Aalto University
School of Science
Master’s Programme in Computer Science and Engineering

Ashiq Ahmed

Measuring Developer Experience of a Digital Platform

Master’s Thesis

Espoo, January 8, 2018

Supervisor:	Professor Marko Nieminen Associate Professor Eduard Petlenkov
Thesis advisor(s):	Fabian Fagerholm, Ph.D

AALTO UNIVERSITY
School of Science
Master's Programme in Computer Science and Engineering

ABSTRACT OF THE
MASTER´S THESIS

Author: Ashiq Ahmed	
Title of the thesis: Measuring Developer Experience of a Digital Platform	
Number of pages: 69	Date: January 8, 2018
Major or Minor: Control and System Engineering	
Supervisors: Professor Marko Nieminen, Associate Professor Eduard Petlenkov	
Thesis advisor: Fabian Fagerholm, Ph. D	

Smart city and smart transportation are concepts that have emerged as an enabling solution which facilitates the grassroots social innovations to mitigate the problems generated by rapid urbanization and population growth. The digital service platform has fostered a new paradigm of transportation by involving all key players to create a novel environment. It is concerned developer are also the user of the platform as they are using the system development tools and methods for further development, that is why developer experience over the platform plays a vital role. Delightful developer experience not only improving the platform performance but also invokes to introduce new innovations. In this research we off to measure developer experience and answering the research questions "how to measure developer experience on top of the digital service platform" and "how to analyse the developer experience".

In the state of measuring developer experience, an application has been developed over the digital service platform and a measurement procedure has been introduced by modifying System Usability Scale (SUS) to more suit the context of the developer. The SUS has been borrowed from UX measurement tools as developers are the user of system, system development tools and methods as well as SUS is a widely accepted tool by the usability researchers for measuring usability. The result of the proposed method showed superior experience from the developer's perspective to develop the application over the living lab bus platform. The result is almost same when it is compared with another method, but it is arguable as it showed small discrepancy. Furthermore, it can be said that, this research provides a straight forward way to measure developer experience on a digital service platform.

The answer of the research questions provides a detail guideline of the measurement process and analysing criteria of developer experience. Moreover, it comes out with few recommendations that can be helpful for the developers of the platform to improve the platform in future, so that it could ensure the delightful experience for the developers.

Keywords: Smart Transportation, Digital Platform, Developer Experience, Developer Experience Measurement	Publishing language: English

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else. This thesis has been done in Aalto university as an exchange student and will be submitted to Aalto university and Tallinn University of Technology for completing the Master’s degree program.

Author: Ashiq Ahmed

30.12.2017

Acknowledgements

First and foremost, I would like to thank my parents who have been supported me throughout my whole life. I am also thankful to Dr. Eduard Petlenkov (Tallinn University of Technology) who was admirably patient, and supported consistently throughout the way of this research. I am in huge debt to Professor Marko Nieminen (Aalto University) for his valuable time, numerous meetings, advices and encouragement all along the way of the research. I am grateful to my instructor Dr. Fabian Fagerholm (Helsinki University) and this research would not be possible without his help. I am also thankful to my friends Balaji G., and Younus Ali for their valuable comments and reviews.

Finally, I would like to thank Mr. Tuomo Parjanen (PayiQ) and Mr. Immo Heino (VTT Technical Research Centre of Finland) for sharing their valuable insight which helps to proceed the research further.

List of abbreviations and terms

LLB	Living Lab Bus
LL	Living Lab
SC	Smart City
DX	Developer Experience
GPS	Global Positioning System
API	Application Programming Interface
MQTT	Message Queue Telemetry Transport
IoT	Internet of Things
UCD	User Centric Design
SUS	System Usability Scale
SDK	System development Kit
L3	London Living Lab
UX	User Experience

Table of contents

List of figures

List of tables

1 Introduction

Though modern cities are well equipped with an advanced transportation system and facilitate it with a reliable public transportation system, it still cannot meet the needs of a smooth commute for the passenger [1]. The transportation domain has experienced fundamental changes during the last decade as the Internet and smart devices are now used. As the cities are overpopulated, meeting the transportation needs of the people makes the city messier and polluted [40, 42, 43] with greenhouse gases. Radical reduction of greenhouse gas emissions is a big challenge these days. However, increasing public transportation, and introducing more renewable technology based vehicles can solve this problem. Moreover, new vehicle technologies and traffic behaviour can be controlled through smart transport system [16].

Smart transportation is the integral factor of a Smart City (SC) concept. Being smarter, commuting should be a smoother, environmentally friendly, and user-friendly transportation system with less traffic congestion [49]. It is considered, both private and public transportation are very important to improve sustainable regional transportation systems as well as ensure safety. The growth of public transportation systems creates a significant impact on economic growth for the region [9]. The development of technologies has enhanced the attractiveness of the recent public transportation system and has brought a dramatic improvement in the provision of services. The introduction of user generalized mobile device applications, location-based web services, open data, open interfaces, and the third-party development of service have tried to smoothen the journey of the regular commuter by modernizing the services. But personal commute still needs a more detailed guidance [13, 19].

Recent modern public transportation systems incorporate a large number of sensors including GPS, cameras, and road sensors to collect information [6, 4, 13]. The data generated and received by the transportation system has multiple dimensions. It is not limited to acknowledge the routes, available vehicles, journey completion times, distances, and road congestion. The information on the automobiles through sensors installed into the vehicle like speed, fuel consumption, weather conditions, door opening and closing, and driving style of the driver is also collected. The information can be collected from an open web-based API over the Internet and cloud.

Integration and optimization of this vast amount of data generated from the transportation system is the challenging part [9, 13] as it is continually changing, and a decision has to make on the fly by the system for running the public commute smoothly. The system has to provide a guideline for further commuting, allocate resources, route optimizations, and strategic decisions for next stage of transportation. The advancement of IoT and big data associated with the infrastructure and the platform made a huge impact on recent transportation system. The transportation system has improved incredibly with the association with Big data applications as it can diagnose problems automatically, do dynamic route planning, and solve the challenges while in motion [6, 9].

Although the current transportation system is providing a few options to smoothen the journey, the generated data is not enough to create value for the passenger, and the entire platform is scattered. Passengers need to access different interfaces for collecting required information on a daily commute and that is troublesome. The Smart service platform brings all the necessary options into one common digital platform and to facilitate the user on a daily commute. The digital platform is the concept of delivering real time information to a user through an information system [9]. Different actors (users, vehicles, and infrastructures) over a digital service platform have a close tie with each other, and these interactions generate a significant amount of data from different heterogeneous sources. If the user can access all the information from one interface, which comprehensively integrates with the data sources, then that would facilitate the optimization of operations as well as the enhanced passenger experience. One can quickly get the schedule of the vehicle, route information, location, travel time, speed, door information, fuel consumption ratio, temperature of the local area, traffic, driving style of the driver, vehicles physical information and cameras, and suggestions for spending spare time at the station. All the data from different heterogeneous sources will come from open API, and the interface will connect to the cloud server.

The Living Lab Bus (LLB) is a novel digital service platform for the transportation system. It is working for smoothening the journey of daily commuters with variety of services. The ecosystem of this platform involved with web portal where services are offered for users and innovative electric buses that itself working as a part of platform information service provider. The LLB is a system that brings all the stakeholder onto the same page and facilitate the user perspectives. It has a close tie with the private companies and research organizations as well as with the public sector. The LLB platform uses open APIs provided by the third parties, which makes this platform more apt for the purpose of smoothening the journey of a regular passenger.

Like other digital service platforms, LLB provides a technological model which enables the stakeholders to structure business models, and facilitates value exchange and trust built-up among the stakeholders [66]. It provides a channel with open connectivity to the user for introducing new services with rapid scaling possibilities and facilitate all the stakeholders by exchanging values [67]. The user centric approach leads to the development of noble service platform in which user experience is focused, whereas the service oriented development strives to improve developer experience (DX) [39]. The platform must create value to the developers and facilitate them with leverage the functionality of the platform. If the platform does not offer what developers need, developers might not want to use the platform. LLB platform must provide excellent developer experience (DX) for keeping the developer interested to develop service over the platform.

Though DX is very essential for any service platform, but DX is not well versed these days. According to Nielson&Norman group [68], designers of system are not the users of the system. Fagerholm et al. [36] gave similar opinion. According to them, developers are the user of development tools, methods and systems which are being used to develop the system for the end user. But, Kuusinen [38], explained developers are being rarely seen as a user of the development tools based on a study result. Developer has dualistic nature into the system; as the user of the system, system tools and the designer of the system what makes the developer special [38].

According to [36] Developer Experience (DX) is similar to UX, but DX only concern about developers who are responsible for designing or developing the system for the end user. The developer has to have interest, feelings, passion, focus, and understand the values of the application that are created for the user. As such, the platform has to generate profits for the developer, giving flexibility, and draw attention and interest in the development of the platform.

Measurement of DX is complicated and can be a misleading over the system. It completely depends on developer's experience, technical knowledge and individual skill. It is not necessary that all the developer experiences need to be same on the same platform, as work areas and skills are different from each other. In [36, 71, 88], suggested UX tools can be used to measure developer experience as developers are also the user of the system, tools and development methods. They pointed out UCD, user persona and UX measurement tools can be used for the measurement process.

In this research, we will measure the DX of LLB platform. The System Usability Scale (SUS) will be used here, which is a simple quantitative tool among usability researchers to measure usability for its accuracy and simplicity. But, we introduced a modified version of System Usability Scale (SUS) to fit in developer experience measurement process. The research will be forwarded by developing an application over LLB ecosystem and measured developer experience by inducing our proposed method.

The research will be preceded further to find out the answers to the following research questions.

1) How to measure developer experience on top of the digital service platform.

2) How to Analyse the developer experience

This research aims to provide guidelines to the developers of the LLB platform for its further development which will help to decide the most convenient way to ensure the developer experience.

The sole purpose this research to introduce an approach of DX measurement on a digital service platform. The role involved with the research to measure the DX by employing the proposed method, verify the proposed method by comparing with another DX measurement method which is derived from Fagerholm et al. [36] conceptual definition and develop an application over the service platform. The measurement result and verification based on the input of a single person.

2 Background Work

Smart transportation system is an integral part of smart cities. LLB provides a way to develop application for the smart transportation systems as a part of the system. The intention of this research to analyse the developer experience by developing an application top of LLB platform based on user needs. This section briefly describes the smart city, smart transportation, living lab, digital service platform, persona, user centric design and developer experience.

2.1 Smart City

The concept of the Smart city (SC) lies on the connection between the human, innovative technologies, academia, technology and business institutions [40]. It is considered that the advancement of the Internet of Things (IoT) have changed the way of living and has an influence of developing SC [44]. In broader sense, it can be said that the SC concept is an ecosystem [49] that has provided smart space for the city dwellers within their context, with the broader community and the whole city. It brings all the stakeholders on the same page by a connection string where humans are acting as a network platform and connected via various networks.

In last couple of decades people are migrating more to the urban area [40]. Half of the people of the world is living in metropolitan area now. As the number of people increases in urban spaces, the cities are becoming messier and a more disordered place [41]. Cities are facing new problems like air pollution, traffic congestion, scarcity of resources, waste management problems, and aging and deteriorating infrastructures. Even insufficient health services are also a physical, technical and material problem [42, 43]. Apart from these issues, there are a lot of social, natural, political, and organizational obstacles associated with different stakeholders. Above all, high levels of interdependency, social, and political complexities make the issues more complicated in those cities.

By Keeping eyes on these problems, it is considered new technology based solutions, innovative planning and living could help to soothe but cannot make radical changes [46]. Though K. Su et al. [44] claimed Smart Cities (SCs) would be the future of urban development, but it does not imply that those cities will be better cities to live. Even the performance could not be measured by counting the smart initiative has been taken by those cities, but it could reflect the efforts have been taken to improve the quality of life of a citizen [46].

Basically, SC is a concept of making a region smart by bringing in standard information and an integrated management system [44]. It is a collective way of effective integration of smart ideas, plans, construction modes, management methods, and above all, smart development approaches. Digital grid management of urban territories, resources, social economics, environment, digital information processing, and application infrastructures will be required to ensure SC management services are efficient with convenient and harmonic operations.

K. Su et al. [44] divided the construction of the cities into three levels, including the installation of an application system, construction of the public platform, and general network infrastructure. The authors put more emphasis on the development of an application system in the creation of an intelligent wireless city, including the home, public services and social management. Transportation, medical, urban management, green city construction, and tourism will be included of the application system. For the development of a SC, few components have a significant impact on it, such as the economy, urban structure, geographic area, population density, congestion, and transportation [46]. It is understandable that geographical location has effects on SC strategy, traffic, and congestion along with population density and helps to determine the path of a SC implementation. Above all, urban structural components and economic development influence the SC development.

The rapid population growth by migrating to urban areas all over the world requires a deeper understanding of Smart city concept [45]. It not only ensures the liveable condition for the city dweller but also reduce the challenges that the city faces with high population growth. This expansion creates an urgent drive to find a smarter solution and many cities around the world are already adopting the concept, but the conversion of the whole city is challenging. The adaptation process not only requires cost large amount of money but also need strong technical support. Moreover, SC solution is completely user oriented and in any SC citizen is the most critical asset. Citizen of the cities are directly involved with the innovation process, testing, verifying and validation. For that reason, a specific region is selected as a testbed of those innovation and experiment and users are part of the testing, verification and validation process [56].

2.2 Smart Transport

Communities are transforming into SCs with innovative planning, management, and operations [20]. Smart transportation systems are the part of SCs that smoothen commuting of the citizens of those cities. The system is not only limited to the scheduling of vehicles [16] but also it provides useful information that is beneficial for the regular commuters.

Smart transportation is an interconnected network of vehicles, sensors, and social networks. The effectiveness of the transportation system depends on the responsiveness of the system, data requests handled on the fly, and the decision-making procedure [26]. It is a collaborative design with the involvement of different stakeholders [27, 28] in combination with cloud-based architectural design and the Internet of Things (IoT).

Baek, J.S. et al. [31], proposed a collaborative service in which users can collaborate with each other to fulfil their goal by providing an innovative service. Their proposed method will reduce the carbon footprint, as it is the primary concern of Smart city concept, and socially connect people with each other. Paul Holleis et al. [30] explained a mobility pattern for personal moving via social networks and created a system of mobility to use sustainable transportation modes as well as generate the win-win situation for all its stakeholders.

It is understandable that advancement of technology forces the redevelopment of the current public transport system. It is giving the opportunity to create a novel system to facilitate the user with high-tech, and powerful user-centric systems. Moreover, opening a new ecosystem with social involvement makes efficient commuting more feasible [4]. IoT blessing also makes the system smarter as the involvement of microcontrollers, transceivers for digital communication, and suitable protocol stacks make communication ease [21]. As IoT decreased miscommunication, the interaction between the actors of the smart transportation system increased [4], helped generate the data set, and increased the amount of data, which helped improve the transportation system. Moreover, introducing different advanced tools, methods, and technologies like Apache Spark Apache Hadoop, MapReduce and Distributed File System (HDFS) [23] not only optimize the data efficiently for the end user but also help to share the data with third parties.

Though Smart transport system improved, but S. Wibowo and S. Grandhi [29], explained more adequate research on Smart transportation is required, and they made a benchmark of comparing the performance of Smart transportation systems existing in different cities. Moreover, it reveals the existing situations and widens the scope of improvement.

2.3 Living Lab

Previously, scientists and experts were only responsible to develop innovative technologies, but in recent era anyone can be a driver of creative technological development [56]. Because of user's involvement with experts in innovation process, user experience becomes the most important factor of any technology driven applications. In most cases, these techniques are failing to serve, not because of advanced technology or lack of superior knowledge. It is failing because it cannot meet the needs of the real user [52, 53]. The Living Lab (LL) is the platform for testing the product and services practically, providing the opportunity to get direct feedback from the user, and have data to modify user feedback. Moreover, it is useful for promoting an open environment to bring in new technologies [53].

