



**Jaypee University of Information Technology**  
**Solan (H.P.)**  
**LEARNING RESOURCE CENTER**

Acc. Num. *SP02038* Call Num:

**General Guidelines:**

- ◆ Library books should be used with great care.
- ◆ Tearing, folding, cutting of library books or making any marks on them is not permitted and shall lead to disciplinary action.
- ◆ Any defect noticed at the time of borrowing books must be brought to the library staff immediately. Otherwise the borrower may be required to replace the book by a new copy.
- ◆ The loss of LRC book(s) must be immediately brought to the notice of the Librarian in writing.

Learning Resource Centre-JUIT



SP02038

SP2038

*SP2038*



# PARKING PAY STATION

By

**ANKIT AGGARWAL - 021403**



**MAY-2006**

**Submitted in partial fulfillment of the Degree of Bachelor of  
Technology**

**DEPARTMENT OF COMPUTER SCIENCE AND  
INFORMATION TECHNOLOGY**

**JAYPEE UNIVERSITY OF INFORMATION  
TECHNOLOGY-WAKNAGHAT**

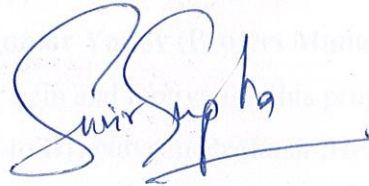


## CERTIFICATE

This is to certify that the work entitled, "PARKING PAY STATION" submitted by ANKIT AGGARWAL in partial fulfillment for the award of the degree of Bachelor of Technology in Information Technology of Jaypee University of Information Technology has been carried out under our supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.



Mr. Mukesh Yadav  
**Project Manager**  
**HCL Technologies**



Mr. Suvir Gupta  
**Project Leader**  
**HCL Technologies**



## ACKNOWLEDGEMENT

Any successful project consumes dedication, cooperation, coordination and several resources like knowledge, energy and time. The credit of these things that I had got partially in completing our project approaches.

I am grateful to the **HCL Technologies Ltd.** for giving me an opportunity to work for them and be a part of such a big project and providing me a very conducive environment to work in. My sincere obligations go to **Mr. Parveen Kumar Malik (Group Project Manager)** who has been a constant support to me during the course of my training. I would also like to thank **Mr. Mukesh Kumar Yadav (Project Manager)** and **Mr. Suvir Gupta (Technical Lead)**, without their help and motivation this project would not have been possible. I am extremely thankful to **Dr. Naveen Prakash ,Head Of Department (Computer Science and Information Technology)**, who has always put forward his due to support students. I am equally indebted to **Mr.Satish Chandra.** He always extended his kind support to me.

Acknowledgements are also to my family, friends and all those people who have been continuous support to me in the successful completion of this project.



## TABLE OF CONTENTS

<b>PROFILE OF HCL TECHNOLOGIES</b>	<b>5</b>
<b>LIST OF ABBREVIATIONS</b>	<b>7</b>
<b>ABSTRACT</b>	<b>8</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>9</b>
<b>CHAPTER 2 PROJECT PLANNING</b>	<b>12</b>
<b>CHAPTER 3 SOFTWARE REQUIREMENT SPECIFICATIONS</b>	<b>19</b>
<b>CHAPTER 4 CONCEPTUAL AND PHYSICAL DESIGN</b>	<b>31</b>
<b>CHAPTER 5 SYSTEM TESTING</b>	<b>34</b>
<b>CHAPTER 6 I/O SCREENS</b>	<b>36</b>
<b>CHAPTER 7 CONCLUSION</b>	<b>38</b>
<b>CHAPTER 8 SCOPE OF DEVELOPMENT</b>	<b>39</b>
<b>GLOSSARY</b>	<b>40</b>
<b>BIBLIOGRAPHY</b>	<b>42</b>

## Profile of HCL Technologies

### Who is HCL Technologies Ltd.?

HCL is one of India's leading global IT services and product engineering companies, providing value-added, software-led IT solutions and services to large- and medium-scale organizations

### History

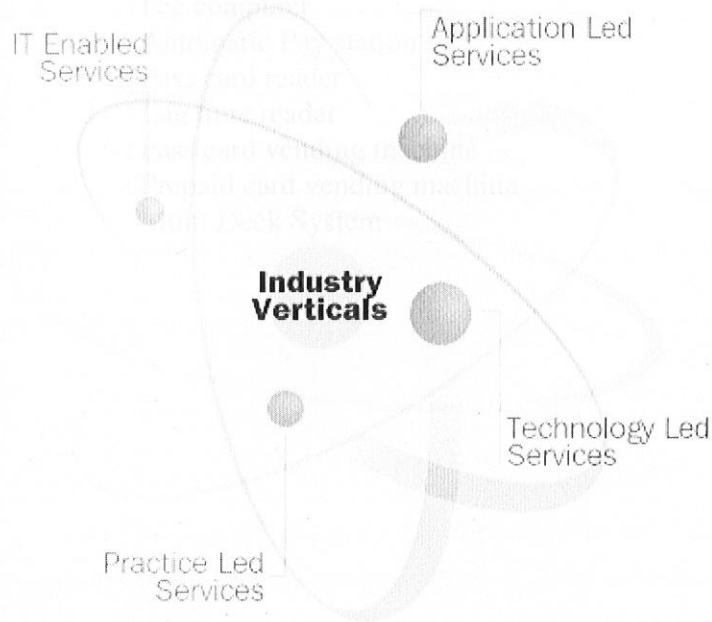
HCL's history goes back to 1975 when six entrepreneurs- Shiv Nadar and five of his associates, decided to systems and services business in India. They opened their first office in the suburbs of New Delhi, naming their business Hindustan Computers Ltd in 1976.

By 1985, HCL was heavily involved in UNIX technology, porting UNIX using the AT&T binaries. UNIX add-on products such as SORT and Checkpoint Restart were soon added to the company's technology development portfolio. In 1987, HCL became involved with ARCNet-based LAN hardware and software. The 1990's saw a rapid rise of software development services within HCL. The R&D division took a lead role in providing those services, using the offshore development model. This resulted in the emergence of HCL's software factories and geographic divisions as key corporate entities, serving clients around the world. HCL Technologies expanded its sphere of influence through HCL Perot Systems, a joint venture with Perot Systems Corporation; its 100% subsidiary, HCL Comnet Systems & Services Ltd.; DSL Software Ltd, a joint venture with Deutsche Bank AG; HCL Enterprise Solutions, Inc., a joint venture between HCLT and Computech Corporation; and HCL E Serve Technologies Ltd, a 100% subsidiary of HCLT.



