

## Designing a Smart Car Parking System (PoC) Prototype Utilizing CCTV Nodes: A vision of an IoT parking system via UCD process

Muftah Fraifer\*, Mikael Fernström

Interaction Design Center, dept. Computer Science and Information Systems, University of Limerick, V94 T9PX, Ireland

### ARTICLE INFO

Article history:

Received: 04 April, 2017

Accepted: 02 June, 2017

Online: 16 June, 2017

Keywords:

Smart parking

Video processing

Interaction systems

UCD

HCI

IoT

### ABSTRACT

Using smart parking systems has become very important, and particularly so for metropolitan areas, because of the benefits for drivers in many aspects, such as time, frustration, stress, and anger, in addition to the increased consumption of fuel while searching for a vacant parking space. This paper proposes a review of recent advances in sensing and communication technology concerning smart parking systems. It includes a brief study of the selected topics and provides an implementation process of those selected systems. Moreover, this work proposes a design approach for a smart car parking system prototype based on utilizing CCTVs (nodes), it is also illustrates the algorithms used for computer vision detection through simulation and real environments, as the system has been deployed in both these environments. Furthermore, the system has been tested and evaluated by stakeholders via a user-centred design process by applying a qualitative research; the promising results demonstrate the effectiveness of our prototype. Finally, this paper discusses the benefits of engaging the stakeholders to develop the prototype.

## 1. Introduction

Installing a parking system at any car park can be expensive, but recently, many researchers have been exploring techniques, such as vehicular ad hoc networks (VANET), or integrated RFID with Wireless Sensor Networks (WSN). As the Internet of Things (IoT) has evolved, further possibilities to provide services have emerged. Here, IoT can connect objects, such as smart-phones, sensors, and actuators, as nodes (in our case, CCTV cameras). The development of IoT may transform a number of domains dramatically, for example, automation, transportation, healthcare, and financial services. IoT can also be applied to new concepts and developments for smart homes to provide improved comfort, security, and quality of life. Some parking systems need extensive infrastructure investment, for example, digging holes in the ground at each parking spot to install a sensor. However, we are proposing an approach that can utilise an existing system instead, i.e., CCTV (node) coverage. The proposed system has many advantages. Firstly, it is cost efficient, as it is using the existing CCTV cameras (nodes). Secondly, no infrastructure changes are needed for the parking lots. Finally, and according to the interviews with stakeholders that were done during the evaluation process using the User-centred Design (UCD) method, the prototype can be

useful for small and large parking areas whether the parking area is active or not where the user can easily identify and locate the vacant parking spots on the fly. This paper presents a prototype of a low-cost and flexible smart parking monitoring system using an embedded micro-web server, with IP connectivity for accessing and controlling devices and appliances remotely [1].

## 2. Contributions

We introduce a novel IoT parking system prototype based on computer vision and build a model of the system operation.

The general approach to design a usable Interactive Technology (IoT) smart parking system for stakeholders (end-users). The methodology for accomplishing this goal was to apply User Centred Design (UCD), which included observation, surveys with stakeholders, and in-depth interviews with users. This article and research contribution covers different aspects:

- It review and presents a related works of different parking systems.
- Design, implement, verify, validate, test and evaluate a smart parking system utilizing CCTVs (camera nodes) facilitating provision of mobile services in indoor and outdoor environments within a university campus as an example.

\*Corresponding Author: Muftah Fraifer, Interaction Design Centre -Dep of CSIS, V94 T9PX, Ireland | Email: [muftah.fraifer@ul.ie](mailto:muftah.fraifer@ul.ie)

- In the field of IoT, facilitating the parking systems process by enabling a camera (thing) to be a smarter thing using a Galileo board (microprocessor) by providing a solution by links smart objects to the Internet.
- In the field of HCI, the knowledge and insights about UCD and how it can be put into practice to design a usable and effective IoT Parking System.

### 3. Related Work

To provide an overview of the different technologies used by others, the following is a summary of the systems we have reviewed. Foshini et al. discussed the design and implementation of an M2M application in the field of road traffic management study which used an integrated IoT retractable bollard barrier management system to allow vehicular access to restricted city areas, based on a standard infrastructure and software components [2]. Pala and Nihat used RFID for finding the nearest vacant parking space and reporting to the driver [3]. Tang et al. developed a system using crossbow motes that have a low unit cost; the system enabled a car to detect entry to the car park and guide the driver to an empty parking space by signs displayed to the driver [4]. Benson et al. proposed that an RF transceiver and micro controller system could monitor the availability of car-parking spaces and send this information to drivers and car park administrators [5]. Funck et al. proposed a system using CCTV cameras fitted in car parks to automatically detect car parking spaces. However, all these methods have the problem that sometimes they are not accurate [6]. Panayappan et al. described a parking system using VANET, which located available parking spaces. The system depended on roadside units to relay parking messages and GPS coordinates to locate parking positions [7]. Kianpisheh et al. presented a smart parking system using ultrasonic sensors. For each individual car park, a sensor was fixed in the ceiling above each parking space. Such ultrasonic sensors operate based on echolocation. The sensor emits a sound, which is reflected back to the sensor; the presence of a vehicle changes the timing of the reflection [8]. Mathur et al. discussed a centralized solution, with cars equipped with ultrasonic sensors driving past parking spaces to collect occupancy data and upload the data to a database; the cars that require parking query the database [9]. In 2010, Hazrin et al. proposed a smart parking using SMS services to communicate with drivers. A parking reservation system was developed whereby users could reserve a parking spot with an SMS message, including GPS information [10]. Bong et al. proposed a system with image processing algorithms, the motivation being to minimise the cost. The system was called Car-Park Occupancy Information System (COINS) [11]. Klappenecker et al. modelled a parking lot as a continuous-time Markov chain. The parking area was modelled as a grid, and schemes for information aggregation and dissemination over the grid were proposed [12]. Jain et al. proposed a system with a Gate-

PC Controller and Embedded Gate Hardware, RFID System, Modular Parking Management Platform [13]. Lee et al. proposed the use of a combination of magnetic and ultrasonic sensors for the accurate and reliable detection of vehicles in a parking lot, also describing a modified version of the Min-max algorithm for the detection of vehicles using magnetometers [14].

