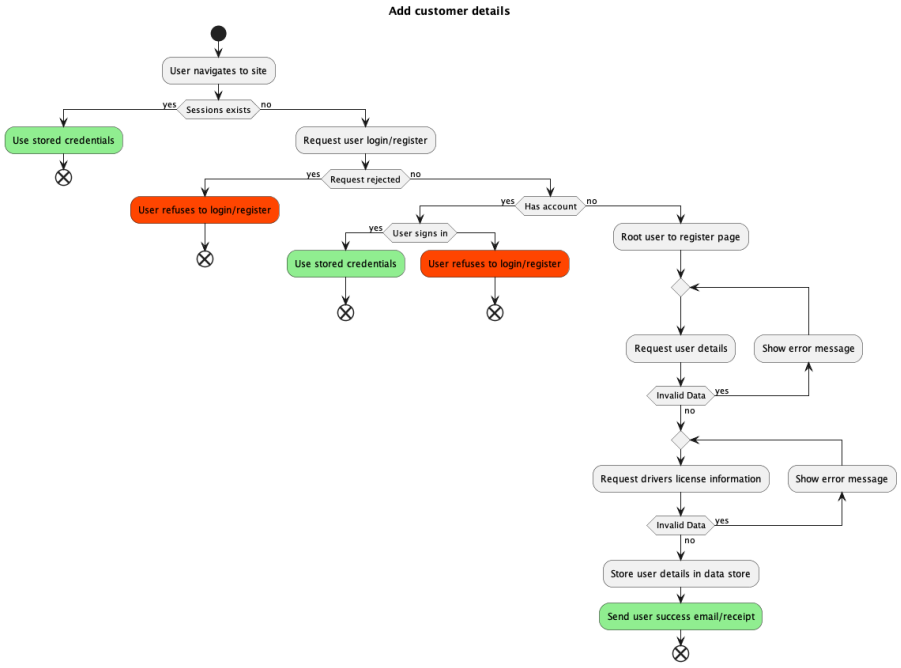

Figure 10: Activity diagram for adding a new user, this includes sign in/up.

