DISSENY I IMPLEMENTACIÓ D'UN SISTEMA DE COBRAMENT CONTACTLESS EN UN EXPENEDOR ATOMÀTIC

Treball de Final de Grau
Presentat a

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

per

Petar Hlad Colic

En requeriment dels
estudis del grau en

Ciències i Tecnologies de la Telecomunicació

Tutor: Marcel Fernández Muñoz

Barcelona, Setembre 2016

ABSTRACT

RESUM

Aquest document presenta el disseny i el desenvolupament d'un sistema de pagament sense contacte en un expenedor automàtic. Mitjançant un ordinador de placa petita per controlar els motors de l'expenedor i per poder llegir targetes de radiofreqüència aquest projecte es centra en el desenvolupament del software que controlarà el sistema d'accionat de l'expenedor i també ...

RESUMEN

"Agraïments"
Autor Frase

RECONEIXEMENTS

I would like to thank to all the GPI team –Javier Ruiz, J. R. Casas, Albert Gil and Carles Ventura– for their support during this half year of work. Especially, I would like to thank Javier Ruiz for advising this TFG project and J.R. Casas for giving me the opportunity to work in this project.

També voldria agrair als meus pares la seva infinita paciència amb les llargues tardes davant l'ordinador, i la seva ajuda quan més falta feia. A la meva parella pel seu suport, tant matemà tic ccom del meu humor. I als meus companys de classe, amb els qui encara tenim unes braves pendents.

ÍNDEX

CAPÍTOL 1

INTRODUCTION

Els expenedors automàtics de begudes fredes tenen una vida útil bastant llarga.

Cada dia el nombre de transaccions amb efectiu es van reduïnt. Els diners en efectiu ocupen espai i, sobretot les monedes, pesen molt. És per això que les transaccions s'encaminen cap al *cashless* (en anglès, sense efectiu).

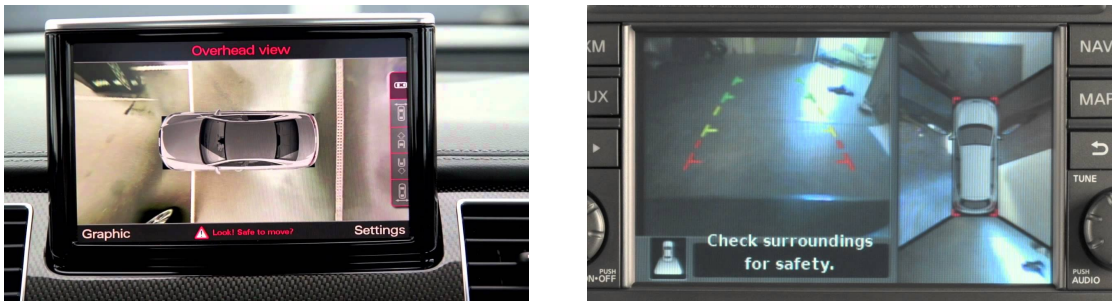
Technological developments in Electronics and Signal Processing fields are also being implemented in the nowadays manufactured vehicles. A considerable number of the vehicle functions have moved to be controlled by electronic actuators. This is the case of, for example, fuel valves, brakes, door and windows opening/closing, etc. In addition, most vehicles are including an on-board computer that monitors all the vehicle parameters –speed, remaining fuel, range, etc.–, and inform the driver about any detected potential danger. All those systems that help the driver and made a safer driving are known as DAS, Driving Assistant Systems.

Recently, a new group of technological developments for the automotive industry has sprung: the Advanced Driving Assistance Systems (ADAS). This category embraces those advanced systems that offer the driver a guidance about the environment of the vehicle. Some known examples are parking assistant –ultrasound system that alerts the driver when the rear distance to an object is below a certain limit–, lane departure warning -alerts the driver when any wheel crosses a lane delimiter line-, traffic assistance –the vehicle detects and interprets the traffic signals–, among others.

This project will focus on a specific ADAS: 360° vision system. These device range offer the driver a complete top view –or bird's eye view– of the vehicle surroundings, allowing the driver to see through blind spots around the vehicle. The most common set-up includes several cameras located around the vehicle, and a display where the driver can view a composite image made from the camera data.

360° vision systems, also called bird's eye systems, are currently offered in high-end passenger cars. This systems can be combined with other features such as lane departure warning or pedestrian detection. Figure 1.1 shows two screenshots from different commercial 360° vision systems.