### 4. Research Methodology and Background

Our general approach is based on UCD. This includes observations of user behaviour, surveys with stakeholders, and in-depth interviews with users (ranging from the parking users to system operators and facilities management staff).

This is carried out in parallel with technical probes, i.e., incremental development and integration of system components. At present, we have a low-fidelity prototype where parking and CCTV nodes are emulated in an office environment (using toy cars) and then actual parking spaces. We explore the desired services informed and shaped by our user groups. We strive towards using open source software as far as possible, and in our preliminary sketches of the system, we may also be using Google APIs, including both maps and calendar functions that can be used for reserving parking spaces based on the user's planned activity on campus. It is thought that a minimally intrusive system that users' experience as augmenting their activity may be more likely to succeed. To address this, we are using part of the University of Limerick campus as a living lab to engage with users and evaluate a range of services. The campus has full coverage with open Wi-Fi, 3G, CCTV and a range of architecturally complex spaces with buildings interspersed with car parking [1].

### 5. Hardware and Software Requirements

- Micro-controller Intel Galileo board or, as in our case, desktop computer with Linux.
- CCTV cameras (nodes) A,B,C, etc.
- Parking Server Computer (PS).
- Software components with Java and Intel OpenCV library, Parking Server PS (LAMP stack)

Figure 1 below shows a Galileo board, which is used to run the OpenCV code directly. Also, the code can be compiled and run using the live-video streaming link for the live scene of the parking areas to monitor the parking status in real time.

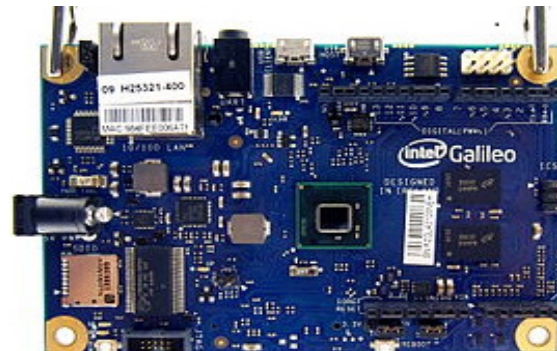


Figure 1: Galileo board

## 6. Application Architecture Layers

From the high level of the IoT based smart parking system; there are three main layers ; As shown in Figure 3.

- Application layer: the main purpose of this layer is data processing and service providing.
- Transmission layer: which is about transferring data over long distances and for wide areas.
- Perception layer: the system aims to collect and process the data from the physical world (in our case is using computer vision technology)



Figure 2 : Prototype CCTV camera node with the cover shield to protect the Galileo board

## 7. Design Goals

Based on our literature review and having access to the existing infrastructure on campus, we set out to explore a smart parking system based on CCTVs/Web-cams (nodes) that can provide drivers with convenient and simple parking service information. The prototype has been evaluated and tested using stakeholders, and the UCD process has been used to gather the end-users' needs and requirements. The process is also used to improve the prototype iteratively.

The system must fulfil a range of criteria. Firstly, it has to provide real-time navigation support to the driver, for example, by utilizing Google maps. Secondly, it has to provide a friendly parking information service to both the parking user and the parking facility's operator/owner. Finally, using CCTV cameras (nodes) for this purpose as well as for security purposes, it is an economical option. The main function the system will perform is to gather information about the parking status in real-time, and help drivers to navigate directly to vacant parking spaces [1]. Figure 3 and Figure 4 shows (admin GUI ) when run the system in outdoor and indoor environments

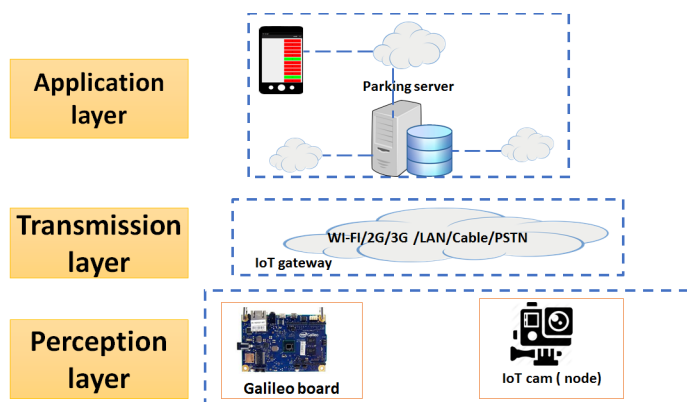


Figure 3: smart parking IoT application architecture

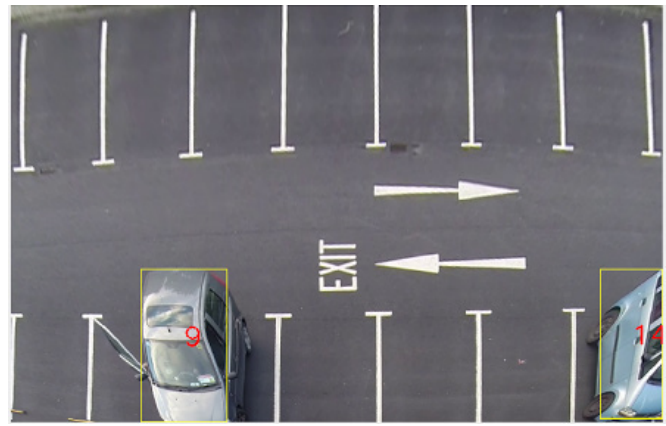


Figure 4: snapshot of testing scenarios in out-door environment covered by one node from admin GUI (monitoring (parking status on real-time )

## 8. User-Centred Design and its Influence on our Porotype

The UCD methodology in this process is used as a framework of processes in which the needs of stakeholders of a product or service are given extensive attention at each stage of the design process. It begins by defining the problem, which in our case is congestion during peak times and the effect on drivers, as it appears difficult for the driver to find a parking spot [1].