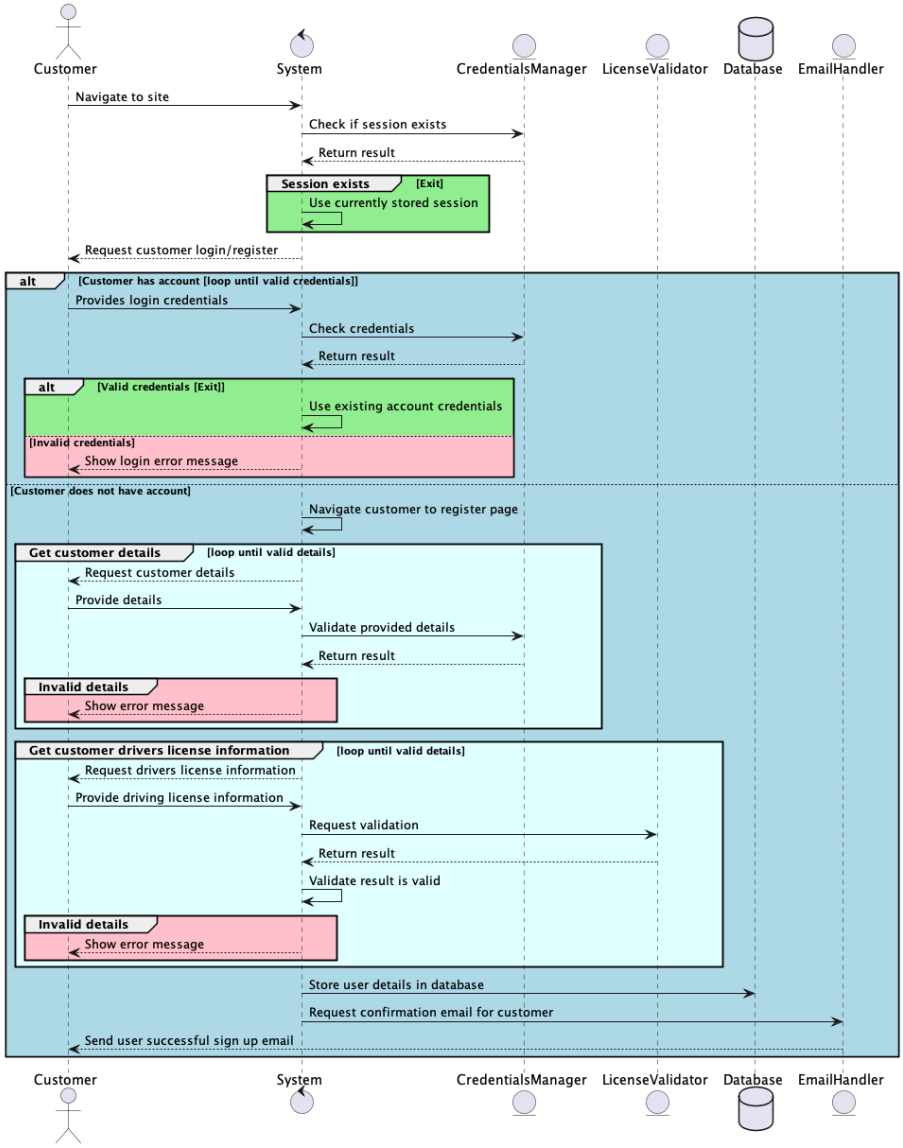

Figure 11: Sequence diagram for adding a new user, this includes sign in/up.

