2021 IEEE International Conference on Computers, Software, and Applications Conference (COMPSAC2021), July 12-16 2021, online

A human-centric approach to building a smarter and better parking application

Chenlin Li, Yuting Yu, Jeremy Leckning, Weicheng Xing, Chun Long Fong
John Grundy, Devi Karolita, Jennifer McIntosh and Humphrey O. Obie
Department of Software Systems, Faculty of IT, Monash University, Australia
chenlinli.andy@gmail.com,pt4300@hotmail.com,jleckning5@gmail.com,wxin9@student.monash.edu,fchunlong@gmail.com
{john.grundy,devi.karolita,jenny.mcintosh,humphrey.obie}@monash.edu

***Abstract*—Finding a parking space can be very stressful and time consuming. A variety of different vehicle parking applications have been developed but many fail to support diverse end-users. We captured diverse *human-centric issues* from user reviews and literature, and then created personas that encompass a wide representative range of parking app user groups. Using these personas, user stories were created, categorized and parking app tasks prioritized. We used these to develop a prototype new "smart parking app". A cognitive walk-through was employed using each of the personas and user stories to evaluate the app. With more human-centric factors taken into account in the design and development of the app, we found that majority of the human-centric frustrations identified were resolved, when compared with a commonly used parking app.**

I. Introduction

With rapid population growth and urbanization, major cities around the world are facing multiple sustainability challenges [1], [2]. To help address some of these issues, many cities are looking to incorporate new technology to improve the quality of life of their inhabitants – these cities are known as "smart cities". *Sustainable mobility* is one of the major goals of most smart cities [3], as issues such as high traffic congestion, high air pollution caused by vehicles, and the lack of available parking spaces are common in modern cities and lead to a frustrating experience for drivers. One way to improve sustainable mobility is to improve the management of parking spaces, making the development of smart parking systems a high priority when developing smart cities.

Smart parking is not a new concept and has already been adopted by many cities. However, while there are many solutions on the market that could be considered smart parking systems, most lack full consideration of *human-centric issues* during the development stage, manifesting in apps that are difficult for many people to use. Human-centric issues are important to include when developing smart parking apps, as different end-users have different, complex needs [2]. For example, end-users of a mobile application will typically vary in age, cultural background, language, have different mental and physical challenges, have different cognitive styles, and require different kinds of support to perform the same task. A diverse range of end-users means that building a smart application suitable for all users is a complex task. For example, a person with visual limitations such as colour blindness or short sightedness, might well have trouble using the application if the icons and font sizes are too small or the colours are not easily differentiable. An elderly person might have trouble navigating a complex user interface and understanding complex instructions including unknown jargon. Additionally, a person with mobility challenges will benefit from being able to find a mobility car park close to their target destination. These require a more human-centric development approach when developing smart city, including parking, apps.

In this paper, we focus on capturing a range of human-centric frustrations with current parking applications. We then provide an improved smart parking solution by designing and building a prototype for a "better smart parking application". To capture the smart parking app's diverse end-user needs, we developed a set of representative personas, informed by negative reviews on current parking apps and relevant literature. We used these personas to create a number of user stories that clearly identify and gather the common and different requirements for the smart parking application for these users. Using these requirements, we designed a prototype on Figma and implemented the design prototype as a mobile application using React Native. We then evaluated the prototype by having our test subjects go through a cognitive walk-through of the app based on the personas. The main contributions of this work include:

- we identify challenges with current smart city parking apps in failing to meet diverse end-user human-centric needs;
- we develop a set of personas to represent diverse parking app end-users;
- we develop a set of user stories to better support these diverse end-user smart parking app needs;
- we prototype such an app using React Native and perform a cognitive walk-through evaluation of it; and
- we discuss lessons learned from to improve human-centric smart city app development.

