

LAMEDUCKS COFEEE SOFTWARE ARCHITECTURE DOCUMENT – TEAM 007

INTRODUCTION

The system is being developed for managing VIP card program for LameDucks Coffee Atlanta, GA. The system will be developed as an android application that will enable cart managers to access the VIP program details from their android smart phones. The system will enable users to order and pre-order from their smart phones. Owners will also have the option to generate reports as required.

SOFTWARE ARCHITECTURE

The system will be developed as an Android application. Users will be able to load the application via the play store to install on their phones. The following are the system requirements.

- a. Java 1.6+
- b. Android 4.0+
- c. External Datasource – Parse.

All users will access the system via the Internet. The connection to the system will be using the HTTP protocol. Data is stored in the Parse datastore as a JSON string. There are no specific login or authentication access control requirements for the system. The Coffee Cart Reward System (CCRS) has the following components.

- a. Account Management – This module will provide features for the cart managers to add/edit and delete VIP customer information.
- b. Product Management – This module will allow cart managers to manage products that are sold by LameDucks Coffee
- c. Coffee Cart Management – This module will allow owners to add new cart locations.
- d. Purchase and Pre-order system – This module will allow users to purchase and order products either for pickup instantly or at a later date via CCRS.
- e. Reporting – This module will allow managers to generate reports for products sold.

Below is the software architecture diagram for the system.

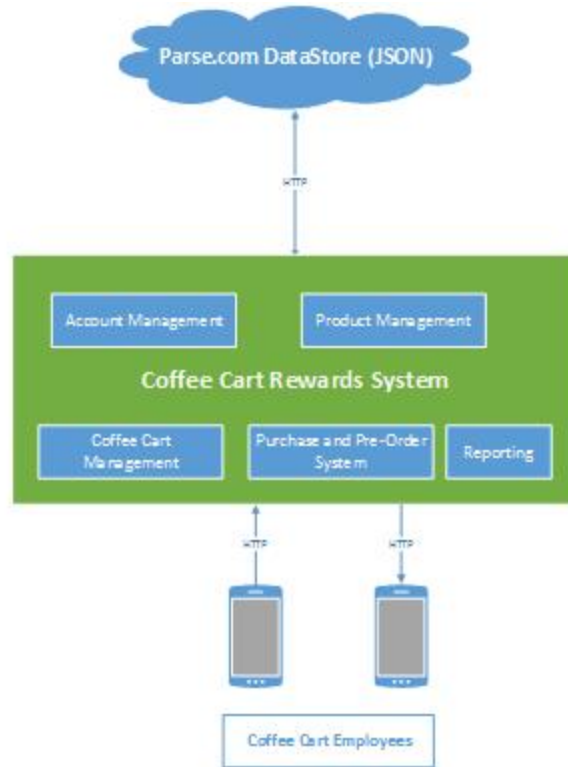


Fig 1: Software Architecture diagram

External Data Source (Parse)

The system will use Parse as it's cloud-based backend service. This allows the developers to focus on building the CCRS and leaving the database management to Parse. Parse allows the system to store any JSON objects into MongoDB database. Parse periodically backups their database from their slave instance (which is periodically updated from its master), thereby reducing the chances of losing data.

DEVELOPMENT PROCESS

The development will follow the Rational Unified Process. In the early stages, requirement and use case documents will be used for planning and designing the system. These documents will be gradually refined throughout the process. For the system implementation, the development of the application will be done on Github using feature branches and pull requests for code reviews.

PERFORMANCE

The CCRS will allow multiple users to use the system simultaneously and guarantee data correctness for ordering and user management.

The system will only store the minimal fields required and calculate any values that can be derived (such as VIP Status/points etc..) in real-time. This will ensure data does not change with changes in business rules. Data de-normalization can be done to speed up the process in future when we experience database bottlenecks due to increased growth/usage of the app.

CONSTRAINTS

Users of the software are required to have Internet access. In case of connectivity failure, the system will make a best effort to retain data when the connection is back online but data persistence on such cases cannot be guaranteed.

COMPATIBILITY

The system will be compatible with newer versions of the platform with minimal changes (if any feature provided by Android that are used by the application changes). The system shall also be backward compatible with most recent versions of android.

RECORDS RETENTION

Account management system within CCRS allows managers to add/edit and delete users. Records will not be retained in the data store if users are deleted from the system. All orders linked with the users will also be deleted. Changes will be stored locally before syncing with the Parse.com data store.