6.2 Taking a payment

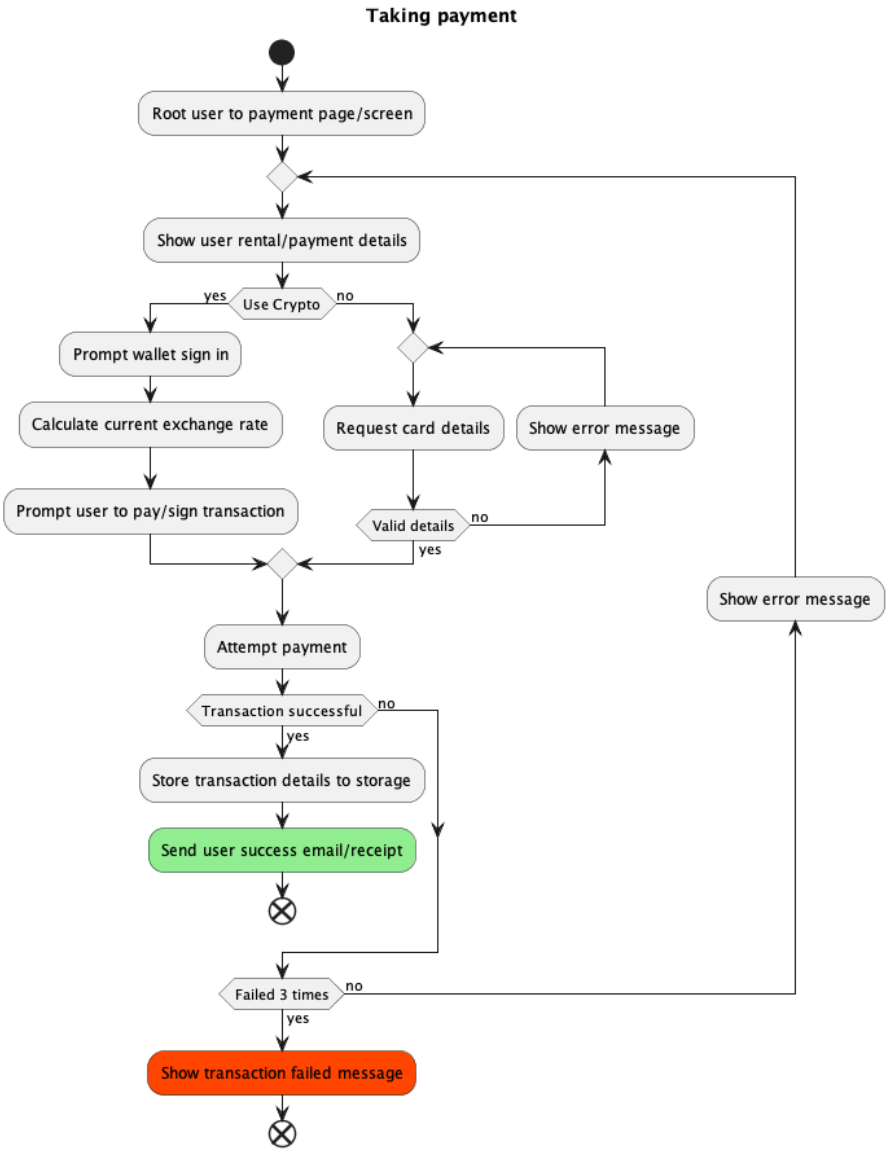

Figure 12: Activity diagram for taking a payment.

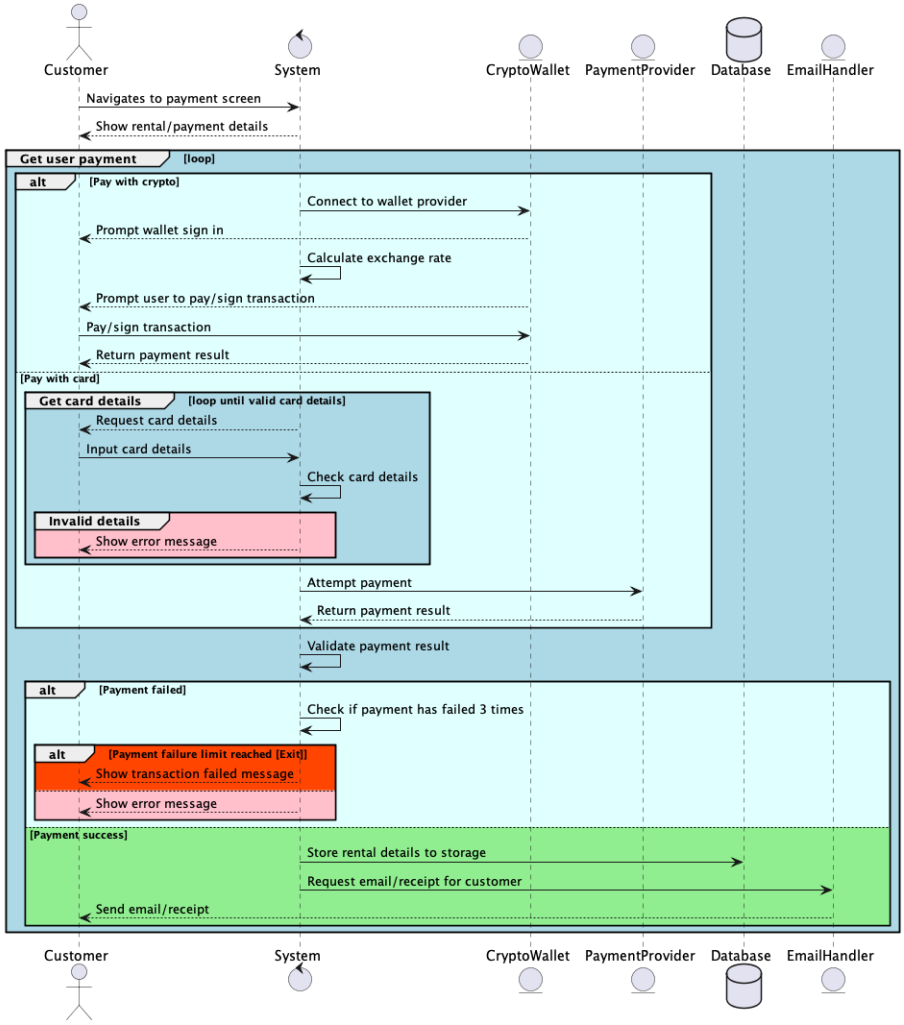

Figure 13: Sequence diagram for taking a payment.