The concept of Living Lab (LL) is not new, and it is a way to bring government, industry, and academia on the same page [67]. Through it, all can collaborate for the development of new solutions and create strategies for a specific region. LL concept is entirely user-centric with the adaptation of new technologies. Many describe LL in different ways. DEll'Era et al. [50] define LL as a design research methodology with the involvement of real users with real lifestyles that aim to co-create the innovation. Eriksson et al. [53] and Schuurman et al. [65] define LL approach as a user-centric methodology that can sense the user needs with prototyping, validation of complexity, and evolvement to real-life situations. Feuerstein et al. [66] proposed LL is a systematic innovation approach where all the stakeholders are directly participating in producing a development process. In the Living Lab Handbook [63], it is described as an open innovation environment where user-driven applications play a vital role in the co-creation process with real needs in respect of new services, product, and social infrastructure. Konsti-Laasko et al. [64] described LL as a concept of continuous R&D processes that focused on creating innovation in a multi-contextual view and satisfy real world needs.

Analysing the concept from [53, 63,50] it can be defined LL approaches are used to develop sustainable innovation or services as follows:

- Developing methods that are on top of the user-centric design, context, socialization, and prototyping and validation.
- Open innovation processes for a sustainable setting.
- Experiment with standard research protocol and data collection; validate the result within the context of the Living Lab and a reliable resource to satisfy scientific criteria.

More precisely, LL can be a mean of reaching the goal of SC concept as it accelerates the innovation with the help of different stakeholders while they are collaborating to create, validate, and test new technology with the real-life context [53]. This concept makes the pathway to generate the values and simulate the innovation reaching toward the SC dream through tailoring the user-centred innovation with the involvement of different stakeholders. The Living Lab is a tangible asset that provides the technological innovation, facilitates the users, and fulfils their needs, which creates a guideline for entrepreneurs. Above all, it always leaves a scope to improve changes while it’s needed [56].

The Living Lab is the small model of a Smart city, which is almost the same structure incorporated with related sensors, APIs, applications and generates similar data in small scale. But it has a broad scope to analyse the overall system with retrieved data, understand, and project in large-scale applications. Esteve et al. provided the way of covering up the gap between the Living Lab project and individual innovation methods and showed how it created an impact with the involvement of users in the innovation process on the overall system. They proposed interviewing persons involved with the Living Lab, and the user could be a way of research design.

2.4 Digital Service Platform

In the era of continuous innovation digital platform presented a compelling solution to the stakeholders in terms of value proposition and business model [82]. The definition of digital platform like a technological model to enable business models, facilitates value exchange between different stakeholders and building trust. Moreover, it ensures compelling user experience with open connectivity and massive scalability without degrading performance [91]. Digital platform creates a channel for the end user, business owners, and technology leaders which allow rapid scaling based on mutual trust, expectations and cost-efficiency. It works as grease for flywheel by facilitating all the stockholders with rapid exchange of values [83].

Application programming interfaces (APIs) are the by mean of making digital platform working. Though APIs are working as a basis of business model and back force of building an ecosystem, but it does not imply that digital platform cannot be functional without APIs. It can be developed with incremental model in total ecosystem [84]. As digital platforms are target oriented and becoming the channel of service, APIs become strongest part of business performance [74]. Even it has been proven that API uses as part of digital platform not only improved business performance but also improved the collaboration with different stakeholders. The collaborative platform can be a long-term goal to achieve, but can hit short term goal while developing and leverage the joint business of the stakeholders [74]. The main goal of the platform to make changes and innovate faster connecting facility with data devices and applications.

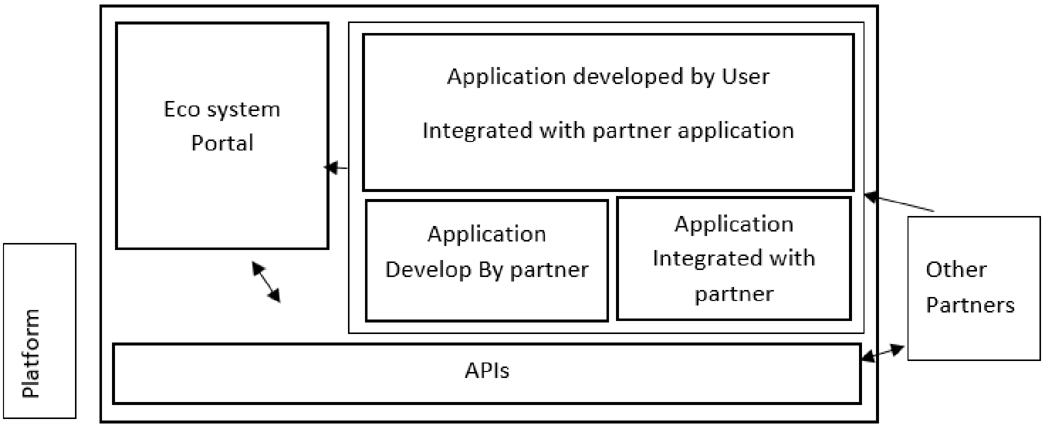


Figure 2.1 Conceptual Structure of Digital Service Platform

2.5 Persona and User Centric Design

Persona is the most understood and accepted method for clarifying the user’s inclination to provide better user-oriented services [33]. A persona is a user model that serves as a repository of all the user expectations for a provided function [35]. Persona is the abstract model of a user preference of the system or the services that they would like to use, and it is the primary driving point of requirement engineering. Persona holds the all the decisions of user surroundings and preference lists, parts of the component they would like to choose, and portability. Jon Orwant [34] defines persona as a user model that possesses all the knowledge of a lifelong friend, and the user must understand it. According to Orwant, user characteristics can be predictable, like why he is doing something and what he is going do in next.

User Centric Design (UCD) plays a vital role in product or service development. In fact, UCD makes the product more user-oriented. The customer need-based approach is an established formula for the development purpose of new services or products by knowing the preferences and behaviours. Persona development is a part of the UCD method. Persona scenarios make the products or services more goal-oriented and reduce conflicts [32]. Persona is applied in the early phase of service or product design. Though it is difficult to understand the user expectation and it changes as time goes by, system boundaries can quickly follow it. Therefore, the user experience needs also to consider after-the-service or product development. In [33], S. Hosono et al. proposed persona-centric service design where stakeholders are related to the service function that reduces the disparity of service and the customer expectation.

In the Smart city and the Living Lab, projects and stakeholders are intimately involved. Users are closely affected, and their expectation can be reflected with the service design. Enterprises are finding more opportunities to improve the product with an agile methodology. Hence, products or services are becoming more user-centric and reducing the potential risk of the product. Eventually, users will practically use the product and give their feedback, so service developers can identify system boundaries and explore more features by summarizing user persona.

2.6 Developer Experience

DX is the concept [36] of capturing the feelings, motivation, characteristics, and activities of the developer while developing the systems. In figure 2.1, developer experience and interaction has been illustrated. DX is quite similar to user experiences [36], except it measures how the developers are feeling, their mentality, activities, characteristics, and vision of the outcome in development. It is inspired by the UX practice and recognize the developer as a user of the development tools for developing application or system for the end user [72]. Naturally, it can be said that the developer has dualistic nature over the system by applying UX definition; being a user of the system tools and the producer of the system [38].

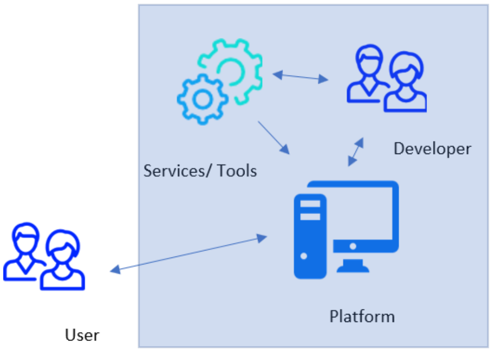


Figure 2.2 Developer Experience

Fagerholm and Münch [36] proposed a conceptual framework for explaining developer experience. Figure 2.2 illustrated the conceptual framework below. According to them, software development is a constructive work, so the developer may have a distinct idea about the infrastructure, viewpoint and feelings of the work, and knowledge of the values that are created by achieving the target. It is a correlation between cognitive, practical, and conation factors, where cognitive factors are related to the execution of the development itself, association with socializing, connection and overall working feelings with the team while developing the product. Conation factors include elements referring to their motivation, planning, and goal-oriented approach.

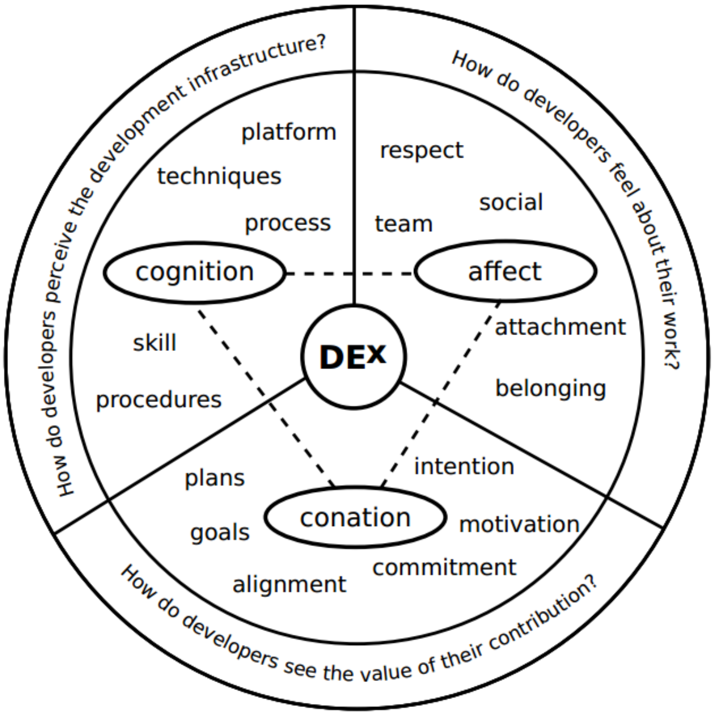

How do developers perceive the development infrastructure?
How do developers feel about their work?
How do developers see the value of their contribution?
platform
techniques
process
respect
social
team
cognition
affect
DEX
skill
procedures
attachment
belonging
plans
goals
alignment
conation
intention
motivation
commitment

Figure 2.3 Conceptual Framework of Developer Experience [36]

Palviainen et al. [37] explained the DX differently and connected it with activity theory. They divided the DX into categories which are involved with tools, development rules, objects, actors, community, and work division between team in the developing phase. Kati Kuusinen [38] did a comprehensive survey and showed the improvement of development tools as one way of achieving DX as these tools are providing better support to the developers in their development activities. Justin baker [89], expressed not only tool improvement is necessary for achieving excellent DX but also pointed out scalability, easiness and reliability needs to improve DX.

DX has many possible application and audiences on different sector. In service development platform developers are concerned the real user of the of the system [72]. In service development platform, DX can be defined as the sum of the interaction between the developers and the platform [71]. According to Pamela fox [73], it is the cumulative value of both positive and negative between the developers and tools, API or library.

To get grip in real world pace, product designer need to think in user centric way and need to focus user needs. User centric approach may lead to develop better service platform where user experience is focused, and service oriented development strives to improve DX [90]. The platform needs to create values where the developers can easily leverage the functionalities of the platform. So, platform developers need to understand the emotional state of the developers which is known as empathy [75]. The platform developers need to think: what kind of functionalities and usability are expected by the end users? What kind of emotional bonding they are expecting?

All the developers are not homogeneous, they have different skill set. For an instance, developers who are responsible for developing tools for the browser and operating system by using C++; the end user are web app developer more interested on JavaScript and web service APIs. It can be argued that both share same DNA, but skill set is different. So, to ensure better DX, it is necessary to know the end user. To make the process happen, experimenting and using over own product and feedback process collection process could be applicable [72].

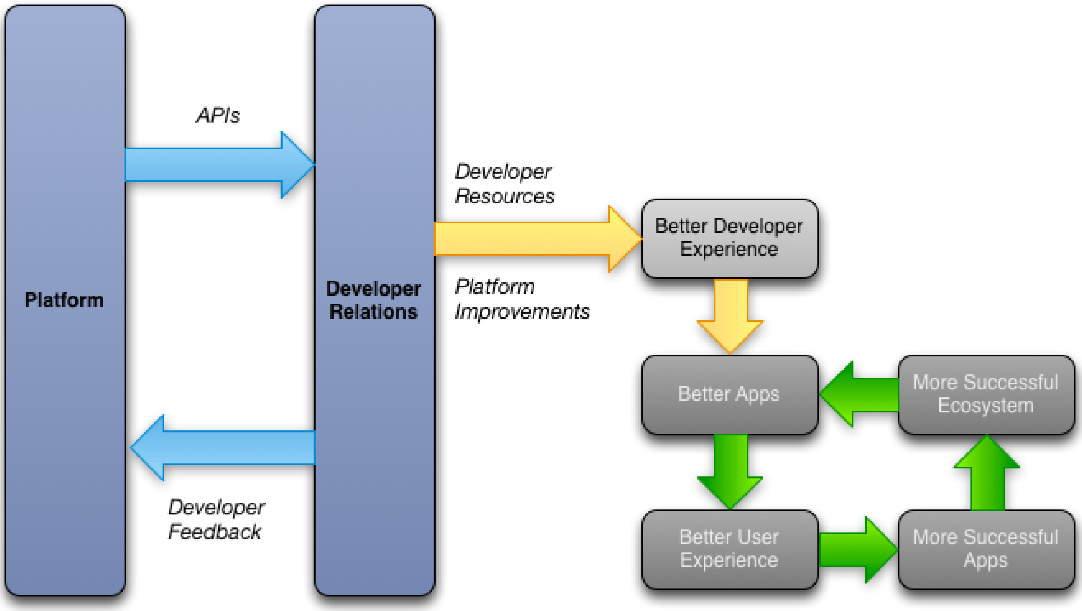


Figure 2.4 The virtuous cycle of Developer Relations [86]

In figure 2.2 virtuous cycle of developer relation has been illustrated. By Analyzing virtuous cycle of developer relation over the platform, it can be said developers can have affection with the development platform, if and only if the platform is empathetic to the developers, ensure usability, and can meet up developer's expectation with leverage functionalities. If it does, then it can influence the developer to develop successful application for the ecosystem

The measurement process of DX still tricky and need more focused research over it. In practice field, DX measurement process has been explained by Mike Brevoot [88]. According to him, developer relation depend on developer success and DX depends on UX. As DX encompasses all the aspect of UX in the ecosystem, so UX tools can be applicable in the measurement process. He proposed User Persona, developer's journey map and watch, ask and listen formula could be a way to measure DX. Pamela Fox [73], proposed a straight forward approach for measuring developer experience and pointed out that the journey map could be used in the measurement procedure of DX on any ecosystem. Roonie [71] proposed another method in practice, and according to him, accumulation of User persona and measurable usability factor could determine the type of experience what developers could have. Fagerholm et al. [36], pointed out developer experience could be measured by focusing positive experience, appropriate and efficient use of the system or platform. It can be transformed as experience of the systems, functionality and usability of the systems.

3 Living Lab Bus

In this chapter we first describe the living lab bus project briefly. Then we describe the objectives and goal of the project, technologies that are being used to develop the facilities, working procedure of the LLB developer portal, and the LLB data collection protocols.

3.1 The Living Lab Bus

The LLB focuses on using the public transportation service seamlessly and encourages the users to use this flexible, fully customer-oriented chain of the transportation system that is developed for smoothening the commute with cost-effectiveness and in a low emission manner. The LLB project is involved with the environment-friendly design and the involvement of various parties, but it mainly focuses three areas: smoothening the multimodal transportation chain, user experience and comfort and technology-oriented.

The LLB project provides is a creative platform which is designed to enable the continuous development of services for the sake of making end users life easier on daily public commute. In fact, it is promoting the mass communication and giving solutions in the real-world environment by implementing the pilot project. Overall, it verifies the plan for the end-user based on experience and continuous feedback from the user. It is mainly concentrated small or large operator groups and developers. A group of interested actors is involved with the LLB project regarding regional business growth. Mainly, the target groups of the LLB project are: the users of the service, public transport operators, and the business organizations. Moreover, The LLB development and testing environment will be used in national and EU projects as well as other international scopes. The introduction of accelerating services the development environment will be a competitive advantage for future innovation on public transport sector.

3.2 Objectives, and Goal of LLB project

The overall objective of the LLB project is to develop a common platform that contributes to the development of travel chain services and smoothen the public transportation system by using new technologies. The LLB project has different stakeholders with different needs and roles. A few key stakeholders are: passengers (end-users), cities, transport operators, service providers and technology suppliers. Those Stakeholders are involved for achieving common goals, and those are:

- A smooth, pleasant and economical movement.
- Functional, safe and environmentally friendly transportation.
- Environmental and operating technology.
- User-friendly solutions and new business opportunities.

