

**Department of Electrical, Computer, and Software Engineering**

**Part IV Research Project**

Literature Review and  
Statement of Research Intent

Project Number: 99

Airbnb for Parking

Aiden Burgess


Sreeniketh Raghavan

Project Supervisor: Andrew Meads

Date: 19/04/2021

## **Declaration of Originality**

This report is my own unaided work and was not copied from nor written in collaboration with any other person.

Signature: 

Name: Aiden Burgess

**ABSTRACT:** The shared economy represents a new system for utilising existing assets. Platforms such as Uber and Airbnb utilise cars and apartments to generate income for their owners. However, there is currently no dominant platform for rental parking. This represents an opportunity in the shared economy space for a short-term rental parking solution. In this project we aim to create an MVP which investigates and explores the rental parking space by creating a mobile compatible application. The literature review focuses on pricing, mobile technologies, and existing applications. The implications of this existing research are discussed to produce desirable features and possible improvements above existing solutions. Finally, the areas of research are defined and a project timeline to produce the MVP is shown.

## **1. Introduction**

The idea for this project was formed during a long search for parking on the edge of the Auckland CBD. Even the side streets in the suburbs had no spaces, but the driveways were empty. The theory of this idea is that owners of the driveways would be willing to rent them for money. Despite the perceived demand, no parking sharing platform has become ubiquitous in New Zealand or globally.

This represents an opportunity in the shared economy space for a short-term rental parking solution. In this project we aim to create an MVP which investigates and explores the rental parking space by creating a mobile compatible application.

“The sharing economy differs from the traditional business model in its dominant reliance on the internet platform, its non-ownership of assets, its ability to access idle resources, its lower prices with more customized products/services and its non-conventional workforce as product/service providers” [1] [2].

In the literature review section, we will summarise the existing literature on aspects related to developing a rental parking mobile application. These include pricing, mobile application development frameworks, technologies to implement mobile applications, mobile map technology, usability of mobile applications, and existing applications.

We then discuss why our application will be mobile compatible, what features will be included in the application, and how the application will be different to existing applications. Finally, the statement of research intent will be presented, along with a planned timeline of the project.

## 2. Literature Review

### 2.1. Pricing

Ten studies conducted in eight cities between 1927 and 2011 found that an average of 34% of cars in congested downtown traffic were cruising for parking (Fig. 1.) [3] [4]. This represents a large pricing issue in the parking industry.

Year	City	Share of traffic cruising (%)
1927	Detroit, MI	19
1927	Detroit, MI	34
1960	New Haven, CT	17
1977	Freiburg, Germany	74
1985	Cambridge, MA	30
1993	New York, NY	8
2005	Los Angeles, CA	68
2007	New York, NY	28
2007	New York, NY	45
2011	Barcelona, Spain	18
<b>Average</b>		<b>34</b>

Fig. 1. Cruising information for cities [3]

The price of a parking space is not just the price of a parking but should also include waiting time (cruising time) and walk time to destination. “This waiting cost is the time drivers spend circling the block searching for an open space.” [4].

It is important to optimise pricing as there are negative consequences when parking is both under-priced and overpriced [4]. When parking is under-priced, many people are willing to park, but there are not enough spaces, so this can increase cruise times. It also means that the revenue is not being maximised. When parking is overpriced, “curb spaces remain empty, nearby stores lose potential customers, employees lose jobs, and governments lose tax revenue” [4].

In 2011, San Francisco switched from a static price per hour parking to prices which “vary by time of day and from block to block.” [4]. Philip Goodwin proposed two strategies based on price and quantity. One strategy was to change the price of the parking to reduce traffic. Another was to have a target traffic level and adjust prices to achieve that target [5].

Dynamic pricing is changing the pricing of a product or service over time, across consumers, or across products/services. Gibbs et al found that “dynamic pricing has been adopted by some of the leading sharing economy firms and has become an integral part of the sharing economy culture” [6].

Airbnb recently released an AI powered pricing system which considers unique features of the listings and offers a price for each date in the future [6]. However, it gives users flexibility to also choose a price higher or lower than recommended. This contrasts with Uber/Lyft where they have spot pricing which cannot be changed.