II. Motivation

There are a number of industry and research smart parking applications – many focus on using Internet of Things (IoT) integration and system security to improve the management of parking resources and integrate the parking application into the smart city ecosystems [3], [4], [5]. However, the human-centric aspects of the end-users of these application do not appear to have received much attention.We found a number of parking applications that are currently widely used. We found by studying the negative user feedback from this and several other apps that they have a number of common human-centric issues. To confirm and understand these issues we tested this and other exemplar parking apps under a typical daily car parking scenario. We discovered that the search system can only be searched by car park name instead of street address or by current location. Search function logic is also not intuitive and can cause confusion to new or non-technically-oriented users. There is no facility to change icon or font size or shape; some colour choices are not suitable for colour blind users; only English text is supported; and the sequence of parking tasks is rigidly fixed, unsuitable or overly difficult for some end-users' parking needs. The app does not accurately detect park location and often requires entering signposted location code or scanning a QR code. Payment assumes finding a free park first, and pre-booking a park is poorly supported. The app requires users to add their payment details before they can use the application. Such a requirement may not be acceptable to all users who might not want to share their private banking details to a third-party application or be identified directly by their payment details. They might instead prefer to use cash over electronic payment methods, or PayPal or some other anonymous parking payment method. Moreover, the app does not provide customization options for different user groups. Elderly users, colour blind users, and users with a disability might encounter difficulties when using the application due to the lack of UI customization support. The app only supports English language users and does not provide any localization to other languages, which may not fulfill non-native English users' needs, and given most cities are multicultural, parking and other smart city apps should provide better user experience.

III. Our Approach

Taking into account drivers' frustrations, we aimed to improve the accessibility and usability of current parking applications by designing, prototyping and evaluating a better "smart parking" mobile application. Figure 1 shows an outline of this approach used which has four key parts:

Requirements Engineering: creating personas to simulate various diverse end-users, and then creating a set of user stories to meet each persona's different human-centric needs and to resolve each persona's different parking frustrations.

Design: designing app components to meet each user story, cognisant of the persona(s) using each one. Using the different personas and user stories, a full app prototype was designed on Figma. We discussed pros and cons about each design choice before altering and improving them.

Prototyping: based on the Figma design, a mobile application was developed using React Native.

Evaluation: we evaluated our prototype by doing a cognitive walk-through using the personas created. We compared its results to a cognitive walk-through with the same personas and user stories of the example commercial parking app.

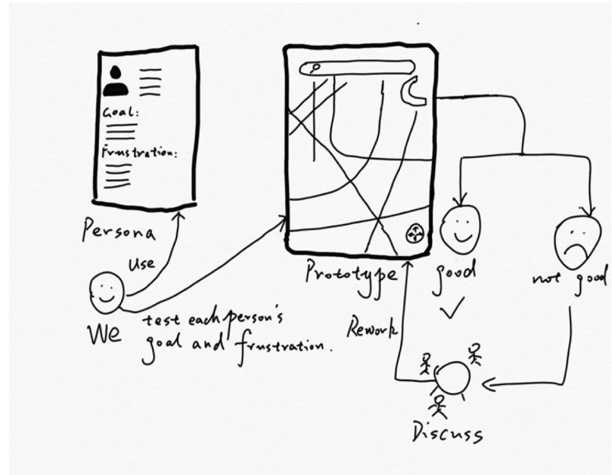


Fig. 1. Outline of our human-centric smart city app development approach

IV. Requirements and Design

A. Requirements Engineering

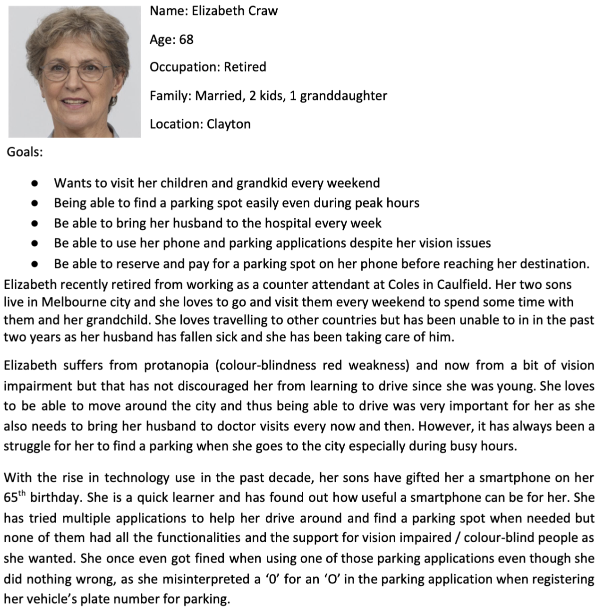
In order to simulate a larger group of end-users, we created five personas. These personas were created after conducting research on some parking applications and discovering the key human-centric issues in each of them. We did this by (i) looking carefully at the user reviews for several parking apps and especially negative reviews that highlighted challenges different users were having with the app; (ii) reviewing existing literature on smart parking app development and relevant requirements they identified [4], [5].