Today, HCL Technologies is a global information technology enterprise headquartered at Noida, India and employing 8,748 people in various locations across India, U.S., Europe, Asia Pacific and Japan. Additionally, HCL Technologies has 16 fully operational state-of-the-art software development centers.

## Services



Currently, HCL Technologies serve the global market through its group companies that have offices in the USA, Europe, Asia Pacific, Japan and India. It has state-of-the-art technology; client/server and mainframe applications development centers in India and is focused on high-end, high value-added services and offshore-centric development in emerging technologies.

## LIST OF ABBREVIATIONS

CPS	: Central Pay Station
EPS	: Exit Pay Station
CRN	: Credit Note
OOO	: Out of Operation
TD	: Ticket Dispenser
PS	: Pay Station
C/W	: Magnetic stripe encoder
T/M	: Fee computer
F/C	: Automatic Pay station
P/R	: Pass card reader
LR	: Lag time reader
P/V	: Pass card vending machine
P/P	: Prepaid card vending machine
MDS	: Multi Deck System



## ABSTRACT

Welcome to the **PARKING PAY SYSTEM**. The purpose of the software is to manage huge parking lots with multiple floors and lanes. The software that has been developed has provided great benefits to the owner as well as the user of the parking lot. The major contribution that has been made by the software is that the whole process, right from the moment a vehicle enters the parking lot until it exits, is automated. The objective of the system is to provide an automated management of a parking lot and to enhance the service quality for the customers using the parking lot.

The complete software developed in Visual C++6.0 (Win32 as SDK) using windows 2000 platform. The whole software runs on windows CE. Windows CE (sometimes-abbreviated WinCE) is a variation of Microsoft's Windows operating system for minimalistic computers and embedded systems. It is supported on Intel x86 and look-alikes, MIPS ARM, and Hitachi SuperH processors.

Windows CE is optimized for devices that have minimal storage — a Windows CE kernel may run in under a megabyte of memory. Devices are often configured without disk storage, and may be configured as a "closed" system that does not allow for end user extension (for instance, it can be burned into ROM). Windows CE conforms to the definition of a real time operating, with a deterministic interrupt latency. It supports 256 priority levels and provides for priority inversion. Unlike UNIX-like operating systems, the fundamental unit of execution is the thread, providing for simpler, faster concurrent programming.

## CHAPTER I

### INTRODUCTION

---

#### **Purpose:**

The purpose of the software is to manage huge parking lots with multiple floors and lanes. This software has provided great benefits to the owner as well as the user of the parking lot. The major contribution that has been made by the software is that the whole process, right from the moment a vehicle enters the parking lot until it exits, is automated. The objective of the system is to provide an automated management of a parking lot and to enhance the service quality for the customers using the parking lot.

The system will provide the client with the following benefits:

- **Inventory management:** The user will be able to know the number of vehicles in the parking lot at any point in time.
- **Cash management:** The owner of the parking lot can see the daily/weekly/monthly records in order to verify the cash given by the attendants (in case of Manual mode).
- **24X7 services:** The parking service can be easily provided round the clock using this system.

Let us enumerate the contributions made by the software:

- **Automated vehicle detection:** As soon as a vehicle is detected at the entry of the Parking lot, the gate is opened for the entry. A ticket is also dispensed from the ticket dispenser.
- **Monitoring vehicle exit:** The system monitors the exit of the vehicle. When a vehicle is detected to be exiting, the user is asked to insert the parking ticket. Thereafter, the fee calculation is done automatically by the Pay Station and after the payment of fee by the user the barrier gates are opened.



- **Generation of Business Totals:** Business totals are generated either on a regular basis or whenever required by the management. It includes the statistics like the daily/weekly/monthly revenue, the peak hours when number of vehicles parked is maximum etc.
- **Inventory Management:** The number of vehicles that are there in the parking lot can be ascertained and on the basis of that the management can efficiently manage the parking lot.
- **Automated acceptance of fee:** The fee that is to be paid by the user can be accepted by the Pay Station in the form of currency like Bank notes and coins. Discount ticket, credit cards and debit cards can also do payment. Hence, it makes it fast and convenient.
- **Automated dispensing of ticket and balance:** The tickets and the balance can be dispensed to the customer by the Pay Station. The balance can be dispensed to the customer in the form Bank notes and coins. If the balance is not sufficient then credit note issued to the customer, which can be claimed.
- **Emergency condition:** In case of emergency strong Alarm will be raised by the Pay station. For example in case of 'Fire', the barrier gates can be opened very quickly so as to evacuate the parking lot and to minimize the loss.

## Overview:

Organization Name	:	HCL Technologies Limited
Client Name	:	(Not to be disclosed)
Client Location	:	Japan, Korea, US
Project Type	:	Development Project
Project Categorization	:	Large Scale Project
Project Manager	:	Mr. Mukesh Kumar Yadav

The client is a leading solution provider for automated parking systems. The management of the parking lot is taken care of using several Pay Stations located in and around the Parking lot itself. Broadly, from the user's perspective, the system does the following things:

- Opening the Gates of the entrance of the parking lot on detecting a vehicle.
- Automated calculation and display of the parking fee.
- Automated acceptance of the parking fee. This includes the acceptance of Coins or bank notes, prepaid cards, credit cards etc.
- Cancellation of payment at any point of time during the transaction.
- Generation of receipt.
- Generation of Exit ticket or opening of barrier gate on completion of the payment.

The tickets generated by the system are in the form of thick paper bearing a magnetic strip that contains the information about the vehicle parking. Using the information recorded on the magnetic strip, the fee is calculated.

## CHAPTER II

### PROJECT PLANNING

---

#### **Introduction:**

The objective of software project planning is to produce a framework that enables the Project Manager to make reasonable estimates of the resources, cost and schedule. Such estimates made within the certain period at the start and should be update regularly with in project growth. In addition, estimates should attempt information to define best and worst case scenarios. The planning objective achieved by discovering information leading to reasonable estimates.

The major issues the project plan addresses are:

#### **Project Estimation:**

For a given set of requirement it is desirable to know how much time and cost it takes to develop the software. These estimates needed before the development initiated. The size of a program is an indicator of the effort and time required to develop the program. In other words, the size of the program indicates the development complexity. Estimating the size is fundamental to estimating the effort, time and cost of a planned software project. Effort estimation is very curial for successful execution for any project. A good and accurate estimation is the key to control project escalation – time overruns and cost overruns. The estimating process is an integral part of the overall project planning.

#### **Estimation Methods:**

It will be under the discretion of the person doing the estimation which technique should be use. However, following recommended guidelines may be used while selecting the estimation technique.

- If the project scope includes the requirement analysis phase, Function Point Analysis (FPA) technique may be used.