Disruptive products usually have lower pricing (e.g. Airbnb, Uber, Lyft) and other benefits, as they often lack some aspects of the traditional products. An example of this is hotels have staff whereas Airbnb listings do not.

There are two main types of owners: professional and non-professional [6]. Professional owners usually own more than one property and have greater returns and occupancy rates. Nonprofessional owners are less likely to change the price of their property to match demand.

Rob Stock identified three participants in the sharing economy: sharing opportunists, “those driven by economic necessity”, and finally commercial/professional participants. Sharing opportunists already own the asset and use the opportunity to make extra income [7] [8].

“Choudary and Parker use the Uber example to explain why platforms are so powerful: they eliminate gatekeepers, unlock new supply and demand and create community feedback loops. Uber, for example, performs a matching service that serves as a virtuous cycle. More demand is met by more opportunistic drivers, which increases geographic coverage, which leads to faster pickups, which encourages more customers to join the platform and more people to sign up as drivers. Driver downtime is lowered and so are prices, which leads to more scale.” [8] [9].

## **2.2. Mobile Application Development Frameworks**

Although many standard development frameworks for an application exist such as Waterfall, Scrum or Agile, Vithani et al. argues that there is a need for a new unique mobile application development lifecycle mode. They reason that because mobile application development involves complex functionality and services like telephony services, location-based services and different connectivity modes, a specific framework should be used to develop mobile applications [10].

They identify some key differences between mobile and desktop application development. Mobile devices have a shorter lifespan [10]. Mobile applications have more complex functionality involving telephone, camera, GPS. There is

a difference in physical interfaces. Desktops use keyboards, mice, touch screens, and other external devices while mobile devices only allow touch panel and keyboard.

Mobile devices also have smaller screens, so they need to use a layered UI and the functionality design also needs to be optimised for the smaller screen [10]. There is also a need to optimise usage of battery and memory in phones.

Therefore, they proposed a new framework called the MADLC Process. This process consists of the following phases: identification, design, development, prototyping, testing, deployment, and finally maintenance.

### 2.3. Technologies to Implement Mobile Applications

Bjorn-Hansen et al. investigated the features and performances between different cross-platform development frameworks (Fig. 2.) [11]. Among the cross-platform approaches, hybrid applications lagged in both launch time and time to render. The interpreted application had a considerably larger size compared to the other applications, and PWAs performed the best in both size and launch time (Fig. 3.).

Feature	Interpreted	PWA	Hybrid	Native
Installable	Yes	Yes <sup>a</sup>	Yes	Yes
Offline capable	Yes	Yes	Yes	Yes
Testable before installation	No	Yes	No	No
App marketplace availability	Yes	Yes <sup>b</sup>	Yes	Yes
Push notifications	Yes	Yes <sup>c</sup>	Yes	Yes
Cross-platform availability	Yes	Limited <sup>d</sup>	Yes	No
Hardware and Platform API access	Yes	Limited <sup>e</sup>	Yes <sup>f</sup>	Yes
Background synchronisation	Yes	Yes	Yes	Yes

Fig. 2. Feature-comparison of cross-platform approaches [11]

Traditionally, web, native, and mobile platform code have been non-interoperable, so there has been no reusability of code [11]. Cross platform development has been popular alternative to reduce resources and need for specialized mobile developers. Reduces dev effort and faster time to market.

The hybrid frameworks Ionic and PhoneGap structure components using HTML and CSS [11]. The interpreted framework React Native uses native interface components. Another interpreted framework Xamarin uses cross-

Measure	Hybrid	Interpreted	PWA
Size of installation	4.53MB	16.39MB	104KB
Launch time	860ms	246ms	230ms
Time from app-icon tap to toolbar render	9242.1ms	862ms	(a) 3152ms (b) 1319ms

Fig. 3. Measurement-comparison of cross-platform approaches [11]

compilation to compile C# into native binaries supported on each platform, so the apps are not dependant on interpreters or web views.

Progressive Web Apps (PWAs) were proposed by Google Web Fundamentals group to bridge gap between mobile and web by introducing features such as offline support, background synchronisation, and home-screen installation [11]. These new functionalities are supported by service workers. However, from their literature review they found there was little academic research on PWAs and for service workers. Currently PWAs are not able to access all hardware and platform features. E.g. calendar and contact list, however new APIs are becoming available to achieve these functionalities.