6.3 Handling the return of a vehicle

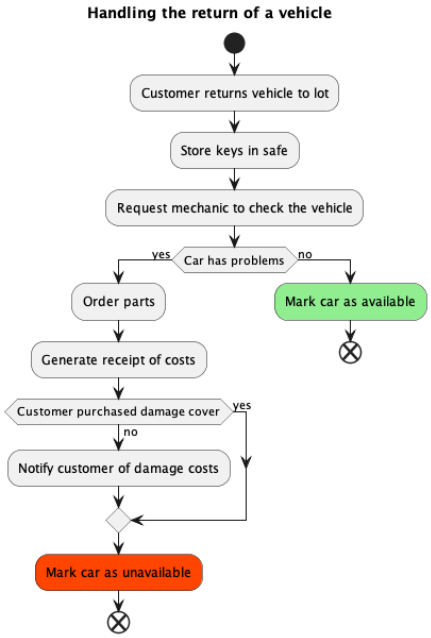

Figure 14: Activity diagram for handling the return of a vehicle.

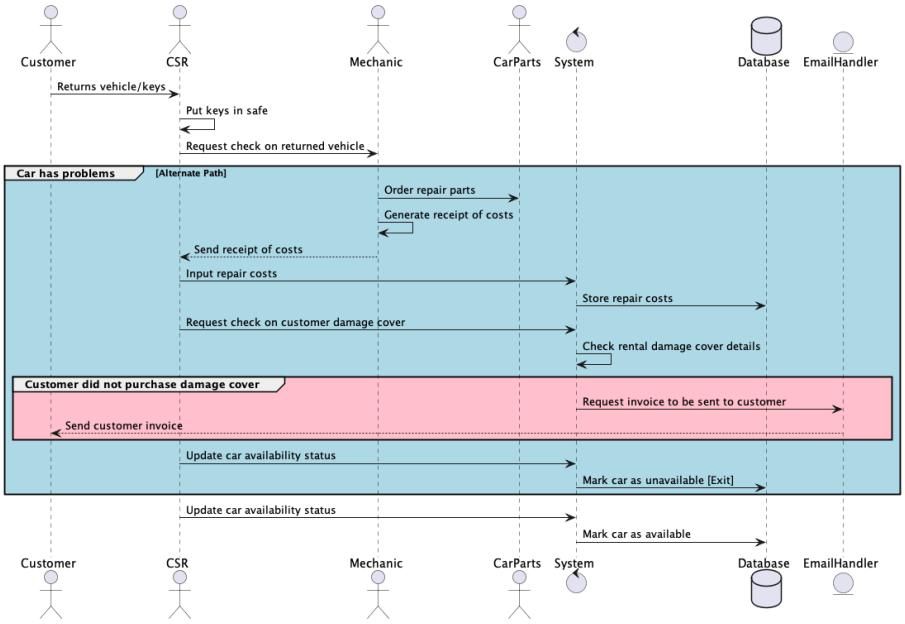

Figure 15: Sequence diagram for handling the return of a vehicle.

6.4 Starting a new hire

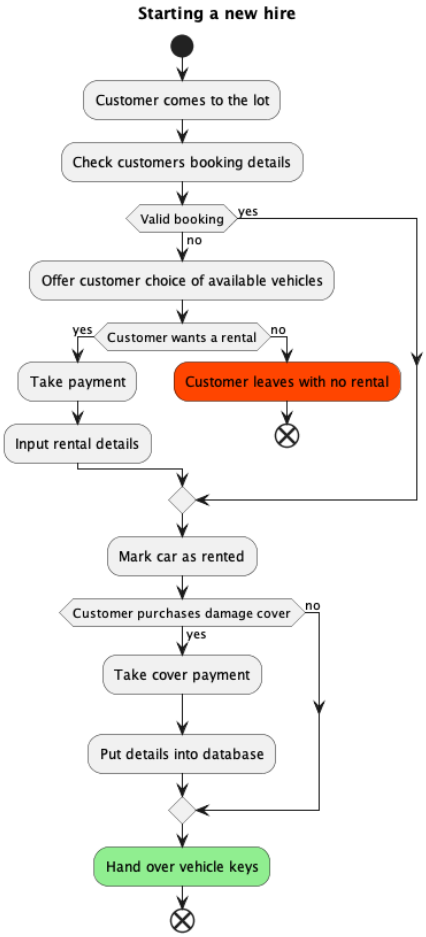

Figure 16: Activity diagram for starting a new hire.