However, 360° vision system for passenger cars can not be straightforwardly applied to all vehicle types. The camera amount, its position, the vehicle length and height and the ROI (Region of Interest) make this algorithms specifically adapted to a vehicle



(a) Audi S8 – 360 View Camera (b) Nissan Rogue – Around View Monitor

Figura 1.1: Two commercial 360° vision systems display view

type. Thus, the aim of this project will be to develop an 360° vision systems for long vehicles, specifically buses.

This vision system is an specific part of the work commissioned by the company Arcol to a multidisciplinary UPC team. Arcol is a mirrors and accessories manufacturer for motorhomes, buses and coaches. Currently, this company is working jointly with the UPC in a 3-year project. This project main goal is to develop a camera-and-display based guidance system for the driver.

This project is carried out jointly in two UPC research groups: The Image Processing Group (GPI) and the Advanced Hardware Architecture (AHA) group. The AHA group main tasks are, on a high profile, to develop all the hardware to obtain and process the signals. The Image Processing Group is responsible of the computer vision algorithms development. During this project first year, the AHA group is developing the recording/display drivers and programming the hardware accelerators needed. Meanwhile, the GPI group is developing the 360° vision system and a pedestrian detection system.

This TFG project contains the part of the work developed in the Image Processing Group developing the 360° vision system, focusing on the image warping algorithms. Thus, this project also develops several parts corresponding to other stitching stages. Stitching algorithm parts developed by other GPI team members, and therefore outside the scope of this project, are only briefly stated for a whole stitching algorithm understanding.

Project goals

The work done in the Image Processing Group started on February 2014. As stated on the Project Proposal and Workplan delivered on February 14th, this TFG work will end at June 30th, although the work in the ARCOL project will be still in development until December 2014.

This project goals had been defined before the development began, and are stated in the Project Proposal and Workplan mentioned above. The goals of this project can be summed up in the following points:

Representative points automatic detection. To perform the stitching a reference points set had to be highlighted on the image. One of this project goals is to define this points and the model shown to each camera on the calibration step.

Estimate the warping parameters. Study the most suitable transformation and the parameters needed to deform the input images to obtain the output stitched image.

Blending the results on the final stitching. Define an algorithm to merge the warped images on the final composition.

Follow the requirements stated by the Arcol project. All the procedures stated before have to be done according with the Arcol UPC project requirements. This requirements and specifications can be found in Chapter **??**.

The work done in this five months has been focused on developing a proper warping algorithm for this specific environment, taking into account all the technical an logistic issues. The specific tasks that had been included in this project scope are explained in Chapter **??**.

Time Plan

The time plan followed during this project development is shown in Figure 1.2 Gantt Diagram. The developed work has been split in 8 work packages. A detailed description of each work package and the internal tasks developed can be found on Appendix **??**.