Charland and Leroux discuss the advantages and disadvantages of native and web code. Native code is usually compiled, “which is faster than interpreted languages such as JavaScript” [12]. This means that the application “paint[s] pixels directly on a screen through proprietary APIs and abstractions for common user-interface elements and controls” [12]. User interfaces are created in webviews utilising HTML and CSS. This leads to a lower performance overall than native code.

However, the benefit of web code is that it is compatible on multiple operating systems, which means that expensive developer time is not repeated, and the code is easier to maintain as there is only one code base. Furthermore, it is often faster and easier to write code for the web.

## 2.4. Mobile Map Technology

Maps can impact user cognitive load. Small displays and limited interaction capabilities often make mobile map-based systems difficult to design and frustrating to use [13]. To make a more usable map, reduce clutter and assist users in finding information they need. Also make relevant information easy to see via contrast to improve usability. Mobile maps need to deliver real time content and support real time interaction. GPS can deliver map content relevant to users personal and situational context. Map interaction: zooming, panning, point of interest selection

Church et al suggest that map and text-based interfaces should be used in different contexts [14]. For their research they implemented two local community Q/A mobile applications with the same functionality but with different user interfaces, one map-based and one text-based. They found that users “produced more queries through the map interface”, but “retrieved content more often” and “answered queries more often” through the text interface.

Although they discovered that participants 47% of participants preferred the map-based application and 41% preferred the text-based application (with 12% expressing no preference), the researchers concluded that a hybrid solution is ideal. “In this way, the speed of navigation that a text-based interface can offer can be coupled with the overview and sense of place that a map can provide” [14]. This was not just an idea that was discussed by the researchers, but some participants also suggested a hybrid model. An example quote from a user is: “I like to start with the map to get a overview and move to the text version afterwards” [14].

## 2.5. Usability of Mobile Applications

Harrison et al. suggested a new usability model for mobile applications, as two widely used frameworks: Nielsen and ISO were “derived from traditional desktop applications” [15]. Issues that the standard frameworks do not address are mobile context, connectivity, small screen size, different display resolution, limited processing capability and power, and data entry methods.

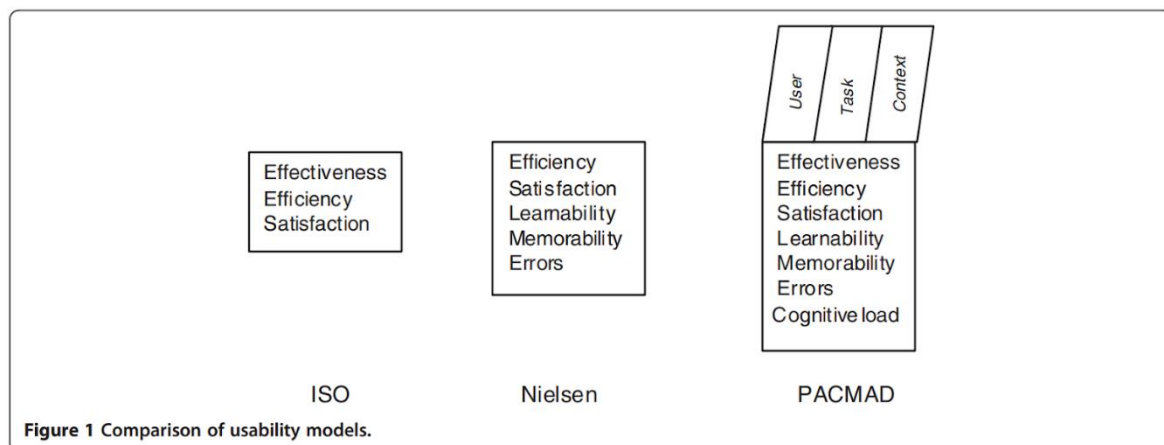


Fig.4. PACMAD usability model [15]

Their new model PACMAD (People At the Centre of Mobile Application Development) “aims to address some of the shortcomings of existing usability models when applied to mobile applications” [15]. The PACMAD model combines the Nielsen and ISO models into one framework to evaluate usability (Fig. 4.). This model has three factors of usability: user, task, context of use. The model evaluates usability with respect to seven attributes: effectiveness, efficiency, satisfaction, learnability, memorability, errors, and cognitive load.