Another aspect to this research-study is carried out to explore the approach of UCD with a focus on stakeholders' involvement in the development of a smart parking technology prototype. We tried to alleviate this everyday issue of finding a parking space through the proposed prototype. Although some of the stakeholders' suggestions may involve some costs, the stakeholders have been identified and a thorough investigation of their needs has been conducted. By performing tasks and needs analyses, designers can develop alternative design solutions to be evaluated by users. The data analysis was based on a qualitative content analysis from a interaction design perspective.

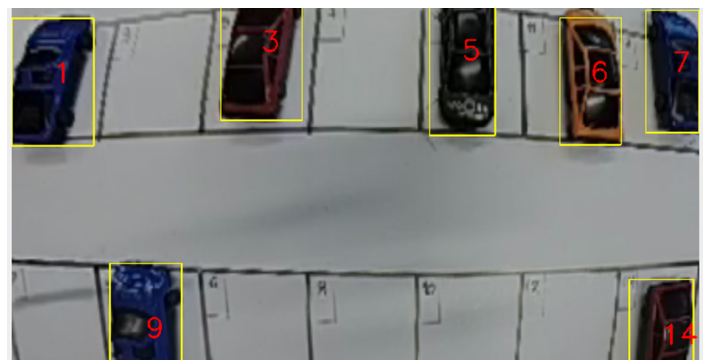


Figure 5: snapshot of Testing scenario in in-door environments from admin GUL (monitoring (parking status on real-time )

## 9. Prototype of the Proposed System

There are three main components:

- firstly, an app installed on the user's smartphone that connects the user with a parking server (PS);



- secondly, an IoT cams (which in our prototype is a Web Camera) nodes that is connected to a microcontroller (Intel Galileo);
- and lastly, a PS (Parking Server) hosting a database.

When the user installs the app on their Smartphone, they are given unique ID. The app can also hold a user’s profile and preferences including all the information entered by the user during the registration with the system and the server. This allows the app to provide services that match a user’s parking preferences with actual available spaces in the actual parking scenario at UL [1].

Our main scenario is the aforementioned openly accessible areas. Each parking lot is divided into areas that are covered by nodes, as shown in Figure 4 below.

Our prototype uses IoT camera nodes. An IoT cam can be set to report changes to the scene as a grid of polygons. At the same time, the user can interact with the system through the app. The IoT cam uses a DHCP connection to connect with the PS, which is on a fixed IP.

A simple blob-detection algorithm (using Intel’s OpenCV library) is used for computer vision, which monitors and reports changes in a defined grid of polygons (i.e. if parking spaces are vacant or occupied). IoT cams connect to the PS, which aggregates the data and updates the database of available car parking spaces in close to real-time. It should be possible to define and redefine parking areas by system operators. The system will operate in close to real-time, so we can aim to improve the parking experience rather than forcing and regimenting it. Status information regarding the parking area within the coverage of any IoT cam (nodes) is continuous, and changes are sent to the PS. We are using both scenarios (indoor and outdoor environments); the first is only to test the feasibility of the system, which depends on toy cars in a model parking lot, as shown in Figure 3, while the second prototype was tested on real cars with an actual car parking area, as shown in Figure 4 (both snapshots were taken from admin GUI). The camera, in a simulated environment (CCTV), is fixed in position with a side view of the parking space. The height of the camera must be sufficient to obtain a clear plan view of the parking

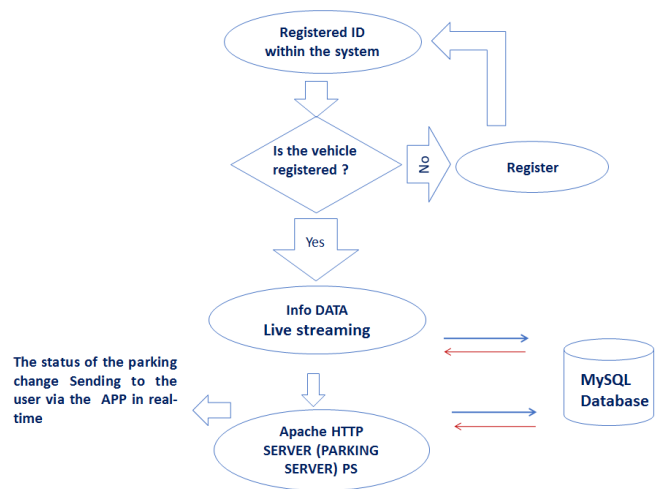


Figure 6: The operations triggered by the car-in and car-out events.

space, as in Figure 1. The IoT cam is positioned with an angled view of the parking space acquiring a fixed scene [1].

### 10. Interaction with MySQL Database

We are using a LAMP server as the open-source platform, which uses Linux as the operating system, Apache as the Web server, MySQL as the relational database management system, and PHP for scripting.

The system has real-time communication between the mobile app and the server, using a RESTful API. Each car park can be identified with labels, for instance, (Node A, Space 1, 2, 3, 4 etc.), so when a driver enters the parking area, they can easily identify each parking space. The main technique of the system is to monitor the system through IoT cams, while the video is processed by analysing the live streaming of data and changes are stored in a MySQL database. Figure 7 illustrates the UML use-case diagram the interaction of the prototype Open CV Algorithms (1)(2)(3)(4).

