FCT – Unesp

Documento de Requisitos

Sistema de Estacionamento “FCT_Estacionatron_3000”

Guilherme Eduardo Silva Batalhoti
5/10/2022

Sumário

Introdução

Propósito do Documento de Requisitos

Este documento tem o objetivo de apresentar os requisitos do sistema "FCT_Estacionatron_3000", detalhando e expandindo cada um. Então, serve de acordo entre os envolvidos: cliente/desenvolvedor. O público-alvo do sistema é o dono de um estacionamento de veículos que deseja automatizar seu estacionamento.

Escopo do Produto

O produto foi desenvolvido visando automatizar as operações de um estacionamento para evitar problemas comuns. O produto é capaz de manter registro das entradas e saídas dos clientes registrando as datas e horários de entrada e saída. Fazendo o cálculo de quanto deve ser cobrado do cliente.

Definições, Acrônimos e Abreviações

O objetivo desta sessão é apenas mostrar o significado de variáveis utilizadas no documento:

- $vagas_{totais}$ – Vagas totais do estacionamento;
- $vagas_{carros}$ – Vagas que são destinadas apenas à carros;
- $vagas_{motos}$ – Vagas que são destinadas apenas à motos;
- $disponíveis_{tipo\_veículo}$ – Vagas disponíveis por tipo de veículo;
- $vagas_{ocupadas\_veículo}$ – Vagas ocupadas por um tipo de veículo;

Visão Geral do Restante do Documento

Na sessão 2 há informações sobre a descrição geral do produto, elucidando suas funcionalidades, características do usuário, restrições e dependências do produto.

Na sessão 3, há um detalhamento dos requisitos descritos no índice 2, com informações essenciais para o desenvolvedor em fase de projeto e manutenção do *software*.

Descrição Geral

Perspectiva do Produto

O sistema “FCT_Estacionatron_3000” visa facilitar o dia a dia de um estacionamento. Além de manter registro sobre as entradas de veículos no estacionamento, também fazer os cálculos de tempo de permanência de veículo e valor de cobrança para a permanência do veículo.

Funcionalidades do Produto

O produto é capaz de registrar a entrada de um veículo no estacionamento, calcular o valor a ser pago pela permanência do veículo, e registrar a saída do veículo, além disso, o sistema mantém registro e controle das vagas disponíveis para cada tipo de veículo (carro ou moto).

Características do Usuário

O usuário que utilizará o sistema deve ter conhecimentos básicos de sistema operacional, como navegação pelo sistema, manejo de arquivos, abrir um programa e instalar um programa.

Restrições Gerais

O software deve ser projetado para execução em no sistema operacional Windows 10. O software deve basear-se na versão 8 do Java Runtime Environment (JRE). Analogamente, será necessário no mínimo 600MB de memória RAM livre apenas para o “FCT_Estacionatron_3000”, enquanto este estiver em execução.

Suposições e Dependências

É necessária a instalação do Java Runtime Environment (JRE) versão 8 em qualquer máquina que vise executar o “FCT_Estacionatron_3000”.

Requisitos Funcionais

RF1.1 – Registrar a data e hora da entrada de um veículo e registrar a data e hora da saída do mesmo veículo, registrando a placa e modelo do veículo, e nome completo, telefone de contato e CPF do motorista;

RF1.2 – Calcular a cobrança na hora da saída do veículo, o custo de permanência de até 4 horas dentro do estacionamento é de R$8,00 e cada 1 hora adicional custa R$1,00;

RF1.3 – O horário de abertura do estacionamento é às 6:00 e o de fechamento é às 18:00, com tolerâncias de 10 minutos a mais dos horários citados;

RF1.4 – Oferecer a opção de diária na hora da entrada com pagamento na entrada, ou seja, permanência de até 12 horas ou até o horário de fechamento (18:00), o custo da diária é de R$14,00;

RF1.5 – Manter registro do número de vagas totais para cada tipo de veículo (os veículos podem ser carros ou motos, as vagas de veículos são definidas como: $vagas_{totais} = vagas_{carros} + vagas_{motos}$);