- In case of conversion type of project, Line of code (LOC) technique can be used for project size estimation.
- If the project scope includes the coding phase and exclude the high level design and low level design phase, it will be under the discretion of the person doing the estimation to estimate the time required to code and test each module of the application being developed and to calculate the effort required to complete the project.
- If none of the aforementioned method is applicable for estimating the size of the project size, a suitable size estimation method devised and its rationale explained. The basis for arriving at the model (which could be pilot project data, data related to similar prior projects stored in process database etc.) needs to be given. The types of future projects for which this model could be re-used needed to be laid down.

The following techniques for the effort estimation for the development of the software system.

#### **Estimation by WBS:**

The product to be estimated is decomposed into Work Breakdown Structure (WBS) elements and the effort calculated for each of these elements. The total effort needed is the sum of all these elements. Therefore, this method called the “effort by elements method”. In WBS, the project requirements are repeatedly broken down into the detail requirements to arrive at the components comprising that system.

#### **How to identified task**

- In WBS repeatedly detail out the requirements into subtasks until each task is about one basic unit of reference (person month or person week) or decompose the requirements into the components. Certain task can fix in a project. These already identified in the template. Task are listed under the WBS can be used as decamped requirements.

- For each decomposition unit, identify similarities with functional unit of previous developed systems and use any historical effort data available from these completed systems.
- For decomposition unit not strongly related to those previous projects, use personal experience to estimate the effort of units.
- From the effort estimate (in person month or person week) for the entire project by adding the effort estimates for all the decomposition units.
- Certain predefined tasks are fixed with the project.

#### **How to use WBS sheet:**

- Details requirements into sub system / sub tasks on a separate sheet.
- Divide each subtask into SDLC phase of requirement analysis / HLD / DD /ERS/Testing /Coding. This will also help in sub dividing the tasks.
- Adding up the effort in all the activities will specify the effort for the entire project.
- Now for each component identified list down the tasks and functionality of that component. (These can also be contribute to the trace- ability matrix later)
- For each task identify the inputs, outputs and other function points. This data can be directly be used in the tool.
- The requirement analysis phase should absorb the study phase of the project.
- For maintenance type of projects each change request is identified as a component.

#### **SOFTWARE QUALITY ASSURANCE PLANS:**

Software quality assurance plans is the planned and systematic set of activities to ensure that software process and software products conform to organization requirements, standards, and procedures. Process includes all of the activities involved in designing, developing, enhancing, or maintaining a product. Products include the software, associated data, documentation, and all supporting project paperwork supporting product delivery.

### **Defining Quality Assurance:**

A planned and systematic pattern of all actions necessary to provide adequate confidence that the project, or product, conforms to established technical, schedule, financial, and contractual requirements. The intent is to build-in quality throughout the project or process. The goal, at the conclusion of the project, is to ensure that the product or system meets stated commitments of budget (i.e., fiscal and all other resources), schedule, functional, and technical results. Practices and procedures for developing products and conducting projects must ensure that satisfactory results will be achieved in a consistent, predictable, reliable, and measurable manner.

### **Objectives of Quality Assurance:**

- Offer improvements for the methods of ensuring software quality,
- Provide assurance that time and financial constraints are accommodated.
- Provide assurance that customer requirements will be met,
- Identify technical deficiencies for resolution,
- Identify risks for mitigation,
- Identify and implement other necessary changes to the development process.
- Enumerate successes or novel practices that may be carried forward into future projects.
- Utilize techniques for environmental, unit, integration testing.
- Reduce the time and cost of software testing while improving quality.

Quality Assurance provides project oversight on all aspects of information technology projects. The Quality Assurance staff monitors, evaluates and measures the progress of major project development. Performance of these functions ensures operational improvements and reduces risk so that quality projects are delivered and all critical project deadlines are met.



### **Risk Management:**

Risk involves the possibility of suffering harm or loss- chances of uncertain occurrences that will adversely affect project objectives. Risk management includes the process concerned with identifying, analyzing and responding to risk factors throughout the life of the project.

### **Method of Risk Management:**

The activities of the risk management are:

- Risk identification
- Risk Analysis
- Risk Prioritization
- Risk Planning
- Risk Monitoring

At the proposal stage macro level, risks are identified and analyzed and in the proposal review checklist a section on risk assessment is include. At the beginning of the planning phase identification and detailed analysis of the risks associated with the project is done and documented. These risks reviewed both periodically and on event driven basis throughout the project life cycle and the status has been updated. New risks identified in the tracking stage have to be analyzed, documented and monitored from on them.

### **Risk Identification:**

Before we can identified the right risks that might occur during a software project, it is important to identify all risks that are obvious to both managers and practitioners. It is also possible to categories risks in many different ways. At macroscopic level Project risks, Technical risks and Business risks can defined. One of the best methods for understanding each of the risks is to use a set of questions that will help the project planner to understand the risk.

### Risk Analysis:

The purpose of this step is to analyze the probability of the identified risks occurring and its impact on the project. The following scales is used for rating the probability of risk.

**Table 3.1 Probability and probability Rating Table**

Probability of Occurrence	Probability Rating
Less than 30 %	1
Less than 50 % But Greater than or equal to 30 %	3
Less than 70 % But Greater than or equal to 50 %	5
Less than 90 % But Greater than or equal to 70 %	7
Greater than 90 %	9

### Risk Prioritization:

The higher the impact and probability of occurrence, the higher is the risk priority and the need of action. The risk is prioritized according to the following matrix, which gives the **Index of risk expectation**.

**Table 3.2 Risk Prioritization based on Risk Index**

Probability \ Normalized Impact	0.1	0.3	0.5	0.7	0.9
1	0.1	0.3	0.5	0.7	0.9
3	0.3	0.9	1.5	2.1	2.7
5	0.5	1.5	2.5	3.5	4.5
7	0.7	2.1	3.5	4.9	6.3
9	0.9	2.7	4.5	6.3	8.1

The need for action required (with regard to each risk index) is as follows:

- Extreme: This risk requires immediate attention, and both contingency planning and mitigation planning must be planned and implemented.
- High: Preventive solutions must be searched for and evaluated.
- Medium: Additional QA steps must be planned. If the risk is realized related risk management technique must be implemented.
- Low: The risk must be monitored.

### **Risk Planning:**

After determining the risks and their priority, the resulting preventive or corrective steps must be planned. The extend of the activities to be planned depends on the priority assigned. First identify the cause of the risk and remove or reduce the cause. Then consider the alternate strategies to avoid the risk. Finally defined action steps is to be taken in the event of risk occurring.

Risk management plan should document the risk, its probability, impact and the index of the risk expectation along with the method / procedure that will be used to manage the risk throughout the project. as per the Risk management plan template.