## 2.6. Existing Applications

Existing solutions for rental parking in New Zealand are Parkable, Sharedspace, and Anyspace. Sharedspace focuses on event, office and parking spaces, however the parking is rented on a monthly basis. Anyspace is like Sharedspace, offering office, storage and parking spaces rented on a weekly basis. Parkable is focused on short term rental parking spaces (hourly or daily), with some support for long term options.

Therefore, Parkable is the most similar existing application to this project. It also focuses on a mobile interface, with the desktop website not containing functionality except for pointing to the mobile application. “Parkable is an excellent example of a home-grown app-based disruptive transaction platform” [8].

## 3. Discussions

From the literature review, it was found that there is a lack of research on the impacts and implications of a shared economy version of parking. Much research has been conducted on the successes of Uber and Airbnb, however a dominant parking application has not fully disrupted the parking industry yet, which explains the lack of discussion in this area.

A similar feedback loop to Choudary and Parker’s example of Uber can be theorised for a rental parking application. More demand is met by more opportunistic parking space owners, which increases geographic coverage of parking spaces, leading to closer parking spaces to destinations, encouraging more customers and owners to join the platform.

Min et al. examined the features of Uber and found “adding the unique features into the Uber mobile application, such as providing more accurate GPS location information on the map” and more specific search terms and refinement options “would help to enhance users’ perceived advantages of using the Uber mobile application” [Min2018]. The rental parking application should provide similar functionality around GPS location accuracy and search refinement.

Features that they found to increase observability were “information about the requested service such as estimated time of arrival, cost for the trip” and “transaction history” [16].

The MADLC development process proposed by Vithani et al. does not appear unique. The phases of the process are very similar to those of the Waterfall process. The justification of a new process for mobile application development did have some valid points but was lacking and not clear in others. For example, “complex functionality” is not unique to mobile applications, and many desktop applications are now also utilising GPS, camera, and voice sensors. Furthermore,

the preliminary results were weak, the article claims the process helped developers to plan and execute more effectively, but with no reasoning or evidence [10].

An important measure of usability is how much bandwidth and data an application uses [13]. To measure these attributes, the average mobile download/upload speeds in NZ should be used to benchmark the application.

Unlike other parking applications, we will experiment with a time slicing pricing system, where instead of being charged each hour, the user will be charged every 15 minutes instead. This adds value to the user by providing more perceived value as they are not charged as much for time they do not use. This is also a pricing advantage compared to traditional parking systems.

Another area where our application can differentiate from existing applications such as Parkable is by not relying solely on a map-based interface. Instead, this should be combined with text-based components. Church et al. concluded that hybrid-based applications are ideal as they combine advantages from both systems in terms of ease of use and aesthetics [14].

#### **4. Statement of Research Intent**

The intent of this research project is to produce a minimum viable product (MVP) of a rental parking system for the New Zealand Market. This MVP should support a mobile user interface to allow users to rent a parking space in New Zealand.

By exploring the implementation of such a product, we hope to explore whether problems in the parking industry can be solved using a shared economy proposition. These problems are sparse parking space, high cruising time, excess parking at residential properties due to parking mandate by council and decreasing parking space per capita in the Auckland CBD [17].

There are also technical areas of research which will be analysed. The mobile application framework (interpreted, hybrid, native, or PWA) will be determined as well as the larger tech stack to fit user requirements that are discovered. Target demographics will also be determined to be used for generating specific usability requirements.

The MVP will be produced as an example to demonstrate these learned concepts. To determine the usability and usefulness of the application, usability heuristics will be analysed, and it will be evaluated via user testing with target demographics.

There can also be an investigation of the benefits and disadvantages of time slice pricing, where the granularity for parking time is reduced from one hour. Measurements of demand and user preference can be used to determine the optimum time slice.