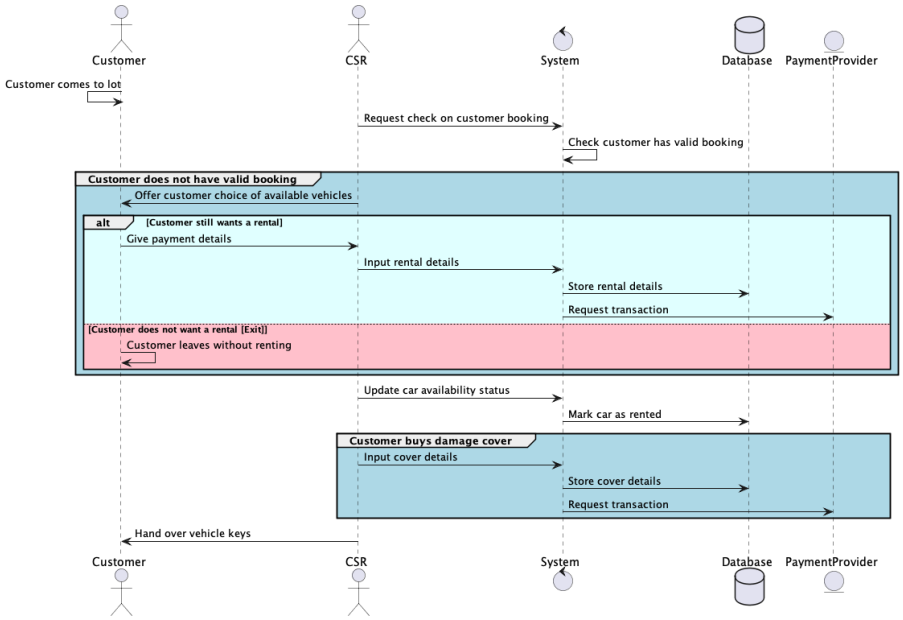

Figure 17: Sequence diagram for starting a new hire.

6.5 Adding a new car

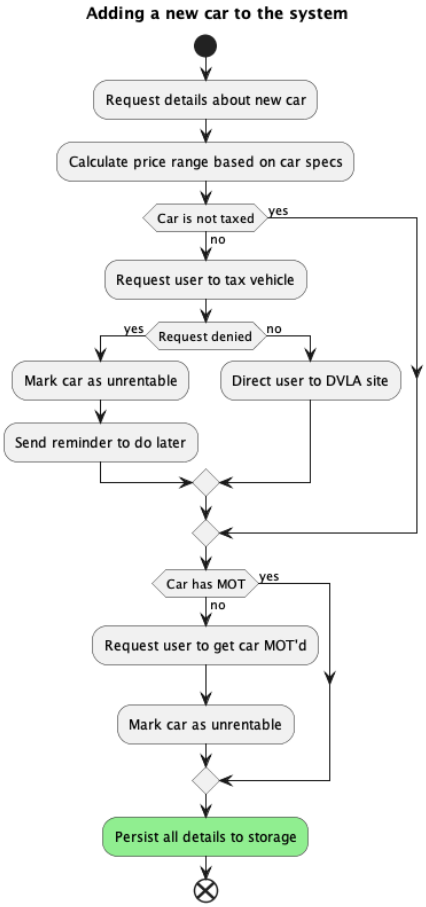

Figure 18: Activity diagram for adding a new car to the system.