The goal of the project is to create an ecosystem where new services can be developed, tested with the co-operation of the researcher, enterprises, and the users. The project is shaped to understand the requirements of different parties. The ecosystem has given a platform to produce a modified business plan and model for earning through cooperation models and opportunities for the development of innovations in transportation services. Moreover, it is a gateway to develop new services, tested in the real environment that creates a dimension to the commercialization of the services with accessible, faster, visual effect estimation, and has a scope to connect with the user using Mobility-as-a-Service solutions directly.

3.3 Technological Aspect of the LLB Project

The core concept of the project is to give the user a high-end technological experience and user-centric interface to ensure the delightful user experience with their regular commute. Also developing Mobility-as-a-Service that aim to meet the needs of the user door-to-door mobility services as well as encouraging sustainable mobility. The LLB enables an agile development environment with experiments, together with the professional. The development oriented companies and public-sector decision-making companies accelerating the coming of solutions in the market, and promote the acquisition of credible references. Structural overview of the LLB ecosystem is illustrated in figure 3.1 below.

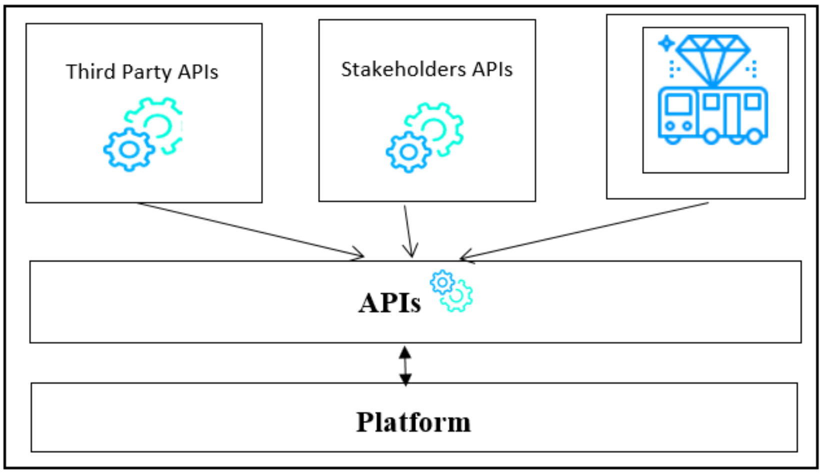


Figure 3.1 Structural Overview of LLB ecosystem

The LLB ecosystem is a combination of LLB developer portal, API's, the hardware installed buses, stakeholders and other third-party service providers. In the figure 3.2 illustrated below is the technical overview of the LLB ecosystem.

More precisely, LLB platform is an open development environment with the connection of physical buses and LLB developer portal. That is given a scope to introduce new models to test, explore new ideas, business models, as well as service development and earning. It is a simple way to generate new services and verify with a flexible and controlled manner.

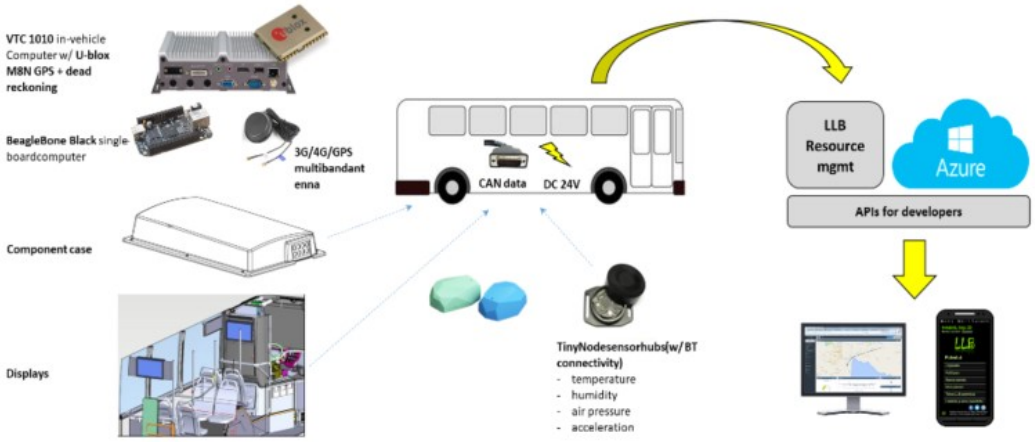

VTC 1010 in-vehicle Computer w/ U-blox M8N GPS + dead reckoning
BeagleBone Black single-boardcomputer
3G/4G/GPS multibandant enna
Component case
Displays
CAN data
DC 24V
TinyNodesensorhubs(w/ BT connectivity)
- temperature
- humidity
- air pressure
- acceleration
LLB Resource mgmt
Azure
APIs for developers

Figure 3.2 Working procedure of LLB platform in big picture [87]

3.4 Bus as a platform

The bus is acting as a part of platform in the LLB ecosystem and is incorporated with many sensors, cameras, hybrid vehicle computers, Beagle Bone, GPS systems and multiband GPS antennae. The bus can collect information both inside and outside using sensors and cameras. As the buses are incorporated with both inner and outer sensors, those sensors are collecting separate kinds of information that helps the passenger to be aware of the situation before getting off at the bus stop.

Through internal sensors, air quality (temperature, humidity, air pressure, CO2), acceleration, vibration, acoustics (noise/sound level), people counting (many technologies), CAN data and real-time video can be measured. Outer sensors and peripheral types of equipment can measure weather data (temperature, humidity, air pressure), light (luminosity, sunshine), air quality (CO2, O3, NO2), position (GPS), and road surface (temperature, water layer thickness, black ice).

The bus also has a hybrid vehicle computer with high configuration graphics and multimedia enhancement illustrated in figure 3.2. This minicomputer has a built-in CAN bus that can monitor the vehicles operating real-time information and has the ability to communicate two-way voice connections. Moreover, it can transfer a high amount of data over the air. The bus has a screen that is connected to the computers, which show the information about the journey, the passenger's needs, and bus information. It is also a source of promoting business and income with mobility as a service platform.

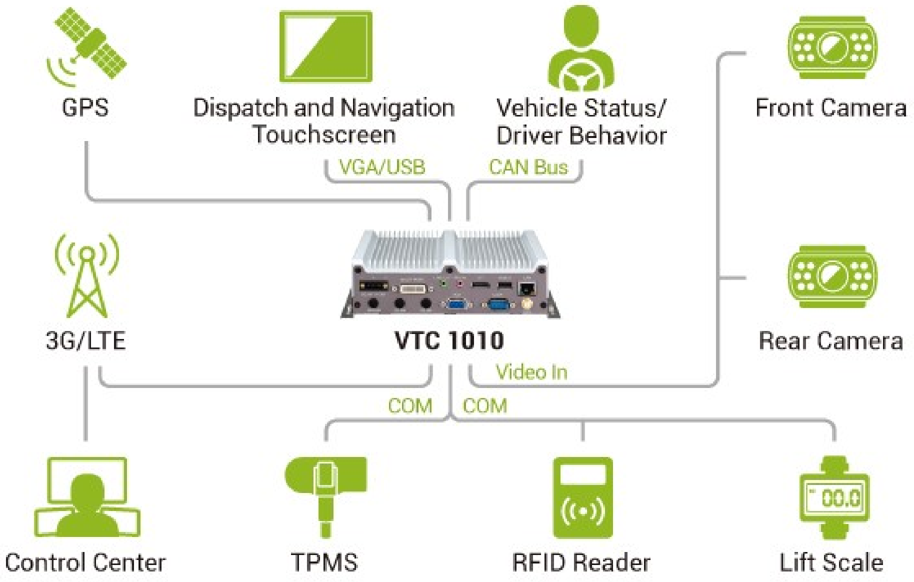


Figure 3.3 VTC 1010 and Connectivity [85]

The bus is also incorporated with a UBLOX GPS, which can measure the location of the bus accurately with a maximum 2-meter error rate. Even the bus stop is covered with Smart sensors, so it becomes easy to view the bus's actual position and shows how far it is from the next stop. It gives the user an estimate to schedule the commute and facilitate the commuter on their daily commute. The bus also has HSL LIJ component that sends vehicle information provided by the service provider.

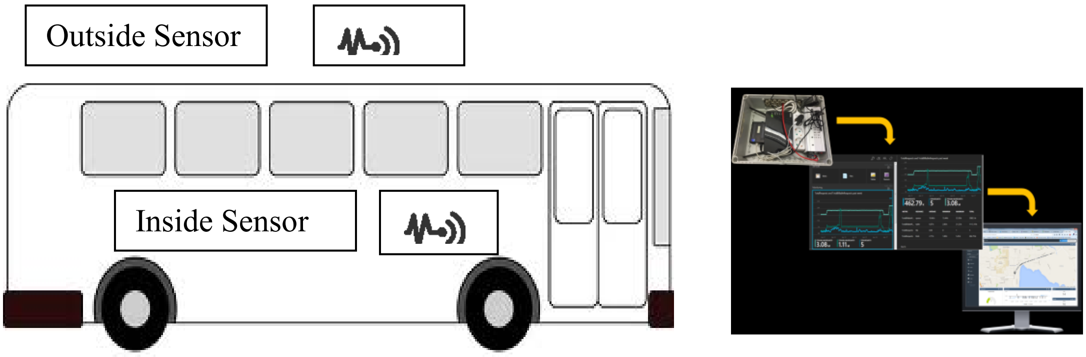


Figure 3.4 LLB platform and dataflow

3.5 Data Collection and Processing Procedure

As illustrated earlier in chapter 3.4, the bus is acting as a part platform and is incorporated with many sensors, cameras, GPS systems, screens, multiband GPS antennae, and a hybrid vehicle computer as illustrated in figure 3.4. Through those sensors and devices, it generates huge amount of data and information. All this information is collected and processed by the hybrid vehicle computer VTC1010, along with the vehicles operating real-time details by built-in CAN bus data.

The buses have LIJ system from HSL, which provides specific real-time information of the vehicle. Both VTC 1010 and LIJ system is connected to the Microsoft Azure cloud system by using MQTT protocol and provide real time information of the system. LLB MQTT and HSL MQTT broker protocol established the connection with the cloud platform and MQTT client. MTQQ clients collect information from both HSL and the LLB sources, merge that data and store it into a data stack. Data collected from the buses through API, are processed and merged for data visualization, and then written into real-time data API in Json format for further processing.

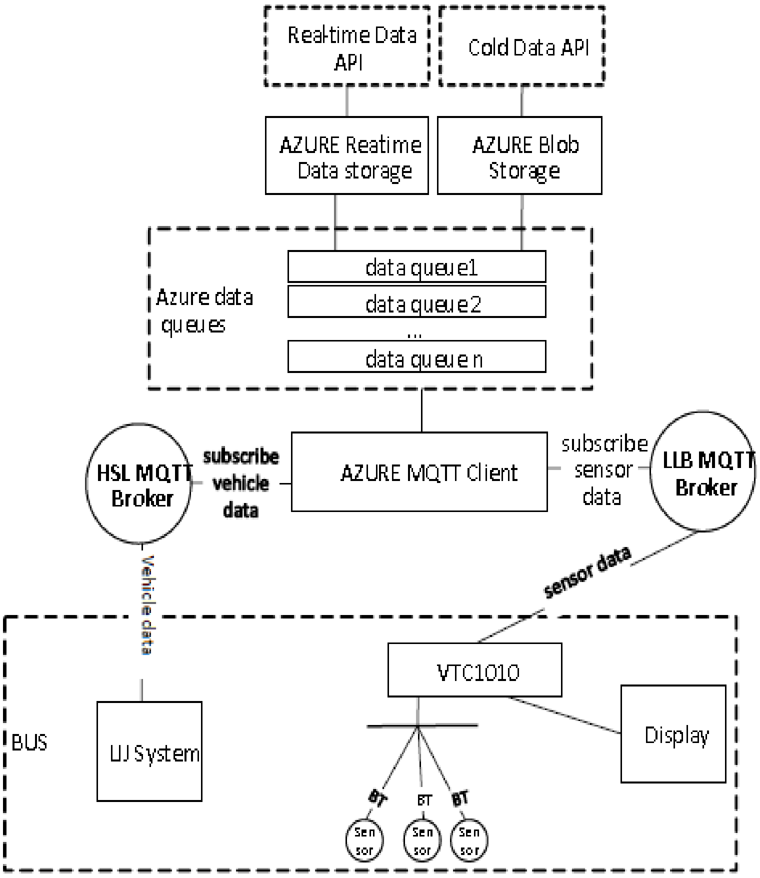
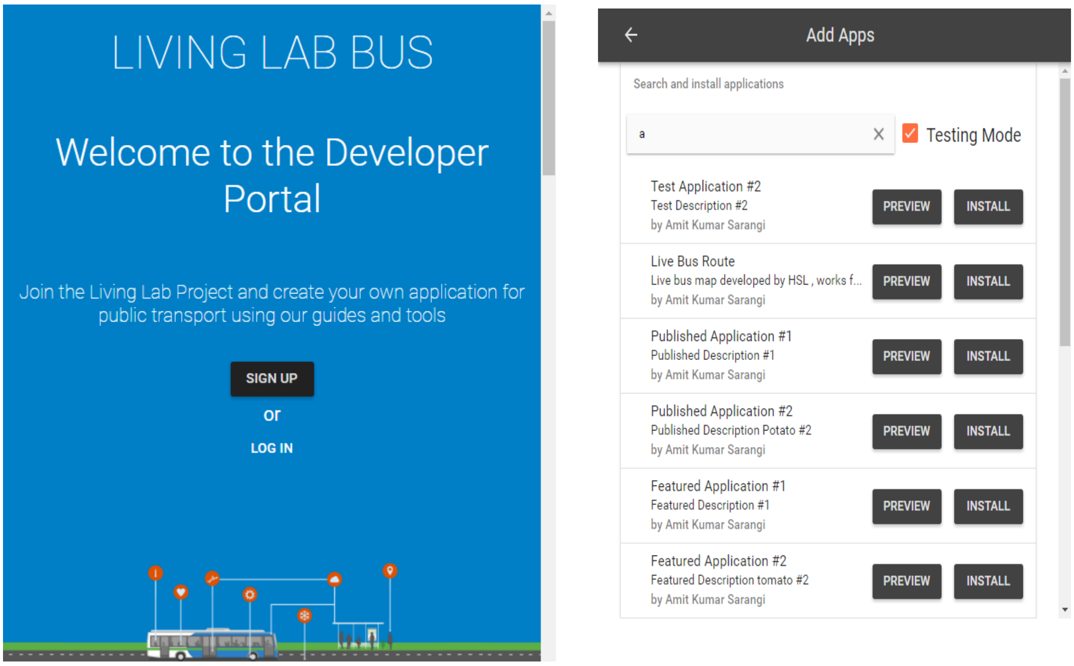


Figure 3.5 Data Collection and Processing in LLB Platform [87]

3.6 LLB Portal and Developer Platform

The LLB developer portals provide the visual application of the LLB ecosystem. One can experience the ecosystem by using the LLB platform and applications. It is an open platform to test new applications and develop new ideas on the mobility of a service as shown in figure 3.5. Service providers have an excellent opportunity to explore the needs of the user and generate their business model around those needs. Service providers can get real-time feedback from the user within a short time and have the information to make improvements.

LLB developer portal is a platform where developers can submit application to solve the issues which are facing by the end users on regular commutes. It provides a platform to registers the application for the end user to use and guide line to develop it. Developers need to register for accessing the portal, can use the SDK which cut out the development time gives a proper structure for the applications. LLB portal giving lots of API support created by the stakeholders to help developers and smoothing the development processes. A lot of existing third-party APIs can be tested over this platform as well. The testing environment provides a great scope to test the application on real environment. After the development, the application need to submit for reviewing and acceptance testing. Once it gets though those process, the application will be published for real using. LLB platform provides detail guideline with hands on practice example to develop application over the platform and focused to keep the develop process as simple as possible. By doing so it provides a scope to the end user, to generate personalized application over the platform according to the needs since the portal is open to develop and extend.