### **Risk Monitoring:**

Risk monitoring is a project tracking activity with three primary objectives (a) to access whether a predicted risk does, in fact occur, (b) to ensure the risk aversion steps defined for the risk are properly applied and (c) to collect information that can used for further risk analysis. Risks have to be tracked both in periodically and event driven reviews. The status and the priority of risks have to be accessed and updated in the plan. Also new, risk, which were not identified earlier, may surface and these have to be go through the steps of analysis and prioritization and updated in the plan.



## CHAPTER III

# SOFTWARE REQUIREMENT SPECIFICATIONS

---

### Functional Requirements

#### ➤ For Vehicle owner

The following features should be provided to the user.

#### 1. Opening of shutters of the various device on detecting a vehicle or person

The shutters of the various devices like the magnetic card reader, Coin input device; Bank Note input device should open on

- Detecting a person in case of the CPS
- Detecting a Vehicle in case of EPS.

On detecting a person or a vehicle the ticket input guidance message should be played.

#### 2. Automated Calculation and Display of fee.

The fee to be paid should be calculated and displayed automatically after the parking ticket is inserted into the system. This could be either an entry ticket or an exit ticket (if Current time > exit time).

#### 3. Automated Acceptance of fee

As soon as a customer inserts a valid entry ticket, the fee will be calculated and after a short period of time, the fee will appear on the screen. Payment can be done by Coins, Bank notes, prepaid card, discount card, count discount card, points card, service card or by the combination of all these techniques. During the course of the payment, the screen will be updated. The *fee paid* will increase with each coin or note inserted, the *still to pay* field will decrease at the same time. After the full payment, the ticket is printed as the Exit ticket. Receipt is also printed after the full payment.

### **3.1.Coins**

Payment can be made by different types of coins. Up to a maximum of eight different types (configurable) of coins are accepted by the system.

### **3.2.Bank Notes**

Payment can be made by bank notes. Up to a maximum of three different types of notes can be used in the system. The value associated with each type is configurable.

### **3.3.Prepaid Cards (Japanese)**

Payment can be made through a prepaid card, which has a value stored in them. As soon as a prepaid card is inserted to make a payment, the amount is deducted from the card and displayed on the screen. Maximum Prepaid Amount per transaction is configurable.

### **3.4.Discount Card (Japanese)**

Discount could be of four types

- **Flat:** Some fixed amount is deducted from the fees.
- **Percentage:** Some percentage of fees is discounted.
- **Time:** Some time is reduced.
- **Rate:** The rate at which the fee is calculated is lowered.

When a discount ticket is inserted in the system while making payment, the discount to be given should be calculated and displayed on the screen. The discount ticket is then swallowed. For one transaction the number of different discounts (Different Discount tickets) that can be given are configurable.

### **3.5.Count Discount Card**

This is similar to the discount ticket apart from that it contains the number of times the discount ticket can be used. Whenever a count discount is used to make payment the number of times is reduced from the card. Only when this number becomes zero, the ticket is swallowed. For one transaction the count discount ticket can be used only once.

### 3.6. Service Card or Points card

It is similar to a discount card, when a payment is made the entry ticket is converted to a service card. The service card contains points. Discount per point is configurable. Depending on the number of point's discount is given to the user.

## 4. Cancellation of Payment

The user can cancel the current transaction at any stage before the Completion. On cancellation, the following things should happen.

- Payment cancellation guidance message should be broadcast.
- Same denomination of Coins and notes should be returned to user. In case same denomination of notes are not available then same amount should be returned with different denomination.
- In Case the money required to be paid is not there then a credit note should be given to user.
- Discount information if any should be written on the entry ticket.
- Points used in the transaction (from the point card) should be converted to discount type (hours, Amount, Percentage etc) and written on the entry ticket.
- Amount paid by the prepaid card should be written on the entry ticket.
- If note dispenser is not present, then the amount to be returned should be paid by coins up to the maximum change per transaction (the maximum change per transaction should be a configurable item) and for rest of the amount a credit note should be given.
- Details of the payment made by cards in this transaction (before cancellation) should be printed on the receipt – after the payment is completed in any of the subsequent transactions.





## 5. Generation of Receipt

There should be a facility for providing receipts to user after the payment. The receipts should be optional and is done only on user request. There should be a provision to print the following types of receipts.

### Transaction Receipt

- **Normal operation:** A transaction receipt can be printed after successful transaction i.e. after making the complete payment for current transaction. In case of a CPS it should be possible to generate the receipt at any time by inserting the exit ticket into the system.
- **Cancellation of operation:** A transaction receipt can also be printed if the payment operation is cancelled.

**Note:** For compatibility with the old system the receipt should always be printed on the entry ticket also after the completion of payment. In case of the EPS it can be configured to give the entry ticket back as receipt to the customer. This functionality is not possible in case of the CPS, because the lag time reader swallows the exit ticket.

### Credit Note

A credit note is issued in the following situations

- **In case of balance not payable situation**

A credit note is issued in the case the amount required to be paid is not present in the PS

- **In case there is an upper limit on the change that can be dispensed from the particular unit**

There can be a configurable maximum limit to which the change can be returned by the CPS and the rest will be made into a CRN. E.g. 10,000 yen note inserted, fees is 1500 yen and the maximum

configuration is 1500 yen, then 1500 yen will be returned and the balance will be printed onto the CRN. Whether the CRN is to be printed or not in this case is also configurable.

#### **6. Generation of Exit Ticket or Opening of barrier gate on completion of payment**

- In case the PS is working as a CPS, On completion of the payment the PS should convert the parking ticket to an exit ticket (Parking ticket after the payment is completed).
- In case the PS is working as an EPS then it should open the barrier Gate.
- In Case of EPS the entry ticket can be converted to a service ticket depending on the configuration.

#### **7. Season Ticket Renewal**

The PS should provide for the renewal of the Japanese version of Season ticket.

For Japanese Version

- The PS should provide for the renewal of all 8 types of season tickets.
- For renewal the type of the ticket cannot be changed, same type can be renewed into same type.
- The season ticket can be renewed for 1, 3 or 6 months.
- Season Ticket Renewal is not possible at EPS.

Note: Season card from one parking lot can be used in the other parking lots also.

#### **8. Anti – Cheating Checks in user operations**

The following checks should be provided to prevent cheating.

## **8.1. For Entry Ticket**

### **8.1.1. Ghost Ticket**

If a user takes a ticket from the ticket dispenser and does not come on the gate coil (i.e.) does not enter the parking. Then make this ticket invalid. A database for all invalid tickets should be maintained. Before accepting the payment the ticket is checked against the invalid ticket database, if the inserted ticket is invalid then it should be rejected.