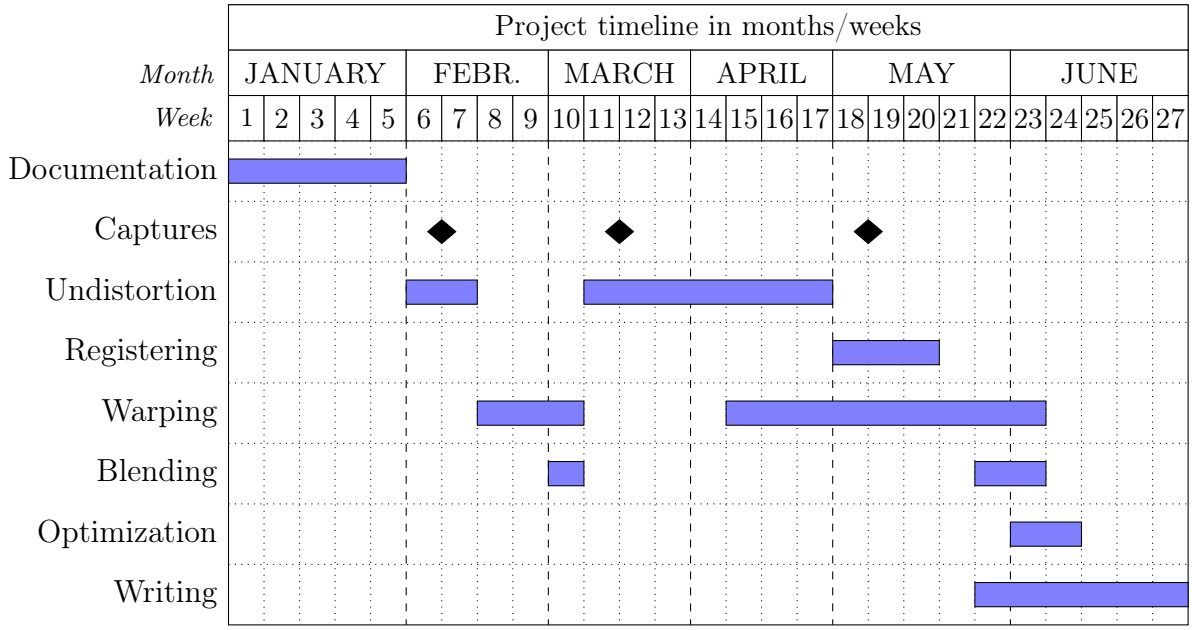


Figura 1.2: Project Gantt Chart

CAPÍTOL 2

REQUERIMENTS

L'objectiu del projecte és dissenyar un sistema que es pugui integrar en expenedors automàtics existents i permeti la compra de begudes sense necessitat de fer ús de diners en efectiu en el moment de la compra.

Requeriments generals de la solució

Millors prestacions La solució no pot empitjorar les prestacions que tenia l'expenedor automàtic en l'estat original. Només n'ha de millorar.

Reversibilitat En cas de necessitat, s'ha de poder retrocedir en la configuració de l'expenedor automàtic i deixar-lo en l'estat original.

Seguretat El sistema ha de ser prou segur per evitar frau o robatoris de diners.

Per assolir aquests requeriments el projecte es dividirà en tres parts:

Aplicació de Servidor És el sistema central de la solució. Serà la part que gestionarà les compres, transaccions monetàries, usuaris.

Aplicació de Client És el sistema que controlarà directament l'expenedor. Quan l'usuari s'identifiqui i faci una sel·lecció, l'aplicació es comunicarà amb el sistema central perquè autoritzi la compra i pugui fer que l'expenedor serveixi una beguda.

Implementació i integració del hardware És tota la part del projecte que tracta la part física del projecte. En altres paraules, la integració del dispositiu que contindrà l'aplicació de client, el/s dispositiu/s que servira/n per identificar l'usuari i tots els circuits elèctrics per controlar l'expenedor i el seu estat.

Un cop dividit el projecte en tres blocs, perquè la solució sigui prou segura contra fraus, es proposa que el sistema sigui un sistema de prepagament on el saldo de cada usuari l'emmagatzema i gestiona el sistema central. També es proposa que l'usuari s'identifiqui mitjançant una etiqueta NFC.

Requeriments específics

Aplicació de Servidor

Per tal de tenir una solució que compleixi els requeriments generals, es proposen una sèrie de requisits específics per a l'aplicació de servidor:

Gestió d'expenedors Ha de tenir control sobre els expenedors. Ha de gestionar quin producte tindrà l'expenedor a cada carril i a quin preu. Ha de poder deshabilitar la venda d'un cert carril. Ha de poder visualitzar l'estat en el que es troba un expenedor (s'ha quedat sense monedes per donar canvi, s'ha avariat, s'ha quedat sense producte ...).

Gestió d'usuaris Ha de poder crear nous usuaris, eliminar-ne, modificar-ne les dades. També ha de poder identificar un usuari a partir d'algun identificador amb què aquest s'identifica quan fa ús d'un expenedor.

Gestió del saldo disponible Ha de control·lar el saldo disponible del que disposa cada usuari. L'ha de poder consultar, modificar-lo quan hi ha un pagament o una recàrrega de saldo.

Gestió de les compres Ha d'autoritzar la compra que fa un usuari tenint en compte el seu saldo disponible i la disponibilitat del producte a l'expenedor.

Seguretat en les transaccions El sistema ha de ser segur en termes de transaccions de diners i rècarregs de saldo per compra. S'han d'evitar suplantacions d'identitat, intrusions en el sistema, fraus i demés posibles accions que perjudiquin al propietari de l'expenedor o als usuaris.

Detecció i gestió d'incidencies En sistemes que involucren parts mecàniques, elèctriques, i parts que es comuniquen per xarxa, no és tan extrany que en certs moments les coses no funcionin com haurien de funcionar. Així el sistema hauria de ser capaç de detectar la gran part d'incidències produides per errors de comunicació o fallades en l'expenedor i solucionar-les en la mesura que sigui possible.