LIVING LAB BUS

Welcome to the Developer Portal

Join the Living Lab Project and create your own application for public transport using our guides and tools

Add Apps

Search and install applications

a

☑ Testing Mode

Test Application #2
Test Description #2
by Amit Kumar Sarangi

Live Bus Route
Live bus map developed by HSL , works f...
by Amit Kumar Sarangi

Published Application #1
Published Description #1
by Amit Kumar Sarangi

Published Application #2
Published Description Potato #2
by Amit Kumar Sarangi

Featured Application #1
Featured Description #1
by Amit Kumar Sarangi

Featured Application #2
Featured Description tomato #2
by Amit Kumar Sarangi

Figure 3.6 LLB Platform’s Developer Portal

4 Methodology

In this chapter we first go over the structure of research, investigation process of DX on LLB platform, followed by User scenario that we collected by researching on user expectation. Then the description of the method that we chose to evaluate the DX.

4.1 Structuring the Research

The central goal of this research was to define the developer's experiences while developing the service for the SC people on their regular commute. According to Fagerholm et al. [36], DX concept is matched with some part of the user experience concept. The modified System Usability Scale (SUS) has been used to measure the developer experience, which provides a straightforward result of seeking the DX measurement. Though the SUS is considered to be a versatile tool for usability professionals to measure usability of the system. In spite of having different methodology to measure DX, a modified version of SUS has been introduced here by believing that the developers are the user of the system development tools, methods and architecture.

The design and development process of the research is based on the Information Systems Research Framework of Hevner et al. [70], illustrated in figure 4.1. The Environment factors is discovered and discussed in chapter 2 and 3. Knowledge Base is described in this chapter 4. In next step, assessing and refining is covered on chapter five. In last step, the observations of the research is presented in chapter 6.

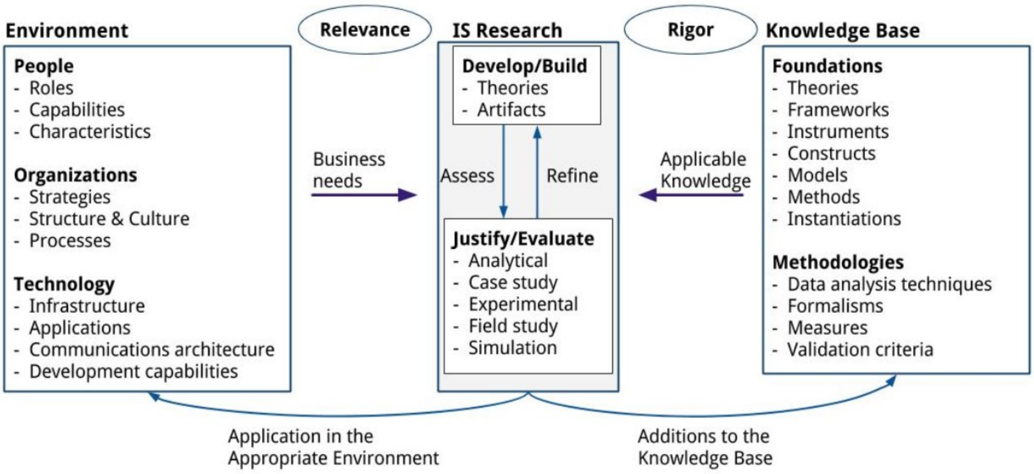


Figure 4.1 Information Systems Research Framework [70]

4.2 Investigation Process of DX

Answering two research questions, “how to measure developer experience on top of the digital service platform” and “how to analyse the developer experience” requires profound understanding on DX, digital platform, and UCD. To gain the understanding a micro-service will be developed on top of a digital service platform by user expectations and analysed developer experiences while developing the service for the LLB ecosystem. Developing system with the involvement of the user, gives the developer scope to achieve developer experience with upstream activities [81].

As discussed in chapter 2.1,2.2,2.3, 2.4 and 2.6, the LLB platform is a digital service platform and It is the part of a SC, Smart transportation, and the LL concept. It has close ties with the user, service providers, user’s consent to the service, and other stakeholders. According to Nielsen&Norman Group [68], system designers are not the user of the system, but in-service development platform developers are also the user of the system development tools for further development of the application for the end user [36]. User-centric design is mainly focused here as it considered the service as the reflection of user expectation. For that reason, the developer experience is measured in a user-centric way. The System Usability Scale (SUS) will be used to measure the developer experiences, as the SUS provides a straightforward result. That is why it is considered a versatile tool for usability professionals, and it can involve a wide range of people including developers as well as the general user who does not have any experience of using the service [76].

The service will be developed on top of a digital service platform of the LLB by using recent web technologies. The LLB platforms, API, and third-party APIs will be connected to make the service more user-centric according to the expectations. The developer experience will be measured, which is like the user experience, by experiencing the differences and effectiveness through the service platform. A user scenario that has been created about the artefact will be developed on top of the LLB platform. The user scenario is given following.

4.3 User Scenario

Mr. Browne is a young professional, working in a bank as an IT specialist. He lives in Tistila, his workplace is in Tapiola, and he takes bus no. 11 for his daily commute. He is using the LLB app and installed it on his mobile device for the ease of daily commute to and from work.

He likes to watch the deals and offers, provided by the business owners according to his preferences and wants to accept reasonable offers on his way home. He buys his bus tickets through the LLB app (through PayIQ). Once he hops on the bus and validates his ticket, the platform identifies him inside the bus, and he gets a notification of sales from Interflora (a flower chain shop providing offers on their flower bouquets and vases) that is in IsoOmena, on his way to home. They are providing a sale price on different flower bouquets. He chose an offer from the given list and paid the bill through the LLB apps. Once he accepted the offer, Interflora is notified that he is on the way to collect it from the store. Interflora has thirteen minutes to ready the bouquet for pick up, as he is thirteen minutes away from the shop on bus number 11. Since he paid the bill through the LLB app, both users receive a receipt and payment code that will enable Mr. Browne to pick up the bouquet quickly without wasting time in queue (both ends are using same payment system PayIQ).

After picking up the bouquet, Mr. Browne can check the schedule for the next bus and any specific info from the apps. As he is carrying the flowers, which are very much temperature sensitive, he wants to keep the bouquet fresh, so he needs specific information about the incoming bus environment. Mr. Browne doesn't want the flowers to freeze in the cold, nor does he want to wait in the bus stop since it is only a 20-second walk to the shop. Therefore, he wants to schedule the commute in a convenient way for him. Again, Mr. Browne wants a free seat available on the bus as he does not want the flower bouquet to get crushed. Overall, he wants a comfortable journey. He checks the information about temperature, humidity, air pressure and available seats on the bus.

The LLB app provides not only a convenient commute for the end user, but also tiding up business provider, and technology providers.

4.4 Developer Experience Measurement by Using Modified SUS

According to ISO 9241-11, the measurement of usability should cover effectiveness, efficiency, and satisfaction (the users' subjective reactions to using the system). In context of developer experience; effectiveness is the involvement of the actors, the community, the work division in developing phase, and the overall output of the system. Efficiency is the number of resources that are needed to perform the specific task with the desired output. Satisfaction is the collective outcome of different factors like the convenient way of using resources, user-centric design with the desired output, and seamlessly fulfilling the need. Although analysing the developer experience is very difficult it also can be misleading the experience result across the system. This does not mean that all the developer experiences need to be same since the developers are doing different tasks and have different skills with varying levels of experience.

From that perspective, SUS is going be used which will give a great foreground to measure the developer experience. The SUS questions are slightly modified to fit with more developer perspectives and would help practitioners to measure the developer experience with the interpretation of the SUS score.

The original SUS have ten statements and these statements are modified to fit with the developer experience while the system is being developed. All the observations have a 5-point scale of strength, with the result that can be a range between 0 and 100. The higher scores indicate a better usability, and lower scores suggest the opposite. The scoring of the survey is tricky because there are both positive and negative statements. The Brooke (1996) [76] method will be used to score the SUS. The SUS score will be used to analyse the developer experience and evaluate it.

The first statement of SUS defines the system is user-friendly and can accomplish the task seamlessly. That is why the user would like to use the system frequently. On the developer's point of view, the system is the LLB platform, and it satisfies the user needs. Developers would like to develop new functionalities and applications onto the service platform that meets the user needs and fulfils their expectations. The statement can be modified in the way "I think I would like to develop the system on top of the service platform frequently."

The second statement of SUS defines the system is unnecessarily complicated to use for the user. Once the system is hard to use, it will lose its acceptability to the user. The user does not like the complicated system, even if the operation is logically or arithmetically correct. From the developer perspective, it is stated that the development of the system on top the existing service platform is exceptionally complicated. The statement can be modified in such a way "I found the development of the system unnecessarily complex on top of the service platform."

The third statement of SUS defines the system as straightforward to use, and following the UCD helps develop it. It can meet the user's expectations. From the developers' perspective, it is shown in a way that the development or enhancement of the system on top of the service platform was simple. The statement can be modified such as "I thought the development of the system was easy on top of the LLB service platform."

The fourth statement states the complexity of the system and the need to get some help from a technical person to be able to use this method. The developer viewpoint can be described as the system is very complicated to develop and that is why developers would need help from external resources. The statement can be modified to "I think that I would need the support from external resources to be able to develop this system."

The fifth statement comments about the integrity of the system with a variety of functionalities. The developer perspective states that the development of various features and integration with the system is accessible on the top service platform. It can work independently as a system or can have a dependency on the service platform. The modified statement would stand as “I found that the development of various functions and integration with the system seamless on the LLB platform.”

The sixth statement states the inconsistency of the system. In the developer’s viewpoint, it can be said that the current system has some differences, which can be the cause of interruptions for further development. If the digital service platform has deviations, it would be difficult to develop any additional construction on top of the platform. The modified statement stands as “I thought that the LLB platform had too much inconsistency and that it is hard to continue further development in this system.”

The seventh statement comment’s about the ease of the system and that it takes a minimum amount of time to learn how to use it. From the developer's point of view, it can be described as a system development learning curve that is so small that most can learn the development of the service platform. The modified statement stands “I would imagine that most people would learn further development on top of this LLB platform very quickly.”

The eighth statement defines the system as cumbersome to use. The word 'cumbersome' is not commonly used. Despite being cumbersome, the word awkward can be used to explain the system by itself. The statement stands like: the system is not very user-friendly, not easy to learn, and does not able to make a connection with the user. As the developers are also the users of the system development tools, it creates some problems with the developers as well. If the system behaves in such a way, it will be difficult for further development on the digital service platform. The statement can be translated in developer’s viewpoint as “I found the development of the system on top of the LLB platform is very awkward.”

The ninth statement states that the system is easy to use, and that the user feels confident, that the method followed UCD, and has a little learning curve as well. The developer viewpoint can be said that the developer has a strong knowledge of the system, sufficient technical knowledge, and the skill to develop the system. Moreover, the service platform must have detailed documentation. The statement can be translated in developer’s viewpoint as “I felt very confident developing the system.”

The tenth statement states the need for specific knowledge and learning required before using the system. It can be said that developers need to learn a prerequisite and should have technical knowledge of the system before starting the development of the system. The modified statement stands “I needed to learn a lot of things before I could get going to develop with this system.”

Original SUS Statements [76]	Modified SUS Statements
I would like to use this system often	I would like to develop the system on top of the service platform frequently
I found the system complex	I found the development of the system is unnecessarily complex on top of the service platform
I found the system was to use	I thought the development of the system was easy on top of the service platform the LLB
I think that I would need the support of a technical person to be able to use this system	I think that I would need the support from external resources to be able to develop this system
I found that the various functions in this system were well integrated	I found that the development of various functions and integration with the system seamless on the LLB platform
I thought that there was too much inconsistency in this system	I thought that the LLB platform has too much inconsistency that it is hard to continue further development in this system
I imagine that most people would learn to use this system very quickly	I would imagine that most people would learn further development on top of this the LLB platform very quickly
I found the system cumbersome to use	I found the development of the system on top of the LLB platform is very awkward
I felt confident using the system	I felt very confident developing the system
I needed to learn a lot of things before I could get going with this system	I needed to learn a lot more before I could start to develop with this system

Table 1 Modified SUS statements on developer context

4.5 DX Measurement Approach Derived from DX Concept

Fagerholm et al. [36], presented a concept of DX by transferring some integral part UX where the end goal is software development. According to them, DX is the combination of positive experience, appropriate use and efficient use of the development tools for developing a product or service. As DX is derived from UX, so both have lots similarities. But in context of the end goal both are completely different where UX focused on using the products or services and DX is focused on Developing.

In digital service development platform, DX can be achieved by following the similar manner as described earlier. So, in service platform view point, it can be said, DX is the summation of the positive experiences achieved by the developer with appropriate, and efficient use of the platform. Conceptually, appropriate use and efficient use refers better understanding of the process of development and product relationship. Simply, it can be said that appropriate and efficient use stands on the functionalities provided by the platform to facilitate the developers for developing the application. Although the developers need to have a clear understanding of the process-product relationships on development process for appropriate use of the functionalities those are being provided.

It is discussed in chapter 2.6, the measurement process of DX is very tricky. Though few measurement processes have been described in practice, but still it has lots scope to research on. A measurement process has been derived here from the conceptual definition of DX [36].

In this measurement method the developers will developer an application for the sake of end user to fulfill their needs by using popular development tools and services provided by the platform. Throughout the development process, the experience of the developer will be mapped over the platform. Then the outcome of the developer journey over the platform will be heuristically evaluated as positive or negative experiences. After that, a math calculation will be done to find out the average of positive experiences achieved over the platform. SUS score scale will be used here and the will SUS score scale will give a qualitative result of DX by applying the value.

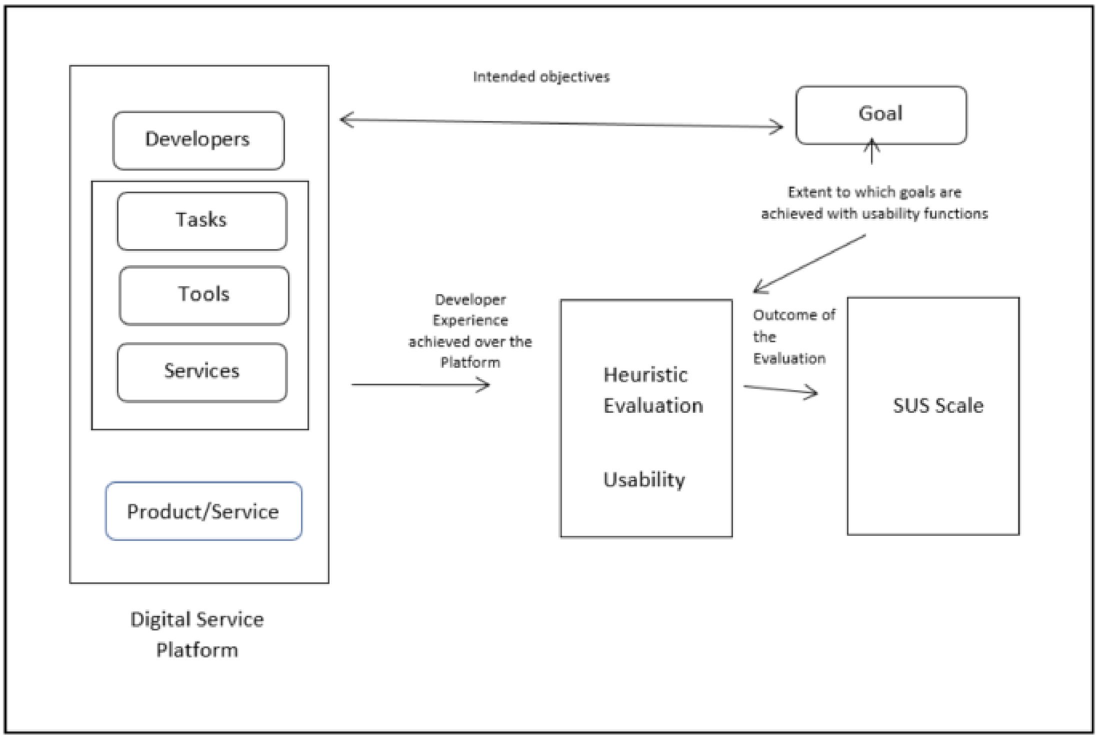

Intended objectives
Goal
Developers
Tasks
Tools
Services
Product/Service
Digital Service Platform
Extent to which goals are achieved with usability functions
Developer Experience achieved over the Platform
Heuristic Evaluation
Usability
Outcome of the Evaluation
SUS Scale

Figure 4.2 DX measurement approach derived from DX concept

5 Results

In this chapter we discussed about the artefact followed by how it matched with user scenario. Then we discuss the measurement process of DX in details with journey map over the platform and SUS scoring reason.