The Ghost ticket check is helpful if the user removes his car from the Ticket loop. Now if the user keeps waiting on the Ticket loop, the above mentioned check will fail. To prevent this kind of cheating the following methods should be used.

### **8.1.2. Minimum Time Check**

The pay – station should not allow any car to go out of the parking before X units of time, where X is configurable.

### **8.1.3. Serial Number Check**

The T/D wise serial number of the last ticket is maintained in the host/ Master PS and is updated only when the car passes the gate loop. If the ticket whose ticket number is greater than the serial number maintained by the host/master PS is inserted in the PS then it should be rejected.

## 9. Error Handling

### 9.1. Automatic Shift to Out of Operation in case of errors

The system goes to out of service due to one of the following

- Coin Box Full.
- Use side panel to put the system forcibly to out of order.
- Forcibly *Out of Order* by communication from the host.
- Due to Trouble
- Signal from the dry contact – if the host is not present then sequencer is used to control the operations.
- Change Exhausted.
- Time Setting to go Out of Service.
- Normal *Out of Operation* or *In Operation* from the *Host*
- The difference between the **forcibly out of operation** and this one is that in the forcible case, when a signal comes to go back to the other mode – it is discarded. The system was forcibly put *out of operation*, then you need to put it back **Forcibly In Operation**.

### 9.2. Recording of Error Information

When an error occurs the following steps should be taken

1. Error Number and generation time should be send to the parent machine in a fixed format. (Specs of the format are not present)
2. Error information and the number should be displayed on the screen.
3. Optionally, depending on the configuration, the error contents, error number and the generation time should be recorded on the journal printer.
4. Electronic file recording – not very clear.
5. The errors should be recorded in a log file, the error log should contain either X records or items for Y days (both configurable) whichever occurs first.



## 4.2 Administrative Requirements

### 10. Money Accounting

#### 10.1. Circulation coin hoppers

The following things should be maintained for each circulation hopper.

- **Presently held number of coins:** This gives the total number of coins contained in the circulation hopper.
- **Initial number of coins:** This gives the number of coins in the circulation hopper the time of putting the hoppers into the machine.
- **Inserted coins:** This gives the number of coins inserted in the hopper during transactions.
- **Dispensed coins:** This gives the total number of coins dispensed during transactions.
- **Refilled coins:** This gives the total number of refilled coins.

#### 10.2. BND note box contents

The following things should be maintained for Bank note box:

- Presently held number of Notes
- Initial number of Notes (at the time of putting the hoppers into the machine at any time)
- Dispensed Notes
- Refilled Notes

On removing any of these devices, totals move one level up and get cleared together with Receipt printing, journal printing as well as memory update occurring.

### **10.3. Coin Box Total / Subtotal**

- The system should provide for printing of Coin Box total and Coin Box Sub Total.
- When the coin box Sub total is printed, the current contents of the coin box are printed.
- When Coin Box total is printed, the current contents of the coin box are printed to paper, the coin box total is added to T-Total and the coin box total is cleared.
- Either pressing a button in the maintenance mode or whenever the coin box is removed from the system should print the Coin Box total.

### **10.4. Note Box Total / Subtotal**

- The system should provide for printing of Note Box total and Note Box Sub Total.
- When the Note Box Sub total is printed, the current contents of the Note Box are printed.
- When Note Box total is printed, the current contents of the Note box are printed to paper, the Note box total is added to T-Total and the Note box total is cleared.
- The Note Box total should be printed by either pressing a button in the maintenance mode or whenever the Note Box is removed from the system.

### **10.5. G – Total/ Sub Total**

- The system should provide for the printing of G -total and G – Subtotal. G- provides the total Sales since the last G- Total was printed.
- On Printing G- Subtotal this total is printed. On Printing the G –Total this total is printed on the paper, the Current G- Total is added to current M-Total and the G- Total is cleared.



### **10.6. M – Total / Sub Total**

- The system should provide for the printing of M -total and M –Subtotal.  
M- provides the total Sales since the last M- Total was printed.
- On Printing M- Subtotal this total is printed. On Printing the M –Total this total is printed on the paper and the current M- Total is cleared.
- The PS should also provide for the printing of previous M- Total

### **11. Maintenance and Diagnostics**

The system should provide the following maintenance and diagnostics operations.

**Magnetic reader Read/ Write Check**

**Ticket text Printing Check**

**Sensor Condition Check**

**Receipt / Journal Printer Check**

**TFT Check**

**Touch Panel Coordinate check**

**Hopper input and dispensing**

**Bank Note Validator Check**

**Note dispenser testing**

**Parking Inventory Control Parameters**

**Maximum number of cars in the parking**

**Changing Between In and Out of Operations**

**Gate Open/Close**



## **Changing Mode of Operations**

The PS machine can be run in the following special modes. The attendant can set the PS to run in any of these modes.

### **Automatic Mode**

When the machine is properly running with all the functionalities assigned to it with out any attendant to take care of it the it is assumes that the Pay Station is in Automatic mode which is running properly.

### **Semi – Automatic Mode**

This mode can be used when the speed of the payment is required to be made faster or in case of trouble related to bank note or coin unit is there. In this mode the fee is calculated automatically after inputting the parking ticket. The attendant then receives the payment and presses the payment complete button. On pressing this button the pay station completes the payment treating the money has been input.

In this payment mode it is possible to use discount ticket, prepaid card etc.

### **Manual Mode**

An attendant is present at the Pay Station in this mode. This normally comes into picture if the ticket reader device is not working properly or the user has lost his parking ticket. In this mode attendant enters the *in* time and the *rate type* from the side panel. After this the fee is displayed on the main screen and the side panel. The payment can then be made as in the normal case (Automatic mode) or in the semi automatic mode. It is also possible to use discount ticket, prepaid card etc in this mode (In case the ticket device is not working it is possible to enter the details of these cards from the side panel). This mode is allowed only on the side panel.

*The attendant in any of the above modes can keep the barrier gate open.*

## Tools and Technologies

This section speaks about the various tools and technologies used to give shape to this project. It encompasses the Hardware as well as the Software used to develop the system.

### Software Requirements

<b>Programming Languages</b>	C++, Win32 API, ActiveX controls
<b>Operating Systems</b>	Windows CE, Windows 2000
<b>Tools</b>	Embedded Visual C++, Visual Source Safe, Windows remote tools

### Hardware Requirements

The hardware required to run the software is embedded in the Pay Station of the client and cannot fully be disclosed. However, some of the main hardware components are enumerated below:

- H8 Board with an SH3 processor.
- Magnetic reader that can read and write the tickets.
- Thermal Printer that prints and issues the receipts etc.
- Coin Unit.
- Currency Note safe.
- TFT screen panel.
- Various Boards like IBK, IBW, IBC etc.