Figure 2 shows one of the personas that was created: Elizabeth Craw, who was used to represent the aging female demographic. We identified a number of parking goals and frustrations "Elizabeth" needed addressed – along with the other persona goals and frustrations – in our prototype.These were used to derive user stories which encapsulated each of their needs. We used five personas and each of them have around five user stories that we created.

User stories were developed for each persona using these goals and frustrations. Some of the user stories are the same or substantially similar for different personas. Some are unique to a single persona. Some examples of user stories created under the "user experience" category include:

"As a person who likes to work methodically, I want the app to have a clear and easy to learn user flow during one transaction so I can understand what I am doing in each step", *"As a food deliverer, I want to be able to easily switch vehicles I am parking so that I can use multiple vehicles to do my delivery"*, and *"As an employee, I want to be able to find a parking space quickly to save valuable time"*.

The user stories we created for the smart parking app were categorized into the following groups: setup, integration, user interface, human-centric issues, user experience, payment, and security-related stories. Some of the user stories for multiple personas had such similar functionalities that we resolved them into a single story instead.



Name: Elizabeth Craw

Age: 68

Occupation: Retired

Family: Married, 2 kids, 1 granddaughter

Location: Clayton

Goals:

- Wants to visit her children and grandkid every weekend
- Being able to find a parking spot easily even during peak hours
- Be able to bring her husband to the hospital every week
- Be able to use her phone and parking applications despite her vision issues
- Be able to reserve and pay for a parking spot on her phone before reaching her destination.

Elizabeth recently retired from working as a counter attendant at Coles in Caulfield. Her two sons live in Melbourne city and she loves to go and visit them every weekend to spend some time with them and her grandchild. She loves travelling to other countries but has been unable to in in the past two years as her husband has fallen sick and she has been taking care of him.

Elizabeth suffers from protanopia (colour-blindness red weakness) and now from a bit of vision impairment but that has not discouraged her from learning to drive since she was young. She loves to be able to move around the city and thus being able to drive was very important for her as she also needs to bring her husband to doctor visits every now and then. However, it has always been a struggle for her to find a parking when she goes to the city especially during busy hours.

With the rise in technology use in the past decade, her sons have gifted her a smartphone on her 65th birthday. She is a quick learner and has found out how useful a smartphone can be for her. She has tried multiple applications to help her drive around and find a parking spot when needed but none of them had all the functionalities and the support for vision impaired / colour-blind people as she wanted. She once even got fined when using one of those parking applications even though she did nothing wrong, as she misinterpreted a '0' for an 'O' in the parking application when registering her vehicle's plate number for parking.

Fig. 2. Example persona (one of five developed)

B. App Interface and Workflow Design

The main design principles considered for our new smart parking app prototype are summarised below.

Fitts' law [6]: As a motivation of Fitts' law – big targets at a close range are acquired faster than small targets at a distance. We placed most of the clickable buttons in the bottom part of the screen so that the user can click them easily. The button sizes are also dynamically changed depending on the screen size. The button size is between 16mm and 19mm square. Our rationale for this is based on a study that shows most older people have better performance results when using an application if the buttons are between 16.51mm and 19.15mm square [7]. Other research into designing interfaces for aging users also advises simplified app layout, simplified textual labels, and careful attention to workflow in the app logic [8], [9], [10]. Figure 3 shows examples of these designs.

Furthermore, we used Norman's seven principles [11], [12], and Shneiderman's eight golden rules [13]. Our prototype prevents users from making mistakes. A better way than a good error message is a careful design that prevents the problem from occurring in the first place. For example, the price of a parking space is shown on multiple pages to keep informing the user of the correct price. This also minimises the user's memory load as the user does not have to remember information from one part of the screen to another [7]. This is important because humans only have limited capacity for information processing due to limits of our short-term memory [6]. As a result, users are able to make the correct decision on whether to pay for a parking space or not. The user can easily reverse their interactions on the prototype. This feature relieves anxiety, since users know that errors can be undone, and encourages exploration of unfamiliar options [6]. Moreover, the buttons in the prototype are all meaningful such that the