5.1 Artefact Description

The application was developed on top of the LLB platform, as described in the methodology section chapter 4.2, using the LLB SDK, and recent current web technologies. Node JS, Handlebars, and MongoDB were used on the server side, and HTML5, CSS3, and Angular JS were used on the client side. A rest API was developed for the server-side application, and the application is connected with LLB and its stakeholder's APIs.

5.1.1 Frontend Application

In LLB platform user can add their application as an addon. It can be personalized by login and adding preferable addon to the platform. Like all the application, My Deal is an application hosted in LLB platform as an addon. Users need to add the addon on their application interface to view the offered deal by the dealer. When the user adds it, the application asks to input their wish list. The application will save the wish list and ask for integration with the PayIQ application hosted over LLB platform If the user already installed it. Otherwise, the application asks to install PayIQ addon, if the user would like to integrate payIQ payment system with the application.

On LLB platform, user can check the schedule and route information for regular commuting, and buy ticket through PayIQ. Figure 7, and 8 in Appendix A- Application interface illustrated that. Once the user hops on the bus and verifies the ticket, the app identifies the route and shows the available deals according to the user wish list, figure 1,3 in Appendix A- Application interface refers that. The user has option to accept the deal and make payment through PayIQ. Figure 4, and 5 in Appendix A- Application interface illustrated that. The app also shows the available bus to commute and bus specific information along with outside environment information. Figure 8 and 9 in Appendix 1- Application interface refers that.

5.1.2 Backend Application

Advertisers are the business owners who can post advertisements on the system. The advertiser can select the time duration of the commercial, displays of the relevant information, and store the advertisement info into an advertisement collection. Figure 5 in Appendix A- Application interface refers the offer adding process. The advertiser has the authentication to create, modify, and delete operations for the advertisement. After payment for the commercial has been made the administrator can review the ad before approving it and set up the payment section, PayIQ, which is the payment API uses. Once the super administrator has authorized the advertisement and payment platform, it will be shown on the application for a specific time duration that was selected by the advertiser.

5.1.3 Matching with User Scenario

If the application work flow matched with the user scenario from methodology section chapter 4.3, the application is running in a similar manner and meets user expectation. Walking through the user scenario and the app itself, it can be said it solves the major functionality that the user wanted like; the users can check the schedule and route for commuting which HSL provides relative information, buy the product on the way that they wanted, can pick up easily from the shop, take another bus and check bus information along with environmental info.

Once the user reviews the data for the commute and selects the route, they can buy the preferred ticket through. PayIQ provides the flexibility to buy the ticket on the application for the users. After purchasing the ticket users get on the bus and validate the ticket. While the user verifies it, the application identifies the user inside the bus and recognizes the route of the journey through HSL. Instantly the user gets a targeted advertisement on the course of the trip according to the provided preference list. The user can set a preference list while registering to the system and the application sorts the advertisements according to it. Users can decide to accept an offer and buy the product through the application. PayIQ provides a convenient way to make the payment online and keep the process simple. Once the user chooses to buy the product and has paid online, both parties get a notification and payment receipt from the application. The user leaves at the bus stop to collect the goods that are secured from the advertiser. The shops also have some time to get the product available for a quick pick up. After picking up the product, the user can take another bus from the bus stop without wasting time. The application provides the schedule for the next bus and specific information. The VTT API is providing all the bus information along with environment info. As VTT API is not ready yet, raw data and replication of VTT API have been used in the application.

5.2 Measuring Developer Experience

Measuring developer experience is quite hard as it is easy to get different results from one developer to another. However, this experience firmly depends on technical knowledge and experiences. Sometimes these differences can mislead the whole measurement system. Despite these complexity, a measurement has done on usability and the developer experience.

SUS is one of common tool which is used to measure usability and renowned for its accuracy and simplicity [76]. In spite of having different way of measuring developer experience described in chapter 2.6, a modified SUS is introduced. As discussed in chapter 4.3, the modified SUS gives a strong background to measure the developer experience and the question has been modified to fit with the developer perspective, which will help to measure the developer experience.

The SUS statements will give the numerical score which is going to use as a benchmark for measuring the developer experience. It provides ten questions, having a response system with a scale of 0 to 5 points. The SUS is considered to be extremely reliable in usability measurement with an overall usability score for any application or product within a scale of 0 to 100. In this research, each statement has been gone through, analysed, and given a rating to measure the developer experience.

5.2.1 Experience Achieved on LLB developer portal

Developers are tending to use any digital service platform depends on different factors. In between those the most important factor is that how easy to use the platform. The registration process need to be easy and precise. If the registration process takes longer period and the process is hard, then there are strong possibilities to switch the platform for the user. In case of LLB platform, registration process in developer portal is easy and it does not take more that more than one minute. Developers can register directly by giving user name, e-mail and password. Along with that, registration can be made with google and Facebook id.

Development stack must not take longer period to install and learning for further development. Most important is that, it must be recent popular technology oriented and It would be nice if developers are free to choose any development stack on their choice. LLB platform using popular technology NodeJS and has given clear direction for setting up the development stack. The setup process of the development stack and learning curve to develop application over the stack does not take longer period and eventually it can be said it was easy. Moreover, LLB platform is not reluctant to use any web 2.0 technologies over it which gives a lot of flexibility to the developers.

The workflow and the visual display of the platform need be easy for user acceptance. If it creates confusion and hard to use, then users could have switch another platform as it loses acceptability to the user. UCD and usability makes difference here. Based on experience over the LLB development platform, it can be said that the workflow of the platform is very simple and easily understandable.

Regarding to use the LLB platform as a digital service platform, it is practical to develop new services with the LLB provided SDK toolkit for further development. The LLB SDK affords a precise folder structure and coding pattern with clear direction for the construction. Above all, SDK cut out the development time as it provides a get going option for developing the application over the platform. Developers are creating the application by using this development kit and place all files inside the relevant directories. SDK provides the user the necessary flexibility to use the LLB APIs and functionalities, or use other third-party APIs, instead of writing their own codes from scratch. The developers can have access to location, notification, and exit option in the LLB SDK. The developer can also get access to the real device location of the user if the user has given the appropriate permissions using the Location API shown in figure 5.1. The necessary code for gaining access to a Location is -

```
llb_app.request('location')
llb_app.addListener('location',
function(result){
     if(result.status == 'success')
     {
          console.log(result.data)
     }
})
```

Figure 5.1 LLB Helper Function

For preceding the development task and integration with the LLB platform APIs and other APIs provided by the stakeholders were smooth. Developer portal has given clear documentation to integrate with, and each stakeholder has provided the same. But the direction is not all together in developer portal, it is scattered, and developers needed to search and study each stakeholder provided document on different web addresses. Though the documentation is scattered and needed lot of efforts to achieve, but following the documentation, integration with the LLB platform and development over it makes it relatively easy. Integration with APIs using the LLB SDK becomes easy by writing basic integration code illustrated in figure 5.2. The connection with the LLB API can be produced with the following code.

```
llb_app.fetch = function(url) {
     base_url='https://llb.sis.uta.fi/api/v1/httpsprox
     y/'
     return fetch(base_url+url)
          .then(res => res.json())
          .then(res => {
               if(res.code == 200) {
                    return
                    Promise.resolve(res.data.data);
               }
               else {
                    console.warn(res.details);
                     return
                     Promise.reject(res.details);
               }
          })
}
```

Figure 5.2 LLB API Integration

Integration with HSL API has been made with the same way by following code. Figure 5.3 illustrated the integration process HSL API.

```
llb_app.fetch = function(url) {
     base_url='https://api.digitransit.fi/routing/v1/r
outers/finland/index/graphql'
          return fetch(base_url+url)
          .then(res => res.json())
          .then(res => {
               if(res.code == 200) {
                    return
                    Promise.resolve(res.data.data);
               }
               else {
                    console.warn(res.details);
                    return
                    Promise.reject(res.details);
               }
          })
}
```

Figure 5.3 HSL API Integration

Development an application on top of the LLB and deployment to the developer portal was easy. It can be said developer with less knowledge on web technologies can develop an application according to their needs by following the documentation. Even a few examples have given on the development portal for making the learning process ease on top of the LLB platform by using its SDK. Developers need to open a developer account on the LLB service platform and submit the application to the platform for the review.

An administrator will verify and validate the application and approve it for the user. After the administrator has approved it, the application is open for the user to use in real life.

LLB platform providing rest API, end points of the APIs are well defined, and the data format of those services are Json which is recent popular and mostly used. According the experience of using the LLB platform, it can be said the API service was satisfactory.

Response time is very important for an application. LLB API, HSL API and PayIQ API was used for developing the application on top of the platform. Though response time for each API was different, but it was fast and efficient. HSL API response time was very fast, as it takes couple of get and post request but surprisingly it responded within 300ms. LLB API provided by VTT still in development phase, and it updates real the information on each second. The response time of LLB API was 1.36s. PayIQ API was used for making payment, it was a bit time consuming than other two APIs. The overall application response time is 1.50s which is fast. According to [79], if the application takes more than four to six second as response time, there strong possibilities to switch the application for the user. In case of API service, developers are inclining to response in same way if the API taking longer period to response.

LLB developer portal provides test environment to test applications for the developer. Through it, developers are getting scope to test the application in real environment. Though LLB tried to provide the testing environment but it must be extended. A sandbox would help developers to test the application over the platform where recent testing environment is too limited.

The service need to be profoundly describe in development portal. It can be said, LLB portal have a clear description of the available services, but the catch is that there is no clear direction how to use those services. Developers need to find those service and using process from stakeholders provided portal. In case of our application development process we found the services from the development portal but for using procedure and other direction we had to search from stakeholder's portal which was time consuming. LLB portal can easily eliminate this problem by grouping together all available services.

The documentation of the LLB platform is rich enough to develop application over it but still it is in the development phase and scattered. The stakeholders are providing services through the platform but did not integrate the documentation on the portal. Developers need to search for documentations those are provided by the stakeholders on different development portal. It is understandable that each stakeholder is published individual services and those are platform independent. It would be helpful for the developers, if the platform group together all the available documentation in development portal.

Though LLB portal has different API services, but it does not have any API catalogue. If we relate it with the service description, it can be said that services need to be group together as LLB portal proving services through APIs. It would be helpful for the user to find available services and chose the best API according to it is provided services.

LLB portal provided few examples of using the portal and development. We would say those sample more directed to use of SDK not using the APIs. There is no direction of using service provided by the APIs in the portal. Though HSL and PayIQ provided details guidance of using APIs but those are not included to the development portal. As LLB providing those services through the portal, it could be helpful for the developers to get all the information inside the portal along with coding samples, snippets and proper documentation of using those services.

User support system plays a very important role in usability. LLB developer portal ensured to provide user support on request. User's need to write the problems to the LLB support system which they are facing to develop application over LLB platform. According to our experience the support service was very fast, and it is very helpful for further development of the application.

The error reporting service of the platform is up to the mark and the structure of the error reporting feedback is clear and effective to solve the issues. Eventually, the error message was self-explanatory which would help the developers to solve the problems.

5.2.2 The SUS Statements and Assessment

The first statement of the SUS states, "I think I would like to develop the system on top of the service platform frequently." If this statement is transformed for developers, then it stands "developers are interested in developing current and any forthcoming applications on the LLB service platform." Though the developer could have chosen another platform since it entirely depends on the developer's interest and the technology demand of the system. If the platform provides flexibility and guidance to develop an application that meets the expectation of the developer, then developers tend to choose that platform for development. The LLB platform offers a flexible environment to develop the app with proper guidance. People with less knowledge in web technology can develop an application over the LLB platform, and that is why it is marked as strongly agreed with the statement.

The second statement of the SUS states "I found the development of the system is unnecessarily complex on top of the service platform." The transformation of the statement in respect to developer's perspective refers to the complexity involved in the development of an application using the LLB platform. However, developers are not entirely satisfied with it because the platform does not provide enough documentation. The documentation of the LLB platform is not adequate to carry out the complex function, whereas it is good to perform a small task, but the user's expectation has no limits. Developers need to study the codes of the platform to figure out the development instead of following the documentation. The developer portal provides few example projects, but those projects are mainly focused on SDK. It would have been bringing better experience for the user, if the platform provides few examples of using LLB APIs. It would also helpful for increasing the confidence of the developers for further development on the platform. By analyzing those issues, it has been marked as agreed, but still LLB platform decreased complexity of the development by using the LLB platform.

The third statement of the SUS is "I thought the development of the system was easy on top of the service platform the LLB" which can be understood as "Developers found the development of the LLB platform is relatively easy compared to any other platform." It can be said the third statement is quite the opposite of the second statement. The LLB ecosystem provides robust development environment with guidance, excellent support system, and providing test environment for the developers to test an application over it. The direction is not adequate, but the LLB SDK provides a bare bones structure and few helper functions that will reduce the development time and effort for the developer. The work flow of the platform is very straight forward which make the development very easy. The Assessment scale elaborates more, and the developer marked it as agree with the third SUS statement.

The fourth statement "I think that I would need the support from external resources to be able to develop this system," suggests the developer might need external support for further development of the system. This external support may be proper documentation and support service of the platform or any experienced developer support to carry out the development procedure. Though this statement declares both documentation and experienced developer support as an external support, the assumption is that following written procedures and getting help from the platform support system is natural for developers instead of getting help from experienced developers. Moreover, dedicated community for the platform could help the developer, but the platform is in developing phase that is why community helps is unavailable here recently. It is mentioned earlier that the documentation of the platform is not adequately maintained and is scattered. Also, the LLB platform itself uses different third-party APIs, so proper documentation plays a vital role here. Moreover, each stakeholder developed a different API instead of merging and generating a single API, which increases the confusion more. It is mentioned in [80], a API catalogue and proper service description can help the developers for finding the resource easily, but in recent phase those are missing. As a result, developers need to go through the entire supporting APIs from different resource by searching and the LLB documentation. So, the documentation is quite hard to follow, but it can be said there does not require any experienced developer help.

The fifth statement "I found that the development of functionalities and integration with the system seamless on the LLB platform" suggests that this platform is easy to learn, flexible, and to carry out further development is simple. It can be said that the platform functionality and integration with the existing module is also relatively easy to develop. As stated in earlier statements, LLB platform provides an easy way to integrate with the LLB platform APIs and other APIs provided by the stakeholders. The system has given documentation to integrate with, and each stakeholder has provided the same. By following the documentation, integration with the LLB platform and development over it makes it relatively easy. Integration with APIs using the LLB SDK becomes very easy by writing few basic integration codes. Along with integration it gives a few interactive functionalities with makes the development work smooth. Even the work flow and application integration with the platform is very easy. Developers just need to open a developer account on the service platform and submit the application to the platform for the review. An administrator will verify and validate the application and approve it for the user. The same procedure has been followed here according to the guidelines, used to develop an application on top of the LLB SDK, and submitted for verification. After the administrator has approved it, the application is open for the user to use in real life. The Assessment scale elaborates it more intensely and the developer agrees with the fifth SUS statement.

The sixth statement “I thought that the LLB platform has too much inconsistency, that it is hard to continue further development in this system.” This statement says that the LLB platform creates a lot of barriers that hindering to be carrying out the development of the system. It provides a clear workflow to continue the development. Since the platform is not yet mature, but further development on top of the LLB platform is not that hard. In earlier statements, it has been shown that the integration was smooth with APIs and helper functions. According to this experience of integration, it can be said that it was fluid and effortless. The developer disagrees with the sixth SUS statement by analyzing those factors. Moreover, the LLB platform offers flexibility to add on different third-party APIs. So, it can be said further development of the LLB platform is not difficult.