## CHAPTER IV

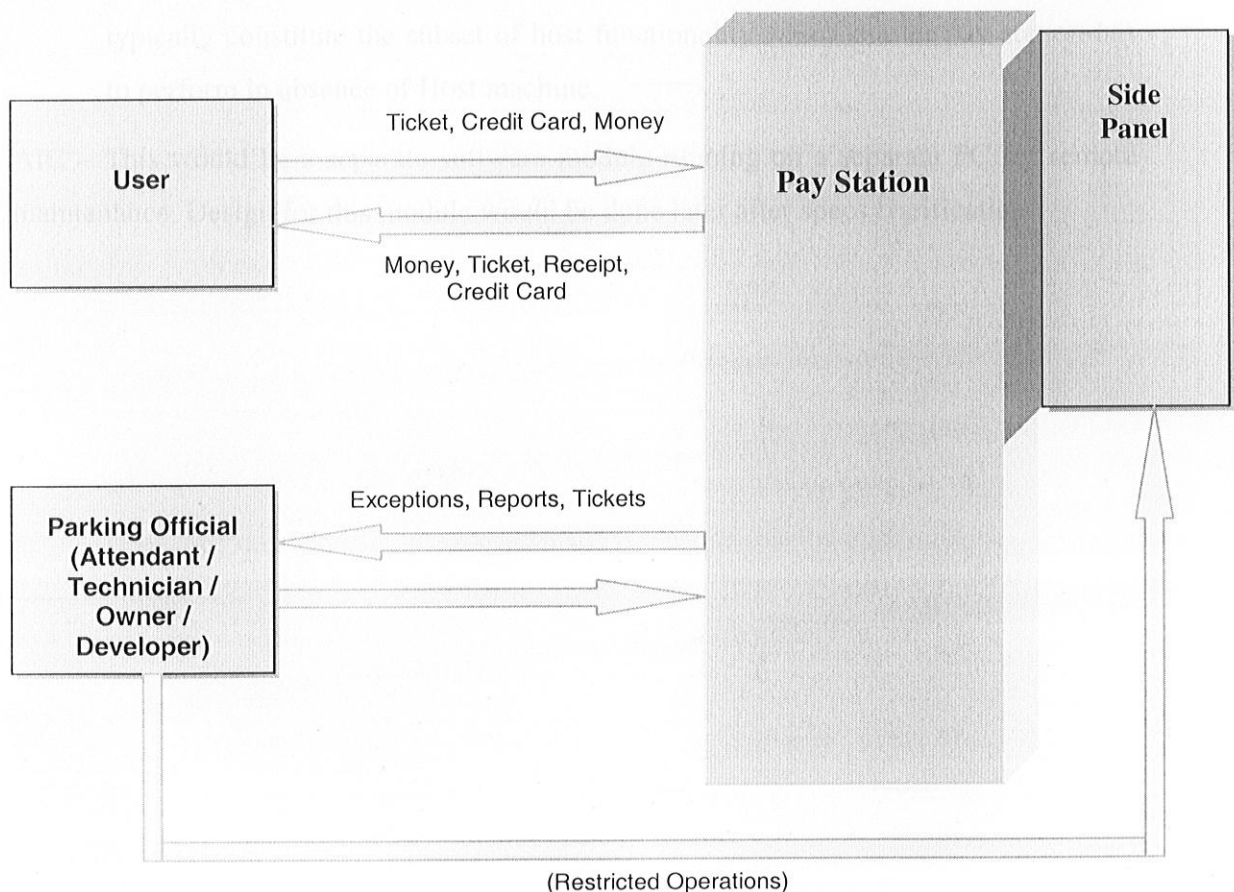
### CONCEPTUAL AND PHYSICAL DESIGN

#### Design Overview:

This is an Automatic pay Station located within a facility for easy access by parking patrons. Utilizing mag-stripe technology, it computes and displays the parking fee, accepts payment, issues change and provides a receipt without the aid of a cashier.

This pay station is to be developed on Windows CE for an SH3 processor based board.

The main entities interacting with the PS machine are the General User, and the parking Officials (the technician, owner/controller, attendant and the developer). This can be described with the help of the following diagram:





## **System Architecture:**

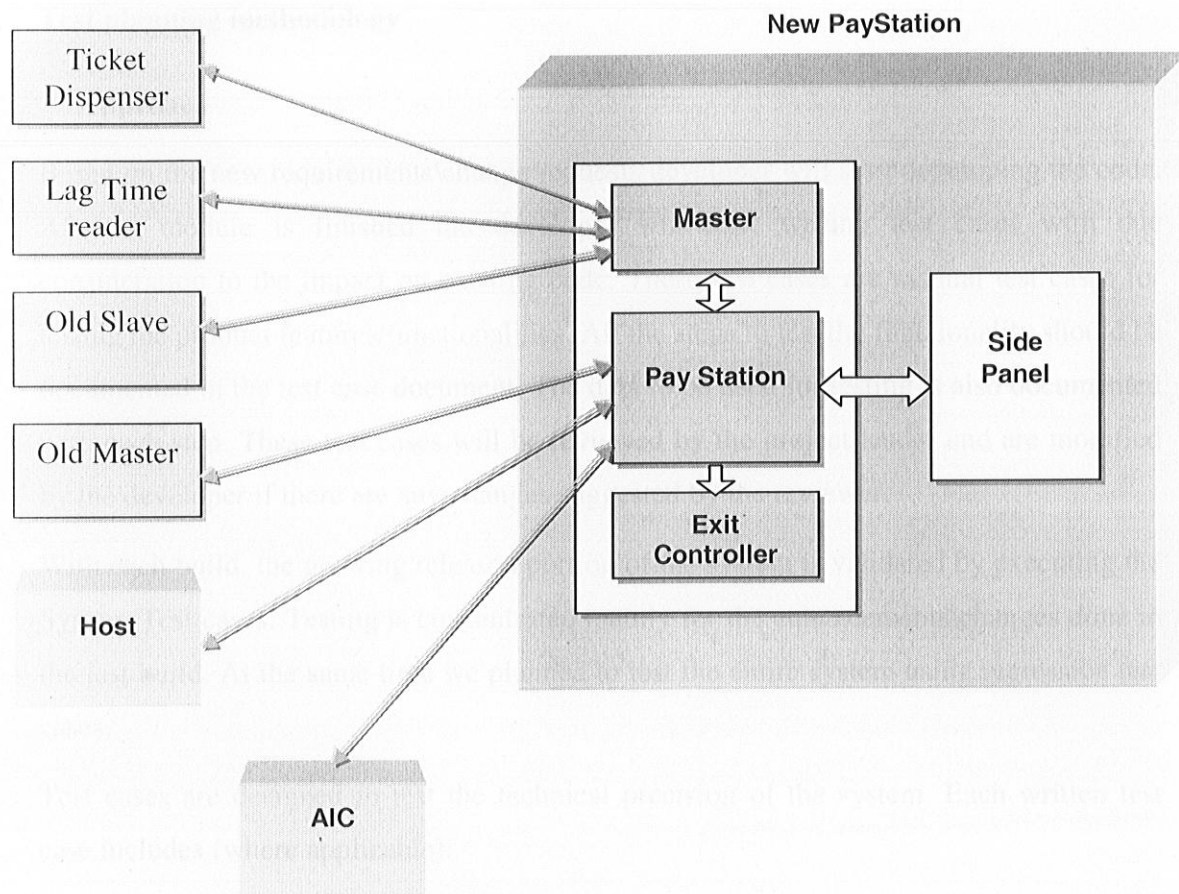
Overall system under our scope of development can be categorized into following major entities:

