Car Sharing on Demand

Ricardo Duarte

*BSc Hons in Contemporary Software Development* 2024/2025

# 

Computing Department, ATU Donegal, Port Road, Letterkenny, Co. Donegal, Ireland.

Car Sharing on Demand:

A car sharing platform for users and providers

Author: Ricardo Duarte

Supervised by: Pauric Dawson

June 2024

# Declaration

I hereby certify that the material, which l now submit for assessment on the programmes of study leading to the award of Bachelor of Science in Applied Computing, is entirely my own work and has not been taken from the work of others except to the extent that such work has been cited and acknowledged within the text of my own work. No portion of the work contained in this thesis has been submitted in support of an application for another degree or qualification to this or any other institution. I understand that it is my responsibility to ensure that I have adhered to ATU’s rules and regulations.

I hereby certify that the material on which I have relied on for the purpose of my assessment is not deemed as personal data under the GDPR Regulations. Personal data is any data from living people that can be identified. Any personal data used for the purpose of my assessment has been psudonymised and the data set and identifiers are not held by ATU. Alternatively, personal data has been anonymised in line with the Data Protection Commissioners Guidelines on Anonymisation.

I consent that my work will be held for the purposes of education assistance to future students and will be shared on the ATU Computing website (http://www.atucomputingdonegal.com/) and Research THEA website (https://research.thea.ie/). I understand that documents once uploaded onto the website can be viewed throughout the world and not just in Ireland. Consent can be withdrawn for the publishing of material online by emailing Thomas Dowling; Head of Department at jade.lyons@atu.ie to remove items from the ATU Computing website and by email emailing Denise McCaul; Systems Librarian at denise.mccaul@atu.ie to remove items from the Research THEA website. Material will continue to appear in printed formats once published and as websites are public medium, ATU cannot guarantee that the material has not been saved or downloaded.

Signature of Candidate: Date:

# Acknowledgements

I would like to thank…

# Abstract

Written here is no more than 250 words of summary of the problem, hypothesis and main conclusions. The abstract will entice people to read the rest of the document.

# Acronyms

|  |  |  |
| --- | --- | --- |
| Acronym | Definition | Page |
| JWT  V2X  V2P  JVM | JSON Web Token  Vehicle 2 Everything  Vehicle 2 Person  Java Virtual Machine | 1  1  2  2 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Table of Contents

[Declaration 4](#_Toc194234408)

[Acknowledgements 5](#_Toc194234409)

[Abstract 6](#_Toc194234410)

[Acronyms 7](#_Toc194234411)

[Table of Contents 8](#_Toc194234412)

[Table of Figures 9](#_Toc194234413)

[Table of Tables 10](#_Toc194234414)

[Table of Code Listings 10](#_Toc194234415)

[1. Introduction 1](#_Toc194234416)

[1.1. Purpose 1](#_Toc194234417)

[1.2. Background 1](#_Toc194234418)

[1.2.1. V2X 2](#_Toc194234419)

[1.2.2. JWT 2](#_Toc194234420)

[1.2.3. Kotlin Multiplatform 2](#_Toc194234421)

[1.3. Aims & Objectives 3](#_Toc194234422)

[1.3.1. Cross Platform User Application 3](#_Toc194234423)

[1.3.2. Server application 4](#_Toc194234424)

[1.3.3. Car application 4](#_Toc194234425)

[1.4. Report Outline 5](#_Toc194234426)

[2. Another Chapter 5](#_Toc194234427)

[2.1. Additional Chapters guide 5](#_Toc194234428)

[2.2 Images 6](#_Toc194234429)

[2.3 Referencing 7](#_Toc194234430)

[2.4 Tables 8](#_Toc194234431)

[2.5 Code and Formulae 8](#_Toc194234432)

[Appendices 11](#_Toc194234433)

[Appendix A: References i](#_Toc194234434)

[Appendix B: Code Listing ii](#_Toc194234435)

