Pentru proiectul final am ales aplicatia Virtual cards. Este o aplicatie de mobil care permite transferul cardurilor de loialitate pe telefon. Sigur le stiti, gen Catena, Sensiblue, Ecco. Genul de carduri care iti umplu portofelul, le folosesti rar iar atunci cand ai nevoie de ele… ia-le de unde nu-s. Stiti ce are intotdeauna o femeie in geanta?

Cum se foloseste Virtual cards?

Se downloadeaza din google play sau app store. Apoi cardurile pot fi scanate si salvate in aplicatie. La alegere, se poate introduce codul manual. Exista doua feluri de carduri pentru aceasta aplicatie: cardurile fizice, de plastic care se salveaza scanandu-le sau introducand manual codul de bare si carduri virtuale, in lipsa celor fizice. Cardurile virtuale se genereaza completand un formular. Se pot adauga oricate carduri. Acestea se prezinta la casa de marcat si se va aplica discountul.

Utilizatorul poate seta daca va primi notificari si de la ce comercianti.

Deasemenea, utilizatorul poate folosi optiunea display route pentru a ajunge la magazinul dorit.

Aplicatia ofera posibilitatea crearii de liste de cumparaturi. Se pot crea liste pentru fiecare comerciant in parte sau o lista mare, globala. Lista de cumparaturi poate fi share-uita pe diferite canale. Messenger, sms, whatsapp.

Obiectivul pe care mi l-am pus cand m-am apucat de testare a fost sa gasesc toate bug-urile. Problema mea e ca si altii au avut acelasi tel si o buna parte din buguri s-au fixat inainte ca eu sa le raportez in Mantis.

Strategia pe care mi-am facut-o a fost sa explorez aplicatia, sa ajung sa cunosc in detaliu ce ofera.

Am luat fiecare functionalitate pe rand, am facut un mind map cu structura aplicatiei pe care o s ail vedeti in slide-urile viitoare.

Apoi am scris in xl un set de date pentru fiecare input (date valide, invalide, nule).

Am scris suita de smoke test cu principalele functionalitati apoi am rulat-o.

Am gasit cateva buguri pe care le-am raportat in Mantis ai am incheiat procesul cu un test report.

M-am folosit de urmatoarele tool-uri:

Test link pentru creat si rulat test case-uri.

Mantis pt raportat buguri, mindmp-urile cu xmind.

Am folosit aplicatia QR Code & Barcode Scanner pentru a verifica daca codurile cardurilor scanate sunt valide.

Screenshoturile le-am editat cu Photoscape.

Pe langa tipurile clasice de testare (Smoke, exploratory, negative) am folosit urmatoarele:

Connectivity testing: prin care am observant cum se comporta aplicatia in offline. Cardurile pot fi citite fara internet, dar alte actiuni ar trebui sa genereze niste mesaje de eroare.

Notification testing: am incercat sa vad daca aplicatia tine cont de dorinta utilizatorului, am ales sa primesc notificari, sa nu primesc, sa primesc doar de la anumiti comercianti…

Localisation testing- aplicatia permite aleagea mai multor limbi si am incercat sa vad daca toate paginile, butoanele, mesajele sunt traduse si ca arata bine layout-ul.

Location testing- folosit la optiune display route.

O alta metoda noua pentru mine a fost interruption testing. Se pare ca developerii trebuie sa tine cont de f multe aspect cand creaza o aplicatie de mobil. (Cum se comporta aplicatia atunci cand suna telefonul, cand utilizatorul primeste un mesaj, cand ramane fara baterie…)

Aplicatia permite interactiunea cu alte aplicatii gen Facebook, whasapp. . de ex log in via FB, send shopping list via whatsapp.

Aici este mindmapul despre care vorbeam la testing approach. Am luat fiecare functionalitate si am testat fiecare bucata. In suita de smoke test au rezultat 19 test case-uri in care am incercat sa gasesc pasii pe care un user obisnuit le-ar face in primul rand. Install, add card, create account / log in, remove card, Search si uninstall. Dintre aceste test-case-uri, 94.7% au trecut, unul a avut rezultat failed (se refera la mandatory fields de la formularele de add card. )

In urma tuturor tipurilor de testare am identificat 13 buguri, unul major, restul normale ca severitate, distribute cat de cat uniform. Cele mai multe le-am descoprit prin connectivity testing si location testing (mesaje netraduse si comportament nepus la punct in offline)

Concluzia la care am ajuns este ca Virtual cards este o aplicatie faina, stabila, folositoare, user friendly, face ce promite. Dovada sunt reting-urile de pe app store si google play.

Iar lectiile pe care le-am invatat au fost ca testatul nu este usor, si este vital pentru a livra un produs de calitate. Pe multa lume am auzit ca trebuie doar sa dai niste clickuri… nu e chiar asa.. trebuie sa stii ce clickuri sa dai astfel incat sa acoperi mare parte de aplicatie cu cat mai putine clickuri. A fost nevoie de rabdare si creativiate, sis a te pui in pielea unui utilizator obisnuit.