- Pay Station – This software module would be for the main SH3 board with Windows CE installed
- Side Panel – This software module would be for the additional SH3 board on side panel with Windows CE installed.
- Exit controller – This software module would be for the logical unit, which controls the exit mechanism from the pay station within parking facility. This would typically be present in EPS.
- Master – This module is again for the logical unit i.e. Master pay station. This will typically constitute the subset of host functionality which master pay station has to perform in absence of Host machine.

AIC – This would be a separate software module running on a separate PC for remote maintenance. Design for this module would be done later after specs clarification.

### Block Diagram for the System:

Here is the sub-system overview of the major entities identified in the System.



### **Test planning methodology**

#### **Approach**

Based on the new requirements\change requests developer will start developing the code. After a module is finished the developer will start writing test cases with due consideration to the impact on existing code. These test cases are manual test cases for testing the product features/functionalities. All the steps to test the functionality should be documented in the test case document. The data to be used for testing is also documented with each step. These test cases will be reviewed by the project leader and are modified by the developer if there are any changes suggested by the reviewer.

With each build, the growing released portion of the system is validated by executing the System Test cases. Testing is concentrated mainly for the enhancements\changes done in the last build. At the same time we planned to test the entire system using regression test cases.

Test cases are designed to test the technical precision of the system. Each written test case includes (where applicable):

- Test Case Description (or Test Procedure)
- Reference to Specification documents
- Test Data
- Expected Result
- Actual Result
- Status (Pass/Fail/NA)



## Test methodology

Test cases will be executed as soon as coding is completed and available for testing. Some times a particular functionality of a module may be incomplete or in some cases we don't have the required hardware to execute some specific test cases, then those test cases will not be executed at offshore but if possible must be executed by onsite team.

Test Methodology
Unit Testing
System Testing
Regression Testing

### Unit Testing

Testing performed to isolate and expose faults and failures as soon as the coding is completed. Each developer will do unit testing of the unit developed by him/her.

### System Testing

As soon as all modules are available and unit testing is completed, the integration of the source changes is done and the execution of full system testing begins. Tester will do system testing by executing all the test cases prepared against the requirements\change requests. It is required to follow the test steps and record the result (Pass/Fail) in the test case document against each test case. If any bug\issue found during system testing set Severity, in conformance with the guidelines for assigning severity.

### Regression Testing

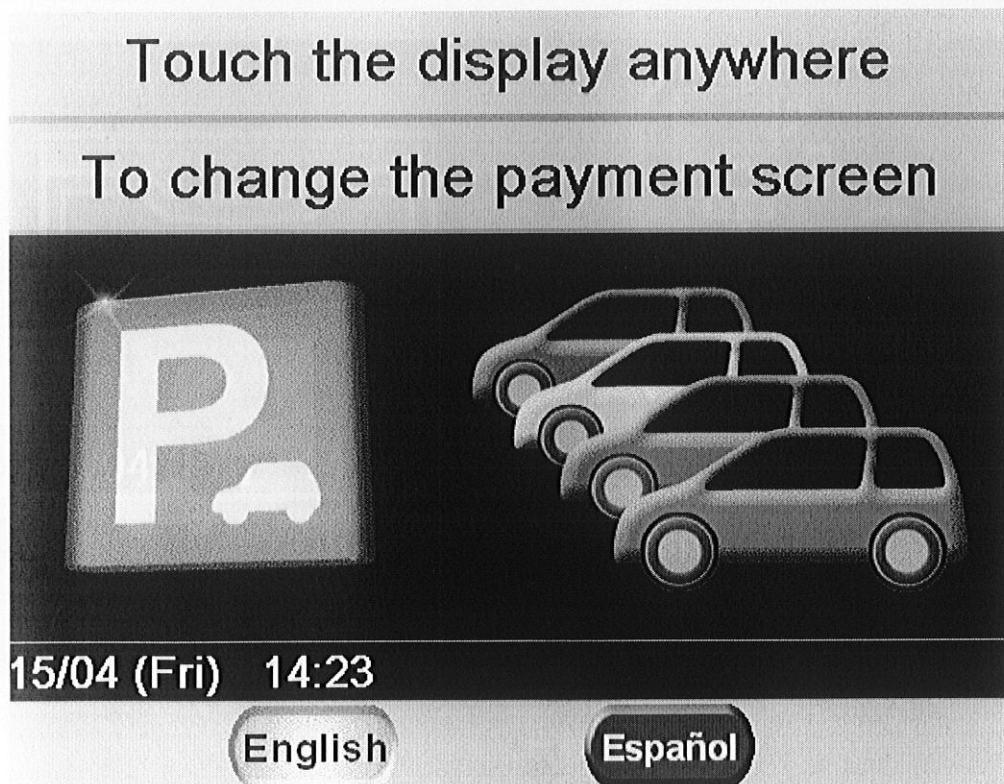
Testing with the selected test cases from the set of complete test cases, which are the minimum test cases required to test the overall functionality of the product is the regression testing. Regression testing is accomplished at each iteration and selected test cases will be performed on portions of the functionality to detect unexpected impact resulting from program modification and defect corrections.

**GUI Screens:**

As the project is on the embedded technology and the actual execution of the software takes place on the Pay Stations of the client, there is no window or snapshot of the screen that can be taken off the system. Although the bitmap images of the screens that show up on the touch-screen panel of the Pay Station are presented ahead.