# Table of Figures

[Figure 1 Letterkenny Institute of Technology Logo 13](#_Toc429427538)

[Figure 2 Letterkenny Institute of Technology Logo 13](#_Toc429427539)

# Table of Tables

[Table 1. Table Formatting Guidelines 15](#_Toc429429027)

[Table 2 Second Sample Table 16](#_Toc429429028)

# Table of Code Listings

[Code Listing 1 MDBean Message Handling 16](#_Toc429428842)

# Introduction

Car sharing today presents today, according to (Turoń, 2023) “one of many solutions to help solve mobility in dense urban areas”, sine it aims to provide an easy and affordable solution for short-term trips and allow an individual to easily access a car sharing service over a smartphone in real time. The author of the article also highlights that in 2023 car sharing solutions existed in “over 59 countries offered by hundreds of different providers”, which could highlight the scalability of such solutions in today’s world and evidence how much this numbers could grow soon (JetBrains, 2024)given the trend.

This project aims to demonstrate a feasible, and secure platform for car sharing. The objective is to design and develop a solution, that is scalable, maintainable, secure and cross platform for that allows for an intuitive and easy to use experience for both providers and users.

## Purpose

In 2024 in Germany car sharing services “covered more than 1200 cities with over 300 providers”, according to (Wandering, 2024), which highlights the extension of the car sharing services on of the largest EU countries. He also highlighted that car sharing services can be “a valid option for people who don’t own a car or drive sporadically” and that base free “free floating” providers are a great alternative to “more typical car rentals when it comes to short trips” (Wandering, 2024).

Giving that, the purpose of this project is to introduce a platform into the car sharing eco system, that allows to provide a general, safe and easy to use solution into the eco system of the car sharing model.

## Background

The basis of the purposed solution is the implementation of a cross-platform solution in Kotlin, for a car sharing platform, based on a V2X design which favours the usage of JWT for password less authentication between user and car.

### V2X

The constant evolution of information networks, such as 5G, pushes for to design vehicle wireless technologies capable of communicating with everything from the cloud, people or even other vehicles, and, according to (Hasan *et al*., 2020), V2X as a technology aims to “improve trafﬁc and resource efﬁciency, incidents and road pollution”. However, this technologies is not free from challenges and the author mentions that the biggest concerns are regarding “security and privacy”. In this project the idea is to use V2X to connect vehicles to users both through an off-board entity (cloud server) and directly V2P using JWT as a technology for a safe and password less authentication between user and car.

### JWT

According to (Jones, *et al.*, 2015), the standard defines a JSON Web Token as a “URL-safe means of representing claims to be transferred between two parties”, a JWT is composed by an header containing the signing algorithm and type, a payload which represents the data object (in JSON format) being transmitted between entities, and the signature that validates the token itself generated via HMAC-SHA256 algorithm using a private key. According to (Rana, *et al.*, 2023) which also states that “JWT with the HMAC-SHA256 algorithm provides a high level of security in data exchange”, which in this project is used to mitigate the security challenges from the V2X nature of the project. The JWT can also be used for authorization and authentication which will allow for a user to initialize a safe and unique session desired vehicle.

### Kotlin Multiplatform

Kotlin multiplatform is a Kotlin capability that aims for “reducing time spent writing and maintaining the same code for different platforms”, according to (JetBrains, 2024), this allows for the same code to be shared for both web, mobile and desktop while attempting to preserve most of the benefits of native code. Kotlin provides multiple frameworks that will assist this project such as Compose, which is a cross-platform framework that allows developers to easily develop cross platform applications with shared code for both the models (in Kotlin) and UI (declarative). Kotlin also provides KTOR which is a framework for the web, which allows to both write client and server code, KTOR allows to compile for the JVM and natively which provides a great tear of flexibility for the solution being developed.