Aplicació de client

De la mateixa manera que amb l'aplicació de servidor, es proposen un seguit de requeriments específics per l'aplicació de client:

Control sobre l'expenedor Ha de poder controlar els circuits de l'expenedor per exendre llaunes. També ha de poder analitzar els paràmetres dels circuits per a poder preveure la quantitat de producte restant que conté l'expenedor i a més detectar i evitar fallades.

Comunicació amb el sistema central S'ha de comunicar amb el sistema central per a recollir la informació necessaria per funcionar i per a poder realitzar les vendes. També ha de comunicar al servidor la informació sobre l'estat de l'expenedor.

Interfície fàcil d'usar L'ús de l'aplicació de cara a l'usuari es farà a través d'una interfície gràfica d'usuari que ha de ser senzilla d'utilitzar de la mateixa manera que ho és un expenedor automàtic usual.

Seguretat en les comunicacions Les comunicacions han de tenir un mecanisme per evitar intents de frau i demés accions perjudicials per al propietari de l'expenedor i altres usuaris.

Implementació del Hardware

Tenint en compte els requisits generals de la solució i els requisits proposats per a l'aplicació de client, es proposen un seguit de requeriments específics per a la implementació del hardware dins l'expenedor:

Reversibilitat de la integració Un cop integrat el sistema, en cas de necessitat, s'ha de poder extreure el hardware i deixar l'expenedor en l'estat original. Això vol dir que no es pot alterar irreversiblement el cablejat de l'expenedor.

Volum reduït No pot ocupar molt espai. Ha de caber dins de la porta de l'expenedor.

Implementació modular Ha de ser prou facil de posar i treure de l'expenedor. Fàcil de reparar i reposar blocs defectuosos.

No interferir amb el funcionament mecànic de l'expenedor La implementació no ha d'interferir en el funcionament mecànic usual que té un expenedor com són l'obertura de l'expenedor per recollir les monedes o reposar producte.

Evitar creuament amb una compra amb monedes Es pot donar el cas que es vulgui efectuar una compra sense efectiu i s'hagin introduït monedes a l'expenedor al mateix temps. Ha d'evitar que es produeixin problemes per aquest creuament.

Aïllament físic amb l'exterior Ha d'estar físicament de l'exterior per protegir el sistema i evitar possibles intents de manipulació.

Interfície d'usuari Ha de tenir una interfície de manera que l'usuari pugui visualitzar la informació que dóna l'aplicació de client i també alguna manera d'introduir la seva selecció.

Lector NFC Ha d'incloure un lector NFC per a la identificació dels usuaris.

CAPÍTOL 3

FUNCIONAMENT D'UN EXPENEDOR AUTOMÀTIC DE BEGUDES FREDES

Per a poder dissenyar la solució i poder entendre les millores, primer s'ha d'esbrinar i entendre el funcionament original d'un expenedor automàtic.

Descripció de les parts

Descripció de les parts.

Cas d'ús

Presentem el cas d'ús usual:

1. Un usuari s'acosta fins l'expenedor i observa quins productes estan disponibles i quin preu tenen. Cada botó de selecció té una etiqueta al costat que indica el producte i el seu preu. La disponibilitat del producte la indica un indicador lluminós sota el botó (si emet llum, no està disponible.
2. Un cop l'usuari ha fet la seva elecció, introdueix monedes amb un valor igual o superior a l'import del producte.
3. L'usuari apreta el botó de selecció del producte que ha escollit.
4. L'expenedor retorna el canvi (si n'hi ha) i serveix la beguda que l'usuari ha seleccionat.

Exposem els casos d'ús quan alguna cosa va com no hauria d'anar:

L'usuari apreta el botó d'un producte que no està disponible: No passa res. És com si no hagués apretat cap botó.

L'usuari apreta el botó d'un producte pel qual no té saldo suficient: No passa res. És com si no hagués apretat cap botó.

L'indicador de ïmport exacte"està encès: L'expenedor només accepta monedes del tipus que pot retornar i no retorna canvi.

Circuit elèctric

Circuit elèctric.

CAPÍTOL 4

DISSENY DE SOFTWARE

En aquesta part es presenta el disseny de l'aplicació amb les decisions de disseny explicades.

Per al projecte sencer s'ha escollit programar en llenguatge *Python* perquè és molt senzill de programar i és molt més llegible.

Aplicació de servidor

Per a l'aplicació de servidor s'ha fet servir un *framework* per a desenvolupar webs en Python anomenat Django.

S'ha seguit el patró MVC (Model-Vista-Controlador).