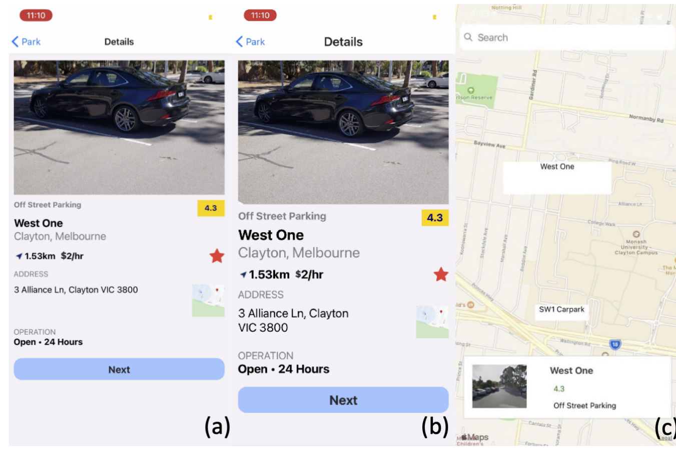


Fig. 3. Examples of incorporating support for aging users (a) and (b), and location awareness (c).

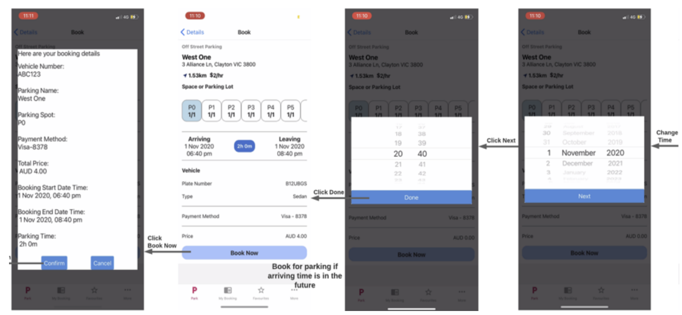


Fig. 4. Examples of reinforcing key information, redoing choices and confirmation of selections.

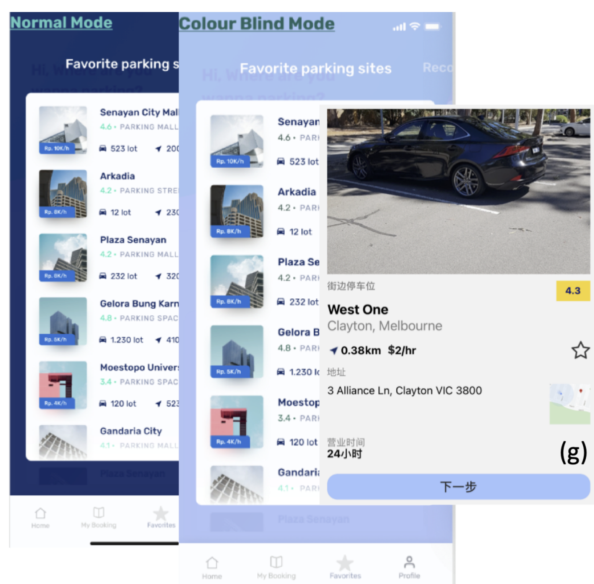


Fig. 5. Examples of incorporating support for colour blind users and non-English speaking users.

users can easily identify what each button does. Figure 4 illustrates examples of these design decisions.

The prototype is designed to have multi-language support to allow users to use the app in their preferred language [14]. The possibility of having multiple languages allows a wider range of users to use it effectively. Currently, it demonstrates this by supporting both Mandarin and English, but more languages could be added to the application to offer universal accessibility. Care also needs to be taken to support colour blind users and other users with particular sight or other challenges with smart city mobile applications [15], [16]. Figure 5 illustrates examples of these design decisions.

V. Prototype App

A. App Architecture and Design

The app architecture consists of the client, the servers, parking bay sensors and the database. Future enhancements could include interaction with variable time parking signs, multi-mode transport interaction, overstay detection and warning, and vision-based number plate recognition [4], [3]. We wanted to ensure that the application could support as many end-users as possible hence the decision to create a smart mobile application. The mobile app client utilizes the React Native framework which allows development to cater for both iOS and Android devices using the same code base. The client communicates with the server via RESTful APIs to send and receive user and parking data. The architecture contains two servers which respectively act as an intermediary between client-database and sensor-database. The database contains information about the users and data about all the parking locations supported. Sensors using image processing technologies [4] will be installed at the supported parking locations and would be used to update our database in real time. The server consists of different classes that provide user, park, payment, and sensor functionalities. For example, to get all of the nearby carparks to show up on the screen, the application would make a call to “CarparkService”. The app then maps these using a React Native mapping component, and displays a list of nearby carparks to the user.

