

Coffee Cart Rewards Management System

Process Plan for Team 22

1 Introduction

The Coffee Cart Rewards Management System will allow the owners of the LameDucks coffee shop to centrally manage a rewards system for their Customers across the numerous coffee carts they will have distributed around the city of Atlanta.

The goal of the system is to keep track of the total number of accumulated VIP points for every Customer. VIP points are earned through the purchases at any of the LameDucks coffee carts. Certain discounts will be automatically applied to any GOLD level Customer who has earned 5000 VIP points or more, or who has earned 500 points or more over the last 30 days.

The system will be constructed as an Android application that will run on a set of standardized devices that will be used by the Employees operating the coffee carts.

2 Process Description

Activity 1. Customer Request Documentation: Define software requirements in terms of the problem

- Input: T-Square Project Description.
- Output: Process Plan document summarizing the project scope
- Exit: Scope Approval

Activity 2. Specifications: Detailed description of the product.

- Input: Process Plan document from Activity 1.
- Output: Inception Phase Documents (Vision, Use Case Model, Risk Assessment)
- Exit: Specifications Approval

Activity 3. Initial Prototype: A quick model to evaluate feasibility of concept and objectives

- Input: Process Plan document from Activity 1 and Inception Phase Documents from Activity 2.
- Output: Non-Functional prototype. High level design for Development.
- Exit: Initial Design approval.

Activity 4. Additional Specifications: Detailed description of the product.

- Input: Prior Activities Output
- Output: Updated Documents for Elaboration Phase Documents (Use Case Model, Risk Assessment)
- Exit: Updated Documents approval and updates placed in version control.

Activity 5. Design: Detailed description of the program components and their interaction.

- Input: Latest Project Documents and Information from Piazza Q/A.
- Output: Create Software Architecture and incorporate the main requirements into the design of the Executable Prototype.
- Exit: Executable Prototype design is approved and specifications finalized.

Activity 6. Test Plan Development: Process for certifying that the product meets specifications.

- Input: Latest Project Documents and Executable Prototype.
- Output: List of Test Cases and directions in a Test Plan document.
- Exit: Approved Test Plan.

Activity 7. Coding: Writing and compiling programs.

- Input: Requirements Document and any New Information from Piazza Q/A.
- Output: Updated Documents for Construction Phase, along with the Initial Beta Version of the Android App and Testing Results.
- Exit: Android App contains most of the required functionality.

Activity 8. Testing: Verifying program logic and function.

- Input: Test Plan document.
- Output: Test results.
- Exit: All test have passed. Or return to Activity 7.

Activity 9. Documentation: User manual for the product.

- Input: Outputs from prior activities.
- Output: Final User Manual
- Exit: Final User Manual is approved.

3 Team

Team Members

- Tony Potter (@mistapotta)
- Murali Raju @muraligit)
- Omar Ramos (@orware)
- Chandra Sapkota (@cdsapkota)

List of Roles

Project Manager

The Project Manager leads and organizes the team so that everyone can work effectively and concentrate on their individual tasks.

Development Lead

The Development Lead is an experienced problem solver, particularly with the language needed to complete the project (Java). He/she will provide architectural direction on the best way to build the software so that it meets the requirements and help prevent team mates from running into coding roadblocks.

Documentation Lead

The Documentation Lead is an effective writer and communicator that can think like a user and create documentation that is easy to understand and makes our software easier to use.

QA Manager

The QA Manager leads the testing efforts and helps ensure that the software meets requirements and will work properly under normal use.

Developer

A Developer assists the Development Lead with the coding process.

Documentation

The Documentation role assists the Documentation Lead with the documentation process (basic documentation, needed code comments, etc.).

Tester

A Tester assists the QA Manager with the testing efforts and can report issues or contribute additional tests.

Role Assignments

<i>Team Member</i>	<i>Primary Role</i>	<i>Additional Roles</i>
<i>Tony Potter</i>	QA Manager	Documentation, Developer
<i>Murali Raju</i>	Development Lead	Documentation, Tester
<i>Omar Ramos</i>	Project Manager	Developer, Documentation, Tester
<i>Chandra Sapkota</i>	Documentation Lead	Developer, Tester

4 Estimates

<i>Team Member</i>	<i>Hours</i>	<i>Activities</i>
<i>Tony Potter</i>	70	1, 2, 4, 5, 6, 7, 8
<i>Murali Raju</i>	80	2, 3, 4, 5, 7
<i>Omar Ramos</i>	60	1, 2, 4, 7, 8, 9
<i>Chandra Sapkota</i>	60	2, 4, 7, 8

Lines of code:

Once complete, we expect there to be approximately 4800 lines of code and documentation (LOC) within the project (assuming each developer is generating 75 lines of code and documentation for approximately 4 days during each of the 4 weeks in the project period).

<i>50 LOC/day * 4 Developers * 16 Day Period</i>	3200 Total LOC
<i>75 LOC/day * 4 Developers * 16 Day Period</i>	4800 Total LOC
<i>100 LOC/day * 4 Developers * 16 Day Period</i>	6400 Total LOC

***LOC Estimate Reference:** CS-6300 Udacity P1L1 Videos.