Model de dades

Per a desenvolupar una aplicació web mínimament complexa, primer s'ha de començar dissenyant el model de dades.

Cada *model* en Django representa una taula a la base de dades.

User Cada usuari està associat a un `User` i contindrà les dades que l'identificaran com el seu nom, correu electrònic, nom d'usuari i contrassenya, DNI ...

Credit Conté el saldo disponible de l'usuari i, per tant, està associat a un `User`, i cada `User` té associat com a molt un `Credit`. Està en una taula separada perquè està pensat perquè l'aplicació pugui ser integrada en una base de dades d'usuaris existent.

Card Representa la targeta amb la què s'identificarà l'usuari. Per tant està associada a un usuari. Igual que les targetes de crèdit, un usuari pot tenir més d'una targeta associada.

Vender Representa un expenedor. Conté la informació bàsica de l'expenedor (model, número de sèrie, localització geogràfica), i també la informació sobre el seu estat.

ApiKeyToken Per a garantir un mínim de seguretat en les comunicacions, l'aplicació de client usarà un codi per a identificar l'expenedor des del qual s'està comunicant. Aquest identificador serà únic per a cada expenedor. `Vender` i `ApiKeyToken` estan en taules separades per raons d'implementació de codi d'autenticació en Django.

Column Representa un carril d'un expenedor. Cada carril té un preu de venda i un tipus de producte. També pot estar actiu o inactiu, i també pot estar buit o ple.

Purchase Cada cop que l'usuari fa una selecció i tot va bé (la informació proporcionada per realitzar la compra és correcta i l'usuari té suficient saldo disponbible) es crea una instància de `Purchase` que representa la compra que ha fet l'usuari amb la informació corresponent (targeta amb què s'ha comprat, producte, carril, preu, data i hora, identificador de la compra, i si ha estat pagada). L'identificador de la compra és un identificador aleatori que assigna l'aplicació de client per raons que comentarem més endavant. Tot i que `Column` ja conté la informació de carril, producte i preu, es guarda per separat ja que al llarg del temps aquests paràmetres podrien canviar.

Payment Quan un `Purchase` ha estat creat, s'ha de procedir al pagament de la compra, per tant un pagament està associat directament a una compra. `Payment` representa l'acció de pagar la compra. Inclou el preu de la compra, la quantitat descomptada (si es fa descompte) i el preu final a pagar. També inclou l'usuari a qui se li ha de cobrar, el dia i hora i la informació per determinar si el pagament està pagat i és valid, o si l'han hagut de reemborsar.

RefundPetition Quan un procés de compra es queda a mitges, o l'expenedor pateix una fallada tècnica, o per qualsevol altre motiu és necessari reemborsar l'import d'una compra a l'usuari, primer s'ha de crear una petició de reemborsament. És així perquè d'aquesta manera les peticions es poden generar de manera automàtica mentre que l'autorització de la petició es realitza amb més cura (un sistema més curós o bé amb supervisió humana). Conté l'identificador del pagament que es vol reemborsar, el sol·licitant (ja sigui un empleat o bé un expenedor de manera automàtica), la raó de la petició, l'hora i la data, i la informació per determinar si ha estat denegada o finalment s'ha realitzat el reemborsament.

Refund Representa el reemborsament d'un pagament un cop la petició és acceptada. Així doncs conté l'entitat que ha autoritzat el reemborsament, la petició que ha estat autoritzada i l'hora i la data de l'autorització.

Controlador

El controlador és on es troba tota la lògica del sistema.

Les classes que cadascuna desenvolupa una acció diferent:

GetUserDataFromCard Retorna la informació de l'usuari (nom i saldo disponible) a partir de l'identificador de la seva targeta.

GetVenderColumns Retorna la informació de cada carril de l'expenedor que l'ha demanat (tots els paràmetres que conté el model `Column`).

ProductPurchase Quan l'usuari ha fet la selecció del producte, es comprova que el producte estigui disponible, que el producte del carril seleccionat coincideixi amb

el producte que conté, que el preu també coincideixi i que el saldo disponible de l'usuari sigui suficient. Si tot és correcte, es crea una instància de `Purchase` i el seu `Payment` associat i se li descompta l'import de la compra al saldo de l'usuari. Finalment se li retorna una resposta de que tot ha anat bé a l'expenedor perquè serveixi una beguda o bé se li retorna una resposta d'error informant de què és el que ha anat malament.