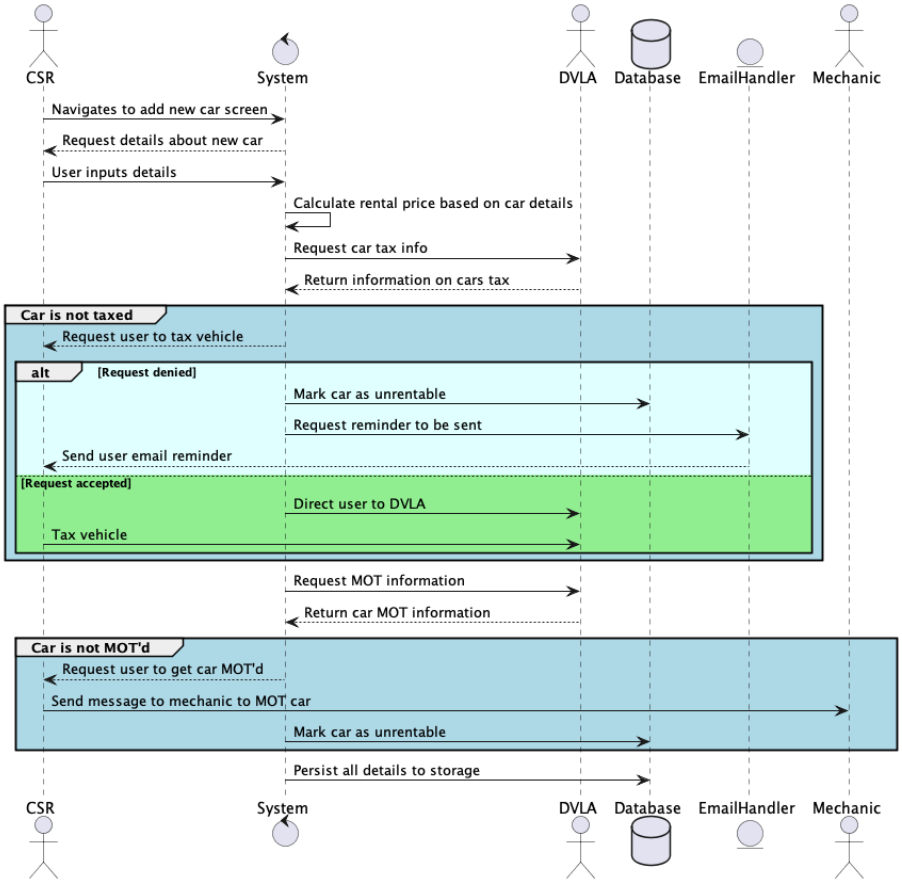

Figure 19: Sequence diagram for adding a new car to the system.

7 Conclusion

In conclusion, this report has covered the initial design phase for ACME to modernise their outdated systems. It's important to note that technological advancements are constantly happening. Technology such as self-driving cars, and policy changes to cryptocurrency and vehicles will play a massive role in ACME's future. However, I think that the newly suggested system will put them in a much better position in the future to tackle these challenges and thrive.

The suggested system in this document, is more than achievable and will help ACME hit it previously stated business goals, as well as offer a better user experience for both customer and staff members.

8 References

[1] PlantUML. (2023) *PlantUML in a nutshell* [online]. Available at `https://plantuml.com/` (accessed on 28th May 2023).

[2] LaTeX. (2023) *The LaTeX project* [online]. Available at `https://www.latex-project.org/` (accessed on 28th May 2023).

[3] Statista. (2023) *Crypto ownership by country 2019-2023 — Statista* [online]. Available at `https://www.statista.com/statistics/1202468/global-cryptocurrency-ownership/` (accessed on 28th May 2023).

[4] Best, R. (2021) *Cryptocurrency adoption among consumers - statistics & facts — Statista* [online]. Available at `https://www.statista.com/topics/7705/cryptocurrency-adoption-among-consumers/#topicOverview` (accessed on 28th May 2023).

[5] Nagari, S. (2023) *Cryptocurrency Adoption Statistics in the UK* [online]. Available at `https://www.banklesstimes.com/uk/buy-cryptocurrency/crypto-adoption/` (accessed on 28th May 2023).

[6] Laoyan, S. (2022) *What is Agile methodology? (A beginner's guide)* [online]. Available at `https://asana.com/resources/agile-methodology` (accessed on 29th May).

[7] Arkbauer. (2023) *Software development life-cycle (SDLC)* [online]. Available at `https://arkbauer.com/blog/software-development-life-cycle-sdlc/` (accessed on 29th May).