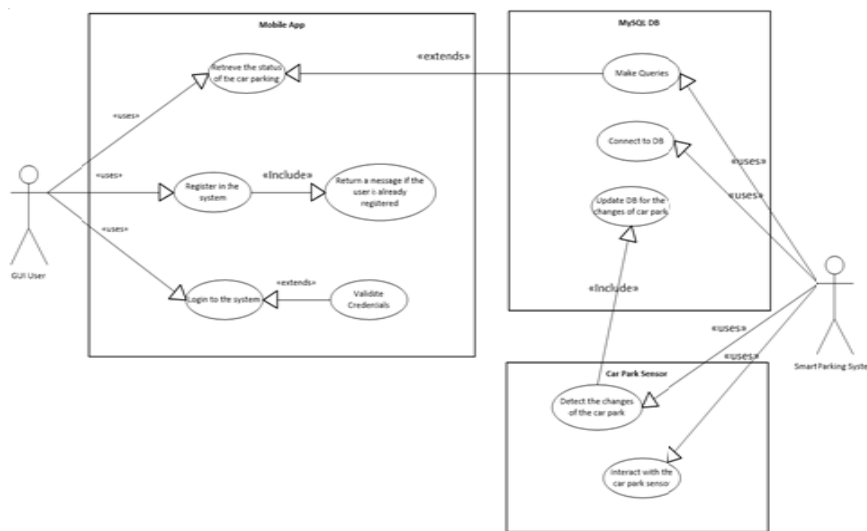


Figure 7: UML -USE CASE Diagram for the prototype

## 11. OpenCV Algorithms and Senescing Techniques

- The code depends on the OpenCV library, which has been used for video processing, to detect vacant parking spaces.
- The code retrieves frames from streaming video by applying a prospective transform Matrix on source images.
- For each pixel (x,y) of the destination image, the functions compute the coordinates of the corresponding “donor” pixel in the source image and copy the pixel value. A perspective transformation is then applied to the image. The function *warpPerspective* transforms the source image using the specified [3x3] matrix:

$$dst(x,y)=src\left(\frac{M_{11x}+M_{12y}+M_{13x}}{M_{31x}+M_{32y}+M_{33x}},\frac{M_{21x}+M_{22y}+M_{23}}{M_{31x}+M_{32y}+M_{33}}\right) \quad (1)$$

Where:

*src*: input image  
*dst*: output image  
*M*: 3x3 transformation matrix.

- Image loading, including erosion, smoothing, noise reduction, blurring, contrast fix, and brightness are applied to the uploaded images to get a uniform output from the camera input image.

$$dst(x,y)=\min_{(x',y'):\text{element}(x,y)\neq 0} src(x+x'+y+y') \quad (2)$$

- We are using the ISODATA method (Iterative Self Organizing Data Analysis Technique) for thresholding of image segmentation to get a high degree of accuracy and to consistently find the contours for each polygon (car) [2].
- A fixed-level threshold is applied to each array element. (THRESH BINARY)

As below

$$dst(x,y)=\begin{cases} maxval & \text{if } src(x,y) > thresh \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

*src* : input array (single-channel, 8-bit or 32-bit floating point).  
*dst* : output array of the same size and type as *src*.  
*thresh*: threshold value.

*maxval*: maximum value to use with the THRESH BINARY and THRESH BINARY INV thresholding types.

- One layer of the algorithm has been applied to detect vacant spaces in the parking lot using a contour method (rectangular detection) to identify the number of cars in the parking lot. The *findRect* method matches the detected shapes to the shape of a parked car, detecting the cars as rectangles.
- Rectangle X and Y coordinates as in (4), where variables are points, namely, first point and second point, and then a rectangle is drawn around the image.

$$\text{Point}(x,y) \rightarrow \text{Point2f}(x*2^{-shift}, y*2^{-shift}) \quad (4)$$

- This data is sent to the MySQL database through a MySQL connector, with a list of all parking areas, number of possible spaces, etc., so the system can count how many parking spots are available. Using the marked parking spaces as bearings, the code can identify the central position of a car as illustrated in the pseudocode below.

## 12. Illustration the stages of Implementation of the Code

The algorithm passes through different stages; starting by getting the prospective transform matrix of the image. Then the code retrieves frames from streaming video using the prospective transform Matrix as clarified in the pseudo code below.

### Pseudo code for the detection algorithm

```
// If a contour is detected in the parking spot, the contour
// represents a parked car (occupied parking spot)
If (contours.size() != 0 )
{
    // checking each contour
    For (int idx = 0; idx < contours.size(); idx++)
    {
        //To get the boundaries of the rectangular contour.
        Rect contourRect=Imgproc.boundingRect(contours.get(idx));

        // If bounding rect is between the two aspect ratio x,y.
        If
        ((contourRect.width < min_x)||(contourRect.height < min_y)||(contourRect.width > max_x)||(contourRect.height > max_y))
        continue;

        // To get the aspect ratio.
        double ratio = [(double)contourRect.height
        /((double)contourRect.width];
        if ((ratio < x)||(ratio > y))
        continue;
        If (the detected contour is a car)
        End If
        {
            // Draw a rectangle of the car.
            R = rectangle of car,
            // Get the position of each car. Pi = central position of car(i),

            // Give each occupied parking spot an ID.
            ParkID(i) = occupied_parking_spot(i),
            End If
            End for
        }
    }
}
```

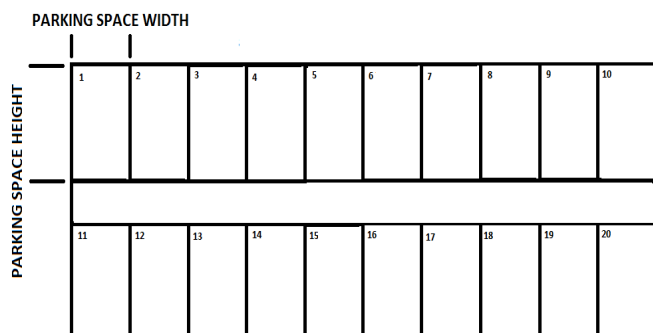


Figure 8: OpenCV algorithm design

### 13. System Implementation

A smart parking system based on CCTV (computer vision) has been proposed and tested in this paper. There are two Java applications; one is called *SPSystemApp* (OpenCV) and the other is called *SPMobileApp* (GUI). The interaction between them is via the MySQL database. The main purpose of the first application is to detect the status or changes of the car park and send this information periodically to the database in real time. At the same time, the GUI retrieves these changes for the user.

