CSE 5324
Software Engineering: Analysis, Design, and Testing

Inception of
Smart Checkout Android Application

Developed by
Gupta, Yesha Pawankumar
Shah, Nishit Haresh
And
Alankar Raju, Ranjan

Under Supervision of
Prof. Christoph Csallner
SMART CHECKOUT ANDROID APPLICATION

GOAL:

Our goal is to expedite the checkout procedure across the campus libraries. This is intended to be done by making use of camera enabled android compatible phone.

CONSTRAINT:

1. Users (UTA maverick/Student) should have a valid net id and password.
2. Device should be camera enabled and compatible to run android application.
3. Device should have access to wireless internet.

EXECUTIVE SUMMARY:

Vision is to help UTA-Mavericks (UTA Students) experience quick and easy way to check out items (books) from library. This can be accomplished by scanning the barcode from their phone and validating themselves to check out books. There exist APIs’ which scan a barcode from mobile’s camera and display its metadata from internet (Amazon or Google). Smart Checkout Android Application will be build upon them. Along with displaying the Meta data from internet, application will fetch data from local database which will be similar to existing UTA Library Database. User’s net id and password are validated to check out a book from library. When user scans a book, he is given options to reserve the book, checkout the book or continue scanning more books. This way, user will not have to wait for a long time in queue to check out and reserve books. On succesfull checkout of books the system will also send a message to the tracking device allowing free exit to the user.

REQUIREMENTS:

2. Device-Database connectivity, Device-Internet connectivity.
4. Scan option which will invoke the camera, capture barcode and display item detail.
5. A check out option for library.
6. Once the item is checked out, tracking system is updated.
7. A reserve option to reserve a book in library.
8. Add to cart option, for multiple books.
USE CASE MODEL:

Login: Login refers to connecting to internet and validation of user’s net id and password. This log in will connect to database (Library). User is prompted to enter the log in information.

Scan: Scan houses of capturing the bar code information from the item and retrieving data from the database. This requires file locking mechanism to avoid conflict between multiple users. Scan function is basically the initial step for browsing items.

Check out: Check out will update the database and unlock the files if necessary. It will update the tracking system.

Exit: In this module the user will have to log out of the application.

SUPPLIMENTARY SPECIFICATION:

There are few non-functional requirements.

1. User account should be active (i.e. Valid net id and password).
2. User should have camera enabled phone with ability to connect to a network.
4. Any book in user’s cart will be locked (reserved) by that user.

GLOSSARY:

List of Keywords used throughout the project.
1. Scan: Scan is an action that will start the camera and wait for user to point on barcode. Furthermore, it will retrieve useful information from database.

2. Reserve: This will update the database and makes the book reservation for a particular user.

3. Add to cart: This will temporarily reserve books in user’s account.

RISK LIST:

1. Speed might decrease rapidly as number of users grows and database size increases (Local Database can go out of bounds because for mobile applications database tends to be of relatively small sizes). This can be resolved by implementing a network database.

2. It might fail to support concurrent transactions. However, this can be resolved in future.

ITERATION:

The project is divided into 3 iterations using Iterative Model. Below is the list of modules to be built in each iterative cycle.

1. Scanning barcode of an item, database connectivity and displaying book’s information from the local database.

2. Authenticating user, reserve books, add to cart and checkout operations on books.

3. Integration testing, J unit test cases, validation and null checks.