The seventh statement “I would imagine that most people would learn further development on top of this the LLB platform very quickly,” suggests the ease of the platform for the developers. As it is stated in earlier, developer portal providing details guidance for development the application over it. Though the guidance is scattered, hard top follow and need efforts to find it. if the developers are able to follow the guidance from difference resources, then it can be said it is not difficult to make further development. LLB portal providing few samples for the sake of developer which is also useful for further development. Moreover, developers can get community help from LLB stakeholders while using stakeholder’s APIs. As it is stated earlier that LLB stakeholders have bigger community portal to help the developer while LLB platform still have lacking. Apart from these difficulties, the developers are strongly agreed with the statement which states learning for further development on the LLB platform was comfortable, which is similar to the reasoning of statement three, which is opposite of statement two and six.

The eighth statement “I found the development of the system on top of the LLB platform very awkward.” This comment is the combination of the second and sixth statement, but the Assessment score indicates a substantial difference as a result. The Assessment score shows a strong disagreement with the account that means its opposite to the statements. The transformation of the description states that the development on the LLB platform was relatively clear, which is similar to the third statement. It is already illustrated that the platform is easy to learn and made the developer's life easy by providing tools for further development. Therefore, it can be matched with the comments that are made on statements number three and five.

The ninth statement “I felt very confident developing the system,” implies the comfort level of the developer for developing the application on top of the LLB platform. It can be said that the ninth statement is the combination of the first, third, fifth and seventh remarks. It is already stated earlier that the platform is easy to learn and by providing tools for further development made the developers lives easy. As well, in the previous statements the construction was easy on top of the service platform by using development SDK, and the integration of the third-party APIs was smooth. The documentation for developers and designers are separate and provide details guidance which increases the confidence level of the developers. Most importantly LLB platform developed trust with the developer by providing secure system. It also secured the data transection with apps by securing the APIs. Apart from those, LLB provides user support system and user can get help within really short time. Moreover, the LLB platform API presents a lot of helper functions like the location of the user and notification, which were very useful for this application. The Assessment score shows a substantial agreement with the report, which means the developer’s comforts are ensured by the platform, the reason of the agreement, and is illustrated in earlier statements.

The tenth statement “I needed to learn a lot of things before I could start to develop with this system,” which states developers need to learn a lot of technologies to develop the application on top of the LLB platform. According to the experience of using this platform it can be said, people with less knowledge on web technologies could develop an application on the platform. It is already mentioned in previous statements that the developer will be able to create applications on the LLB system by following the documentation. Most of the APIs do have the proper documentation, and the users need to learn those APIs by going through the documentation. Along with that most of the stakeholders have bigger community to help the developer to solve arising questions. If the platform was compact with API catalogue with all third-party services API with the proper documentation, it would be much easier for the developer to develop any application over it. Though there have been some issues, but still the SUS assessment score marked as disagreement that means developers do not need to learn a lot of things to get started with the system.

Statement number	Modified SUS Statements	Assessment on SUS Scale
1	I think I would like to develop the system on top of the service platform frequently	5
2	I found the development of the system is unnecessarily complex on top of the service platform	2
3	I thought the development of the system was easy on top of the service platform the LLB	5
4	I think that I will need the support from external resources to be able to develop this system	1
5	I found that the development of functionalities and integration with the system seamless on the LLB platform	4
6	I thought that the LLB platform has too many inconsistencies and that it is really hard to continue further development in this system	2
7	I would imagine that most people would learn further development on top of this the LLB platform very quickly	5
8	I found the development of the system on top of the LLB platform is very awkward	1
9	I felt very confident developing the system	5
10	I needed to learn a lot of things before I could get going to develop with this system	2

Table 2 Modified SUS Statements and Assessment Scale

By analysing those statements and accessing the score is given by each statement, it can be calculated the total assessment score of the SUS. The total score is a numerical representation which refer the developer experience. The total score of the SUS statements is 90.

6 Discussion

In this chapter we first present the observations we made during the research and then detail the actions are recommend for LLB to take, and then discuss the limitations of the research and contemplate future work.

6.1 Observations

This research is aimed to answer the questions: “How to measure the developer experience on top of the digital service platform” and “How to analyse the developer experience”. The research is formulated by following the Information Systems Research Framework of Hevner et al. [70]. The whole research based on finding the best possible way to analyse developer experience over digital platform. The observation of the research which are achieved by introducing the modified SUS have gone through in this section.

Regarding DX and software development process the following items were realised and considered:

1. Few meeting with the stakeholders have been taken place
2. Hand on exercise of the platform being introduced
3. Communication between teams, stakeholders and organization levels need to be improved
4. The chosen methods and tools need to evaluate which can improve platform’s interactivity
5. Sharing experience between teams are sort be the best activity
6. Integration should be start with small and unambiguous application with the system

6.2 SUS score scale

The SUS score scale is more like a typical school grading system which means the statements provides the numerical value that can compare with the SUS scale. In figure 6.1 SUS score scale is illustrated. The SUS scores below 50 are not acceptable and the range between 50 and 70 is marginal with categories high and low. Scores between 70 to upper 80 is sufficient. A truly superior rating is better than 90. Scores less than 70 considered as a candidate for extended analysis and continued improvement needed over it. Moreover, it should be judged to be passable at best. With the adjective rating scales, it can also make a further contrast in the marginal scores, by dividing them into "low marginal" and "high marginal."

The figure below illustrates the comparison of acceptability score, quartile ranges, and the adjective rating scale.

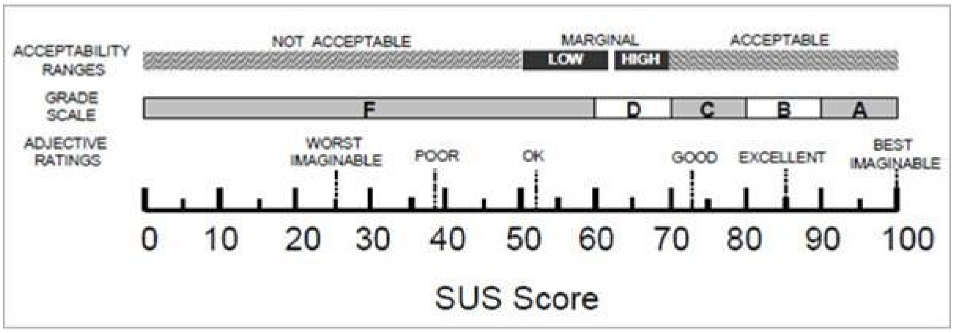


Figure 6.1 SUS Score Scale [76]

It is necessary to test enough participants to avoid misinterpreting the SUS data because of an insufficient sample size. Any interpretation of a SUS score also needs to consider that in the range of scores is half of the nominal value. Thus, a score of 50 does not represent a product that is "half as good" as a product that scores 100 but instead is likely an indication of a grave usability failure for that product.

As there have individual success metrics with associated SUS scores, anecdotal evidence with a participant's SUS score can be an exclusive performance with the platform. It is noteworthy that the limitation of this is entirely on the participant who showed to perform well on tasks, but did not that make a difference on the SUS score. Even developer experience and technological knowledge have a deep-seated effect on the score. In these cases, the SUS scores may very well be inflated, representing perceived success on the part of the user even though they failed.

Our SUS score is 90 and according to the SUS scale it is a superior experience to develop the application on the LLB platform but as we stated earlier it is showing individual success metrics which is entirely depends on the individual experience. Though the result is promising, the score may be varied depending on the developer and knowledge in related fields. Even if the number of developers increases, the effect can be altered as well.

6.3 Validation SUS Score

Validation of our proposed method is the most challenging part due to lack of papers focusing on developer experience measurement, but few proposed methods has been discussed in chapter 2.6. To describe the measurement process Pamela Fox [73], proposed a straight forward approach by analysing developers journey map throughout the development work. Mike Brevoot [88], proposal based on journey map, countable UX tool and introduce Kano model to evaluate it. Roonie [71] and Fagerholm et al. [36], proposed DX could be measured by using UX measurement tools. According to them DX is the cumulative sum of experience, appropriate use and efficient use of the system tools.

In chapter 5.2.2, we achieved DX measurement quantitative result from introduced SUS method over LLB platform. The measurement score scale has been elaborated in chapter 6.2 and achieved a qualitative result. To validate the result achieved from introduced SUS method is compared with a method which is derived from Fagerholm et al. [36] conceptual definition of DX described in chapter 4.5. The validation process combines the measurement of DX over the platform by following the derived method from Fagerholm et al. concept, then comparing the result with proposed modified SUS result.

In chapter 4.5, a method of measuring DX has been described which is derived from Fagerholm et al. [36] proposed DX conceptual definition. In chapter 5.2.1, the journey map has been evaluated and found some positive experience along with some negative. A heuristic evaluation has been done on those experiences achieved through out the journey of the application development over the platform. Then a simple math calculation has been done over the experience list to find out the average of positive experience and employ the score to SUS score scale by converting it to decimal number which gives qualitative result. The result showed 73.40 percent and by applying it to SUS score scale a qualitative result has been found as sufficient. The result indicates that the platform needs to improve a lot for providing excellent DX to the developer over the platform.

On the other hand, in chapter 5.2.2 our proposed method showed that the experience over LLB platform was delightful and excellent which is completely different from the derived method from Fagerholm et al. [36] proposed DX conceptual definition. The difference between these two results is relatively big. Though both methods have been analysed over individual success matrix which could be a cause of misleading. On the other hand, it can be said, the introduced SUS method is not able to measure the exact DX in all aspect of digital service platform. But the derived method from Fagerholm et al. [36] proposed DX conceptual definition [36] is not a proven method that is why the result of that method is arguable. Moreover, the application was developed over the platform was simple and small. if the application were complex, it could have altered the result.

6.4 Remarks and Recommendation for LLB Platform

Though our introduced method result showed best experience achieved by the developer, but still the platform has some deficiencies. According to the journey map in chapter 5.2, it can be said that developing applications and integration was seamless and effortless over the LLB platform, but the platform did not clear example or direction for doing that. Example of using LLB APIs and some code snippets could be helpful for the developers.

Although the documentation of the platform is rich, but it is not organised and scattered. Developers need to search on different stakeholder's platform for seeking help which is time worthy and cause of healthy learning curve. Developers time is limited and if the platform stretched it too thin, then developers could have switched to another platform. So, for offering great DX, platform could offer an organised and centralized documentation. If the platform does so, then it would be more helpful for the developers. At the same way centralized support system over the whole ecosystem ensuring stakeholders involvement could be also helpful for the developers.

It has been discussed earlier in chapter 2.6, DX is not only depending on functionality, usability and visual appearance of the platform. The platform has to create emotional attachment with the developer, otherwise it will be hard to achieve DX. For that reason and get grip on developer expectation for the platform, platform needs to conduct regular survey on developer's journey map. Watch, ask listen formula could works well here. If the platform does so, then it could ensure delightful DX for the developers.

Although DX is not a feature of the platform, it can only achieve by establishing emotional attachment and continuous interaction with the developers. As described in chapter 2.6, The designer of the platform need to understand that all the developers are not homogeneous. Though developers share same DNA, but skill set differs and have different expectation from the platform. To make a generalized platform, designers of the platform need to analyze different developer's persona. If may be varied like some have 25 years of experience developing backend application by using C++ or someone have some technical knowledge love to code in JavaScript. By doing that it can be ensured that the application replicates all developer's expectation with simplistic manner.

Above all, the platform need to use its own APIs, need to provide examples of using it, code snippets, and introducing learning platform by using it's given snippets can be nice move to get closer to the developers. As LLB only providing SDK and example of SDK using right now. By introducing discussed functionalities and facilities could help the platform to engage with the developers.

Recommendation for LLB platform

1. Developer persona need to create
2. Application need to be more simple
3. Eat your own dog food
4. Conduct regular survey on developers' journey over the Ecosystem
5. API catalogue and Documentations need to improve and group together
6. Support system could to be more ecosystem oriented and centralised

6.5 Revisiting the Research Questions

The following research questions are addressed in this paper in chapter 1, and we proceeded the research to find out the answer of those questions.

"How to measure developer experience on top of digital service platform"

To measure developer experience, a detailed analysis has been done over LLB platform and it is discussed in chapter 3, LLB platform is digital service platform. The measurement process required profound understanding on DX, digital platform and UCD. To achieve the knowledge a detail analysis has been done in chapter 2. As DX is inspired by the UX practice and it is recognised that developers are the user of the system development tools [72], so for getting DX over the platform, a micro-service has been developed on top of a digital service platform. It has been sorted out that developing application over the platform is the most appropriate way to engage with the platform [73].

To develop the micro-service over the platform, at first the most important user expectations were determined and discovered which service aspects its related. Then tasks have been grouped together and have taken stakeholders concern. After then, few meeting with the stakeholders had been taken place to know the services are being provided by themselves and familiarised ourselves with the technology. Finally, user scenario has been written other goal of the service.

Throughout the development process over the platform, few questions were answered, and developers' reaction has been recorded over the journey map. The interest of developing service over the platform is measured, recorded the feeling of the developer while development process has been carried out, outlooks, motivation, and perception have been taken account while the distribution or integration has been made with the ecosystem. It has been tried to point out the trust, project awareness, goal-oriented achievement and over all the negative impact while developers improve project environments.

"How to analyse the developer experience "

In chapter 4.3, it is already stated that measuring and analysing DX over any system is very complicated. The result of this analysis differs between person to person. Though it is considered that the result depends on developer's personal skills, technical knowledge and experience level, but the result could be varied in between experienced developer also.

To analyse the developer experience, a method has been introduced by modifying SUS which provides both quantitative and qualitative results. The SUS provides a straightforward result [76] and it is considered a versatile tool for usability professionals. It can involve a wide range of people including developers as well as the general user who does not have any experience of using the services [77].

A modified list of SUS statement has been created suits with developer perspective, which gave quantitative values of the experiences over the LLB platform. By comparing the total given value by the SUS statements with the SUS scale, the developer experience can be analysed. There have been asked 10 questions and got answers for those based on the experiences over the LLB platform, then analysed the answers, gave score and assessed with SUS scale. As discussed in chapter 2, UX completely depends on user's perception, feeling, and response over the system, on the same way DX depends on developer's perception, feelings, responses, and overall reflection of the system.

It can be said that analysing individual developer experience is very hard as it could be changed over each epoch and transection of development phase. Therefore, analysing a group of developer's experiences could be both cumulative and episodic experience over the platform. Moreover, psychological, social, and motivational factors can influence developer experience. It affects both on individual and team performance.

6.6 Limitation and Weaknesses of the Research

One of the limitations of this study is that the result shown here is based on individual success metrics which entirely depends on the individual experience. Though the result is promising, but the score may be varied depending on the developer knowledge and skill in related fields. Even, if the number of developers increases, the effect can be altered.

Lacking's of time striving us to prepare user scenario by analysing fewer user persona and limited resource access forced us to develop small scale application over the platform. Scaling the application on bigger perspective could have lead us in different result.

Although the SUS score shows satisfactory results, developers may be still facing challenges while developing systems on the platform because LLB platform requires a healthy learning curve of different API usage for non-technical person. The stakeholders of LLB platform are working together to achieve a common goal, and striving to ensure the best DX over the platform. As it is considered DX is most important factor for any digital service platform.

6.7 Future Work

In future work, the research period should be long enough that it could give enough time to be implemented an extended version of the application with team oriented environment and make an observation how the induced methods perform and fits in large group of developers.

Finally, in any future work, it would be interesting to analyse DX over complete version of LLB platform with real user interaction.

7 Conclusion

In this research, a comprehensive study has been done on the digital service platform LLB, and analyse developer experience. As discussed in chapter 1, measuring developer experience is not an easy task, as it varies person-to-person, experience level, and knowledge. The measurement process can be conducted by employing UX and usability tools [36, 71, 88]. The modified version of SUS has been used as stated in chapter 1, for measuring developer experiences over the LLB platform. It gives a concrete measurement of DX, but on comparing with a method derived from Fagerholm et al. [36] concept showed same answer. Though it is arguable that the result could be varied on stablished and more trusted platform. Conducting survey with the involvement of multiple developers could also alter the result. The method presented both qualitative and quantitative results and established a baseline for the developers with acceptability scores for further development on top of the LLB platform.