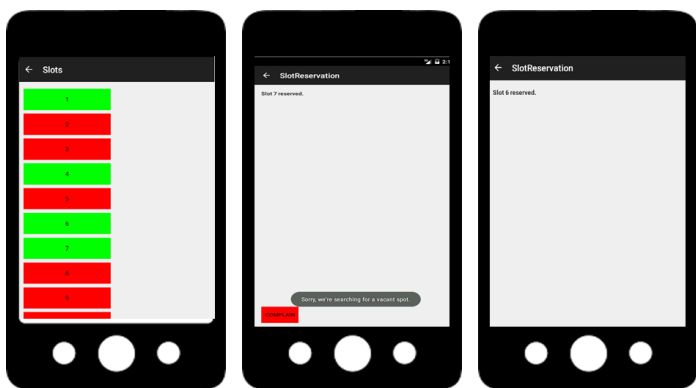


Figure 9: a,b,c Shows ( a snapshots )AVD with a reservation scenario ( it shows the interactive process by giving another parking spot

Figure 3 shows the implementation of the system, which shows red and green buttons where the user can reserve the parking space. The test cases used to evaluate the functionality of the system are listed in Table 1.

Table 1. Testing the system with different functions

| Test scenario  | Result |
|--|--------|
| MySQL connector can make a database update record according to status change | Pass   |
| Can the user reserve a specific parking space                                | Pass   |
| View the number of available parking spots in real-time                      | Pass   |
| Parking area status will update info to PS when a car leaves the car park    | Pass   |
| Able the user to register in the system using GUI                            | Pass   |
| Ability to do several tasks ( Registration/reserve)                          | Pass   |

### 14. The System and its Maximum Effectiveness

Regarding the implementation of the work of the system on the Galileo board (microcomputer ); Which distinguishes its use during its work in real time in addition to being a small board next to each camera (node) and do not need to connect video over the Internet to a central computer, where the delay here in the live broadcast to nearly half a minute or more. What also distinguishes our use of the motherboard in this system is the possibility of operating it in a real car par. We used a special cover to protect the processor from atmospheric conditions as shown in the picture. Again, we want to clarify that this article came as part of a research with several purposes to demonstrate the proof of concept and the prototype of the system of a complete illustration of the general perception to take advantage of the work of cameras not only in the parking lot, but also anywhere to include temporary parking on both sides of roads and others Through our experience of the reality of the work of the system and found that its work is in the best conditions and give good results identical to the reality of the case in terms of the number of cars and their location and the possibility of detection by the system if the speed of the Internet is not less than 120 Mbps ; to be able to stream data with a smooth and fast way to emulate reality and thus the quality of images through video broadcast in real time of the parking lot , which is dedicated to a particular camera (node) or more. Also, The image resolution required is no less (1400X960) pixels .To enable the system to operate with acceptable efficiency. The system handles (capturing, analyzing and grabbing frames from the connected nodes) up to 22 frames per second ; This is what distinguishes using OpenCV in this research.

### 15. System Requirements and Implantation Techniques

The system can detect vacant spaces using two different methods. The first method is using the IoT-Cam, where it pushes any changes in vacancy to the server (the microcontroller/server (in this scenario) while the PS can get the parking area status. The second way is using smart cameras that just capture an image, and all the processing happens on the server where the server can get an image from the camera providing the server is able to get smart camera status.

The IoT cameras (web cameras nodes) are connected via Wi-Fi to the Internet, so we can get the IP-number for live streaming of the video, which covers a certain area of the parking space; When the code is executed online, it assigns the link to the IoT cam to get the video stream analysed on the Galileo board or via the Linux machine..

### 16. System Reliability in Different Scenarios

Below is a collection of screenshots for application during use; The captured images were placed within the Android framework for clarity. Several scenarios have been considered; For example, the first user booked a certain parking place number 7 as in Fig x.a ; Upon arrival he found the place vacant, In this scenario, the user has only to click on the button called a complaint. Although such

behavior is unlikely to occur. Because it will usually apply fines to prevent such violations as any other system which provides services to the public as it shows in Fig x.b Which shows a second option to the user automatically.

#### 14.a System Efficiency

With regard in the reliability of our prototype. The system has been tested for several different times and for several different scenarios. The first tests were done in the laboratory (indoor environments) .While other tests were conducted in a real atmosphere. The tests carried out in the selected paper for the conference were carried out with a group of stakeholders and the number of them was 12 completed in the laboratory. All tests were ethically proven, and approved by ethical committee in the university. In terms of tests and reliability of experimental system. The system worked well and especially when the cars in both cases (toys and real) was properly parked. We noticed that those cars that are not parked along the lines of the stand cannot be detected by the system. The dark atmosphere makes the system fail to perform its functions, so we mentioned the need to light the parking lot in order to increase its efficiency. It is worth mentioning here. We connected the system to one of the cameras in a nearby parking lot .by Enabling VLC media player for Real Time Streaming Protocol (UDP); latency was quite high somethings, So there is some delay in the video displayed directly.

Based on the speed of the Internet and the quality of the images transmitted from the parking lot be the quality of the system and its efficiency and reliability.

We noticed that the system is delayed in this scenario to approximately 35 to 45 seconds. Sometimes the image is frozen in a fixed format as a result of the speed of the Internet or low performance of the processor. As the main purpose of this research is to clarify the potential of the work of the system in ideal circumstances.

#### 14.b Various Activities for Application and Features Available

Regarding the real implementation process of the system, the stakeholders themselves, who are students, staff, and visitors contributed voluntarily from the experience of the system in practice. This is what distinguishes this work. As the method used in the design process is User-centred Design (UCD).The images below illustrate the practical tests of the system [10a] [10b] [10c ] Low-fidelity prototype (paper prototyping) technique was used; as it is a method in the UCD process to ensure that stakeholders meet with satisfaction and acceptance. Then; The application is designed to run on the Android system. In a simple way, the application contains several activities. The most important is the possibility of registering in the system.