RF1.5.1 – Vagas de carros são apenas para carros, e vagas de motos são apenas para motos, ou seja, não é possível fazer o intercâmbio de tipo de vagas;

RF1.6 – Manter registro do número de vagas ocupadas e o número de vagas disponíveis no momento, sendo o número de vagas disponíveis para qualquer tipo de veículo calculador por:

$dispon\acute{\imath}veis_{tipo\_veículo} = vagas_{totais\_veículo} - vagas_{ocupadas\_veículo}$;

RF1.6.1 – Notificar quando o número de vagas disponíveis de um determinado tipo de veículo for igual a 0, ou seja, quando não houver nenhuma vaga disponível;

RF1.6.2 – Não permitir a entrada de mais veículos de um determinado tipo se o número de vagas disponíveis do mesmo tipo de veículo for igual a 0;

RF1.7 – Notificar se um veículo não foi retirado no fechamento do estacionamento, mostrando os dados do veículo e hora da entrada, justamente da instrução de contatar o motorista;

RF1.8 – Calcular a taxa caso um veículo não seja retirado até o horário de fechamento do estacionamento, sendo cobrado para diárias e entradas normais;

RF1.8.1 – É cobrado a taxa do período de horas que o veículo permaneceu no estacionamento, acrescido de R$30,00 de taxa de não retirada;

RF1.9 – Gerar um ticket de identificação com os dados do veículo e do motorista, horário de entrada e com um código numérico, e enviar para a impressora (externo) para ser entregue ao motorista do veículo na entrada;

RF1.10 – Liberar a saída apenas mediante apresentação do ticket de identificação, validação do código numérico e pagamento do valor inteiriço calculado pelo sistema;

RF1.11 – Permitir a liberação manual do veículo apenas em caso de perda do ticket de identificação:

RF1.11.1 – A liberação deve ser feita apenas após o motorista preencher um formulário externo, com seus dados pessoais e dados do veículo;

RF1.11.2 – Cruzar os dados do motorista na hora a entrada e os informado na saída, e cruzar com os dados do veículo informado, a placa do veículo na saída deve ser a mesma da placa que o motorista entrou;

RF2.1 – O número de vagas para cada tipo de veículo pode ser modificado (aumentar ou diminuir) a qualquer momento, não podendo ser menor do que 0;

RF2.1.1 – A diminuição do número de vagas para qualquer tipo de veículo só pode ocorrer se o número de vagas ocupadas pelo tipo de veículo for maior ou igual ao de vagas totais para o tipo de veículo: $vagas_{totais\_veículo} \geq vagas_{ocupadas\_veículo}$;

RF3.1 – Tratar pagamentos em dinheiro: capturar a quantia recebida e informar o troco;

RF3.2 – Tratar pagamento com cartão de crédito: capturar a informação do cartão de crédito por um leitor de cartões e autorizar o pagamento com o serviço de autorização de crédito (externo) via conexão por modem;

RF3.3 – Tratar pagamento por PIX: capturar o comprovante de transferência do cliente (externo) e autorizar o pagamento via recebimento do comprovante de transferência com valor da taxa correto;

RF3.4 – Se não houver o pagamento inteiriço da taxa, não liberar o veículo.

Requisitos não funcionais

Por tratar-se de um *software* de aplicação acadêmica, não existe necessidade de especificar requisitos não funcionais.

Apêndices

- Todos os arquivos referentes ao projeto devem ser colocados no repositório do GitHub: https://github.com/Team-Dire/FCT_Estacionatron_3000
- O agendamento das tarefas será agendado e seu progresso deve ser registrado por meio de *Issues* e *Pull Requests*, no link: https://github.com/orgs/Team-Dire/projects/7
- O programador deve utilizar IntelliJ IDEA na produção de seus artefatos e códigos.
- O analista e o projetista devem utilizar Astah Community 7.2.0 na produção de seus artefatos. Utilizar o *template* para Java 8.
- Casos de uso, relatórios de SQA e demais arquivos textuais podem ser elaborados com a ferramenta de preferência do membro da equipe.