## Aims & Objectives

The main goal of this project is to develop a proof of concept of a car sharing system that focus on simplicity, safety and scalability as well as generalization. This goal can is divided into the following primary aims:

### Cross Platform User Application

One of the parts of the desired system is a cross-platform application that can ran in multiple platforms, where a user can fetch for available cars, get the car details such as availability, amount of fuel and location for example, and request a session for a specific car, lock and unlock the car and end the current session. The goals for this section are:

* Have a cross-platform application developed using Kotlin Multiplatform (with Compose for declarative UI), and KTOR as the framework for the https client code.
* The application fetches the server application for the available cars in each location and lists the cars available with their specific details (such as availability, charge or fuel, and exact location) in a list (or map) and cost.
* It allows the user to select a desired car and request a session for it.
* It allows the user to unlock or lock the car while a session is established.
* It allows the user to end the session on demand and generates a report.

### Server application

The second and most central component of the system is a server application that can fetch data from the car app for the car details, and feed the user application, accordingly, allow to get request from a user to authenticate with a specific car, generate a signed JWT and provide it to the user so it can authenticate with the car. This aim is divided into the following goals:

* Have a server-side application developed using Kotlin with KTOR as the framework for the https server code.
* The server will fetch for all the cars data in a fixed.
* The server will accept requests from the client application to feed the fetched data.
* The server can accept requests to create a session between a car and a user.
* The server can generate a signed JWT and send it to the user so it can authenticate with the car.
* The server can receive requests from the car and user about closure of sessions.

### Car application

The last part of the system is a car application that constantly sends status data back to the server, it can interact with the user app to initialize a session, receive commands for lock and unlock and end the session on user demand. The goals are:

* To have application developed using Kotlin which uses KTOR as the framework for the https client code.
* The application can send data to the server on demand or on a fixed interval updating its status.
* It can receive a JWT and validate it to initialize a session with a user.
* It can receive commands from a client within a session to unlock and lock.

## Report Outline

Provide a summary of what each of the following chapters will entail

Figure 1.2 Letterkenny Institute of Technology Logo

# Another Chapter

The chapters should all begin on a new page. Each chapter should contain an Introduction and a Conclusion for that chapter. Page numbers appear at the bottom right of each page. Page numbers appear from Chapter 1 and onwards. Appendices should be paginated using roman numerals (I, II, etc.).

* Bullets should be aligned with the text.
* Bulleted items should have one blank line above and below.
* If there are only 1 or 2 items a bulleted list is not required.

Any paragraph after the list, image or table should resume its normal position for the given header. When using any acronym such as Some Silly Acronym (SSA) it must be expanded on its first occurrence within the text. All acronyms should appear in an acronyms list preceding the main chapters.

## Additional Chapters guide

Additional chapters may include but not be limited to:

Chapter 2 – Design

* System requirements – Functional, Non-functional requirements/user stories, software and hardware requirements
* UML Diagrams, application structure, flowchart, storyboards

Chapter 3 – Implementation

* Environment set up; technologies used.
* Building the application piece by piece
* UI Design
* Connecting the front end to the backend

Chapter 4 – Testing

* Method of testing (White box, black box, unit tests, manual user tests, automated tests)
* Test Cases including test results.

Chapter 5 – Conclusion and Future work

* Results and outcomes
* Future work
* Issues found and addressed.

# 2.2 Images

In some sections of the project an image may be required. Any image utilised must be referred to within the main body of text. In Figure 1.1 the ATU logo can be seen with an appropriate caption.

Text

Description automatically generated

Figure 1.1 Letterkenny Institute of Technology Logo

To provide a caption for an image, table or equation the item should be selected. The Microsoft References ribbon should be selected. From there the Insert Caption button should be selected. The label should be set to the most appropriate one. In this example the Figure label was selected. New labels can be created as necessary. Remember not to simply copy and paste from above. Instead insert the image into the file and select it. Add the caption as described.