The research processes have been approached by raising two research questions stated in chapter 1, and preceded to find the answer to those questions. The finding of those question provides a baseline of DX measurement and analysis process on digital platform. It also provides a way to verify the result. According to the results obtained from the SUS scale, the developer experience on the LLB platform can be illustrated as "superior". The results thus obtained has shown some discrepancies, but it is understandable since the LLB platform is still in the development phase and the measurement parameter depends on the scale of the application as well. Even, the fact that testing of the application was done with raw data which could have been the reason to these discrepancies. The result finding procedure also comes out with recommendations which could be helpful for the platform developers to increase the developer experience of the LLB platform. As discussed chapter 2, the LLB platform is part of the smart transportation system, which is solely devoted to upgrading the user's lifestyle in the transportation sector and this sector is an integral part of SC concept. Therefore, it can be said this research encouraged developers to develop the application in favour of SC concepts which will create an impact on people's regular lives by mitigating the recent problems. Furthermore, it can be said DX is not a feature, it need to be achieved by the platform. By engaging with the developers, showing empathy and giving the power to develop application over the platform can be a way to achieve that. This research opening the door for the LLB designer to improve the platform, so it could provide a delightful DX to all its developers.

References

1. N. A. Khan, V. Brujic-Okretic, and S. Khaddaj, “Intelligent Integration Framework for Smart Transport System,” 2016 12th International Conference on Intelligent Environments (IE), 2016.

2. G. Maria, E. Baccaglini, D. Brevi, M. Gavelli, and R. Scopigno, “A drone-based image processing system for car detection in a smart transport infrastructure,” 2016 18th Mediterranean Electrotechnical Conference (MELECON), 2016.

3. A. Janota and J. Spalek, “Where is the smart transport going?” 2016 IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (SAMI), 2016.

4. R. Marrero-Almonte, E. Marin-Tordera, X. Masip-Bruin, R. Nuez-Baldoma, J. Batlle-Ferrer, and G. Ren, “A smart drive to future transport systems,” 2014 International Conference on Connected Vehicles and Expo (ICCVE), 2014.

5. S. Wibowo and S. Grandhi, “A multicriteria analysis approach for benchmarking smart transport cities,” 2015 Science and Information Conference (SAI), 2015.

6. D. Advani, “How Melbourne Is Fast-Tracking Smart Transit Using Big Data,” The Huffington Post. [Online]. Available: http://www.huffingtonpost.com/deepak-advani/ibm-how-melbourne-is-fast-_b_4178465.html. [Accessed: 10-Jan-2017].

7. A. A. C. CSAIL, “MIT Big Data Initiative launches transportation challenge, privacy working group,” MIT News, 12-Nov-2013. [Online]. Available: http://news.mit.edu/2013/mit-big-data-initiative-launches-transportation-challenge-privacy-working-group-at-white-house-event. [Accessed: 10-Jan-2017].

8. Baek, Joon Sang, Ezio Manzini, and Francesca Rizzo. "Sustainable collaborative services on the digital platform: Definition and application." DESIGN RESEARCH SOCIETY INTERNACIONAL CONFERENCE, Montreal. 2010.

9. Gage, Benjamin C. "Real time design, development and delivery collaborative apparel solution digital platform." U.S. Patent No. 6,741,904. 25 May 2004.

10. Gunnar Stefansson, Kenth Lumsden, (2008) "Performance issues of Smart Transportation Management systems", International Journal of Productivity and Performance Management, Vol. 58 Iss: 1, pp.55 - 70

11. ADB, Changing Course: A New Paradigm for Sustainable Urban Transport, 1st ed. Oakland: O Books Sun & Moon.

12. J. Zhu and S. Roy, "MAC for dedicated short range communications in intelligent transport system," IEEE Communications Magazine, vol. 41, pp. 60-67, 2003.

13. M. Naphade, G. Banavar, C. Harrison, J. Paraszczak and R. Morris, "Smarter Cities and Their Innovation Challenges", Computer, vol. 44, no. 6, pp. 32-39, 2011.

14. A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, "Internet of Things for smart cities", IEEE Internet Things J., vol. 1, no. 1, pp. 22-32, Feb. 2014.

15. L. Schewel and D. Kammen, "Smart Transportation: Synergizing Electrified Vehicles and Mobile Information Systems", Environment: Science and Policy for Sustainable Development, vol. 52, no. 5, pp. 24-35, 2010.

16. Deakin, Elizabeth; Frick, Karen Trapenberg; & Skabardonis, Alexander. (2009). Intelligent Transport Systems. ACCESS Magazine, 1(34), 29 -34. UC Berkeley: University of California Transportation Center. Retrieved from: https://escholarship.org/uc/item/3mb3n3j4

17. Regan, Michael A., et al. Intelligent transport systems: Safety and human factors issues. No. 01/01. 2001.

18. O. Vermesan and P. Friess, Internet of things- From research and innovation to market deployment, 1st ed. Aalborg: River Publishers, 2014.

19. T. Vaa, M. Penttinen and I. Spyropoulou, "Intelligent transport systems and effects on road traffic accidents: state of the art," in IET Intelligent Transport Systems, vol. 1, no. 2, pp. 81-88, June 2007.

20. M. Naphade, G. Banavar, C. Harrison, J. Paraszczak and R. Morris, "Smarter Cities and Their Innovation Challenges", Computer, vol. 44, no. 6, pp. 32-39, 2011.

21. A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, "Internet of Things for smart cities", IEEE Internet Things J., vol. 1, no. 1, pp. 22-32, Feb. 2014.

22. L. Schewel and D. Kammen, "Smart Transportation: Synergizing Electrified Vehicles and Mobile Information Systems", Environment: Science and Policy for Sustainable Development, vol. 52, no. 5, pp. 24-35, 2010. ==15

23. D. Advani, “How Melbourne Is Fast-Tracking Smart Transit Using Big Data,” The Huffington Post. [Online]. Available: http://www.huffingtonpost.com/deepak-advani/ibm-how-melbourne-is-fast-_b_4178465.html. [Accessed: 10-Jan-2017].

24. Yovanof, G. S., & Hazapis, G. N. (2009). An architectural framework and enabling wireless technologies for digital cities & intelligent urban environments. Wireless Personal Communications, 49(3), 445-463. Available at http://www.springerlink.com/content/g1v63025217mt8x0/.

25. Moser, M. A. (2001). What is smart about the smart communities movement? EJournal, 10/11(1). Available at http://www.ucalgary.ca/ejournal/archive/v10-11/v10-11n1Moser-print.html

26. Shabbir Ahmed and Salil S. Kanere. 2006. SKVR: scalable knowledge based routing architecture for public transport networks. In Proceedings of the 3rd international workshop on Vehicular ad hoc networks (VANET '06). ACM, New York, NY, USA, 92-93.

27. K. Ashokkumar, Baron Sam, R. Arshadprabhu, Britto Cloud Based Intelligent Transport System. Procedia Computer Science Volume 50, 2015, Pages 58–63 Big Data, Cloud and Computing Challenges

28. Z. Li, C. Chen and K. Wang, "Cloud Computing for Agent-Based Urban Transportation Systems," in IEEE Intelligent Systems, vol. 26, no. 1, pp. 73-79, Jan.-Feb. 2011. doi: 10.1109/MIS.2011.10

29. S. Wibowo and S. Grandhi, "A multicriteria analysis approach for benchmarking smart transport cities," 2015 Science and Information Conference (SAI), 2015.

30. Holleis, P., Luther, M., Broll, G., Cao, H., Koolwaaij, J., Peddemors, A., Ebben, P., Wibbels, M., Jacobs, K. and Raaphorst, S., 2012, May. TRIPZOOM: A system to motivate sustainable urban mobility. In Proc. International Conference on Smart Systems, Devices and Technologies (SMART '12), ThinkMind.

31. Baek, J.S., Manzini, E. and Rizzo, F., 2010, July. Sustainable collaborative services on the digital platform: Definition and application. In DESIGN RESEARCH SOCIETY INTERNACIONAL CONFERENCE, Montreal.

32. Iga, A., et al., 2007, A style of craftsmanship (In Japanese), Tokai University Press.

33. Hosono, S., Hasegawa, M., Hara, T., Shimomura, Y. and Arai, T., 2009, March. A methodology of persona-centric service design. In Proceedings of the 19th CIRP Design Conference–Competitive Design. Cranfield University Press.

34. Orwant, J. (1996). For want of a bit the user was lost: Cheap user modeling. In: IBM Systems Journal, 35(3&4), 398-416.

35. SIGCHI Bulletin Vol.29 No.3, July 1997: A Critical Examination of Separable User Interface Management Systems: Constructs for Individualization, Homepages.cwi.nl, 2017. [Online]. Available: http://homepages.cwi.nl/~steven/sigchi/bulletin/1997.3/reynolds.html.

36. Fagerholm, F. and Münch, J., 2012, June. Developer experience: Concept and definition. In Software and System Process (ICSSP), 2012 International Conference on (pp. 73-77). IEEE.

37. Palviainen, J., Kilamo, T., Koskinen, J., Lautamäki, J., Mikkonen, T. and Nieminen, A., 2015, April. Design framework enhancing developer experience in collaborative coding environment. In Proceedings of the 30th Annual ACM Symposium on Applied Computing (pp. 149-156). ACM.

38. Kuusinen, K., 2015, December. Software developers as users: Developer experience of a cross-platform integrated development environment. In International Conference on Product-Focused Software Process Improvement (pp. 546-552). Springer International Publishing.

39. API Design 102: The Developer Experience | API Design and Strategy, Apiacademy.co, 2017. [Online]. Available: http://www.apiacademy.co/resources/api-design-102-the-developer-experience/.

40. Nam, T. and Pardo, T.A., 2011, September. Smart city as urban innovation: Focusing on management, policy, and context. In Proceedings of the 5th international conference on theory and practice of electronic governance (pp. 185-194). ACM.

41. Johnson, B. (2008). Cities, systems of innovation and economic development. Innovation: Management, Policy & Practice, 10(2-3), 146-155.

42. Toppeta, D. (2010). The Smart City Vision: How Innovation and ICT Can Build Smart, "Livable", Sustainable Cities. The Innovation Knowledge Foundation. Available from http://www.thinkinnovation.org/file/research/23/en/Top peta_Report_005_2010.pdf.

43. Washburn, D., Sindhu, U., Balaouras, S., Dines, R. A., Hayes, N. M., & Nelson, L. E. (2010). Helping CIOs Understand "Smart City" Initiatives: Defining the Smart City, Its Drivers, and the Role of the CIO. Cambridge, MA: Forrester Research, Inc. Available from http://public.dhe.ibm.com/partnerworld/pub/smb/smarterplanet/forr_help_cios_und_smart_city_initiatives.pdf.

44. Su, K., Li, J. and Fu, H., 2011, September. Smart city and the applications. In Electronics, Communications and Control (ICECC), 2011 International Conference on (pp. 1028-1031). IEEE.

45. Chourabi, H., Nam, T., Walker, S., Gil-Garcia, J.R., Mellouli, S., Nahon, K., Pardo, T.A. and Scholl, H.J., 2012, January. Understanding smart cities: An integrative framework. In *System Science (HICSS), 2012 45th Hawaii International Conference on* (pp. 2289-2297). IEEE.

46. Neirotti, P., De Marco, A., Cagliano, A.C., Mangano, G. and Scorrano, F., 2014. Current trends in Smart City initiatives: Some stylised facts. *Cities*, *38*, pp.25-36.

47. Moser, M.A., 2001. What is smart about the smart communities movement. *EJournal, 10*, *11*(1), pp.1-11.

48. Yovanof, G.S. and Hazapis, G.N., 2009. An architectural framework and enabling wireless technologies for digital cities & intelligent urban environments. *Wireless personal communications*, *49*(3), pp.445-463.

49. Neirotti, P., De Marco, A., Cagliano, A.C., Mangano, G. and Scorrano, F., 2014. Current trends in Smart City initiatives: Some stylised facts. *Cities*, *38*, pp.25-36.

50. Dell'Era, C. and Landoni, P., 2014. Living Lab: A methodology between user-centred design and participatory design. Creativity and Innovation Management, 23(2), pp.137-154.

51. Larios, V.M., Gomez, L., Mora, O.B., Maciel, R. and Villanueva-Rosales, N., 2016, September. Living labs for smart cities: A use case in Guadalajara city to foster innovation and develop citizen-centered solutions. In Smart Cities Conference (ISC2), 2016 IEEE International (pp. 1-6). IEEE.

52. Chen, K.B., Tsui, H.L., Yang, C.T., Ting, L.H. and Houng, H., 2010, June. A Living Lab model for user driven innovation in urban communities. In Technology Management Conference (ICE), 2010 IEEE International (pp. 1-7). IEEE.

53. Eriksson, M., Niitamo, V.P. and Kulkki, S., 2005. State-of-the-art in utilizing Living Labs approach to user-centric ICT innovation-a European approach. Lulea: Center for Distance-spanning Technology. Lulea University of Technology Sweden: Lulea.

54. Leminen, S., Westerlund, M. and Nyström, A.G., 2012. Living Labs as open-innovation networks. Technology Innovation Management Review, 2(9).

55. Feurstein, K., Hesmer, A., Hribernik, K.A., Thoben, K.D. and Schumacher, J., 2008. Living Labs: a new development strategy. European Living Labs-a new approach for human centric regional innovation, pp.1-14.

56. Liedtke, C., Jolanta Welfens, M., Rohn, H. and Nordmann, J., 2012. LIVING LAB: user-driven innovation for sustainability. International journal of sustainability in higher education, 13(2), pp.106-118.

57. Alba, M., Avalos, M., Guzmán, C. and Larios, V.M., 2016, September. Synergy between smart cities' hackathons and living labs as a vehicle for accelerating tangible innovations on cities. In Smart Cities Conference (ISC2), 2016 IEEE International (pp. 1-6). IEEE.

58. J. Schöning, Y. Rogers, J. Bird, L. Capra, J. A. McCann, D. Prendergast, and C. Sheridan, “Intel Collaborative Research Institute - Sustainable Connected Cities,” in Distributed Communities on the Web, vol. 7683, no. 28, Berlin, Heidelberg:Springer Berlin Heidelberg, 2012, pp. 364–372.

59. T. Magedanz and S. Covaci, “UNIFI - NGNI,” daadunifi.org, 2014. [Online]. Available: http://www.daadunifi.org.

60. S. C. Alliance, Smart City Alliance. Alberta : Smart City Alliance, [Online]. Available: https://smartcityalliance.ca/

61. Sustainable Singapore Blueprint, Our Home, Our Environment, Our Future. 2015. Ministry of the Environment and Water Resources. [Online]. Available:http://www.mewr.gov.sg/ssb/files/ssb2015.pdf

62. M. Network, METROLAB Network. Washington DC: Carnegie Mellon University, 2016. [Online]. Available:http://metrolab.heinz.cmu.edu/

63. Living Lab Handbook (2010) [WWW document]. URL www.ve-forum.org/apps/comm.asp?$1=528

64. Konsti-Laasko, S., Pihkala, T. and Kraus, S. (2012) Facilitating SME Innovation Capability through Business Networking. Creativity and Innovation Management, 21, 93–105.

65. Schuurman, D., Lievens, B., De Marez, L. and Ballon, P. (2012) Towards Optimal User Involvement in Innovation Processes: A Panel-Centered Living Lab Approach. Technology Management for Emerging Technologies (PICMET), 2012 Proceedings of PICMET'12. IEEE, New York, pp. 2046–2054.

66. Feurstein, K., Hesmer, K.A., Hribernik, K.-D. and Schumacher, J. (2008) Living Labs: A New Development Strategy. In Schumacher, J. and Niitamo, V.-P. (eds.), European Living Labs – A New Approach for Human Centric Regional Innovation. Wissenschaftlicher Verlag Berlin, Berlin, pp. 1–14.

67. 3 Ways to Get Your Own Digital Platform, Harvard Business Review, 2017. [Online]. Available: https://hbr.org/2016/07/3-ways-to-get-your-own-digital-platform.

68. Designers are not users (Video), Nielsen Norman Group, 2017. [Online]. Available: https://www.nngroup.com/videos/designers-are-not-users/.

70. Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. 2004. Design science in information systems research. MIS Q. 28, 1 (March 2004), 75-105.

71. J. Riggins, Why API Developer Experience Matters More Than Ever | Nordic APIs |, Nordic APIs, 2017. [Online]. Available: https://nordicapis.com/why-api-developer-experience-matters-more-than-ever/.

72. Michael Mahemoff, PATTERNS OF DEVELOPER EXPERIENCE, 2017. [Online]. Available: http://softwareas.com/patterns-of-developer-experience/.

