Coupons Nao!

Project Management Plan

COP4331_001, Spring 2011

Team Name: The A-Team
Modification History:

VERSION	DATE	WHO	COMMENT
v0.0	02/05/11	Taylor Kourim	First draft of project management plan
v0.1	02/10/11	Daniel Kaplan	Filled out Risk Management
v0.2	2/14/11	Daniel Kaplan	Filled out Technical Progress Metrics and Plan for tracking, control, and reporting of progress
v0.3	2/15/11	Taylor Kourim	Added PERT chart, applicable standards

Team Members:

- Stephen Bryant - sbryant31@gmail.com
- Taylor Kourim - tkourim@knights.ucf.edu
- Daniel Kaplan - kaplan@knights.ucf.edu
- Isaiah Walker - senthose@gmail.com
- Chris Nergard - cnergard@gmail.com

Contents of this Document

Project Overview

The purpose of the project is to create an Android application that enables the user to find coupons and deals for businesses around their location. Users have the ability to add local businesses and coupons to our database, and the legitimacy will be verified by a confidence rating. When users use a coupon at a store, they can rate it up or down depending on their success.

Reference Documents

- Concept of Operations
- System Requirement Specifications
- Test Plan

Applicable Standards

- Coding Standard
  - Brackets
    - Opening bracket should be on line with method/loop/if statement, closing bracket should be on it's own line, example:

      ```
      for {
              …
      }
      ```
    - Single statement if statements and loops should have brackets
  - Comments
    - 1-2 line comments: use `//`
    - 3+ line comments: use `/* */`
    - Javadoc comments required for all methods
    - A good amount of commenting is required for each code to explain thought process and logic
  - Naming Convention
    - Constants: all uppercase
      - Example: `public static final int MAX_SIZE = 100;`
    - Variables: lower upper
      - Example: `String customerName;`
    - Classes: capitalize first letter
      - Example: `public class Coupon {`
    - Methods: lower upper
      - Example: `setCouponExpirationDate();`
- Document Standard
  - The style of all webpages is governed by an external style sheet, which can be found at http://couponsnao.sourceforge.net/styles.css.
  - Artifacts will mimic the style of the webpage with the exception of the background and table colors. All artifacts will be delivered as .PDF.
- Artifact Size Metric Standard
  - To ensure proper and quality documentation, each section of every artifact must contain at least three sentences detailing and explaining that part of the project, as long as that section is applicable. Lists are expected to be at least three lines long as well.

Project Team Organization

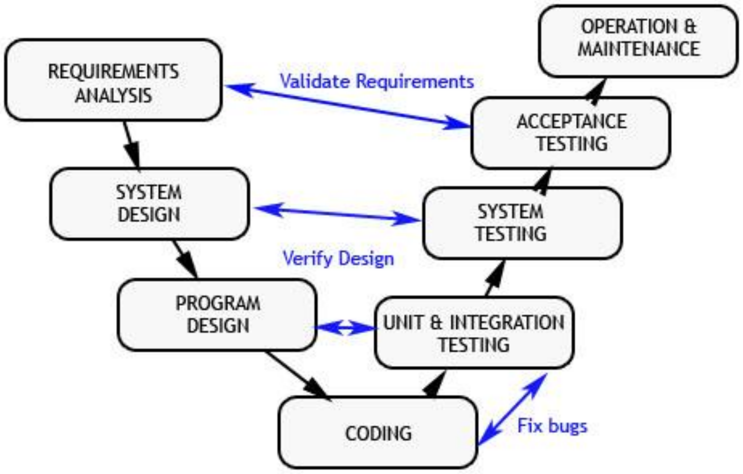
The team members that work on this project are listed at the top of the page. Taylor Kourim is the project manager and the webmaster, responsible for overseeing that everything is completed according to schedule. Daniel Kaplan is the lead developer, responsible for overall code and assigning who works on which piece of the code. Daniel is also a designer, along with Stephen Bryant. The two are in charge of developing the high-level design for the code along with all the accompanying UML diagrams. Chris Nergard and Isaiah Walker are the system testers, and will be primarily responsible for testing all the code to ensure it works properly.

The team will have weekly meetings where we all come together and discuss the tasks at hand and make decisions on the project. Team members who work together are responsible for finding times to meet up outside of team meetings to complete their tasks. Additional team meetings will be scheduled when needed. Outside of meetings, team members communicate through a private Facebook group.
The artifacts are to be completed by everyone. The project manager assigns team members to work on each artifact and is responsible for making sure they are all uploaded by the due date. The standard is that two members should work on each artifact.