[8] Coursera. (2023) *What Is the Software Development Life Cycle? SDLC Explained* [online]. Available at `https://www.coursera.org/articles/software-development-life-cycle` (accessed on 29th May).

[9] Atlassian. (2023) *What is the Agile methodology?* [online]. Available at `https://www.atlassian.com/agile` (accessed on 29th May).

[10] Cunningham, W. (2001) *Manifesto for Agile Software Development* [online]. Available at `https://agilemanifesto.org/iso/en/manifesto.html` (accessed on 29th May).

[11] University of Minnesota. (2022) *Agile Methodology: Advantages and Disadvantages* [online]. Available at `https://ccaps.umn.edu/story/agile-methodology-advantages-and-disad` (accessed on 29th May).

[12] anthillsoftwareleeds. (2021) *Why Collaboration Is Key To Customer Satisfaction* [online]. Available at `https://anthill.co.uk/insights/why-collaboration-is-key-to-customer` (accessed on 29th May).

[13] Sommerville, I. (2016). Software Engineering. Tenth edition. USA:Pearson.

[14] Miles, R., Hamilton, K. (2006). Learning UML 2.0. USA:O'Reilly.

[15] Stripe, Inc. (2023) *Stripe — Payment Processing Platform for the Internet* [online]. Available at `https://stripe.com/gb` (accessed on 29th May).

[16] L, Georgieva. (2021) *Digital Inclusion and the Elderly: The Case of Online Banking* [online]. Available at `http://lrec-conf.org/workshops/lrec2018/W14/pdf/2_W14.pdf` (accessed on 29th May).

[17] Glassdoor. (2023) *Salary: Software Developer in United Kingdom 2023 — Glassdoor* [online]. Available at `https://www.glassdoor.co.uk/Salaries/software-developer-salary-SRCH_KO0,18.htm` (accessed on 8th June)

[18] Ashmore, s. (2014). Introduction to agile methods. USA:Pearson.

[19] Codecademy Team. (2023) *What is CRUD?* [online]. Available at `https://www.codecademy.com/article/what-is-crud` (accessed on 8th June)

[20] Consensys. (2023) *The crypto wallet for Defi, Web3 Dapps and NFTs — MetaMask* [online]. Available at `https://metamask.io/` (accessed on 8th June)

[21] Meta Platforms Inc. (2023) *React Native · Learn once, write anywhere* [online]. Available at `https://reactnative.dev/` (accessed on 8th June)

[22] Alphabet Inc. (2023) *Flutter - Build apps for any screen* [online]. Available at `https://flutter.dev/` (accessed on 8th June)

[23] Financial Conduct Authority. (2023) *Regulation of Digital Assets in the UK* [online]. Available at `https://www.fca.org.uk/news/speeches/regulation-digital-assets-uk` (accessed on 8th June)

[24] Stripe Inc. (2023) *Stripe API reference - curl* [online]. Available at `https://stripe.com/docs/api` (accessed on 8th June)

9 Appendix

9.1 Appendix A - An example of a 7 phased SDLC

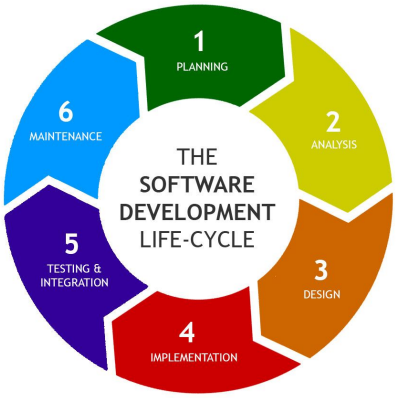

Figure 20: SDLC with 7 phases [5]