B. Prototype Development

After finishing our multi-persona requirements gathering and analysis, we created a set of Figma-based user interface mock up prototypes for evaluation. Design and prototyping are important stages, ensuring that the usability and accessibility features are properly captured and implemented. Five authors each created a few designs taking into account the human-centric design principles, personas and user stories as described above. We then picked out designs and features that were the most appropriate for the app, validating with cognitive walk-through with the personas and their user stories. The final design created on Figma was a high-fidelity interactive prototype derived from the different designs chosen. We implemented a working prototype of our proposed smart parking app using the React Native framework. This framework allows web, iOS and Android applications to be largely specified with the same code base. Some iOS- and Android-specific components did have to be used to achieve some features, especially the mapping component. We built a fully functioning app prototype and tested its functionality against all user stories on both Android and iOS platforms. Due to the COVID-19 restrictions, rather than connecting our app to a real local council parking bay availability system, we mocked up available parks for a nearby geographic area.

VI. Evaluation

A. Evaluation Method

Due to the COVID-19 pandemic situation during the time of our app design and development, we were unable to test our prototype application with a set of target end-users face-to-face as we had planned. As a result, we decided to manually evaluate the prototype app by using a cognitive walk-through process with each of our personas and their user stories.

Firstly, we assigned a different author the persona and user stories than that they had helped to create. We carefully read the goals and frustrations that the different personas had for using a smart parking app. Based on their requirements as codified in their user stories, we tested the prototype app to check if the prototype could fulfil their goals while preventing or mitigating their frustrations. We tested it for both our prototype and the commonly used app discussed in Section 2. Using the data gathered, we determined if the user had a better or worse user experience with either of the apps, and areas where further work was needed. The process was then repeated for all of the personas.

B. Evaluation Results

During the cognitive walk-through, each of the personas responded differently and provided a variety of constructive feedback including both positive and negative comments, as summarised in table I. Through the cognitive walk-through process [17], we found that a majority of the positive feedback was attributed to the navigation function, payment confirmation page, and adjustable fonts and language. These functionalities greatly improved the user's experience as they followed more human-centric designs and supported different end-users in different ways. Some personas with disabilities or language barriers felt that the application's support for multi-language and color-blindness were very helpful for them. We analysed key differences when evaluating our prototype app and the commercial parking app. Our prototype contained a lot more human-centric designs and functionality. Some of the features designed in our app may not be core functionalities for a parking application but they play an important role for a user's experience.

C. Discussion

Evaluation of our more human-centric prototype parking app was mostly positive. The prototype addressed a majority of the five representative personas' frustrations with existing current day parking application. Most personas approved of the adjustable User Interface and believed that such a design had

TABLE I
EVALUATION RESULTS FROM COGNITIVE WALK-THROUGH WITH PERSONAS.

Persona	Satisfied Function	Issues found
Elizabeth Craw – aging woman	Navigation page displays the nearest car park which saves her time to find one. Figure 3 (c) shows example. Have the option to enable color-blindness support. Figure 5 demonstrates the difference between the two modes Adjustable fonts allowed her to view the app more easily. Figure 3 (a) and (b) demonstrate the difference between enlarged text interface and normal text interface. Payment confirmation page removes ambiguity and give assurance to her action. Figure 4 demonstrates this.	The booking function does not currently guarantee slot reservation. User Interface is still rough in places.
Ratna Smith – young single mother, English as 2nd language	Multi-language support is very helpful to her as she is non-native speaker. Figure 5 demonstrates this. Adjustable UI provides a clean and simple user experience. Integration of multiple car park in one app removes the hassle of using multiple parking app for different car parks. Navigation function saves her time to find a car park.	Her first language is not included in multi-language support, she wishes more languages could be added in the future.
Joe Marshall – aging male with mobility challenges	Simple and clear user interface. Can customize settings (font size/language). Multiple payment methods. Parking information summary and confirmation. Navigation function allows Joe to find the nearest car park to reduce walking distance.	User guide is not provided Disabled parking slot not marked.
Dave Thompson – mid-30s, multiple cars, colour blind	Car list for user to switch between different cars. Parking information summary and confirmation reduce ambiguity. Simple parking process is swift.	Payment history review not implemented.
Libby Balliol – young female student	Finding and booking an empty parking slot. Private and secure payment method.	User guide is not provided.

greatly improved their user experience. This especially helped the elderly and the vision impaired who might have difficulties reading smaller texts and normal color design. In addition to the customizable User Interface, multi-language support was another part that was appreciated by most personas as they are non-native English speakers and struggled to use existing parking application that did not implement localization support. The navigation function in our demo eased the pain to find parking locations near the user's destination point and gained a very positive feedback from our personas as it addressed their long-standing issues.