Deliverables

ARTIFACT	DUE DATES
Meeting Minutes	Every Tuesday by 9:00pm
Concept of Operations	February 8, 2011
Project Management Plan	February 9, 2011
Software Requirement Specification	February 11, 2011
Test Plan	February 11, 2011
High-Level Design	March 1, 2011
Detailed Design	March 5, 2011
Final Test Results	April 15, 2011
User's Manual	April 18, 2011
Source, Executable, Build Instructions	April 18, 2011
Project Legacy	April 22, 2011
Individual Lessons Learned	April 25, 2011

Software Life Cycle Process



The process model our group has adopted is a slight adaptation of the V model. The black arrows represent the normal sequence of events, beginning with planning the system out in its entirety before coding, testing, and releasing. The blue arrows represent optional deviations from the black path. During each part of the process we may find the need to go back and either amend the requirements, fix a bug, or tweak the design of our system. Once we have used the black path to reach one step of the process, we can travel freely along the blue path connected to that node as we see fit. So for example, we can code our program, head into unit testing, find a bug, and then go back to coding to fix that mistake an infinite number of times. However, we cannot travel along the Validate Requirements path until we have taken the black path all the way to Acceptance Testing.

As is the case with all models, this model will likely not be perfect and it will be difficult to adhere to this model exactly, so some flexibility is implied. If critical issues arise during integration testing that stem from a problem in the system design, we likely will not follow this model exactly but will jump backwards to System Design and restart on the black path again. Therefore, this model is a soft requirement of the way we will develop our application.

Tools and Computing Environment

All team members are running Ubuntu, Windows Vista, or Windows 7. The main programming language used is Java SE 6. The IDE is Eclipse (Galileo or higher) with the Android plug-in installed. Libraries we will utilize are all Java standard libraries, Android standard libraries, and the Google Maps API.

Configuration Management

Version control and change control is handled through Subversion. Our project is hosted on Sourceforge.net. Windows users utilize TortoiseSVN to access the repository, and Sourceforge keeps a history of the revisions made. Stephen Bryant is the responsible administrator of the Sourceforge and the repository, although all team members have access and administrative powers.

Quality Assurance

Quality assurance is maintained through routine unit and system testing. Each team member is responsible for coding different parts of the system, and for doing their own unit testing on their work when they finish. The system testers are responsible for testing the pieces together and making sure they interact correctly. So while quality assurance rests on the shoulders of all team members, the system testers are ultimately responsible for assuring the quality of the program adheres to the standards outlined in the SRS. The test schedule is outlined in the Test Plan.

Risk Management

There is always a risk when we use SQL queries we will need to implement a system on the server and possibly android as well to verify the input that users give for forms. Another risk is someone brute forcing the database and gaining direct access to it. To minimize damage done by this type of attack we will hash all user passwords using a MD5 hashing algorithm. If possible we will also add a cap to how many login attempts are able to be made on the SQL server.

Table of Work Packages, Time Estimates, and Assignments

ACTIVITY	TIME ESTIMATE (IN DAYS)	ASSIGNMENT
Documentation Package A		
Concept of Operations	5	Stephen Bryant, Taylor Kourim
Project Management Plan	6	Daniel Kaplan, Taylor Kourim
Test Plan	7	Chris Nergard, Isaiah Walker
Software Requirements Specification	7	Stephen Bryant, Daniel Kaplan
Documentation Package B		
High Level Design	10	All team members
Detailed Design	8	Stephen Bryant, Daniel Kaplan
Code Package		
User Interface	5	Taylor Kourim, Isaiah Walker
Database	6	Stephen Bryant, Daniel Kaplan
Google Maps Integration	10	Daniel Kaplan, Chris Nergard
Backbone Structure	7	
Documentation Package C		
Test Results	5	Chris Nergard, Isaiah Walker
User's Manual	4	Stephen Bryant, Isaiah Walker
Source Code	1	Daniel Kaplan, Chris Nergard
Build Instructions	1	Chris Nergard, Isaiah Walker
Project Legacy	1	Taylor Kourim, Stephen Bryant
Final Presentation	1	All team members
Individual Lessons Learned	1	All team members

PERT Chart

ACTIVITY	PREDECESSOR	TIME ESTIMATES			EXPECTED TIME
		O	**M**	**P**	
High level design	Concept of Operations	2	5	7	4.83
Database design	High level design	2	4	6	4.00
Backend design	High level design	4	7	11	7.16
User Interface (UI)	Concept of Operations	3	5	8	5.16
Database Content	Database Design	1	3	4	2.83
Final Testing	Database Design, Database Content, UI, Backend design	3	4	7	3.66

Technical Progress Metrics

We will be tracking our progress using Mantis Bug Tracker. This program not only lets us keep track of problems that arise out of our coding but it also allows us to fill in our requirements for each phase of development.

We will fill in a all the requirements for each phase of development and assign each requirement to a different group member to be completed. This group member can break down the requirements even more and can ask for help if needed for the sub requirements.

Plan for tracking, control, and reporting of progress

Every week we have a team meeting to review our progress for the week.

We have a SVN set up through source forge to keep track of all versions of our project. This SVN lets all members have the newest version of our program as well as keeps track of changes done.

We are setting up a bug tracking software to keep track of our progress and any problems that arise as we develop the software.

We have set up a Facebook group for us to communicate with each other outside our group meetings. This Facebook group lets us post information that might be relevant to our project and also gives us the ability to have group chats with all members online.