#### 4.1. Project Timeline

2021																															
Mar				Apr				May				June				July				Aug				Sept				Oct			
W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Market Research / Stakeholder Analysis																															
								Requirements																							
												Design																			
																Implementation															
																				User Testing											
																								Evaluation							

Overall, the project is expected to take 25 weeks in total, ending in the second week of September. The outline of the project timeline (Fig. X) is a rough idea of the steps needed and an estimate of each stage. It should be noted that each stage may overlap in the actual timeline. The stages that were identified as necessary for this project were: market research/stakeholder analysis, requirements, design, implementation, user testing and evaluation.

#### 5. Conclusions

In conclusion, there appears to be much research on shared economy platforms such as Uber and Airbnb, but there is a lack of research with respect to rental parking. The main reason behind this could be that there is currently no large application which has disrupted the industry.

Therefore, there is an opportunity to introduce a new parking application utilising existing usability research, mobile map technology, and a new shorter time slice feature for pricing. This project aims to develop an MVP of such an application to investigate the impact on parking problems such as cruising time, prices, and per capita parking space. Furthermore, technical aspects such as which mobile framework should be used. Usability and usefulness will be measured via user testing and PACMAD analysis.

#### Acknowledgements

The author would like to thank their project supervisor Andrew Meads for providing support and advice while producing this report. They would also like to thank their project partner Sreeniketh Raghavan for providing ideas, research and support while working together on this project.

## 6. References

- [1] R. Belk, "You are what you can access: Sharing and collaborative consumption online," *Journal of Business Research*, vol. 67, pp. 1595-1600, 2014.
- [2] Z. Mao and J. Lyu, "Why travelers use Airbnb again?," *International Journal of Contemporary Hospitality Management*, vol. 29, p. 2464-2482, 9 2017.
- [3] D. C. Shoup, *The high cost of free parking*, Planners Press, American Planning Association, 2011.
- [4] G. Pierce and D. Shoup, "Getting the Prices Right," *Journal of the American Planning Association*, vol. 79, p. 67-81, 1 2013.
- [5] P. B. Goodwin, "Traffic reduction," in *Handbook of transport systems and traffic control*, Emerald Group Publishing Limited, 2001.
- [6] C. Gibbs, D. Guttentag, U. Gretzel, L. Yao and J. Morton, "Use of dynamic pricing strategies by Airbnb hosts," *International Journal of Contemporary Hospitality Management*, vol. 30, p. 2-20, 1 2018.
- [7] R. Stock, "The share club," *Dominion Post*, 11 2017.
- [8] K. Jenkins, "Platforms in Aotearoa: our fast-growing sharing economy," *Policy Quarterly*, vol. 14, 3 2018.
- [9] S. P. Choudary and G. Parke, "How to build a successful platform business," *INSEAD Entrepreneurship blog*, 6 2016.
- [10] T. Vithani and A. Kumar, "Modeling the mobile application development lifecycle," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2014.
- [11] A. Biørn-Hansen, T. A. Majchrzak and T.-M. Grønli, "Progressive Web Apps: The Possible Web-native Unifier for Mobile Development," in *Proceedings of the 13th International Conference on Web Information Systems and Technologies*, 2017.
- [12] A. Charland and B. Leroux, "Mobile application development," *Communications of the ACM*, vol. 54, p. 49-53, 5 2011.
- [13] V. Setlur, C. Kuo and P. Mikelsons, "Towards designing better map interfaces for the mobile," in *Proceedings of the 1st International Conference and Exhibition on Computing for Geospatial Research & Application - COM.Geo*, 2010.
- [14] K. Church, J. Neumann, M. Cherubini and N. Oliver, "The 'Map Trap'?", in *Proceedings of the 19th international conference on World wide web - WWW*, 2010.
- [15] R. Harrison, D. Flood and D. Duce, "Usability of mobile applications: literature review and rationale for a new usability model," *Journal of Interaction Science*, vol. 1, p. 1, 2013.
- [16] S. Min, K. K. F. So and M. Jeong, "Consumer adoption of the Uber mobile application: Insights from diffusion of innovation theory and technology acceptance model," *Journal of Travel & Tourism Marketing*, vol. 36, p. 770-783, 9 2018.
- [17] MRCagney, "The Economic Impacts of Parking Requirements in Auckland," *Auckland Council*, 8 2013.