When the Pay Station boots up, the following (animated) screen is shown:

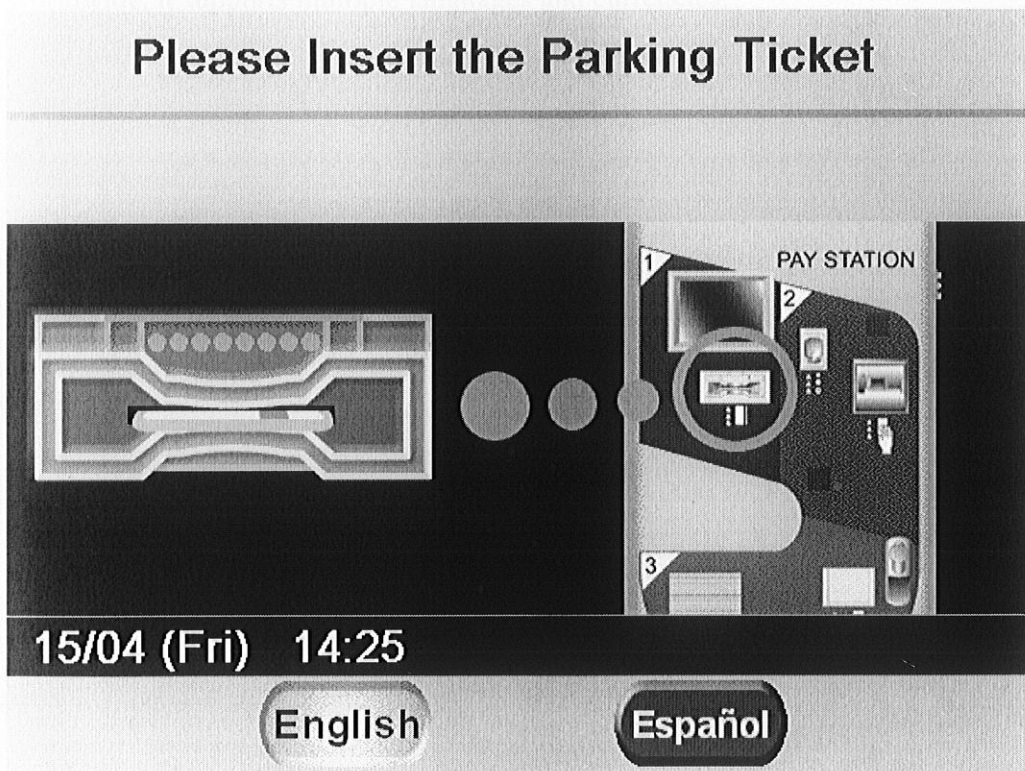
Idle Screen of the Pay Station



The above screen shows animated graphics asking the customer to touch the screen anywhere to change to the Payment Screen. As can be seen in the figure, there are two languages that can be displayed- **English** and **Spanish**. Many other languages can also be displayed such as **French, Deutsch, Italian, Japanese, Chinese, and Malay** etc. At any point, any two of these languages can be set for the Pay Station to display.

On touching the Touch Screen Panel anywhere, the following screen is shown that asks for inserting the Parking Ticket:

'Insert the Parking Ticket' screen





## CHAPTER VII

### CONCLUSION

---

The **Automated Parking Pay Station** that had been developed is fairly complex from programming point of view, but very powerful and easy for a user to use. This software provides a great deal of flexibility to the user. User can pay the parking fee by various means of payment. He can pay the parking fee by either cash or by using various types of cards. Apart from the user, this software provides facilities to the parking owner. No attendant is required at the Entry/Exit gate. Attendant is required only in case if there is some problem. The owner can maintain easily the record of every day transactions. As this software used worldwide, it supports multiple languages and currencies.

**Future Enhancement:**

At present, the software is employed to manage huge parking lots with multiple floors and lanes. The terminals (or Pay Stations) at different floors and lanes communicate with each other using TCP/IP protocol. The system is presently running on the client side and now all the enhancement is going on. When first the product is developed it is only for the Japan and therefore it support only Japanese currency and Japanese language. But after the further development it will support many languages and currency.

Since this software will be used by the various countries, so everytime a new requirement is generated according to the country in which it is used. The software can be enhanced even for a larger area and the various parking lots that are situated at diverse locations which can be managed from a single location. Internet can also be used for this purpose to make it cost-effective.

Presently the exit area of the parking station. Its functionality is similar to the CPS. The additional functionality provided is that it controls the opening of the barrier gate after payment is made for parking.

**Master PS (Master Pay Station)**

This could either be an FPS or CPS with additional functionality for doing parking management operations concerning complete parking lot.

• Status information view for all the slave devices

• Money collection related status view for all the Pay Stations

• Sending slaves report of operations etc.

**Slave PS (Slave Pay Station)**

This could either be an FPS or CPS which is monitored and maintained by the host side of the master PS.

## GLOSSARY

### Definitions and Acronyms

#### **Pay Station:**

The pay station should be able to work as any of the following depending on the current configuration.

#### **CPS (Central Pay Station)**

Present inside the parking station. This is the central place where all payment operations happen after presenting the parking ticket. The basic features provided to the user are:

- Generation of Exit tickets on payment
- Ticket Renewals
- Money Uploads onto tickets

#### **EPS (Exit Pay Station)**

Present at the exit area of the parking station. Its functionality is similar to the CPS. The additional functionality provided is that it controls the opening of the barrier gate after payment is made for parking.

#### **Master PS (Master Pay Station)**

This could either be an EPS or CPS with additional functionality for doing parking management operations concerning complete parking lot like:

- Status information view for all the slave devices
- Money collection related status view for all the Pay Stations
- Sending slaves in/out of operations etc

#### **Slave PS (Slave Pay Station)**

This could either be an EPS or CPS, which are monitored and maintained by the host machine or the master PS.



### **Lag Time Reader**

Present at the exit area of the parking station. It checks the validity of the exit ticket. The user has to present the exit ticket here while exiting the pay station. If a valid exit ticket is presented at the Lag time reader then the barrier gate is opened. Lag time reader is required in parking having CPS, in case of the EPS it is the responsibility of the EPS to open the barrier gate.

### **Ticket Dispenser**

Present at the entry area of the parking station. It dispenses entry tickets for parking of vehicles.

### **Barrier Gate**

Gate, which is used to let cars in and out of the parking.

### **Host Machine**

This is a central machine, which does the following operations:

- Maintains the business transaction status of all pay stations
- Status control (barrier gates, In/out of operations etc)
- Parking inventory management

The exact functionality of the host machine is given in the protocol document sent by client.

### **Sequencer**

This is a PLC circuit which provides signals through dry contacts .If the host is not present then sequencer is used to control certain operations. The basic operations supported are:

- Opening / Closing the barrier gate.
- Putting Pay stations *Out of Service*

## BIBLIOGRAPHY

Thinking in c++.....Bruce Eckel

Programming Windows .....Charles Patzold