Where it is easy for the subscriber to enter his personal data including the number plate of his/her car. The car number has been added to its registration rules through suggestions for some stakeholders to follow the violators of the parking laws that want to apply such a system to facilitate the service to find a vacant position. The second activity relates to the possibility of seeing vacant positions in real time with the advantage of the possibility of reserving a particular position. The application is linked to a database as described above.

And each interaction with the system appears in the picture short message as shown in the figure below.

As previously reported, all trials were completed as shown in the table below. All the tests were in the appropriate conditions in terms of speed of the Internet and the quality of cameras and the weather and lighting. In addition to the emphasis on who was driving cars for the purpose of testing the need to be in a straight position parallel to the place allocated and not be tilted even a little and these are one of the biggest disadvantages of the system now.

### 17. Graphical user interface GUI for User and Admin

The system has a car parking mobile app as a key component, with a user interface for everyday use. Through the app, car parking information and services can be delivered to the user in a functional and user-friendly way and that are relevant to the user's present goals and location and according to the user's dynamic profile. Below is a collection of screenshots of the application during use.



Figure 10: GUI shows the status of parking area

When a user registers with the system, they are given a unique ID. The emulator used (an Android Virtual Device (AVD) ) as in Fig[9 ] and Fig[10, 11].also the figure below shows real the app on a real device .



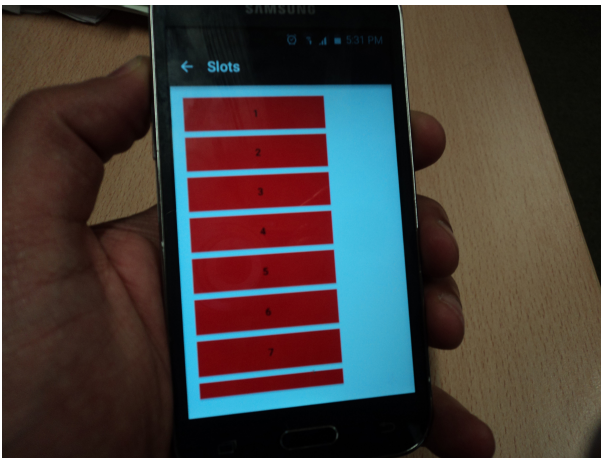


Figure 10 run the app on real device

Also, should someone else have deliberately or by mistake taken a pre-booked space, the system is currently able to allocate the user another parking space by the user clicking on the complain button as illustrated below in Figure 8(a,b,c) where red indicates an occupied and green a vacant spot.

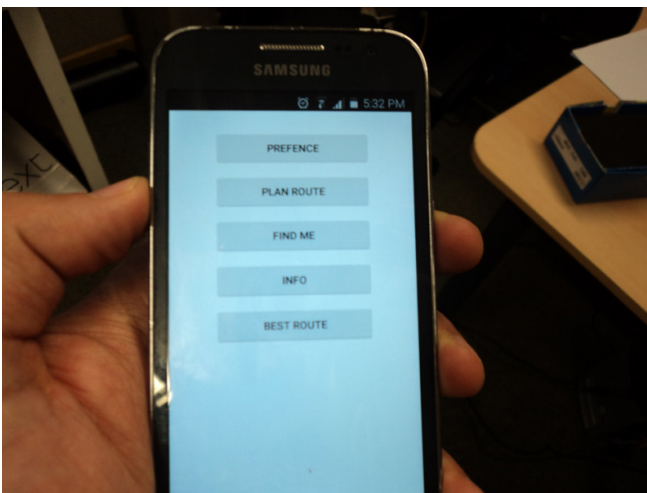


Figure 11 : run the app on real device for registration process

The figure below shows different snapshots of the admin GUI which is designed to help the system administrator to monitor the work and performance of the system.

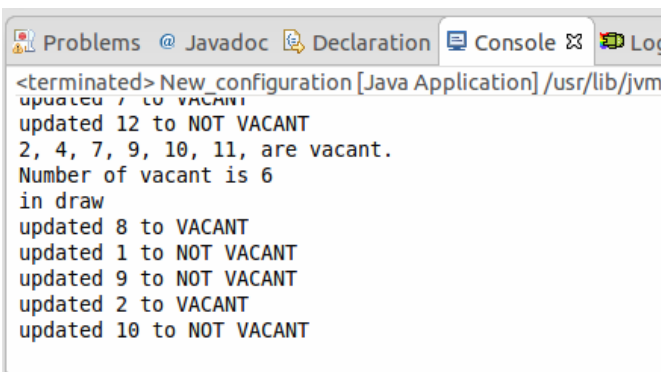


Figure 12 : Snapshot 1 of the admin GUI (monitoring parking status on real-time)

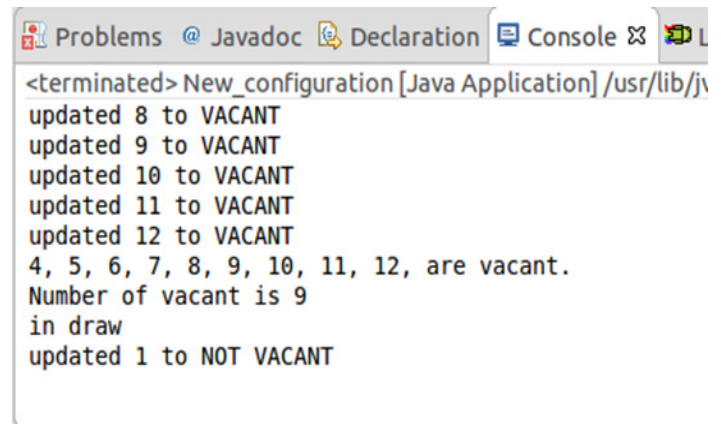


Figure 13: snapshot 2 of admin GUI (monitoring parking status on real-time)

## 18. Discussion

This smart parking system requires cooperation from multiple users searching for parking spaces at the same time, but there is no barrier to the parking spots. In addition, there are some situations where the system may not work. So far, the system is not designed for use at night, as the car park in our scenario is a scarce resource only during office-hours. If night-time operation was required, night vision cameras or artificial illumination would have to be used. [1]

To create a potentially successful parking locator system, we first considered user needs and requirements [15]. Users are not a homogenous group, and an individual user's goals can change rapidly. The heterogeneity of users may range from residents/workers, short/long-term visitors, deliveries, services and emergency access. As a whole, users may engage in a range of disparate and dynamic activities and roles. In addition, accidents and temporary disruptions that may render dedicated parking spaces temporarily unusable, which presents the problem of getting the system quickly updated to reflect any changes.