73. The Developer Experience, Speaker Deck, 2017. [Online]. Available: https://speakerdeck.com/pamelafox/the-developer-experience.

74. Building a Digital Platform in the API Economy, MuleSoft, 2017. [Online]. Available: https://www.mulesoft.com/lp/whitepaper/api/digital-platform.

75. APIs for Humans: The Rise of Developer Experience (DX) | HelloSign Blog, Hellosign.com, 2017. [Online]. Available: https://www.hellosign.com/blog/the-rise-of-developer-experience.

76. Brooke, J. 1996. SUS: A "quick and dirty" usability scale. In Usability evaluation in industry, Edited by: Jordan, P. W., Thomas, B. A. Weerdmeester and McClelland, I. L. 189–194. London: Taylor & Francis. [Google Scholar]

77. A. Bangor, P. Kortum and J. Miller, "An Empirical Evaluation of the System Usability Scale", International Journal of Human-Computer Interaction, vol. 24, no. 6, pp. 574-594, 2008.

78. ISO 9241-11:1998(en)Ergonomic requirements for office work with visual display terminals (VDTs) — Part 11: Guidance on usability

79. Building an Awesome Developer Experience, 2017. [Online]. Available: https://apigee.com/api-management/#/video/878.

80. Using Google Maps as an interface for the library catalogue | Library Hi Tech | Vol 26, No 1, Emeraldinsight.com, 2017. [Online]. Available: http://www.emeraldinsight.com/doi/abs/10.1108/07378830810857780.

81. E. O'Neill, User-developer cooperation in software development: Building Common Ground and Usable system. Springer, 2013, pp. 39 - 42.

82. New isn't on its way. We're applying it right now. | Accenture, Accenture.com, 2018. [Online]. Available: https://www.accenture.com/us-en/_acnmedia/PDF-29/Accenture-Five-Ways-To-Win-WithDigital-Platforms-Executive-Summary.pdf.

83. 3 Ways to Get Your Own Digital Platform, Harvard Business Review, 2017. [Online]. Available: https://hbr.org/2016/07/3-ways-to-get-your-own-digital-platform.

84. Paolo Malinverno, Mark O'Neill and K. Moyer, Gartner Reprint, Gartner.com, 2017. [Online]. Available: https://www.gartner.com/doc/reprints?id=1-4DCI6R6&ct=170911&st=sb.

Non-Academic References

85. NEXCOM In-vehicle Computers Provide Improved Waste Collection Services, Nexcom.com, 2018. [Online]. Available: http://www.nexcom.com/news/Detail/nexcom-in-vehicle-computers-provide-improved-waste-collection-services.

86. Why Do We Pay These People Anyway? – Google Developers – Medium", Medium, 2018. [Online]. Available: https://medium.com/google-developers/why-do-we-pay-these-people-anyway-d7ed706d6d55.

87. Living Lab Bus – Open innovation and test platform, Livinglabbus.fi, 2018. [Online]. Available: http://livinglabbus.fi/LLB_slideset.pdf.

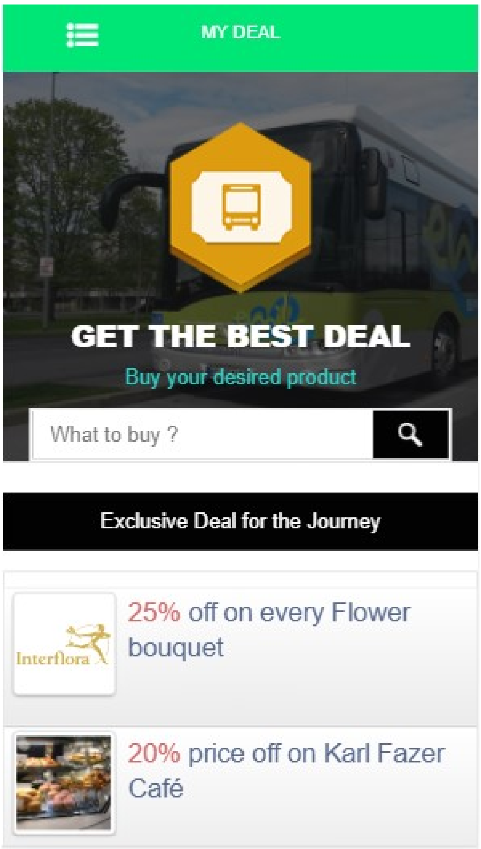
88. D. Design, Developer experience is user experience - DevRel.net, DevRel.net, 2017. [Online]. Available: https://devrel.net/developer-experience/developer-experience-user-experience.

89. Developer Experience (DX) — Devs Are People Too - DZone Agile, dzone.com, 2017. [Online]. Available: https://dzone.com/articles/developer-experience-dx-devs-are-people-too.

90. API Design 102: The Developer Experience | API Design and Strategy, Apiacademy.co, 2017. [Online]. Available: http://www.apiacademy.co/resources/api-design-102-the-developer-experience/.

91. Everything you need to know about Digital Platforms, Product marketing in the digital era, 2018. [Online]. Available: http://stephane-castellani.com/everything-you-need-to-know-about-digital-platforms/.

Appendix A – Application Interfaces



25% off on every Flower bouquet

20% price off on Karl Fazer Café

Figure 1 Front Page

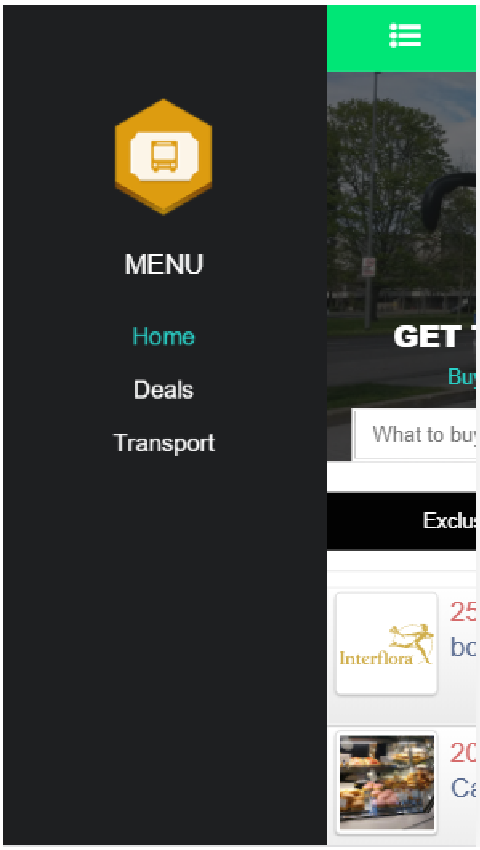
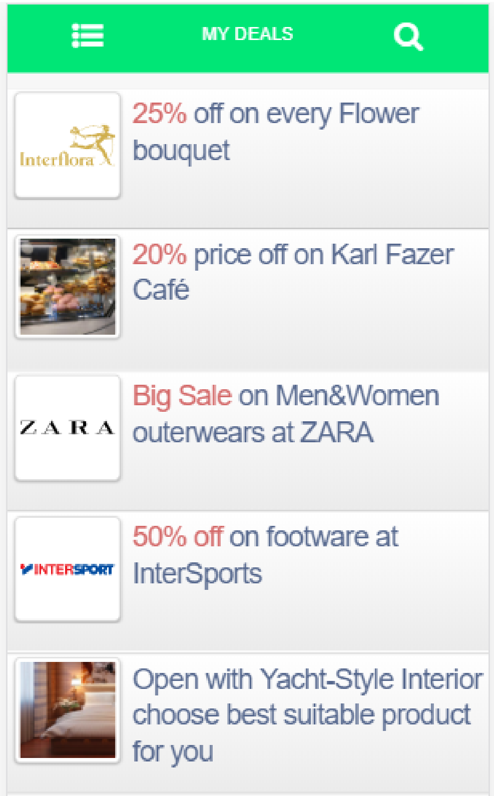


Figure 2 Menus of the application



25% off on every Flower bouquet

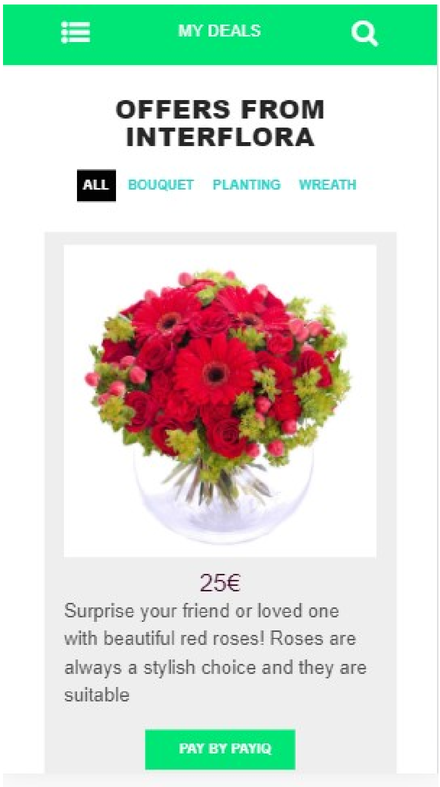
20% price off on Karl Fazer Café

Big Sale on Men&Women outerwears at ZARA

50% off on footware at InterSports

Open with Yacht-Style Interior choose best suitable product for you

Figure 3 Deals offered by the application

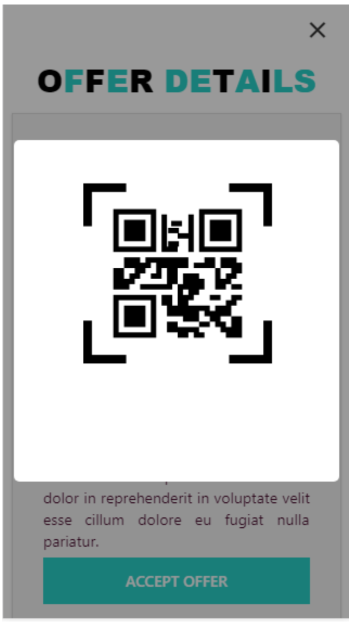


OFFERS FROM INTERFLORA

25€

Surprise your friend or loved one with beautiful red roses! Roses are always a stylish choice and they are suitable

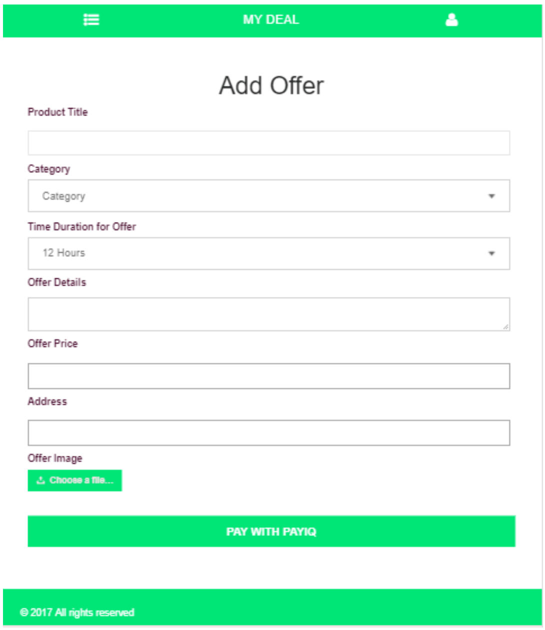
Figure 4 Single offer details



OFFER DETAILS

dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Figure 5 User Received QR code for quick pick up



MY DEAL

Add Offer

Product Title

Category

Category

Time Duration for Offer

12 Hours

Offer Details

Offer Price

Address

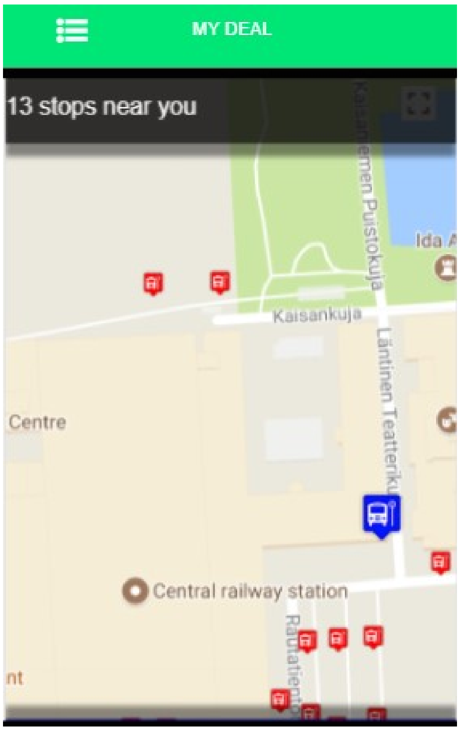
Offer Image

Choose a file...

PAY WITH PAYIQ

© 2017 All rights reserved

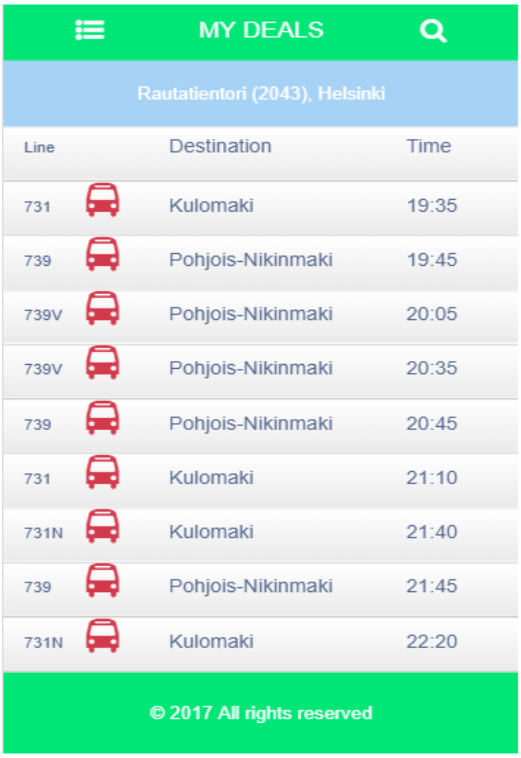
Figure 6 Offer adding page for the client



MY DEAL

13 stops near you

Figure 7 Bus stops near to the user



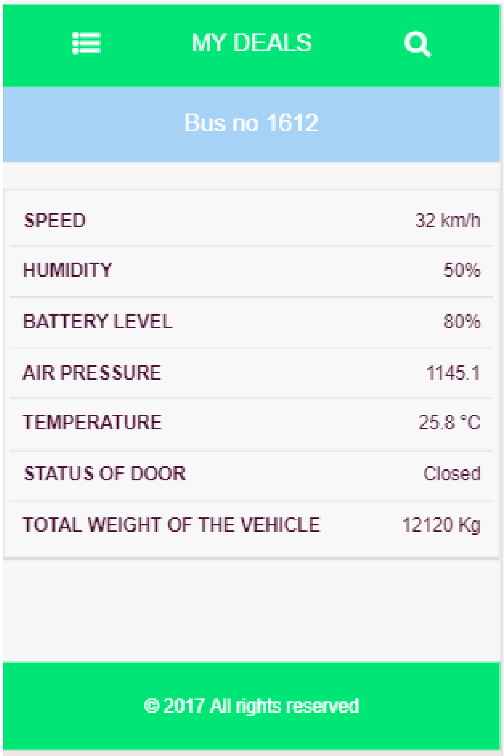
MY DEALS

Rautatientori (2043), Helsinki

Line		Destination	Time
731	🚌	Kulomaki	19:35
739	🚌	Pohjois-Nikinmaki	19:45
739V	🚌	Pohjois-Nikinmaki	20:05
739V	🚌	Pohjois-Nikinmaki	20:35
739	🚌	Pohjois-Nikinmaki	20:45
731	🚌	Kulomaki	21:10
731N	🚌	Kulomaki	21:40
739	🚌	Pohjois-Nikinmaki	21:45
731N	🚌	Kulomaki	22:20

© 2017 All rights reserved

Figure 8 Available buses to the stop with time table and destinations



MY DEALS

Bus no 1612

SPEED	32 km/h
HUMIDITY	50%
BATTERY LEVEL	80%
AIR PRESSURE	1145.1
TEMPERATURE	25.8 °C
STATUS OF DOOR	Closed
TOTAL WEIGHT OF THE VEHICLE	12120 Kg

© 2017 All rights reserved

Figure 9 Bus Specific Information