Where an image does not have a clearly defined border, one should be added. Care should be taken to ensure that all details of images are clearly visible both when in print and when in electronic format. Careful selection of colours should be considered for this purpose. Images as shown in Figure 1.2 should always be referred to from the main text.



## Referencing

The text of any given chapter may refer to an interesting idea presented in another book, paper, journal or whitepaper. On-line sources should not normally constitute more than 50% of your references. ALL of your references must be peer-reviewed or whitepapers. Further details on this will be given in the Research Workshop and may be obtained from your supervisor. This is supported by research (Bloggs, 2012) carried out how best to reference. The idea is succinctly expressed by Murphy:

“A reference in a thesis should be of the previously demonstrated Harvard Style.” (Murphy, 2011a)

Notice that the quote has indentations on both sides and is surrounded by quotes. If the quote abstracts only part of a sentence double dots should be placed before or after to show where there is missing text. Further, where additions for clarification are used in the text square brackets should be used. According to Murphy:

“A reference in a thesis should be of… Harvard Style.” (Murphy, 2011b)

While the example above is provided for demonstration purposes it is obviously not a good idea to provide the same quote twice so for the purposes of this example we will assume this quote was taken from a different book by the same author. Further the general use of a single or a small number of sources multiple times is referred to as ‘over-reliance on a source’ and is deemed plagiarism.

## Tables

The text of any chapter may include tabular data. In order to aid legibility some simple guidelines should be adhered to. Refer to Table 2.1.

Table 2.1 Table Formatting Guidelines

|  |  |
| --- | --- |
| **Format** | **Description** |
| Size | The table should be able to fit into one page and should not overrun. |
| Margins | The table should not extend past the normal margins of the page |
| Colour | Colour may be used but consideration should be given to both on screen display and printed display. |
| Design | Simple designs are best. At all times consider that the information in the table is more important than the ‘flashy’ design. |

The title for an image or code must appear directly underneath and on the same page. If this is not possible then move the item within the text to ensure that the caption remains with the item. The title for a table must appear directly above the table.

Refer to Table 2.2 for the second short table sample.

Table 2.2 Second Sample Table

|  |  |
| --- | --- |
| **Format** | **Description** |
| Size | The table should be able to fit into one page and should not overrun. |

## Code and Formulae

Where code requires listing within the text it should be treated as an image in that it is sectioned off with a border and has a caption directly underneath. Refer to Code Listing 2.1 below.

…

TextMessage msg = null;

try {

if (message instanceof TextMessage) {

msg = (TextMessage) message;

System.out.println("A Message received in TMDB: "

+ msg.getText());

}

else {

System.out.println("Message of wrong type: "

+ message.getClass().getName());

}

} catch (JMSException e) {

e.printStackTrace();

mdc.setRollbackOnly();

} catch (Throwable te) {

te.printStackTrace();

}

…

Code Listing 2.1 MDBean Message Handling

Notice that only minimal commenting is provided within the text. The code is shown in Cambria, 10 point. This reduces the overall text size and clearly distinguishes it from the main text.

If a single line of formula is required it can be referred to within the text as formula (Equation 2.1) for example with the formula example shown slightly indented and with the formula number to the far right.

Equation 2.1

A single line space above and below the formula (Tsiolkovsky, 2000) also aids legibility. Note also that the font size is increased by one point. A further item to note is that equations may also be referenced.

# Appendices

# Appendix A: References

Monowar, H., Sibin, M., Takayuki, S. & Hongsheng, L., 2020. Securing Vehicle-to-Everything (V2X) Communication Platforms. *IEEE Transactions on Intelligent Vehicles,* 5(4), p. 22.

Turoń, K., 2023. Factors Affecting Car-Sharing Services. *Smart Cities,* p. 17.

Wandering, S., 2024. *Carsharing 2024: Providers Compared.* [Online]   
Available at: https://www.navit.com/resources/carsharing-2024-providers-compared  
[Accessed 2025].

# Appendix B: Code Listing