A number of aspects of our prototype could be usefully extended and improved: **Navigation** – a Google-map style guiding of drivers from their current location to the actual target carpark would be very helpful. **Integrating with a live carpark database** – fetching real time data from existing carparks by utilizing their sensor and image processing technology would assist the end-user, e.g. identifying actual park occupied, OCR recognition of plate, etc. **Guide for first time users** – when a user launches the app for the first time, it would be helpful to optionally guide them through all the functionalities in the app via a simple tutorial. This would assist several of our target use groups [18]. **Improved search function** – users should be able to search for a location and all vacant or potentially vacant carparks near that location should show up. Users might want to book a park in the general area but perhaps not a specific one. **Multi-language support** – the app needs support other languages other than English and Chinese and it would be good to automatically detect this from phone settings or other apps. **Notification support** – the app should notify the users when their parking time is about to expire with enough time to return to the parking space. **Extension of parking time** – where allowable and where parking space demand is not high, users should be able to extend parking times seamlessly. **Vibration support** – the app should have proper vibration support implemented for users that are unable to hear notification sounds, utilising native platform approaches and phone settings. **Disabled driver support** – we want to provide better app support mode for drivers with disabilities e.g. find parks close to their final destination, and be cognisant of their challenges. **Feedback reporting** – the app should provide an option to report not just software bugs but also give feedback on poor user interface designs or poor user experience, ideally supporting proactive adaptation of the app font, icon, language, terminology etc., based on the user feedback.

Key lessons learned and implications for practice:

Include diverse range of users in requirements gathering: Smart city supporting apps by their nature need to support a wide range of diverse users with different age, socio-economic status, language, education, gender, ethnicity, personality and so on. Diverse user needs have to be considered from very early on in smart city app development. **Support performing the same tasks in different ways:** Smart city supporting apps often require all users to perform tasks in a single way. However, where possible, apps should be designed to allow users with different cognitive styles, physical challenges, age, etc., to perform tasks with the app in the best workflow and screen designs suitable for them. **Accessibility is hard to retrofit:** A wide range of accessibility challenges present for users. Colorblind users, users with sight, hearing and other challenges, users with mild cognitive challenge, very young and old user, different languages used, different understanding of terminology, and many others must be accommodated carefully in the app design. Tasks being performed by these differently-abled users need to be accommodated, e.g., drivers with mobility challenges, aging drivers, and drivers with multiple vehicles due to job or family circumstances need to be supported in different ways to other users. Such support is very hard to add in after an initial app release. **App reviews, rich personas and user stories help:** We found our diverse target end-user personas, researched from a variety of sources, app review analysis of end-user views on current parking app problems, and defining sets of overlapping user stories for each target end-user persona greatly helped to guide our app design and development. These approaches do not replace using real smart city target end-users, but we found they aided in understanding and appreciating the variety of end-user requirements and current parking app frustrations.

VII. Related Work

Various smart parking systems have been developed leveraging IoT-based sensors in various ways [3], [4]. For example, the use of sensors to detect specific parks, inform of vacant parks, with the aim of helping drivers and smart city managers to better find, utilise and monitor parking. Most do not seem to support diverse human user differences. Various design principles have been proposed and trialled to improve support for aging users, including mobile apps in various domains [7], [19] However, many existing apps, including all of the parking apps we reviewed, do not appear to follow many of these principles. Colour blind and other sight-challenged user needs and design principles have been well-studied [16], [15]. Many widely used parking apps also fail to accommodate colour blind users and provide limited or no support to those with eyesight challenges. Multi-lingual support has been studied but many apps fail to properly support different spoken languages [14], let alone provide different labels, icons, prompts and dialogue to users with different educational level, age, etc.

There are a variety of different human-centric issues that affect how a person interacts with an application and a few examples of them are personality, gender, emotions, engagement, entertainment, ethnicity, culture, age, values, physical and mental challenges [2]. A more human-centric approach to software engineering is needed to be able to capture accurate requirements when designing an application whilst taking into account all of these different human factors.