9.2 Appendix B - Use case diagram with all actors

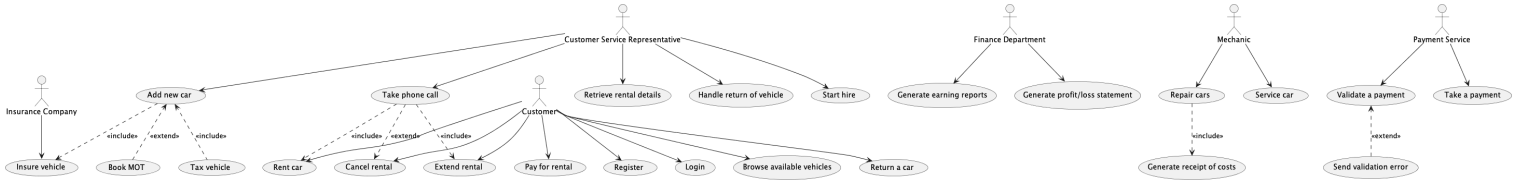

Figure 21: Full use case diagram, showing all actors.

9.3 Appendix C - Diagram descriptions

9.3.1 Adding customer information

As part of adding a information I have also modelled the login/register, I did this because without being signed in a customer would not be able to rent.

Activity - The diagram shows 3 happy paths and 2 sad paths. The sad paths are the user quitting the process, however it is worth noting that a user could quit the application during any stage of the process. The first happy path is when a sessions exists. This points to some kind of session/cookie management in the future. The second is credentials stored in the database of an existing user.

Sequence - In the sequence diagram a user can technically exit the login/register process at any stage, however I did not map this in the diagram as every stage would become bloated. Both EmailHandler and CredentialsManager would be module of the system, however I have separated them for more clarity. License validator would be an external API Call.

9.3.2 Taking payment

Activity - Taking a payment refers to online payments. If the user fails multiple times (3) then the system should stop the user from trying again, at least for a short while. This is to stop spamming and potential malicious activity. This is also includes the path of paying with crypto and cards to help reach some of Aston's business goals.

Sequence - The user navigates to the payment screen and is shown details about rental by th UI. Then the system consistently loops to get the user payment. This is broken when the payment succeeds or the systems payment failure limit is reached (shown in the dark red). This diagram introduces the CryptoWallet entity which the system will have to interact with if it wants to accept cryptocurrency payments.

9.3.3 Handling the return of a vehicle

Activity - If the user returns the car with no issues then this process is very simple and the car is marked as available for other customers to rent. If the car does have problems, and the customer doesn't have accidental damage cover, then they will be charged for the damages to the vehicle. I have marked this path as red, as it is not the ideal path, however technically the vehicle has been returned.

Sequence - The sequence diagrams shows how staff communicate in person as well as what the system does. The keys are stored in a safe on the car lot premises for security. The mechanic checks the vehicle for issues on return, order parts from a supplier and generates a receipt of costs if needed. The system in this case is mainly used to store data in the new database, and send a receipt to the customer at the end if there was any damages they need to pay for.

9.3.4 Starting new hire

Activity - When a starting a hire the customer must be at the car lot. I have modelled a situation where the customer does not have a valid rental and they can then be offered an in person choice of available vehicles. This will help capture people who just walk in ad hoc, and help accommodate a different sales path. I have also added a path where the customer can purchase damage cover. This would cover the individual if they were in an accident that was partly their doing, so standard insurance wouldn't cover it. However this is not required to start a hire.
Sequence - The sequence diagram follows the activity diagram closely. The system communicates with the database and payment provider to achieve the desired outcome. Note that I have not fully modelled customer payment transactions as that was already done in the **Taking Payment** section of this report. There is one fail case and that is if the customer does not have a valid rental and does not wish to rent any vehicles that are available.

9.3.5 Adding new car to the system

Activity - When adding a car a user has two optional stages, taxing the vehicle and MOT'ing the car. These are needed for the car to be rentable, however don't necessarily have to be done when adding the vehicle. However the diagram shows that if these criteria are not met the car is marked as unrentable until they have been fulfilled.
Sequence - The actors/entities in this diagram are similar to the first. The DVLA in this case would be some sort of API call, or potentially a physical call a member of staff has to make. For the sake of ease in this case I have modelled it as an API to check both tax and MOT status of the vehicle. The car is marked as unrentable if either of these cases are not fulfilled and that would be a simple flag in the database.

9.4 Appendix D - Word count

Word count **(Number Here)** - includes removal of captions, contents, list of figures/tables, Table 1, references and appendices.