RequestCancelPayment Quan alguna cosa ha anat malament a la banda de l'expenedor (s'ha encallat el mecanisme, ha fallat la xarxa, ...) pot ser que s'hagi cobrat la compra a l'usuari i no s'hagi servit la beguda. És per això que cal un mecanisme per demanar reemborsament sota certes circumstàncies. Quan es rep una sol·licitud de reemborsament aquesta s'emmagatzema si fins el moment no existia cap altra sol·licitud associada al mateix pagament.

Vista

La vista en si és la informació que rep l'aplicació de client des de l'aplicació de servidor.

Aplicació de client

Per a l'aplicació de client, com ha de tenir una interfície gràfica d'usuari, s'ha decidit utilitzar `PyQt4` que és una llibreria gràfica. Té integrada la part gràfica i també la comunicació entre fils d'execució.

L'aplicació de client es divideix en 6 grans blocs diferenciables:

comms S'encarrega d'implementar la comunicació amb el servidor.

card_reader S'encarrega de controlar el lector de targetes. Cada cop que detecta una targeta avisa al fil d'execució principal juntament amb l'identificador de la targeta.

column_controller S'encarrega de controlar els mecanismes dels carrils de l'expenedor i de detectar les possibles fallades mecàniques.

db_controller S'encarrega de controlar la base de dades local. En aquesta base de dades s'emmagatzema la informació dels carrils i també la informació de compres fallides per les quals el servidor encara no ha confirmat la creació d'una petició de reemborsament.

graphic_ui : S'encarrega de crear la interfície gràfica d'usuari i de gestionar totes les accions de l'usuari i derivar-les cap el fil d'execució principal.

cocacolero : S'encarrega d'instanciar, controlar i coordinar la resta de blocs. Aquest bloc es troba en el fil d'execució principal

Cas d'ús

A continuació s'exposa l'explicació del cas d'ús usual de l'aplicació per part de l'usuari.

1. L'usuari arriba i apropa la seva targeta al lector.
2. L'usuari és correctament identificat i el seu nom i el seu saldo disponible apareixen per pantalla. Els botons de selecció dels productes disponibles queden habilitats.
3. L'usuari fa la selecció pitjant amb el dit sobre el botó del producte que ha escollit
4. L'usuari disposa de suficient saldo disponible i la compra s'efectua amb èxit. L'expenedor serveix la beguda, es deshabiliten els botons i finalitza el procés.

Exposem el mateix cas d'ús des del punt de vista de l'aplicació.

1. `card_reader` detecta una targeta i comunica l'identificador a `cocacolero`.
2. `cocacolero` demana la informació de l'usuari associat a la targeta detectada a l'aplicació del servidor. El servidor l'identifica amb èxit i retorna la informació demanada (nom i saldo). `cocacolero` habilita els botons de `graphic_ui` dels productes disponibles.
3. Quan l'usuari fa la selecció, `graphic_ui` comunica a `cocacolero` la selecció.
4. `cocacolero` genera un identificador aleatori per a la compra, envia al servidor la petició de la compra. El servidor comprova que la informació és correcta i que l'usuari té suficient saldo. El servidor respon amb èxit. `cocacolero` comunica a `column_controller` el carril que s'ha d'activar i aquest li respon amb èxit quan la beguda s'hagi servit. `cocacolero` deshabilita els botons de `graphic_ui` i finalitza el procés.

Ara exposem els casos d'ús quan alguna cosa no funciona com està previst:

L'usuari no vol realitzar cap compra després d'identificar-se: L'usuari pitja el botó `Log Out` i finalitza el procés.

L'usuari no s'ha identificat amb èxit: S'ignora

Es detecta una targeta durant un procés de compra: S'ignora

L'usuari fa una selecció per la qual no disposa de saldo suficient: el servidor ho detecta i ho comunica a l'aplicació de client. L'aplicació descarta la compra i finalitza el procés.

La informació de la compra és incorrecta: El mateix cas que el de saldo insuficient.

El servidor no pot processar la compra per qualsevol altre motiu: El mateix cas que el de saldo insuficient.

El servidor no respon dins del termini establert: S'anul·la el procés. S'emmagatzema la informació de la compra i s'envien peticions al servidor periodicament fins que el servidor crea la petició de reemborsament amb èxit o la descarta per qualsevol motiu.

L'aplicació detecta una fallada tècnica i no s'ha servit el producte: El mateix cas que quan es perd la resposta del servidor.

CAPÍTOL 5

DISSENY DE HARDWARE

Bla bla