•        Select an application at your choice

•        Think on how would you test it, what testing types are applicable

•        Test the application

•        Write the smoke test suite in TestLink, under your dedicated folder under Final Project S18

•        Run the tests in TestLink

•        Report the bugs under Mantis, project Final Project S18

•        Create a test data set in an Excel file

•        Create a test report using the provided template

•        Create a presentation (10 minutes) containing the following info: app description (1 slide), how do did you approached testing (1 slide), tools used (1 slide), testing types covered (1 slide), test cases overview (1 slide), bugs overview (1 slide), test cases results (1 slide), conclusions/test report (1 slide) and lessons learnt (1 slide)

•        Deliverables: Test Cases (TestLink – Your Name Project), Test Data Set (Excel), Test Report(.doc), presentation (.ppt), Mantis bugs

* ar trebui sa ai cel putin 15-20 de seturi de date (carduri)
* ca si test data ai putea sa incerci diverse nume, coduri de bare corecte sau eronate, search in lista, diverse logouri (poze) in diverse formate, sterge carduri, adaugare card existent
* am vazut ca se poate crea si ceva gen shopping list
* oferte de la comercianti - aici nu poti controla tu ce primesti dar ar merita investigat daca sunt ceva oferte constante pentru a putea testa
* avantajul legat de buguri este ca daca intri pe playstore o sa si vezi unele deja raportate de utilizatori :-)
  + Test Techniques: Manual, dynamic, black-box, functional testing:

Exploratory testing, Component testing, System testing, Usability Testing, interruption, boundary

Test cases Virtual Cards

Test on different devices and OS

**Usability tests verify that the system is easy to use and that the user**

interface appearance is appealing.

Data Integrity Testing**: It verifies that data are being stored by the system in a manner**

where the data is not comprised by updating, restoration, or retrieve processing.Validation to be performed can include checking data fields for alphabetic and numericcharacters, for information that is too long, and for correct date format.

**. For Type Ahead Option**

If your UI contains drop down lists, check whether “type ahead” is included or not. Let’s take an example, there are more than 100 names in the list like list of countries. The drop down list should include “type ahead” function to facilitate selection of the desired name.

Calendar allows you to add birthday in the future –ex: belladonna card

The behaviour of the “Your card is not on the list? Add card” option from the bottom of the list of all available cards is different from the behaviour of the same option from the bottom of added cards. The first gives user the possibility to add a private card, the second leads the user to tha list of all available cards.

* **Different range of mobile devices** with different screen sizes and hardware configurations like a hard keypad, virtual keypad (touch screen) and trackball etc.

**types of testing are performed on Mobile applications.**

* [**Usability testing**](http://www.softwaretestinghelp.com/usability-testing-guide/)– To make sure that the mobile app is easy to use and provides a satisfactory user experience to the customers
* **Compatibility testing**– Testing of the application in different mobiles devices, browsers, screen sizes and OS versions according to the requirements.
* **Interface testing**– Testing of menu options, buttons, bookmarks, history, settings, and navigation flow of the application.
* **Services testing**– Testing the services of the application online and offline.
* **Low-level resource testing**: Testing of memory usage, auto-deletion of temporary files, local database growing issues known as low-level resource testing.
* [**Performance testing**](http://www.softwaretestinghelp.com/introduction-to-performance-testing-loadrunner-training-tutorial-part-1/)– Testing the performance of the application by changing the connection from 2G, 3G to WIFI, sharing the documents, battery consumption, etc.
* **Operational testing**– Testing of backups and recovery plan if a battery goes down, or data loss while upgrading the application from a store.
* [**Installation tests**](http://www.softwaretestinghelp.com/software-installationuninstallation-testing/)**–** Validation of the application by installing /uninstalling it on the devices.
* [**Security Testing**](http://www.softwaretestinghelp.com/category/security-testing/)– Testing an application to validate if the information system protects data or not.

**Test Cases for Testing a Mobile App**

In addition to functionality based test cases, Mobile application testing requires special test cases which should cover following scenarios.

* **Battery usage**– It’s important to keep a track of battery consumption while running application on the mobile devices.
* **The speed of the application-**the response time on different devices, with different memory parameters, with different network types etc.
* **Data requirements**– For installation as well as to verify if the user with the limited data plan will able to download it.
* **Memory requirement**– again, to download, install and run
* **The functionality of the application**– make sure application is not crashing due to network failure or anything else.

### Conclusion

Designing the right test strategy, choosing the right mobile simulators, devices and mobile testing tools can make sure that we have 100% test coverage and help us include security, usability, performance, functionality and compatibility based tests into our test suites.

Sorting cards by last/frecqvent usage

1. To validate that the application resumes at the last operation in case of a hard reboot or system crash.

the application provides an available user guide for those who are not familiar to the app

BUGS

Log in via FB while offline error message missing

Nu ma pot conecta prin contul de Facebook.Pe net mi spune "eroare la server" când vreau sa modific parola.

Utila ?i merge cu condi?ia sa ai ecranul curat oglinda ???? fara pete de amprente, grasime ?i alte minuna?ii?? altfel nu-l cite?te scannerul ??

O aplicatie care probabil imi face e economie pe luna de cel putin 50 ron

linkul din notificari duce exact la fereastra pt obtinerea cardului pt care este notificarea

toate detaliile sunt memorate in form, mai putin ce e in drop down

notifications unwanted