VIII. Summary

We identified some of the key limitations of current smart parking applications with lack of accommodating diverse end-user human-centric issues. We created five personas to simulate key target end-user groups and created user stories from each of the different personas' parking frustrations and goals. We developed a rich design prototype with Figma and iOS and Android smart parking apps using the React Native framework. We evaluated our prototype apps by having test subjects go through a detailed cognitive walk-through based on each personas' characteristics and user stories. Most evaluation outcomes were positive and user experience was improved and our prototype is more inclusive for a diverse range of users regardless of their cultural background, age category, fluency in different languages, and their mental and physical health.

IX. Acknowledgements

Grundy is supported by ARC Laureate Fellowship FL190100035, Obie by ARC Discovery Grant DP200100020 and Karolita by an Australia Awards Scholarship.

References

[1] B. N. Silva, M. Khan, and K. Han, "Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities," *Sustainable Cities and Society*, vol. 38, pp. 697–713, 2018.

[2] J. Grundy, H. Khalajzadeh, J. McIntosh, T. Kanij, and I. Mueller, "Humanise: Approaches to achieve more human-centric software engineering," in *Evaluation of Novel Approaches to Software Engineering: 15th International Conference, ENASE 2020, Prague, Czech Republic, May 5–6, 2020, Revised Selected Papers 15*. Springer International Publishing, 2021, pp. 444–468.

[3] L. Mainetti, L. Patrono, M. L. Stefanizzi, and R. Vergallo, "A smart parking system based on iot protocols and emerging enabling technologies," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE, 2015, pp. 764–769.

[4] A. Khanna and R. Anand, "Iot based smart parking system," in *2016 International Conference on Internet of Things and Applications (IOTA)*. IEEE, 2016, pp. 266–270.

[5] T. Fabusuyi and V. Hill, "Designing an integrated smart parking application," *Transportation Research Procedia*, vol. 48, pp. 1060–1071, 2020.

[6] I. S. MacKenzie, "Fitts' law as a research and design tool in human-computer interaction," *Human-computer interaction*, vol. 7, no. 1, pp. 91–139, 1992.

[7] Z. X. Jin, T. Plocher, and L. Kiff, "Touch screen user interfaces for older adults: button size and spacing," in *Int. Conference on Universal Access in Human-Computer Interaction*, 2007, pp. 933–941.

[8] J.-M. Díaz-Bossini and L. Moreno, "Accessibility to mobile interfaces for older people," *Procedia Computer Science*, vol. 27, pp. 57–66, 2014.

[9] P. Gregor, A. F. Newell, and M. Zajicek, "Designing for dynamic diversity: interfaces for older people," in *Proceedings of the fifth international ACM conference on Assistive technologies*, 2002, pp. 151–156.

[10] L. Ivan, E. Loos, and G. Tudorie, "Mitigating visual ageism in digital media: Designing for dynamic diversity to enhance communication rights for senior citizens," *Societies*, vol. 10, no. 4, p. 76, 2020.

[11] D. Norman, *The design of future things*. Basic books, 2009.

[12] D. Norman, *The design of everyday things: Revised and expanded edition*. Basic books, 2013.

[13] B. Shneiderman and C. Plaisant, *Designing the user interface: strategies for effective human-computer interaction*. Pearson Education, 2010.

[14] J. Roturier, *Localizing Apps: A practical guide for translators and translation students*. Routledge, 2015.

[15] J. S. Sierra and J. Togores, "Designing mobile apps for visually impaired and blind users," in *The Fifth international conference on advances in computer-human interactions*. Citeseer, 2012, pp. 47–52.

[16] M. G. Carcedo, S. H. Chua, S. Perrault, P. Wozniak, R. Joshi, M. Obaid, M. Fjeld, and S. Zhao, "Hapticolor: Interpolating color information as haptic feedback to assist the colorblind," in *2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 3572–3583.

[17] D. Fichter, "Heuristic and cognitive walk-through evaluations," *Medford*, vol. 23, no. 3, pp. 53–56, 2004.

[18] D. Williams, M. A. U. Alam, S. I. Ahamed, and W. Chu, "Considerations in designing human-computer interfaces for elderly people," in *13th Int. Conference on Quality Software*, 2013, pp. 372–377.

[19] S. H. Kurniawan, A. King, D. G. Evans, and P. Blenkhorn, "Personalising web page presentation for older people," *Interacting with computers*, vol. 18, no. 3, pp. 457–477, 2006.