In addition, if a car is parked across two spaces, then the system may detect this as two separate shapes, neither of which is the correct shape to be a car and may be ignored [1].

## 19. Evaluation and Testing Our Prototype upon Empirical Study via Qualitative Research

In particular, intended smart parking solutions for urban environments necessarily and primarily need to involve all the stakeholders in an early phase of the solution development. The stakeholders involved in the UCD process were in general enthusiastic, and they were happy to be able to contribute to the system development. That is in line with research that found that users who are involved in the process indicated that the system was designed to meet their needs. The prototype was evaluated and tested in two iterations using the UCD methodology. The first iteration on the UCD process was informed by 15 participant interviews. This qualitative study builds on a case study to propose a methodological framework for the implementation of a smart parking IoT system.



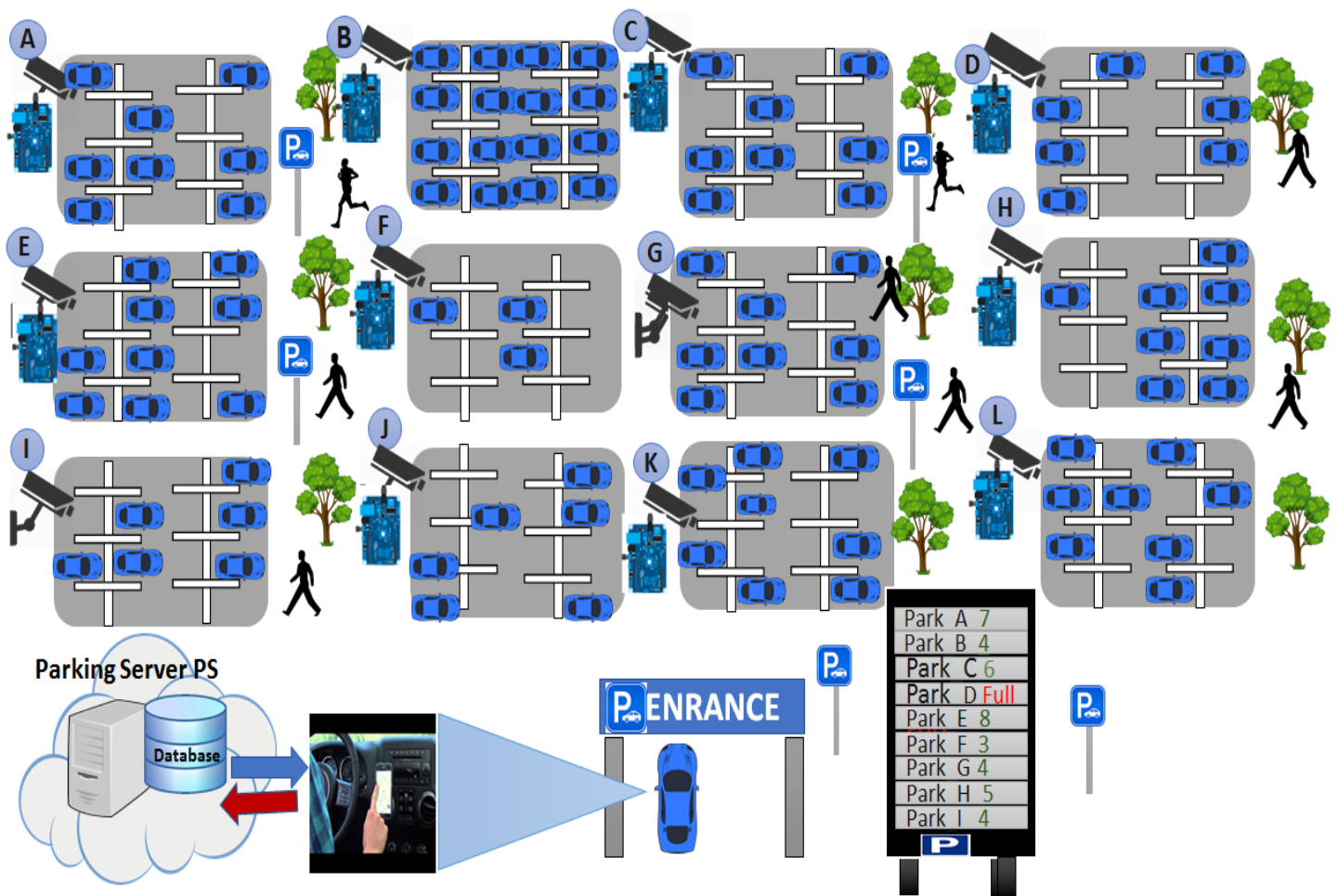


Figure 14: General outline of the IoT prototype using different nodes

The majority of participant stakeholders were impressed by the prototype and believed that the system would be useful, at least to alleviate the issue of finding parking spaces. They believed that even if the system is able to tell the users that there are no vacant spaces available, which would be useful information instead of the driver wasting time and fuel searching for a space when none is available. However, they had a variety of ideas regarding how to improve the prototype. We took this into consideration in the figure (9) above as a general prospective for our IoT parking system prototype.

## 20. Empirical Study and Ideas from Stakeholders Used to Improve the Prototype:

Our findings from the empirical data indicate that there is a real need for a smart parking system to alleviate the issues when finding a parking spot. Some stakeholders suggested letting all drivers know about the fully occupied parking areas via large information screens while others suggested using small screens connected to our prototype instead of the apps at each parking space/area to help the drivers avoid crowding and congestion. In addition, some recommended utilising large screens at each entrance to the university to provide information to the drivers. Also, interviewees recommended enabling the user to click on a

button to confirm that he/she has arrived. Furthermore, stakeholders suggested applying rules about carpooling to convince the students and staff to set up a car-sharing system (a website would be created to support this). Moreover, some suggested using the system to cover all the parking spaces to let the driver know about the parking status. Besides, some stakeholders advised strongly against using a reservation option in the app. (It should be used to get information about the status of parking spaces, they believed, so they need it only to allow them to go directly to the less occupied ones). And to make the system more accurate, there was a suggestion from stakeholders regarding security staff - many stakeholders suggested that there was a need for crew safety, which are employed by the Buildings and Estates department to intervene if a driver finds someone has taken his/her place.

Finally, the ultimate problem may emerge from users, whose mistakes or deliberate ignorance may challenge the system. This raises the question of whether the system serves the users or if the users are serving the system. While it is technically possible to design a system that leaves almost no room for a user's ad-hoc behaviour, such a system may be too complex to be deployed (e.g. physical barriers preventing entry or exit from unauthorized individual locations) [1].

## 21. The Final Visualization of our Model System

According to the qualitative research with stakeholders, we can conclude that the system would be beneficial for two cases. On the one hand, it would be useful when the parking area is large and by guiding a driver to vacant spaces. Figure 4 represents the model, and a general perception of the system in the final image was that different size lots are desired so the user can easily recognize the vacant parking space. On the other hand, it is also beneficial in a case when a parking area is small and active.

### Conclusion and Future Work

To conclude, the objectives of this research have been achieved. The aggravation involved in searching for vacant parking spaces can be minimised by informing the driver about the status of parking areas. Although much work has been done on smart parking, the work reported in this paper is based upon IoT cams (nodes). We propose a novel architecture for low cost and flexible smart parking using CCTVs (nodes). The system is intended to use previously installed CCTV cameras at any parking space instead of using dedicated systems, which would be more expensive.

Upgrades are currently being done to ensure better reliability and scalability for the proposed system. Some adjustments are being done which focus on making the algorithms effective to cover more parking spots. More iterations of UCD will be applied to the process regarding evaluation of our prototype in order to improve the quality of service. Also, In our future work, we will consider the security aspects of our prototype system as well as implement the system in large scales in real environments.

### Conflict of Interest

The authors declare no conflict of interest.

### Acknowledgment

The authors would like to thank our colleagues in Interaction Design Centre- IDC in UL for their support in pursuing this work. We also would like to acknowledge the support of the Ministry of Higher Education and Scientific Research-Tripoli for their funding of the project. Also, part of the article is taken from our previous work as referenced in [1].

### References

- [1] This paper is an extension of work originally presented in WF-IoT Conference By M. Fraifer and M. Fernström, "Smart car parking system prototype utilizing CCTV nodes: A proof of concept prototype of a novel approach towards IoT-concept based smart parking," in *Internet of Things (WF-IoT)*, 2016, IEEE 3rd World Forum on, 2016, pp. 649-654.
- [2] L. Foschini, T. Taleb, A. Corradi, and D. Bottazzi, "M2M-based metropolitan platform for IMS-enabled road traffic management in IoT," *Communications Magazine*, IEEE, vol. 49, pp. 50-57, 2011.
- [3] Z. Pala and N. Inanc, "Smart parking applications using RFID technology," in *RFID Eurasia*, 2007 1st Annual, 2007, pp. 1-3.
- [4] V. W. S. Tang, Y. Zheng, and J. Cao, "An intelligent car park management system based on wireless sensor networks," in *Pervasive Computing and Applications*, 2006 1st International Symposium on, 2006, pp. 65-70
- [5] J. P. Benson, T. O'Donovan, P. O'Sullivan, U. Roedig, C. Sreenan, J. Barton, et al., "Car-park management using wireless sensor networks," in *Local Computer Networks*, Proceedings 2006 31st IEEE Conference on, 2006, pp. 588-595.

- [6] S. Funck, N. Mohler, and W. Oertel, "Determining car-park occupancy from single images," in *Intelligent Vehicles Symposium*, 2004 IEEE, 2004, pp. 325-328.
- [7] R. Panayappan, J. M. Trivedi, A. Studer, and A. Perrig, "VANET-based approach for parking space availability," in *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*, 2007, pp. 75-76.
- [8] A. Kianpisheh, N. Mustaffa, P. Limtrairut, and P. Keikhosrokiani, "Smart Parking System (SPS) Architecture Using Ultrasonic Detector," 2012.
- [9] S. Mathur, S. Kaul, M. Gruteser, and W. Trappe, "ParkNet: a mobile sensor network for harvesting real time vehicular parking information," *Proceedings of the 2009 MobiHoc S*, vol. 3, p. 3, 2009.
- [10] N. Hanif, M. Badiozaman, and H. Daud, "Smart parking reservation system using short message services (SMS)," in *Intelligent and Advanced Systems (ICIAS)*, 2010 International Conference on, 2010, pp. 1-5
- [11] K. C. T. a. K. C. L. D.B.L. Bong, "Integrated Approach in the Design of Car Park Occupancy Information System (COINS)."
- [12] H. L. Andreas Klappenecker , , Jennifer L. Welch, "Finding available parking spaces made easy," journal homepage: [www.elsevier.com/locate/adhoc](http://www.elsevier.com/locate/adhoc), 2012.
- [13] S. Jian, K. S. Yang, and C. L. Lee, "Modular RFID parking management system based on existed gate system integration," *WSEAS Transactions on Systems*, vol. 7, pp. 706-716, 2008.
- [14] S. Lee, D. Yoon, and A. Ghosh, "Intelligent parking lot application using wireless sensor networks," in *Collaborative Technologies and Systems*, 2008. CTS 2008. International Symposium on, 2008, pp. 48-57.
- [15] T. Lowdermilk, *User-centered design: a developer's guide to building user-friendly applications*: " O'Reilly Media, Inc.", 2013.